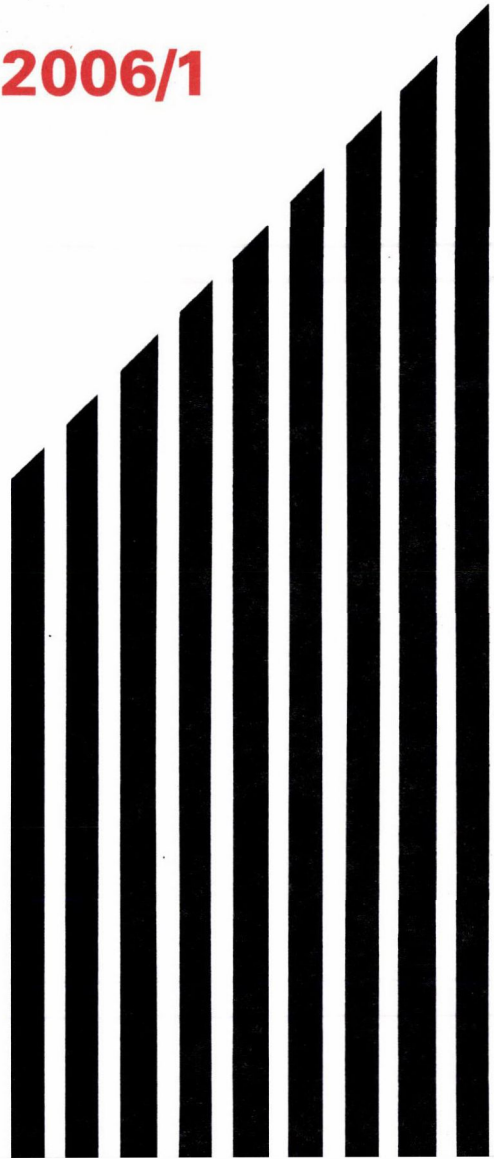
Alkalmazott matematikai lapok

2006/1



A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI TUDOMÁNYOK
OSZTÁLYÁNAK KÖZLEMÉNYEI

23. KÖTET

ALKALMAZOTT MATEMATIKAI LAPOK

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

ALAPÍTOTTÁK
KALMÁR LÁSZLÓ, TANDORI KÁROLY, PRÉKOPA ANDRÁS, ARATÓ MÁTYÁS

FŐSZERKESZTŐ
PÁLES ZSOLT

FŐSZERKESZTŐ-HELYETTESEK
BENCZÚR ANDRÁS, SZÁNTAI TAMÁS

FELELŐS SZERKESZTŐ
VIZVÁRI BÉLA

TECHNIKAI SZERKESZTŐ
KOVÁCS GERGELY

23. kötet

Szerkesztőség és kiadóhivatal: 1027 Budapest, Fő u. 68.

Az Alkalmazott Matematikai Lapok változó terjedelmű füzetekben jelenik meg, és olyan eredeti tudományos cikkeket publikál, amelyek a gyakorlatban, vagy más tudományokban közvetlenül felhasználható új matematikai eredményt tartalmaznak, illetve már ismert, de színvonalas matematikai apparátus újszerű és jelentős alkalmazását mutatják be. A folyóirat közöl cikk formájában megírt, új tudományos eredménynek számító programokat, és olyan, külföldi folyóiratban már publikált dolgozatokat, amelyek magyar nyelven történő megjelentetése elősegítheti az elért eredmények minél előbbi, széles körű hazai felhasználását. A szerkesztőbizottság bizonyos időnként lehetővé kívánja tenni, hogy a legjobb cikkek nemzetközi folyóiratok különszámaként angol nyelven is megjelenhessenek.

A folyóirat feladata a Magyar Tudományos Akadémia III. (Matematikai) Osztályának munkájára vonatkozó közlemények, könyvismertetések stb. publikálása is.

A kéziratok a főszerkesztőhöz, vagy a szerkesztőbizottság bármely tagjához beküldhetők. A főszerkesztő címe:

Páles Zsolt, főszerkesztő
1027 Budapest, Fő u. 68.
A folyóirat e-mail címe: aml@math.elte.hu

Közlésre el nem fogadott kéziratokat a szerkesztőség lehetőleg visszajuttat a szerzőhöz, de a beküldött kéziratok megőrzéséért vagy továbbításáért felelősséget nem vállal.

Az Alkalmazott Matematikai Lapok előfizetési ára kötetenként 850 forint. Megrendelések a szerkesztőség címén lehetségesek.

A Magyar Tudományos Akadémia III. (Matematikai) Osztálya a következő idegen nyelvű folyóiratokat adja ki:

1. Acta Mathematica Hungarica,
2. Studia Scientiarum Mathematicarum Hungarica.

Alkalmazott Matematikai Lapok **23** (2006), 1–4.

MARTOS BÉLA OPTIMALIZÁLÁSELMÉLETI MUNKÁSSÁGÁNAK MÉLTATÁSA AZ EGERVÁRY-EMLÉKPLAKETT ÁTADÁSA ALKALMÁBÓL

RAPCSÁK TAMÁS

Budapest

Martos Béla 1920-ban született Budapesten. Miután kitanulta az elektroműszerész szakmát; a háború után tanári oklevelet szerzett az *Eötvös Loránd Tudományegyetem* matematika-fizika szakán. Pályájának kezdeti szakaszát igencsak a változatosság jellemezte. Volt elektroműszerész, gépi hurkoló és társadalombiztosítási tisztviselő, majd 1949–1962 között osztályvezető a *Központi Statisztikai Hivatal*ban, az *Országos Tervhivatal*ban és főmérnök a *Kohó- és Gépipari Minisztérium*ban, miközben szakcikkeket írt munkaügyi kérdésekről a *Statisztikai Szemlé*be, a *Munkaügyi Szemlé*be és a *Közgazdasági Szemlé*be. 1962-tól kezdődően az *MTA Közgazdaságtudományi Intézet*ében dolgozott. Húsz éven át (1968–88) főszerkesztője volt a *Szigma* matematikai-közgazdasági folyóiratnak. 1990 óta nyugállományban van.

Martos Béla szigorú értelemben vett tudományos pályafutása az ötvenes évek vége felé vette kezdetét, és ez a kezdet, noha *Martos Béla* akkor már közel 40 éves volt, ragyogó sikerekkel indult. Első olyan eredményét, amelyre a nemzetközi tudományos világ is felfigyelt, az optimalizáláselmélet területén érte el, amihez a közgazdaságtan úgynevezett optimalizáló modelljei is kapcsolódnak. Abban az időben a kutatások fő iránya a legegyszerűbb modelltípus, a lineáris optimalizálási feladat vizsgálata volt, aminek a megoldására szolgáló szimplex algoritmus ismerete ma már a közgazdász alapműveltség része.

Martos Béla észrevette, hogy az a feladatosztály, ahol két lineáris függvény hányadosának a szélsőértékeit kell meghatározni, sok lényeges tulajdonságban megegyezik a lineáris feladatokkal. Ennek alapján, a világon elsőként oldotta meg a lineáris törtprogramozási, vagy az általa bevezetett terminológiát használva, a hiperbolikus programozási feladatot, megelőzve az ugyanezzel foglalkozó amerikai és német matematikusokat.

Martos Béla visszaemlékezése szerint a hiperbolikus optimalizálás létrejöttének a következő a háttere:

1958 táján a *Nehézipari Minisztérium* megbízta az *MTA Kibernetikai Kutatócsoportját* (az *MTA SZTAKI* ősét) a magyar bauxit-alumíniumipar négy vertikuma (bauxitbányászat, timföldgyártás, alumíniumkohászat, félkész termékek) arányainak és külgazdasági összefüggéseinek a vizsgálatával. Abban az időben már működött a szovjet kooperáció, amelynek keretében a hazai timföldet a *Szovjetunió*ban kohósították, és az alumíniumtömböt visszaszállították.

A kutatócsoportot *Martos Béla* vezette, közgazdászok (*Kornai János*, *Nagy András*), matematikusok (*Krekó Béla*, *Mentes Imre*) és alumíniumgyártáshoz értő szakemberek voltak a tagjai. A kisszámú termék és technológiai-kereskedelmi variáns lehetővé tette olyan kisméretű optimalizálási modell kidolgozását, amelyet az akkori technikai feltételek mellett (elektroncsöves szovjet számítógép) kezelni lehetett.

A műszaki és gazdasági (lineáris) korlátozó feltételek felállításával nem is volt nagyobb gond, de a célfüggvény meghatározása nehézséget okozott. A probléma gyökere az volt, hogy amíg a ráfordításokat forintban lehetett számításba venni, a hozamokat a világpiaci (dollár) áron kellett értékelni. Ma nevetségesnek tűnhet, hogy ez akkor gondot okozott, hiszen erre való a devizaárfolyam. Csakhogy akkoriban a hivatalos dollárárfolyam, szovjet mintára, olyan alacsony volt, hogy alkalmazási lehetősége fel sem merült. Más, reális (fekete) árfolyamot viszont csak bizonytalanul lehetett volna becsülni, és emiatt bármilyen eredményt nehéz lett volna elfogadtatni.

Végül, a kutatócsoport a feladatot paraméteres optimalizálással oldotta meg, minthogy azonban nem volt olyan paraméter érték, devizaárfolyam, amely mellett a megrendelőknek tetsző eredmény adódott volna, a zárójelentést titkosították és elsüllyesztették.

A munka során ötlött fel *Martos Béla* számára, hogy ha különböző dimenziójú mennyiségeket összeadni-kivonni nem lehet is, de elosztani egymással igen, és ily módon elő lehet állítani az alumíniumipar belső devizaárfolyamát, azaz hogy optimálisan mennyiért lehetne dollárt kitermelni. Ez viszont a hiperbolikus optimalizálási feladat megfogalmazását jelentette, amire abban az időben nem létezett megoldó algoritmus. Ez volt az a kihívás, amelyre körülbelül egyidőben, különböző és egymástól függetlenül kifejlesztett módszerekre alapozva, három megoldás született. Erről részletesebben lehet olvasni *Martos Béla* (1975) könyvében.

Martos Béla idevágó eredményei 1960-ban jelentek meg magyar nyelven, de az eredmény fontosságát mutatja, hogy pár éven belül a teljes cikk angol nyelvű változatát megjelentette az egyik vezető amerikai szakmai folyóirat, a *Naval Research Logistic Quarterly* [3] is. *Martos Bélának* ez a cikke indította el a törtprogramozási kutatásokat, aminek bibliográfiája ma már több száz adatot tartalmaz.

Martos Béla további vizsgálódásaiban érdekes újítást vezetett be: ahelyett, hogy adott feladatosztályhoz keresett volna megoldási algoritmust, adott algoritmushoz, nevezetesen a szimplex algoritmushoz keresett olyan feladatosztályt, amelyre az algoritmus működik. Ezen vizsgálódásai kapcsán jutott el a linearitás különböző általánosításainak vizsgálatához. *Martos Béla* az elsők között vezette be a ma is gyümölcsöző explicit kvázikonkáv és kvázimonoton függvény fogalmakat és vizsgálta ezen függvényosztályok szerepét az optimalizálási feladatokban és szimplex alapú megoldási módszerekben, különös tekintettel a poliedrikus megengedett halmazokra [4], [5], [6]. Kiemelendő eredménye a pozitív szubdefinit kvadratikus függvény osztály bevezetése és jellemzése [7], [8]. Ezek az eredmények is számos további kutatás kiindulási pontjául szolgáltak.

1966–67-ben *Ford* ösztöndíjjal az *Egyesült Államok*ban, *Cambridge*-ben (az *MIT*-n) és *Stanford*ban folytatott tanulmányokat. 1969-ben (majd 1980-ban újra) a *Purdue Egyetem*en (*Indiana, USA*) volt vendégprofesszor egy-egy szemeszterre. 1970-ben elnyerte a matematikai tudományok kandidátusa fokozatot az optimalizáláselméleti eredményeit összegző disszertációjával. Pár évvel később, 1975-ben, angol nyelvű monográfiában foglalta össze kutatási eredményeit *Nonlinear Programming: Theory and Methods* [9] címmel. Hazai és külföldi szakemberek szerint ez a könyv abban az időben a legjobbak között volt. Jóllehet azóta 30 év telt el, és az adott témában több új monográfia is napvilágot látott, könyvét ma is rendszeresen hivatkozzák a szakterület kutatói.

Operációkutatási és optimalizáláselméleti kutatásai széles spektrumot ívelnek át: a tiszta matematikától a releváns alkalmazásokig. A hatvanas években intenzíven tanulmányozta a népgazdaság dinamikus modelljeit, amiről 1967-ben a társszerzőivel (*Andorka Rudolf*fal és *Dányi Dezső*vel) könyvet jelentettek meg *Dinamikus népgazdasági modellek* címmel. Tudományos teljesítményét nagyra értékelte a szakma, amit jelez, hogy beválasztották a *Mathematical Programming Society* elnökségébe, melynek több éven keresztül volt tagja.

Jóllehet nem témám, de nem tehetem meg, hogy ne tegyek említést *Martos Béla* másik kutatási területéről. *Martos Béla* a hetvenes évek elején kezdett foglalkozni a szabályozáselmélet közgazdasági alkalmazásaival. Ebből a témából első cikkét *Kornai János*sal közösen írta a gazdasági rendszerek vegetatív működéséről, melynek angol nyelvű változata a világ egyik legrangosabb matematikai-közgazdasági folyóiratában, az *Econometrica*ban jelent meg 1973-ban. *A gazdasági szabályozási struktúrák* című közgazdasági akadémiai doktori értekezése, melyet 1985-ben sikeresen védett meg, *Economic control structure: a non-Walrasian approach* címmel a *North-Holland* kiadónál jelent meg.

Irodalomjegyzék

[1] Komlósi, S., Dr. Martos Béla szakmai életrajza és tudományos munkássága. Emlékkötet a JPTE Közgazdaságtudományi Kar megalakulásának 20. évfordulója alkalmából díszdoktorrá avatottak tiszteletére, *Studia Oeconomica Auctoriate Universitatis Pécs Publicata* (1992) 5–8.

[2] Komlósi, S., Martos Béla szakmai életrajza, *Szigma* **XXIV** (1993) 1–2, 91–94.

[3] Martos, B., Hiperbolikus programozás, *Az MTA Matematikai Kutatóintézetének Közleményei* **5B** (1960) 383–406; in English: Hyperbolic programming, *Naval Research Logistic Quaterly* **11** (1964) 135–155.

[4] Martos, B., The direct power of adjacent vertex programming methods, *Management Science* **12** (1965) 241–252.

[5] Martos, B., Nem lineáris programozási módszerek hatóköre, *A Magyar Tudományos Akadémia Közgazdaságtudományi Intézetének Közleményei* **20** (1966).

[6] Martos, B., Quasi-convexity and quasi-monotonicity in nonlinear programming, *Studia Scientiarum Mathematicarum Hungarica* **2** (1967) 265–273.

[7] Martos, B., Subdefinite matrices and quadratic forms, *SIAM Journal on Applied Mathematics* **17** (1969) 1215–1233.

[8] Martos, B., Quadratic programming with a quasiconvex objective function, *Operations Research* **19** (1971) 87–97.

[9] Martos, B., *Nonlinear programming: theory and methods* (North-Holland, Amsterdam, Akadémiai Kiadó, Budapest, 1975).

RAPCSÁK TAMÁS
MTA SZTAKI
OPERÁCIÓKUTATÁS ÉS DÖNTÉSI RENDSZEREK LABORATÓRIUM ÉS OSZTÁLY
1518 BUDAPEST, PF. 63.
rapcsak@oplab.sztaki.hu

REVIEW ON BÉLA MARTOS' ACTIVITY IN THE FIELD OF OPTIMIZATION THEORY – ON THE OCCASION OF HIS BEING AWARDED EGERVÁRY COMMEMORATIVE PLAQUE

TAMÁS RAPCSÁK

The profile is a short summary on Béla Martos' professional life, illustrating the background of developing one if his outstanding results, hyperbolic optimization.

Alkalmazott Matematikai Lapok **23** (2006), 5–15.

RÓZSA PÁL MÉLTATÁSA AZ EGERVÁRY JENŐ-EMLÉKÉREM ÁTADÁSA ALKALMÁBÓL

GALÁNTAI AURÉL

Budapest

1. Pályafutásának állomásai

Rózsa Pál 1925. január 20-án született Budapesten. Középiskolai tanulmányait a Toldy Ferenc gimnáziumban, egyetemi tanulmányait pedig a Budapesti Műszaki Egyetemen végezte. 1949-ben, gépészmérnöki diplomájának megszerzése után tanársegédi kinevezést kapott a Miskolci Nehézipari Műszaki Egyetem Borbély Samu által vezetett Matematika Tanszékére. Innen 1950-ben a Közoktatásügyi Minisztérium Felsőoktatási Főosztályára került, ahol a műszaki egyetemek alaptárgyainak

előadója lett. 1951–1955 között az MTA Alkalmazott Matematikai Intézetében volt Egerváry Jenő aspiránsa [5]. 1955 és 1963 között ugyanitt tudományos munkatárs, illetve 1960-tól tudományos főmunkatárs. 1960 és 1963 között Egerváry Jenő utódaként a „Mátrixelmélet és alkalmazásai" csoport vezetője is volt. 1963 és 1968 között a KFKI Matematikai Főosztályát irányította. 1968-ban egyetemi tanárrá és a BME Építőmérnöki Kar Matematika Tanszéke vezetőjévé nevezték ki. 1978-ban átkerült a BME Villamosmérnöki Kar Matematika Tanszékre, amelynek a vezetését is ellátta 1982–1990 között. 1995-ben történt nyugdíjazása óta a BME Villamosmérnöki és Informatikai Kar Számítástudományi és Információelméleti Tanszékének (az előbbi tanszék jogutódjának) emeritus professzora.

2. Tudományos eredményei

Rózsa Pál a lineáris algebrában, az alkalmazott matematika különböző területein és különféle mérnöki és természettudományi alkalmazásaiban ért el számottevő eredményeket. Témakörök szerinti bontásban a következő területeken dolgozott.

Mátrixelmélettel foglalkoznak a [7], [18], [84], [23], [22], [38], [67], [68], [26], [27], [28], [34], [35], [86], [85], [48], [87], [30], [45], [46], [32], [47], [44], [42], [41], [43] dolgozatok.

A mátrixelméletben főként a blokkmátrixok, sávmátrixok és egyéb speciális mátrixok (pl. periodikus tridiagonális mátrixok, tranzitív mátrixok) tulajdonságaival foglalkozott és e terület nemzetközileg elismert vezető kutatójává vált, amit az is mutat, hogy rendszeresen publikál a lineáris algebra vezető kutatóival, Peter Lancaster és Ludwig Elsner professzorokkal. A blokkmátrixok körében tartott vendégprofesszori előadásokat a Braunschweigi Egyetemen [14], valamint a McMaster Egyetemen is [20]. Sávmátrixok szerkezetével és inverzével kapcsolatos [27] és [28] dolgozatait, amelyek egy sor, a Pisai Egyetem munkatársaival közös dolgozat ([34], [35], [86], [85], [48], [87]) létrejöttét inspirálták, a terület fontos eredményeiként tartják számon (lásd pl. R. Bevilacqua: Structural and computational properties of band matrices, in: Complexity of Structured Computational Problems, Applied Mathematics Monographs, CNR, Giardini, Pisa, pp. 131–188 c. áttekintő cikkét). A speciális szerkezetű mátrixok témakörében elért eredményeiért kapta meg a matematikai tudományok doktora fokozatot is [25].

Differenciálegyenletek numerikus megoldásával foglalkoznak a [6], [49], [16], [116], [117], [118], [21], [66] dolgozatok. Itt főként a diszkretizált parciális differenciálegyenletek direkt megoldásait tanulmányozta. A végeselem-módszer pontosságának növelésével foglalkozik a [66] dolgozat.

Az elméleti fizika különféle kérdéseit vizsgálják a [13], valamint a Jánossy Lajossal, Lee Annával és Otto Litzman professzorral (Brno) közösen írt [60], [61], [71], [72], [62], [73], [74], [75], [76], [52], [77], [78], [79], [81], [80] dolgozatai.

Műszaki mechanika témakörben az [2], [4], [10], [45], a Tassi Gézával írt [99], [100], [102], a Szabó Jánossal írt [97], [98], a Michelberger Pállal írt [51], valamint a Peter Lancasterrel írt [69] dolgozatokat publikálta.

Hidak és szerkezetek mechanikájával kapcsolatosak Tassi Gézával közösen írt [101], [104], [105], [115], [106], [107], [108], [109], [112], [113], [114], [110] munkái. Együttműködésük tapasztalatairól „A mérnök és matematikus együttműködése tartószerkezeti feladatok megoldásában" című [111] dolgozatukban is beszámoltak.

Elektrotechnikával foglalkoznak a [82], [83] és a Kerényi Dénessel közös [64], [65] dolgozatok. Vízépítéstannal kapcsolatosak az [50] és [33] cikkek.

Geotechnikai témájúak az Imre Emőkével írt [53], [54], [55], [56], [58], [57], [59] munkák.

Az operációkutatás, ezen belül a többkritériumú döntések körébe tartoznak a Farkas Andrással írt [39], [40] dolgozatok.

Vegyipari alkalmazásokkal foglalkoznak a Jung Gittával, Sárkány Györggyel, illetve Tettamanti Károllyal írt [63], [88], [90], [89] tanulmányok.

A szabályozáselmélet kérdéseivel foglalkozik a [91], [92], [94], [93], [95], [96], [70], [37] dolgozat. E témakörben fő kutatási partnere Naresh K. Sinha professzor (McMaster University) volt.

Rózsa Pál alkalmazott matematikai munkásságát az alkalmazások sokszínűsége és igényessége jellemzi. Az ilyen tárgyú dolgozatai szinte kivétel nélkül az adott szakterület vezető folyóirataiban jelentek meg. Fontos jellemzője ennek a tevékenységének a társszerzőivel kialakított, hosszú időn (évtizedeken) át is töretlen, elkötelezett együttműködés. Pályafutásának minden állomásán kutatók egész sorával alakított ki gyümölcsöző együttműködést.

Nemzetközi elismerését a következő tényekkel lehet még jellemezni. Több mint 50 konferencián volt előadó, vagy meghívott előadó. Egyéni meghívás alapján több mint 80 előadást tartott mintegy 20 országban. Vendégkutató volt a Newcastle upon Tyne Egyetemen (1971), a McMaster Egyetemen (1973–74), a Pisai Egyetemen (1988, 1990) és a Calgary Egyetemen (1993).

Rózsa Pál kezdeményezte a Bolyai János Matematikai Társulat nemzetközi „Numerikus módszerek" konferencia sorozatát, amelynek az 1968, 1973, 1977, 1986, 1990 és 1994 években szervezőbizottsági elnöke is volt. Ez a nagy elismerésnek örvendő konferenciasorozat nyújtott lehetőséget sok hazai matematikusnak arra, hogy megismerje a szakterület külföldi vezető kutatóit, illetve, hogy saját eredményeit közzétegye.

Rózsa Pál jelenleg is igen aktív és eredményes kutatási tevékenységet folytat. 1995-ben történt nyugdíjazása óta (máig) 26 elméleti és alkalmazási tárgyú dolgozatot publikált.

3. Oktatói tevékenysége

Rózsa Pál jelentős és elismert tanári munkássággal rendelkezik. Pályafutása során mérnök-, matematikus-, fizikus- és más szakos hallgatók tömegeit tanította és tanítja mai is. Az egyetemi matematikaoktatásba korán, már 1947-48-ban mint demonstrátor bekapcsolódott. Gépészmérnököket tanított Miskolcon az 1949/50-es tanévben. A Közoktatásügyi Minisztérium referenseként is főként a matematika oktatásával foglalkozott. 1956-tól külső előadóként, illetve másodállásban, az ELTE TTK-n matematikus- és fizikushallgatókat tanított. Ezt a tanári tevékenységét 1968-ban, a BME-re történt kinevezése után speciális előadások formájában folytatta. Bevezetés a mátrixelméletbe c. speciális előadása igen népszerű volt és sok hallgatót vonzott. A diplomás szakemberek továbbképzésébe már 1960-ban bekapcsolódott. 1960–62 között előadásokat tartott az Országos Atomenergia Bizottság Atomtechnikai Tanfolyamán [12]. A szakmérnök-képzésben az Építőmérnöki Karon 1968-tól, a Villamosmérnöki Karon pedig 1978-tól vett részt. 1973-ban mátrixelméleti előadássorozatot tartott a Matematikai Kutató Intézet „A számítástechnika matematikai alapjai" c. kétéves tanfolyamán.

A „Budapest Semesters in Mathematics" program keretében „Válogatott fejezetek az analízis köréből" címmel angol nyelvű előadásokat tartott USA-beli és kanadai egyetemi hallgatók részére 1985–1991 között.

1995-ben történt nyugdíjazása sem törte meg tanári aktivitását. Rózsa Pál ma is rendes előadásokat tart a BME-n, valamint angol nyelven a Western Maryland College (most McDaniel College) kihelyezett budapesti tagozatán.

Vendégprofesszorként a következő egyetemeken tanított: Technische Universität, Braunschweig (1966), McMaster University, Hamilton (1981–82, 1992–93), George Washington University, Washington D.C. (1991–92).

Előadásaihoz minden esetben jegyzetet, vagy könyveket írt [19], [3], [12], [14], [17], [20], [36], [31]. Ezek közül különösen kiemelkedő jelentőségű a háromszor kiadásra került „Lineáris algebra és alkalmazásai" c. könyve [19], amely a témakör alapvető hazai szakkönyve.

A matematika népszerűsítésével, ill. a matematikaoktatás kérdéseivel több dolgozatban ([1], [8], [103], [29]) is foglalkozott.

4. Tagságok, elismerések

Pályafutása során Rózsa Pál fontos tisztségeket töltött be az egyetemi életben és a szakmai közéletben. Tagja a Bolyai János Matematikai Társulatnak, az Amerikai Matematikai Társulatnak (AMS) és a GAMM-nak. Alapítója az ILAS-nak és ugyanezen szervezet nemzetközi bizottságának is tagja. Otto Litzman professzorral együtt végzett eredményes kutatási együttműködéséért a Brnoi Purkyné (jelenleg Masaryk) Egyetem Ezüst Érmét kapta 1979-ben. A Munka Érdemrend ezüst foko-

zatát 1985-ben kapta meg. Nyugdíjba vonulása alkalmából 1995-ben a Köztársasági Érdemrend tiszti keresztjét kapta eredményes munkásságáért.

* * *

Rózsa Pál személyében nemcsak egy kiváló, az elméletben és alkalmazásokban egyaránt sok eredményt elérő és ma is aktív matematikust köszönthetünk, hanem Egerváry Jenő azon tanítványát is, aki igen sokat tett Egerváry munkásságának folytatásáért, nemzetközi és hazai elismertetéséért. Rózsa Pál rendezte sajtó alá Egerváry posztumusz mátrixelméleti dolgozatait a ZAMP-ban és a Publicationesben. Különösen a ZAMP-ban megjelent dolgozat az, amelyből a nemzetközi szakmai közösség Egerváry mátrixelméleti munkásságáról egyáltalán valamilyen képet kapott. Rózsa Pál Egerváry Jenő munkásságát több dolgozatban ([9], [11], [24]) és a [15] könyvrészletben is ismertette. Kutatóként is számos dolgozatában fejlesztette tovább Egerváry eredményeit. A teljesség igénye nélkül említjük meg Ludwig Elsnerrel közös [38] dolgozatát, amelyben Egerváry rangszámcsökkentési tételének egy új bizonyítását, ill. jellemzését is megadják, vagy a Naresh K. Sinha professzorral közös [96] dolgozatát, amelyben Egerváry egy eljárását alkalmazzák. Rózsa Pál tanári tevékenységében is megőrizte és továbbvitte Egerváry „mátrixelméleti iskoláját". Ennek (egyik) nyilvánvaló bizonyítéka Rózsa Pál három kiadásban is megjelent kiváló „Lineáris algebra és alkalmazásai" c. könyve, amelyből sokan ismerték, ismerhettük meg a matematika ezen területét és alkalmazásait.

Hivatkozások

[1] Rózsa, P., A matematikai oktatás feltételei műszaki egyetemeinken és főiskoláinkon, *Magyar Technika* **6** (1951), 88–90.

[2] Rózsa, P., Elasztikusan kapcsolt korpuszkuláris rendszerek kis rezgéseinek vizsgálata a mátrix-számítás segítségével, *MTA Alkalmazott Matematikai Kutató Intézetének Közleményei* **2** (1953), 51–82.

[3] Rózsa, P., *A nomográfia elemei* (Felsőoktatási Jegyzetellátó Vállalat, Budapest, 1955).

[4] Rózsa, P., A mátrix-számítás alkalmazása rudak és lemezek sztatikájában, *MTA Matematikai Kutató Intézetének Közleményei* **1** (1956), 593–621.

[5] Rózsa, P., *A mátrixelmélet néhány új tételéről és azok alkalmazásairól* (kandidátusi értekezés, matematikai tudományok, 1956).

[6] Rózsa, P., Lineáris differenciál- és differenciaegyenletek általános kezdeti feltételeket kielégítő explicit megoldásáról, *MTA Matematikai Kutató Intézetének Közleményei* **2** (1957), 127–143.

[7] Rózsa, P., Megjegyzések egy sztochasztikus mátrix spektrálfelbontásához, *MTA III. (Matematikai és Fizikai) Osztályának Közleményei* **7** (1957), 199–206.

[8] Rózsa, P., Német Matematikai Társaság 1957. évi drezdai kongresszusa, *Magyar Tudomány* **3** (1958), 55–56.

[9] Rózsa, P., Egerváry Jenő munkásságáról, *Matematikai Lapok* **10** (1959), 195–225.

[10] Rózsa, P., O primenenii kletocnyh matric v mehanike korpuskularnyh sistem, *Uspehi Matematiceskih Nauk* **14** (4(88)) (1959), 207–211.

[11] Rózsa, P., In memoriam Egerváry Jenő, *MTA III. (Matematikai és Fizikai) Osztályának Közleményei* **10** (1960), 1–3.

[12] Rózsa, P., *Matematika* (az Országos Atomenergia Bizottság Atomtechnikai Tanfolyamán tartott előadások, Felsőoktatási Jegyzetellátó Vállalat, Budapest, 1960).

[13] Rózsa, P., Über die Verallgemeinerung einer Routhschen Erscheinung, *ZAMM* **41** (1960), 114–115.

[14] Rózsa, P., *Einführung in die Theorie der Hypermatrizen* (Gastvorlesung, Braunschweig, 1966).

[15] Rózsa, P., Egerváry Jenő, in: *Műszaki nagyjaink III*, 337–380 (1967).

[16] Rózsa, P., Ein Rekursionsverfahren zur Lösung linearer Differentialgleichungssysteme mit singulären Koeffizientmatrizen, *Internationale Schriftenreihe zur Numerischen Mathematik* **9** (1968), 117–123.

[17] Rózsa, P., *Lineáris algebra I*, BME Építőmérnöki Kar Szakmérnöki Tagozat, Mérnöki Továbbképző Intézet Kiadványa (Tankönyvkiadó, Budapest, 1969).

[18] Rózsa, P., On Periodic Continuants, *Linear Algebra and its Applications* **2** (1969), 267–274.

[19] Rózsa, P., *Lineáris algebra és alkalmazásai* (Műszaki Könyvkiadó, Budapest, 1974, második kiadás: 1976, harmadik kiadás: Tankönyvkiadó, Budapest, 1991).

[20] Rózsa, P., *Theory of Block Matrices and its Applications* (McMaster University, Department of Applied Mathematics, Hamilton, 1974).

[21] Rózsa, P., A Direct Method for the Numerical Solution of Elliptic Partial Differential Equations, in: J. Miller (ed.), *Topics in Numerical Analysis III*, 369–381 (Proc. Roy. Irish Acad. Conf., Trinity Coll., Dublin, 1976, 1977).

[22] Rózsa, P., Linear Matrix Equations and Kronecker Products, in: *Proceedings of the Fourth Symposium on Basic Problems of Numerical Mathematics, Plzen, 1978*, 153–162 (Prague, 1978).

[23] Rózsa, P., Lineare Matrizengleichungen und Kroneckersche Produkte, *ZAMM* **58** (1978), T395–T397.

[24] Rózsa, P., Jenő Egerváry (1891–1958), a Great Personality of the Hungarian Mathematical School, *Periodica Polytechnica, Electrical Engineering* **28** (1984), 287–298.

[25] Rózsa, P., *Vizsgálatok speciális szerkezetű mátrixok köréből* (doktori értekezés, matematikai tudomány, 1984).

[26] Rózsa, P., Band Matrices and Semi-Separable Matrices (D. Kravvaritis – P. Lancaster – J. Maroulas, Conference Report), *Linear Algebra and its Applications* **84** (1986), 414–417.

[27] Rózsa, P., Band Matrices and Semi-Separable Matrices, in: *Numerical Methods. Colloquia Mathematica János Bolyai 50, 1986*, 229–237 (Amsterdam–Oxford–New York, 1987).

[28] Rózsa, P., On the Inverse of Band Matrices, *Integral Equations and Operator Theory* **10** (1987), 82–95.

[29] Rózsa, P., 200 Years of Teaching Mathematics at the Technical University of Budapest, *Int. J. Math. Educ. Sci. Technol.* **25** (6) (1994), 805–809.

[30] Rózsa, P., Block Matrices and their Applications, in: A. Easton and J. Steiner (eds.), *The Role of Mathematics in Modern Engineering (Melbourne, 1994)*, 85–103 (Lund, 1994).

[31] Rózsa, P., Chapt. Mathematical Preliminaries, in: P. Rózsa and T. Szirtes, *Applied Dimensional Analysis and Modelling*, 1–26 (McGraw Hill, New York, 1997).

[32] Rózsa, P., Kronecker Polynomials and their Applications, *Computers and Mathematics with Applications* **38** (1999), 1–10.

[33] Rózsa, P., Ambrus, S., Szőlősi-Nagy, A. and Kontur, I., On the Observability of the Linear Diffusion Water Model, *Cybernetics and Systems: An International Journal* **18** (1987), 57–70.

[34] Rózsa, P., Bevilacqua, R., Favati, P. and Romani, F., On the Inverse of Block Tridiagonal Matrices with Applications to the Inverses of Band Matrices and Block Band Matrices, *Operator Theory: Advances and Applications* **40** (1989), 447–460.

[35] Rózsa, P., Bevilacqua, R., Favati, P. and Romani, F., On Band Matrices and their Inverses, *Linear Algebra and its Applications* **150** (1991), 287–295.

[36] Rózsa, P. and Bézi, M., *Lineáris algebra* (Tankönyvkiadó, Budapest, 1976).

[37] Rózsa, P., Boros, T., Mironovskij, L. and Mihajlov, N., A Uniform Algorithm for the Transformation of Multivariable Systems Into Canonical Forms, *Linear Algebra and its Applications* **147** (1991), 441–467.

[38] Rózsa, P. and Elsner, L., On Eigenvectors and Adjoints of Modified Matrices, *Linear and Multilinear Algebra* **10** (1981), 235–247.

[39] Rózsa, P. and Farkas, A., An Analysis of Rank Preservation and Reversal in the Analytic Hierarchy Process, *Periodica Polytechnica Series Social and Management Sciences* **4** (1996), 63–78.

[40] Rózsa, P. and Farkas, A., Data Perturbations of Matrices of Pairwise Comparisons. Optimization with Data Perturbations, II, *Annals of Operations Research* **101** (2001), 401–425.

[41] Rózsa, P. and Farkas, A., On the Non-Uniqueness of the Solution to the Least Squares Optimization of Pairwise Comparison Matrices, *Acta Polytechnica Hungarica* **1** (2004), 1–22.

[42] Rózsa, P., Farkas, A. and György, A., On the Spectrum of Pairwise Comparison Matrices, *Linear Algebra and its Applications* **385** (2004), 443–462.

[43] Rózsa, P., Farkas, A. and Lancaster, P., On Approximation of Positive Matrices by Transitive Matrices, *Computers and Mathematics with Applications*, to appear.

[44] Rózsa, P., Farkas, A. and Lancaster, P., Consistency Adjustments of Pairwise Comparison Matrices, *Numerical Linear Algebra with Applications* **10** (2003), 689–700.

[45] Rózsa, P., Farkas, A. and Stubnya, E., Spectral Properties of Input Spectral Density Matrices, in: I. Zobory (ed.), *Proceedings of the 6th Miniconference on Vehicle System Dynamics, Identification and Anomalies VSDIA'98, Budapest, 9-11 November, 1998*, 467–476 (1998).

[46] Rózsa, P., Farkas, A. and Stubnya, E., Spectral Properties of Symmetrically Reciprocal Matrices, *ZAMM* **79** (1999), S3.

[47] Rózsa, P., Farkas, A. and Stubnya, E., Transitive Matrices and their Applications, *Linear Algebra and its Applications* **302–303** (1999), 423–433.

[48] Rózsa, P., Favati, P., Lotti, S. and Romani, F., Generalized Band Matrices and their Inverses, *Calcolo* **28** (1991), 45–92.

[49] Rózsa, P. and Frey, T., Konvergenzschnelle des Differenzenverfahrens der Poissonschen und der biharmonischen Differentialgleichungen, *Periodica Polytechnica (Maschinen und Bauwesen)* **4** (1960), 384–422.

[50] Rózsa, P. and Gálai, A., Mathematical Analysis of a Reservoir, in: *International Conference on Computer Aided Design (CAD) in Hydraulic and Water Resources Engineering HYDROCAD 86*, 113–116 (Budapest, 1985).

[51] Rózsa, P., Galambosi, F. and Michelberger, P., Preliminary Approximate Analysis of Bending Stresses in Lattice Like Vehicle Structures, *Periodica Polytechnica, Transportation Engineering* **10** (1982), 67–82.

[52] Rózsa, P., Hild, E. and Litzman, O., The Green Function in the Microscopical Theory of the Interaction of the Light with a Dielectric Slab, *Scripta Facultatis Scientiarum Naturalium Universitatis Purkynianae Brunensis* **12** (9 (Physica)) (1982), 409–420.

[53] Rózsa, P. and Imre, E., Consolidation Around Piles, in: *Proceedings of the 3rd Seminar on Deep Foundations on Bores and Auger Piles, Ghent*, 385–391 (1998).

[54] Rózsa, P. and Imre, E., Modelling for Radial Consolidation, in: *Proceedings of the XIIth ECSMGE, Amsterdam*, 1017–1027 (1999).

[55] Rózsa, P. and Imre, E., Some Radial Consolidation Models, in: *Proceedings of the XVth ECSMGE, Istanbul* (submitted in 2001).

[56] Rózsa, P. and Imre, E., Modelling for Consolidation Around the Pile Tip, in: *Proceedings of the 9th Conference of Piling and Deep Foundations, Nice*, 513–519 (2002).

[57] Rózsa, P. and Imre, E., Dissipation Test Evaluation with a One-Dimensional Analytical Consolidation Model, *Periodica Polytechnica* (2004).

[58] Rózsa, P. and Imre, E., Dissipation Test Evaluation with a Point-Symmetrical Consolidation Model, in: *2nd International Conference on Geotechnical Site Characterization, Porto*, 513–519 (2004).

[59] Rózsa, P. and Imre, E., Point-Symmetric Consolidation Models for the Evaluation of the Dissipation Test, in: *11th IACMAG, Turin* (2005).

[60] Rózsa, P. and Jánossy, L., Maximum Likelihood Determination of the Scattering Constant of an Emulsion Track in the Presence of Noise, *Il Nuovo Cimento, Serie X* **20** (1961), 817–835.

[61] Rózsa, P., Jánossy, L. and Lee, A., A Coulomb-szórás paraméterének becslése fotoemulzióban végzett mérések alapján, *MTA Matematikai Kutató Intézetének Közleményei, B sorozat* **6** (1961), 467–497.

[62] Rózsa, P., Jánossy, L. and Lee, A., On the Problem of the Evaluation of Emulsion Tracks in the Presence of Spurious Scattering, *Il Nuovo Cimento Serie I* **3** (1965), 281–296.

[63] Rózsa, P., Jung, G. and Sárkány, G., Ellenáramú szétválasztó vegyipari alapműveletek elméleti fokozatszámának meghatározásáról, *MTA Matematikai Kutató Intézetének Közleményei* **2** (1957), 227–245.

[64] Rózsa, P. and Kerényi, D., Über die Berechnung der Stoßsspannungsverteilung in Maschinenwicklungen, Teil I, *Archiv für Elektrotechnik* **51** (1956), 92–100.

[65] Rózsa, P. and Kerényi, D., Über die Berechnung der Stoßsspannungsverteilung in Maschinenwicklungen, Teil II, *Archiv für Elektrotechnik* **53** (1956), 95–99.

[66] Rózsa, P., Kónya, I. and Szabados, T., Applications of Orthogonal Polynomials for Improving the Accuracy of the Finite Element Method, in: *Numerical Methods. Colloquia Mathematica Societatis János Bolyai 50, 1986*, 395–406 (Amsterdam–Oxford–New York, 1987).

[67] Rózsa, P. and Lancaster, P., On the Matrix Equation $AX + X^*A^* = C$, *SIAM Journal on Algebraic and Discrete Methods* **4** (1983), 432–436.

[68] Rózsa, P. and Lancaster, P., Eigenvectors of H-Selfadjoint Matrices, *ZAMM* **64** (1984), 439–441.

[69] Rózsa, P. and Lancaster, P., The Spectrum and Stability of a Vibrating Rail Supported by Sleepers, *Computers and Mathematics with Applications* **31** (1996), 201–213.

[70] Rózsa, P., Lastman, G. and Sinha, N., On the Selection of States to Be Retained in a Reduced-Order Model, *Proceedings of the Institution of Electrical Engineering* **131 Pt.D** (1984), 15–22.

[71] Rózsa, P. and Litzman, O., Allgemeine Behandlung primitiver idealer und nichtidealer Kristallgitter mit Anwendung der Theorie der Hypermatrizen, *Physica Status Solidi* **2** (1962), 28–41.

[72] Rózsa, P. and Litzman, O., Vibrations of Large Imperfections in Crystals, *Proceedings of the Physical Society* **85** (1965), 285–292.

[73] Rózsa, P. and Litzman, O., Long-Range Forces in the Dynamics of Crystals with Defect, *Physica Status Solidi (b)* **58** (1973), 451–456.

[74] Rózsa, P. and Litzman, O., The Localization of Local and Gap Modes by the Sign Count Method, *Czechoslovak Journal of Physics B* **25** (1975), 768–777.

[75] Rózsa, P. and Litzman, O., The Interaction of Light with a Semiinfinite Dielectric as a Phonon Problem: The Generalized Snellius Law and Fresnel Formulae, *Surface Science* **66** (1977), 542–558.

[76] Rózsa, P. and Litzman, O., Reflectivity and Transitivity of a Spatially Dispersive Crystal, *Czechoslovak Journal of Physics* **30** (1980), 816–826.

[77] Rózsa, P. and Litzman, O., The Reflexion and Transmission of the Electromagnetic Wave by a Slab in the Soft X-Ray and Vacuum UV Regions, *Czechoslovak Journal of Physics B* **33** (1983), 1303–1314.

[78] Rózsa, P. and Litzman, O., Optics of Thin Films in the Soft X-Ray and Vacuum UV Regions, *Czechoslovak Journal of Physics B* **34** (1984), 53–68.

[79] Rózsa, P. and Litzman, O., Reflectivity and Transmitivity of a Stack of Thin Films in the Soft X-Ray Region, *Optica Acta* **31** (1984), 1351–1359.

[80] Rózsa, P. and Litzman, O., Ewald's Extended Theory of Diffraction on a Crystal of Finite Thickness, *Acta Crystallographica A* **46** (1990), 897–900.

[81] Rózsa, P. and Litzman, O., The Inversion of a Matrix Occuring in the Dynamical Theory of Diffraction of Electrons, *Scripta Facultatis Scientiarium Naturalium Universitatis Purkynianae Brunensis* **20** (8 (Physics)) (1990), 369–376.

[82] Rózsa, P. and Lovass-Nagy, V., Matrix Analysis of Transient Voltage Distributions in Alternating Ladder Networks, *Proceedings of the Institute of Electrical Engineering* **110** (1963), 1663–1670.

[83] Rózsa, P. and Lovass-Nagy, V., Die Berechnung von Ausgleichsvorgangen auf langskompensierten Fernleitungen, *Archiv für Elektrotechnik* **49** (1964), 260–270.

[84] Rózsa, P. and Reimann, J., Random Flight on Spatial Lattice Points, *Periodica Polytechnica, Civil Engineering* **17** (1973), 47–53.

[85] Rózsa, P. and Romani, F., A Reduction Theorem for the Characteristic Polynomial of Periodic Block Tridiagonal Matrices, in: *Numerical Methods. Colloquia Mathematica Societatis János Bolyai 59, Miskolc, Hungary, 1990*, 37–47 (1990).

[86] Rózsa, P. and Romani, F., On Periodic Block-Tridiagonal Matrices, *Linear Algebra and its Applications* **167** (1992), 35–52.

[87] Rózsa, P., Romani, F. and Bevilacqua, R., On Generalized Band Matrices and their Inverses, in: J. Brown, M. Chu, D. Ellison, and R. Plemmons (eds.), *Proceedings of the Cornelius Lanczos International Centenary Conference, December 12–17, 1993*, 109–121 (1993).

[88] Rózsa, P. and Sárkány, G., Ellenáramú szétválasztó vegyipari alapműveletek elméleti fokozatszámának meghatározásáról II, *MTA Matematikai Kutató Intézetének Közleményei* **4** (1959), 277–298.

[89] Rózsa, P. and Sárkány, G., Analytical Determination of the Number of Theoretical Stages in Binary Rectification and Countercurrent Extraction for Nonlinear Operating Lines, *Periodica Polytechnica, Chemical Engineering* **17** (1973), 335–358.

[90] Rózsa, P., Sárkány, G. and Tettamanti, K., The Analytical Calculation of the Number of Theoretical Plates, *Periodica Polytechnica, Chemical Engineering* **14** (1970), 321–331.

[91] Rózsa, P. and Sinha, N., Efficient Algorithm for Irreducible Realization of a Rational Matrix, *International Journal of Control* **20** (1974), 739–751.

[92] Rózsa, P. and Sinha, N., Minimal Realization of a Transfer Function Matrix in Canonical Forms, *International Journal of Control* **21** (1975), 273–284.

[93] Rózsa, P. and Sinha, N., Decoupling and Pole Displacement in Linear Multivariable Systems: A Direct Algebraic Approach, *Problems of Control and Information Theory* **5** (1976), 329–358.

[94] Rózsa, P. and Sinha, N., Some Canonical Form for Linear Multivariable Systems, *International Journal of Control* **23** (1976), 865–883.

[95] Rózsa, P. and Sinha, N., Algorithm for Transformation of a Polynomial Matrix to Canonical Form, *Problems of Control and Information Theory* **10** (1981), 175–180.

[96] Rózsa, P. and Sinha, N., Some Systems Theory Applications of Egerváry's Algorithm for Transformation of a Matrix to the Hermite Normalform, *Problems of Control and Information Theory* **10** (1981), 267–279.

[97] Rózsa, P. and Szabó, J., Die Matrizengleichung von Stabkonstruktionen (Im Falle kleiner Verschiebungen), *Acta Technica Academiae Scientarum Hungaricae* **71** (1971), 133–148.

[98] Rózsa, P. and Szabó, J., Grosse Verschiebungen von Stabkonstruktionen, *Acta Technica Academiae Scientiarum Hungaricae* **73** (1972), 53–60.

[99] Rózsa, P. and Tassi, G., Primenenie teorii matric k rascetu statisticeski neopredelimyh sterznevyh sistem v uprugoplasticeskoj stadii, *MTA Matematikai Kutatóintézetének Közleményei* **3** (1958), 43–62.

[100] Rózsa, P. and Tassi, G., Rugalmas-plasztikus állapotú sztatikailag határozatlan rúdszerkezetek számítása a mátrixelmélet alkalmazásával, *Építőipari és Közlekedési Műszaki Egyetem Tudományos Közleményei* **4**(2) (1958), 21–42.

[101] Rózsa, P. and Tassi, G., Über eine Anwendung einpariger Matrizen bei der Lösung eines statisch unbestimmten Systems in elasto-plastischen Bereich, in: *II. Magyar Matematikai Kongresszus, II. kötet, VI. szekció*, 53–54 (1960).

[102] Rózsa, P. and Tassi, G., Eine Matrizenmethode zur Lösung statisch unbestimmter Systeme in elasto-plastischen Bereich, *Wissenschaftlische Zeitschrift der Technischen Universität Dresden* **10** (1961), 1329–1331.

[103] Rózsa, P. and Tassi, G., Előszó, in: Kármán Tódor and Biot, M. A., *Matematikai módszerek műszaki feladatok megoldására* (Műszaki Könyvkiadó, Budapest, 1963).

[104] Rózsa, P. and Tassi, G., Kétfázisú rúdmodell vizsgálata diszkontinuitások és képlékeny alakváltozások figyelembevételével, *Műszaki Tudomány* **54** (1978), 81–87.

[105] Rózsa, P. and Tassi, G., Analytische Behandlung des Spannbetontragers aufgrund des Mörschschen Schubmodells, *Acta Mechanica* **41** (1981), 1–9.

[106] Rózsa, P. and Tassi, G., Háromfázisú diszkrét rúdmodell analitikus vizsgálata, in: *Műszaki Mechanikai Tanszéki Kutatócsoport IV. Tudományos Ülésszaka, Tanulmányok*, 120–129 (1986).

[107] Rózsa, P. and Tassi, G., Forces in Prestressed Concrete Bridges Constructed by Free Cantilevering, *Periodica Polytechnica Ser. Civil Engineering* **36** (1992), 355–361.

[108] Rózsa, P. and Tassi, G., Forces in a Concrete Bridge of Special Arrangement, *Budapesti Műszaki Egyetem Építőmérnöki Kar Vasbetonszerkezetek Tanszéke Tudományos Közleményei* (1998), 199–208.

[109] Rózsa, P. and Tassi, G., Treatise on Forces in Cable-Stayed and Extradosed Concrete Bridges, *Budapesti Műszaki Egyetem Építőmérnöki Kar Vasbetonszerkezetek Tanszéke Tudományos Közleményei* (2000), 289–298.

[110] Rózsa, P. and Tassi, G., Forces Caused by Post-Tensioning in Continuous Concrete Girders, *Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Hidak és Szerkezetek Tanszéke Tudományos Közleményei* (2004), 121–130.

[111] Rózsa, P. and Tassi, G., A mérnök és matematikus együttműködése tartószerkezeti feladatok megoldásában, in: *ÉPKO 2004, Csíksomlyó*, 377–388 (2004).

[112] Rózsa, P., Tassi, G. and Hernandez Cruz, A., Contribution to the Forces in Slipformed Concrete Structures, *Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Hidak és Szerkezetek Tanszéke Tudományos Közleményei* (2001), 175–182.

[113] Rózsa, P., Tassi, G. and Hunyadi, M., Adjustment of Cable-Stayed Bridges, *Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Hidak és Szerkezetek Tanszéke Tudományos Közleményei* (2002), 151–160.

[114] Rózsa, P., Tassi, G. and Hunyadi, M., Analytical Solution of Basic Equations of Theory of Structures for Cable-Stayed Bridges, *Rakenteiden Mekanikka* **37** (3) (2004), 18–33.

[115] Rózsa, P., Tassi, G. and Magyari, B., Matrix Analysis of a One-Dimensional Discrete Problem (Forces in a Pressed Sleeve Splice), *Acta Mechanica* **56** (1985), 17–29.

[116] Rózsa, P. and Tóth, I., Über Die Numerische Lösung Partieller Differentialgleichungen mit Anwendung der Theorie der Hypermatrizen, *ZAMM* **44** (1964), T64–T66.

[117] Rózsa, P. and Tóth, I., Über die direkte numerische Lösung von Partiellen Differentialgleichungen mit Anwendung der Hypermatrizen, *Apl. Mat.* **10** (1965), 289–292.

[118] Rózsa, P. and Tóth, I., Eine direkte Methode zur numerischen Lösung der Poissonschen Differentialgleichung mit Hilfe des 9-Punkte-Verfahrens, *ZAMM* **50** (1970), 713–720.

GALÁNTAI AURÉL
MISKOLCI EGYETEM
MATEMATIKAI INTÉZET
3515 MISKOLC–EGYETEMVÁROS
matgal@gold.uni-miskolc.hu

THE WORK OF PÁL RÓZSA

AURÉL GALÁNTAI

This paper give a survey of the research and educational activity of Prof. Dr. Pál Rózsa on the occasion that he is honoured by the Jenő Egerváry medal.

Alkalmazott Matematikai Lapok **23** (2006), 17–37.

AZ ÁLTALÁNOSÍTOTT LPT(k) ALGORITMUSCSALÁD EGYFORMA PÁRHUZAMOS GÉPEK ÜTEMEZÉSÉRE

DÓSA GYÖRGY ÉS VIZVÁRI BÉLA

Veszprém, Budapest

Egyforma párhuzamos gépek ütemezésével és Graham klasszikus LPT algoritmusának egy újabb általánosításával foglalkozunk. (Korábbi [1] cikkünkben már megadtunk egy másfajta általánosítást.) Az LPT sorrend szerint, egyszerre mindig k számú munkát ütemezünk úgy, hogy a teljes átfutási idő növekedése minimális legyen, vagyis az adott állapot mellett minden lépésben lokálisan optimálisan ütemezzük a soron következő k munkát. Fő eredményünk, hogy minden $2 \leq m \leq 4$ gépszám és minden k érték esetén megadjuk az algoritmuscsalád pontos hatékonyságbecslését, az élességet bizonyító példákkal együtt. Végül tesztfeladatokkal demonstráljuk az algoritmuscsalád gyakorlati hatékonyságát.

1. Bevezetés

Az ütemezéselmélet gyakran vizsgált feladata az egyforma párhuzamos gépek ütemezése, amely a következő: Adott munkák egy $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ halmaza. Mindegyik munkát m gép valamelyikével kell elvégezni. A T_i munka elvégzésének ideje bármely gépen $l(T_i)$. Ha egy gép egy munkát elkezd elvégezni, akkor azt be is kell fejeznie. A munkák egy ütemezésén $\mathcal{T}$ valamely $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ partícióját értjük: az i-edik gép végzi a P_i -ben lévő feladatokat. Várakozási idők nincsenek. Az egy gépen elvégzendő munkák sorrendje az ütemezési feladat szempontjából közömbös. A $\mathcal{P}$ ütemezés teljes átfutási ideje

$$\mathcal{L}(\mathcal{P}) = \max_{1 \leq i \leq m} l(P_i), \tag{1}$$

ahol $\mathcal{T}$ tetszőleges X részhalmazára $l(X) = \sum_{T \in X} l(T)$, vagyis $l(P_i)$ az i-edik gépen az utolsó munka befejezési ideje, ahol feltettük, hogy a gépek a 0 időpontban kezdenek dolgozni. A $\mathcal{P}^*$ ütemezés optimális, ha $L(\mathcal{P}^*) \leq L(\mathcal{P})$ a $\mathcal{T}$ halmaz tet-

szőleges $\mathcal{P}$ ütemezése esetén. Optimális ütemezés biztosan létezik, esetleg több is lehet. Az $L(\mathcal{P}^*)$ értéket jelöljük C^*-gal, ami csak $\mathcal{T}$-től és az m számtól függ. A problémával kapcsolatos legfontosabb eredményeket összefoglalja [5].

A feladat az NP-teljes feladatosztályba tartozik, sok heurisztikus módszert dolgoztak ki rá. Ezek között egyik legkorábbi, és egyben a mai napig legnépszerűbb, a Graham-féle LPT (Longest Processing Time) algoritmus [3, 4]. Az algoritmus népszerűségének oka, hogy rendkívül egyszerű, és másrészt sok esetben igen hatékonyan működik. Az algoritmus a következő: Először a munkákat a műveleti idő szerint monoton nemnövekvő sorrendbe rakjuk, majd ebben a sorrendben ütemezzük őket. A következő munka mindig olyan gépre kerül, hogy a lehetséges legkorábbi időpontban fejeződjön be. Az LPT algoritmus vizsgálatával foglalkozó későbbi cikkekről található egy áttekintés korábbi [1] cikkünkben, ahol megvizsgáltuk az algoritmus egy lehetséges általánosítását is. Most egy másik általánosítással foglalkozunk. A jelen cikkben szereplő általánosított algoritmus során (a kezdeti sorbarendezést követően) minden lépésben egyszerre k számú munkát ütemezünk úgy, hogy ezek együttesen a lehető legkevésbé növeljék az átfutási időt. Az algoritmus érdekessége és fő eredményünk, hogy $2 \leq m \leq 4$ gépszám esetén minden k paraméterre pontos elméleti hatékonyság értékeket tudunk majd megadni. Példákkal illusztráljuk majd a módszer gyakorlati hatékonyságát is. Ezek konklúziója, hogy az új algoritmuscsalád sok feladatosztály esetén hatékonyabb az eredeti algoritmusnál.

A cikk szerkezete a következő: A 2. fejezetben megadjuk az algoritmuscsalád pontos definícióját, a 3–6. fejezetekben hatékonyságbecslésekkel foglalkozunk, végül a 7. fejezetben teszteredményeket közlünk.

2. Az általánosított algoritmuscsalád

Jelöljük egy tetszőleges $\mathcal{A}$ algoritmus által a $\mathcal{T}$ feladatra meghatározott ütemezés átfutási idejét $\mathcal{L}_{\mathcal{A}}(\mathcal{T})$-vel. Bevezetjük a következő mennyiséget:

$$R_m(\mathcal{A}) = \sup_{\mathcal{T}} \left\{ \frac{\mathcal{L}_{\mathcal{A}}(\mathcal{T})}{C^*} \right\},$$

és ezt az $\mathcal{A}$ algoritmus elméleti hatékonyságának nevezzük. Ez lényegében azt a szorzót jelenti, ahányszorosa lehet az algoritmus által kapott átfutási idő az optimumértéknek, a lehető legrosszabb esetben. A legrosszabbhoz közeli esetek az összes esetnek csak igen kis százalékában fordulnak elő. Graham LPT algoritmusára igaz a következő

1. TÉTEL. $R_m(\mathrm{LPT}) = \left(\frac{4}{3} - \frac{1}{3m}\right)$. □

A „sup" helyett a képletben „max" is állhat. Legyen ugyanis

$$\mathcal{T}^* = \{2m-1, 2m-1, 2m-2, 2m-2, \ldots, m+1, m+1, m, m, m\}. \tag{2}$$

A halmaz $2m+1$ feladatból áll: kettő-kettő van mindegyik fajtából és a hosszúságuk eggyel csökken, kivéve az utolsó fajtát, amelyből három darab van. Az optimális megoldásnál az utolsó három feladatot egy gépre tesszük, ahol az átfutási idő így $3m$ lesz, a többi feladatot pedig párokba rendezzük úgy, hogy egyet mindig a sor elejéről, egyet pedig mindig a sor végéről veszünk, a két munka hossza együttesen szintén $3m$. Ezzel szemben az LPT algoritmus az első m számú munkát mind különböző gépekre ütemezi, ezek után második munkaként a következő m számú munkát, a gépek fordított sorrendjében, ekkor minden gépen éppen $3m-1$ az átfutási idő. Az utolsó munka elhelyezése után az átfutási idő $4m-1$ lesz. [2] cikkünkben megmutattuk, hogy az előbbi $\mathcal{T}^*$ példán kívül nincs is más olyan feladathalmaz, amelyet LPT ilyen „rosszul" ütemez, minden más feladathalmaz esetén a heurisztikus megoldás értéke kisebb, mint az optimumérték $\left(\frac{4}{3}-\frac{1}{3m}\right)$-szerese.

Az LPT algoritmus a következő munkát a lehető legkorábbi időpontra ütemezi. Helyezzünk el most egyszerre $k \geq 1$ számú munkát úgy, hogy ezek együttesen a lehető legkevésbé növeljék az átfutási időt. $k=1$ esetén nyilván az eredeti LPT algoritmust kapjuk. Az algoritmus formális leírása a következő, ahol m a gépek, n a feladatok száma, $1 \leq k \leq n$ rögzített pozitív egész.

Az általános LPT(k) algoritmus

1. Rakjuk sorba a munkákat csökkenő időtartam szerint. Legyen $n_1 = n$.
2. Legyen $k_1 = \min\{k, n_1\}$.
3. Helyezzük el a soron következő k_1 számú munkát az összes lehetséges módon. Ezek közül válasszuk ki azokat az elhelyezéseket, amikor az átfutási idő növekedése minimális.
4. Az előbbiek közül válasszuk ki azokat az ütemezéseket, amikor a második legnagyobb átfutási idő maximális, ezek közül azokat, amikor a harmadik legnagyobb átfutási idő maximális, és így folytassuk; végül utoljára amikor az $(m-1)$-edik legkisebb átfutási idő maximális (a legkisebb átfutási idő értéke ekkor már adódik).
5. Az így kapott elhelyezések közül válasszunk tetszés szerint, és ütemezzük a kiválasztott módon a munkákat.
6. Legyen $n_1 := n_1 - k_1$. Ha $n_1 > 0$, akkor menjünk a 2. lépésre, egyébként vége.

Figyeljük meg, hogy LPT(k) a soron következő k számú munkát mindig lokálisan optimálisan helyezi el, amin azt értjük, hogy ha nem lenne több munka, csak a soron következő k darab, akkor a pillanatnyi helyzet mellett ennek a hátralévő k darabnak az elhelyezése optimális. Jegyezzük meg, hogy Graham eredeti cikkében is szerepel egy általánosítási lehetőség: A k darab munka optimális ütemezését csak egyszer, az algoritmus elején javasolja, majd a hagyományos LPT-vel folytatja a módosított algoritmust.

2. TÉTEL. *Az* LPT(2) *algoritmus az eredeti LPT algoritmussal azonos ütemezést határoz meg.*

Bizonyítás. Ha egy gépre kerül a soron következő két munka, akkor is; és ha két különböző gépre, akkor is ugyanúgy „helyeződnek el a munkák", mint az LPT algoritmus esetében. □

Definíció. Legyen $\mathcal{A}$ egy ütemezési algoritmus, m a gépek száma. Legyen p, q tetszőleges pozitív egész szám, ahol $p > q$. Egy (p/q) példán olyan $\mathcal{T}$ feladathalmazt értünk, amelyre teljesül, hogy $\mathcal{L}_{\mathcal{A}}(\mathcal{T}) \geq p$, és $C^* = q$. Minimális (p/q) példán olyan $\mathcal{T}$ (p/q) példát értünk, amely minimális az elemszám tekintetében.

1. LEMMA. *Legyen $\mathcal{T}$ egy tetszőleges (p/q) példa. Ekkor tetszőleges $\beta > 0$ esetén, a $\mathcal{T}$-beli munkák mindegyikét β-val megszorozva $(\beta p/\beta q)$ példát kapunk.* □

Az is könnyen látható, hogy az $R_m(\mathcal{A})$ szám, ami azon $\frac{p}{q}$ értékek szupremuma, amelyekhez létezik (p/q) példa, egyben azon $\frac{p}{q}$ értékek infimuma, amelyekhez nem létezik (p/q) példa.

3. TÉTEL. *Tetszőleges rögzített k esetén van olyan m pozitív egész szám, amelyre teljesül a következő egyenlőtlenség: $R_m\big(\mathrm{LPT}(k)\big) \geq \frac{4}{3} - \frac{1}{3m}$.*

Bizonyítás. Az LPT eljárásra $R_m(\mathrm{LPT}) = \frac{4}{3} - \frac{1}{3m} = \frac{4m-1}{3m}$. Tekintsük a (2) példát, ahol $q = 3m$, valamint $p = 4m - 1$. Minden munkából vegyünk k darabot, és k-szorozzuk meg a gépek számát. Ekkor az eredeti LPT algoritmus által adott ütemezéshez hasonlót kapunk, csak az eredetileg szereplő munkák helyén mindegyikből k példány lesz egymás utáni gépekre ütemezve; így a hatékonysági arány is ugyanaz. □

Más a helyzet, ha a gépek m számát rögzítjük, ahogyan ez az alábbi tételből kitűnik:

4. TÉTEL. *Rögzített feladathalmaz esetén $\lim_{k\to\infty} R_m\big(\mathrm{LPT}(k)\big) = 1$, vagyis a k szám növelésével az algoritmus elméleti hatékonysága 1-hez tart.*

Bizonyítás. $k \geq n$ esetén már optimális megoldást kapunk. □

Ezután az egyszerűség kedvéért egy T munka $l(T)$ átfutási idejét többnyire egyszerűen csak T-vel fogjuk jelölni, a legrövidebb ideig tartó munkát A-val jelöljük.

3. Az általános eset

Az alábbiakban minden rögzített m esetén becslést adunk az $\mathrm{LPT}(k)$ elméleti hatékonyságára, és belátjuk, hogy ez k növelésével 1-hez tart. Általában a (2) példa megfelelő módosításai adják majd az élességet bizonyító példákat. Szükségünk lesz a következő lemmára:

2. LEMMA. *Legyen a gépek száma $m \geq 2$ és $\mathcal{T}$ minimális (p/q) példa az $\mathrm{LPT}(k)$ algoritmus esetén. Legyen $A \in \mathcal{T}$ minimális hosszúságú munka. Ekkor $A \geq \frac{m}{m-1}(p - q)$.*

Bizonyítás. A munkák összhossza legföljebb mq. Az egyik gép átfutási ideje az LPT(k) algoritmus által adott ütemezés esetén p. Legyen $\mathcal{T}_1$ azon munkák halmaza, amelyeket az algoritmus utoljára (egyszerre) ütemezett, és ennek az ütemezésnek az ideje legyen a $*$ időpont. Legyen $\mathcal{T}_0 = \mathcal{T}_1 \setminus \{A\}$. Helyezzük el a $*$ időpontban $\mathcal{T}_0$ elemeit úgy, hogy az átfutási idő a lehető legkevésbé növekedjen. Mivel $\mathcal{T} \setminus \{A\}$ nem (p/q) példa $\mathcal{T}$ minimalitása miatt, nem értük el a p átfutási időt. Az ezen munkák ütemezése utáni pillanat legyen a $**$ időpont. Most helyezzük el az A munkát a lehető legkorábbi időpontra. Mivel nem kaphattunk jobb ütemezést, mintha az LPT(k) algoritmus a $\mathcal{T}_1$ elemeit egyszerre helyezte volna el a $*$ időpontban, most is legalább p az átfutási idő, és ez azon a gépen adódik, ahova az A munka került. A többi gép között a legrövidebb átfutási idő legföljebb $q - \frac{p-q}{m-1}$, és ide került az A munka a $**$ időpontban, amikor is elértük a p átfutási időt, ezért A hossza legalább $p - \left(q - \frac{p-q}{m-1}\right) = \frac{m}{m-1}(p-q)$. □

5. TÉTEL. $R_m\big(\mathrm{LPT}(k)\big) \leq \frac{4m-1}{3m}$ *ha* $2 \leq k \leq 2m$. *A becslés éles, ha*

$$2m \equiv 0, 1, 2 \pmod{k}.$$

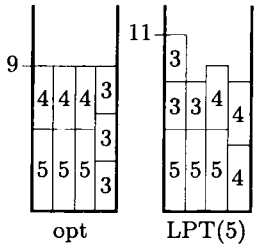
Bizonyítás. Tegyük fel, hogy van olyan $\mathcal{T}$ minimális feladathalmaz, ahol $q = 3m$ és $p > 4m - 1$. A legrövidebb munka hossza a 2. Lemma szerint több, mint m. Emiatt az optimális ütemezésnél minden gépen legföljebb 2 munka van, így a munkák száma legföljebb $2m$. Bármely két munka együttes hossza nagyobb, mint $2m$. Tegyük fel, hogy mindegyik munka hossza kisebb, mint $2m$. Emiatt egyik gépre sem kerül három munka addig, amíg nincs mindegyiken legalább kettő munka. Ekkor az LPT(k) algoritmus ugyanazt az ütemezést eredményezi, mint az LPT algoritmus, és ez optimális megoldás is egyben: A Gantt táblán az alsó sor balról jobbra a felső jobbról balra haladva van feltöltve csökkenő magasságú téglalapokkal. Ha van $2m$-nél hosszabb átfutási idejű munka, az ilyenek az optimális ütemezésnél egyedül vannak a gépeiken. Az LPT(k) algoritmus is csak azután helyezne az ilyen munkák gépeire még egy munkát, amikor a $2m$-nél rövidebb munkák már mind legalább ketten vannak egy-egy gépen, azonban ekkor már nem maradt több munka. Ezzel beláttuk az állítás első részét.

Ameddig az algoritmus alkalmazása során legfeljebb $2m$ munkát ütemeztünk, azok ugyanoda kerülnek, mint ahova őket az LPT algoritmus helyezi. A (2)-beli $\mathcal{T}$ halmaz esetén $2m + 1$ munka van, és ez adja az élességet. Abban az esetben, ha $2m$ k-val osztva 0, 1 vagy 2 maradékot ad, utoljára legfeljebb három munkát helyezünk el egyszerre, ezek egyforma hosszúságúak. Emiatt nem tudjuk őket jobban elhelyezni, mint ha ugyanoda tesszük ezeket is, mint az LPT algoritmus, emiatt a hatékonyság értéke ekkor $\frac{4m-1}{3m}$. □

A becslés éles, ha gépek száma 2 vagy 3, valamint ha $k \leq 4$. Ha a gépek száma $m = 4$, akkor a (2)-beli $\mathcal{T}$ halmazra az LPT(5) algoritmus jobb ütemezést ad, mint az LPT algoritmus. Az alábbi tétel szerint az LPT(5) algoritmus elméleti hatékonysága négy gép esetén valóban jobb, mint $\frac{15}{12}$. Nem végeztünk olyan vizsgálatot, hogy pontosan melyek azok a k számok, amikor $2 < k \leq 2m$ és az előbbi tételben

szereplő becslés nem éles, ugyanis arra fogunk koncentrálni, hogy hogyan változik az elméleti hatékonyság értéke a k paraméter növelésével, a gépek számának rögzítése mellett. Viszont ha a gépek száma legfeljebb négy, és $2 < k \leq 2m$, akkor az előbbi eset az egyetlen kivétel, amikor is LPT (5) hatékonysági szorzója jobb, mint az LPT algoritmusé.

6. TÉTEL. $R_4\big(\mathrm{LPT}(5)\big) = \frac{11}{9}$.



1. ábra

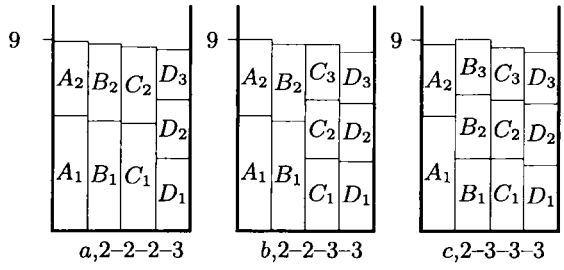
Bizonyítás. Az ábrán látható feladathalmaz (11/9) példa. Tegyük fel, hogy $\mathcal{T}$ minimális $(p/9)$ példa, ahol $p > 11$. A legrövidebb téglalap (a legrövidebb ideig tartó munkának megfelelő téglalap a Gantt táblán) magassága $> \frac{4}{3} \cdot 2 = \frac{8}{3}$, emiatt minden optimális gépen legfőljebb három munka lehet. Ha minden optimális gépen fegfőljebb két munka van, akkor feltehető, hogy azok úgy helyezkednek el, hogy az alul levők hossza balról jobbra, a felül levőké jobbról balra haladva csökken. Nevezzük ezt a sorrendet reguláris sorrendnek. Az első öt munka az algoritmus által ugyanoda kerül, és utána a többi munka is, vagyis az algoritmus a munkákat optimálisan ütemezi. Emiatt van olyan optimális gép, amelyiken három munka van. Ezeken a gépeken a leghosszabb munka hossza is legfeljebb $9 - 2 \cdot \frac{8}{3} = \frac{11}{3}$, a legkisebbnek a hossza pedig legfeljebb 3. Az algoritmus végrehajtása után bármelyik gépre helyezzük át az A munkát, annak teteje a 11 magasságot meghaladja, ezért az A munka nélkül is a gépek átfutási ideje legalább 8, hiszen $A \leq 3$. Emiatt a munkák hosszúságainak összege legalább $11 + 3 \cdot 8 = 35$. Tehát amelyik optimális gépen két munka van, a nagyobbik hossza legalább 4. Belátjuk a következő lemmát:

LEMMA. *Legyen $\mathcal{P}$ a $\mathcal{T} \setminus \{A\}$ olyan ütemezése, amikor a teljes átfutási idő legfeljebb* 11. *Ekkor A elhelyezhető úgy, hogy az ütemezés teljes átfutási ideje továbbra is legfeljebb* 11 *marad.*

Bizonyítás. Legyen a P_i gép átfutási ideje legalább akkora, mint valamelyik optimális gép átfutási ideje. Tegyük az A munkát valamelyik P_i-től különböző gépre. Tegyük fel, hogy az ütemezés átfutási ideje meghaladja a 11-et, ekkor a P_i-től különböző három gép egyike az, ahol több az átfutási idő, mint 11, azaz az optimumértéknél több, mint kettővel nagyobb, a P_i gépen legalább akkora, emiatt a másik két gép átfutási idejénél legalább 2 egységnyi hiány keletkezik, ezért egyiküknek az átfutási ideje kisebb, mint 8. Tegyük akkor ide az A munkát, és nem lépjük túl a

11 egységet, hiszen $A \leq 3$. □

Folytatjuk a tétel bizonyítását. Feltesszük, hogy az optimális gépeken a munkák hossza az ütemezés sorrendjében csökkenő. Vegyük sorra a lehetséges eseteket aszerint, hogy hány optimális gépen van kettő, ill. három munka. Mindegyik esetben azt mutatjuk meg, hogy az LPT(5) algoritmus által adott teljes átfutási idő legfeljebb 11.



2. ábra

a) 2-2-2-3 munka van az optimális gépeken. Feltehető, hogy az első három gépen a munkák sorrendje reguláris. A három leghosszabb idejű munka az A_1, B_1 és a C_1. Helyezzük el az első öt munkát, ekkor a negyedik gépen kettő munka lesz. Az ütemezetlen munkák között szerepel az A_2 és a D_3 munka, mert mindkettő esetén van másik öt olyan munka, amelyeknek a hossza legalább akkora. D_3 hossza legfeljebb 3. Tegyük az A_2 munkát az első gépre, itt ugyanazok a munkák vannak most, mint az optimális ütemezésnél, emiatt ugyanakkora a gép átfutási ideje. A másik három munka közül a D_3-tól különböző két munka elhelyezhető most a második és harmadik gépen úgy, hogy ne lépjük túl a 11 időt, (sőt 9-et sem), mert most ide legfeljebb olyan hosszúságú munkák kerülnek, mint amilyenek az optimális ütemezésnél voltak. Most alkalmazzuk a lemmát, és kapunk egy legfeljebb 11 átfutási idővel rendelkező ütemezést.

b) 2-2-3-3 munka van az optimális gépeken. Feltehető, hogy az első két gépen a munkák sorrendje reguláris. A két leghosszabb idejű munka A_1 és B_1. Az utolsó két gépen levő munkák hossza kisebb, mint 4, emiatt az öt legkisebb mindegyikének a hossza is kisebb, mint 4. Helyezzük el az első öt munkát. Tegyük fel, hogy ezek között az A_2 munka nem szerepel. Ha az első öt munka között ott van C_3, akkor C_2 és C_1 is, D_3 pedig nincs. Ugyanígy, ha D_3 ott van, akkor C_3 nincs. Emiatt a maradék öt munka közül az egyik az A_2, a másik négy hossza kisebb, mint 4, és van olyan is e négy között, amelyiknek a hossza legfeljebb 3.

Tegyük az A_2 munkát az első gépre, ekkor a gép ütemezése nem változott. A másik négy munka közül a három nagyobbikat próbáljuk úgy elhelyezni a másik három gépen, hogy egyiknek az átfutási ideje se lépje túl a 11-et. Ezeken a gépeken most csak olyan munka van, mint az optimális ütemezésnél, és hiányzik még egy legalább $\frac{8}{3}$ hosszúságú munka. Ha valamely gép átfutási ideje több lenne, mint 11, akkor a másik két gép közül van olyan, amelynek az átfutási ideje kisebb, mint

$27 - 11 - \frac{8}{3}$ fele, emiatt kisebb, mint 7. Az előbb említett három nagyobbik munka mindegyikének a hossza kisebb, mint 4. Ha tehát valamelyik gép átfutási ideje túllépné a 11-et, akkor a túlnyúló munkát tegyük a 7-nél kisebb átfutási idejű gépre. Ekkor egyik gép átfutási ideje sem több, mint 11. Alkalmazzuk a lemmát, és kapunk egy 11-nél nem nagyobb idejű ütemezést.

Hátramaradt az az eset, hogy A_2 az első öt munka között van. Ez csak úgy lehet, hogy az öt leghosszabb munka: A_1, A_2, B_1 és B_2, valamint a másik két optimális gép egyikéről a leghosszabb munka, például C_1. Mivel A_1 és B_1 a két leghosszabb munka, ezek az első öt elhelyezésekor külön gépre kerülnek. A másik három munka két gépre kerül. Legyen közülük X az, amelyik öt közül egyedül marad. Tegyük még erre a gépre a D_2 és D_3 munkákat. Ennek a gépnek az átfutási ideje most legalább $D_1 + D_2 + D_3$, mert $X \geq D_1$. Ha $X + D_2 + D_3 > 11$ lenne, akkor $X > D_1 + 2 \geq 4\frac{2}{3}$, ami nem lehet, hiszen B_1, B_2 és C_1 mindegyike ennél kisebb. Maradt még három munka. Tegyük a két nagyobbat az első két gépre, ezek legfeljebb akkorák, mint az optimális ütemezéskori kisebbik munkák, így egyik gép átfutási ideje sem több, mint 11, és a lemma alkalmazható.

c) 2-3-3-3 munka van az optimális gépeken. Ha A_1 és A_2 egy gépre került az első tíz munka elhelyezésekor, tegyük olyan másik gépre az utolsó munkát, amelyen csak két munka van. E három munka együttes hossza legfeljebb $3 \cdot \frac{11}{3} = 11$. Ha A_2 marad utoljára, akkor ha A_1 gépén nincs más munka, oda befér. Ha van, akkor van olyan másik gép, ahol csak két munka van, és ide elfér, mert e három munka mindegyikének a hossza legfeljebb $\frac{11}{3}$. Ha az A_1 és A_2 két különböző gépre került az első tíz munka elhelyezésekor, akkor ha a másik két gép közül valamelyiken csak két munka van, ide elfér az utolsó munka. Ha viszont e másik két gépen három-három munka van, az A_1 és A_2 gépén pedig még egy-egy munka, legyen az A_2 gépén például még rajta kívül az X munka. Legyen az Y olyan munka, amelyik ugyanazon az optimális gépen van, mint az X. Ekkor $Y + X + A \leq 9$. Ha $A_2 + X + A > 11$, akkor $A_2 > Y + 2$, emiatt $A_2 > 4\frac{2}{3}$, ami ellentmond annak, hogy A_2 az első optimális gépen a kisebbik munka. Eszerint az utolsó munka elfér még azon a gépen, amelyiken A_2 van, és az átfutási idő nem lesz nagyobb, mint 11.

d) 3-3-3-3 munka van az optimális gépeken. Mivel a legkisebb munka hossza több, mint $\frac{8}{3}$, a leghosszabb munka hossza is kevesebb, mint $\frac{11}{3}$. Helyezzük el az első öt munkát, majd a következő ötöt. Mivel minimális példáról van szó, egyik gép átfutási ideje sem lépi túl a 11-et. Van két olyan gép, amelyeken legfeljebb csak két munka van. Tegyük ezekre az utolsó két munkát. Mivel három munka átfutási ideje együtt kisebb, mint 11, ellentmondást kaptunk. □

Az 5. Tétel szerint $R_m(\mathrm{LPT}(2m)) = \frac{4m-1}{3m}$. Ezt általánosítja az alábbi tétel:

7. TÉTEL. $R_m(\mathrm{LPT}(\gamma m)) = \frac{2m-1+\gamma m}{m+\gamma m}$, *ahol* $\gamma \geq 2$, *tetszőleges pozitív egész.*

Bizonyítás. A becslés értéke legalább ennyi, ugyanis a (2) példát bővítsük ki $(\gamma - 2)m$ számú m méretű munkával, és megkapjuk az előbbi értéket. Másrészt tegyük fel, hogy van olyan $\mathcal{T}$ minimális feladathalmaz, ahol az optimális megoldás értéke $m + \gamma m$, a heurisztikus megoldás értéke pedig több, mint $2m - 1 + \gamma m$.

A legrövidebb munka hossza több, mint m. Emiatt minden optimális gépen legföljebb γ munka van, így a munkák együttes száma legföljebb γm, ezekre viszont algoritmusunk optimális megoldást ad, ellenmondást kaptunk. □

KÖVETKEZMÉNY. $R_m(\mathrm{LPT}(\gamma m + l)) \leq \frac{2m-1+\gamma m}{m+\gamma m}$, *ahol* $\gamma \geq 2$, *tetszőleges pozitív egész, valamint* $0 \leq l < m$ *nemnegatív egész szám.*

Bizonyítás. A munkák száma az előbbi módon kiszámítva megint legföljebb γm, ezekre viszont a heurisztikus algoritmusunk optimális megoldást ad, ellenmondást kaptunk. □

Ha k értéke nagy, akkor a becslést egy kicsit javítani tudjuk a következő három tétel szerint.

8. TÉTEL. $R_m(\mathrm{LPT}(\gamma m + 1)) \leq \frac{2m+\gamma m}{m+\gamma m+1}$, *ahol* $\gamma \geq 2$, *valamint* $\gamma \geq m - 3$.

Bizonyítás. Tegyük fel, hogy van olyan $\mathcal{T}$ feladathalmaz, ahol $q = m + \gamma m + 1$, míg $p > 2m + \gamma m$. A legrövidebb A téglalap magasságára $A > m$. Emiatt az optimális ütemezésnél minden gépen legföljebb $\gamma + 1$ munka van, ami összesen legföljebb $\gamma m + m$ munka. Emiatt pontosan két ütemben helyezzük el a munkákat. A két ütem közti időpont $*$. A téglalapok összterülete legföljebb $m(m + \gamma m + 1)$. A $*$ időpontban még nem értük el a $2m + \gamma m$ magasságot, különben a többi munka elhagyható lenne.

A $*$ időpontban egyik gépen sem lehet $\gamma + 2$ vagy több munka, mert az ezeknek megfelelő téglalapok együttes hossza meghaladná a $\gamma m + 2m$ magasságot. Ezért mindegyik gépen legfeljebb $\gamma + 1$ munka van, és van ahol éppen ennyi, mert már $\gamma m + 1$ munkát elhelyeztünk. Tekintsünk egy ilyen gépet. Az első γ munka együttes ideje több, mint γm. Legalább ekkora az átfutási idő a többi gépen, különben a $(\gamma + 1)$-edik munkát nem ide tettük volna. Ezért a második ütemben egyik gépre sem kerülhet kettő vagy több munka, ugyanis a $*$-kor meglevő több, mint γm átfutási időhöz így még hozzá jön több, mint m (az egyik munka ideje), ez már több, mint $\gamma m + m$, ekkora átfutási idő mindegyik gépen van a végén, mert akkor a másik munka nem erre a gépre kerülne, ez összesen több, mint $m(m + \gamma m)$, és még ehhez hozzájön a másik munka ideje, ami így összesen meghaladja a téglalapok összterületét. Ugyanígy látható be, hogy az eljárás végén egyik gépen sem lehet $\gamma + 2$ vagy több munka. Tehát $*$ időpontban néhány gépen pontosan $\gamma + 1$ számú munka van, a többi gépen ennél kevesebb. Az algoritmus második lépésében a maradék munkák csak olyan gépekre kerülnek, ahol még legfeljebb γ munka van, továbbá ezek a $*$ időpont után elhelyezett munkák mind különböző gépre kerülnek. Ebből következik, hogy a maradék munkákat az eredeti LPT szabály szerint rakjuk le. Így a minimalitás miatt feltehető, hogy csak egyetlen feladatnak, és éppen egy minimális idejűnek az átfutási ideje fogja csak meghaladni az algoritmus végén a $\gamma m + 2m$ értéket.

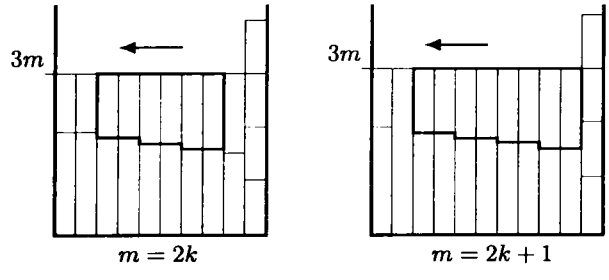
Legyen $A = m + \varepsilon$, ahol $\varepsilon > 0$. Az eljárás végén egy olyan gép van, ahol az átfutási idő több, mint $\gamma m + 2m$, ami az optimumértéknél több, mint $(m - 1)$-gyel több. A többi gépen az átfutási idők átlaga kisebb, mint $m + \gamma m$, hiszen az átfutási idők

összege a téglalapok összterületét nem haladhatja meg. A $*$ időpontban amelyik gépen pontosan $\gamma+1$ munka van, ott az átfutási idő több, mint $\gamma m+m+(\gamma+1)\varepsilon$. Legalább egy ilyen gép van, ezen gép(ek)en az előbbi átlagmagassághoz képest legalább $(\gamma+1)\varepsilon$-nyi többlet keletkezik. Az algoritmus végén lesz legalább egy olyan gép, ahol az átfutási idő nem éri el a $m+\gamma m$ értéket, de az ilyen gépek száma legföljebb $m-2$. Mivel az eredeti feltevés szerint teljesül $\gamma \geq m-3$, ezért $\gamma+1 \geq m-2$, így az algoritmus végén lesz olyan gép, amelyiknek az átfutási ideje kisebb, mint $m+\gamma m-\varepsilon$. Ellentmondáshoz jutottunk, mert akkor az A munkát erre a gépre helyezve a heurisztikus megoldás értéke csökkenne. □

Az alábbi példa mutatja, hogy szükséges a $\gamma \geq m-3$ feltétel. Legyen $m=6$, $\gamma=2$ és $\mathcal{T}=\{9,9,8,8,7,7,6,6,5,5,5,5,5,5\}$. Ekkor a munkák optimális elhelyezésekor mindegyik gép átfutási ideje 15 lesz: $P_1^*=P_2^*=\{9,6\}$, $P_3^*=P_4^*=\{8,7\}$, $P_5^*=P_6^*=\{5,5,5\}$. Az LPT(13) algoritmus először elhelyez 13 munkát a következőképpen: $P_1=P_2=\{9,5\}$, $P_3=P_4=\{8,6\}$, $P_5=\{7,7\}$, $P_6=\{5,5,5\}$. Ekkor az első öt gép átfutási ideje 14, az utolsóé 15. Az utolsó munka ütemezése után az átfutási idő 19 lesz. Mivel $\frac{19}{15}>\frac{24}{19}$, emiatt az előbbi tételben szereplő becslés hat gép és $\gamma=2$ esetén már nem teljesül. Ekkor így a hatékonyság értéke legalább $\frac{19}{15}$.

9. TÉTEL. $R_m\big(\mathrm{LPT}(\gamma m+1)\big) \geq \frac{2m+\gamma m}{m+\gamma m+1}$, *ahol* $\gamma \geq 2$.

Megjegyzés. Most a $\gamma \geq m-3$ feltételre nincs szükség. Állításunkból következik, hogy az $\mathrm{LPT}(\gamma m+1)$ algoritmus hatékonyságbecslése $\gamma \geq \max\{2, m-3\}$ esetén éles.



3. ábra

Bizonyítás. Legyen először $\gamma=2$. Megmutatjuk, hogy ha a (2) példához még egy darab m hosszú téglalapot hozzáveszünk, akkor a hatékonyság értéke $\frac{2m+\gamma m}{m+\gamma m+1}=\frac{4m}{3m+1}$. Az ábrákon a (2) feladathalmaz optimális elhelyezése látható, valamint a hozzávett még egy téglalap az előbbiek fölött. Az első két gépen levő munkák ideje $2m-1$ és $m+1$, a következő két gépen levő munkáké $2m-2$ és $m+2$, és így tovább. Ha a gépek száma páros: $m=2k$, akkor az utolsó előtti gépen levő mindkét munka ideje $3k$; ha $m=2k+1$, akkor az utolsó előtti és azelőtti gépen levő munkák ideje $3k+2$ és $3k+1$. Az utolsó gépen mindkét esetben négy darab m hosszúságú

munka van. Az LPT($2m+1$) algoritmus az előbbi elhelyezéseket produkálja. Az optimális elhelyezéseket a következőképpen kapjuk: A rövidebb ideig tartó munkákat kettő géppel balra tesszük, az első esetben ez a 3.-tól az $(m-2)$.-ig terjedő munkákra, a másik esetben szintén a 3.-tól az $(m-1)$.-ig terjedő munkákra vonatkozik, az ábrán ezek a vastag vonallal keretezett részek. Az érintett gépeken az átfutási idő megnövekszik 1-gyel. Ha a gépek száma páros, akkor tegyük az utolsó előtti gépen levő két munkát az előtte levő két gépre. Ekkor kimaradt hat munka: az első két gépről származik két $m+1$ hosszúságú, és az utolsóról négy m hosszúságú munka. Helyezzük el ezeket a maradék két gépre egyformán. Ekkor az átfutási idő mindegyik gépen $3m+1$ lett. Ha a $m=2k+1$, akkor még az utolsó három gép ütemezése maradt hátra, és nyolc munkát nem helyeztünk még el: Az első két gépről származik két $m+1$ hosszúságú munka, az utolsó előtti és azelőtti gépről két $3k+2$ hosszúságú munka, valamint az utolsóról négy m hosszúságú munka. Tegyük a $3k+2$ hosszúságú munkákat egy gépre, a többit pedig osszuk el a maradék két gép között egyformán. Az átfutási idő most is mindegyik gépen $3m+1$ lett. Így kaptunk egy $\frac{4m}{3m+1}$ hatékonyságú példát. Tetszőleges $\gamma \geq 2$ esetén a szokásos módon kapjuk a megfelelő példát: Adjunk a feladathalmazhoz még $(\gamma-2)\cdot m$ számú m méretű munkát. □

KÖVETKEZMÉNY. $R_m(\mathrm{LPT}(\gamma m+1)) = \frac{2m+\gamma m}{m+\gamma m+1}$, *ahol* $\gamma \geq 2$ *és* $\gamma \geq m-3$ *(vagyis például* $m \leq 5$ *esetén a 8. Tétel becslése éles).*

10. TÉTEL. $R_m(\mathrm{LPT}(\gamma m+l)) \leq \frac{2m+\gamma m}{m+\gamma m+1}$, *ahol* $\gamma \geq m$ *tetszőleges pozitív egész, valamint* $1 \leq l < m$ *nemnegatív egész szám.*

Bizonyítás. Indirekt módon tegyük fel, hogy $\mathcal{T}$ olyan minimális feladathalmaz, amelyre $\mathrm{LPT}(\gamma m+l) > 2m+\gamma m$, de $C^* = m+\gamma m+1$. Válasszunk ezek között a feladathalmazok között olyat, amikor az l szám minimális. Mivel $l=1$ esetén az állítás adódik az előző tételből, $l>1$. A 2. Lemma szerint a legrövidebb téglalap magassága $A > m$. Így optimális ütemezésnél minden gépen legföljebb $\gamma+1$ munka van, tehát a munkák száma legföljebb $m+\gamma m$. Mivel $\gamma m+l$ vagy ennél kevesebb számú feladat esetén optimális megoldást kapnánk, a feladatok száma ennél több. A $*$ időpontban összesen $\gamma m+l$ számú munkát helyeztünk már el a gépekre, a többi munkát már mind elhelyezi egyszerre az algoritmus a következő lépésben. $*$-kor van olyan gép, amelyiken γ -nál több munka van, vagyis a $\gamma \geq m$ feltétel miatt a $*$ időpontban van olyan gép, amelyiken m-nél több munka van. Ezek között van két olyan, amelyek az optimális ütemezés esetén azonos gépeken vannak. Módosítsuk úgy a feladathalmazt, hogy e két munka helyett vegyünk egyetlen olyat, aminek a végrehajtási ideje egyenlő az előbbi két munka idejének összegével. Legyen ez a $\mathcal{T}'$ feladathalmaz. Most végezzük el az $\mathrm{LPT}(\gamma m+l-1)$ algoritmust a $\mathcal{T}'$ feladathalmazzal. A $*$ időpontban minden gép átfutási ideje ugyanannyi lesz, mint az előbb, és a maradék téglalapok ugyanoda kerülnek, mint ahova $\mathrm{LPT}(\gamma m+l)$ helyezte az előbb a $\mathcal{T}$ maradék téglalapjait. (Most használtuk fel először az algoritmus 4. pont-

jának teljesülését!) Így most is ugyanolyan hatékonyságú példát kaptunk, mint az előbb, ez pedig ellentmond az l szám minimális választásának. □

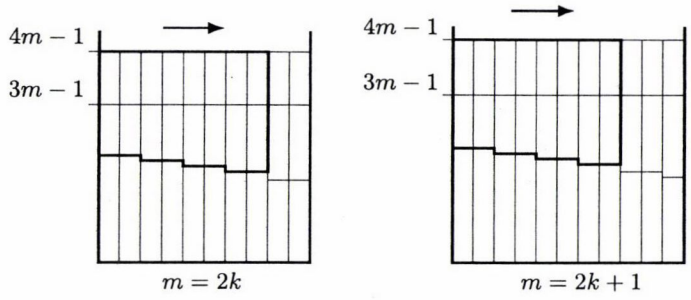
Az eddigieket összefoglalva elmondhatjuk, hogy minden m-re és k-ra van eléggé pontos felső becslésünk, ami ráadásul éles is, ha $2 \leq k \leq 2m$, és $2m$ k-val osztva legfeljebb kettő maradékot ad, illetve ha $k \geq \max\{2m, (m-3)\,m\}$, és k m-mel osztva 0 vagy 1 maradékot ad.

Ha $m = 2$, akkor az előző tételeket alkalmazva minden k-ra pontos becslést tudunk megadni. Ha a gépek száma három, akkor a „kicsi" k értékek 2 és 6 közöttiek, a „nagyok" pedig 9-től kezdődnek, ezek mindegyikére pontos becslésünk van. $k = 7$ esetére alkalmazható a 10. Tétel előtti következmény, csak $k = 8$ esete maradt ki, erre a következő tételben fogunk pontos becslést megadni. Ha a gépek száma négy, és a k értéke legföljebb 8 vagy legalább 16, akkor alkalmazhatók az előbbi eredmények. Az előbbi határok között tudjuk még az előző tételek alapján a pontos hatékonysági értéket 9, 12, és 13 esetén, a többi k-ra az elméleti hatékonyság értéke szabálytalan. A következő tételben megvizsgáljuk a $k = 3m - 1$ esetet. A következő fejezetekben 2, 3 és 4 gép esetén *minden* k-ra meghatározzuk az elméleti hatékonyság pontos értékét.

11. TÉTEL. $R_m(\mathrm{LPT}(3m-1)) = \frac{5m-2}{4m-1}$.

Bizonyítás. Tegyük fel, hogy $\mathcal{T}$ olyan feladathalmaz, ahol $q = 4m - 1$, míg $p > 5m - 2$. Ekkor $A > m$. Emiatt optimális gépen legföljebb 3 munka van. Mivel $3m$-nél kevesebb munka esetén az eljárás optimális megoldást ad, pontosan $3m$ a munkák száma, és minden optimális gépen három munka van. Tegyük fel, hogy a $*$ időpontban van olyan gép, amelyre két munkát, C_1-et és C_2-t ($C_1 \geq C_2$) osztottunk. Az utolsó munkát erre a gépre téve az átfutási idő meghaladja az $5m - 2$ értéket, ugyanis ha a legkorábbi időpontra ütemezzük az algoritmusunk szerint, akkor is meghaladja ezt az értéket. Emiatt teljesül a $C_1 + C_2 + A > 5m - 2$ egyenlőtlenség, ahonnan $2C_1 + A > 5m - 2$. Mivel a C_1 munka mellett optimális ütemezésnél még két másik munka van ugyanazon a gépen, teljesül a $C_1 + 2A \leq 4m - 1$ egyenlőtlenség is. A két egyenlőtlenséget összevetve $A < m$ adódik, ami ellentmondás. Ellentmondásra jutunk akkor is, ha a $*$ időpontban van olyan gép, amelyiken csak egy munka van: Jelöljük ezt az egy munkát C-vel. Ekkor $C + A > 5m - 2$, de akkor optimális ütemezésnél is egyedül kellene a C munkának lennie. Az előzőek alapján a $*$ időpontban mindegyik géphez legalább három munka lett már rendelve, ami ellentmond annak, hogy eddig a pillanatig még csak $3m - 1$ munkát osztottunk szét a gépek között.

A becslés éles voltát a (2) példa azon módosításával látjuk be, ahol (2)-t kiegészítettük $m - 1$ darab m hosszú téglalappal. Ábránkon (2) LPT algoritmus szerinti elhelyezése látható, kiegészítve a hozzávett téglalapokkal. Minden gépen három munka van, a legrövidebb munka ideje mindig m. Az első két gépen levő másik két munka ideje $2m - 1$ és m, a következő két gépen pedig $2m - 2$ és $m + 1$, és így tovább. Ha a gépek száma páros: $m = 2k$, akkor az utolsó előtti két gépen levő további munkák időtartama $3k$ és $3k - 1$; ellenkező esetben az utolsó gépen levő



4. ábra

mindkét további munka időtartama $3k+1$. A heurisztikus megoldásban a vastag vonallal keretezett téglalapok két géppel jobbra kerülnek, ahol az átfutási idő 1-gyel csökken.

Ha a gépek száma páros, akkor marad még két gép (az első és a második), valamint hat munka: az első két gépről származik még egy-egy $2m-1$ hosszúságú, és az utolsó kettőről két $3k-1$ és két m hosszúságú munka. Az egyik gépre kerül a két $2m-1$ hosszúságú, a másikra két $3k-1$ és egy m hosszúságú munka. Ezek a munkák helyeződnek el az LPT$(3m-1)$ algoritmus első lépésében, az átfutási idő ekkor minden gépen pontosan $4m-2$, és még hátramaradt egyetlen m idejű munka, ezeket elhelyezve lesz az algoritmus átfutási ideje egyik gépen $5m-2$.

Ha a gépek száma páratlan, akkor még három gép maradt, az első kettő és az utolsó, valamint kimaradt még kilenc munka. Az első két gépről származó két $2m-1$ hosszú munkát tegyük egy gépre. Hátra van még kettő $3k$, kettő $3k+1$ és három m hosszúságú munka. Helyezzünk el két gépre három-három munkát a következőképpen: egy $3k$, egy $3k+1$, és egy m hosszúságút. Most mindegyik gépen pontosan $4m-2$ az átfutási idő, és ezt az elhelyezést valósítja meg az LPT$(3m-1)$ algoritmus az első lépésében. Megint maradt még egyetlen m idejű munka, ezt elhelyezve lesz az algoritmus átfutási ideje éppen $5m-2$. □

12. TÉTEL. $R_m\big(\mathrm{LPT}(k)\big) < \frac{m+k}{k+1}$ *ha* $k \geq 2m$.

Bizonyítás. Legyen $k=\gamma m+l$, ahol $0 \leq l < m$. Teljesül a következő egyenlőtlenség-lánc: $R_m\big(\mathrm{LPT}(\gamma m+l)\big) \leq \frac{2m+\gamma m}{m+\gamma m+1} < \frac{m+\gamma m+l}{\gamma m+l+1} = \frac{m+k}{k+1}$. □

KÖVETKEZMÉNY. *Az előbbi tétel szerint rögzített m esetén az algoritmusunk elméleti hatékonysága 1-hez tart:* $\lim_{k\to\infty} R_m\big(\mathrm{LPT}(k)\big) = 1$.

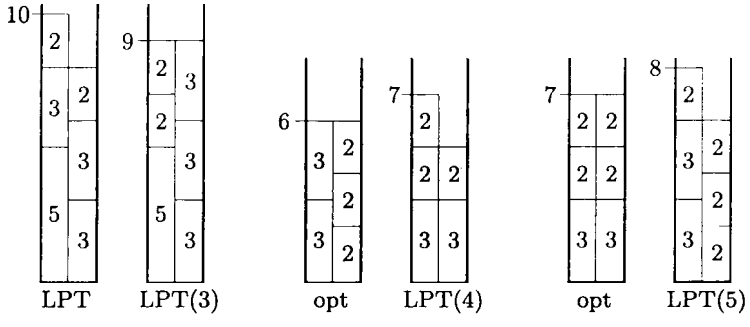
A következő három fejezetben megvizsgáljuk két, három, illetve négy gép esetén az összes lehetséges esetet.

4. $m = 2$ esete

Legyen $m = 2$, $\mathcal{T} = \{5, 3, 3, 3, 2, 2\}$. Az 5. ábra bal oldali része mutatja, hogy az LPT algoritmusnak 10, míg LPT(3)-nak 9 egység a teljes átfutási ideje.

Az 5. Tétel szerint az LPT(3) és LPT(4) algoritmusok hatékonysága $R_2 = \frac{7}{6}$: Álljon $\mathcal{T}$ a következő téglalapokból: $\mathcal{T} = \{3, 3, 2, 2, 2\}$. Ekkor az LPT(3) és LPT(4) által meghatározott átfutási idő 7, míg az optimális megoldás értéke 6 egység (5. ábra). A legelső k, amelyre javul algoritmusunk hatékonysága, a $k = 5$. Az előző fejezetbeli 6., 7. és 8. Tételek alapján tetszőleges k-ra megadtuk az algoritmusunk hatékonyságát. $m = 2$ esetén azonban az algoritmus hatékonyságát közvetlenül és egyszerre, tetszőleges $k \geq 2$ esetére is meg tudjuk határozni, az alábbi tétel szerint:

13. TÉTEL. *Ha* $k \geq 4$, *akkor* $R_2\big(\mathrm{LPT}(k)\big) = \frac{k+3}{k+2}$.

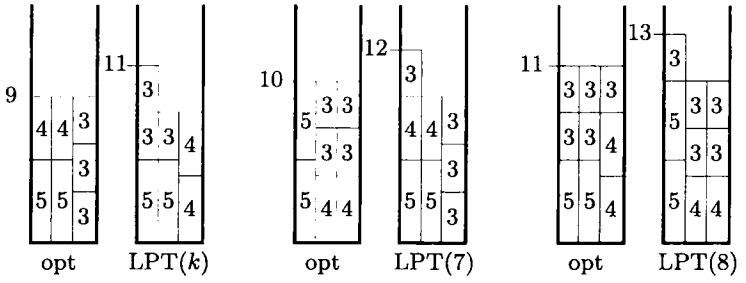


5. ábra

Bizonyítás. $\frac{k+3}{k+2}$-es példának megfelel a következő: Álljon a $\mathcal{T}$ feladathalmaz 2 darab 3 egység magasságú, és $k - 1$ darab 2 egység magasságú téglalapból. Ekkor az LPT(k) által meghatározott átfutási idő $k + 3$, míg az optimális megoldás értéke $k + 2$. (Az optimális és heurisztikus megoldások ugyanis $k = 4$ és $k = 5$ esetén az 5. ábrán szerepelnek. Nagyobb k számok esetén pedig az előbbi téglalapok fölé kerül még néhány sor a 2 hosszúságú téglalapból, páros k esetén a középső, páratlan k esetén pedig a jobb oldali ábrát véve alapul.) Meg kell mutatnunk még, hogy $R_2\big(\mathrm{LPT}(k)\big) \leq \frac{k+3}{k+2}$. Tegyük fel ezzel ellentétben, hogy létezik olyan $\mathcal{T}$ feladathalmaz, ahol az optimális megoldás értéke éppen $k + 2$, a heurisztikus megoldás értéke több, mint $k + 3$. A téglalapok összterülete legföljebb $2 \cdot (k + 2)$. A 2. Lemma miatt a legrövidebb téglalap magassága több, mint 2. Emiatt $k + 2$-nál kevesebb téglalap van, $k + 1$-nél kevesebb nem lehet, mert ekkor LPT(k) optimális megoldást határozna meg, így pontosan $k + 1$ darab téglalap van. Az első k számú téglalapot LPT(k) ezekre vonatkozóan optimálisan helyezi el, ezek $\mathcal{T}$ minimalitása miatt beleférnek $k + 3$ sávmagasságba. Az utolsó téglalap az alacsonyabb helyre kerül, ekkor növekszik a sávmagasság $k + 3$ fölé. A másik magasság ezért kisebb, mint $k + 1$, az utolsó téglalap elhelyezése előtt ezért mindkettőé kisebb, mint $k + 1$. Legyen k páratlan.

Ekkor az első k darab téglalap elhelyezésekor az egyik gépre legalább $\frac{k+1}{2}$ feladat került, ezek együttes átfutási ideje több, mint $k+1$, ellentmondást kaptunk. Most legyen az k páros szám. Mivel összesen $k+1$ számú téglalap van, az optimális megoldásnál van olyan gép, amelyikre legalább $\frac{k+2}{2}$ számú munka kerül, ezek együttes átfutási ideje több, mint $k+2$, megint ellentmondást kaptunk. □

5. $m = 3$ esete



6. ábra

Ha $m = 3$, az LPT algoritmus éles felső becslése $\frac{11}{9}$. A $\mathcal{T} = \{5,5,4,4,3,3,3\}$ feladathalmaznál az optimum értéke 9, míg az LPT által adott ütemezésé 11. Az LPT(k) algoritmus ugyanezt az elhelyezést valósítja meg $3 \le k \le 6$ esetén, és az 5. Tétel szerint $R_3\big(\mathrm{LPT}(k)\big) = \frac{11}{9}$ $(3 \le k \le 6)$. Továbbá a 6. Tételben beláttuk, hogy $R_3\big(\mathrm{LPT}(6+3\gamma)\big) = \frac{11+3\gamma}{9+3\gamma}$. A 7. és 8. Tételek szerint $R_3\big(\mathrm{LPT}(7+3\gamma)\big) = \frac{12+3\gamma}{10+3\gamma}$, ahol $\gamma \ge 0$. A becslés éles voltát bizonyító példát úgy kaptuk, hogy a (2) példához hozzávettünk még egy legrövidebb idejű munkát, az optimális, illetve heurisztikus megoldásokat mutatja $\gamma = 0$ esetén a 6. ábra. Nagyobb γ esetén a szokásos módon járunk el: hozzáveszünk a feladathalmazhoz 3γ számú, 3 egység időtartamú munkát.

A 11. Tétel szerint $R_3\big(\mathrm{LPT}(8)\big) = \frac{13}{11}$. Az éles példa $\mathcal{T} = \{5,5,4,4,3,3,3,3,3\}$. Három gép esetén az LPT$(11+3\gamma)$ algoritmus elméleti hatékonysága legfeljebb $\frac{15+3\gamma}{13+3\gamma}$, ha $\gamma \ge 0$, a 10. Tétel szerint. Most megmutatjuk, hogy ez a becslés éles.

14. TÉTEL. *Ha* $\gamma \ge 0$, *akkor* $R_3\big(\mathrm{LPT}(11+3\gamma)\big) = \frac{15+3\gamma}{13+3\gamma}$.

Bizonyítás. Csak azt kell belátnunk, hogy a hatékonyság értéke legalább ekkora. Legyen először $\gamma = 0$. Tekintsük a $\mathcal{T} = \{4,4,4,3,3,3,3,3,3,3,3,3\}$ feladathalmazt. Az optimum értéke 13: mindegyik optimális gépen egy darab 4, és három darab 3 hosszúságú munka van. Az LPT(11) által kapott megoldás átfutási ideje 15: az első gépre kerülnek a 4 hosszúságú munkák, a másik két gépre négy-négy darab 3

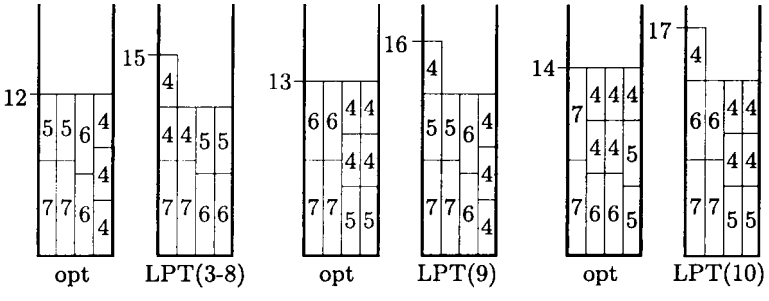
hosszúságú munka, az algoritmus első ütemében. Ekkor minden gép átfutási ideje 12, és még hátravan egy darab 3 egység hosszú munka elhelyezése. Nagyobb γ számok esetén elég az előbbi $\mathcal{T}$ halmazhoz még 3γ darab 3 egység hosszúságú téglalapot kell hozzávenni, ekkor az optimális és a heurisztikus megoldás értéke is 3γ-val növekszik. □

Az előző tételek eredményeit foglalja össze az alábbi

15. TÉTEL. *Legyen* $k \geq 8$. *Ha* $k = 3\gamma$ *vagy* $k = 3\gamma + 1$ *alakú, akkor* $R_3\big(\mathrm{LPT}(k)\big) = \frac{5+k}{3+k}$, *és* $R_3\big(\mathrm{LPT}(k)\big) = \frac{4+k}{2+k}$ *ha* $k = 3\gamma + 2$ *alakú.* □

6. $m = 4$ gép esete

Négy gép esetén az LPT algoritmus éles felső becslése $\frac{15}{12}$. Az LPT(k) algoritmus ugyanezt az elhelyezést valósítja meg $3 \leq k \leq 8$, $k \neq 5$ esetén (a $k = 5$ esetet a 6. Tételben külön megvizsgáltuk, ekkor az éles becslés értéke $\frac{11}{9}$). Ezért a $\frac{15}{12}$ éles hatékonysági becslés adódik az 5. Tétel szerint, ha $3 \leq k \leq 4$, és ha $6 \leq k \leq 8$ (7. ábra).



7. ábra

A 7. Tétel szerint $R_4\big(\mathrm{LPT}(8 + 4\gamma)\big) = \frac{15+4\gamma}{12+4\gamma}$.

$R_4\big(\mathrm{LPT}(9 + 4\gamma)\big) = \frac{16+4\gamma}{13+4\gamma}$. A Graham-féle (2) példa esetén LPT(9) optimális megoldást ad, és eljárásunk hatékonysága $m = 4$ esetén most először jobb, mint az LPT hatékonysága. $\gamma = 0$ esetén a 7. ábra mutatja, hogy az elméleti hatékonyság legalább $\frac{16}{13}$, másrészt a 7. Tétel szerint legföljebb ennyi. Nagyobb γ számok esetén az éles példát megint úgy kapjuk, hogy hozzáveszünk 4γ számú 4 időtartamú munkát az előbbi példa elemeihez, ezek a téglalapok fognak az előzőek fölött elhelyezkedni. Ezután az általános részben még nem tisztázott esetek következnek.

16. TÉTEL. $R_4\big(\mathrm{LPT}(10)\big) = \frac{17}{14}$.

Bizonyítás. A 7. ábra jobb oldali példája szerint az elméleti hatékonyság legalább ekkora. Tegyük fel, hogy $q = 14$ egység, míg $p > 17$. Most is igaz, hogy $A > 4$.

Minden optimális gépen legföljebb 3 munka van, ezért 11 vagy 12 a téglalapok száma. A $*$ időpontban egyik gépen sem lehetett háromnál több munka. Az első tíz téglalap lerakásakor ezért 3, 3, 2, 2 vagy pedig 3, 3, 3, és 1 munka került a gépekre. Ha 12 a munkák száma, akkor minden optimális gépen három munka van, ezért minden munka ideje kisebb, mint 6. Ezért a $*$ időpontban amelyik gépen két munka van, ott az átfutási idő kisebb, mint 12, ahol viszont három, ott ennél több. Így a maradék két munka olyan gépekre kerül, ahol legfeljebb két munka van még. Ebből kapjuk, hogy $2C_1 + A > 17$, másrészt $C_1 + 2A \leq 14$, ezekből $A < \frac{11}{3}$ következik, ami ellentmondás. Tegyük fel ezért, hogy 11 a munkák száma. Ha 3, 3, 3, 1 munka került a gépekre, legyen C annak a gépnek az átfutási ideje, ahol egy munka van. Ekkor $C + A > 17$, emiatt ez a munka egyedül van az optimális megoldásál is, így a munkák száma legföljebb $3 \cdot 3 + 1 = 10$, ami túl kevés. Maradt az a lehetőség, hogy 3, 3, 2, 2 a gépeken levő munkák száma. A harmadik gépen lévő két munka legyen A_1 és A_2, az utolsó gépen levők pedig B_1 és B_2. Legyen $A_1 \geq A_2$ és $B_1 \geq B_2$. Ha az A_1 munka mellett az optimális megoldásnál még lenne két munka, akkor az előbbi egyenlőtlenség-rendszert kapnánk: $2A_1 + A > 17$, $A_1 + 2A \leq 14$, ami ellentmondásra vezet. Ezért A_1 mellett az optimális megoldásnál legfeljebb egy munka lehet, és ugyanez igaz B_1-re is. Ez csak úgy lehet, ha A_1 és B_1 ugyanazon az optimális gépen vannak és nincs több azon a gépen. A_1 és B_1 együttes ideje legfeljebb 14, ezért egyikük nem hosszabb, mint 14 fele, legyen például $A_1 \leq 7$. Ekkor $A_1 + A_2 + A > 17$, amiből kapjuk: $A_2 + A > 10$. Másrészt $A_2 + 2A \leq 14$, amiből következik $A < 4$, ellentmondás. □

Az előbbiekhez hasonlóan most LPT$(10 + 4\gamma)$ hatékonyságára szeretnénk pontos becslést kapni. Az LPT(14) algoritmus esetében a várható $\frac{21}{18}$ értéktől eltérően $\frac{20}{17}$ a pontos becslés értéke.

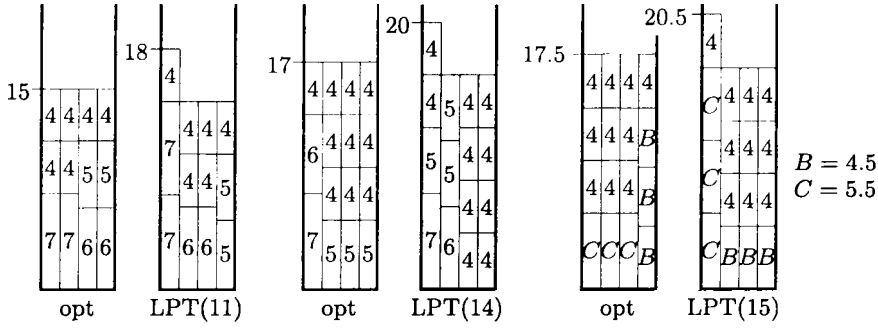
17. TÉTEL. $R_4\big(\text{LPT}(14 + 4\gamma)\big) = \frac{20+4\gamma}{17+4\gamma}$, *ahol* $\gamma \geq 1$.

Bizonyítás. Legyen $\gamma = 0$. A 8. ábrán középének példája mutatja, hogy az elméleti hatékonyság legalább ennyi. Tegyük fel, hogy $q = 17$, $p > 20$. Ekkor $A > 4$. A téglalapok száma 15 vagy 16. Könnyen látható, hogy az eljárás végén egyik gépen sem lehet 5 vagy több munka. Ha 16 téglalap van, minden optimális gépre négy munka jut, így mindegyik hossza legfeljebb 5, ezért a heurisztikus megoldás értéke legföljebb 20, ami ellentmondás. Ezért pontosan 15 munka van. A legutolsó munka olyan gépre kerül, ahol négynél kevesebb munka van. Legyen most is $A = 4 + \varepsilon$, és a 8. Tételben leírtakhoz hasonlóan ellentmondáshoz jutunk. Ezután $\gamma > 0$ esetén a szokásos módosításból: 4γ darab 4 egység hosszú munkát a feladathalmazhoz hozzávéve adódik az állítás azon része, hogy az elméleti hatékonyság értéke legalább ekkora. Az ellenkező irányú egyenlőtlenség a 10. Tételből adódik. □

Az általános részben láttuk, hogy $R_4\big(\text{LPT}(11)\big) = \frac{18}{15}$. A 8. ábrán bal oldali példája bizonyítja az élességet. Hátra van még az LPT$(11 + 4\gamma)$ algoritmus hatékonyságának megkeresése. Az LPT(15) algoritmus esetében $\frac{22}{19}$ helyett meglepő módon $\frac{20.5}{17.5}$ a pontos becslés értéke.

18. TÉTEL. *Négy gép esetén* LPT(15) *elméleti hatékonysága* $R_4\big(\mathrm{LPT}(15)\big) = \frac{20.5}{17.5}$.

Bizonyítás. A 8. ábra jobb oldali példája szerint az elméleti hatékonyság legalább ekkora. Tegyük fel, hogy $q = 17.5$, míg $p > 20.5$. Ekkor $A > 4$ és minden műveleti idő kisebb, mint 5.5. Pontosan 16 téglalap van. A $*$ időpontban mindegyik gépen van legalább három munka. Legyen C a legkisebb átfutási idő, ekkor teljesülnek az alábbi egyenlőtlenségek: $C + A > 20.5$, másrészt $\frac{C}{3} + 3A \leq 17.5$, amiből következik $A < 4$, ellentmondást kaptunk. □



8. ábra

19. TÉTEL. *Ha* $\gamma \geq 1$, *akkor* $R_4\big(\mathrm{LPT}(15 + 4\gamma)\big) = \frac{20+4\gamma}{17+4\gamma}$.

Bizonyítás. A 4 darab 5 hosszú és 16 darab 4 hosszú téglalapból álló példa ekkora hatékonyságot ad. Az optimum értéke 21, ahol minden gépen ugyanolyan munkák vannak. LPT(19) 24 egységnyi magasságot használ fel. A hatékonyság értéke legfeljebb ekkora a 10. Tétel miatt. □

A „nagy" k számokra vonatkozó becsléseket foglalja össze a

20. TÉTEL. *Ha* $k \geq 16$, *akkor* (i) $R_4\big(\mathrm{LPT}(k)\big) = \frac{7+k}{4+k}$, *ha* $k = 4\gamma$ *vagy* $k = 4\gamma + 1$ *alakú*, (ii) $R_4\big(\mathrm{LPT}(k)\big) = \frac{6+k}{3+k}$, *ha* $k = 4\gamma + 2$ *alakú*, (iii) $R_4\big(\mathrm{LPT}(k)\big) = \frac{5+k}{2+k}$, *ha* $k = 4\gamma + 3$ *alakú*. □

A következő táblázatban az $m = 2, 3, 4$ esetén kapott eredményeket szemléltetjük. Ha a $k > 23$, a táblázat alsó részén fellelhető szabály szerint következnek az elméleti hatékonyság pontos értékei.

	$m=2$	$m=3$	$m=4$
$k=1$	7/6	11/9	15/12
$k=2$	7/6	11/9	15/12
$k=3$	7/6	11/9	15/12
$k=4$	7/6	11/9	15/12
$k=5$	8/7	11/9	11/9
$k=6$	9/8	11/9	15/12
$k=7$	10/9	12/10	15/12
$k=8$	11/10	13/11	15/12
$k=9$	12/11	14/12	16/13
$k=10$	13/12	15/13	17/14
$k=11$	14/13	15/13	18/15
$k=12$	15/14	17/15	19/16
$k=13$	16/15	18/16	20/17
$k=14$	17/16	18/16	20/17
$k=15$	18/17	20/18	41/35
$k=16$	19/18	21/19	23/20
$k=17$	20/19	21/19	24/21
$k=18$	21/20	23/21	24/21
$k=19$	22/21	24/22	24/21
$k=20$	23/22	24/22	27/24
$k=21$	24/23	26/24	28/25
$k=22$	25/24	27/25	28/25
$k=23$	26/25	27/25	28/25

Az $R_m(\mathrm{LPT}(k))$ értékek

7. Az algoritmuscsalád numerikus vizsgálata

Illusztráció céljából az alábbiakban közlünk néhány futási eredményt, ahol a cikkben szereplő algoritmusokat hasonlítottuk össze. Egy-egy feladatosztályon belül vizsgáltuk az algoritmusok működését, ahol m a gépek számát, n a feladatok számát jelenti, amelyeknek az időtartamát a $[p_1, p_2]$ intervallumból választottuk egyenletes eloszlás szerint, kerekítéssel. Azt vizsgáltuk, hogy az egyes algoritmusok száz esetből hányszor adtak minimális eredményt. Az első táblázat annak illusztrálására szolgál, hogy mi történik, ha a gépek száma rögzített, és a k paramétert növeljük. A felső sorban az LPT algoritmus mellett a k paraméter növekvő értékei szerepelnek.

	m	n	p_1, p_2	LPT	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1.	2	29	9,18	0	2	0	8	8	5	8	69	75
2.	2	21	9,18	0	6	0	12	2	91	10	10	0
3.	2	16	9,18	94	71	96	26	96	91	98	65	98
4.	3	17	9,18	1	1	2	0	2	24	12	23	87
5.	3	29	9,18	1	0	0	4	1	0	1	0	100
6.	3	30	9,18	97	79	60	73	96	14	78	96	76

Mi történik a k paraméter növekedése esetén?

Az első három esetben kettő, a következő három példában három volt a gépek száma. A munkák időtartamát mindegyik esetben a $[9, 18]$ intervallumból választottuk. Az első és negyedik sorban arra látunk példát, amikor az algoritmus hatékonysága a k paraméter növekedésével együtt növekszik. A 2. és 5. példa azt illusztrálja, hogy az előbbi szituáció nem általános: az LPT(k) hatékonyságát erősen befolyásolja a munkák számának és a k paraméternek az oszthatósági viszonya. Amikor k osztója a feladatok számának, LPT(k) hatékonyan meg tudja oldani a feladatot (mint például a 2. sorban $k = 7$ esetén), vagy például akkor is, ha a maradék -1 (5. sor, $k = 10$ esete). Ilyen irányú alaposabb vizsgálatokat (ami a k paraméter és a munkák n számának oszthatósági viszonyait érinti) a jelen cikk keretei között nem végeztünk. A 3. és 6. példa különleges abban, hogy itt minden második, illetve a 6. esetében minden harmadik k-ra majdnem mindig optimális megoldást kapunk, a többi k-ra pedig ezeknél rosszabb megoldásokat.

	m	n	p_1, p_2	LPT	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1.	2	17	9,18	0	0	0	0	0	0	0	100	10
2.	3	20	9,18	0	0	0	0	0	3	3	0	98
3.	4	18	9,18	0	1	0	0	4	9	0	97	28

Egy speciális eset: k értéke a munkák számának a fele.

A második táblázatunk egy érdekes jelenséget mutat. A k paramétert úgy választottuk meg, hogy akkora legyen, mint a munkák számának a fele. Az esetek túlnyomó részében ekkor az algoritmusunk optimális megoldást adott.

Az algoritmus bonyolultsága a k paraméter növelésével exponenciálisan növekszik. A számítógépes realizáció során az algoritmus általános lépésénél, tehát a következő k számú munka helyének a keresésénél korlátozás és szétválasztás típusú algoritmust alkalmaztunk. Ennek során, a következő k számú munka optimális elhelyezése után adódó teljes átfutási időre egy egyszerű felső becslést használtunk: a következő k darab munkának az LPT algoritmus szerinti elhelyezésével adódó

becslést. A program futása például öt gép, $k = 10$ és $n = 30$ munka esetén Pentium 1-es típusú géppel körülbelül négy másodpercig tartott, négy gép esetén pedig körülbelül fél másodpercig. Ez azt mutatja, hogy kis vagy közepes méretű feladat esetében az algoritmus számítógépes futása hamar véget ér. A kapott eredményeket összefoglalva azt mondhatjuk, hogy a k paraméter növelésével az algoritmus hatékonysága bizonyos esetekben jelentősen növekszik, de jelentősen befolyásolja a hatékonyságot a k és n oszthatósági viszonya.

Hivatkozások

[1] Dósa, Gy. and Vizvári, B., Az Általánosított LPT $(k)'$ algoritmus egyforma párhuzamos gépek ütemezésére, *Alkalmazott Matematikai Lapok* **21** (2004) 269–290.

[2] Dósa, Gy., Graham's example is the only tight one for $P\|C_{\max}$, *Annales Univ. Sci. Budapest.* **47** (2004) 143–146.

[3] Graham, R. L., Bounds for certain multiprocessor anomalies, *Bell System Tech. J.* **45** (1966) 1563–1581.

[4] Graham, R. L., Bounds on multiprocessor timing anomalies, *SIAM J. Appl. Math.* **17** (1969) 416–429.

[5] Vizvári, B., Bevezetés a termelésirányítás matematikai elméletébe, *Egyetemi jegyzet* (ELTE, Budapest, 1992).

(Beérkezett: 2001. március 26.)

DÓSA GYÖRGY
VESZPRÉMI EGYETEM
dosagy@almos.vein.hu

VIZVÁRI BÉLA
ELTE OPERÁCIÓKUTATÁSI TANSZÉK
vizvari@cs.elte.hu

THE GENERAL ALGORITHM LPT(k) FOR SCHEDULING IDENTICAL PARALLEL MACHINES

GYÖRGY DÓSA AND BÉLA VIZVÁRI

The paper is devoted to investigate the general algorithm LPT(k). First the tasks are ordered to the LPT order. Then at the same time commonly k tasks are scheduled in a (locally) optimal way (when the increase of the makespan is minimal), and this step is iterated. As our main result, we give the tight value of the theoretical efficiency of the algorithm for every k and every $2 \leq m \leq 4$, where m is the number of machines. Numerical results are also given.

Alkalmazott Matematikai Lapok **23** (2006), 39–57.

ÚJABB STATISZTIKAI VIZSGÁLATOK AZ ORNSTEIN–UHLENBECK-FOLYAMATRÓL I. ELMÉLETI HÁTTÉR

FEGYVERNEKI SÁNDOR

Miskolc

Ebben a dolgozatban azzal a klasszikus problémával foglalkozom, hogy hogyan becsülhetők együtt a stacionárius Gauss–Markov- (Ornstein–Uhlenbeck-) folyamat paraméterei. Ezen folyamat esetén számos probléma adódik a maximum-likelihood becslések eloszlásának vizsgálatával és a paraméterekre adandó konfidenciaintervallumok meghatározásával, ha a csillapítási tényező tart nullához. A cikk célja előkészíteni a szimulációs vizsgálatokhoz szükséges elméleti eredményeket. Bebizonyítom a maximum-likelihood becslés egyértelmű létezését (stacionárius esetben) és megadom a becslés meghatározására szolgáló numerikus módszert. A dolgozat további részében az aszimptotikus esetek vizsgálatához szükséges eloszlásokat, becsléseket és sorfejtéseket adom meg különös tekintettel a végpontokhoz kapcsolódó becslésekre.

1. Bevezetés

A fizikai folyamatok egy jelentős részében a folyamat lefolyását nem a

$$\frac{dx(t)}{dt} = -\lambda x(t) \qquad (\lambda > 0)$$

differenciálegyenlet írja le (melynek megoldása $x = x_0 e^{-\lambda t}$, $x_0 \in \mathbf{R}$), hanem egy ún. sztochasztikus differenciálegyenlet

$$d\xi(t) = -\lambda \xi(t)\, dt + \sigma_w dw(t) \quad \big(E\big(\xi(t)\big) = E\big(w(t)\big) = 0\big), \tag{1}$$

vagy integrálalakban

$$\xi(t) - \xi(t_0) = -\lambda \int_{t_0}^{t} \xi(s) ds + \sigma_w \big(w(t) - w(t_0)\big),$$

ahol $\lambda > 0$ és $\sigma_w > 0$. Továbbá a $\xi(t_0)$ normális eloszlású és független a $w(t)$ standard Wiener-folyamattól, ha $t > t_0$. Ekkor igaz a következő tétel.

1.1. TÉTEL (Arató [1]). *A $\xi(t)$ sztochasztikus folyamat akkor és csak akkor stacionárius Gauss–Markov-folyamat, ha az (1) sztochasztikus differenciálegyenlet megoldása a következő értelemben:*

(i) *Ha $\xi(t)$ folytonos stacionárius Gauss–Markov-folyamat, akkor létezik $\lambda > 0$ és egy Wiener-folyamat, melyre $E\big(w(t)\big) = 0$, $E\big(w^2(t)\big) = t$ úgy, hogy (1) teljesül és*

$$R(t) = E\big(\xi(s+t)\xi(s)\big) = \sigma_\xi^2 e^{-\lambda t} \quad (t > 0), \tag{2}$$

ahol

$$E\big(\xi^2(s)\big) = \sigma_\xi^2 = \frac{\sigma_w^2}{2\lambda}. \tag{3}$$

(ii) *Ha $\lambda > 0$ és $\sigma_w > 0$, akkor csak az (1) $\xi(t)$ folytonos stacionárius megoldása stacionárius Gauss–Markov-folyamat, amelynek kovarianciafüggvényére teljesül (2) és (3). Amikor $\xi(t)$ $t > t_0$ esetén definiált, akkor $\xi(t_0)$ olyan normális eloszlású, amelyre*

$$E\big(\xi(t_0)\big) = 0, \quad E\big(\xi^2(t)\big) = \sigma_\xi^2 = \frac{\sigma_w^2}{2\lambda},$$

és független a $w(t)$ Wiener-folyamattól, ha $t > t_0$. □

Legyen $m \in \mathbf{R}$ és $\eta(t) = \xi(t) + m$, azaz

$$d\big(\eta(t) - m\big) = -\lambda\big(\eta(t) - m\big)\,dt + \sigma_w dw(t). \tag{4}$$

Ekkor az $\eta(t)$-folyamat három paraméter, az m, λ és a σ_w^2 által meghatározott. Feladatunk ezek meghatározása (becslése). A feladat két paraméterre egyszerűsödik a következő állítás alapján.

1.2. TÉTEL (Arató [1]). *Legyen $0 = t_0 < t_1 < \cdots < t_n = T$ egy felosztása a $[0, T]$ intervallumnak, ekkor*

$$\lim_{\max(t_k - t_{k-1}) \to 0} \sum \big[\xi(t_k) - \xi(t_{k-1})\big]^2 = \sigma_w^2 T \quad (1 \text{ valószínűséggel}). \tag{5}$$ □

(5) alapján a σ_w^2 „diffúziós együttható" egyetlen realizáció alapján 1 valószínűséggel meghatározott.

Két feltétel kell ahhoz, hogy a paraméterek jól becsülhetők legyenek egyetlen trajektória alapján. Az egyik az általánosított Markov-tulajdonság, a másik a metrikus tranzitivitás.

A stacionárius Gauss–Markov-folyamat teljesíti a Markov-tulajdonságot. A metrikus tranzitivitás pedig abból adódik, hogy

$$R(t) = \frac{\sigma_w^2}{2\pi} \int_{-\infty}^{+\infty} \frac{e^{iut}}{|\lambda + iu|^2}\,du,$$

így a folyamat spektrál sűrűségfüggvénye

$$\frac{\sigma_w^2}{2\pi}\frac{1}{|\lambda+iu|^2}.$$

A független megfigyeléssorozatok esetén a maximum-likelihood módszer alkalmazásához szükség van a mintaelemek együttes sűrűségfüggvényének az ismeretlen paraméterektől függő seregének meghatározására. A folyamatok statisztikájában ennek felel meg a trajektóriák függvényterén a folyamat által generált, paraméterektől függő mértékseregnek valamilyen standard mértékre vonatkozó Radon–Nikodym-deriváltja.

Gauss-folyamatok esetén érvényes a Feldman–Hájek-féle dichotómia elv: *ugyanazon a téren értelmezett két Gauss-mérték vagy szinguláris, vagy ekvivalens.* Ez a tény azt sugallja, hogy standard mértékül valamilyen analitikusan jól leírható és a vizsgálandó folyamattal egyszerű összefüggésben álló Gauss-mértéket válasszunk.

Az (5) összefüggés alapján tudjuk, hogy az azonos diffúziós együtthatójú stacionárius Gauss–Markov-folyamatok ekvivalens mértékeket generálnak. Továbbá ezek a mértékek ekvivalensek az ugyanilyen szórásnégyzetű Wiener-mértékkel. A $\xi(t)$ $(0 \le t \le T)$, realizációk $\mathbf{R}_\xi$ tere felfogható, mint a $\xi(0)$ valós számegyenes és a $\xi(t)-\xi(0)$ realizációk terének szorzata. Jelölje $\mathbf{W}$ a Wiener-féle mértéket $(x(0), \sigma_w^2)$ paraméterekkel a $0 < t \le T$ intervallumon értelmezett folytonos függvények terén, $\mathbf{L}$ pedig a Lebesgue-mértéket a számegyenesen. Legyen $\mathbf{V} = \mathbf{L} \times \mathbf{W}$. Ha $\mathbf{P}_\xi$ jelöli a $\xi(t)$ stacionárius Gauss–Markov-folyamathoz tartozó mértéket, akkor a $\mathbf{P}_\xi$ mérték abszolút folytonos a $\mathbf{V}$ mértékre nézve, és a $\mathbf{V}$ mérték szerinti Radon–Nikodym-deriváltja:

$$(6)\quad \frac{d\mathbf{P}_\xi}{d\mathbf{V}}(x) = \sqrt{\frac{\lambda}{\pi}}\frac{1}{\sigma_w}\exp\left\{-\frac{\lambda^2}{2\sigma_w^2}\int_0^T x^2(t)\,dt + \frac{\lambda T}{2} - \frac{\lambda}{2\sigma_w^2}\left[x^2(T)+x^2(0)\right]\right\}.$$

Legyen $\xi(t)$ az (1) egyenlet megoldása, míg $\eta(t)$ a (4) egyenleté. Továbbá legyen $\mathbf{P}_0$ és $\mathbf{P}_m$ a $\xi(t)$- és $\eta(t)$-folyamatok által generált mértékek a $[0,T]$ intervallumon folytonos függvények terén, akkor

$$(7)\quad \frac{d\mathbf{P}_m}{d\mathbf{P}_0}(x) = \exp\left\{-\frac{\lambda m}{2\sigma_w^2}\left[x(0)+x(T)+\lambda\int_0^T x(t)\,dt + m\left(1+\frac{\lambda T}{2}\right)\right]\right\}.$$

A (6) és (7) formulákat úgy kell értelmezni, hogy majdnem minden Wiener-trajektóriára megadják a Radon–Nikodym-derivált értékét.

Ha az m és a λ paraméter ismeretlen, akkor a Radon–Nikodym-derivált a következőképpen adható meg:

$$(8)\quad \sqrt{\frac{\lambda}{\pi}}\frac{1}{\sigma_w}\exp\left\{-\frac{\lambda}{\sigma_w^2}\left[s_1^2+\frac{1}{2}\lambda T s_2^2 + (m-m_1)^2 + \frac{\lambda T}{2}(m-m_2)^2 - \frac{1}{2}\sigma_w^2 T\right]\right\},$$

ahol

$$m_1 = \frac{\eta(0) + \eta(T)}{2},$$

$$m_2 = \frac{1}{T}\int_0^T \eta(t)\,dt,$$

$$s_1^2 = \frac{\left[\eta(0) - m_1\right]^2 + \left[\eta(T) - m_1\right]^2}{2} = \frac{\left[\eta(T) - \eta(0)\right]^2}{4},$$

$$s_2^2 = \frac{1}{T}\int_0^T \left[\eta(t) - m_2\right]^2 dt.$$

A (8) formula alapján következik, hogy m_1, m_2, s_1^2 és s_2^2 rendszer egy elégséges statisztika. A maximum-likelihood egyenletek pedig a következők:

$$\frac{\sigma_w^2}{2\lambda}(1 + \lambda T) - s_1^2 - \lambda T s_2^2 - (m - m_1)^2 - \lambda T(m - m_2)^2 = 0, \tag{9}$$

$$2(m - m_1) + \lambda T(m - m_2) = 0. \tag{10}$$

A becslések eloszlásának vizsgálatához meg kell adnunk a karakterisztikus függvényt. A helyzetet bonyolítja, hogy ez csak abban az esetben adott, amikor $m = 0$. Tehát az elégséges statisztikák karakterisztikus függvényéről a következőt tudhatjuk:

$$\psi(u, v, w, z) = E\left(\exp\left[i(um_1 + vs_{01}^2 + wm_2 + zs_{02}^2)\right]\right) = \frac{\sqrt{2\lambda e^{\kappa}\Lambda}}{\sqrt{T\psi(v, z)}}$$

$$\exp\left[\frac{1}{2}\left\{-\frac{uw + w^2}{\Lambda} - \sigma_w^2 T + \left(\frac{iu}{2} - T\sigma_w^2\frac{2vw + \frac{iw\lambda}{\sigma_w^2}}{\Lambda}\right)(\Psi_1 + \Psi_2)\right\}\right],$$

ahol

$$\Psi_1 = \left[\frac{iu\sigma_w^2}{2}\left(1 + e^{-\Lambda}\right) - iw\sigma_w^2\frac{1 - e^{-\Lambda}}{\Lambda}\right]\frac{(\kappa - T\sigma_w^2 iv + \Lambda)e^{\Lambda} - (\kappa - T\sigma_w^2 iv - \Lambda)}{T\psi(v, z)},$$

$$\Psi_2 = \left[\frac{iu\sigma_w^2}{2}\left(1 + e^{\Lambda}\right) - iw\sigma_w^2\frac{1 - e^{\Lambda}}{\Lambda}\right]\frac{(\kappa - T\sigma_w^2 iv + \Lambda) - (\kappa - T\sigma_w^2 iv - \Lambda)e^{-\Lambda}}{T\psi(v, z)},$$

$$\Lambda = \sqrt{\kappa^2 - 2T^2\sigma_w^2 iz}$$

$$\psi(v, z) = \frac{1}{T^2}\left[(\kappa - T\sigma_w^2 iv + \Lambda)^2 e^{\Lambda} - (\kappa - T\sigma_w^2 iv - \Lambda)^2 e^{-\Lambda}\right],$$

$$s_{01}^2 = \frac{[\eta(0) - m]^2 + [\eta(T) - m]^2}{2} \quad \text{és} \quad s_{02}^2 = \frac{1}{T}\int_0^T [\eta(t) - m]^2\, dt.$$

A dolgozat hátralévő része a következőképpen szerveződik. A 2. szakaszban belátom, hogy mindig létezik egyértelmű, pozitív maximum-likelihood becslés a csillapítási tényezőre. Megadok két numerikus eljárást ennek meghatározására. A 3. szakaszban az s_1^2 statisztika eloszlását adom meg. Ez fontos, mert a szakirodalom alapján a reciproka felmerült a κ becsléseként, ha $\kappa \to 0$. Ezenkívül a segítségével könnyen bemutatható, hogy a 0 közelében nincs nullától különböző alsó határa a κ paraméterre készített konfidenciaintervallumnak. A 4. szakaszban néhány, az elégséges statisztika rendszerhez kötődő eloszlás paraméterének a becslését adom meg. Az 5. szakasz pedig az aszimptotikus ($\lambda \to 0$ és $\lambda \to +\infty$) esetek könnyebb megértését és vizsgálatát elősegítő általánosított sorfejtéseket tartalmazza. A szimuláció és az eredmények leírását, kiértékelését a cikk II. része tartalmazza. (Ez utóbbi a folyóirat következő számában jelenik meg – a szerk.)

2. A maximum-likelihood becslés meghatározása

Ha mind a két paraméter ismeretlen, akkor láttuk, hogy a (9)–(10) egyenletrendszer adja meg a maximum-likelihood becsléseket. Vezessük be a $\kappa = \lambda T$ jelölést és átalakítva az egyenletrendszert kapjuk, hogy

$$2(m - m_1) + \kappa(m - m_2) = 0, \tag{11}$$

$$[2s_2^2 + 2(m - m_2)^2]\kappa^2 + [2s_1^2 + 2(m - m_1)^2 - \sigma_w^2 T]\kappa - \sigma_w^2 T = 0. \tag{12}$$

Az első egyenletből az m könnyen kifejezhető:

$$m = \frac{2m_1 + \kappa m_2}{2 + \kappa}. \tag{13}$$

Ennek segítségével κ-ra egy negyedfokú egyenlet adódik:

$$2s_2^2\kappa^4 + [8s_2^2 + 2s_1^2 + 2(m_1 - m_2)^2 - \sigma_w^2 T]\kappa^3 + \tag{14}$$
$$+ [8s_2^2 + 8s_1^2 + 8(m_1 - m_2)^2 - 5\sigma_w^2 T]\kappa^2 + [8s_1^2 - 8\sigma_w^2 T]\kappa - 4\sigma_w^2 T = 0.$$

Számunkra csak a $\kappa > 0$ (stacionárius) eset fogadható el, így meg kell vizsgálnunk a gyökök elhelyezkedését. A következőkben belátjuk, hogy csak egy pozitív gyök van, s megadunk két módszert is, amelyek alkalmasak ennek a gyöknek a meghatározására.

Jelöljük a (13) negyedfokú egyenlet együtthatóit a következőképpen:

$$A = 2s_2^2,$$

$$B = 8s_2^2 + 2s_1^2 + 2(m_1 - m_2)^2 - \sigma_w^2 T,$$

$$rC = 8s_2^2 + 8s_1^2 + 8(m_1 - m_2)^2 - 5\sigma_w^2 T,$$

$$rD = 8s_1^2 - 8\sigma_w^2 T,$$

$$rE = -4\sigma_w^2 T.$$

Ekkor a következő összefüggéseket tudjuk felírni:

$$A > 0,$$

$$4B = 32s_2^2 + 8s_1^2 + 8(m_1 - m_2)^2 - 4\sigma_w^2 T,$$

$$C = 4B - 24s_2^2 - \sigma_w^2 T,$$

$$D = C - 8s_2^2 - 3\sigma_w^2 T - 8(m_1 - m_2)^2,$$

$$E < 0.$$

Továbbá ha $B < 0$, akkor $C < 0$, és ha $C < 0$, akkor $D < 0$, azaz rögtön látszik, hogy a (14) egyenletben szereplő polinomnak pontosan egy előjelváltása van. A Descartes-féle előjelszabály alapján az algebrában jól ismert a következő – a gyökök számára vonatkozó – állítás, amely felírható a Pólya–Szegő [11] II. kötet, V. rész, 36–37. feladat alapján.

2.1. TÉTEL (Pólya–Szegő [11]). *Ha z a polinom pozitív zérushelyeinek a száma, w pedig az együtthatókból álló sorozat előjelváltásainak a száma, akkor $w - z$ egy nemnegatív páros szám.* □

Ezzel beláttuk a következő állítást:

2.2. ÁLLÍTÁS. *A (14) negyedfokú egyenletnek pontosan egy pozitív gyöke van, azaz a stacionárius megoldás egyértelmű.* □

A negyedfokú polinom gyökei meghatározhatók a jól ismert algebrai módszerrel, ha az együtthatók pontosan adottak. Mivel nekünk nincs szükségünk az összes gyökre, így egyszerűbb a numerikus közelítés.

Szidarovszky [12] dolgozata alapján gyökkereső eljárásként azonnal alkalmazható a Newton-módszer. Mivel számunkra csak a (14) egyenlet legnagyobb gyöke érdekes, így az eljárás a következő:

Legyen α_n az

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \quad (a_n \neq 0),$$

polinom legnagyobb gyöke és $x_0 > \alpha_n$. Az x_0 kezdeti közelítésből elkészítjük az

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Newton sorozatot. Legyen $\alpha_n^{(k)} = x_k - \alpha_n$. A [12] dolgozat állításaiból következik, hogy

$$\alpha_n^{(k)} > 0 \quad \text{(minden} \quad k \in \mathbf{N} \quad \text{esetén)},$$

$$\alpha_n^{(k)} \leq \left(\frac{n-1}{n}\right)^k (x_0 - \alpha_n), \quad \text{azaz} \quad x_k \to \alpha_n,$$

$$\alpha_n^{(k)} \leq (n-1)(x_{k-1} - x_k),$$

mely utóbbi becslés nem élesíthető általános esetben.

Az x_0 érték, mint a polinom gyökeinek felső korlátja sokféleképpen meghatározható. Lagrange-tételként ismert a következő állítás a polinom pozitív gyökeinek egy felső korlátjának a meghatározására.

2.3. TÉTEL (Kuros [10]). *Legyen az*

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

valós együtthatós polinom, ahol $a_n > 0$. *Legyen továbbá* a_k $(k \leq n-1)$ *az első negatív együttható. Legyen végül* B *a negatív együtthatók abszolút értékei közül a legnagyobb. Ekkor az*

$$1 + \sqrt[(n-k)]{\frac{B}{a_n}}$$

szám az $f(x)$ *polinom pozitív gyökeinek felső korlátja.* □

Megjegyzés. Ha nincs negatív együttható, akkor nincs pozitív gyök. Ez, mint láttuk, a mi esetünkben nem fordulhat elő.

Habár ezek alapján a maximum-likelihood becslés már meghatározható, mégis vizsgáljuk meg a (14) egyenlet negyedfokú polinomját, mint függvényt, mert látni fogjuk, hogy közvetlenül is belátható, hogy csak egy pozitív gyöke van az egyenletnek. Egyszerű átalakítások után értékes összefüggések derülnek ki, amelyek fontosak a szimulációs eredmények feldolgozásánál. Továbbá megadunk egy másik módszert a stacionárius gyök meghatározására.

Legyen

$$f(x) = 2s_2^2 x^4 + \left[8s_2^2 + 2s_1^2 + 2(m_1 - m_2)^2 - \sigma_w^2 T\right]x^3 +$$
$$+ \left[8s_2^2 + 8s_1^2 + 8(m_1 - m_2)^2 - 5\sigma_w^2 T\right]x^2 + \left[8s_1^2 - 8\sigma_w^2 T\right]x - 4\sigma_w^2 T.$$

Abból, hogy $f(0) = -4\sigma_w^2 T$, tudjuk, hogy létezik pozitív gyök (a főegyüttható pozitív). Továbbá vegyük észre a (11)–(12) egyenletrendszerben, ha $m_1 = m_2$, akkor egyrészt $m = m_1 = m_2$, másrészt ekkor κ-ra az egyenlet másodfokúra redukálható. Ebben az esetben a $\kappa = -2$ kétszeres gyöke a negyedfokú egyenletnek. Ezenkívül

$$f(-2) = 16(m_1 - m_2)^2.$$

Ezek adták azt az ötletet, hogy rendezzük át függvényünket a következő alakba:

$$f(x) = \left\{2s_2^2x^2 + \left[2s_1^2 + 2(m_1 - m_2)^2 - \sigma_w^2 T\right]x - \sigma_w^2 T\right\}(x+2)^2 - 8(m_1 - m_2)^2 x.$$

Ebből is jól látszik, hogy

(1) ha $m_1 = m_2$, akkor a -2 kétszeres gyök, és ezenkívül létezik egy pozitív és egy negatív gyök is;

(2) ha $m_1 \neq m_2$, akkor az $f(x) = 0$ egyenlet átrendezhető a következőképpen:

$$\frac{x}{(x+2)^2} = \frac{1}{8(m_1 - m_2)^2}\left\{2s_2^2x^2 + \left[2s_1^2 + 2(m_1 - m_2)^2 - \sigma_w^2 T\right]x - \sigma_w^2 T\right\}. \tag{15}$$

A (15) egyenlet bal oldala könnyen ábrázolható, hiszen a következő jellemzők állapíthatók meg:

Racionális törtfüggvény, amelynek kétszeres pólus helye van -2-nél, zérushelye 0-nál, maximuma van 2-nél és a maximum $\frac{1}{8}$, inflexiós helye van 4-nél, a $(-\infty, 4]$ intervallumon alulról konkáv, míg a $[4, +\infty)$ intervallumon konvex. A határértéke $\pm\infty$ esetén 0 (1. ábra). A függvényvizsgálat alapján jól látható, hogy a függvény egy viszonylag szűk intervallumban változik, ha $x > 0$.

A (15) egyenlet jobb oldalának az alakja is könnyen meghatározható, hiszen egy parabola, amelynek főegyütthatója pozitív, míg a konstans negatív, így pozitív és negatív x-tengelymetszete is van.

Ez viszont azt jelenti, hogy az $x > 0$ félsíkon pontosan egy metszéspontja van a két függvénynek.

A pozitív gyök meghatározására a következő iterációt is használhatjuk:

$$m^{(0)} = m_2,$$

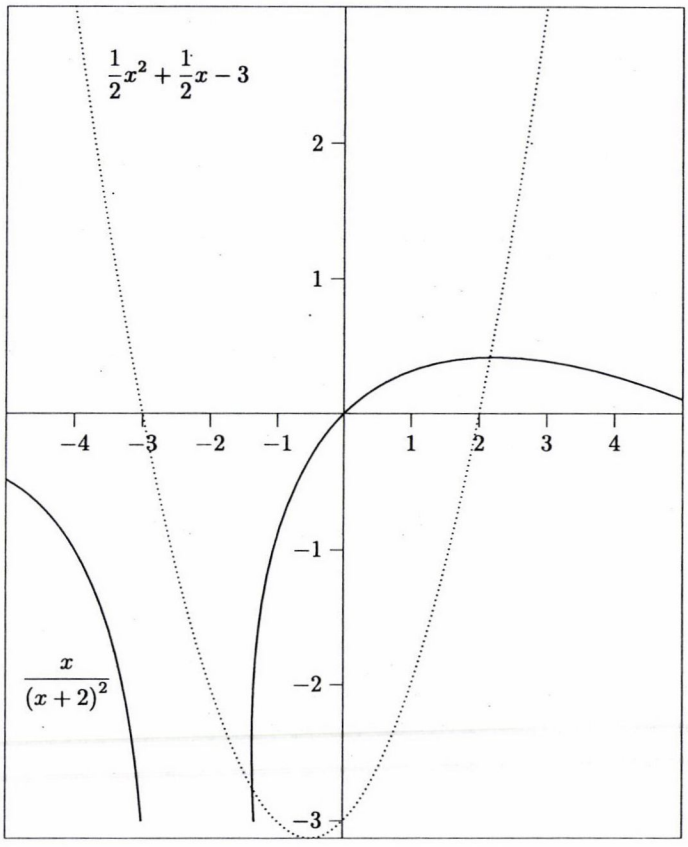
$$\kappa^{(k)} = \frac{\sigma_w^2 T - 2s_1^2 - 2(m^{(k)} - m_1)^2 +}{2\left[2s_2^2 + 2(m^{(k)} - m_2)^2\right]} +$$

$$+ \frac{\sqrt{\left[2s_1^2 + 2(m^{(k)} - m_1)^2 - \sigma_w^2 T\right]^2 + 8\sigma_w^2 T\left[s_2^2 + (m^{(k)} - m_2)^2\right]}}{2\left[2s_2^2 + 2(m^{(k)} - m_2)^2\right]},$$

$$m^{(k+1)} = \frac{m_1 + \kappa^{(k)} m_2}{2 + \kappa^{(k)}}.$$

Ez a módszer egyszerűen az első egyenletből kifejezett m értéket behelyettesítve megoldja a másodfokú egyenletet κ-ra, s ezt folytatja, amíg valamilyen megállítási szabály alapján eléri a megfelelő pontosságot. A továbbiakban „gyökös" iterációként hivatkozunk rá.

A szimuláció során összehasonlítjuk a két módszer esetén az iterációk számát és a gyorsaságot. Itt az összehasonlításból csak annyit emelünk ki, hogy a szimuláció



1. ábra.

során alkalmazott pontosság mellett a két módszer által meghatározott gyökközelítések eltérése kisebb volt, mint a pontosság. Továbbá a számolási idő – a használt számítógépes körülmények mellett – csak akkor volt összehasonlítható, ha ugyanazt a gyökkeresést 10^5 alkalommal megismételtük.

3. Az elégséges statisztikák eloszlásáról

Legyen a (ξ_1, ξ_2) véletlen vektor sűrűségfüggvénye

$$f(x_1, x_2) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \exp\left\{-\frac{(x_1-m)^2 - 2\rho(x_1-m)(x_2-m) + (x_2-m)^2}{2\sigma^2(1-\rho^2)}\right\}.$$

Tehát a (ξ_1, ξ_2) véletlen vektor normális eloszlású, ahol

$$E(\xi_1) = E(\xi_2) = m, \quad D^2(\xi_1) = D^2(\xi_2) = \sigma^2 \quad \text{és} \quad r(\xi_1, \xi_2) = \rho.$$

Készítsük el a ξ_1, ξ_2 mintaelemekből (nem függetlenek) az átlagot (jelölje η_1) és a tapasztalati szórásnégyzet kétszeresét (jelölje η_2), azaz

$$\eta_1 = \frac{\xi_1 + \xi_2}{2}, \qquad \eta_2 = \frac{(\xi_1 - \xi_2)^2}{2}.$$

Az (η_1, η_2) véletlen vektor sűrűségfüggvényének meghatározásához a megfelelő transzformációk:

$$y_1 = \frac{x_1 + x_2}{2}, \qquad y_2 = \frac{(x_1 - x_2)^2}{2}.$$

Ez a transzformáció leképezi az $A = \mathbf{R}^2$ halmazt a $B = \{(y_1, y_2) \mid y_1 \in \mathbf{R}, y_2 \geq 0\}$ halmazra. De ez a transzformáció nem egyértelmű, és a B halmaz minden elemének (kivéve, amikor $y_2 = 0$) két elem felel meg az A halmazban. Így az inverzfüggvények két csoportja tartozik a transzformációhoz:

$$x_1 = y_1 - \sqrt{\frac{y_2}{2}}, \qquad x_1 = y_1 + \sqrt{\frac{y_2}{2}},$$

$$x_2 = y_1 + \sqrt{\frac{y_2}{2}}, \qquad x_2 = y_1 - \sqrt{\frac{y_2}{2}}.$$

Továbbá az A halmaz nem írható fel két olyan diszjunkt halmaz uniójaként, amelyek mindegyike esetén a transzformációnk a B halmazra képez. A problémát az A halmaznak azok a pontjai okozzák, amelyek az $x_1 = x_2$ egyenletű egyenesen fekszenek. Ekkor ugyanis $y_2 = 0$. Azonban mondhatjuk azt, hogy $f(x_1, x_2) = 0$ minden olyan pontban, ahol $x_1 = x_2$. Ezt megtehetjük anélkül, hogy az eloszlás megváltozna, hiszen ezen pontok valószínűsége 0.

Legyen tehát $A = \mathbf{R}^2 \backslash \{(x_1, x_2) \mid x_1 = x_2\}$. Ez a mintatér az

$$A_1 = \{(x_1, x_2) \mid x_2 < x_1\} \quad \text{és} \quad A_2 = \{(x_1, x_2) \mid x_2 > x_1\}$$

diszjunkt halmazok uniója. Ekkor a transzformációnk kölcsönösen egyértelmű az A_i $(i = 1, 2)$ halmazok mindegyikéről az új

$$B = \{(y_1, y_2) \mid y_1 \in \mathbf{R},\ y_2 > 0\},$$

halmazra. Az (η_1, η_2) véletlen vektor sűrűségfüggvénye most már meghatározható. A transzformációk Jacobi-determinánsaira teljesül, hogy

$$|J_1| = |J_2| = \frac{1}{\sqrt{2y_2}}.$$

Tehát az (η_1, η_2) véletlen vektor sűrűségfüggvénye

$$g(y_1, y_2) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \frac{2}{\sqrt{2y_2}} \exp\left\{-\frac{(2-2\rho)(y_1-m)^2 + (1+\rho)y_2}{2\sigma^2(1-\rho^2)}\right\} =$$

$$= \frac{\sqrt{2}}{\sqrt{2\pi}\sigma\sqrt{1+\rho}} \exp\left\{-\frac{(y_1-m)^2}{\sigma^2(1+\rho)}\right\} \frac{1}{\sqrt{2\pi}\sigma\sqrt{1-\rho}} \frac{1}{\sqrt{y_2}} \exp\left\{-\frac{y_2}{2\sigma^2(1-\rho)}\right\}.$$

Ebből jól látható, hogy η_1 és η_2 sztochasztikusan függetlenek (ez ismert független valószínűségi változók esetére). Továbbá η_1 eloszlása normális, amelyre

$$E(\eta_1) = m \quad \text{és} \quad D^2(\eta_1) = \frac{\sigma^2(1+\rho)}{2}.$$

Míg

$$\frac{\eta_2}{\sigma^2(1-\rho)},$$

eloszlása 1-szabadságfokú χ^2.

Legyen $\eta = a\xi$ $(a > 0)$, ekkor az eloszlásfüggvényekre, illetve a sűrűségfüggvényekre teljesül, hogy

$$F_\eta(x) = F_\xi\left(\frac{x}{a}\right), \quad f_\eta(x) = \frac{1}{a} f_\xi\left(\frac{x}{a}\right),$$

azaz ha

$$f_\xi(x) = \frac{1}{\Gamma\left(\frac{1}{2}\right)\sqrt{2}} \frac{1}{\sqrt{x}} e^{-\frac{x}{2}},$$

ahol $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$, és ha $a = \sigma^2(1-\rho)$, akkor

$$f_\eta(x) = \frac{1}{\sqrt{2\pi\sigma^2(1-\rho)x}} \exp\left\{-\frac{x}{2\sigma^2(1-\rho)}\right\}.$$

Továbbá ez azt jelenti, hogy

$$\frac{\eta_2}{2} = \frac{(\xi_1 - \xi_2)^2}{4},$$

sűrűségfüggvénye

$$\frac{1}{\Gamma\left(\frac{1}{2}\right)\sigma\sqrt{1-\rho}\sqrt{x}}\exp\left\{-\frac{y_2}{\sigma^2(1-\rho)}\right\}.$$

A függetlenség és a normális, illetve a χ^2-eloszlás karakterisztikus függvénye alapján az (η_1, η_2) véletlen vektor karakterisztikus függvénye

$$E\big(\exp\{it_1\eta_1 + it_2\eta_2\}\big) = \frac{1}{\sqrt{1-2it_2\sigma^2(1-\rho)}}\exp\left\{it_1 m - \frac{t_1^2\sigma^2(1+\rho)}{4}\right\}.$$

A Radon–Nikodym-deriváltakból adódó elégséges statisztikák között előfordul a

$$\frac{{\xi_1}^2 + {\xi_2}^2}{2} = \left(\frac{\xi_1+\xi_2}{2}\right)^2 + \left(\frac{\xi_1-\xi_2}{2}\right)^2 = {\eta_1}^2 + \frac{\eta_2}{2}.$$

Az eddigiek alapján viszonylag gyorsan meghatározható

$$\left(\frac{\xi_1+\xi_2}{2}, \frac{{\xi_1}^2+{\xi_2}^2}{2}\right)$$

együttes karakterisztikus függvénye. Az egyszerűség kedvéért legyen $m = 0$ (az elégséges statisztikák együttes karakterisztikus függvénye is ebben az esetben adott a differenciálegyenlet módszer alapján). Tehát

$$\varphi(t_1,t_2) = E\left(\exp\left\{it_1\eta_1 + it_2\left({\eta_1}^2 + \frac{\eta_2}{2}\right)\right\}\right) =$$

$$= \int_{-\infty}^{+\infty}\int_{0}^{+\infty} e^{it_1y_1+it_2(y_1^2+y_2)}\frac{1}{\pi\sigma^2\sqrt{1-\rho^2}\sqrt{y_2}}e^{-\frac{y_1^2}{\sigma^2(1+\rho)}-\frac{y_2}{\sigma^2(1-\rho)}}\,dy_2\,dy_1 =$$

$$= \frac{1}{\sqrt{1-it_2\sigma^2(1-\rho)}}\int_{-\infty}^{+\infty} e^{it_1y_1+it_2y_1^2}\frac{1}{\sqrt{\pi}\sigma\sqrt{1+\rho}}e^{-\frac{y_1^2}{\sigma^2(1+\rho)}}\,dy_1 =$$

$$= \frac{1}{\sqrt{1-it_2\sigma^2(1-\rho)}}\int_{-\infty}^{+\infty}\frac{1}{\sqrt{\pi}\sigma\sqrt{1+\rho}}e^{it_1y_1+it_2y_1^2-\frac{y_1^2}{\sigma^2(1+\rho)}}\,dy_1.$$

Alakítsuk át az integrandusz kitevőjében lévő kifejezést a következőképpen:

$$it_1y_1 + it_2y_1^2 - \frac{y_1^2}{\sigma^2(1+\rho)} =$$

$$= -\frac{1}{\sigma^2(1+\rho)}\left[y_1^2 - it_2\sigma^2(1+\rho)y_1^2 - it_1\sigma^2(1+\rho)y_1\right] =$$

$$= -\frac{1}{\sigma^2(1+\rho)}\left[\left(1 - it_2\sigma^2(1+\rho)\right)y_1^2 - it_1\sigma^2(1+\rho)y_1\right] =$$

$$= -\frac{1-it_2\sigma^2(1+\rho)}{\sigma^2(1+\rho)}\left[y_1^2 - \frac{it_1\sigma^2(1+\rho)}{1-it_2\sigma^2(1+\rho)}y_1\right] =$$

$$= -\frac{1-it_2\sigma^2(1+\rho)}{\sigma^2(1+\rho)}\left[\left(y_1 - \frac{it_1\sigma^2(1+\rho)}{2-2it_2\sigma^2(1+\rho)}\right)^2 - \left(\frac{it_1\sigma^2(1+\rho)}{2-2it_2\sigma^2(1+\rho)}\right)^2\right].$$

Ezen átalakításokat és azt felhasználva, hogy a

$$h(y_1) = \sqrt{\frac{1-it_2\sigma^2(1+\rho)}{\pi\sigma^2(1+\rho)}}\exp\left\{-\frac{1-it_2\sigma^2(1+\rho)}{\sigma^2(1+\rho)}\left[\left(y_1 - \frac{it_1\sigma^2(1+\rho)}{2-2it_2\sigma^2(1+\rho)}\right)^2\right]\right\}$$

függvény formálisan egy olyan normális eloszlás sűrűségfüggvényének tekinthető, amelynek várható értéke

$$\frac{it_1\sigma^2(1+\rho)}{2-2it_2\sigma^2(1+\rho)},$$

míg szórásnégyzete

$$\frac{\sigma^2(1+\rho)}{2-2it_2\sigma^2(1+\rho)},$$

s így

$$\int_{-\infty}^{+\infty} h(y_1)\,dy_1 = 1.$$

Tehát a karakterisztikus függvény

$$(16)\quad \varphi(t_1,t_2) = \frac{1}{\sqrt{1-it_2\sigma^2(1-\rho)}}\,\frac{1}{\sqrt{1-it_2\sigma^2(1+\rho)}}\exp\left\{\frac{-t_1^2\sigma^2(1+\rho)}{4-4it_2\sigma^2(1+\rho)}\right\}.$$

A 3. szakasz eredményeit a következőképpen összegezhetjük.

3.1. ÁLLÍTÁS. *Ha adott két korrelált, normális eloszlású valószínűségi változó, akkor a belőlük képzett számtani átlag és tapasztalati szórásnégyzet független. Továbbá a tapasztalati szórásnégyzet eloszlása egy olyan 1-szabadságfokú χ^2 eloszlás, melynek a várható értéke*

$$\frac{\sigma^2(1-\rho)}{2}.$$

Ezenkívül a két valószínűségi változó négyzetösszege felbontható két olyan független 1-szabadságfokú χ^2-eloszlású valószínűségi változó összegére, melyek várható értéke

$$\sigma^2(1-\rho), \qquad \textit{illetve} \qquad \sigma^2(1+\rho). \qquad \square$$

Megjegyzés. Ha m tetszőleges, akkor a (16) karakterisztikus függvény a következő:

$$\frac{1}{\sqrt{1-it_2\sigma^2(1-\rho)}}\frac{1}{\sqrt{1-it_2c}}\exp\left\{-\frac{1}{c}\left(m^2-\frac{(2m+it_1c)^2}{4-4it_2c}\right)\right\},$$

ahol $c=\sigma^2(1+\rho)$. A $t_1=0$ esetén pedig

$$\frac{1}{\sqrt{1-it_2\sigma^2(1-\rho)}}\frac{1}{\sqrt{1-it_2\sigma^2(1+\rho)}}\exp\left\{\frac{it_2m^2}{1-it_2\sigma^2(1+\rho)}\right\},$$

azaz a két valószínűségi változó négyzetösszege felbontható olyan centrális és nemcentrális 1-szabadságfokú χ^2-eloszlású valószínűségi változók összegére, amelyek függetlenek.

4. A paraméterek becslése

Az eddigiek alapján látható, hogy a λ paraméter becsléseinek vizsgálatához szükségünk van az

$$(m_1-m_2)^2,\quad s_1^2,\quad s_2^2$$

statisztikákhoz kötődő eloszlások paramétereinek a becslésére. A 3. szakasz alapján tudjuk, hogy s_1^2 eloszlása mindig 1-szabadságfokú χ^2, míg Cox [5] szerint, ha a három paraméter mindegyike ismeretlen, akkor az s_2^2 a Karhunen–Loeve sorfejtés alapján reprezentálható, mint független nemcentrális χ^2 eloszlású valószínűségi változók súlyozott összege, ezért ebben a szakaszban összefoglaljuk szimulációk kiértékeléséhez, az aszimptotikus esetek vizsgálatához szükséges fogalmakat és eredményeket.

A szimuláció statisztikai vizsgálatához szükséges a χ_1^2-eloszlású valószínűségi változó reciprokának az eloszlása és jellemzése.

Legyen ξ χ_1^2-eloszlású és $\eta=\frac{1}{\xi}$, ekkor az eloszlásfüggvény, ha $x>0$,

$$F_\eta(x)=2-2\Phi\left(\sqrt{\frac{1}{x}}\right),$$

ahol Φ a standard normális eloszlás eloszlásfüggvénye.

Ezután meghatározzuk a skálaparaméter maximum-likelihood becslését:

Tehát a $\xi_1,\xi_2,\ldots,\xi_n$ független minta, amelyhez tartozó eloszlásfüggvény

$$F(x)=2-2\Phi\left(\sqrt{\frac{\sigma}{x}}\right),\quad \text{ha}\quad x>0.$$

A megfelelő sűrűségfüggvény

$$f(x) = \frac{\sqrt{\sigma}}{x\sqrt{2\pi x}} e^{-\frac{\sigma}{2x}},$$

amelyből a likelihood függvény

$$L(\xi_1, \xi_2, \ldots, \xi_n; \sigma) = -\sum_{i=1}^{n} \ln f(\xi_i) = -\frac{n}{2} \ln \sigma - \frac{3}{2} \sum_{i=1}^{n} \ln \xi_i + \frac{n}{2} \ln(2\pi) + \frac{\sigma}{2} \sum_{i=1}^{n} \frac{1}{\xi_i}.$$

Ebből

$$\frac{dL}{d\sigma} = -\frac{n}{2\sigma} + \sum_{i=1}^{n} \frac{1}{2\xi_i} = 0,$$

amelyből a skálaparaméter maximum-likelihood becslése

$$\hat{\sigma} = \frac{n}{\sum_{i=1}^{n} \frac{1}{\xi_i}}.$$

Cox [5] alapján tudjuk, hogy az s_2^2 statisztika esetében a nemcentrális χ^2-eloszlás paramétereinek a becslésével kell foglalkoznunk. A nemcentrális χ^2-eloszlás tulajdonságainak, illetve a paraméterbecsléseknek egy kitűnő összefoglalása található a [9] könyvben. A paraméterek becslése során számos probléma vetődik fel, viszont számunkra az a fontos eset, amikor a szabadsági fok $r = 1$, és

$$E(\xi) = \mu, \quad D^2(\xi) = \sigma^2 \quad \text{és} \quad \eta = \xi^2,$$

ekkor az eloszlásfüggvény, a sűrűségfüggvény, a várható érték és a szórásnégyzet a következő:

$$F_\eta(x) = \Phi\left(\frac{\sqrt{x} - \mu}{\sigma}\right) - \Phi\left(\frac{-\sqrt{x} - \mu}{\sigma}\right) \quad (x > 0),$$

$$f_\eta(x) = \frac{1}{2\sqrt{x}\sigma}\left[\varphi\left(\frac{\sqrt{x} - \mu}{\sigma}\right) + \varphi\left(\frac{-\sqrt{x} - \mu}{\sigma}\right)\right] \quad (x > 0),$$

$$E(\eta) = \mu^2 + \sigma^2,$$

$$D^2(\eta) = \sigma^2(4\mu^2 + 2\sigma^2),$$

ahol φ a standard normális eloszlás sűrűségfüggvénye.

A vizsgálatok azt mutatják, hogy már ekkor sem egyszerű numerikusan meghatározni μ és σ maximum-likelihood becslését (l. [9]). Viszont a momentumok

módszerének hatásfoka „kicsi", ha a $\vartheta = \dfrac{\mu^2}{\sigma^2}$ nemcentrálási paraméter közel van az egyhez vagy kisebb egynél. Egyszerűbb meghatározni a paramétereket, ha a

$$\sqrt{\eta} = |\sigma\xi + \mu| \tag{17}$$

eloszlásának paramétereit határozzuk meg. Ekkor a maximum-likelihood becsléseket meghatározó egyenletrendszer felírható a következő alakban:

$$\hat{\mu}^2 + \hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m} \eta_i^2,$$

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} \eta_i \operatorname{th}\left(\frac{\hat{\mu}\eta_i}{\hat{\sigma}^2}\right),$$

ahol $\eta_1, \eta_2, \ldots, \eta_m$ független minta eloszlása megegyezik a (17) valószínűségi változó eloszlásával. Az egyenletrendszer megoldására pedig a [6] cikkben leírt „ping-pong" algoritmust használjuk.

A szimulációs vizsgálataink egyik iránya, hogy a λ milyen „kis" értékei mellett használható a χ_1^2-eloszlás a konfidenciaintervallum készítésére. Hiszen ha λ értéke „nagy", akkor a határeloszlás-tételek szerint a normális eloszlás, míg a köztes értékekre a [2] cikkben megadott táblázatot használjuk.

Ha egy sztochasztikus modell csak közelítőleg teljesül, azaz ha a mintaelemek eloszlása csak megközelítően ismert, akkor szokás ún. robusztus módszereket alkalmazni. A [6] cikkben egy olyan robusztus paraméterbecslési eljárást adtam meg egy eloszlástípus hely- és skálaparaméterére, amely többek között jól alkalmazható a χ_1^2 típus reciprokának a vizsgálatára is. Továbbá jól használható a nemcentrális χ_1^2 esetén is. Ennek rövid leírása a szimulációs eredmények mellett található (a cikk II. részében).

5. Sorfejtések közel nemstacionárius esetben

A (15) egyenlet bal oldalán lévő függvény nagyban segíti az aszimptotikus kiértékelést, így megadjuk az általánosított sorbafejtését:

$$\frac{x}{(x+2)^2} = \begin{cases} \dfrac{1}{4}x - \dfrac{1}{4}x^2 + \dfrac{3}{16}x^3 - \dfrac{1}{8}x^4 + \dfrac{5}{64}x^5 + O(x^6), & \text{ha } x \to 0, \\ \dfrac{1}{x} - \dfrac{4}{x^2} + \dfrac{12}{x^3} - \dfrac{32}{x^4} + O\left(\dfrac{1}{x^5}\right), & \text{ha } x \to +\infty. \end{cases}$$

Az elégséges statisztikák várható értékének és szórásnégyzetének általánosított sorbafejtése és az ebből adódó határértékek, ha $\kappa \to 0$, illetve $\kappa \to +\infty$:

$$E(m_1^2) = \begin{cases} \dfrac{1}{2\kappa} - \dfrac{1}{4} + \dfrac{1}{8}\kappa - \dfrac{1}{24}\kappa^2 + O(\kappa^3), & \text{ha } \kappa \to 0, \\ \dfrac{1}{4\kappa} + \dfrac{1}{4\kappa \exp(\kappa)}, & \text{ha } \kappa \to +\infty. \end{cases}$$

$$E(m_2^2) = \begin{cases} \dfrac{1}{2\kappa} - \dfrac{1}{6} + \dfrac{1}{24}\kappa - \dfrac{1}{120}\kappa^2 + O(\kappa^3), & \text{ha } \kappa \to 0, \\ \dfrac{1}{\kappa^2} - \dfrac{1}{\kappa^3} + \dfrac{1}{\kappa^3 \exp(\kappa)}, & \text{ha } \kappa \to +\infty. \end{cases}$$

$$E(m_1 m_2) = \begin{cases} \dfrac{1}{2\kappa} - \dfrac{1}{4} + \dfrac{1}{12}\kappa - \dfrac{1}{48}\kappa^2 + O(\kappa^3), & \text{ha } \kappa \to 0, \\ \dfrac{1}{2\kappa^2} - \dfrac{1}{2\kappa^2 \exp(\kappa)}, & \text{ha } \kappa \to +\infty. \end{cases}$$

$$E((m_1 - m_2)^2) = \begin{cases} \dfrac{1}{12} - \dfrac{1}{120}\kappa^2 + O(\kappa^3), & \text{ha } \kappa \to 0, \\ \dfrac{1}{4\kappa} - \dfrac{1}{\kappa^3} + \left(\dfrac{1}{4\kappa} + \dfrac{1}{\kappa^2} + \dfrac{1}{\kappa^3}\right)\dfrac{1}{\exp(\kappa)}, & \text{ha } \kappa \to +\infty. \end{cases}$$

$$E(s_1^2) = \begin{cases} \dfrac{1}{4} - \dfrac{1}{8}\kappa + \dfrac{1}{24}\kappa^2 + O(\kappa^3), & \text{ha } \kappa \to 0, \\ \dfrac{1}{4\kappa} - \dfrac{1}{4\kappa \exp(\kappa)}, & \text{ha } \kappa \to +\infty, \end{cases}$$

$$E(s_2^2) = \begin{cases} \dfrac{1}{6} - \dfrac{1}{24}\kappa + \dfrac{1}{120}\kappa^2 + O(\kappa^3), & \text{ha } \kappa \to 0, \\ \dfrac{1}{2\kappa} - \dfrac{1}{\kappa^2} + \dfrac{1}{\kappa^3} - \dfrac{1}{\kappa^3 \exp(\kappa)}, & \text{ha } \kappa \to +\infty. \end{cases}$$

$$D^2(s_2^2) = \begin{cases} \dfrac{1}{45} - \dfrac{1}{60}\kappa + O(\kappa^2), & \text{ha } \kappa \to 0, \\ \dfrac{1}{2\kappa^3} - \dfrac{9}{4\kappa^4} + \dfrac{3}{\kappa^5} + O\left(\dfrac{1}{\kappa^4 \exp(\kappa)}\right), & \text{ha } \kappa \to +\infty. \end{cases}$$

A 3. szakaszban ismertetett eredmények alapján látható, hogy érdemes megadni a $\sigma^2(1-\rho)$ és $\sigma^2(1+\rho)$ értékekhez kötődő sorfejtéseket is, ha

$$\sigma^2 = \frac{1}{2\lambda}, \quad \text{és} \quad \rho = e^{-\lambda}.$$

$$\frac{1-e^{-\lambda}}{2\lambda} = \begin{cases} \frac{1}{2} - \frac{1}{4}\lambda + \frac{1}{12}\lambda^2 + O(\lambda^3), & \text{ha } \lambda \to 0, \\ \frac{1}{2\lambda} - \frac{1}{2\lambda \exp(\lambda)}, & \text{ha } \lambda \to +\infty. \end{cases}$$

$$\frac{1+e^{-\lambda}}{2\lambda} = \begin{cases} \frac{1}{\lambda} - \frac{1}{2} - \frac{1}{4}\lambda - \frac{1}{12}\lambda^2 + O(\lambda^3), & \text{ha } \lambda \to 0, \\ \frac{1}{2\lambda} + \frac{1}{2\lambda \exp(\lambda)}, & \text{ha } \lambda \to +\infty. \end{cases}$$

Irodalomjegyzék

[1] Arató, M., *Linear stochastic systems with constant coefficients* (Springer-Verlag, Berlin, 1982).

[2] Arató, M., Kuki, A. and Szabó, A., Exact Distribution of Estimators of Parameters in Ornstein–Uhlenbeck Processes, *Computers Math. Applic.* **31** (1996), 45–54.

[3] Baxter, G., A strong limit theorem for Gaussian process, *Proc. Amer. Math. Soc.* **7** (1956), 522–525.

[4] Chan, N. H. and Wei, C. Z., Asymptotic inference for nearly nonstationary $AR(1)$, *Ann. Statist.* **15** (1987), 1050–1063.

[5] Cox, D. D., Gaussian likelihood estimation for nearly nonstationary $AR(1)$ processes, *Ann. Stat.* **19** (1991), 1129–1142.

[6] Fegyverneki, S., A special joint estimation of location and scale with applications, *Publ. Univ. of Miskolc, Series D. Natural Sciences, Mathematics* **39** (1999), 21–27.

[7] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J. and Stahel, W. A., *Robust statistics: the approach based on influence functions* (Wiley, New York, 1986).

[8] Hogg, R. V. and Craig, A. T., *Introduction to mathematical statistics* (MacMillan, New York, 1969).

[9] Johnson, N. L. and Kotz, S., *Distributions in statistics* (Wiley, New York, 1970).

[10] Kuros, A. G., *Felsőbb algebra* (Tankönyvkiadó, Budapest 1967).

[11] Pólya, G. and Szegő, G., *Problems and Theorems in Analysis II* (Springer, New York 1976).

[12] Szidarovszky, F., Valós gyökökkel rendelkező polinomok gyökeinek meghatározása Newton módszerével, *MTA III. Osztály Közleményei* **21** (1972), 63–69.

[13] Striebel, C. T., Densities for stochastic processes, *Ann. Math. Stat.* **30** (1959), 559–667.

(Beérkezett: 2001. május 15.)

FEGYVERNEKI SÁNDOR
MATEMATIKAI INTÉZET
MISKOLCI EGYETEM
3515 MISKOLC
MISKOLC-EGYETEMVÁROS
matfs@gold.uni-miskolc.hu

NEW STATISTICAL INVESTIGATIONS OF ORNSTEIN–UHLENBECK PROCESS. I. THEORETICAL BACKGROUND

SÁNDOR FEGYVERNEKI

An asymptotic analysis is presented for estimation in the three-parameter Ornstein–Uhlenbeck process, where the parameters are the local mean, the drift and the variance coefficient. Section 1 is a short overview about properties of stationary Gauss–Markov process, Radon–Nikodym derivatives, sufficient statistics and their distribution. The maximum likelihood estimate of the parameter vector is a solution of a rather complicated system of equations. In Section 2 we describe the methods for solving maximum-likelihood equations. In the rest of the paper we give some basic results which are necessary to statistical investigations and simulations. We determine the distributions of some simple statistics, that is, the distribution of the mean and variance of the sample where the sample size is two and the elements are correlated and normal distributed. In Section 4 we give the maximum likelihood estimator for the parameter of reciprocal chi-square distribution and noncentral chi-square distribution. At the end of the paper generalized series expansions are given for the expectation and variance of some sufficient statistics.

Alkalmazott Matematikai Lapok **23** (2006), 59–72.

SS-TÍPUSÚ IGAZMONDÓ–HAZUG FEJTÖRŐK GRÁFELMÉLETI MEGKÖZELÍTÉSBEN

NAGY BENEDEK

Debrecen

Olyan igazmondó–hazug fejtörők gráfjait vizsgáljuk részletesen, amikben minden szereplő csak egyszerű mondatokat mondhat valamely szereplő típusáról. A gráf-reprezentációban a tiszta rejtvényekről – amelyekben az elhangzó mondatokon kívül nincs plusz információnk – minden információ benne van. A lehetséges gráfok több érdekes tulajdonságát is bemutatjuk. Megmutatjuk, hogy nincs olyan tiszta egyszerűmondatos igazmondó–hazug fejtörő, aminek egyetlen megoldása van. A gráfban az élek kétféle súlyúak lehetnek, irányításuk viszont nem játszik szerepet. Bevezetjük a maximális és a minimális fejtörők fogalmát, amelyek a teljes gráfokhoz, illetve a feszítőerdőkhöz köthetőek. Mutatunk egy algoritmust, amellyel bármely tiszta rejtvénynek meghatározhatjuk a lehetséges megoldásait a gráf összefüggő komponenseinek vizsgálatával.

1. Bevezetés

A logikai fejtörők mindig érdekelték az embereket, köztük minden korosztály talál a tudásszintjének megfelelőt. R. Smullyan az egyike volt az elsőknek, aki tudományos, logikai szempontból is vizsgálta őket, nagysikerű könyveket írva a témakörben, amikből többet magyar nyelvre is lefordítottak ([4], [5], [6], [7]). Aszalós László [1]-ben, illetve PhD dolgozatában egy, a mesterséges intelligenciában használatos, az ún. tabló-módszerrel, illetve Prolog nyelvű programokkal oldott meg különböző típusú fejtörőket. Ebben a cikkben csak olyan igazmondó–hazug fejtörőkkel fogunk foglalkozni, amelyekben az igazmondók minden atomi állítása igaz, a hazugoknak pedig minden atomi állításuk hamis. Az alapdefiníciók után elkészítjük e rejtvények gráf-reprezentációját, megvizsgáljuk szerkezetüket. A fejtörőket a lehetséges megoldásaikat megtartva bővíthetjük ún. élhozzáadó lépések segítségével, amíg el nem érünk az ezekkel a megoldásokkal rendelkező maximális feladványig. A feladványokat a gráfjaik segítségével, új módszerrel oldjuk meg, amiben az ún. kiértékelő nyilak játszanak jelentős szerepet.

2. Alapdefiníciók

Azért, hogy matematikai szempontból pontosan el tudjuk érni a célunkat, szükségünk van néhány fogalom meghatározására. Néhány definíció egyelőre általánosabb, mint amire most szükségünk lesz, ezekről később még szólunk.

2.1. Definíció. Atomi állításnak (vagy egyszerű mondatnak) nevezünk egy részállításokra nem bontható állítást.

2.2. Definíció. Egy embert (erősen) igazmondónak (az angol Strong szóból röviden: S-truthteller) nevezünk, ha minden atomi állítása igaz, gyengén igazmondónak, ha legalább egy atomi állítása igaz (feltéve, hogy megszólal). Hasonlóan egy ember erősen hazug, ha minden atomi állítása hamis, illetve (gyengén) hazug, amennyiben legalább egy atomi állítása hamis (ha mond egyáltalán valamit).

Könnyen beláthatjuk, hogy aki nem mond semmit, az bármilyen típusú lehet a fentiek közül.

Evidens, hogy az (erősen) igazmondó állításai logikai és művelettel összekötve igazat adnak, a gyengén igazmondó állításait összekapcsolva logikai vagy művelettel szintén igaz az eredmény. Ennek megfelelően az erősen hazug ember állításainak diszjunkciója is hamis, míg egy gyengén hazug állításainak konjunkciója is hamis lesz.

Ebben a cikkben mi csak olyan típusú atomi állításokkal foglalkozunk, ami egy a feladványban szereplő személy igazmondó, illetve hazug voltát állítja, szereplőink pedig erősen igazmondók, illetve erősen hazugok. (Az „erősen" jelzőt innen kezdve nem mindig fogjuk kitenni.) Formálisan tehát:

2.3. Definíció. Egy fejtörőt SS-típusúnak nevezünk, ha benne minden szereplő csak „a j. szereplő hazug", illetve „a j. szereplő igazmondó" alakú kijelentéseket tesz.

Egy fejtörőt tisztának nevezünk, ha a benne szereplők által mondott mondatokon kívül nincs egyéb információnk a megoldáshoz.

Ebben a cikkben tiszta fejtörőkkel fogunk foglalkozni, szemben pl. [8]-cal, ahol olyan programot prezentáltunk, amely nem tiszta fejtörőket is képes generálni.

2.4. Definíció. Egy fejtörő megoldásának hívjuk azt a függvényt, amely a benne szereplő emberek mindegyikéhez hozzárendeli az {igazmondó, hazug} halmaz egyik elemét, és így minden ember által mondott állítás az adott ember típusának megfelelően alakul.

Két megoldást különbözőnek nevezünk, ha legalább egy szereplőhöz nem ugyanazt az értéket rendelik.

2.5. Definíció. Egy feladvány megoldható, ha van megoldása, továbbá egy fejtörőt akkor nevezünk jónak, ha pontosan egy megoldása van.

A jó fejtörőkben minden ember típusa egyértelműen kiderül a megoldás során. Valójában az ilyeneket szeretjük, használjuk feladványokként pl. rejtvényújságok-

ban. Amelyik fejtörőnek nincs megoldása, azokat valójában nem is szoktuk annak tekinteni.

A fejtörőket szemléletessé tehetjük a gráf-reprezentáció segítségével.

2.6. Definíció. Egy fejtörő gráf-reprezentációján a következő irányított gráfot értjük: A gráf csúcsai legyenek a fejtörőben szereplő emberek. Az éleket pedig kétféle nyíllal jelöljük: folytonos nyíllal, ha valaki azt állítja valakiről, hogy igazmondó; és szaggatott vonallal, ha azt állítja, hogy hazug. (A nyíl az állítást tevőtől mutat arra, akiről az állítás szól.)

2.1. Megjegyzés. Tulajdonképpen a kétféle éltípus egy olyan súlyozott gráfot jelenít meg, amelyben minden él kétféle súlyú lehet, célszerű pl. a szaggatott nyílnak 1-es, míg a folytonos nyílnak 2-es súlyt megfeleltetni.

2.2. Megjegyzés. Általános esetben az állításoknak megfelelő logikai formulát diszjunktív normál formájúvá alakítjuk, és a gráf-reprezentációban az egy csúcsból induló „és-éleket" összekötjük. Így a feladvány „és–vagy gráfjában" minden információ benne lesz.

2.7. Definíció. Egy feladvány gráfjában kiértékelő nyílnak hívunk egy élt akkor, ha valamelyik végpontjában levő ember típusa már ismert, és ez alapján meg tudjuk mondani a másik végpontban levő ember típusát.

2.8. Definíció. Két feladványt ekvivalensnek nevezünk, ha gráfjaik izomorfak.

2.9. Definíció. Az P és a Q feladványokat (gyengén) ekvivalensnek nevezünk, ha P megoldása(i) pontosan ugyanaz(ok), mint a Q megoldása(i).

Most nézzünk meg részletesebben a címben is szereplő feladványtípust.

3. Erősen igazmondó – erősen hazug (vagyis SS-típusú) fejtörők

Ebben a cikkben olyan feladványokkal foglalkozunk, amelyben minden igazmondó erősen igazmondó, illetve minden hazug erősen hazug (innen, az erősen szavakból az SS-típus). Tehát a feladvány a következő alakot ölti:

Adott n számú ember, akik mindegyike vagy igazmondó, vagy hazug. Az igazmondók csak igaz egyszerű mondatokat tudnak mondani, a hazugok pedig csak hamisakat. Minden ember a következő kétféle atomi mondatokat mondhatja: az n ember közül valamelyikről azt állítja, hogy igazmondó; vagy pedig azt állítja, hogy hazug. Egy ember több állítást is tehet. (Ekkor minden állítása külön-külön a típusának megfelelő.)

Formálisan:

Jelöljük az i. embert B_i-vel, ekkor B_i tetszőleges B_j emberről a következő állításokat teheti:

(1) B_i azt mondja, hogy B_j igazmondó;

(2) B_i azt mondja, hogy B_j hazug;
(3) – (B_i a B_j-ről nem mond semmit);
ahol $1 \leq i, j \leq n$.

3.1. Megjegyzés. Mivel ebben a fejtörőtípusban minden állítás atomi, ezért helytálló az „egyszerűmondatos" fejtörők elnevezés is.

Ahhoz, hogy behatóbban megvizsgálhassuk e fejtörőtípus lehetséges gráfjait, vezessük be a következő rövidítést. Jelöljük az erősen igamondó típust az I, az erősen hazug típust a H betűvel.

3.1. LEMMA. *Az SS-típusú fejtörők gráfjában (a hurokélektől eltekintve) minden él kiértékelő nyíl.*

Bizonyítás. Mivel minden ezekben a fejtörőkben szereplő ember típusa erős. ezért minden egyes állítása (vagyis minden egyes él) pontosan meghatározza a típusát. Esetekre bontva ez a következőképpen néz ki:

a kiértékelő nyíl	a kiértékelés során megtudjuk a másik ember típusát is	
$I \longrightarrow$	$I \longrightarrow I$	(igaznak kell lennie az állításnak)
$I \dashrightarrow$	$I \dashrightarrow H$	(igaznak kell lennie az állításnak)
$H \longrightarrow$	$H \longrightarrow H$	(hamisnak kell lennie az állításnak)
$H \dashrightarrow$	$H \dashrightarrow I$	(hamisnak kell lennie az állításnak)
$\longrightarrow I$	$I \longrightarrow I$	(igaz az állítás: igazmondó mondta)
$\dashrightarrow I$	$H \dashrightarrow I$	(hamis az állítás: hazug mondta)
$\longrightarrow H$	$H \longrightarrow H$	(hamis az állítás: hazug mondta)
$\dashrightarrow H$	$I \dashrightarrow H$	(igaz az állítás: igazmondó mondta)

Az előbbi lemma akkor nyújthat segítséget, ha a megoldás már részben ismert (pl. ez gyakran előfordul nem tiszta fejtörők esetén).

3.2. Megjegyzés. A kiértékelő nyilakat megfigyelve láthatjuk, hogy a folytonos nyíllal összekötött csúcsok típusa megegyezik, míg a szaggatott nyíllal összekötötteké ellenkező. A nyíl irányítása pedig nem játszik szerepet ezekben a rejtvényekben. Ezért a továbbiakban az ilyen típusú fejtörők gráfjainak éleit nem tekintjük irányítottnak. A nyíl helyett a vonal szót is használjuk.

Ha pl. két különböző kiértékelő nyíl egy csúcsban az I és a H típust is megjeleníti, akkor ellentmondást kapunk, így nem lehet megoldás.

3.2. LEMMA. *Egy megoldható fejtörő gráfjában nem lehet két csúcs mindkét típusú éllel összekötve.*

Bizonyítás. Legyen két csúcs közt mindkétféle él. Ekkor tegyük fel, hogy az egyik ember típusát ismerjük. A másik típusának meg kell egyeznie ezzel, hiszen folytonos él van köztük, másrészt ellenkezőnek kell lennie, hiszen szaggatott él is vezet köztük. Ez ellentmond annak, hogy a feladvány megoldható.

Ennek a lemmának közvetlen következményei a fejtörők nyelvére átfogalmazva a következők.

3.1. KÖVETKEZMÉNY. *Ha egy SS-típusú fejtörőnek van megoldása, akkor benne senki nem állíthatja senkiről azt is, hogy igazmondó és azt is, hogy hazug.*

Bizonyítás. Tegyük fel, hogy C azt állítja a D-ről, hogy igazmondó, és azt is állítja a D-ről, hogy hazug. Ekkor e két állításból pontosan az egyik igaz. C nem lehet igazmondó, hiszen akkor minden állításának igaznak kellene lennie, másrészt a C hazug sem lehet, mert akkor minden állításának hazugságnak kell lennie. Márpedig a fejtörőben csak igazmondók, illetve hazugok szerepelhetnek. Ez ellentmondás.

3.2. KÖVETKEZMÉNY. *Megoldható SS-típusú fejtörőben senki sem állíthatja olyanról, hogy igazmondó, aki róla azt állítja, hogy hazug.*

Bizonyítás. Tegyük fel, hogy C azt állítja a D-ről, hogy igazmondó, és D azt állítja C-ről, hogy hazug. Ekkor C nem lehet igazmondó, hiszen akkor D is az lenne, aki most éppen hazugságot állít. C hazug sem lehet, mert akkor a D is hazug lenne, miközben éppen igaza van. Ez ellentmondás.

3.3. LEMMA. *Egy megoldható SS-típusú fejtörő nem tartalmazhat szaggatott hurokélt, vagyis senki sem állíthatja önmagáról, hogy hazug.*

Bizonyítás. Tegyük fel, hogy C önmagáról azt állítja, hogy hazug. Világos, hogy C igazmondó nem lehet, mert ekkor hazug is lenne, ami ellentmondás. Ha viszont C hazug lenne, akkor ez az állítása igaz lenne, ami szintén ellentmondás (ez a jelenség egyébként Hazug-paradoxonként ismert).

A gráf nyelvén bizonyítva: ha van szaggatott hurokél a gráfban, akkor mivel ez az él is kiértékelő nyíl a végpontjainak különböző típusúnak kell lennie, ez ellentmond annak, hogy a hurokélnek csak egyetlen végpontja van, ami nem lehet egyszerre H és I típusú is. Tehát ennek a fejtörőnek nincs megoldása.

3.4. LEMMA. *Egy egyszerűmondatos igazmondó–hazug rejtvényben minden szereplő állíthatja magáról, hogy igazmondó.*

Bizonyítás. Bármely típusú csúcsot összeköthetünk saját magával folytonosan, hiszen a típusa megegyezik a sajátjával, ez nem befolyásolja a feladvány megoldása(i)t.

3.5. LEMMA. *Egy n szereplős erősen igazmondó-erősen hazug rejtvényben minden ember maximum $2n$ különböző állítást tehet, vagyis összesen maximum $2n^2$ állítás hangozhat el.*

Egy n szereplős, megoldható erősen igazmondó-erősen hazug rejtvényben minden ember maximum n különböző állítást tehet. Ez összesen maximum n^2 állítást jelent.

Bizonyítás. A lemma első része triviális, hiszen bármely szereplő legfeljebb két különböző atomi állítást mondhat ugyanarról a személyről. Most bizonyítsuk a második részt. A 3.2. Lemmával (illetve annak 1. Következményével) beláttuk, hogy bármely ember bármely másikról pontosan egy állítást tehet. A 3.4., illetve 3.5. Lemmából következik, hogy saját magáról is pontosan egy állítást tehet. Tehát bármely szereplő összesen annyi állítást tehet, ahány szereplős a rejtvény.

3.6. LEMMA. *A többszörös éleket elég egyszeresen figyelembe vennünk, az eredetivel megegyező lesz a megoldás.*

Bizonyítás. Az előző lemmák alapján bármely két csúcs között legfeljebb egyféle él vezethet, amelynek az irányítása nem lényeges. A kiértékelő nyilakat figyelembe véve bármely két csúcsot összekötő él ugyanannyi információt hordoz, mintha két vagy akár több ugyanilyen típusú él lenne köztük.

A továbbiakban bármely két csúcs között legfeljebb egy azonos típusú élt veszünk figyelembe.

4. A fejtörők gráfjának manipulálása

A gráf manipulálásán most elsősorban élek hozzáadását értjük. A fejezet végén azonban élek elhagyásával kapcsolatos eredményeket is közlünk.

Ahhoz, hogy többet tudjunk mondani e gráfokról ebben a részben olyan módszert mutatunk, amelynek segítségével úgy vehetünk éleket a gráfhoz, hogy a megoldás(ok) ne változzon(anak) meg. Ezeket a lépéseket mutatja a következő táblázat.

4.1. Definíció. A következő lépéseket élhozzáadó lépéseknek nevezzük, amennyiben a kiegészítő él még nem szerepelt a gráfban. Legyen A, B és C tetszőleges három csúcs, ekkor

ha az eredeti fejtörőben benne vannak a következő élek,	akkor a kiegészítő él
1. A —— B —— C	A —— C
2. A —— B - - - - - C	A - - - - - C
3. A - - - - - B - - - - - C	A —— C.

4.1. LEMMA. *Ha adott a P rejtvénynek egy megoldása, akkor ez megoldása annak az R rejtvénynek is, amit a P-ből valamely az – előbb felsorolt – élhozzáadó lépéssel kaptunk.*

Bizonyítás. Ha az első lépést alkalmaztuk, akkor az eredeti megoldásban a B és az A valamint a B és a C típusának meg kell egyeznie, tehát az A és C közé a folytonos él behúzható.

Ha a második lépést alkalmaztuk, akkor az A és a B azonos típusú, míg a B és a C különböző, így az A és a C is különböző, jogos köztük a szaggatott nyíl. A harmadik lépés alkalmazásakor a B az A-val és a C-vel is ellentétes típusú, így mivel csak kétféle típus van, az A és C típusa egyforma, köztük folytonos vonal lehet.

4.1. KÖVETKEZMÉNY. *Az élhozzáadó lépésekkel az eredetivel gyengén ekvivalens fejtörőket kapunk.*

4.2. LEMMA. *Ha valamelyik élhozzáadó lépéssel olyan rejtvényt kapunk, amelyben szaggatott hurokél jelenik meg, vagy két csúcs mindkétféle éllel össze lesz kötve, akkor az eredeti rejtvénynek nincs megoldása.*

Bizonyítás. A 3.2., illetve a 3.3. Lemma szerint ilyen fejtörő nem lehetséges megoldással, mivel a 4.1. Lemma alapján ami az eredetinek megoldása az ennek is, így az eredeti rejtvénynek sem lehet megoldása.

4.2. Definíció. Egy SS-típusú feladványt maximálisnak mondunk, ha élhozzáadó lépéssel nem adható gráfjához újabb él.

4.1. TÉTEL. *Minden megoldható feladványhoz pontosan egy olyan maximális fejtörő tartozik, amit az eredetiből élhozzáadó lépésekkel megkaphatunk.*

Bizonyítás. Tegyük fel, hogy a Q és az R is olyan maximális fejtörő gráfja, amit a P-ből nyertünk élhozzáadó lépések segítségével. Ekkor mivel Q és R maximálisak, egyikbe sem lehet új élt felvenni ily módon. Tegyük fel, hogy a Q-ban az A és a B csúcs össze van kötve. Ekkor belátjuk, hogy ezek a csúcsok az R-ben is össze vannak kötve ugyanilyen módon. Ha az A és B folytonos vonallal van összekötve, akkor típusuk megegyezik a megoldásban. De ugyanez a megoldása a P-nek is a 4.1. Lemma alapján, sőt ez a megoldás megoldása az R-nek is. Tehát mivel az R maximális, és benne az A és B csúcsok típusa megegyezik, e két csúcsnak az R-ben is össze kell kötve lennie folytonos éllel. Hasonló gondolatmenettel belátható az az eset is, amikor a két csúcs szaggatott vonallal van összekötve. Megfordítva a Q és az R szerepét láthatjuk, hogy ha egy él az R-ben benne van, akkor a Q-ban is benne kell, hogy legyen. Ezzel beláttuk, hogy a Q és az R gráf megegyezik.

4.2. TÉTEL. *Egy fejtörőből kapott maximális gráfban az eredetileg összefüggő részekben minden csúcs mindegyikkel össze lesz kötve, az eredetileg nem összefüggő részek között pedig továbbra sem lesz él.*

Bizonyítás. Először azt bizonyítjuk, hogy az összefüggő részben minden csúcs össze lesz kötve. Legyen az eredeti gráfunk P, a belőle kapott maximális gráf pedig R. Egy gráfban azt a részt nevezzük összefüggőnek, amelyen bármely csúcsból bármelyikbe eljuthatunk. Tegyük fel tehát, hogy az A és B csúcsok közt vezet út P-ben. Ha a csúcsok közvetlenül (is) össze voltak kötve, akkor készen vagyunk, ha nem, tekintsünk egy utat köztük: $A = B_0, B_1, \ldots, B_j = B$. Alkalmazva a megfelelő élhozzáadó lépést az A, B_1, B_2-re nyerünk egy olyan utat, amely a B_1-et kihagyva egy éllel rövidebb az eredetinél. Ezt folytatva végül elérjük, hogy közvetlen él legyen A és B közt, aminek a maximális gráf egyértelműsége miatt R-ben szerepelnie kell. Ha az A és a B közt nem vezetett út a P-ben, akkor R-ben sem vezethet, hiszen az élhozzáadó lépésekkel az összefüggőség nem változik.

Tekintsük most azt a gráf-reprezentációt, amelyben a szaggatott éleknek 1, a folytonosaknak 2 súlyú éleket feleltetünk meg.

4.3. TÉTEL. *Egy SS-típusú fejtörőnek pontosan akkor van megoldása, ha teljesül a következő: ha két csúcs közt vezet út, akkor köztük vagy minden út páros hosszúságú, vagy minden út páratlan hosszúságú.*

Bizonyítás. Tegyük fel indirekt, hogy az A és a B csúcsok közt vezet páros és páratlan hosszúságú út is. Ekkor először vegyük a páros hosszúságú $A = B_0, B_1, \ldots, B_j = B$ utat. Készítsük el az A és B közti közvetlen élt ezen út alapján a 4.2. Tétel bizonyításában szereplő módon. Ez egy 2 súlyú (azaz folytonos) élt fog eredményezni. (Az első és a második élhozzáadó lépés használatával 2-vel kevesebb lesz az új út költsége, mint az egy lépéssel hosszabb úté, míg a harmadik lépés használatakor az új, egy éllel rövidebb út költsége megegyezik az előzőével.) Hasonlóan a páratlan hosszú útból kiindulva azt kapjuk, hogy a maximális gráfban a két csúcs közt 1 súlyú (azaz szaggatott) él is vezet. Ez viszont a 3.2. Lemma alapján azt jelenti, hogy a maximális fejtörő nem oldható meg, ekkor viszont a 4.1. Lemma alapján az eredeti feladványnak sincs megoldása. A másik irány bizonyításához készítsük el a maximális fejtörő-gráfot. Ha két csúcs közt páros hosszú az út, legyenek összekötve 2 súlyú éllel, páratlan hosszúságú út esetén pedig 1-súlyú éllel. Most megkonstruálunk egy megoldást. Válasszunk ki egy-egy csúcsot minden komponensből, legyen ezek típusa I. Legyen továbbá minden ezekkel 2 súlyú éllel összekötött csúcs értéke I, valamint az 1 súlyú élekkel kapcsolódó csúcsok értéke 1. Könnyen belátható, hogy ez megoldása az eredeti fejtörőnek is.

4.2. KÖVETKEZMÉNY. *Ebben a reprezentációban két egymásból elérhető csúcsról azt mondhatjuk, hogy pontosan akkor azonos típusúak, ha az út költsége kettejük közt páros, illetve pontosan akkor ellenkező típusúak, ha köztük az út költsége páratlan.*

Bizonyítás. Az előző tételben beláttuk, hogy ha két csúcs közt vezet út, akkor a köztük vezető minden út költségének paritása ugyanaz. A fejtörőből készített maximális gráfban páros útköltség esetén 2 súlyú (folytonos) él, páratlan útköltség esetén 1 súlyú (szaggatott) él vezet a két csúcs közt. Ez a kiértékelő nyilakat használva éppen az állításunkat jelenti.

Térjünk most vissza az eredeti gráfjainkhoz, használjunk újra szaggatott, illetve folytonos nyilakat.

4.4. TÉTEL. *Adott egy fejtörő, tekintsük a gráfjának azt a részgráfját, amely csak a szaggatott nyilakat tartalmazza. Ha a fejtörőnek van megoldása, akkor ez a részgráf páros.*

Bizonyítás. Tegyük fel, hogy adott a feladvány egy megoldása, tekintsük ekkor az igazmondók és a hazugok halmazait. A kiértékelő nyilak tulajdonsága miatt szaggatott él csak a két halmaz között vezethet, halmazon belül nem, tehát a szaggatott élek páros gráfot alkotnak. (1.8.1. Definíció [3]-ban.)

4.3. KÖVETKEZMÉNY. *Egy feladványhoz rendelt maximális fejtörő gráfjában a szaggatott élek bizonyos értelemben maximális páros gráfokat jelentenek. Ha a fejtörő gráfja összefüggő, akkor teljes páros gráfot kapunk. Ha nem összefüggő az eredeti gráf, akkor összefüggő részenként kapunk egy-egy teljes páros gráfot a szaggatott élekkel.*

4.5. TÉTEL. *Ha egy rejtvénynek van megoldása, akkor a folytonos élek részgráfjai teljes diszjunkt gráfok a fejtörőhöz rendelt maximális gráfban.*

Bizonyítás. Vegyük az összes lehetséges megoldást, nyilvánvaló (a 4.2. és a 4.3. Tételek alapján), hogy két csúcs közt pontosan akkor vezet folytonos él, ha típusuk minden lehetséges megoldásban megegyezik. Ekkor viszont a 4.2. Tétel és bizonyítása alapján ezen részek teljes gráfok.

4.3. Definíció. Egy megoldás inverzén azt a függvényt értjük, amely minden emberhez pontosan az ellenkező típust rendeli, mint az adott megoldás.

4.3. LEMMA. *Minden megoldásfüggvény különbözik az inverzétől.*

Bizonyítás. Triviális, minden szereplőben eltérőek.

4.6. TÉTEL. *Legyen egy feladvány gráfja P. Egy f függvény pontosan akkor a P megoldása, ha azonos típusú csúcsok közt csak folytonos, ellentétes típusúak között csak szaggatott él van a hozzárendelés után.*

Bizonyítás. Tegyük fel, hogy f megoldás. Ha két azonos típusú csúcs közt szaggatott él vezet, az ellentmondásra vezet, csakúgy, mint két eltérő típusú közt levő folytonos él (kiértékelő nyilak). Tehát ha f megoldás, akkor az állítás igaz.

Most tegyük fel, hogy azonos típusú csúcsok közt csak folytonos, ellentétes típusúak között csak szaggatott él van. Ekkor minden igazmondó által mondott mondat igaz, hiszen belőle folytonos él csak igazmondóhoz (azonos típusú), szaggatott pedig csak hazughoz (ellenkező típusú) vezet. Hasonlóan minden hazug minden állítása hamis, ugyanis igazmondóságot csak hazugról (folytonos él azonos típusúhoz); azt hogy hazug, pedig csak igazmondóról (szaggatott él a másik típusúhoz) állít. Tehát az f megoldása a fejtörőnek.

4.7. TÉTEL. *Tiszta erősen igazmondó–erősen hazug jó fejtörő nincs.*

Bizonyítás. Tegyük fel, hogy van ilyen fejtörő, legyen a gráfja P. Megmutatjuk, hogy a megoldás mellett annak inverze is megoldás. Mivel a feladvány tiszta, a gráf tartalmaz minden információt. Ha nincs megoldás, akkor a fejtörő nem lehet jó. Tehát van megoldása, legyen f egy megoldás. Ekkor a 4.6. Tétel alapján azonos típusú csúcsok közt csak folytonos, ellentétes típusúak között csak szaggatott él van. Vizsgáljuk meg most a g függvényt, ami az f inverze. A P gráf csúcsaiba írt típusokat megváltoztatva, a különbség annyi, hogy amely él eddig két I csúcsot kötött össze, az most két H csúcsot köt össze, és fordítva. A 4.6. Tétel feltételei most is teljesülnek. Tehát g is megoldás. Mivel van két különböző megoldása, a P nem lehet jó feladvány.

Az előzőekben a feladvány gráfját újabb élekkel egészítettük ki. Vajon mit tapasztalunk, ha elhagyunk egyes éleket a gráfból? Mennyire függetlenek egymástól a fejtörő élei?

4.4. Definíció. Egy fejtörő minimális fejtörőjének nevezzük azt a fejtörőt, ami az eredetivel gyengén ekvivalens (pontosan ugyanazok a lehetséges megoldásai), és a legkevesebb elhangzó mondatot tartalmazza. Hasonlóan értelmezzük a minimális gráf fogalmát is.

Az előző definíció, valamint a 3.3. Lemma alapján azt mondhatjuk, hogy az ellentmondásos fejtörők minimális gráfja egyetlen – szaggatott hurok- – élt tartalmaz.

4.4. LEMMA. *Egy megoldható feladvány minimális fejtörőjének gráfjában minden komponensben eggyel kevesebb él van, mint csúcs.*

Bizonyítás. Könnyen belátható, hogy komponensenként egy feszítőfa, bármely élek elhagyásával pontosan egy minimális fejtörőt fog reprezentálni.

Összefüggő gráf esetén egy minimális fejtörő éppen egy feszítőfát jelent, nem összefüggő esetben pedig egy feszítőerdőt.

4.8. TÉTEL. *Bármely SS-típusú rejtvény esetén a feszítőerdőhöz tartozó rejtvény mindig megoldható.*

Bizonyítás. A feszítőerdőben bármely két csúcs között legfeljebb egy út vezethet. Ez a 4.3. Tétel értelmében elégséges feltétele a megoldhatóságnak.

Az előző tétel miatt vigyáznunk kell az élek elhagyásaival, mert lehet hogy egy rejtvénynek nem volt megoldása, ekkor a gráfjának feszítőerdeje nem ekvivalens az eredeti rejtvénnyel.

4.9. TÉTEL. *Ha egy hurokélmentes gráffal rendelkező SS-típusú rejtvénynek nincs megoldása, akkor van legalább két olyan, a gráfjához tartozó feszítőerdő, amelyekhez tartozó rejtvényeknek nincs azonos megoldása.*

Bizonyítás. Tegyük fel, hogy a P rejtvény nem megoldható és gráfja hurokélmentes. Ekkor a 4.3. Tétel alapján lennie kell benne 2 csúcsnak melyek közt vezet páros, illetve páratlan hosszú út is. Készítsük el a P gráf R feszítőerdejét úgy, hogy benne a fenti 2 csúcs között páros hosszú út vezessen, míg a Q feszítőerdőben legyen a 2 csúcs közti út páratlan hosszú. Nyilvánvalóan az R gráfot tekintve minden megoldásban azonos értékű lesz a fenti két csúcs. Ezzel szemben a Q megoldásaiban mindig különböző értékű lesz ez a két csúcs.

5. Erősen igazmondó – erősen hazug fejtörők megoldása

Most módszert adunk az ilyen típusú fejtörők megoldására.

Itt jegyezzük meg, hogy a vizsgált fejtörők egyszerűen megoldhatóak az összes lehetséges I–H sorozat ellenőrzésével, azonban n szereplő esetén ez 2^n darab „próbálkozást" jelent. Itt egy próbálkozás ellenőrzéséhez a fejtörő minden állítását meg

kell vizsgálnunk, ami egyáltalán nem hatékony. Ezért most egy jóval hatékonyabb algoritmust adunk, felhasználva az előző eredményeinket.

A következő algoritmus segítségével egy tiszta fejtörő lehetséges megoldásait állíthatjuk elő.

5.1. Algoritmus.

Adott a fejtörő gráfja, P.

1. Nézzük meg hány komponensből áll a gráf (2.1.7. Definíció [3]-ban).
2. Komponensenként jelöljünk ki egy-egy csúcsot.
3. A kiválasztott csúcsokhoz külön-külön rendeljük hozzá az $\{I, H\}$ valamelyik elemét.
4. Ciklus komponensenként: amíg az adott komponens minden csúcsához nincs érték rendelve az $\{I, H\}$ halmazból, az élek, mint kiértékelő nyilak segítségével az összefüggő részekben határozzuk meg a csúcsok típusait a már ismertekből kiindulva.
5. Ellenőrzési fázis, csak az első megoldás előállításakor: ciklus komponensenként: minden élre vizsgáljuk meg, hogy a végpontjaiban szereplő értékek azonosak (folytonos él esetén), illetve különbözőek (szaggatott él esetén).
   Ha valamelyik élre nem stimmel, akkor NINCS MEGOLDÁS és STOP.
6. Amennyiben nincs szükségünk minden megoldás előállítására, akkor már van(nak) megoldás(ok), STOP.
7. Vissza a 3. lépésre és ott rendeljük az I, H értékeket az eddigiektől eltérően a kiválasztott csúcsokhoz, ha ez lehetséges. Ha ez már nem lehetséges, akkor minden megoldást előállítottunk. STOP.

5.1. TÉTEL. *Ha a P nem megoldható, akkor az 5.1. Algoritmus ellentmondásra vezet. Ha a P megoldható, akkor pedig megoldás(oka)t állít elő.*

Bizonyítás. Tegyük fel, hogy a P nem megoldható. Ez azt jelenti a 4.6. Tétel alapján, hogy nincs olyan függvény, hogy azonos típusú csúcsok közt csak folytonos, ellentétes típusúak között csak szaggatott él van a hozzárendelés után. Ez azt jelenti, hogy lennie kell olyan csúcsnak, hogy a kiértékelő nyilak segítségével ez a csúcs I és H típusú is lesz, ami ellentmond annak, hogy mindenki csak egyféle típusú lehet. Ilyen esetben az algoritmus nem állít elő függvényt. Tegyük fel, hogy a P megoldható. Ekkor azt állítjuk, hogy ha az algoritmus előállítja az f függvényt, akkor az f megoldás. Ekkor a kiértékelő nyilakkal nem kapunk ellentmondást egyik komponensben sem. Tehát azonos típusú csúcsok közt csak folytonos, ellentétes típusúak között csak szaggatott él van a hozzárendelés után, így f megoldás.

5.2. TÉTEL. *Legyen k a komponensek száma a fejtörő gráfjában. Ekkor lehetséges megoldások száma 2^k. Az algoritmus pedig mindet elő tudja állítani.*

Bizonyítás. Először azt bizonyítjuk, hogy az algoritmus ennyi különböző megoldást állít elő. Az algoritmus 3. lépésében kiválasztunk minden komponensből egy csúcsot, és mindhez hozzárendelhetjük az $\{I, H\}$ egyik elemét, ezt pontosan

2^k-féleképpen tehetjük meg, és ez mind más megoldást fog eredményezni. Most bebizonyítjuk, hogy ezeken kívül nincs más megoldás. Tegyük fel, hogy f megoldás. Ekkor tekintsünk a P gráfban minden komponensből egy-egy csúcsot, és rendeljük hozzá az f által adott értéket. Ekkor a kiértékelő nyilak segítségével éppen az f megoldást kapjuk meg.

Az előző két tétellel megmutattuk, hogy algoritmusunk helyes és teljes.

5.3. TÉTEL. *A nem tiszta fejtörő megoldása mindig része az ugyanazon P gráfhoz tartozó tiszta rejtvény lehetséges megoldásainak.*

Bizonyítás. Tegyük fel, hogy f nem lehetséges megoldása P-nek. Ekkor a gráfban a hozzárendelés után ellentmondást kapunk, vagyis lesz két különböző típusú csúcs folytonos éllel, vagy két azonos típusú csúcs szaggatott éllel összekötve. Tehát valamely a P gráf által hordozott információ nem teljesül. Ez tehát nem lehet megoldás, akkor sem, ha a gráfon kívül még más információnk is van.

Most vizsgáljuk meg az algoritmus hatékonyságát. Az első megoldás előállításához a gráf egy feszítőerdejét kell „végigjárni". A megoldás ellenőrzéséhez a gráf összes élét meg kell vizsgálnunk. A többi megoldás előállításához már csak a feszítőerdők bejárására van szükség az algoritmus szerint.

Megjegyezzük, hogy a megoldhatóság vizsgálata történhet a 4.3. Tétel alapján, de ez nem feltétlenül gyorsabb, mint ahogy az algoritmus egy lehetségesnek tűnő megoldást leellenőriz.

Az algoritmus a további megoldások előállításában gyorsítható a következőképpen:

Vegyük a gráf k komponensét, ezeken belül (mint kisebb, tiszta SS-típusú rejtvények) a megoldások inverze is megoldás (lásd 4.3. Lemma és 4.7. Tétel). Tehát minden komponensre adott az algoritmus által adott első megoldás és annak az inverze. Az összes megoldás pedig ezek tetszés szerinti kombinációja, vagyis az eredeti rejtvény bármely megoldása előáll a komponensek első, vagy azok inverz megoldásaiból.

6. Összefoglalás

Az egyszerűmondatos fejtörők definíciója után azok gráfjait vizsgáltuk. Több észrevételt tettünk arra, hogy milyen lehet egy megoldható feladat gráfja. Ezen kívül bevezettük a kiértékelő nyilakat, amik segítségével a már ismert típusú csúcsokból indulva a velük összefüggő gráfrészben szereplő csúcsok típusát meghatározhatjuk. Az élhozzáadó lépésekkel pedig olyan gráfokat állíthatunk elő, melynek megoldásai megegyeznek az eredetiével. Ezek a lépések a lokális információkat kihasználva segítenek jellemezni a gráfot és megtalálni a megoldást. Beláttuk, hogy ahhoz, hogy egyértelmű megoldásunk legyen még plusz információra lenne szükségünk. A módszer akkor is kiválóan működik, ha nem szólal meg minden szereplő.

Egy meg nem szólaló szereplőt helyettesíthetünk például más atomi kijelentéssel. Ekkor, ha a fejtörő jó, akkor egyértelműen kiderül e kijelentés típusa, vagyis igazságértéke a megoldás során. Tehát a módszert szinte változatlan formában használhatjuk akkor is, ha a szereplők nem csak egymás típusára, hanem például az „ez az út Budára vezet", illetve (és/vagy) ennek ellenkezőjét (is) állítják. Ekkor ennek a mondatnak is megfeleltetünk egy csúcsot a gráfban.

A gráf tulajdonságait használó módszerhez hasonlót más típusú fejtörőkre is alkalmazhatunk.

Köszönetnyilvánítás

A szerző ezúton is szeretné megköszönni a lektor és az érdeklődő kollégák tanácsait, észrevételeit.

Irodalom

[1] Aszalós, L., *Smullyan logikai rejtvényei és automatikus megoldásuk*, Technical Report No. 2000/14 (University of Debrecen, Institute of Mathematics and Informatics, 2000).

[2] Shasha, D., *Dr. Ecco talányos kalandjai* (Typotex, Budapest, 1999).

[3] Hajnal, P., *Gráfelmélet* (Polygon, Szeged, 1997).

[4] Smullyan, R., *Forever Undecided (A Puzzle Guide to Gödel)* (Alfred A. Knopf, New York, 1987).

[5] Smullyan, R., *A hölgy vagy a tigris?* (Typotex, Budapest, 1997).

[6] Smullyan, R., *Mi a címe ennek a könyvnek?* (Typotex, Budapest, 1996).

[7] Smullyan, R., *Seherezádé rejtélye* (Typotex, Budapest, 1999).

[8] Kósa, M. and Nagy, B., Logical puzzles (Truth-tellers and liars), in: *5th International Conference on Applied Informatics (ICAI'01)*, 105–112 (Eger, 2001).

(Beérkezett: 2001. december 3.)

NAGY BENEDEK
DEBRECENI EGYETEM
ÉS
ROVIRA I VIRGILI EGYETEM
TARRAGONA
SPANYOLORSZÁG
nbenedek@inf.unideb.hu

SS-TYPE TRUTHTELLER–LIAR PUZZLES AND THEIR GRAPHS

BENEDEK NAGY

In this paper we define the SS-type puzzles. In these puzzles each person can say only simple statements about a person type. A truth-teller can have only true assertions, and a liar can make only false statements. We present the graph-representations of these puzzles. The graph of a puzzle has all information about the puzzle in case of clear puzzles. We present some interesting properties of the possible graphs of puzzles. We prove that there is no clear SS-type puzzle with unique solution. We present an algorithm which can provide all possible solutions of a clear puzzle. In case of non-clear puzzles, we must choose among the possible solution of the clear puzzle with the same graph.

Alkalmazott Matematikai Lapok **23** (2006), 73–83.

MŰSZAKI BERENDEZÉSEK VIZSGÁLATA FAKTORANALÍZIS SEGÍTSÉGÉVEL

GYARMATI JÓZSEF

Budapest

Cikkemben műszaki berendezések paramétereit elemzem a többváltozós matematikai statisztika eszközeivel. Az elemzés során a korrelációs viszonyokat vizsgálva csoportosítom a paramétereket, valamint a korrelációt kiváltó okokat azonosítva, a mintát alkotó berendezések meghatározó vetületeit megállapítva inputadatokat szolgáltatok ugyanazon eszközhöz tartozó többszempontú döntési problémához. Vizsgálatom célja a többváltozós analízis egy újszerű alkalmazásának a bemutatása. A számításokat SPSS 11. statisztikai programcsomag segítségével végeztem el.

1. Bevezetés

A katonai beszerzések gyakorlatában gyakran előforduló probléma két vagy több haditechnikai eszköz közül a legmegfelelőbb kiválasztása. A kérdéses eszközök több vetületük alapján jellemezhetők, a közülük történő választás tehát egy többszempontú döntési probléma, melynek általános modelljét az (1) mutatja:

$$
\begin{array}{cc} & \begin{array}{cccc} A_1 & A_2 & \cdots & A_n \end{array} \\ \begin{array}{cc} C_1, & w_1 \\ C_2, & w_2 \\ \vdots & \vdots \\ C_m, & w_m \end{array} & \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array}\right] \\ & \begin{array}{cccc} x_1 & x_2 & \cdots & x_n \end{array} \end{array}, \tag{1}
$$

ahol: A_j a j-edik haditechnikai eszköz (alternatíva), $j = 1, 2, \ldots, n$;
C_i az i-edik vizsgálati szempont, $i = 1, 2; \ldots, m$;
w_i az i-edik szempont fontosságát jelző súlyszám, $i = 1, 2, \ldots, m$;

a_{ij} az j-edik haditechnikai eszköz i-edik szempont szerinti értéke (utilitása); $i = 1, 2, \ldots, m$ és $j = 1, 2, \ldots, n$;

x_j a j-edik haditechnikai eszköz pontszáma, $j = 1, 2, \ldots, n$.

Az alternatívák pontszámai a vizsgálati szempontok és ezek súlyszámainak, valamint az alternatívák szempontonkénti értékeinek a függvénye. Ismertnek tekinthetők az alternatívák szempontonkénti értékei, a súlyszámok számítására több módszer áll rendelkezésre, a szempontrendszer kialakítására viszont csak általános jellegű ajánlások vannak, melyek teljesítési szintje szubjektív. Mivel a pontszám és ebből következően az alternatívák sorrendje függ a szempontoktól, ezért megállapítható, hogy a sorrendet a szempontrendszert felépítő szakértői csoport szubjektivitása jelentős mértékben befolyásolja. Dolgozatomban olyan módszert kínálok, amely ezen szubjektivitást csökkentve segítséget nyújt egy többszempontú döntési probléma esetén a szempontrendszer kialakításában.

Vizsgálatom során abból a megfigyelésből indultam ki, hogy műszaki berendezések egyes paraméterei között korrelációs viszony tapasztalható. Feltevésem szerint a korrelációt kiváltó ok az eszközre jellemző és az eszköz valamilyen funkcionális képességével van összefüggésben. A többszempontú döntési eljárás során a vizsgálati vetületeknek vagyis a szempontoknak pontosan ezeket a funkcionális képességeket kell takarniuk. Tehát, ha megfelelő nagyságú minta alapján korrelációs viszony szerint csoportosítjuk a vizsgált eszközök paramétereit, és az egyes csoportokon belül meghatározzuk a korrelációt kiváltó okokat, akkor egy olyan tulajdonságlistát kaphatunk, amely alapjául szolgálhat a szempontrendszer kialakításában.

A 2. pontban bemutatom a probléma megoldásában alkalmazott matematikai módszereket, a 3. pontban pedig az ismertetett módszerek segítségével gépjárművek paramétereit elemzem. Vizsgálatom során nem volt célom új matematikai eredmények bemutatása, csak a 3. pontban bemutatott példán keresztül az előzőekben felvetett probléma megoldására kínálok megoldást a többváltozós analízis egy újszerű alkalmazásával.

2. A matematikai modell

A problémát két módszer segítségével oldottam meg: a faktoranalízis és a főkomponens elemzés. Jelen fejezetben ezen két eljárás lényegét közlöm.

A *faktoranalízist* a [3] alapján ismertetem, ezen kívül a módszertannal és az alkalmazással foglalkoznak az [1, 2] irodalmak.

Rendelkezzenek egy N számú sokaság egyedei n ismérvvel és reprezentálja a j-edik ismérv változatainak az eloszlását $X_j \mathrm{N}(m_j, \sigma_j)$ valószínűségi változó. Legyen x_{ij} az i-edik egyed j-edik ismérv szerinti értéke, ahol $i = 1, \ldots, N$ és $j = 1, \ldots, n$. Standardizáljuk az X_j valószínűségi változó reprezentációit a

$$z_{ij} = \frac{x_{ij} - \overline{x}_j}{s_j}$$

egyenlet szerint, ahol $\overline{x}_j$ az X_j valószínűségi változó realizációinak a számtani közepe és s_j a korrigált tapasztalati szórása. Hasonlóan legyen a Z_j valószínűségi változó az X_j-ből képzett N(0, 1) változó. A standardizált értékek mátrixa:

$$\mathbf{Z} = \begin{bmatrix} z_{11} & \dots & z_{1n} \\ \vdots & & \vdots \\ z_{N1} & \dots & z_{Nn} \end{bmatrix}.$$

A standardizált értékek segítségével határozzuk meg a korrelációs együtthatókat:

$$r_{jk} = \frac{\sum_{i=1}^{N} z_{ij} z_{ik}}{N}.$$

A korrelációs együtthatók mátrixa:

$$\mathbf{R} = \begin{bmatrix} r_{11} & \dots & r_{1n} \\ \vdots & & \vdots \\ r_{n1} & \dots & r_{nn} \end{bmatrix}.$$

Az **R** mátrix a faktoranalízis kiinduló pontja. A matematikai modell feltételei szerint n számú X_j valószínűségi változóval rendelkezünk, melyek egymással korreláltak. A korrelációt független virtuális változók, un. faktorok hatása okozza. A faktorok következő típusait különböztethetjük meg:

1. *Közös faktorok,* amelyekben egyszerre több ismérv (valószínűségi változó) jelentkezik és ezek egymással korreláltak. Jelölésük: $F_1, \dots, F_m$.
2. *Specifikus faktorok,* amelyekben csak egy változó hatása figyelhető meg. Jelölésük: S_j és $j = 1, \dots, n$.
3. *Hibafaktorok,* amelyekben nem figyelhető meg egyetlen egy ismérv hatása sem. Jelölésük: E_j és $j = 1, \dots, n$.

A Z_j változót az eljárás segítségével a faktorok lineáris kombinációiként a (2) egyenletek szerint írhatjuk fel:

$$\begin{aligned} Z_1 &= a_{11}F_1 + a_{12}F_2 + \dots + a_{1m}F_m + b_1S_1 + c_1E_1, \\ Z_2 &= a_{21}F_1 + a_{22}F_2 + \dots + a_{2m}F_m + b_2S_2 + c_2E_2, \\ &\vdots \\ Z_n &= a_{n1}F_1 + a_{n2}F_2 + \dots + a_{nm}F_m + b_nS_n + c_nE_n, \end{aligned} \tag{2}$$

ahol: a_{jk} a Z_j k-adik faktorához tartozó faktorsúlya;
b_j a Z_j specifikus faktorához tartózó faktorsúlya;
c_j a Z_j hibafaktorához tartozó faktorsúlya.

A modell szerint a közös, a specifikus és a hibafaktorok várható értékei nullák, valamint páronként függetlenek, amennyiben X_j normál eloszlású, ha ez nem teljesül, akkor csak korrelálatlanságról beszélhetünk.

Az a_{jk} faktorsúly a k-adik faktor hozzájárulását fejezi ki a Z_j változó s_j^2 szórásnégyzetében, vagyis

$$s_j^2 = a_{j1}^2 + a_{j2}^2 + \ldots + a_{jm}^2 + b_j^2 + c_j^2, \qquad (j = 1, 2, \ldots, n).$$

Az eljárás eredményeként a közös faktorok együtthatómátrixát, vagyis a faktorsúlyok mátrixát kapjuk:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nm} \end{bmatrix}.$$

A közös faktorokhoz tartozó faktorsúlyok négyzetösszegét kommunalitásnak nevezzük és h_j^2-*tel* jelöljük:

$$h_j^2 = a_{j1}^2 + a_{j2}^2 + \ldots + a_{jm}^2.$$

Mivel Z_j standardizált, ezért szórásnégyzete 1, ezért:

$$h_j^2 + b_j^2 + c_j^2 = 1,$$

vagyis a kommunalitások segítésével kifejezhető, hogy a közös faktorok milyen mértékben magyarázzák az eredeti változók szórásnégyzeteit, vagyis információt ad az eljárás eredményének az információtartalmáról. A faktorsúlyok meghatározását az [1, 2, 3] irodalmak tartalmazzák.

Az eljárás eredményeként egymással korrelált valószínűségi változókat az eredeti változószámnál kevesebb, egymással páronként korrelálatlan, virtuális változó, vagyis faktor segítségével írhatunk le. Az eredmények segítségével a rendszert leíró változók számát lehet csökkenteni, valamint lehetőség nyílik a rendszer belső struktúrájának analitikus vizsgálatára.

A *főkomponensanalízis* lényegét a [2] szerint mutatom be. Az eljárás ismertetése során csak a faktoranalízistől való eltérés bemutatására törekszem. Az $X_i\mathrm{N}(m_i, \sigma_i)$ valószínűségi változóból, a faktoranalízishez hasonlóan képezzük $Z_i\mathrm{N}(0, 1)$ valószínűségi változót. A főkomponensanalízis során Z_i olyan lineáris kombinációit keressük, ahol teljesül, hogy maximális szórásúak, korrelálatlanok és lineáris kombinációik együtthatóinak a négyzetösszege egységnyi:

$$Z_1 = a_{11}C_1 + a_{12}C_2 + \ldots + a_{1n}C_n,$$

$$Z_2 = a_{21}C_1 + a_{22}C_2 + \ldots + a_{2n}C_n,$$

$$\vdots$$

$$Z_n = a_{n1}C_1 + a_{n2}C_2 + \ldots + a_{nn}C_n,$$

ahol: a_{ij} a Z_i j-edik főkomponenséhez tartozó együtthatója, $i, j = 1, \ldots, n$;
C_j a j-edik főkomponens.

A felbontásból látható, hogy a valószínűségi változók és a főkomponensek (virtuális változók) száma megegyezik, de mivel a modell feltétele szerint:

$$D^2C_j > D^2C_{j+1} \qquad (j = 1, \ldots, n-1), \tag{7}$$

az utolsó néhány komponens elhagyásával Z_j szórásnégyzetei magas százalékban magyarázhatók a megmaradt főkomponensek segítségével. Az elhagyott főkomponensek számának a meghatározásához a [2] Bartlett tesztet javasol.

A kommunalitások jelen esetben az el nem hagyott főkomponensekhez tartozó együtthatók négyzetösszegét jelentik.

3. Gépjárművek vizsgálata főkomponens- és faktoranalízis segítségével

A vizsgálatot 49 darab közúti és terepjáró gépjármű 17 paraméterére végeztem el SPSS 11 statisztikai programcsomag segítségével. A járművek között – személygépkocsitól a nyerges vontatóig – minden teherbírási kategória megtalálható. Az egyes típusok gyártásának kezdeti időpontjai között nincs jelentős eltérés. Amennyiben ez nem állna fenn, az analízis eredménye nem lenne feltétlenül megbízható, mivel a műszaki-tudományos fejlődés a vizsgált berendezések paramétereit folyamatosan módosítja, ami befolyásolhatja a korrelációk nagyságát. Az adathalmaz elemzését főkomponens- és faktoranalízis segítségével is elvégeztem.

A főkomponensanalízis elvégzésekor az egyes főkomponensek szórásnégyzetei alapján meg kell határozni, hogy az elemzés során hány főkomponest válasszunk ki. Ezt a program számára meg lehet adni a főkomponesek szükséges számával és a korrelációs mátrix legnagyobb még figyelembe vett sajátértékének a segítségével. A sajátérték rendre megegyezik a hozzá tartozó főkomponens szórásával. Mivel a sajátértékek és ezzel a szórásnégyzetek a (7) egyenlőtlenség szerint csökkennek, ezért az első néhány főkomponens az eredeti változók teljes szórásnégyzetét általában nagy százalékban magyarázza. A határt az elhagyott és a bennmaradt főkomponensek között ott célszerű meghúzni, ahol a szórásnégyzetek között nagy lesz a különbség. Az eljárás eredményeként kapott együtthatómátrix információtartalmát jól mutatja a bennmaradt főkomponensek szórásnégyzet összegének és a főkomponensek teljes szórásnégyzetének – ami megegyezik az eredeti változók teljes szórásnégyzetével – a hányadosa százalékos formában kifejezve (1. táblázat).

Esetemben a korrelációs mátrix legkisebb még figyelembe vett sajátértékének 0,8-et adtam meg, így az első öt főkompones került kiválasztásra, melyek a teljes szórásnégyzetet 81,609%-ban magyarázzák. Az eredményeket a 2. táblázat mutatja.

A 2. táblázat szerint a *első főkomponenssel* a legnagyobb korrelációs viszonyban a fogyasztás van, mellette korrelációt tapasztalhatunk az öntömeg, a teherbírás, a vontatmány tömeg, a motor teljesítmény, a vonóerő és az összgördülőtömeg esetében. Ezek a jellemzők beláthatóan műszaki megfontolások szerint is szoros

	Teljes szórásnégyzet			Kiválasztott főkomponensek szórásnégyzetei			Rotált főkomponensek szórásnégyzetei		
Főkomponens	Saját-érték	A teljes szórás-négyzet %-ában	Kumulált %	Saját-érték	A teljes szórás-négyzet %-ában	Kumulált %	Saját-érték	A teljes szórás-négyzet %-ában	Kumulált %
1	8,434	49,610	49,610	8,434	49,610	49,610	5,682	33,426	33,426
2	1,940	11,411	61,022	1,940	11,411	61,022	2,869	16,878	50,304
3	1,498	8,811	69,833	1,498	8,811	69,833	2,475	14,561	64,865
4	1,127	6,631	76,464	1,127	6,631	76,464	1,491	8,769	73,635
5	,875	5,145	81,609	,875	5,145	81,609	1,356	7,974	81,609
6	,655	3,851	85,460						
7	,546	3,210	88,670						
8	,446	2,625	91,295						
9	,414	2,436	93,731						
10	,323	1,901	95,632						
11	,233	1,368	96,999						
12	,201	1,182	98,182						
13	,130	,765	98,946						
14	8,570E−02	,504	99,451						
15	4,245E−02	,250	99,700						
16	2,871E−02	,169	99,869						
17	2,224E−02	,131	100,000						

1. táblázat. A teljes szórásnégyzet magyarázata

összefüggésben állnak egymással. A korrelációt kiváltó ok a *teherbírási kategória*, ezért ez a főkomponens ezt a szempontot reprezentálja.

A *második főkomponens* a max. sebességgel korrelál, emellett gyenge korrelációt tapasztalhatunk a jármű befoglaló méreteivel (hosszúság, magasság) és a teljesítmény dotációval (a gördülőtömeg és a motorteljesítmény hányadosa). Az összefüggés járműtechnikai okai beláthatók, de az összefüggés nagysága miatt szorosabb korrelációt lehetne várni főképpen a sebesség és a teljesítmény dotáció között. A jelenség magyarázható azzal, hogy a járművek legnagyobb sebességét a méreten és a tonnára vetített teljesítményen kívül forgalombiztonsági és jogszabályi előírások is meghatározzák. Haszongépjárművek esetén a motor egy meghatározott sebesség elérésekor automatikusan leszabályoz, így műszakilag akadályozza meg a biztonságosnak ítélt sebesség, általában 100 km/h túllépését. A főkomponens a *közúti mozgékonyság* szempontját reprezentálja.

A *harmadik főkomponens* legnagyobb korrelációban a terepjárással van, melletted korrelációt tapasztalhatunk az oldaldőlés és a mászóképesség esetében, melynek oka egyértelmű, kérdéses viszont az ellenkező előjel. A vizsgált járművek között közútiak és terepjárók egyaránt találhatóak. A kialakítást egy alternatív ismérv segítségével vettem figyelembe, melynek értéke 1, ha a jármű közúti és 0, ha te-

	Főkomponens				
	1	2	3	4	5
fogyasztás	**,909**	,161	−7,802E−02	,161	1,338E−02
motor teljesítmény	**,885**	,265	,153	,258	−3,659E−02
vontatmány tömeg	**,871**	7,775E−02	5,517E−02	−,140	−,139
összgördülőtömeg	**,790**	,370	,279	,296	9,541E−02
max. vonóerő	**,781**	,180	,291	,199	5,001E−02
öntömeg	**,771**	,490	,151	,236	7,637E−02
teherbírás	**,713**	,361	,235	,359	,127
max. sebesség	−,117	**−,810**	5,093E−02	,101	,338
magasság	,316	**,736**	,296	−,138	−5,701E−02
teljesítmény dotáció	−,403	**−,642**	−,368	−,254	−,177
hosszúság	,281	**,635**	,164	,373	,188
terepjárás	−,157	8,221E−02	**,821**	,104	−,141
oldaldőlés	−,305	−,207	**−,816**	,139	−7,722E−02
mászóképesség	−,458	−,120	**−,734**	4,032E−02	−,124
fajlagos fogyasztás	−,307	1,696E−02	8,954E−02	**−,864**	2,966E−02
nyomatéki rugalmasság	−,145	−,198	−9,210E−02	−7,568E−02	**,868**
fordulókör átmérő	,426	,331	,232	,223	,567

2. táblázat. Rotált együtthatómátrix

repjáró. A harmadik főkomponens által reprezentált szempont a *terepjáró képesség* szempontja.

A *negyedik főkomponens* egyedül a fajlagos fogyasztással korrelál. Ez a paraméter csak a motor korszerűségére jellemző, ezért nem lehet más itt is megjelenő paraméterrel való összefüggést tapasztalni. Vizsgálható lenne a fogyasztás, de ennek az értékét a fajlagos fogyasztáson kívül még számos tulajdonság alakítja ki, például a teljesítménydotáció, az erőátvitel áttételi viszonyai, a gumiabroncsok kialakítása és a hajtott tengelyek száma, stb. A negyedik főkomponens ennek megfelelően csak a *fajlagos fogyasztást* reprezentálja.

Az *ötödik főkomponens* csak a nyomatéki rugalmassággal van korrelációs viszonyban. Az előző főkomponens tapasztaltakhoz hasonlóan ez is csak a motor jellemzője.

Az első két főkomponenshez tartozó paraméterek közül néhány (öntömeg, teljesítmény dotáció) mindkét komponenssel mutat némi korrelációs viszonyt, amiből nagyrészt a második főkomponenst és az ide megállapított szempontot teszi bizonytalanná. A fordulókör átmérő nem korrelál egyértelműen egyik főkomponenssel sem, valamint nem egyértelmű a korreláció a jármű hosszúsága esetében. A vizsgált gépjárművek között van terepjáró személygépkocsi, pick up, nyergesvontató, valamint kettő-, három-, és négytengelyes gépjármű. Az eltérő funkciókból adódóan jelentősek az eltérések a gépjárműveket jellemző fő méretek arányaiban, amivel

magyarázható a jármű befoglaló méreteinek a korrelálatlansága. Nincs figyelembe véve a tengelyek száma, a nyomtávolság és a tengelytávolság aránya, valamint a szerkezeti kialakítás, amivel a fordulókör korrelálatlansága magyarázható.

A minta nagysága lehetővé tette a faktoranalízis elvégzését is. Az együtthatómátrixot főfaktor módszerrel állítottam elő. A létrehozandó közös faktorok számát, hasonlóan, mint azt a főkomponensanalízisnél láttuk, a program számára előre meg kell adni. Célszerű több faktormegoldást megvizsgálni, megkeresve azt, ahol a kumulált szórásnégyzetek maximálisak lesznek, ugyanis ez a faktormegoldás magyarázza legnagyobb mértékben az eredeti változók teljes szórásnégyzetét.

Esetemben az optimális faktormegoldás a főkomponensanalízis eredményéhez hasonlóan öt közös faktort tartalmaz, amelyek 71,52%-ban magyarázzák az eredeti változók teljes szórásnégyzetét. Faktoranalízis esetében a minta megfelelőségét a program Kaiser–Meyer–Oldkin teszt segítségével ellenőrzi. Ennek értéke jelen esetben 0,794 volt, ami jó faktorelemzést mutat. Az analízis eredményei a 3. táblázatban láthatók.

	Faktor				
	1	2	3	4	5
vontatmány tömeg	**,850**	9,278E−02	,147	,107	−,111
fogyasztás	**,812**	,429	,189	−3,899E−02	9,282E−03
motor teljesítmény	**,756**	,534	,250	,190	−,114
öntömeg	**,611**	,510	,508	,191	9,398E−04
max. vonóerő	**,579**	,505	,164	,343	−4,138E−02
összgördülőtömeg	,573	**,642**	,336	,333	−1,561E−02
teherbírás	,497	**,638**	,341	,257	3,476E−02
fajlagos fogyasztás	−,174	**−,607**	2,180E−03	,110	1,716E−02
magasság	,304	−7,684E−03	**,734**	,325	−,117
max. sebesség	−7,879E−02	−6,996E−02	**−,621**	−4,155E−02	,411
teljesítmény dotáció	−,225	−,454	**−,618**	−,380	−6,794E−02
hosszúság	,191	,425	**,525**	,176	6,213E−02
oldaldőlés	−,225	−2,987E−02	−,197	**−,887**	−2,051E−02
mászóképesség	−,314	−,217	−,114	**−,759**	−4,891E−02
terepjárás	−9,629E−02	−1,859E−02	,120	**,543**	−6,386E−02
nyomatéki rugalmasság	−,119	−2,887E−02	−,142	−5,436E−02	**,627**
fordulókör átmérő	,274	,432	,339	,257	,458

3. táblázat. Rotált együtthatómátrix

Az eredmények hasonlóságokat mutatnak a főkomponensanalízis eredményeivel. Hasonlóságot tapasztalhatunk a harmadik, a negyedik, az ötödik faktor, és rendre a második, a harmadik és az ötödik főkomponens között. A fordulókör átmérő esetében mindkét eljárásnál ugyanazt tapasztalhatjuk. Az első főkomponensbe sorolt ismérvek a faktoranalízis során két külön faktorba lettek sorolva, kiegészítve az előzőekben függetlennek ítélt *fajlagos fogyasztással.*

Megfigyelve viszont az első két faktort, az ide sorolt paraméterek a vontatmány tömeg és a fajlagos fogyasztás kivételével, mind a két faktorral magas korrelációt mutatnak. Az eredmények pontosítása érdekében ezért a [2] 264. oldal szerinti iterációt végeztem el. Az ott bemutatott példa szerint az analízisből ki kell hagyni azon paramétereket, amelyekhez tartozó legnagyobb együttható egy meghatározott értéket nem ér el. A bennmaradt paraméterekkel pedig újra el kell végezni az analízist. Ez az érték esetemben 0,6 volt. A többszörös iteráció eredményét a 4. táblázat mutatja.

A KMO teszt eredménye 0,826. A két faktor az iterációk eredményeként megmaradt 8 paraméter teljes szórásnégyzetének a 81,822%-át magyarázza. Vagyis az eddigi eljárások közül ez szolgáltatta a legnagyobb pontosságot. A táblázatot megvizsgálva megállapítható, hogy az eredmények megerősítik a főkomponens analízis során kapott első főkomponenst, a *max. vonóerő* kivételével és megerősítik a harmadik főkomponenst. A terepjárás kiesése magyarázható azzal, hogy ezt alternatív ismérv segítségével vettem figyelembe, ami a jelen eljárás esetében nem használható.

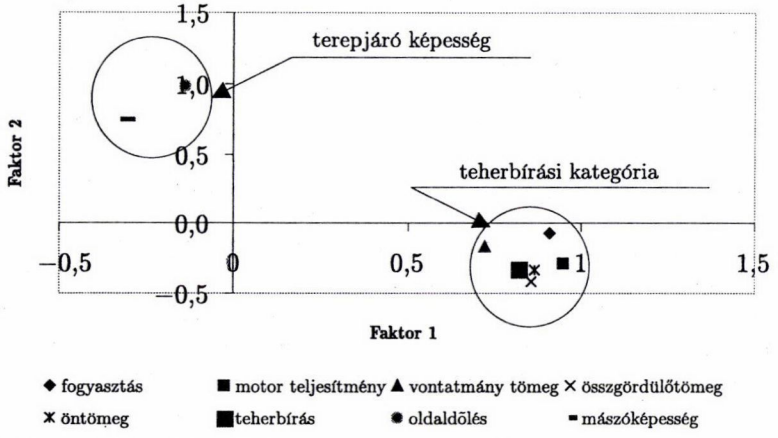
	Faktor	
	1	2
fogyasztás	**,908**	−6,639E−02
motor teljesítmény	**,947**	−,279
vontatmány tömeg	**,724**	−,159
összgördülőtömeg	**,856**	−,419
öntömeg	**,866**	−,335
teherbírás	**,817**	−,324
oldaldőlés	−,141	**,980**
mászóképesség	−,324	**,750**

4. táblázat. Rotált együtthatómátrix

Végezetül megállapítom, hogy a közel valamennyi lehetséges tehergépkocsit reprezentáló minta alapján a vizsgált paraméterekből két olyan csoport szűrhető ki, amelyeket alkotó tagok egymással páronként magas korrelációs viszonyban vannak. Az általuk reprezentált szempontok a következők: *teljesítmény kategória* és *terepjáró képesség*. A mintát alkotó gépjárművekre ezek lesznek az alapvető információt hordozó változók, vagyis ezen két szempont szerint lehet a mintát elkülöníteni. Az eredmények helyességét visszaigazolják a kiindulási adatok, miszerint a mintát különböző teherbírási kategóriájú gépjárművek közúti és terepjáró változatai alkotják.

Figyelembe véve a kiszűrt két csoportot korrelálatlannak tekinthető a *nyomatéki rugalmasság*, nem képezhető egyértelmű csoport a *max. vonóerő*, a *magasság*, a *max. sebesség*, a *teljesítmény dotáció* és a *hosszúság* paraméterekből. A *fordulókör átmérő* esetében egyértelmű korrelációs viszony nem állapítható meg.

Az analízis eredményeként csak két faktor maradt, ezért lehetőség van a faktortér grafikus ábrázolására is (1. ábra).



*1. ábra*1. ábra. A teherbírási kategória és a terepjáró képesség szempontjaihoz tartozó faktortér

4. Összegzés

A két adathalmaz elemzését követően megállapítom, hogy a többváltozós analízis segítségével a műszaki berendezések paraméterei csoportokba foglalhatók. A csoportokon belül a korrelációt kiváltó okok azonosíthatók, és ezen okok alapját képezhetik a vizsgált eszközt leíró szempontrendszer kialakításának. Továbbá az eredmények lehetővé teszik a paraméterek közötti korrelációk segítségével a rendszerben lévő összefüggések feltárását.

Meghatározó az eszközöket leíró paraméterek teljessége, mivel csak ezeknek az információtartalma kerül feldolgozásra, ezért ennek a feltételnek a be nem tartása egyrészről a faktorok által nem teljes paramétercsoportokat és ebből adódóan nem teljes szempontrendszert szolgáltathat.

Az eljárás alkalmazása, illetve az eredmények pontossága függ az egyedek számától. További feltétel, hogy a mintát képző eszközöknek közel azonos funkciókkal és teljesítőképességgel kell rendelkezniük, valamint hasonló műszaki-tudományos színvonalat kell képviselniük.

Az eljárások eredményeként született funkcionális képességeket leíró vetületek segítségével előállnak azok a vizsgálati szempontok, amelyek inputadatát képezhetik egy többszempontú döntési eljárásnak.

Hivatkozások

[1] Éltető, Ö., Meszéna, Gy., Ziermann, M., *Sztochasztikus módszerek és modellek* (Közgazdasági és Jogi Könyvkiadó, Budapest, 1982).

[2] Füstös, L., Meszéna, Gy., Simonné Mosolygó N., *A sokváltozós adatelemzés statisztikai módszerei* (Akadémiai Kiadó, Budapest, 1986).

[3] Jahn, W., Vahle, H., *A faktoranalízis és alkalmazása* (Közgazdasági és Jogi Kiadó, Budapest, 1974)

(Beérkezett: 2003. október 1.)

GYARMATI JÓZSEF
ZRÍNYI MIKLÓS NEMZETVÉDELMI EGYETEM
HADITECHNIKAI ÉS MINŐSÉGÜGYI TANSZÉK
1581 BUDAPEST HUNGÁRIA KRT. 9-11. PF. 15
gyarmati.jozsef@zmne.hu

EXAMINING TECHNICAL EQUIPMENT WITH THE HELP OF FACTOR ANALYSIS

JÓZSEF GYARMATI

In this article, I will investigate the characteristics of technical equipment with the help of factor analysis and principal component analysis. In the process I will use mathematical methods in order to classify the characteristics of technical equipment. Within the individual classes, I will look for the reasons behind the correlation that exists between the characteristics. I will try to prove that those reasons will help identify the main purposes of the equipment under scrutiny.

Alkalmazott Matematikai Lapok **23** (2006), 85–98.

NAGY PONTOSSÁGÚ KÉPFELDOLGOZÁS

CSIRMAZ LÁSZLÓ

Budapest

Egy szürke skálájú képen mintegy háromszáz, nagyjából kör alakú, 10 és 30 pixel közötti átmérőjű folt középpontját és sugarát kell lehető legnagyobb pontossággal meghatározni. A feladat nehézségét az jelenti, hogy ellentétben a képfeldolgozásban szokásos pixeles pontossággal, ennél egy nagyságrenddel nagyobbra van szükség. A megoldásban különböző paraméterű szűrőkkel készített konvolúciók geometriai tulajdonságait használjuk fel; nevezetesen az illeszkedés jóságát a felület lokális maximumában mért görbülete adja meg. A cikkben az újfajta megközelítés elemeit és azok matematikai hátterét ismertetjük.

1. Bevezetés

Egy tipikus képfeldolgozási feladatban egy képen megadott, vagy a megadotthoz hasonló alakzatot kell keresni. Gyakran elegendő csak azt eldönteni, hogy a minta egyáltalán megtalálható-e a képen. A legtöbb ismert eljárás a minta összes előfordulását pár pixeles pontossággal meg is adja. Esetünkben képenként több mint háromszáz, nagyjából kör alakú folt sugarát és helyét kellett pixelnél nagyobb pontossággal megállapítanunk. A feladatot a következő módszer szerint kívánjuk megoldani. Elsőként definiáljuk mintáknak egy egyparaméteres sokaságát: minden szóbajövő r sugárra egy mintát, ami leírja azt, hogyan néz ki egy tipikus r sugarú folt. Ezek után egy gyors, heurisztikus módszerrel megkeressük az összes folt középpontját és sugarát pár pixel pontossággal. Minden sugárra megkeressük, hogy ehhez a sugárhoz tartozó minta hol illeszkedik a legjobban a becsült középpont közelében, majd kiválasztjuk azt a sugarat és a hozzátartozó középpontot, ahol ez az illeszkedés a lehető legjobb. A továbbiakban ennek megvalósítása során felmerülő problémákat ismertetjük azzal együtt, hogyan oldottuk meg, illetve hogyan kerültük meg a problémát.

Foltok helyének közelítő meghatározása nem igényel a képfeldogozásban ismert és használt módszereken túlmutató ötletet, így annak ismertetésétől eltekintünk. A 2. részben foglaljuk össze mindazt a szükséges előismeretet, amit a szűrőkről és a konvolúcióról – a képfeldolgozás alapvető módszereiről – fel fogunk használni. A 3. rész taglalja, hogyan kell megválasztani a különböző sugarakhoz tartozó mintákat, továbbá hogyan kereshetjük meg a foltok középpontját pixel pontossággal. A középpontokat pixelnél nagyobb pontossággal a 4. részben határozzuk meg. Az 5. rész az r sugár meghatározásáról szól. Míg fix sugár mellett a legjobb illeszkedés helyét a konvolúciós felület lokális maximumai szolgáltatják, addig különböző sugarak mellett a lokális maximum értéke nem tükrözi vissza, hogy mennyire jó az illeszkedés. Ennek mérésére a konvolúciós felület másik jellemzőjét, a maximumban mért *görbületet* használjuk. Kiderül azonban, hogy a lokális maximum helyét megfelelő pontossággal becsülő eljárás túl nagy hibát vét a görbület esetében. A 6. részben megmutatjuk, hogyan lehet más módszerrel, megfelelő pontossággal a görbületet is számítani. Az utolsó, 7. részben összefoglaljuk a feladat megoldását, és néhány nyitva maradt kérdést vetünk fel.

2. Szűrők

A képfeldolgozás során a leggyakrabban használt módszer *szűrők* alkalmazása [2, 6]. A képet egy $p(x, y)$ kétváltozós valós síkon értelmezett függvénynek tekintjük; a függvény értéke a kép (x, y) pontbeli szine. Mivel fekete-fehér képről van szó, a szín egy 0 és 1 közötti valós szám, ahol 0 jelenti a feketét, 1 pedig a fehéret. A $p(x, y)$ függvényről feltesszük, hogy a sík megfelelően nagy tartományán (vagy akár a teljes síkon) értelmezve van, és mindazon feltételeknek eleget tesz, amik biztosítják, hogy a felhasznált tételek igazak legyenek (például p akárhányszor deriválható, vagy négyzetesen integrálható, stb.). A feldolgozandó képen a p függvénynek az egész koordinátájú rácspontok egy téglalap alakú részén felvett értékeit találjuk valamekkora hibával. A hiba természetével és a kép minőségének javításával (zajszűrés, kontrasztjavítás) nem foglalkozunk.

Egy *szűrő* mindenütt értelmezett, sima, négyzetesen integrálható f függvény, ami rendszerint, de nem feltétlenül még centrálszimmetrikus is (vagyis $f(x, y) = f(-x, -y)$ minden x, y valós számpárra). Az f-et eltoljuk az (u, v) pontba, majd az eltolt és tükrözött f-fel mint súllyal átlagoljuk a p képet:

$$P(u, v) = \int_{\mathbf{R}^2} p(x, y) f(u - x, v - y)\, dx\, dy. \tag{1}$$

Ez az úgy nevezett *konvolúciós integrál* a $p(x, y)$ kétváltozós függvényhez a $P(u, v)$ kétváltozós függvényt rendeli; szokásos még a $P = \langle p, f \rangle$ jelölés is. Könnyű látni, hogy $\langle p, f \rangle$ szimmetrikus: $\langle p, f \rangle = \langle f, p \rangle$, valamint mindkét argumentumában lineáris, például $\langle p, c_1 f_1 + c_2 f_2 \rangle = c_1 \langle p, f_1 \rangle + c_2 \langle p, f_2 \rangle$. Az f szűrő a konvolúció *magfüggvénye*.

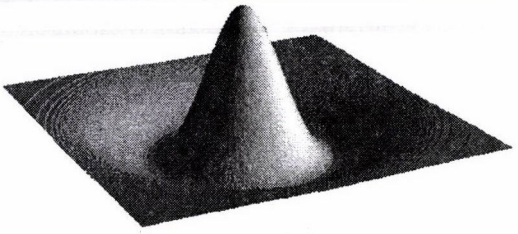
A σ szórású (kétdimenziós) *Gauss-szűrő* az alábbi függvény:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Szokás szerint a konstanst úgy választottuk meg, hogy a szűrő teljes síkon vett integrálja (vagyis a szűrő L_1 normája) éppen 1 legyen. Előszeretettel használnak kis szórású Gauss-szűrőt a kép hibáinak csökkentésére; nagyobb szórású szűrővel a képet „lágyítani" lehet, vagyis az éleket elmosni (blurring). Még nagyobb szórású Gauss-szűrővel vett konvolúció értéke jól méri egy képpont környezetének átlagos szürkeségét. Például az élesítésnek (sharpening) hívott kontrasztjavító eljárásnál a kép (u,v) pontbeli értéket úgy módosítjuk, hogy az ottani átlagtól való eltérés $c > 1$-szeresére nőjön. Az átlagot egy σ_1 szórású Gauss-szűrővel való konvolúció adja meg, az (u,v) pontbeli értéket pedig egy σ_2 szórású Gauss-szűrő (persze σ_1 nagyobb, mint σ_2). A kontrasztjavító eljárás eredménye az (u,v) pontban:

$$\langle p, G_{\sigma_1}\rangle + c \cdot \big(\langle p, G_{\sigma_2}\rangle - \langle p, G_{\sigma_1}\rangle\big) = \big\langle p, (1-c)G_{\sigma_1} + cG_{\sigma_2}\big\rangle,$$

felhasználva a konvolúció linearitását. A javítás tehát egyetlen konvolúcióval is számolható. A két Gauss-szűrő különbsége úgy néz ki, mint egy mexikói kalap (1. ábra).



1. ábra. Két Gauss-szűrő különbsége

Gyakorlatban a számítások gyorsítása érdekében a szűrőket egészen kis méretű, egész értékű mátrixszal közelítik. A kontraszt növelésére például az alábbi 3×3-as mátrixot szokás használni különböző c konstansokkal:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & c & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Visszatérve a $\langle p, f\rangle$ konvolúcióra, ennek (u,v)-beli értékét úgy is tekinthetjük, mint annak a mértékét, hogy ebben a pontban p mennyire hasonlít az f szűrőhöz. Nézzük ugyanis p valamint f eltolt (és tükrözött) képe közötti különbség négyzetes

integrálját a teljes síkon:

$$\int \big(p(x,y) - f(u-x, v-y)\big)^2 \, dx\, dy =$$
$$= \int p^2(x,y)\, dx\, dy + \int f^2(x,y)\, dx\, dy - 2\langle p, f\rangle.$$

A jobb oldalon az első két integrál értéke független az (u,v) ponttól, tehát a négyzetes eltérés a konvolúció (-2)-szeresétől csak egy additív konstansban tér el. Minél nagyobb tehát a konvolúció, annál jobban hasonlít a p az f szűrőfüggvényre. A konvolúciónak ezt a tulajdonságát kihasználva tudunk egy képen adott mintához hasonló alakzatokat keresni. A mintát felvesszük szűrőnek; ahol a konvolúciónak (lokális) maximuma van és értéke meghalad egy bizonyos küszöböt, ott várható a minta megjelenése. Természetesen az eljárás csak az adott minta eltoltjait tudja csak megtalálni. Szerencsére esetünkben a keresendő alakzatok forgásszimmetrikusak, így nem kellett a minta elforgatásából adódó problémával küszködnünk.

Gyakorlatban a konvolúció kiszámítása numerikus integrálást jelent. A p függvény értékét csak egész helyeken ismerjük (és ott is csak valamilyen hibával), ezért az integrált az integrálandó függvény rácspontokon felvett értékeinek összegeként közelítjük:

$$P(u,v) \approx \tilde{P}(u,v) = \sum_{i,j \in \mathbf{Z}^2} p(i,j) f(u-i, v-j).$$

Az f szűrő általában mindenütt (és nem csak a rácspontokon) van definiálva, tehát $\tilde{P}$ értékét tetszőleges (u,v) pontban, és nem csak a rácspontokon tudjuk számítani.

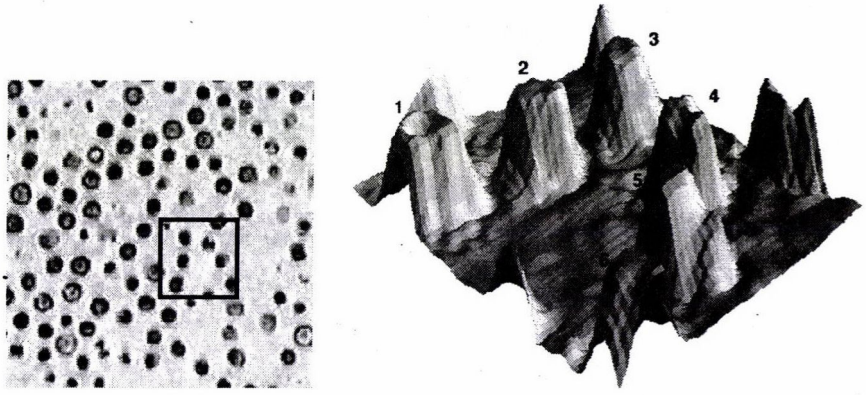
3. Lokális maximum keresése pixel pontossággal

Egy feldolgozandó kép tipikus részletét láthatjuk az 2. ábra bal oldalán. A jobb oldalon a kiemelt rész háromdimenziós képe látható; a foltokat a későbbi utalások kedvéért megszámoztuk.

A keresendő alakzatok nem homogén foltok, hanem gyakran gyűrű alakúak, melyeket egy nagyon keskeny, a háttérnél világosabb csík vesz körül. Az egyes számú (bal alsó sarokban található) folt közepén is megfigyelhető egy kráter. A foltok hozzávetőleges helyét és sugarát egy gyors, heurisztikus algoritmussal megkeressük. Ezután veszünk egy r sugárhoz tartozó f_r szűrőt, és a középpont becsült helyéből indulva megkeressük azt a legközelebbi (u,v) rácspontot, ahol a

$$\tilde{P}_r(u,v) = \sum_{i,j} p(i,j) f_r(u-i, v-j)$$

közelítő összegnek lokális maximuma van. Amennyiben mind a sugarat, mind a folt helyét megfelelően jól becsültük meg, a lokális maximum helye jó közelítést ad a középpontra.

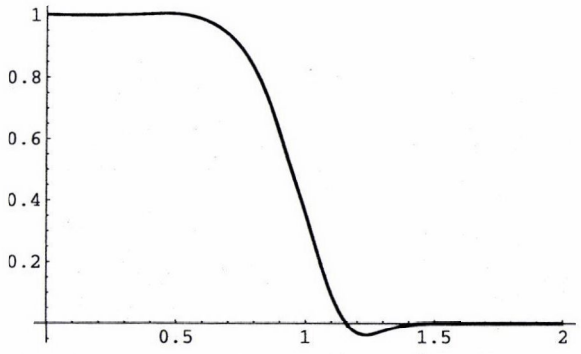


2. ábra. A feldolgozandó kép és részének 3D képe

Mint az előző részben láttuk, az ideális f_r szűrő megegyezik az r sugarú folt alakjával. A foltok valamilyen fizikailag létező objektumok képei, túl nagy változatosságot mutatnak, és nincs is rá esély, hogy valamilyen szép függvénnyel le tudjuk őket írni. Ezért olyan f_r szűrőt használunk, amely egyrészt analitikusan kezelhető, másrészt jól használható a folt középpontjának és sugarának meghatározására. Ez azt jelenti, hogy a szűrő viszonylag nagy és egyenletes meredekségű a folt szélének környékén – vagyis az origótól r távolságban –, attól távolabb és az origó környezetében pedig lényegében vízszintes kell legyen. Minél meredekebb a szűrő, annál jobban kiemelkedik az illeszkedés helye. Ugyanakkor túl meredek szűrő már túlságosan érzékeny a pontos r értékre és a folt szabályos alakjára: kicsit eltérő sugár vagy bizonytalan alakú folt esetén a lokális maximum nagyon eltolódhat az igazi középponthoz képest.

További problémát okoz, hogy nem egyetlen szűrőt kell megadnunk, hanem minden szóbajövő r sugárra egyet-egyet. Különböző szűrők által adott eredményeket kell összevetnünk, ami azt jelenti, hogy a szűrőket megfelelően kell normálni. De milyen normát válasszunk? A kép átlagos szürkeségét azok a szűrők tartják meg, melyek L_1 normája (a tejes síkon vett integrálja) éppen 1. Ha viszont a konvolúciót úgy tekintjük, mint ami a kép és a szűrő közötti négyzetes eltérést méri, akkor a szűrők L_2 normáját kellene egyenlővé tennünk. Persze esetünkben nem szükségképpen a négyzetesen legjobban közelítő szűrő illeszkedik legjobban. Ennek oka a folt alakjában mutatkozó bizonytalanság és a folt közepén megjelenő egyenetlenség. A norma megfelelő megválasztásának kérdését megkerülhetjük, ha az f_r szűrőt f_1 megfelelően felnagyított képének választjuk: $f_r(u,v) = f_1(u/r, v/r)$. Ekkor ugyanis f_r tetszőleges normája f_1 megfelelő normájának r^2-szerese. Ennek a választásnak hátránya viszont, hogy f_r-nek sugár távolságban mért meredeksége lineárisan függ r-től: minél nagyobb a sugár, a szűrő annál kevésbé érzékeny kis elmozdulásokra.

A keresett foltok körszimmetrikusak, tehát szűrőink is azok lesznek. Az előző érvelésnek megfelelően f_r-et úgy választjuk, hogy egy valós egyváltozós $\varphi(x)$



3. ábra. A szűrőt generáló $(1 - x^8/4)e^{-x^4}$ függvény gráfja

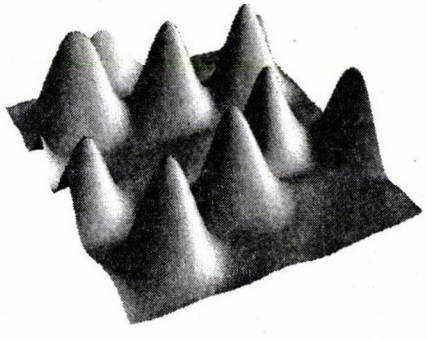
függvényt r-szeresére nyújtva megforgatunk az y tengely körül:

$$f_r(u,v) = \varphi\left(\frac{\sqrt{u^2+v^2}}{r}\right).$$

φ-nek olyan függvényt kell választani, mely a 0 környékén nagyjából konstans, az 1-hez közeledve meredeken csökken, majd nem sokkal 1 fölött gyorsan belesimul az x tengelybe. Az általunk választott

$$\varphi(x) = \left(1 - \frac{x^8}{4}\right)e^{-x^4}$$

függvény $x \approx 0{,}7$-ig vízszintesen halad, onnan elindul lefelé. $x = \sqrt[4]{2} \approx 1{,}18$ körül átmetszi az x tengelyt, 1,5 fölött pedig már elenyészően kicsi (3. ábra). A kis negatív tartomány a folt körüli keskeny világosabb sávnak felel meg. A további elemzéseket erre függvényre tesszük; következtetéseink azonban más szűrőkre is igazak.



4. ábra. A kép $r = 7{,}5$ paraméterű szűrő alkalmazása után

Alkalmazva az f_r szűrőt az $r = 7{,}5$ értékkel, a kapott felület háromdimenziós képét a 4. ábra mutatja. Jól láthatók a markáns süvegek, melyek csúcsa felel meg

a konvolúció lokális maximumainak, vagyis a csúcsok koordinátái adják meg a foltok középpontjait. A szűrő eltüntette az eredeti képen a foltok közepén található bemélyedéseket, a folt egyenetlenségeit és bizonytalanságokat.

A $P = \langle p, f_r \rangle$ konvolúció lokális maximumait – vagyis a süvegek csúcsait – pixel pontossággal egyszerűen meghatározhatjuk. A maximumhoz elegendően közelről indulva rácspontokon lépkedünk. Minden lépésben a rácspontnak olyan szomszédjába megyünk, ahol a P konvolúció nagyobb értéket vesz fel. Amikor megakadunk, megtaláltuk a lokális maximumot. Az algoritmus gyorsítható, ha például minden lépésben először abban az irányban próbálkozunk, amit az előző lépésben használtunk. Robusztusabbá is tehető az algoritmus, ha lépésenként az összes közvetlen (vagy másod-, harmad-) szomszéd közül választjuk ki a maximálisat.

4. Nagyobb pontosság felület illesztésével

A $P_r = \langle p, f_r \rangle$ konvolúció lokális maximumainak helyét pixelnél nagyobb pontossággal is megbecsülhetjük. Tegyük fel, hogy az (u_0, v_0) rácspontban P_r értéke nagyobb, mint a környező rácspontokban. A $P_r(u_0 + x, v_0 + y)$ függvényt egy

$$G(x, y) = A + Bx + Cy + Dxy + \frac{1}{2}Ex^2 + \frac{1}{2}Fy^2 \tag{2}$$

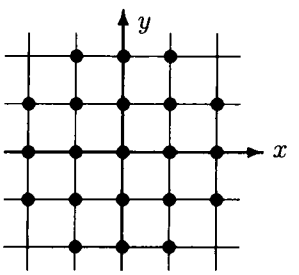
alakú másodfokú felülettel közelítjuk, és P_r lokális maximumát $(u_0 + x_0, v_0 + y_0)$-lal becsüljük, ahol (x_0, y_0) az a hely, ahol $G(x, y)$ a lehető legnagyobb.

Szokás szerint G együtthatóit úgy határozzuk meg, hogy $P_r(u_0 + x, v_0 + y)$ és $G(x, y)$ eltérésének négyzetösszege az origó körüli néhány rácspontban a lehető legkisebb legyen. Ha $\mathcal{A}$ ezen rácspontok halmaza, akkor a minimalizálandó kifejezés

$$\sum_{(i,j)\in\mathcal{A}} \big(G(i, j) - P_r(i + u_0, j + v_0)\big)^2.$$

Ennek az A, B, C, D, E és F együtthatók szerinti parciális deriváltjai el kell tünjenek. Ez hat lináris egyenlet az ismeretlen együtthatókra, amiből azok meghatározhatók. Amennyiben az $\mathcal{A}$ halmaz szimmetrikus mind a két tengelyre, az egyenletrendszer harminchat együtthatójából 24 nulla lesz, és az egyenletrendszer akár kézzel is könnyen megoldható.

Az $\mathcal{A}$-ra jó választás pédául az a 21 elemű halmaz, ami az origó körüli 5-ször 5-ös rácsnégyzet pontjaiból áll annak négy sarkát kivéve:



A $G(x,y)$ együtthatóinak meghatározása után megkeressük G maximumát. Ezt abban az (x_0,y_0) pontban veszi fel, ahol az G-nek x és y szerinti parciális deriváltja egyaránt eltűnik:

$$\frac{\partial G}{\partial x} = B + Dy + Ex = 0, \qquad \frac{\partial G}{\partial y} = C + Dx + Fy = 0.$$

Az egyenletrendszer megoldása

$$x_0 = \frac{CD - BF}{EF - D^2}, \quad y_0 = \frac{BD - CE}{EF - D^2}. \tag{3}$$

A nevezőben álló $d = EF - D^2$ érték a G másodfokú felület fontos jellemzője. Ennek előjele azonosítja a felület típusát a (3) szerinti (x_0,y_0) pontban. Ha d pozitív, G-nek az (x_0,y_0) pontban vagy abszolút maximuma, vagy abszolút minimuma van, attól függően, hogy E (és vele együtt F) negatív vagy pedig pozitív. Ha d nulla vagy negatív, akkor G-nek egyátalán nincs lokális szélsőértéke, és így speciálisan (x_0,y_0) sem lehet az.

Mindezeket összerakva a P_r konvolúció lokális maximumának meghatározására az alábbi algoritmust kapjuk. Az (u_0,v_0) rácspont meghatározása után az $\mathcal{A}$ rácspontokban négyzetesen legjobban illeszkedő G másodfokú felület (2) szerinti együtthatóit kiszámítjuk. Ha a felület $d = EF - D^2$ diszkriminánsa nulla vagy negatív, az illesztett felületnek (u_0,v_0) környékén nincs szélsőértéke, ami az jelenti, hogy a P_r konvolúciónak sincs itt szignifikáns maximuma. Ha d pozitív, a (3) alapján számított (x_0,y_0) pontban van G-nek szélsőértéke. Még érdemes ellenőrizni, hogy x_0 és y_0 abszolút értékben kisebb 1-nél (tulajdonképpen az 1-hez vagy -1-hez közeli érték sem igazán elfogadható). Ha így van, az $(u_0 + x_0, v_0 + y_0)$ pont a P_r lokális maximumának jó közelítése.

5. A sugár meghatározása – Gauss-görbület

Amennyiben ismerjük az r sugarat, az előző pontban ismertetett eljárással a folt középpontját már megfelelő pontossággal meg tudjuk határozni. Feladatunk tehát r meghatározása.

Mint láttuk, a P_r konvolúció az f_r szűrő és a kép négyzetes eltérését méri. Természetesen adódik, hogy azt az r-et fogadjuk el a sugár értékének becslésére, amire P_r-nek előző pontban leírt algoritmus szerint számított lokális maximuma a lehető legnagyobb. A tapasztalat szerint azonban ez nem működik: a lokális maximumban felvett érték r-ben nagyon enyén, de monoton csökken. Ennek oka elsősorban az, hogy a foltok alakja r-rel nem lineárisan változik, míg az f_r szűrőket f_1-ből lineáris nagyítással kapjuk. Így az optimális r sugár esetén f_r már nem feltétlenül illeszkedik négyzetesen a lehető legjobban.

Megoldásként egyik lehetőség az, hogy az f_r szűrőket másképp definiáljuk. Például nagyszámú foltról statisztikát készítve próbáljuk meg azok tipikus alakját meghatározni – ehhez viszont a statisztikában részt vevő foltok pontos sugarát tudnunk kell. Ha ezt a triviálisnak egyátalán nem tünő problémát megoldottuk (például klaszterezéssel csoportosítjuk a nagyjából egyforma sugarú foltokat), újabb problémával kerülünk szembe: a szűrőket normálni kell. Tökéletes alakú szűrő és hiba nélküli $p(x, y)$ értékek esetén persze a szűrők L_2 normáját kell egyenlővé tenni. A mérési hibák hatását az L_1 normán keresztül tudjuk kontrollálni. Mivel az L_1 és L_2 norma teljesen más, a normálást ismét az adatok alapján kell beállítani.

Másik lehetőségünk az, ha az illeszkedés jóságát másképpen mérjük. Ehhez vegyük jobban szemügyre a (2) felület (x_0, y_0) lokális szélsőértékét. Hogy ez a szélsőérték mennyire szignifikáns, a pozitív $d = EF - D^2$ érték mutatja meg. Ha d kicsi, akkor G maximumát egy lapos platón alig kiemelkedve veszi fel. Minél nagyobb a d, a felület annál meredekebb, hegyesebb (x_0, y_0)-ban. Ez nem véletlen, hiszen d éppen G-nek az (x_0, y_0)-beli *Gauss-görbülete*, lásd [3, 4].

Húzzunk érintő síkot a G felület egy (x, y) pontjában, és az egyszerűség kedvéért tegyük fel, hogy a felület az érintési pont egy környezetében az érintő síknak ugyanarra az oldalára esik. (Ez a helyzet például, ha (x, y) lokális maximum.) Ha most az érintő síkot önmagával párhuzamosan ε távolságra eltoljuk a felület felé, akkor a felületet és a sík metszésvonala közel ellipszis alakú lesz. A metszet ellipszis kis- és nagytengelyének félhosszát emeljük négyzetre, a négyzeteket szorozzuk össze, majd a szorzatot normáljuk le. Mivel mindkét tengely hossza $\varepsilon^{1/2}$ nagyságrendű (érintő síkon vagyunk), a normálás $4\varepsilon^2$-nel való osztást jelent. (A mágikus 4-es szorzó megjelenésének okát lásd alább.) A Gauss- – vagy másképpen szorzat- – görbület a normált szorzat reciprokának határértéke, amint ε tart a nullához.

A Gauss-görbületet másképpen is szokás definiálni. A felület (x, y) pontjában az érintő síkra merőleges egyenest emelünk. Ezen a normálison át fektett összes síkon meghatározzuk a felület és a sík metszetének a görbületét. A görbületek közül a minimális és a maximális két egymásra merőleges sík esetén fordul elő. A szorzatgörbület ennek a két szélsőértéknek a szorzata. Az *összeg-* vagy *Minkowski*-görbület pedig a minimális és maximális görbület összege. Az, hogy a Gauss-görbület két

definíciója ekvivalens, abból következik, hogy limeszben a fenti ellipszisek kis- és nagytengelyei éppen a maximális, illetve minimális görbületet adó síkokban vannak, továbbá ezek a görbületek annak a hányadosnak a határértéke, mikor 2ε-t a megfelelő féltengely hosszának négyzetével osztjuk. (Innen adódik a fenti $2\varepsilon \cdot 2\varepsilon = 4\varepsilon^2$ normáló tényező.) Ha az $F(x,y)$ függvény által definiált felület egy pontjában az x és y szerinti parciális derivált is eltünik, akkor abban a pontban a Gauss-görbület

$$\frac{\partial^2 F}{\partial x^2} \cdot \frac{\partial^2 F}{\partial y^2} - \left(\frac{\partial^2 F}{\partial x \partial y}\right)^2, \tag{4}$$

egyezésben azzal, hogy a (2) másodrendű G felület (x_0, y_0) lokális szélsőértékében ez az érték éppen $EF - D^2$, hiszen G második parciális deriváltjai rendre E, F, and D. Hasonlóképpen ugyanebben a pontban a Minkowski-görbület

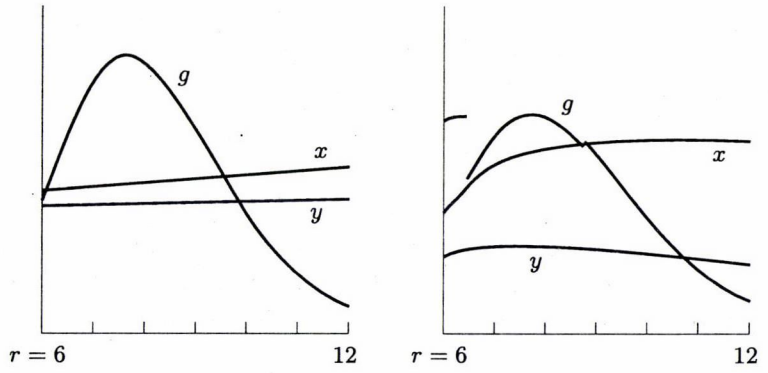
$$\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2}. \tag{5}$$

Mind a Gauss-, mind a Minkowski-görbület a felület csúcsosságát, hegyességét méri. Gauss nevezetes tétele szerint a Gauss-görbület invariáns a felület hajlítgatásaival szemben, azt a felület belső geometriája meghatározza, szoros kapcsolatban van a felületre rajzolt háromszögek szögösszegével. Ugyanakkor egyik irányban hosszan elnyúló felületnél a Gauss-görbület majdnem nulla, függetlenül attól, hogy erre merőlegesen a felület mennyire meredek. Ilyenkor a Minkowski-görbület jobban használható a felület görbültségének mérésére. Esetünkben a feldolgozandó foltok nagyjából kör alakúak, így P_r lokális maximumaiban a minimális és maximális görbület várhatóan közel van egymáshoz. Következésképp a kétfajta görbület ezeken a helyeken egyformán fog viselkedni.

Az illesztett G felület maximumában a görbület egyúttal becslést ad a P konvolúciós felület görbületére is. Minél nagyobb ez a görbület, annál szignifikánsabb a lokális maximum. Ez az észrevétel sugallja, hogy a különböző sugarakat annak alapján hasonlítsuk össze, hogy mekkora a lokális maximumban a görbület. Az optimális sugár a maximális görbülethez tartozik: mind kisebb, mind nagyobb sugárhoz tartozó szűrő ezt a maximumot egy kissé „elkeni", és így a görbület kisebb lesz.

Ezek a meggondolások a következő eljárást sugallják egy folt helyének és sugarának becslésére. Legyen r_0 valamint (u_0, v_0) az előzetesen becsült sugár, illetve középpont. Az r_0 körül választunk különböző r értékeket. Mindegyikkel elkészítjük a folt környékének az f_r szűrővel való P_r konvolúcióját. Az (u_0, v_0)-ból indulva megkeressük azt a rácspontot, ahol P_r-nek lokális maximuma van. Ott a 4. részben leírtak szerint kiszámítjuk a négyzetesen legjobban illeszkedő G másodfokú felületet, és azt, hogy hol veszi fel maximumát, illetve hogy ott mekkora a görbülete. Azt az r-et fogadjuk el a sugár közelítésének, amelyikre a görbület a lehető legnagyobb, és az ekkor adódó maximum helye adja meg a folt középpontját.

Az eljárás eredményét a hármas és négyes számú foltokra az 5. ábra mutatja. A vízszintes tengelyen az r sugár fut 6-tól 12-ig. A g-vel jelölt görbe a Gauss-görbület, az x, illetve y pedig a lokális maximum helye. Mindkét esetben jól látható



5. ábra. Maximumhely és Gauss-görbület a hármas és négyes folt esetén

a görbület maximuma $r = 7{,}6$, illetve $r = 7{,}7$ esetén, tehát ezek az értékek adják meg a foltok sugarát. A hármas foltnál g, x és y várakozásainknak megfelelően viselkedik: x és y nagyjából egyenletesen nő: miközben a sugár $r = 6$-ról $r = 12$-re nő, x mindegy kétharmad, y pedig egynegyed pixelnyit mozdul el. Eközben a g görbület először meredeken nő, majd a maximum elérése után meredeken zuhan. A négyes foltnál x és y nagyjából folytonosan követi r-et, x összesen másfél pixelnyit, y nagyjából egyharmad pixelnyit mozog; a görbületnek viszont $r = 6{,}5$-nél egy nagyobb, $r = 8{,}8$ körül pedig egy kisebb ugrása van. Bár az ugrások ellenére a görbületnek jól meghatározott maximuma van, ami persze a keresett sugarat is megadja, általános esetben a szakadások miatt az eljárás egyátalán nem, vagy csak igen körülményesen alkalmazható.

6. A végső eljárás

A görbületnél adódó szakadások okát vizsgálva nézzük meg egy kicsit pontosabban a fenti eljárást. Először is kiválasztjuk az r sugárhoz tartozó f_r szűrőt, majd az előzetesen becsült rácspontból indulva keresünk egy közeli (u_r, v_r) rácspontot helyet, ahol a konvolúciós integrál $\tilde{P}_r$ közelítő értékének lokális maximuma van. Ezután meghatározzuk azt a G_r másodfokú felületet, amely az (u_r, v_r) rácspontnak egy (21 rácspontot tartalmazó) környezetében négyzetesen a legjobban illeszkedik $\tilde{P}_r$-re. Végül kiszámítjuk G_r-nek a maximumát, valamint azt, hogy a maximumban mennyi G_r görbülete. A görbületben akkor tapasztalunk szakadást, mikor az (u_r, v_r)-beli lokális maximum éppen átugrik egyik rácspontból egy másikba. Az illesztett G_r felület, míg helyesen ad számot arról, hogy hol is található $\tilde{P}_r$ lokális maximuma, már nem elég jó ahhoz, hogy $\tilde{P}_r$-nek a maximumbeli görbületét is jól megbecsülje. $\tilde{P}_r$ görbületét tehát közvetlenül kell kiszámítani. A (4) és (5) képletek szerint ehhez elegendő $\tilde{P}_r$ második parciális deriváltjainak ismerete.

Láttuk a 2. rész végén, hogy ha a szűrő mindenütt értelmezve van (ami esetünkben fennáll), akkor $\tilde{P}_r$ értékét nem csak rácspontokban, hanem tetszőleges (x, y) pontban tudjuk számítani a következőképpen:

$$\tilde{P}_r(x,y) = \sum_{i,j\in\mathbf{Z}^2} p(i,j) f_r(x-i, y-j).$$

Ennek alapján $\tilde{P}_r$ parciális deriváltjai a jobb oldal tagonként deriválásával adódik. Így $\tilde{P}_r$ egy parciális deriváltja annak a konvolúciónak az értéke, melyben a magfüggvény az f_r szűrő megfelelő parciális deriváltja. Így például

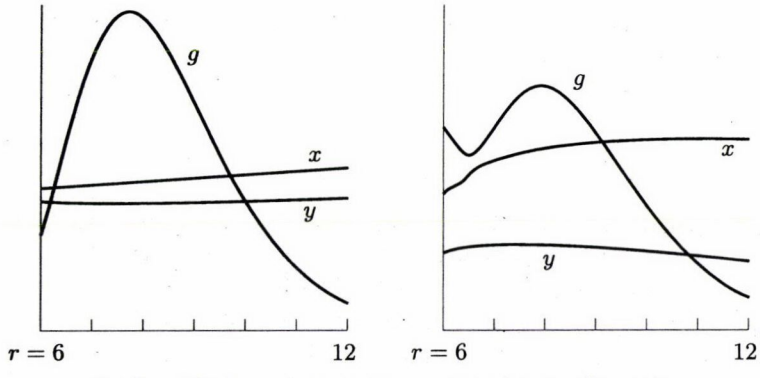
$$\frac{\partial^2}{\partial x^2} P_r = \left\langle p, \frac{\partial^2}{\partial x^2} f_r \right\rangle, \qquad \text{illetve} \qquad \frac{\partial^2}{\partial y^2} P_r = \left\langle p, \frac{\partial^2}{\partial y^2} f_r \right\rangle.$$

Ezeknek a konvolúcióknak az értéke pedig tetszőleges pontban számítható.

Tekintsük a $\tilde{P}_r$ függvénynek az (u, v) pont körüli Taylor sorát, és vegyük abban a legfeljebb másodrendű tagokat:

$$(6)\quad \tilde{P}_r(u+x, v+y) \approx \tilde{P}_r(u,v) + \frac{\partial \tilde{P}_r}{\partial x} x + \frac{\partial \tilde{P}_r}{\partial y} y + \frac{\partial^2 \tilde{P}_r}{\partial x \partial y} xy + \frac{1}{2}\frac{\partial^2 \tilde{P}_r}{\partial x^2} x^2 + \frac{1}{2}\frac{\partial^2 \tilde{P}_r}{\partial y^2} y^2,$$

$\tilde{P}_r$-nek ebben a másodrendű közelítésében az összes együtthatót tetszőleges (u, v) pontban ki tudjuk számítani.



6. ábra. Maximumhely és Gauss-görbület iterálás után

Ha (u, v)-nek az a pontot választjuk, amelyet a 5. részben ismertetett eljárás eredményez, akkor $\tilde{P}_r$ lokális maximumának még jobb közelítését adja a (6) másodrendű felület maximuma. Ebben a maximumban (6) Gauss-görbülete

$$(7)\qquad \frac{\partial^2 \tilde{P}_r}{\partial x^2} \cdot \frac{\partial^2 \tilde{P}_r}{\partial y^2} - \left(\frac{\partial^2 \tilde{P}_r}{\partial x \partial y}\right)^2$$

jobb közelítést ad $\tilde{P}_r$ görbületére. A 6. ábra mutatja az ennek alapján számított maximumhelyet és Gauss-görbületet. A maximum helye pár század pixellel mozdult el a négyzetesen legjobban illeszkedő felület maximumától, míg a görbület értéke jól láthatóan kisimult. $\tilde{P}_r$ görbületének ez a közelítése tehát már használható a folt sugarának megállapítására.

A (6) felület hat együtthatója hat konvolúció kiszámítását jelenti, melyekben a magfüggvények értékeit helyben kell kiszámítani. Amennyiben elfogadjuk, hogy (u, v) megfelelően jól közelíti a lokális maximumot, akkor csak a görbületet kell számítanunk. A Gauss-görbület esetén (7) szerint ez három konvolúciót jelent, míg a Minkowski görbület esetén – a konvolúció linearitása miatt – ez egyetlen konvolúcióval is számítható:

$$\frac{\partial^2 \tilde{P}_r}{\partial x^2} + \frac{\partial^2 \tilde{P}_r}{\partial y^2} = \sum_{i,j \in \mathbf{Z}^2} p(i,j) \left(\frac{\partial^2 f_r}{\partial x^2} + \frac{\partial^2 f_r}{\partial y^2} \right) (x - i, y - j). \tag{8}$$

7. Összefoglalás

A kitűzött feladat, miszerint határozzuk meg a képen található foltok helyét és nagyságát pixelnél nagyobb pontossággal, a bevetésben vázolt módszerrel megoldható. A 3. részben tárgyaltuk, hogy az f_r szűrőknek milyen feltételeket kell kielégíteniük. Fix r érték mellett a legjobb illeszkedést a kép és az f_r szűrő konvolúciójának maximuma adja. A maximum helyét pixelnél nagyobb pontossággal megkaphatjuk, ha a konvolúciós felületet egy négyzetesen legjobban illeszkedő másodrendű felülettel helyettesítjük. Természetesen a négyzetesen legjobban illeszkedő felület helyett használhatnánk a 6. részben bemutatottak mintájára azt a másodfokú polinomot, melynek együtthatói a rácspontban kiszámított parciális deriváltak. A maximumban mind a Gauss-, mind a Minkowski-görbület jó mérőszáma annak, hogy mennyire jó az illeszkedés. Azt az r sugarat kell választanunk, amire a görbület maximális.

A görbület rosszul viselkedhet abban az esetben, mikor az r sugár pár pixellel kisebb a folt valódi sugaránál, és a folt belsejében egyenetlenségek vannak. Ezért a követendő eljárás az, hogy r-et nagyobbra választjuk, majd addig csökkentjük, amíg a görbület nő.

Érdekes lenne megvizsgálni, hogy a p függvény, vagyis a kép hibái hogyan mutatkoznak a lokális maximum, illetve a görbület értékében. Szokás szerint a hibáról feltesszük, hogy pixelenként független, kis szórású, nulla várható értékű normális eloszlásból származik. Elképzelhető, hogy a négyzetesen illeszkedő felület kisebb hibával adja meg a lokális maximumot, mint a harmadrendű tagoknál levágott hatványsor, különösen ha a hiba nagy lehet.

A másodendű közelítést arra használtuk, hogy a konvolúciós felület lokális maximumát megkeressük, vagyis olyan (u, v) pontot, ahol a felület mindkét parciális

deriváltja nulla. Elképzelhető, hogy erre egy közvetlen, gyors algoritmus is adható, mely ezeknek a parciális deriváltaknak explicit alakját használja.

Egy folt sugarának megállapításához a görbület függvény maximumát kell megkeresni. Ha ehhez a (8) szerinti Minkowski-görbületet használjuk, akkor (8) analitikusan deriválható, és a derivált zéróhelyét kell keresni. Egy egyváltozós függvény zérushelyét numerikusan jóval egyszerűbb meghatározni, mint a maximumát. Lehet-e ezt az észrevételt egy használható algoritmussá fejleszteni?

Irodalomjegyzék

[1] Winkler, G., *Image Analysis, Random Fields and Dynamic Monte Carlo Methods* (Springer, Heidelberg, 1996).

[2] Gonzalez, R. C. and Woods, R. E., *Digital Image Processing* (Addison Wesley, 1992).

[3] Hajós, Gy., *Differenciálgeometria*, egyetemi jegyzet (Tankönyvkiadó, Budapest, 1971).

[4] Lánczos, K., *A geometriai térfogalom fejlődése* (Gondolat, Budapest, 1976).

[5] Polzehl, J. and Spokoiny, V., *Image denoising: pointwise adaptive approach*, preprint (Weierstass Institute, 2001).

[6] Fisher, R., Perkins, S., Walker, A. and Wolfart, E., *Hypermedia image processing reference*, `http://www.dai.ed.ac.uk/HIPR2`

(Beérkezett: 2004. január 3.)

CSIRMAZ LÁSZLÓ
KÖZÉP-EURÓPAI EGYETEM
1051 BUDAPEST
NÁDOR U. 9.
csirmaz@ceu.hu

SUBPIXEL IMAGE PROCESSING

LÁSZLÓ CSIRMAZ

During a recent research project we had to develop a method which can determine the center and radius of more than three hundred circular patches on a grayscale image. This had to be done with precision exceeding the resolution of the image. For a fixed radius we used filtering technique to find the patches' location with high precision. The goodness of fit was measured by the Gauss curvature of the convolution surface at the local maximum. The radius was estimated by looking for the maximum of the curvature. The paper discusses the mathematical background, and, through an example, some fine points of the final algorithm. It concludes with ideas for further research.

Alkalmazott Matematikai Lapok **23** (2006), 99–110.

A MEG NEM ÚJULÓ ERŐFORRÁSOK EGY DINAMIKUS LEONTIEF-MODELLJE

FLORISKA ADÉL ÉS DOBOS IMRE

Gödöllő – Budapest

A dolgozatban a sokszektoros dinamikus termelési Leontief-modell egy általánosítását mutatjuk be. A standard nyílt dinamikus Leontief-modellt kibővítjük a meg nem újuló erőforrások mérlegegyenletével. Megvizsgáljuk az így kapott diszkrét lineáris rendszer irányíthatóságát, a végső felhasználást, a fogyasztást tekintve irányítási paraméternek. Nyilvánvalóan a meg nem újuló erőforrások készleteit (pl. vasérc, szén, kőolaj) az elsődleges erőforrások felhasználása csökkenti. Egyenletesen növekvő fogyasztást és termelést feltételezve megvizsgáljuk, hogy a kezdeti erőforrás-készletek mennyi ideig fedezik a termeléshez szükséges ráfordításokat, valamint a rendszer élettartama, működőképessége hogyan függ az egyenletes növekedés ütemétől és a fogyasztástól. Vizsgálatainkhoz felhasználtuk a klasszikus irányításelméleti tételeket valamint a lineáris algebra sajátérték-feladataira vonatkozó eredményeket.

1. Bevezetés

A véges, kimerülő erőforrás-készletek problémájának megoldása nemcsak környezetvédelmi, hanem gazdasági szempontból is igen fontos feladat. A dolgozatban egy környezeti hatásvizsgálatot mutatunk be, a gazdasági tevékenységek és ezen tevékenységekhez szükséges meg nem újuló nyersanyagok közötti kapcsolatot tekintve tanulmányunk alapjának (Turner, Pearce, Bateman [3], Tietenberg [7]).

A közismert egyszerűsítő feltevések ellenére az input-output elemzés igen hasznos eszköz a gazdasági fejlődés stratégiai irányainak vizsgálatára, valamint az egymással összefüggő gazdasági tevékenységek és a természet mennyiségi kölcsönhatásainak tapasztalati tanulmányozására. Mindezeket figyelembe véve, a szakirodalomban jól ismert sokszektoros dinamikus Leontief-modellt kibővítjük a meg nem újuló erőforrások mérlegegyenletével (Leontief [4]). A kapott modellel kapcsolatosan a dolgozatban a következő kérdésekre keressük a választ.

Az így konstruált dinamikus termelési-ökológiai modell irányítható-e, ha a fogyasztást tekintjük irányítási paraméternek? Azaz tudjuk-e befolyásolni a gazdasági folyamatok nyersanyag-felhasználását azáltal, hogy irányítjuk a végső felhasználást a gazdaságban? Ismert, hogy növekvő fogyasztási igényeket csak növekvő termeléssel lehet kielégíteni, aminek következtében viszont a meg nem újuló erőforrás-készletek csökkenni fognak. Továbbá megvizsgáljuk, hogy a kezdeti erőforrás-készletek mennyi ideig fedezik a termeléshez szükséges ráfordításokat, amennyiben mind a termelés, mind a fogyasztás egyenletes ütemben növekszik, valamint a rendszer élettartama, működőképessége hogyan függ az egyenletes növekedés ütemétől.

A dolgozat a következő részekre tagolódik. A bevezetést követő első részben ismertetjük modellünket, majd a további részekben vizsgáljuk a rendszer irányíthatóságát, az egyensúlyi növekedési pálya létezésének feltételeit, a meg nem újuló erőforrás-készletek időbeni alakulását, valamint a készletek kimerülésének időpontját egyenletesen növekvő termelést és fogyasztást feltételezve. Végül modellünk működését egy egyszerű numerikus példán mutatjuk be. Az utolsó részben pedig a dolgozat eredményeit foglaljuk össze.

2. A kibővített dinamikus Leontief-model

A modell leírásakor a hagyományos sokszektoros dinamikus Leontief-modell egyenletéből indulunk ki. Ezt a modellt kibővítjük a meg nem újuló erőforrások mérlegegyenletével. Az így kapott modell eszköz arra, hogy vizsgáljuk a gazdasági folyamatok és a környezet kölcsönhatását. A Leontief-modell alapvető feltevése, hogy az egyes gazdasági szektorok teljes termelésének fedeznie kell a végső fogyasztást, a termelőfogyasztást, azaz a termeléshez szükséges folyó ráfordításokat és a növekedéshez szükséges beruházásokat.

A modellünkkel kapcsolatosan a következő feltevésekkel élünk. A gazdasági ágazatok száma n, és minden egyes ágazat egyetlen terméket állít elő. A termeléshez minden egyes ágazat legfeljebb m-féle elsődleges erőforrást használhat fel (feltesszük, hogy $m < n$).

Ez a gazdasági-ökológiai modell, a gazdaság működését leíró, a termékekre vonatkozó input-output mérlegegyenlettel, valamint az erőforrás-készletek alakulását leíró mérlegegyenlettel jellemezhető. A *termékek mérlegegyenlete* azt fejezi ki, hogy a gazdasági ágazatok teljes kibocsátásának fedeznie kell a termeléshez szükséges ráfordításokat, beruházásokat és a fogyasztást.

$$x_t = Ax_t + B(x_{t+1} - x_t) + c_t. \tag{1}$$

Az *erőforrások mérlegegyenlete* azt fejezi ki, hogy egy adott évben a kitermeléssel csökkentett meg nem újuló erőforrás-készletek jelentik a következő évben rendelkezésre álló készletmennyiséget.

$$R_{t+1} = R_t - Dx_t, \tag{2}$$

ahol

- x_t a termelő ágazatok bruttó kibocsátásának n-dimenziós vektora,
- c_t a végső fogyasztás n-dimenziós vektora,
- A a folyó ráfordítások együtthatóinak $n \times n$-es mátrixa,
- B a tőkeráfordítási együtthatók $n \times n$-es mátrixa,
- D az erőforrás-ráfordítások együtthatóinak $n \times n$-es mátrixa, amely megmutatja, hogy egységnyi termék előállításához mennyi meg nem újuló erőforrás szükséges az egyes fajtákból,
- R_t a meg nem újuló erőforrás-készletek n-dimenziós vektora.

Feltevések. A dolgozatban feltesszük, hogy A, B és D nemnegatív mátrixok, B nemszinguláris mátrix és c_t nemnegatív vektor. A fenti két egyenlet explicit vektoriális alakja

$$\begin{bmatrix} x_{t+1} \\ R_{t+1} \end{bmatrix} = \begin{bmatrix} I + B^{-1}(I - A) & 0 \\ -D & I \end{bmatrix} \begin{bmatrix} x_t \\ R_t \end{bmatrix} - \begin{bmatrix} B^{-1} \\ 0 \end{bmatrix} c_t$$

ahol I az $n \times n$-es egységmátrixot jelöli.

A további vizsgálatok céljából célszerű átírni a fenti diszkrét idejű lineáris rendszert a jólismert mátrix-vektor alakban (Kalman [2])

$$q_{t+1} = A_q q_t + B_q u_t, \tag{3}$$

ahol $A_q = \begin{bmatrix} I + B^{-1}(I - A) & 0 \\ -D & I \end{bmatrix}$ a rendszer $(m+n) \times (m+n)$-es mátrixa, $B_q = -\begin{bmatrix} B^{-1} \\ 0 \end{bmatrix}$ az irányítás $(m+n) \times n$-es hatásmátrixa, $q_t = \begin{bmatrix} x_t \\ R_t \end{bmatrix}$ a rendszer $m+n$-dimenziós állapotvektora, $u_t = c_t$ az irányítás n-dimenziós vektora. A következő részben megvizsgáljuk a (3)-as rendszer irányíthatóságát.

3. A modell irányíthatósága

Egy rendszer *irányíthatóságán* (Elaydi [1]) azt a tulajdonságát értjük, hogy egy megadott időintervallumon belül, a rendszer egy tetszőleges kezdeti állapotból átvihető egy tetszőleges végső állapotba.

A Kalman-féle rangfeltételt alkalmazva (Kalman [2], Elaydi [1]) kapjuk, hogy a (3)-as rendszer akkor és csakis akkor irányítható, ha

$$\operatorname{rang} \begin{bmatrix} B_q & A_q B_q & \dots & A_q^{m+n-1} B_q \end{bmatrix} = m + n.$$

1. LEMMA. *A (3)-as rendszer akkor és csakis akkor irányítható, ha* $\operatorname{rang} D = m$.

Bizonyítás. A mátrixok rangjára vonatkozó tulajdonságokat felhasználva egyszerűen belátható, hogy a (3)-as rendszer irányíthatósága ekvivalens az alábbi rendszer irányíthatóságával:

$$q_{t+1} = (A_q - I)\, q_t + B_q u_t.$$

Jelölje az $n \times nm$-es irányítási mátrixot W, ahol

$$(4) \qquad W := \left[B_q \quad (A_q - I)B_q \quad \dots \quad (A_q - I)^{m+n-1} B_q \right] =$$

$$= \left[\begin{bmatrix} B^{-1} \\ 0 \end{bmatrix}, \begin{bmatrix} B^{-1}(I-A)B^{-1} \\ -DB^{-1} \end{bmatrix}, \begin{bmatrix} \left(B^{-1}(I-A)\right)^2 B^{-1} \\ -DB^{-1}(I-A)B^{-1} \end{bmatrix}, \dots \right.$$

$$\left. \dots, \begin{bmatrix} \left(B^{-1}(I-A)\right)^{m+n-1} B^{-1} \\ -D\left(B^{-1}(I-A)\right)^{m+n-2} B^{-1} \end{bmatrix} \right].$$

Az irányíthatóság feltételének teljesülése, azaz a W mátrix rangja $m+n$, azt jelenti, hogy a W mátrix lineárisan független sorainak a száma pontosan $m+n$-nel egyenlő. Megmutatjuk, hogy a W mátrixnak pontosan akkor van $m+n$ számú lineárisan független sora, ha $\operatorname{rang} D = m$.

Legyen y_1 egy n- és y_2 pedig egy m-dimenziós konstans vektor. A W mátrixnak pontosan akkor van $m+n$ lineárisan független sora, ha minden $[y_1, y_2]\, W = 0$ teljesülése esetén fennáll az $y_1 = 0$ és $y_2 = 0$ egyenlőség.

A (4) jelöléseit alkalmazva az $[y_1, y_2]\, W = 0$ egyenletre kapjuk, hogy

$$(5) \qquad [y_1, y_2] \left[\begin{bmatrix} B^{-1} \\ 0 \end{bmatrix}, \begin{bmatrix} B^{-1}(I-A)B^{-1} \\ -DB^{-1} \end{bmatrix}, \begin{bmatrix} \left(B^{-1}(I-A)\right)^2 B^{-1} \\ -DB^{-1}(I-A)B^{-1} \end{bmatrix}, \dots \right.$$

$$\left. \dots, \begin{bmatrix} \left(B^{-1}(I-A)\right)^{m+n-1} B^{-1} \\ -D\left(B^{-1}(I-A)\right)^{m+n-2} B^{-1} \end{bmatrix} \right] = 0.$$

Mivel $\operatorname{rang} B^{-1} = n$, az $[y_1, y_2]\, W = 0$ egyenlet csak $y_1 = 0$ esetén oldható meg. Behelyettesítve az (5) egyenletbe az $y_1 = 0$ kifejezést, azt kapjuk, hogy

$$[0, y_2] \left[\begin{bmatrix} B^{-1} \\ 0 \end{bmatrix}, \begin{bmatrix} B^{-1}(I-A)B^{-1} \\ -DB^{-1} \end{bmatrix}, \begin{bmatrix} \left(B^{-1}(I-A)\right)^2 B^{-1} \\ -DB^{-1}(I-A)B^{-1} \end{bmatrix}, \dots \right.$$

$$\left. \begin{bmatrix} \left(B^{-1}(I-A)\right)^{m+n-1} B^{-1} \\ -D\left(B^{-1}(I-A)\right)^{m+n-2} B^{-1} \end{bmatrix} \right] = 0.$$

A szorzásokat elvégezve

$$\left[0, -y_2 DB^{-1}, -y_2 DB^{-1}(I-A)B^{-1}, \dots, -y_2 D\left(B^{-1}(I-A)\right)^{m+n-2} B^{-1}\right] = 0.$$

Az egyenlet bal oldalán lévő kifejezésből kiemelve az y_2D-t kapjuk, hogy

$$(6) \qquad y_2D\left[0, -B^{-1}, -B^{-1}(I-A)B^{-1}, \ldots, -\left(B^{-1}(I-A)\right)^{m+n-2}B^{-1}\right] = 0.$$

Amennyiben feltesszük, hogy rang $D = m$, a fenti egyenlőség pontosan akkor teljesül, ha $y_2 = 0$. Ezzel beláttuk, hogy rang $W = m + n$.

Most feltételezzük, hogy rang $W = m + n$, majd belátjuk, hogy rang $D = m$. Ha rang $D < m$, akkor létezik olyan $y_2 \neq 0$, m-dimenziós konstans vektor, amelyre fennáll az $y_2D = 0$ egyenlőség. Ez azt jelenti, hogy az $y_2 \neq 0$-ra (6) egyenlőség is teljesül, tehát rang $W < m + n$, ami viszont ellentmond feltevésünknek.

Megjegyzés. A (3) lineáris rendszer matematikai értelemben vett irányíthatóságát tanulmányoztuk, de valós gazdasági rendszer esetén vizsgálni kell az állapotváltozók nemnegativitását is. A nemnegativitási feltételek teljesülését a következő részben egy speciális esetben vizsgáljuk, feltételezve a termelés és a fogyasztás azonos ütemű egyenletes növekedését.

Megvizsgáljuk, hogy adott irányítás, azaz adott fogyasztás mellett hogyan alakulnak az (1) rendszer pályái, valamint az egyenletes növekedésű pálya hatását a meg nem újuló erőforrások készletére.

4. Az egyensúlyi arányos pálya vizsgálata

A továbbiakban keressük az (1) rendszer adott α ($\alpha \geq 0$) növekedési ütemhez tartozó egyensúlyi megoldását, feltételezve, hogy a termelés is és a fogyasztás is azonos α ütemben növekszik. Ekkor a rendszer megoldása $x_t = (1+\alpha)^t x_0$ és $c_t = (1+\alpha)^t c_0$ alakú, ahol $\alpha \geq 0$ és x_0, c_0 a kezdeti termelési, illetve fogyasztási szerkezetet jelöli. Az előbbi kifejezéseket x_t-re és c_t-re behelyettesítve az (1)-es egyenletbe azt kapjuk, hogy a kezdeti termelési és fogyasztási szerkezetek között az alábbi kapcsolatnak kell fennállnia:

$$(7) \qquad (I - A - \alpha B)x_0 = c_0.$$

A (7) egyenletből következik, hogy az egyenletes növekedésű pályához tartozó kezdeti termelési szerkezet függ mind a növekedés ütemétől, mind pedig a kezdeti fogyasztástól. A továbbiakban feltételeket keresünk a nemnegatív x_0 kezdeti kibocsátás létezésére vonatkozóan.

A további érveléseinkben a következő jelöléseket használjuk: Legyen M egy tetszőleges nemnegatív $n \times n$-es mátrix. $\lambda_1(M)$ jelölje az M mátrix Frobenius-gyökét, amely a mátrix legnagyobb abszolút értékű nemnegatív valós sajátértékét jelenti.

2. LEMMA. *A (7) egyenletnek megfelelő x_0 kezdeti kibocsátási szerkezet létezik és nemnegatív, ha $\alpha \in [0, \alpha_0)$, ahol α_0 az ún. határnövekedési ráta, amelyre* $\lambda_1(A + \alpha_0 B) = 1$.

Bizonyítás. Mivel az $A + \alpha B$ mátrix nemnegatív, a Perron–Frobenius-tételből (Morishima [5]) következik, hogy az $A + \alpha B$ mátrixnak van Frobenius-gyöke. Amennyiben ez utóbbi kisebb mint egy, akkor az $I - A - \alpha B$ mátrixnak létezik nemnegatív inverze.

Produktív termelési rendszereknél, $\alpha = 0$ esetben a $\lambda_1(A + \alpha B) < 1$ egyenlőtlenség nyilvánvalóan teljesül. $\alpha > 0$-ra a Perron–Frobenius-tételek alkalmazásával könnyen belátható, hogy $\lambda_1(A + \alpha B)$ monoton növekedő, folytonos függvénye α-nak, és létezik α^* $\left(\alpha^* = \frac{1}{\lambda_1(B)}\right)$ úgy, hogy $\lambda_1(A + \alpha^* B) \geq 1$. Ezért α értékét növelve $\lambda_1(A + \alpha B)$ is nő, a folytonosság miatt viszont léteznie kell egy pozitív α_0-nak, amelyre $\lambda_1(A + \alpha_0 B) = 1$. Ezzel beláttuk, hogy minden α-ra, $\alpha \in [0, \alpha_0)$ teljesül a $\lambda_1(A + \alpha B) < 1$ egyenlőtlenség, ami viszont azt jelenti, hogy $(I - A - \alpha B)^{-1}$ létezik és nemnegatív. Azaz minden adott α növekedési ütemhez $\alpha \in [0, \alpha_0)$, és adott c_0 fogyasztáshoz létezik egy nemnegatív $x_0(\alpha)$ kibocsátási szerkezet, amelyre fennáll

$$x_0(\alpha) = (I - A - \alpha B)^{-1} c_0. \tag{8}$$

1. ÁLLÍTÁS. *Az α_0 határnövekedési ráta az alábbi sajátérték-feladat megoldása:*

$$B^{-1}(I - A)v = \alpha_0 v$$

valamely nemnulla n-dimenziós v vektorra.

Bizonyítás. A 2. Lemma alapján az α_0 *határnövekedési rátára* fennáll a

$$\lambda_1(A + \alpha_0 B) = 1$$

egyenlőség. Ez azt jelenti, hogy létezik egy olyan nemnulla v vektor, amelyre $(A + \alpha_0 B)v = v$. Ekvivalens átalakításokkal ez utóbbi egyenlőséget az

$$(I - A)^{-1} B v = \frac{1}{\alpha_0} v$$

alakra hozzuk, ami viszont ekvivalens a bizonyítandó sajátérték-feladattal.

2. ÁLLÍTÁS. *Legyen C és D nemnegatív $n \times n$-es mártix úgy, hogy $C \leq D$. Ekkor $C^2 \leq D^2$.*

Bizonyítás. A feltétel azt jelenti, hogy $0 \leq c_{ij} \leq d_{ij}$ minden $i, j \in \overline{1, n}$ esetén. Ez utóbbi feltételt, valamint a mátrixok szorzásának szabályát felhasználva kapjuk, hogy $C^2 = \left[\sum_{k=1}^{n} c_{ik} c_{kj}\right] \leq \left[\sum_{k=1}^{n} d_{ik} d_{kj}\right] = D^2$.

Megjegyzés. Könnyen belátható, hogy az 1. Állítás a C és D mátrixok tetszőleges hatványára is teljesül.

3. LEMMA. *Feltételezzük, hogy létezik az $x_0(\alpha)$ nemnegatív kibocsátási szerkezet. Ekkor*

(i) $x_0(\alpha)$ *α-nak monoton növekedő függvénye, ahol* $\alpha \in [0, \alpha_0)$,

(ii) $x_0(\alpha)$ *nem feltétlenül korlátos függvény a* $[0, \alpha_0)$ *intervallumon.*

Bizonyítás. (i) Legyen $\alpha_1, \alpha_2 \in [0, \alpha_0)$, ahol $\alpha_1 \leq \alpha_2$. Mivel $\alpha_1 \in [0, \alpha_0)$, ezért a 2. Lemma alapján $\lambda_1 (A + \alpha_1 B) < 1$, azaz $(I - A - \alpha_1 B)^{-1}$ létezik és nemnegatív, a Neumann-hatványsora pedig konvegens: $(I - A - \alpha_1 B)^{-1} = \sum_{i=0}^{\infty} (A + \alpha_1 B)^i$. Hasonlóképpen kapjuk, hogy $(I - A - \alpha_2 B)^{-1} = \sum_{i=0}^{\infty} (A + \alpha_2 B)^i$. Az 1. Állítást és a Megjegyzést felhasználva $C = A + \alpha_1 B$ és $D = A + \alpha_2 B$ mátrixokra kapjuk, hogy $\sum_{i=0}^{\infty} (A + \alpha_1 B)^i \leq \sum_{i=0}^{\infty} (A + \alpha_2 B)^i$, azaz $(I - A - \alpha_1 B)^{-1} \leq (I - A - \alpha_2 B)^{-1}$. Beszorozva az egyenlőtlenséget a nemnegatív c_0 vektorral kapjuk, hogy

$$x_0(\alpha_1) \leq x_0(\alpha_2),$$

ami éppen bizonyítandó volt.

(ii) Tegyük fel, hogy $\alpha \to \alpha_0$. A $\lambda_1(A + \alpha B)$ függvény folytonosságát felhasználva $\lambda_1(A + \alpha B) \to 1$. Egyszerű számolással adódik, hogy

$$\lambda_1(E - A - \alpha B)^{-1} = \frac{1}{1 - \lambda_1(A + \alpha B)}.$$

Ezért minden $\alpha \to \alpha_0$ esetén, amennyiben a $\lambda_1(I - A - \alpha B)^{-1}$ sajátértékhez tartozó sajátvektor nem merőleges a nemnegatív c_0 fogyasztásvektorra,

$$\lambda_1(I - A - \alpha B)^{-1} \to \infty.$$

5. Az erőforrások vizsgálata

A gazdaság működéséből következik, hogy a meg nem újuló erőforrások korlátos jellegükből adódóan korlátozzák a gazdaság növekedését. A továbbiakban ezt a kapcsolatot vizsgáljuk, azaz hogy miként alakulnak az egyenletes növekedésű pályához tartozó erőforrás-készletek.

Először meghatározzuk a (2) differenciaegyenlet megoldását, feltételezve, hogy a termelés is és a fogyasztás is azonos, nemnegatív α ütemben nő. Helyettesítsük be az (1) rendszer egyensúlyi megoldását $x_t = (1 + \alpha)^t x_0$ a (2) egyenletbe, majd az erőforrás-készletekre egyszerű számítással a következő formulát kapjuk:

$$R_t(\alpha, c_0) = R_0 - \frac{(1 + \alpha)^t - 1}{\alpha} D x_0(\alpha). \tag{9}$$

4. LEMMA. *Az $R_t(\alpha, c_0)$-val jelölt erőforrás-készletek*
(i) *α-nak monoton csökkenő függvényei minden $\alpha \in [0, \alpha_0)$ növekedési ütem esetén, és*
(ii) *c_0-nak monoton csökkenő függvényei minden nemnegatív c_0 fogyasztásvektor esetén.*

E feltételek a (9) egyenlőség analitikus alakjából közvetlenül következnek.

Megjegyzés. A fenti lemmák közgazdaságtani értelemben azt jelentik, hogy amennyiben a növekedés ütemét csökkentjük, vagy csökkentjük a kezdeti fogyasztási szintet, az erőforrás-készletek tovább tartanak.

6. Az erőforrás-készletek kimerülésének becslése

Az alábbi kifejtésben becsüljük, hogy kezdeti erőforrás-készletek mennyi ideig fedezik a termeléshez szükséges ráfordításokat, amennyiben mind a termelés, mind a fogyasztás egyenletes ütemben növekszik. Keressük azt a legnagyobb t időtartamot a (9) egyenletet használva, amelyre az erőforrás-készletek még nemnegatívak ($R_t \geq 0$). Jelöljük T-vel a keresett időtartamot, amely nem feltétlenül egész szám. Ekkor teljesülnie kell az alábbi egyenlőségnek:

$$\frac{(1+\alpha)^T - 1}{\alpha} = \min_i \left(\frac{(R_0)_i}{\left(Dx_0(\alpha)\right)_i} \right),$$

ahol $(\,\cdot\,)_i$ jelöli a megfelelő vektor i-edik komponensét. T-t kifejezve kapjuk, hogy

$$T(\alpha) = \frac{\ln\left(\alpha \min_i \frac{(R_0)_i}{\left(Dx_0(\alpha)\right)_i} + 1\right)}{\ln(1+\alpha)}, \qquad \alpha \in (0, \alpha_0). \tag{10}$$

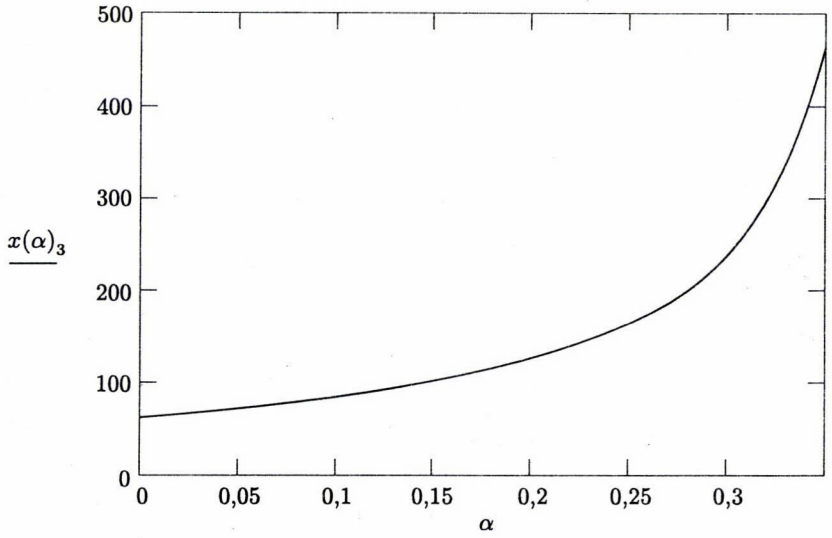
Megjegyzés. Egyszerű matematikai érveléssel belátható, hogy $T(\alpha)$ α-nak monoton csökkenő függvénye, ami pontosan azt jelenti, hogy csökkenő növekedési ütem esetén a készletek tovább tartanak.

7. Egy numerikus példa

A dolgozatban leírt modell működését egy egyszerű számpéldán mutatjuk be. Feltételezzük, hogy három gazdasági ágazatunk van, és minden ágazat működéséhez kétféle erőforrást használ. A folyó ráfordítások, a tőkeráfordítások valamint az erőforrás-ráfordítások együtthatóinak mátrixai legyenek a következők:

$$A = \begin{bmatrix} 0.1 & 0.3 & 0.2 \\ 0.5 & 0.2 & 0.4 \\ 0.2 & 0.4 & 0.5 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.01 & 0.03 & 0.02 \\ 0.05 & 0.02 & 0.04 \\ 0.07 & 0.06 & 0.01 \end{bmatrix} \quad \text{és} \quad D = \begin{bmatrix} 0.6 & 0.2 & 0.1 \\ 0.4 & 0.5 & 0.3 \end{bmatrix}.$$

A 2. Lemma eredményeit felhasználva, fenti paraméterekkel számolva, a határnövekedési ráta értéke 0.404 ($\alpha_0 = 0.404$). Ez azt jelenti, hogy a gazdaság növekedési ütemének csak ennél kisebb értéket választhatunk.



1. ábra. A harmadik ágazat kezdeti termelési szerkezetének alakulása a növekedési ütem függvényében

Legyen az egyenletes növekedés üteme 0.03 (azaz 3%). A kezdeti fogyasztás és erőforrás-készletek vektorainak válasszuk a következő paraméterértékeket:

$$c_0 = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \quad \text{és} \quad R_0 = \begin{bmatrix} 10{,}000 \\ 15{,}000 \end{bmatrix}.$$

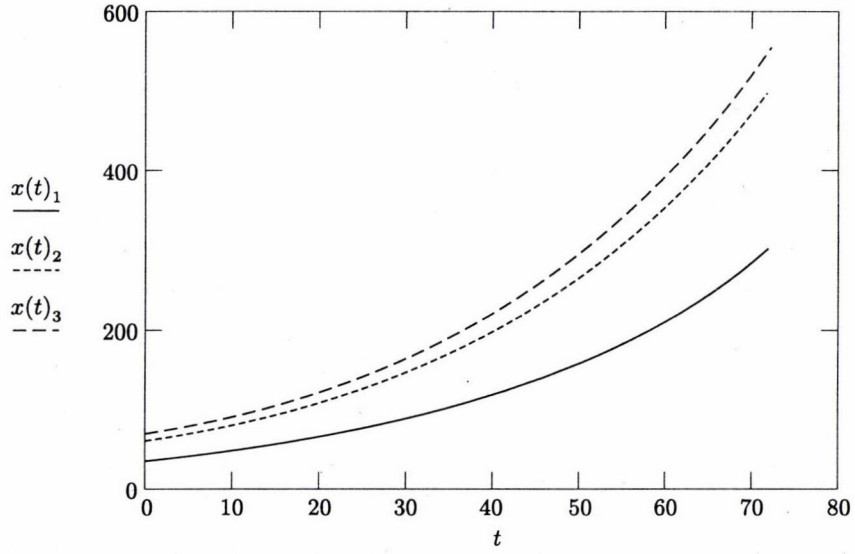
A (8) egyenletet alapján a gazdaság kezdeti termelési szerkezete:

$$x_0(0.03) = \begin{bmatrix} 35.787 \\ 59.479 \\ 66.302 \end{bmatrix}.$$

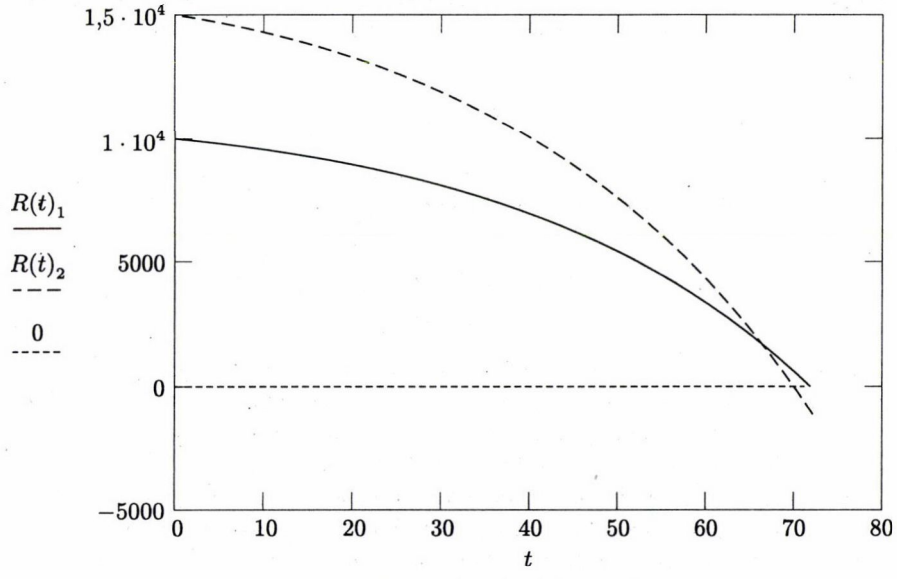
Amennyiben a növekedés ütemét változónak tekintjük, a kezdeti termelési szerkezet ezen növekedési ütem monoton növekvő, nemkorlátos függvénye (lásd 3. Lemma). Ez a tulajdonság a harmadik ágazatra az 1. ábrán látható.

A 2. és 3. ábrákon mutatjuk be a gazdaság egyensúlyi arányos pályájának, valamint az erőforrás-készleteknek időbeni alakulását az egyes ágazatok esetében, feltételezve, hogy a növekedési ütem értéke 0.03. Az ágazatok termelésének időbeni alakulását vizsgálva láthatjuk, hogy a termékkibocsátás monoton növekvő és nemkorlátos, míg az erőforrás-készletek kimerülése az időnek monoton csökkenő függvénye. A felvett adatokkal elvégezve a számolásokat a (9) azonosságban,

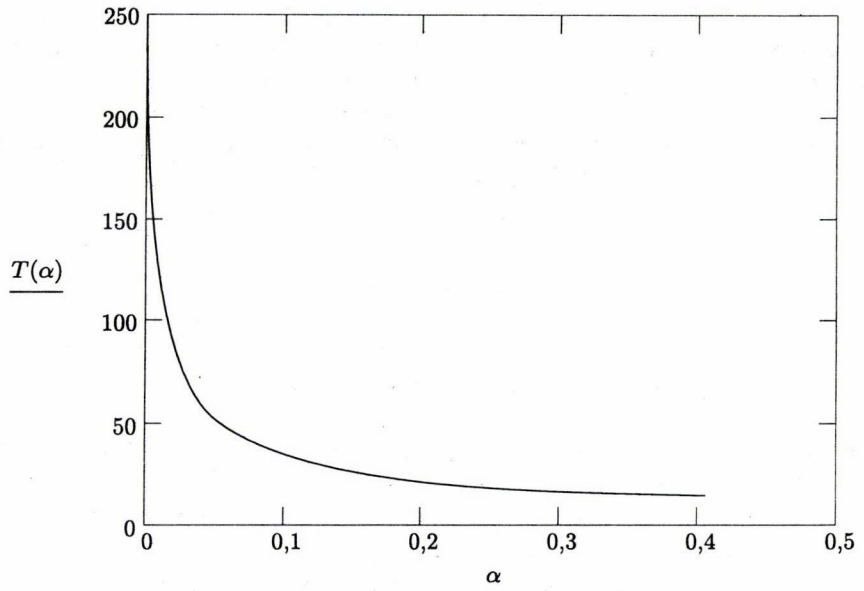
azt kapjuk, hogy az első fajta erőforrás-készletek 72 évig, viszont a második fajta erőforrás-készletek csak 70 évig elegendőek, így a gazdaság 70 évig lesz működőképes.



2. ábra. Az egyensúlyi arányos pálya időbeni alakulása az egyes ágazatok esetében



3. ábra. Az erőforrás-készletek kimerülése



4. ábra. Az erőforrás-készletek kimerülése a növekedési ütem függvényében

A 4. ábrán bemutatjuk, hogy a növekedési ütem értékét a megengedett határok között változtatva hogyan változik az erőforrás-készletek kimerülésének időpontja. Látható, hogy a növekedési ütem értékének növelésével ez az időpont egyre kisebb. A (10) azonosságot használva a példa adataira azt kapjuk, hogy az erőforrás-készletek mintegy 70 évig ($T(0.03) = 70.48$) biztosítják a gazdaság működését.

8. Összefoglalás és további kutatás

A dolgozatban egy általánosított Leontief-modellt vizsgáltunk. Az alapmodellt az erőforrások felhasználásával bővítettük ki. Bebizonyítottuk, hogy a bővített modell irányítható, ha az erőforrások felhasználási mátrixának a rangja megegyezik a sorvektorok számával. A dolgozat második részében az egyensúlyi arányos pálya hatását vizsgáltuk az erőforrások felhasználására. Bemutattuk, hogy a meg nem újuló erőkorrások készlete tovább tart, ha a növekedési rátát és/vagy a fogyasztási szintet csökkentjük. A kutatás egy következő fázisában azt lehetne megvizsgálni, hogy az arányos egyensúlyi pálya hogyan befolyásolja a megújuló erőforrások (pl. víz, levegő) minőségét. Hogyan alakul az egyensúlyi arányos pálya, ha a kormányzat a gazdaság egészére kibocsátási határértékeket állapít meg? További vizsgálatok tárgya lehet az újrafelhasználás bevonása az alapmodellbe. Ekkor az újrafelhasználással csökkenteni lehet a természeti erőforrások felhasználását, ami a fenntartható fejlődés értelmében erőforrásokat takarít meg a következő generációknak.

Az említett lehetséges vizsgálatok figyelmen kívül hagyják az árak hatását az egyensúlyi arányos pályára. Ezen elemzések olyan árrendszer kialakításához nyújthatnak elemzési keretet, amelyekben a termelést környezettudatosan lehetne irányítani. Végül, a vizsgált modell eltekint a technológiai haladástól, vagyis az együtthható márixok időben változatlanok. A technikai haladás hatásának meg nem újuló erőforrásokra gyakorolt hatása lehet egy következő kutatási irány.

Irodalomjegyzék

[1] Elaydi, S. N., *An Introduction to Difference Equations* (Springer, Berlin et al., 1996).

[2] Kalman, R. E., Contributions to the Theory of Optimal Control, *Bol. Socied. Mat. Mexicana* (1960).

[3] Turner, K., Pearce, D. and Bateman, I., *Environmental Economics* (T. J. Press Ltd., Cornwall, 1994).

[4] Leontief, W., *Input-Output Economics* (Oxford University Press, Oxford, 1986).

[5] Morishima, M., *Equilibrium Stability and Growth* (Oxford University Press, Oxford, 1964).

[6] Schoonbeek, L., The Size of the Balanced Growth Rate in the Dynamic Leontief Model, *Economic Systems Research* **2** No. 4 (1990) 345–349.

[7] Tietenberg, T., *Environmental and Natural Resource Economics* (Addison Wesley Longman, Inc., 2000).

(Beérkezett: 2004. szeptember 16.)

FLORISKA ADÉL
MATEMATIKAI ÉS SZÁMÍTÁSTUDOMÁNYI INTÉZET
SZENT ISTVÁN EGYETEM
2100 GÖDÖLLŐ
PÁTER KÁROLY U. 1.
afloriska@mszi.gau.hu

DOBOS IMRE
VÁLLALATGAZDASÁGTAN INTÉZET
BUDAPESTI CORVINUS EGYETEM
1053 BUDAPEST
VERES PÁLNÉ U. 36.
imre.dobos@uni-corvinus.hu

NON-RENEWABLE RESOURCES IN AN OPEN DYNAMIC LEONTIEF MODEL

ADÉL FLORISKA AND IMRE DOBOS

In this paper we study a generalisation of the dynamic Leontief input-output model. We extend the standard dynamic Leontief model with the balance equation of non-renewable resources. Obviously the non-renewable stocks will decrease exploiting primary resources. In this study we examine the controllability of this extended model taking the consumption as control parameter. We suppose a balanced growth path both for consumption and production, and we examine how long these scare resources will cover the inputs of production and how the lifetime of the system depends on the balanced growth rate and on the consumption, as well. To these investigations we use the classic results of the control theory and the eigenvalue problems of the linear algebra.

Alkalmazott Matematikai Lapok **23** (2006), 111–120.

KORSTRUKTURÁLT POPULÁCIÓDINAMIKAI MODELL STABILITÁSA

FARKAS JÓZSEF ZOLTÁN

Budapest

1974-ben Gurtin és McCamy bevezette a Lotka–McKendrick korstrukturált modell egy nemlineáris változatát. Az azóta eltelt 30 évben ez a PDE modell és későbbi általánosításai a populációdinamika egyik legtöbbet kutatott területe lett. Gurtin és McCamy dolgozatukban levezették a modell stacionárius megoldásához tartozó karakterisztikus egyenletet, de stabilitási eredményeket egészen speciális esetektől eltekintve nem tudtak bizonyítani. Nemrég Farkas Miklós levezette ezt a karakterisztikus egyenletet teljesen más formában, melynek segítségével stabilitási eredményeket sikerült igazolnunk általános feltételek mellett. Ebben a dolgozatban megmutatjuk a két egyenlet ekvivalenciáját, majd megadjuk stabilitási eredményeinket a legáltalánosabb formában.

1. Bevezetés

1926-ban A. G. McKendrick új fejezetét nyitotta meg a matematikai populációdinamikának. [9] dolgozatában egy új, ún. korstrukturált modellt vezetett be elsősorban biológiai, demográfiai problémák modellezésére, mely modell, illetve későbbi nehéz, matematikai szempontból is igényes problémákat felvető általánosításai a populációdinamika egyik legtöbbet kutatott területe lett. A teljesség mindenfajta igénye nélkül néhány általunk jól használhatónak tartott monográfia a témakörből, amelyekben az olvasó további hasznos referenciákat találhat: [1], [6], [8], [10], [11].

A McKendrick-modell lényegi újítása, hogy korstruktúrával jellemzi a populációt, mely a leginkább releváns meghatározója az ún. vitális rátáknak, melyekkel jellemezhetők a populáció egyedei. A modell – mely egy parciális differenciálegyenlet a megfelelő peremfeltételekkel – egy fajból álló populáció dinamikáját adja meg, melyben nincs migráció. Természetesen több fajból álló biológiai rendszerek is modellezhetők a megfelelő számú egyenletből álló modellel, ld. pl. [8], [5].

Jelöljük $p(a,t)$-val a t időpillanatban az a-korú egyedek sűrűségét, azaz a populáció összlétszáma a t időpillanatban a

$$P(t) = \int_0^\infty p(a,t)\,da = \int_0^m p(a,t)\,da$$

mennyiség, ahol m jelöli a maximális életkort. Ilyen véges m biológiai okok miatt nyilván mindig létezik, és sok esetben matematikai szempontból könnyebb véges tartójú függvényekkel dolgozni. Megjegyezzük, hogy a két modell (véges, illetve végtelen tartójú függvények) lényegesen különbözően kezelhető.

Mivel a modellünkben nincs migráció, a $t + dt$ időpontban azok az egyedek alkotják a populációt, akik éltek az $a - dt$ időpontban, illetve le kell vonni azon egyedek mennyiségét, akik meghaltak a dt idő alatt. Az a korú egyedek mortalitását $\mu(a)$-val jelölve kapjuk

$$p(a, t + dt) = p(a - dt, t) - p(a - dt, t)\mu(a - dt)\,dt.$$

Értelemszerűen az egyedek az idő előrehaladásával párhuzamosan öregednek, tehát $da = dt$, az egyenletet megfelelően rendezve a következőt kapjuk

$$p(a, t + dt) - p(a, t) + p(a, t) - p(a - dt, t) = -p(a - dt)\mu(a - dt)\,dt,$$

amiből dt (ill. da)-val leosztás után kapjuk $da = dt \to 0$ esetén

$$p'_t(a,t) + p'_a(a,t) = -\mu(a)p(a,t), \quad 0 \le a < m < \infty.$$

Az a korúak fertilitását $\beta(a)$-val jelölve az újszülöttek sűrűsége

$$p(0,t) = \int_0^m \beta(a)p(a,t)\,da, \quad t > 0,$$

megadunk továbbá egy kezdeti koreloszlást $p(a,0) =: p_0(a)$.

Ebben a modellben tehát a vitális ráták csak a kortól függnek. Ezt a klasszikus korstrukturált lineáris modellt ma már jól ismertnek tekinthetjük, jó leírást ad pl. [8] első két fejezete.

Az alkalmazások szempontjából lényegesen érdekesebb, ha a vitális ráták, β és μ nem csak a kortól, hanem a populáció összlétszámától $P(t)$, vagy annak valamilyen súlyozott átlagától, vagy explicite a t időtől függnek. Ezt a nemlineáris modellt Gurtin–McCamy vezette be 1974-ben [7]. A modell biológiai újítása igen jelentős: a rendszer dinamikáját meghatározó születési, illetve halálozási függvények összlétszámtól való függésének figyelembe vétele lehetőséget ad olyan már ismert jelenségek, mint például az Alle-effektus vizsgálatára, amikor az egyedek fertilitása az összlétszám egy bizonyos K kritikus értékéig nő, mert nagyobb a „meeting" valószínűsége, a K értéken túl pedig csökken, például a túlzott létszám miatt csökken a szaporodókedv.

A második esetben, amikor tehát a vitális ráták explicite az időtől függnek, lineáris nem-autonóm parciális egyenletet kapunk, melynek aszimptotikáját a [3] dolgozatban kezdtük el vizsgálni. Igen érdekesnek ígérkezik a periodikus vitális rátákkal rendelkező rendszer megoldásai viselkedésének vizsgálata.

Ahogy már említettük, számos dolgozat foglalkozik a modell olyan általánosításaival, amikor a vitális ráták nem a $P(t)$ összlétszámtól, hanem annak valamilyen súlyozott átlagától, $S(t) = \int_0^m \gamma(a)p(a,t)\,da$, illetve véges sok ilyen $S_i(t)$ súlyozott átlagtól függnek. Ez sok esetben biológiai szempontból még reálisabbá teszi a modellt, hiszen pl. a különböző korú egyedek nem egyformán vesznek részt a szaporodásért vívott harcban.

2. A karakterisztikus függvény

A Gurtin–McCamy által bevezetett modell a következő nemlineáris parciális differenciálegyenlet

$$p'_a(a,t) + p'_t(a,t) = -\mu\big(a, P(t)\big)p(a,t)$$

a következő integrál peremfeltétellel:

$$p(0,t) = \int_0^m \beta\big(a, P(t)\big)p(a,t)\,da,$$

illetve a $p_0(a) := p(a,0)$ kezdeti koreloszlással, amely teljesíti a következő kompatibilitási feltételt: $p_0(0) = \int_0^m \beta\big(a, P(0)\big)p_0(a)\,da$. Kiemeljük, hogy most a vitális ráták véges tartójú függvények m maximális életkorral, ez biológiai szempontból reális, hiszen minden egyed véges életkorú. Gurtin és McCamy [7]-ben bizonyították a megoldások létezését és egzisztenciáját a megfelelő feltételek mellett.

A fenti modell egy stacionárius, időtől nem függő $p_*(a)$ megoldása egyensúlyi helyzete a rendszernek. Könnyen belátható, hogy ilyen egyensúlyi helyzet csak olyan (időtől független) P összlétszámnál valósulhat meg, amely kielégíti a következő egyenletet:

$$R(P) = \int_0^m \beta(a,P)\pi(a,P)\,da = 1.$$

Itt $\pi(a,P) = e^{-\int_0^a \mu(s,P)\,ds}$ annak a valószínűsége, hogy egy egyed megéri az a életkort, illetve $R(.)$ az úgynevezett reprodukciós ráta, az egy egyed által produkált utódok várható értéke.

Ha tehát adottak a μ, β vitális ráták, akkor megoldhatjuk a fenti egyenletet P-re, majd ezen P_* megoldásból(megoldásokból) kapjuk a stacionárius megoldás $p_*(a)$ újszülötteinek számát a következő képlet szerint:

$$p_*(0) = \frac{P_*}{\int_0^m \pi\,(a,P_*)\,da},$$

amiből a stacionárius megoldás

$$p_*(a) = p_*(0)\pi(a, P_*)$$

meghatározható. Persze több stacionárius megoldás lehet attól függően, hogy az $R(.)$ függvény hányszor veszi fel az 1 értéket.

[7]-ben a szerzők levezették a stacionárius megoldáshoz tartozó karakterisztikus egyenletet, de stabilitási eredményeket egészen speciális esetektől eltekintve nem tudtak bizonyítani. Az általuk levezetett egyenlet a következő:

$$1 = \int_0^m r(a)e^{-\gamma a}\,da + g_\gamma\left(\frac{\kappa}{B_0} - \int_0^m r(a)f_\gamma(a)\,da\right), \tag{1}$$

ahol $r(a) = \beta_0(a)\pi_0(a)$,

$$\kappa = B_0\int_0^m \beta_0'(a)\pi_0(a)\,da, \quad f_\gamma(a) = \int_0^a e^{-\gamma(a-\alpha)}\lambda_0'(\alpha)d\alpha, \tag{2}$$

továbbá

$$g_\gamma = \frac{B_0\int_0^m e^{-\gamma a}\pi_0(a)\,da}{1 + B_0\int_0^m \pi_0(a)f_\gamma(a)\,da}, \quad B_0 = \int_0^m \beta_0(a)p_0(a)\,da. \tag{3}$$

A mi jelöléseinkkel $\lambda_0(\alpha) = \mu(s, P_*)$, $\pi_0(a) = \pi(a, P_*)$, $\beta_0(a) = \beta(a, P_*)$, $\gamma = \lambda$.

[5]-ben a szerző egy új karakterisztikus egyenletet vezetett le, amely a következő:

$$K(\lambda) = A_{11}(\lambda)A_{22}(\lambda) - A_{12}(\lambda)A_{21}(\lambda) + A_{12}(\lambda) + A_{21}(\lambda) = 1, \tag{4}$$

ahol

$$A_{11}(\lambda) = \int_0^m e^{-\lambda a}e^{-\int_0^a \mu(s,P_*)\,ds}\,da,$$

$$A_{12}(\lambda) = -p_*(0)\int_0^m e^{-\lambda a}e^{-\int_0^a \mu(s,P_*)\,ds}\int_0^a \mu_P'(s, P_*)\,e^{\lambda s}\,ds\,da,$$

$$A_{21}(\lambda) = \int_0^m e^{-\lambda a}\beta(a, P_*)\,e^{-\int_0^a \mu(s,P_*)\,ds}\,da,$$

$$A_{22}(\lambda) = p_*(0)\int_0^m \beta_P'(a, P_*)\,e^{-\int_0^a \mu(s,P_*)\,ds}\,da -$$

$$-p_*(0)\int_0^m \left(e^{-\lambda a}\beta(a, P_*)\,e^{-\int_0^a \mu(s,P_*)\,ds}\int_0^a \mu_P'(s, P_*)\,ds\right)da,$$

itt $P_* = \int_0^m p_*(a)\,da$ a stacionárius megoldás összlétszáma.

Lássuk be, hogy a két egyenlet ekvivalens.

Az (1) egyenletbe behelyettesítve a (2)–(3) kifejezéseket, a mi jelöléseinkre áttérve, illetve a $p_*(0) = \frac{P_*}{\int_0^m \pi(a,P_*)\,da}$ összefüggést használva kapjuk a következő egyenletet:

$$1 = \int_0^m e^{-\lambda a}\beta(a,P_*)\,\pi(a,P_*)\,da +$$

$$+\left(\frac{\frac{P_*}{\int_0^m \pi(a,P_*)\,da}\int_0^m e^{-\lambda a}\pi(a,P_*)\,da}{1+\frac{P_*}{\int_0^m \pi(a,P_*)\,da}\int_0^m \pi(a,P_*)\int_0^a e^{-\lambda(a-s)}\mu'_P(s,P_*)\,ds\,da}\right)\cdot$$

$$\cdot\int_0^m \beta'_P(a,P_*)\,\pi(a,P_*)\,da -$$

$$-\left(\frac{\frac{P_*}{\int_0^m \pi(a,P_*)\,da}\int_0^m e^{-\lambda a}\pi(a,P_*)\,da}{1+\frac{P_*}{\int_0^m \pi(a,P_*)\,da}\int_0^m \pi(a,P_*)\int_0^a e^{-\lambda(a-s)}\mu'_P(s,P_*)\,ds\,da}\right)$$

$$\cdot\int_0^m \beta(a,P_*)\int_0^a e^{-\lambda(a-s)}\mu'_P(s,P_*)\,ds\,da.$$

Ebből az $A_{ij}(\lambda)$ formulákat bevezetve kapjuk

$$1 = \left(\frac{p_*(0)A_{11}(\lambda)}{1+p_*(0)\frac{A_{12}(\lambda)}{-p_*(0)}}\right)\int_0^m \beta'_P(a,P_*)\,\pi(a,P_*)\,da -$$

$$-\frac{p_*(0)A_{11}(\lambda)}{1+P_*(0)\frac{A_{12}(\lambda)}{-p_*(0)}}\int_0^m e^{-\lambda a}\beta(a,P_*)\,\pi(a,P_*)\int_0^a e^{\lambda s}\mu'_P(s,P_*)\,ds\,da.$$

Innen

$$1 = A_{21}(\lambda) + \left(\frac{p_*(0)A_{11}(\lambda)}{1-A_{12}(\lambda)}\right)\frac{A_{22}(\lambda)}{p_*(0)},$$

ebből pedig átrendezéssel kapjuk a (4) egyenletet.

Látszólag tehát ez az újabb (4) karakterisztikus függvény sem egyszerűbb, mint a korábbi (1)-es, mégis sokkal jobban kezelhetőnek bizonyult, és stabilitási/instabilitási eredményeket sikerült igazolnunk igen általános β, μ függvények esetén is.

Vegyük észre, hogy az új karakterisztikus függvény szempontjából szembetűnő különbség, hogy a vitális ráták véges tartójú függvények-e vagy sem. Utóbbi esetben az A_{ij} együtthatók Laplace-transzformáltak, a karakteriszikus függvény pedig polinom véges sok gyökkel, míg az első esetben exponenciális függvény végtelen sok gyökkel.

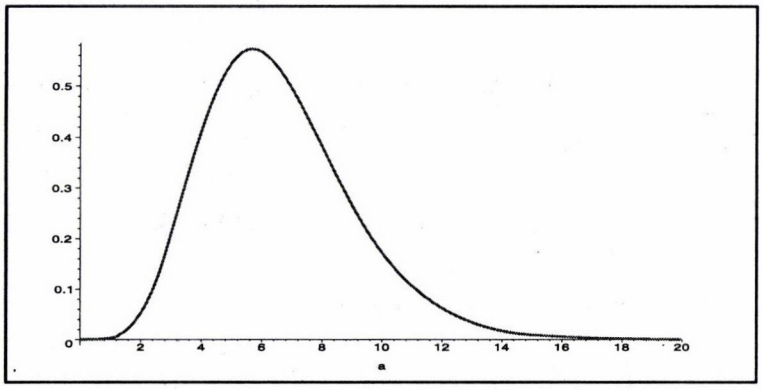
Példa. A következő numerikus példában olyan születési és halálozási rátákat adunk meg, amely választás esetén a (lineáris) rendszernek (végtelen sok) stacionárius megoldása van. A születési és halálozási függvények végtelen tartójúak, így a karakterisztikus függvény polinom lesz.

Legyen

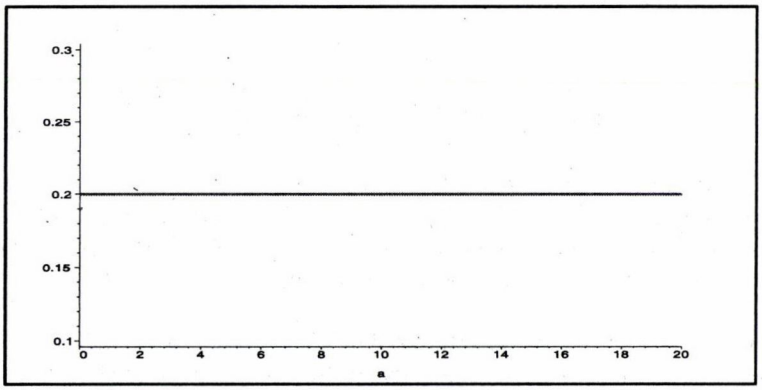
$$\hat{\beta}(a) := a^6(40-a)^2e^{-a}, \quad \mu(a) := 0.2.$$

Ebben az esetben $\pi(a) = e^{-0.2a}$ adódik.

A reprodukciós ráta pedig $\hat{R} = \int_0^\infty \hat{\beta}(a)\pi(a)\,da \sim 235544.91026520347508$. Válasszuk tehát az új β függvényt a következő normálással: $\beta := \frac{\hat{\beta}}{\hat{R}}$.



1. ábra. $\beta(a) := \frac{a^6(40-a)^2e^{-a}}{235544.91026520347508}$



2. ábra. $\mu(a) := 0.2$

A (4)-es karakterisztikus egyenlet lineáris modell esetén a következő egyszerű alakra redukálódik:

$$K(x) = 1 = \int_0^\infty e^{-xa}\beta(a)\pi(a)\,da,$$

aminek láthatóan a 0 mindig gyöke, így a stacionárius megoldás nem lehet aszimptotikusan stabil.

Valóban,

$$K(x) \sim \int_0^\infty 0.42454748815165876777 \cdot 10^{-5} e^{-xa} a^6 (40-a)^2 e^{-a} e^{-0.2a}\,da \sim$$

$$\sim \frac{4.8907870635071090047}{(x+1.2)^7} - \frac{1.7117754722274881516}{(x+1.2)^8} + \frac{0.17117754722274881516}{(x+1.2)^9},$$

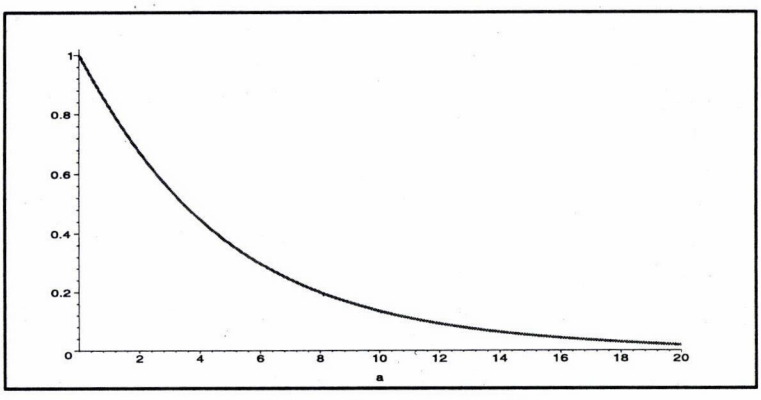
és a $K(x) = 1$ egyenlet megoldásai közelítőleg a következők:

$$x_{1,2} \sim -2.376 \pm 0.545i, \quad x_{3,4} \sim -1.527 \pm 1.226i,$$

$$x_{5,6} \sim -1.025 \pm 0.066i, \quad x_{7,8} \sim -0.470 \pm 0.984i, \quad \boldsymbol{x_0 \sim 0.279 \cdot 10^{-21}}.$$

Lineáris modell esetén, ha létezik stacionárius megoldás, akkor annak tetszőleges (pozitív) konstans többszöröse is ilyen tulajdonságú, így stacionárius megoldások egy I osztályát kapjuk.

$$\widehat{p_*}(a) := \pi(a), \quad I = \left\{c \cdot \widehat{p_*}(a) \mid c \in \mathbf{R}^+\right\}$$



3. ábra. $\widehat{p_*}(a) = \pi(a) = e^{-0.2a}$

3. Stabilitás

Ebben a fejezetben a Gurtin–McCamy-féle nemlineáris modellre vonatkozó stabilitási eredményeinket ismertetjük.

Stabilitás alatt mindig aszimptotikus stabilitást értünk. Stabilitási/instabilitási eredményeink bizonyításainak kulcsa a (4)-es karakterisztikus egyenlet, illetve a [5] dolgozat 2.2-es tétele, miszerint ha a (4)-es karakterisztikus egyenlet minden gyökének valós része negatív, akkor a $p_*(a)$ stacionárius megoldás aszimptotikusan stabil, másrészt ha létezik pozitív valós részű gyöke, akkor a stacionárius megoldás instabil.

Tegyük most fel, hogy a μ halálozási függvény nem függ a populáció összlétszámától, P-től, csak az a kortól. Ez reális feltevés lehet olyan biológiai modellekben, ahol az egyedek pusztulása szempontjából a fajon belüli létszám elhanyagolható, mint például mélytengeri koralloknál. Ebben az esetben az alábbi eredmény igazolható.

1. TÉTEL. *A $\mu(a), \beta(a,P)$ alakú vitális ráták esetén a $p_*(a)$ stacionárius megoldás aszimptotikusan stabil, ha $\beta'_P(.,P_*) < 0$, illetve ha $\beta'_P(.,P_*) > 0$, akkor a stacionárius megoldás instabil.*

A tétel bizonyítása a $\beta(a,P) = b(a)f(P)$ speciális esetben megtalálható a [2] dolgozatban, illetve a [4] dolgozatban a 2-es tétel egy általánosabb modell esetén mond ki hasonló eredményt, amelynek a bizonyítása könnyen átvihető a mi modellünkre.

Vegyük észre, hogy a fenti esetben

$$R'(P) = \int_0^m \beta'_P(a,P)\pi(a)\,da$$

miatt, ha $\beta'_P(.,P_*) < 0$, akkor $R'(P) < 0$, $\pi(a) \geq 0$ miatt, illetve ha $\beta'_P(.,P_*) > 0$ teljesül, akkor $R'(P) > 0$.

Mint látni fogjuk, az ún. reprodukciós ráta $R(.)$ nagymértékben meghatározza a populáció dinamikáját az általános esetben is.

Tegyük tehát most fel, hogy mindkét vitális ráta függ mind az a kortól, mind a P összlétszámtól.

Ebben a teljesen általános esetben a alábbi instabilitási eredmény igazolható:

2. TÉTEL. *Az általános $\mu(a,P)$, $\beta(a,P)$ alakú vitális ráták esetén az $R'(P_*) > 0$ feltételből következik, hogy a P_* összlétszámhoz tartozó $p_*(a)$ stacionárius megoldás instabil.*

Ennek az eredménynek a bizonyítása könnyen adódik az [4] dolgozat 3-as tételének bizonyításából a $\gamma \equiv 1$ választás esetén.

Tekintsük most a $\beta(a,P)$, $\mu(a)$ speciális esetet. Ekkor

$$R'(P) = \int_0^m \beta'_P(a,P)\pi(a)\,da$$

miatt az $R'(P_*) < 0$-ból következik, hogy

$$\int_0^m \beta'_P(a, P_*)\,\pi(a)\,da < 0.$$

Vegyük figyelembe továbbá, hogy a $\pi(.)$ fügvény a 3. ábra szerinti monoton csökkenő pozitív függvény.

Mindezekből következik, hogy a $\beta(., P)$ függvény a populációnak a nagy fertilitású korcsoportjainál a P_* környezetében a P összlétszám növelésével csökken, tehát ezekre a korcsoportokra a $\beta'(., P_*) < 0$ feltétel kell teljesüljön. Ha ez minden a-ra teljesülne, akkor 2-es tétel garantálná a stabilitást.

A modellünk segítségével tehát a fenti egzakt matematikai eredményekből az alábbi következtetéseket tehetjük.

Ha egy populációban a maximális fertilitású korosztályok előtti korosztályok mortalitása kicsi, mint pl. a fejlett világban élő embereké, akkor a populáció nem lesz annyira érzékeny egyes kisebb korcsoportok szaporodási rátájának gyors megváltozására. Ellenkező esetben, tehát ha a maximálisnál jelentősen alacsonyabb fertilitású egyedek mortalitása is jelentős, akkor a populáció egyes kisebb korcsoportjainál beálló megnövekedett szaporodókedv is instabilitást okozhat. azaz a szaporodóképes egyedek mortalitásának növekedése destabilizálja a rendszert.

Ez a viselkedés magyarázhatja olyan populációk létezését, ahol a kevéssé szaporodóképes (fiatal) egyedek határozzák meg a dinamikát, mert az ő mortalitásuk relatíve nagy. Az itteni korcsoportoknál a szaporodókedvben bekövetkező változások a döntőek. Jó például szolgálhatnak erre bizonyos északi-tengeri halfajok, amely populációkban a fiatal egyedek planktonnal, apróbb rákokkal táplálkoznak, míg a felnőtt egyedek a fiatal egyedekkel táplálkoznak, tehát kannibálok.

Hivatkozások

[1] Cushing, J. M., *An Introduction to Structured Population Dynamics* (SIAM, Philadelphia, PA, 1998).

[2] Farkas, J. Z., Stability conditions for the non-linear McKendrick equations, *Applied Mathematics and Computations* **156** (2004) 771–777.

[3] Farkas, J. Z., On the asymptotic behaviour of the non-autonomous Gurtin–McCamy equation, to appear in *Annales Univ. Eötvös Sect. Math.* **46** (2004), 111–120.

[4] Farkas, J. Z., Stability conditions for a non-linear size-structured model, to appear in *Nonlinear Analysis Real World Applications* **6** (2005), 962–969.

[5] Farkas, M., On the stability of stationary age distributions, *Applied Mathematics and Computation* **131** (1) (2002) 107–123.

[6] Farkas, M., *Dynamical Models in Biology* (Academic Press, 2001).

[7] Gurtin, M. E. and McCamy, R. C., Non-linear age-dependent populations dynamics, *Arch. Rat. Mech. Anal.* **54** (1974) 281–300.

[8] Iannelli, M., *Mathematical Theory of Age-Structured Population Dynamics* (Giardini Editori, Pisa, 1994).

[9] McKendrick, A. G., Applications of mathematics to medical problems, *Proc. Edin. Math. Soc.* **44** (1926) 98–130.

[10] Metz, J. A. J. and Diekmann, O., *The Dynamics of Physiologically Structured Populations* (Lecture Notes in Biomath. 68, Springer-Verlag, Berlin, 1986).

[11] Webb, G., *Theory of Nonlinear Age-Dependent Population Dynamics* (Marcel Dekker, New York, 1985).

(Beérkezett: 2004. október 4.)

FARKAS JÓZSEF ZOLTÁN
BUDAPESTI MŰSZAKI EGYETEM
DIFFERENCIÁLEGYENLETEK TANSZÉK
BUDAPEST 1521
farkas@math.bme.hu

STABILITY OF AN AGE-STRUCTURED MODEL

JÓZSEF ZOLTÁN FARKAS

In 1974 Gurtin and MacCamy introduced a nonlinear version of the Lotka-McKendrick age-structured equation. In the last thirty years this PDE model and further generalizations served as basic models of structured population dynamics. Gurtin and MacCamy deduced a characteristic function corresponding to the stationary solution of the system. Recently Miklós Farkas deduced another equation, which enabled us to prove stabiliy results under biologically interpretable conditions on the vital rates. In this note we show the equivalence of the two characteristic equations and then we formulate our stability results in the most general form. At the end we point out an interesting phenomenon of the model.

Alkalmazott Matematikai Lapok **23** (2006), 121–137.

SÚLYOK MEGHATÁROZÁSA PÁROS ÖSSZEHASONLÍTÁS MÁTRIXOK LEGKISEBB NÉGYZETES KÖZELÍTÉSE ALAPJÁN

BOZÓKI SÁNDOR

Budapest

A páros összehasonlítások módszere a többszempontú döntési feladatok megoldásának egy lehetséges eszköze mind a szempontsúlyok meghatározásában, mind az alternatívák értékelésében. A szempontokat páronként összehasonlítva, fontosságaiknak a döntéshozó által megítélt arányait mátrixba rendezve a feladat a súlyvektor meghatározása úgy, hogy annak komponensei valamilyen értelemben jól illeszkedjenek a döntéshozó által megadott értékekhez.

A páros összehasonlítás mátrixból a súlyok kiszámítására leggyakrabban használt sajátvektor módszer (Analytic Hierarchy Process) mellett számos távolságminimalizáló módszer is létezik. Ezek egyike a legkisebb négyzetek módszere, melynek megoldása nemlineáris, nemkonvex függvény feltételes optimalizálását jelenti. A cikkben olyan módszereket mutatunk be a páros összehasonlítás mátrixok legkisebb négyzetes becslésére, amelyek a célfüggvény összes lokális és globális minimumhelyének meghatározására alkalmasak.

1. Páros összehasonlítás mátrixok

A többszempontú döntési modellekben a cél véges számú alternatíva véges számú szempont szerint történő rangsorolása. A pályázatok versenyeztetése, a vállalati stratégiák közül a legjobb kiválasztása, a közbeszerzési eljárások, adott pozícióra a legalkalmasabb személy kiválasztása olyan gyakorlati problémák, amelyek többszempontú döntési feladat megoldására vezetnek. A szempontok általában nem egyformán fontosak, szükség van tehát olyan módszerre, amely a szempontokat fontossági súlyokkal látja el úgy, hogy az a döntéshozó céljaival harmóniában álljon. A feladat egyik nehézsége, hogy a fontosságnak nincs általánosan elfogadott mértékegysége, azt csak valamilyen skálával együtt lehet értelmezni. Előfordul, hogy a döntéshozó közvetlenül, számszerűen meg tudja adni a szempontsúlyokat, ezt egyszerű közvetlen becslésnek is nevezi az irodalom [27]. Nagyobb méretű, összetett

feladatoknál azonban nem várható el, hogy a modellező rendelkezésére bocsássa a számszerűsített értékeket. A probléma kisebb részekre történő bontásával azonban elérhető, hogy a döntéshozónak csak egyszerű, világos kérdéseket kell megválaszolnia, azokból mégis előállítható az egész feladat szempontsúly-rendszere. Axiómaként elfogadjuk a preferencia-modellezésben használt feltételt, miszerint a döntéshozó képes két dolog (ami lehet pl. a szempontok fontossága) összehasonlítására: meg tudja mondani, hogy valamelyik jobb (vagy nagyobb) a másiknál, vagy egyformák.

Condorcet [10] és Borda [2] szavazási feladataikban már az 1780-as években bevezették a *páros* (vagy páronkénti) *összehasonlítás* fogalmát, mint az egyéni preferenciák alapján felállított rangsor két eleme közötti viszonyt. A páros összehasonlítás, mint módszer alkalmazási lehetőségeit Kindler [27] történeti és módszertani áttekintése tárgyalja, melyből itt csak a csak legfontosabbakat emeljük ki. A kísérleti pszichológiában az 1920-as években jelent meg e fogalom Thorndike [34] és Thurstone [35] munkáiban. Churchman és Ackoff [9] eljárásában az elemeket először ordinális értelemben rendezni kell, ezután valamelyiket rögzítve és a többivel kardinális értelemben összehasonlítva számszerű eredmények adódnak. Guilford [21] modelljében pusztán ordinális információk alapján kardinális sorrend állapítható meg. Több döntéshozó (csoportos döntéshozatal) esetére dolgozta ki Kendall [25] a róla elnevezett egyetértési együtthatót.

Bár e dolgozatnak nem célja az emberi racionalitás korlátait és paradoxonait firtatni, megjegyezzük, hogy a páronkénti összehasonlítások a döntéshozókkal történő elvégeztetésének fontos módszertani szempontja, hogy nem mindegy, milyen sorrendben tesszük fel a kérdéseket. A szabályos elrendezés szinte mindig torzít, a véletlenszerű már kevésbé, a Ross-féle elrendezés [32] pedig a véletlennél is kisebb torzítással működik.

A dolgozatban a páros összehasonlítások azon változatát tárgyaljuk, amelyben az elemeket arányskálán hasonlítjuk össze, azaz a döntéshozótól olyan formában várjuk az elemek összehasonlítását, hogy hányszor tekinti az egyiket *jobb*nak vagy *nagyobb*nak a másiknál [33]. A páronkénti összehasonlításokból felépíthető egy négyzetes mátrix, melynek definíciója a következő:

Definíció. (Páros összehasonlítás mátrix.) Jelölje $\mathbb{R}_+^{n\times n}$ a pozitív valós elemekből álló $n \times n$-es mátrixok osztályát. Az

$$\mathbf{A} = \begin{pmatrix} 1 & a_{12} & a_{13} & \dots & a_{1n} \\ 1/a_{12} & 1 & a_{23} & \dots & a_{2n} \\ 1/a_{13} & 1/a_{23} & 1 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & 1/a_{2n} & 1/a_{3n} & \dots & 1 \end{pmatrix} \in \mathbb{R}_+^{n\times n}$$

mátrixot páros összehasonlítás mátrixnak nevezzük, ha minden $i, j = 1, \dots, n$ indexre teljesül, hogy

$$a_{ii} = 1, \tag{1}$$

$$a_{ij} = \frac{1}{a_{ji}}. \tag{2}$$

A mátrix a_{ij} eleme azt mutatja meg, hogy a döntéshozó hányszor jobbnak ítéli meg az i-edik objektumot a j-ediknél. (1) alapján az önmagával való összehasonlítás eredménye mindig 1.

A (2) tulajdonság azon a feltételezésen alapul, hogy ha a döntéshozó számára az i-edik objektum a_{ij}-szer akkora, mint a j-edik, akkor a j-edik pontosan $\frac{1}{a_{ij}}$-szer akkora, mint az i-edik. Az (1)–(2)-ből adódóan n objektum esetén $\binom{n}{2} = \frac{n(n-1)}{2}$ összehasonlítással adható meg a mátrix.

Definíció (Konzisztens páros összehasonlítás mátrix). Ha egy

$$\mathbf{A} = [a_{ij}]_{i,j=1,2,\ldots,n} \in \mathbb{R}_+^{n\times n}$$

mátrixra (1)–(2)-n túl még

$$a_{ij}a_{jk} = a_{ik} \tag{3}$$

is teljesül minden $i, j, k = 1, \ldots, n$ indexre, akkor konzisztens páros összehasonlítás mátrixnak nevezzük. Az (1)–(2) feltételt igen, de (3)-at nem teljesítő mátrixot inkonzisztens mátrixnak nevezzük.

A feladat: az elemek páronkénti összehasonlításának ($\mathbf{A}$ mátrix) ismeretében a $w_1, w_2, \ldots, w_n$ súlyok meghatározása, ahol

$$w_i > 0, \qquad i = 1, 2, \ldots, n$$

$$\sum_{i=1}^{n} w_i = 1. \tag{4}$$

A súlyokat együttesen a $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ súlyvektorral jelöljük.

A problémára több megoldási lehetőség kínálkozik. Az Analytic Hierarchy Process (AHP) [33] módszertanban a mátrix legnagyobb sajátértékéhez ($\lambda_{\max}$) tartozó jobboldali sajátvektor komponensei adják a súlyokat. Más, távolságminimalizáló módszerekben a mátrix valamilyen célfüggvény szerinti legjobb közelítése alapján lehet a súlyokra következtetni. A legkisebb négyzetek módszere [8] és annak relaxált változatai, mint pl. a súlyozott legkisebb négyzetes [8], a logaritmikus legkisebb négyzetes [12, 11], vagy a χ^2-es [22] feladatok mellett olyan megközelítések találhatók, mint a szinguláris felbontás [19], célprogramozás [5], lineáris programozás [7].

Konzisztens mátrixok esetén minden egyes eljárás ugyanazt az eredményt adja. Inkonzisztens esetben a különböző módszerek által eredményezett súlyvektorok kisebb-nagyobb mértékben eltérnek. Golany és Kress [20] több szempont alapján történő összehasonlító elemzéséből kiderül, hogy minden súlyozási módszernek van előnye és hátránya, egyik sem nevezhető „a legjobb"-nak.

A többi módszerrel ellentétben a legkisebb négyzetes feladatról általában nem mondható el, hogy megoldása egyértelmű [22], [23]. A célfüggvény ugyanis nem feltétlenül konvex, és az eddigiekben publikált eljárásoknál ([23], [15]) jelentős nehézséget okoz a stacionárius pontok meghatározása, mivel azok az iterációs elvű numerikus módszereket használják.

A következő fejezetben olyan módszereket tekintünk át, amelyekkel megoldható a páros összehasonlítás mátrixok legkisebb négyzetes közelítése. Az eljárások előnye, hogy minden lokális és globális minimumhelyet megtalálnak, továbbá indulópont választására sincs szükség.

2. A legkisebb négyzetek módszere

Legyen adott az $\mathbf{A}$ $n \times n$-es páros összehasonlítás mátrix:

$$\mathbf{A} = \begin{pmatrix} 1 & a_{12} & a_{13} & \dots & a_{1n} \\ 1/a_{12} & 1 & a_{23} & \dots & a_{2n} \\ 1/a_{13} & 1/a_{23} & 1 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & 1/a_{2n} & 1/a_{3n} & \dots & 1 \end{pmatrix}.$$

Keressük azt a $\mathbf{w} = (w_1, w_2, \dots, w_n) \in \mathbb{R}^n_+$ vektort, amelynek komponenseiből képzett

$$\mathbf{X} = \begin{pmatrix} 1 & w_1/w_2 & w_1/w_3 & \dots & w_1/w_n \\ w_2/w_1 & 1 & w_2/w_3 & \dots & w_2/w_n \\ w_3/w_1 & w_3/w_2 & 1 & \dots & w_3/w_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & w_n/w_3 & \dots & 1 \end{pmatrix},$$

mátrix Frobenius-normában a legjobban közelíti $\mathbf{A}$-t. Az optimalizálási feladat tehát:

$$\min \| \mathbf{A} - \mathbf{X} \|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_{ij} - \frac{w_i}{w_j} \right)^2,$$

$$\sum_{i=1}^{n} w_i = 1,$$

$$w_1, w_2, \dots, w_n > 0.$$

Vezessük be az $x_1, x_2, \dots, x_{n-1}$ új változókat a következőképpen:

$$(5) \qquad x_1 = \frac{w_1}{w_2}, \quad x_2 = \frac{w_1}{w_3}, \quad \dots, \quad x_i = \frac{w_1}{w_{i+1}}, \quad \dots, \quad x_{n-1} = \frac{w_1}{w_n}.$$

Ekkor

$$\frac{w_i}{w_j} = \begin{cases} 1, & \text{ha } i = j; \\ x_{j-1}, & \text{ha } i = 1 \text{ és } 1 < j \leq n; \\ \frac{1}{x_{i-1}}, & \text{ha } j = 1 \text{ és } 1 < i \leq n; \\ \frac{x_{j-1}}{x_{i-1}}, & \text{ha } 1 < i, j \leq n, \end{cases}$$

így az $\mathbf{X}$ mátrix x_i $(i = 1, 2, \ldots, n-1)$ változókkal való felírása a következő:

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 & x_2 & \ldots & x_{n-1} \\ \frac{1}{x_1} & 1 & \frac{x_2}{x_1} & \ldots & \frac{x_{n-1}}{x_1} \\ \frac{1}{x_2} & \frac{x_1}{x_2} & 1 & \ldots & \frac{x_{n-1}}{x_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{x_{n-1}} & \frac{x_1}{x_{n-1}} & \frac{x_2}{x_{n-1}} & \ldots & 1 \end{pmatrix},$$

és az optimalizálási feladat

$$\min \| \mathbf{A} - \mathbf{X} \|_F^2 = f(x_1, x_2, \ldots, x_{n-1})$$
$$x_1, x_2, \ldots, x_{n-1} > 0,$$

alakban írható fel, ahol

$$f(x_1, x_2, \ldots, x_{n-1}) = \sum_{j=2}^{n} \left[(a_{1j} - x_{j-1})^2 + \left(\frac{1}{a_{1j}} - \frac{1}{x_{j-1}} \right)^2 \right]$$
$$+ \sum_{i=2}^{n-1} \sum_{j=i+1}^{n} \left[\left(a_{ij} - \frac{x_{j-1}}{x_{i-1}} \right)^2 + \left(\frac{1}{a_{ij}} - \frac{x_{i-1}}{x_{j-1}} \right)^2 \right].$$

Mivel f nyílt tartományon értelmezett differenciálható függvény, az optimalitás elsőrendű szükséges feltétele olyan pont létezése, amelyre

$$\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = \cdots = \frac{\partial f}{\partial x_{n-1}} = 0. \tag{6}$$

Az f függvény elsőrendű parciális deriváltjai az $x_1, x_2, \ldots, x_{n-1}$ változók racionális törtfüggvényei, hisz maga f is az volt. Adott i $(1 \leq i \leq n-1)$ indexhez tartozó x_i változó csak az $\mathbf{X}$ mátrix $(i+1)$-edik sorában és oszlopában fordul elő, ezért az $\frac{\partial f}{\partial x_i}$ parciális derivált így írható:

$$\frac{\partial f}{\partial x_i} =$$
$$= \frac{\partial \left\{ (a_{i+1,1} - x_i)^2 + (a_{1,i+1} - \frac{1}{x_i})^2 + \sum\limits_{\substack{j=2 \\ j \neq i+1}}^{n} \left[(a_{i+1,j} - \frac{x_i}{x_{j-1}})^2 + (a_{j,i+1} - \frac{x_{j-1}}{x_i})^2 \right] \right\}}{\partial x_i},$$

$$\frac{\partial f}{\partial x_i} = -2\left(a_{i+1,i} - x_i\right) + 2\left(a_{1,i+1} - \frac{1}{x_i}\right)\frac{1}{x_i^2} + \sum_{\substack{j=2 \\ j\neq i+1}}^{n}\left[-2\left(a_{i+1,j} - \frac{x_i}{x_{j-1}}\right)\frac{1}{x_{j-1}} + 2\left(a_{j,i+1} - \frac{x_{j-1}}{x_i}\right)\frac{x_{j-1}}{x_i^2}\right].$$

Mivel $\frac{\partial f}{\partial x_i}$ felírásában a nevezőben x_j^2 $(j = 1, 2, \ldots, n-1,\ j \neq i)$, valamint x_i^3 szerepel, a $\frac{\partial f}{\partial x_i}$-et $\left(x_i^3 \cdot \prod_{\substack{j=1 \\ j\neq i}}^{n-1} {x_j}^2\right)$-nel beszorozva a

$$P_i(x_1, x_2, \ldots, x_{n-1}) = \frac{1}{2}\frac{\partial f}{\partial x_i} x_i^3 \prod_{\substack{j=1 \\ j\neq i}}^{n-1} x_j^2 = \frac{1}{2}\frac{\partial f}{\partial x_i} x_i \prod_{j=1}^{n-1} {x_j}^2, \quad i = 1, 2, \ldots, n-1$$

többváltozós polinomokat kapjuk. A P_i $(i = 1, 2, \ldots, n-1)$ polinomok közös gyökeit a

$$\begin{aligned} P_1(x_1, x_2, \ldots, x_{n-1}) &= 0 \\ P_2(x_1, x_2, \ldots, x_{n-1}) &= 0 \\ &\vdots \\ P_{n-1}(x_1, x_2, \ldots, x_{n-1}) &= 0 \end{aligned} \tag{7}$$

rendszer megoldásai adják.

A döntési feladat szempontjából csak a pozitív valós $(x_1, x_2, \ldots, x_{n-1})$ gyökök érdekesek, így a (6) és (7) rendszerek egyenértékűek abban az értelemben, hogy egy pozitív valós $(x_1, x_2, \ldots, x_{n-1})$ $(n-1)$-es pontosan akkor megoldása (6)-nak, ha (7)-nek is.

Ha egy $(x_1^\star, x_2^\star, \ldots, x_{n-1}^\star)$ $(n-1)$-es f-nek minimumhelye, akkor szükségképpen megoldása a (7) polinomrendszernek is. Fordítva, ha a pozitív $(x_1^\star, x_2^\star, \ldots, x_{n-1}^\star)$ vektor megoldása a (7) polinomrendszernek, az f Hesse-mátrix pozitív definitségének ellenőrzésével tudjuk ellenőrizni, hogy valóban (lokális) minimumhely-e. Ha igen, akkor $(x_1^\star, x_2^\star, \ldots, x_{n-1}^\star)$-ból (5) és (4) alapján felírható a keresett $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ súlyvektor. Kifejezve ugyanis (5)-ből a w_i $(i = 2, 3, \ldots, n)$ súlyokat:

$$w_2 = \frac{w_1}{x_1}, \quad w_3 = \frac{w_1}{x_2}, \quad \ldots, \quad w_i = \frac{w_1}{x_{i-1}}, \quad \ldots, \quad w_n = \frac{w_1}{x_{n-1}},$$

majd az egyenleteket összeadva

$$\sum_{i=2}^{n} w_i = w_1 \sum_{i=1}^{n-1} \frac{1}{x_i}. \tag{8}$$

(4) szerint (8) baloldali kifejezése $(1 - w_1)$-gyel egyenlő, így w_1-re

$$w_1 = \frac{1}{1 + \sum_{j=1}^{n-1} \frac{1}{x_j}},$$

w_i-re $(1 < i \leq n)$ pedig az (5) megfelelő $\left(x_{i-1} = \frac{w_1}{w_i}\right)$ egyenletéből

$$w_i = \frac{w_1}{x_{i-1}} = \frac{\frac{1}{x_{i-1}}}{1 + \sum_{j=1}^{n-1} \frac{1}{x_j}}$$

adódik. Kaptuk tehát, hogy az LSM-optimális $\mathbf{w}$ súlyvektor a (7) polinomrendszer $\left(x_1^*, x_2^*, \ldots, x_{n-1}^*\right)$ megoldásából az alábbi formula szerint számolható:

$$w_1 = \frac{1}{1 + \sum_{j=1}^{n-1} \frac{1}{x_j^*}}, \qquad w_i = \frac{\frac{1}{x_{i-1}^*}}{1 + \sum_{j=1}^{n-1} \frac{1}{x_j^*}}, \qquad i = 2, 3, \ldots, n.$$

3. Polinomrendszerek megoldása

A matematikai (főleg geometriai) és fizikai-mérnöki problémák (kinetika és egyensúly) gyakran vezetnek polinomiális rendszerek megoldására, mely – mint a nemlineáris rendszerek megoldása általában – nem könnyű. Jelen fejezet áttekintést ad négy olyan módszerről, amelyek segítségével kisméretű feladatok megoldhatók. Mivel egy adott polinomrendszer összes megoldását keressük, a Newton-iteráción alapuló algoritmusokat nem tárgyaljuk. Megjegyezzük azonban, hogy valamely polinomrendszer-megoldó algoritmus által szolgáltatott megoldás, mely szükségképpen csak közelítő megoldás lehet, a Newton-iteráció indulóértékéül választva tetszőlegesen pontosítható.

3.1. Rezultáns módszer

Bevezetésül idézzük fel Gauss egyik legfontosabb eredményét.

TÉTEL. *(Az algebra alaptétele.) Minden nemkonstans komplex $f \in \mathbb{C}[x]$ polinomnak van gyöke a $\mathbb{C}$ számtestben.*

Legyenek f és g egyváltozós, valós együtthatós polinomok:

$$\begin{aligned} f(x) &= a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n, \\ g(x) &= b_0 x^m + b_1 x^{m-1} + \ldots + b_{m-1} x + a_m, \end{aligned}$$

ahol $a_0 \neq 0$, $b_0 \neq 0$. Az algebra alaptételéből következően f és g felírhatók gyöktényezős szorzatalakban:

$$f(x) = a_0 \prod_{i=1}^{n} (x - \alpha_i),$$

$$g(x) = b_0 \prod_{j=1}^{m} (x - \beta_j), \tag{9}$$

ahol $\alpha_i, \beta_j \in \mathbb{C}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$.

Definíció. Az f és g polinomok $R(f, g)$-vel jelölt **rezultánsa**

$$R(f, g) = a_0^m b_0^n \prod_{i=1}^{n} \prod_{j=1}^{m} (\alpha_i - \beta_j).$$

(9) alapján

$$g(\alpha_i) = b_0 \prod_{j=1}^{m} (\alpha_i - \beta_j),$$

és hasonlóan,

$$R(f, g) = a_0^m \prod_{i=1}^{n} g(\alpha_i).$$

A definícióból következik, hogy f-nek és g-nek pontosan akkor van közös gyöke $\mathbb{C}$-ben, ha $R(f, g) = 0$. Megjegyezzük, hogy a rezultáns definíciója nem szimmetrikus az argumentumokra nézve, igaz viszont, hogy

$$R(g, f) = b_0^n a_0^m \prod_{j=1}^{m} \prod_{i=1}^{n} (\beta_j - \alpha_i) = (-1)^{nm} R(f, g).$$

$R(g, f)$ egy másik alakban történő felírása:

$$R(g, f) = b_0^n \prod_{j=1}^{m} f(\beta_j).$$

Az alábbi tétel [28] szerint $R(f, g)$ nemcsak f és g gyökeiből, hanem közvetlenül az együtthatókból is számolható.

TÉTEL. *Jelölje D a következő (Sylvester-féle) mátrix determinánsát:*

$$D = \begin{vmatrix} a_0 & a_1 & a_2 & \dots & a_n & & & \\ & a_0 & a_1 & \dots & a_{n-1} & a_n & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & a_0 & a_1 & a_2 & \dots & a_n \\ b_0 & b_1 & b_2 & \dots & b_m & & & \\ & b_0 & b_1 & \dots & b_{m-1} & b_m & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & b_0 & b_1 & b_2 & \dots & b_m \end{vmatrix}_{(n+m)\times(n+m)},$$

ahol az üresen hagyott elemek 0-kat jelentenek. Ekkor

$$D = R(f, g).$$

Példa. Legyenek f és g

$$f(x) = x^2 - 5x - 14,$$

$$g(x) = x^2 - 6x - 7.$$

Az előző tétel szerint

$$R(f,g) = \begin{vmatrix} 1 & -5 & -14 & 0 \\ 0 & 1 & -5 & -14 \\ 1 & -6 & -7 & 0 \\ 0 & 1 & -6 & -7 \end{vmatrix} = 0,$$

tehát f és g-nek kell, hogy legyen közös gyöke ($x = 7$ valóban az). Mindezt anélkül kaptuk, hogy ki kellett volna számítanunk f és g gyökeit.

Legyen most adott a következő egyenletrendszer:

$$f(x, y) = 0, \tag{10}$$

$$g(x, y) = 0, \tag{11}$$

ahol $f, g \in \mathbb{R}[x, y]$ kétváltozós, valós együtthatós polinomok. Ha csak x-et tekintenénk változónak, y-t pedig paraméternek, akkor az f-ben és g-ben, mint egyváltozós polinomokban szereplő tagok sorbarendezhetők x kitevőjének nagysága szerint:

$$f(x, y) = a_0(y)x^k + a_1(y)x^{k-1} + \dots + a_{k-1}(y)x + a_k(y), \tag{12}$$

$$g(x, y) = b_0(y)x^l + b_1(y)x^{l-1} + \dots + b_{l-1}(y)x + a_l(y), \tag{13}$$

ahol $a_0(y), a_1(y), \ldots, a_k(y), b_0(y), b_1(y), \ldots, b_l(y) \in \mathbb{R}[y]$ valós együtthatós egyváltozós polinomjai y-nak. Felírható tehát $R_x(f,g)$, mint az f és g egyváltozós polinomok rezultánsa:

$$R_x(f,g) = \begin{vmatrix} a_0(y) & a_1(y) & a_2(y) & \ldots & a_k(y) & & & \\ & a_0(y) & a_1(y) & \ldots & a_{k-1}(y) & a_k(y) & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & a_0(y) & a_1(y) & a_2(y) & \ldots & a_k(y) \\ b_0(y) & b_1(y) & b_2(y) & \ldots & b_l(y) & & & \\ & b_0(y) & b_1(y) & \ldots & b_{l-1}(y) & b_l(y) & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & b_0(y) & b_1(y) & b_2(y) & \ldots & b_l(y) \end{vmatrix} = P(y),$$

ahol $P(y) \in \mathbb{R}[y]$ valós együtthatós egyváltozós polinomja y-nak.

Tegyük fel, hogy az $x = \alpha, y = \beta$ megoldása a (10)–(11) rendszernek. Behelyettesítve $y = \beta$-t (12)–(13)-be, az $f(x,\beta)$ és $g(x,\beta)$ egyváltozós polinomokat kapjuk, melyek közös gyöke α. Ha az $a_0(\beta), b_0(\beta)$ főegyütthatók nem nullák, akkor az $f(x,\beta)$ és $g(x,\beta)$ polinomok rezultánsa az alábbiak szerint írható fel:

$$R\big(f(x,\beta), g(x,\beta)\big) = \begin{vmatrix} a_0(\beta) & a_1(\beta) & a_2(\beta) & \ldots & a_k(\beta) & & & \\ & a_0(\beta) & a_1(\beta) & \ldots & a_{k-1}(\beta) & a_k(\beta) & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & a_0(\beta) & a_1(\beta) & a_2(\beta) & \ldots & a_k(\beta) \\ b_0(\beta) & b_1(\beta) & b_2(\beta) & \ldots & b_l(\beta) & & & \\ & b_0(\beta) & b_1(\beta) & \ldots & b_{l-1}(\beta) & b_l(\beta) & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & b_0(\beta) & b_1(\beta) & b_2(\beta) & \ldots & b_l(\beta) \end{vmatrix},$$

mely rezultánst kifejtve β-nak egy $P(\beta)$ polinomját kapjuk. Mivel α közös gyöke $f(x,\beta)$ és $g(x,\beta)$-nak, azt kaptuk, hogy $P(\beta) = 0$, azaz β gyöke a P-nek.

Másfelől, tegyük fel, hogy $P(y) = R_x(f,g)$ polinomnak gyöke az $y = \beta$. Ha az $a_0(\beta)$ és $b_0(\beta)$ főegyütthatók nem nullák, akkor $P(\beta) = R\big(f(x,\beta), g(x,\beta)\big)$. De $P(\beta) = 0$, így $f(x,\beta)$-nak és $g(x,\beta)$-nak szükségképpen van közös gyöke.

3.2. Általánosított rezultáns módszer

A rezultáns-módszer hátránya, hogy csak két egyenletből álló, kétváltozós rendszerek redukálhatók egyváltozós polinom valós gyökeinek megkeresésére. Az általánosított rezultánsok elmélete Bezout, Dixon [13], Kapur, Saxena és Young [24] nevéhez fűződik. Az általánosított, vagy Dixon-rezultáns szerepe megegyezik a rezultánséval: adott polinomrendszer esetén olyan indikátort keresünk, amely felírható a polinomok együtthatóinak függvényeként, továbbá pontosan akkor 0, ha a polinomrendszernek van megoldása. Az eljárás leírása lényegesen hosszabb, mint a

rezultáns módszeré, ezért itt inkább az algoritmus implementációjával és a futási tapasztalatokkal foglalkozunk.

Robert H. Lewis [30] létrehozta a *Fermat* computer algebra programcsomagot, melyet kifejezetten nagyméretű polinomiális és mátrix-számításokra tervezett. A 4×4-es mátrixok LSM-approximációjából származó 3 egyenletből álló 3 ismeretlenes polinomrendszer a Fermat szoftver segítségével megoldható [4]. A kapott egyváltozós polinom foka 26 és 137 között változik, az adott mátrix elemeinek függvényében. Ezen egyváltozós polinom pozitív valós gyökeinek megkeresésére a Maple program kielégítő eredményt ad. Az eddigi tapasztalatok alapján a pozitív valós gyökök száma 1 és 10 között mozog. Ez lehetőséget ad arra, hogy egyszerű visszahelyettesítéssel az eredeti 3 egyenletből álló 3 ismeretlenes polinomrendszer 3 egyenletből álló 2 ismeretlenes polinomrendszerre redukálódjon. A 3 egyenletből egyszerre csak kettőt tudunk megoldani a rezultáns módszer segítségével, viszont képezve a 3 lehetséges egyenletpár megoldásainak metszetét az eredeti polinomrendszer közös gyökei immáron rendelkezésünkre állnak.

3.3. Gröbner-bázisok

A polinomgyűrűk és ideálok tanulmányozására vezette be Buchberger [6] a Gröbner-bázis fogalmát, mely elnevezést PhD témavezetője iránti tisztelete jeléül választotta.

Egy adott polinomrendszerhez tartozó Gröbner-bázis egy az eredetivel ekvivalens rendszer, azaz pontosan ugyanazok a gyökei, mint az eredetinek. A Gröbner-bázisbeli polinomrendszer azonban rendelkezik bizonyos tulajdonságokkal is, melyek jól használhatók a polinomokkal való osztás és egyéb vizsgálatok során. A Maple szoftverben írt programjaink futási eredményei azt mutatják, hogy a 3×3-as páros összehasonlítás mátrixokból kapott polinomrendszerre még működik az algoritmus, nagyobb méretekre azonban memória-túlcsordulás miatt leáll.

3.4. Homotópiás módszer

Az utóbbi 25 év során a homotópiás kontinuitási módszerek megbízható és hatékony technikává fejlődtek a polinomiális rendszerek összes megoldásának meghatározására.

Garcia és Zangwill [18], valamint tőlük függetlenül Drexler [14] javasolta elsőként a homotópiás módszerek alkalmazását polinomiális rendszerek összes gyökének numerikus meghatározására. A homotópiás kontinuitás módszer alapgondolata a következő: Adott $\mathbf{P} = (P_1, P_2, \ldots, P_n)$,

$$\begin{aligned} P_1(x_1, x_2, \ldots, x_n) &= 0, \\ P_2(x_1, x_2, \ldots, x_n) &= 0, \\ &\vdots \\ P_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}$$

polinomrendszerhez definiáljuk a $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_n)$

$$\begin{aligned} Q_1(x_1, x_2, \ldots, x_n) &= 0, \\ Q_2(x_1, x_2, \ldots, x_n) &= 0, \\ &\vdots \\ Q_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}$$

polinomrendszert úgy, hogy $\mathbf{Q}$ gyökeit már ismerjük. Legyen $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ és definiáljuk a

$$H(\mathbf{x}, t) = (1-t)\mathbf{Q}(\mathbf{x}) + t\mathbf{P}(\mathbf{x}) = 0$$

parametrikus egyenletrendszert, ahol $0 \leq t \leq 1$. $\mathbf{Q}$ megválasztásánál arra kell ügyelni, hogy a következő tulajdonságok teljesüljenek:

(1) trivialitás: $\mathbf{Q}(\mathbf{x}) = 0$ megoldásai ismertek;

(2) simaság: a $H(\mathbf{x}, t) = 0$ $(0 \leq t \leq 1)$ megoldáshalmaza véges sok sima útból áll, melyek mindegyike t-vel paraméterezhető;

(3) elérhetőség: a $H(\mathbf{x}, 1) = \mathbf{P}(\mathbf{x}) = 0$ rendszer minden izolált megoldása elérhető valamely $t = 0$-ból induló út mentén, mely út kezdőpontja tehát a $H(\mathbf{x}, 0) = \mathbf{Q}(\mathbf{x}) = 0$ rendszer egy megoldása.

Jelölje d_i a P_i polinom teljes fokát,

$$d_i = \deg P_i(\mathbf{x}), \qquad i = 1, 2, \ldots, n,$$

és legyen $d = d_1 \cdot d_2 \cdot \ldots \cdot d_n$. A többváltozós polinomokra vonatkozó Bezout-tétel értelmében a $P_1, P_2, \ldots, P_n$ polinomok teljes fokainak szorzata (d) felső becslést ad a közös gyökök számára (multiplicitással) $\mathbb{C}^n$-ben. Q választására gyakran a következő hatványfüggvények adódnak:

$$\begin{aligned} Q_1(x_1, x_2, \ldots, x_n) &= a_1 x_1^{d_1} = 0, \\ Q_2(x_1, x_2, \ldots, x_n) &= a_2 x_2^{d_2} = 0, \\ &\vdots \\ Q_n(x_1, x_2, \ldots, x_n) &= a_n x_n^{d_n} = 0, \end{aligned}$$

ahol $d_i = \deg P_i(\mathbf{x})$, $i = 1, 2, \ldots, n$, az a_j, b_j, $j = 1, 2, \ldots, n$ pedig tetszőleges, általában véletlenszerűen generált komplex számok. Ezek teljesítik a fenti három tulajdonságot, így a $\mathbf{P}(\mathbf{x}) = 0$ gyökei a $H(\mathbf{x}, t) = 0$ $(0 \leq t \leq 1)$ megoldásaként adódó d számú út végpontjai között keresendők.

A tapasztalatok szerint azonban d értéke nagyságrendekkel nagyobb lehet, mint a keresett gyökök száma, és az utak többsége nem tényleges gyökhöz konvergál, hanem a végtelenbe. A polinomrendszerek gyökszámának jobb becslésére szolgál Bernshtein [1], Kushnirenko [29] és Khovanskii [26] módszere, amely a kisebb számú út vizsgálatával a homotópiás módszert hatékonyabbá teszi.

E cikk szerzőjének lehetősége nyílt Tien-Yien Li és Tangan Gao [31, 17] algoritmusának tesztelésére. Az 1. táblázatban összefoglaltuk a páros összehasonlítás mátrixokra felírt legkisebb négyzetes közelítés feladatából adódó polinomrendszerek megoldásának átlagos adatait és a homotópiás algoritmus futási idejét 1 GHz-es processzoron.

A 3×3-as eset elemzésében [3] található olyan mátrix-konstrukció, amelyhez tartozó LSM-feladatnak négy lokális minimumhelye van. Ezek azonban gyakorlati szempontból nem tűnnek elsődleges fontosságúnak. A 3×3-asnál nagyobb esetben néhány tapasztalati (konkrét feladatra döntéshozó által megadott), valamint véletlenszerűen generált páros összehasonlítás mátrixokat vizsgáltunk. Számításaink szerint a tapasztalati mátrixok esetén a legkisebb négyzetes súlyvektor az esetek döntő részében egyértelmű, de még a véletlenszerűen generált mátrixok esetében is csak elvétve fordult elő 2 megoldásnál több.

A mátrix mérete $(n \times n)$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$
CPU time	0.05 mp.	0.5 mp.	20 mp.	14 perc	10 óra	3 nap
Közös gyökök száma	24	224	1840	14000	$\sim 10^5$	$\sim 10^6$
Közös pozitív valós gyökök száma	1 és 7 között					

1. táblázat. Polinomrendszerek megoldása $(n = 3, 4, \ldots, 8)$

4. Kutatási lehetőségek

Döntési szempontból alapvető fontosságú annak biztosítása, hogy egy páros összehasonlítás mátrixból számolt súlyvektor egyértelmű legyen, ami az 1. fejezetben említett súlyozási módszerek esetében biztosított. A legkisebb négyzetes megoldás egyértelműségére vonatkozó szükséges és elégséges feltétel azonban még nem ismert. A páros összehasonlítás mátrixok egy osztályában a megoldás nemegyértelműségére Farkas és Rózsa [16] adott elégséges feltételt.

Döntéselméleti és alkalmazási szempontból is lényeges kérdés a különböző súlymeghatározó módszerek összehasonlítása. Célunk, hogy a döntési feladatok jellegzetes vonásainak, típusainak leginkább megfelelő módszert ki tudjuk választani. Ezen jellegzetességek feltérképezése és azonosítása jelenleg is kutatás tárgya.

5. Összegzés

A cikkben egy, a többszempontú döntési feladatok szempontsúly-rendszerének kialakítására szolgáló módszert vizsgáltunk. Négy eljárást tekintettünk át a páros összehasonlítás mátrixok legkisebb négyzetes közelítéséből (LSM) adódó súlyok meghatározására. A kapcsolódó nemlineáris célfüggvény nemkonvexitása miatt az optimumhely általában nem egyértelmű. Az általunk tárgyalt módszerek az összes lokális és globális minimumhely megkeresésére alkalmasak. Tapasztalataink alapján a 3×3-as mátrixok esetére használható a rezultáns-módszer és a Gröbner-bázisok, 3×3-as és 4×4-es esetben az általánosított rezultánsokat alkalmazó Fermat szoftver, 3×3-astól 8×8-as méretig pedig a homotópiás kontinuitási módszer.

A kutatás jelenlegi fázisában a 3×3-as esetben tudunk páros összehasonlítás mátrixokat nagy számban generálni, majd azokból automatikusan súlyokat számolni. Ez lehetőséget ad a súlyozás szabályszerűségeinek feltárására, valamint a véletlen és a döntéshozó által megadott mátrixok összevetésére. 3×3-as mátrixokra a tárgyalt 4 módszer mindegyike lényegében azonnali eredményt ad, ezért kis méretű döntési problémák szempont-súlyozásában felhasználhatók.

A 4×4-estől 8×8-as méretig a súlyok számítása egyedileg történik, ezért a statisztikai jellegű elemzés lehetősége korlátozott. A futási eredmények (különösen $n = 7, 8$ esetében) azt mutatják, hogy döntési feladatok valós időben történő megoldására még nem alkalmazhatók, az általunk alkalmazott módszertan a kutatás fázisában van.

A legkisebb négyzetes közelítésből számolt súlyvektor ismeretében lehetőség nyílik e módszer sajátosságainak feltárására valamint más súlymeghatározó módszerekkel való összevetésre. A módszerek előnyeinek és hátrányainak pontosabb ismeretével közelebb kerülünk ahhoz a célhoz, hogy a döntési feladattípusok alapfeltevéseinek megfelelően ki tudjuk jelölni az alkalmazható súlymeghatározó módszerek csoportját.

Köszönettel tartozom Tangan Gao-nak (Michigan State University) a homotópiás algoritmus rendelkezésemre bocsátásáért, valamint Stefán Péternek (Nemzeti Információs Infrastruktúra Fejlesztési Program (NIIF) Szuperszámítógép Központ) a program futtatásában nyújtott technikai segítségéért.

A tanulmány az Országos Tudományos Kutatási Alapprogramok OTKA-T043-276 és OTKA-T043241 számú pályázatainak támogatásával készült.

Hivatkozások

[1] Bernshtein, D. N., The number of roots of a system of equations, *Functional Analysis and its Applications* **9** (1975) 183–185.

[2] Borda, J. C. de, Mémoire sur les élections au scrutin, *Histoire de l'Académie Royale des Sciences* (Paris, 1781).

[3] Bozóki, S., A method for solving LSM problems of small size in the AHP, *Central European Journal of Operations Research* **11** (2003) 17–33.

[4] Bozóki, S. and Lewis, R. H., Solving the Least Squares Method problem in the AHP for 3×3 and 4×4 matrices, *Central European Journal of Operations Research* **13** (2005), 255–270.

[5] Bryson, N. A goal programming method for generating priority vectors, *Journal of the Operational Research Society* **46**, No. 5 (1995) 641–648.

[6] Buchberger, B., An algorithmic method in polynomial ideal theory, in: N. K. Bose (editor), *Multidimensional System Theory*, 184–232 (D. Reidel Publishing Company, Dordrecht, Boston, Lancaster, 1985).

[7] Chandran, B., Golden, B. and Wasil, E., Linear programming models for estimating weights in the analytic hierarchy process, *Computers & Operations Research* **32** (2005) 2235–2254.

[8] Chu, A. T. W., Kalaba, R. E. and Spingarn, K., A comparison of two methods for determining the weight belonging to fuzzy sets, *Journal of Optimization Theory and Applications* **4** (1979) 531–538.

[9] Churchman, C. W., Ackoff, R. L. and Arnoff, L. E., *Introduction to Operations Research* (Wiley, New York, 1957).

[10] Condorcet, M., Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues á la Pluralité des Voix, Paris, 1785.

[11] Crawford, G. and Williams, C., A note on the analysis of subjective judgment matrices, *Journal of Mathematical Psychology* **29** (1985) 387–405.

[12] De Jong, P., A statistical approach to Saaty's scaling methods for priorities, *Journal of Mathematical Psychology* **28** (1984) 467–478.

[13] Dixon, A. L., The eliminant of three quantities in two independent variables, *Proceedings of the London Mathematical Society* **7** (1908) 50–69, 473–492.

[14] Drexler, F. J., Eine Methode zur Berechnung sämtlicher Lösungen von Polynomgleichungssystemen, *Numerische Mathematik* **29** (1978) 45–58.

[15] Farkas, A., Lancaster, P. and Rózsa, P., Consistency adjustment for pairwise comparison matrices, *Numerical Linear Algebra with Applications* **10** (2003) 689–700.

[16] Farkas, A. and Rózsa, P., On the Non-Uniqueness of the Solution to the Least-Squares Optimization of Pairwsie Comparison Matrices, *Acta Polytechnica Hungarica, Journal of Applied Sciences at Budapest Polytechnic Hungary* **1** (2004) 1–20.

[17] Gao, T., Li, T. Y. and Wang, X., Finding isolated zeros of polynomial systems in $\mathbb{C}^n$ with stable mixed volumes, *Journal of Symbolic Computation* **28** (1999) 187–211.

[18] Garcia, C. B. and Zangwill, W. I., Finding all solutions to polynomial systems and other systems of equations, *Mathematical Programming* **16** (1979) 159–176.

[19] Gass, S. I. and Rapcsák, T., Singular value decomposition in AHP, *European Journal of Operational Research* **154** (2004) 573–584.

[20] Golany, B. and Kress, M., A multicriteria evaluation of methods for obtaining weights from ratio-scale matrices, *European Journal of Operational Research* **69** (1993) 210–220.

[21] Guilford, J. P., *Psychometric Methods* (McGraw-Hill Book, New York, 1936).

[22] Jensen, R. E., Comparison of Eigenvector, Least squares, Chi square and Logarithmic least square methods of scaling a reciprocal matrix (Working Paper 153, 1983). http://www.trinity.edu/rjensen/127wp/127wp.htm

[23] Jensen, R. E., An Alternative Scaling Method for Priorities in Hierarchical Structures, *Journal of Mathematical Psychology* **28** (1984) 317–332.

[24] Kapur, D., Saxena, T. and Yang, L., Algebraic and geometric reasoning using Dixon resultants, in: *Proceedings of the International Symposium on Symbolic and Algebraic Computation* (A.C.M. Press, 1994).

[25] Kendall, M. G., *Rank Correlation Methods* (C. Griffin & Co., London, 1948).

[26] Khovanskii, A. G., Newton polyhedra and the genus of complete intersections, *Functional Analysis and its Applications* **12** (1978) 38–46.

[27] Kindler, J. és Papp, O., *Komplex rendszerek vizsgálata – Összemérési módszerek* (Műszaki Könyvkiadó, Budapest, 1977).

[28] Kuros, A. G., *Felsőbb algebra* (Tankönyvkiadó, Budapest, 1971).

[29] Kushnirenko, A. G., Newton polytopes and the Bézout theorem, *Functional Analysis and its Applications* **10** (1976) 233–235.

[30] Lewis, R. H., Computer algebra system *Fermat*, http://www.bway.net/~lewis/

[31] Li, T. Y., Numerical solution of multivariate polynomial systems by homotopy continuation methods, *Acta Numerica* **6** (1997) 399–436.

[32] Ross, R. T., Optimum orders for the presentation of pairs in the method of paired comparison, *Journal of Educational Psychology* **25** (1934) 375–382.

[33] Saaty, T. L., *The analytic hierarchy process* (McGraw-Hill, New York, 1980).

[34] Thorndike, E. L., A Constant Error in Psychological Ratings, *Journal of Applied Psychology* **4** (1920) 25–29.

[35] Thurstone, L. L., The Method of Paired Comparisons for Social Values, *Journal of Abnormal and Social Psychology* **21** (1927) 384–400.

(Beérkezett: 2005. április 8.)

BOZÓKI SÁNDOR
MTA SZTAKI
OPERÁCIÓKUTATÁS ÉS DÖNTÉSI RENDSZEREK LABORATÓRIUM
1518 BUDAPEST, PF. 63
bozoki@oplab.sztaki.hu

WEIGHTS FROM THE LEAST SQUARES APPROXIMATION OF PAIRWISE COMPARISON MATRICES

SÁNDOR BOZÓKI

The method of pairwise comparisons is one of the tools for determining the weights of attributes or evaluating the alternatives in Multi Attribute Decision Making. The decision maker is requested to compare pairwise the importance of the attributes, then the rates are arranged in a matrix. The aim is to find the weight vector which best reflects the values given by the decision maker.

There exist many distance minimizing methods, besides the Eigenvector Method (Analytic Hierarchy Process) for determining weights from a pairwise comparison matrix. One of them is the Least Squares Method, the solution of which leads to the optimization of a nonlinear, non-convex function. In the paper, methods for solving the problem of least squares approximation of pairwise comparison matrices are presented. All methods are suitable for finding all local and global optima.

Alkalmazott Matematikai Lapok **23** (2006), 139–161.

A DUÁL SZIMPLEX ALGORITMUS ELSŐ FÁZISÁNAK VIZSGÁLATA

MAROS ISTVÁN

London

A cikk egy új, a szimplex módszer számára készített duál első fázis eljárás elméleti és számítástechnikai vizsgálatával foglalkozik. Megmutatja, hogy a GDPO algoritmus lényegesen jobb teljesítményt képes nyújtani egy duál megengedett bázis megtalálásában, mint a hagyományos módszer. A 48 feladaton végrehajtott kísérletek meggyőzően igazolják, hogy az algoritmus előnyös elméleti tulajdonságai a gyakorlatban ténylegesen és nagy mértékben érvényesülni tudnak.

1. Bevezetés

A lineáris programozási (LP) feladat megoldására Dantzig [1] által kidolgozott szimplex algoritmus különféle variánsai rendkívül fontos szerepet játszanak az optimalizálás elméletében és gyakorlatában. A Karmarkar [3] által bevezetett és sok szerző (cf. Terlaky et al. [7, 8]) által továbbfejlesztett belsőpontos algoritmusok (BPA) nem homályosították el a szimplex alapú algoritmusok jelentőségét. Bebizonyosodott, hogy a két algoritmus család jól kiegészíti egymást. Bizonyos feladattípusokra a BPA-k a jobbak, másokra pedig a szimplex eljárások hatékonyabbak.

A szimplex módszernek két alapvető változata van: a primál és a duál algoritmus. Kezdetben a primál módszer kapta a több figyelmet. Ennek megfelelően a (primál) szimplex alapú programrendszerek folyamatos algoritmikus és implementációs fejlődésen mentek át és egyre nagyobb feladatokat tudtak megoldani egyre megbízhatóbban és hatékonyabban. A duál szimplex ettől jelentősen elmaradt, és használata elsősorban arra az esetre korlátozódott, amikor egy megengedett (fizibilis) bázis a rendelkezésre állt. Ez volt a helyzet az egészértékű LP feladatok egyszerű branch-and-bound (B&B) módszerrel történő megoldása esetén. Ekkor ugyanis egy származtatott relaxált LP feladatra nézve a megelőző LP relaxáció optimális bázi-

sa egy duál megengedett bázis. Erről indulva a duál módszernek általában sokkal kevesebb iterációra van szüksége, mint a primál módszernek (lásd pl. [4]).

A korszerű B&B algoritmusok lokális technikákat alkalmaznak a kereső fa csúcsainál, mint például *logical testing, implied bounds, added cuts.* Ilyen esetekben már nem igaz, hogy az előd LP feladat optimális bázisa duál megengedett marad a származtatott feladat számára. Így a duál algoritmus duál első fázissal tud csak indulni, de még mindig ez a jobb választás a primállal szemben.

A kedvező tapasztalatok alapján felmerült a kérdés, vajon a duál algoritmus egyenértékű, esetleg jobb alternatívája tud-e lenni a primálnak nagyméretű LP feladatok megoldására. A duál második fázis sikeres fejlesztése [4, 5, 2] után a duál első fázis algoritmikus maradt a hiányzó láncszem. Ezen dolgozva jutott el a szerző a GDPO algoritmushoz [6, 5], amely a fenti kérdésre pozitív választ adott. Elméletileg kimutatható, hogy az új algoritmusnak számos kedvező tulajdonsága van. Az, hogy ezek a valóságban milyen mértékben realizálódnak, még nem volt ismeretes. Jelen cikk ezt a hiányt pótolja és pozitív következtetésekre jut.

A továbbiakban a 2. szakasz a vizsgált általános alakú LP feladatot fogalmazza meg, amit a 3. szakaszban a GDPO algoritmus leírása követ. A 4. szakasz GDPO elméleti vizsgálatát tárgyalja, míg a 5. szakasz a számítástechnikai vizsgálatot mutatja be. A 6. szakasz a következtetéseket összegzi.

2. A feladat megfogalmazása

A gyakorlat szempontjából fontos és az elmélet számára sem közömbös, hogy az LP feladatot a természetes felmerülésnek megfelelő formában lehessen kezelni. Ez szükségessé teszi az előforduló összes típusú változó és feltétel algoritmikus kezelését. Ez nem pusztán esztétikai kérdés, hanem a hatékonyság növelésének a forrása is.

2.1. A primál feladat

Egy LP feladat legáltalánosabb alakja a következő:

$$\min z = c_0 + c_1 x_1 + \cdots + c_n x_{\bar{n}},$$

feltéve, hogy a közös korlátok

$$L_i \leq \sum_{j=1}^{\bar{n}} a_{ij} x_j \leq U_i, \quad i = 1, \ldots, m,$$

és a változók egyedi korlátai

$$\ell_j \leq x_j \leq u_j \quad j = 1, \ldots, \bar{n}$$

is teljesülnek. Az alsó $(L_i$ vagy $\ell_j)$ korlátok közül bármelyik lehet $-\infty$. Hasonlóan, a felső $(U_i$ or $u_j)$ korlátok között is lehet $+\infty$.

Egyszerű átalakítások és a közös feltételekhez egy-egy z_i logikai változó hozzáadása után a feltételek az alábbi alakra hozhatók:

$$z_i + \sum_{j=1}^{\bar{n}} a_{ij} x_j = b_i, \quad i = 1, \ldots, m, \tag{1}$$

ahol a változók egyedi korlátai:

(2)

Megengedett tartomány					Típus	Hivatkozás
		z_i, x_j	$=$	0	0	Fix
0	$\leq$	z_i, x_j	$\leq$	u_j	1	Korlátos
0	$\leq$	z_i, x_j	$\leq$	$+\infty$	2	Nem-negatív
$-\infty$	$\leq$	z_i, x_j	$\leq$	$+\infty$	3	Szabad

A változók típusára $\text{type}(x_j)$ vagy $\text{type}(z_i)$ módon fogunk hivatkozni.

Az LP feladat az (1) feltételrendszerrel mátrix alakban is felírható:

$$\min \ \mathbf{c}^T\mathbf{x} \tag{3}$$

$$\text{s.t.} \ \ \mathbf{A}\mathbf{x} + \mathbf{I}\mathbf{z} = \mathbf{b} \tag{4}$$

és a változók eleget tesznek (2)-nek.

Az x változókat *strukturális*, a z változókat pedig *logikai* változóknak nevezzük. Miután c_0 nem játszik szerepet az optimalizálásban, (3)-ban már nem tüntettük fel.

Technikai és algoritmikus szempontból a logikai és strukturális változók egyenrangú szerepet játszanak és általában nem szükséges megkülönböztetni őket. Ha a (4) bal oldalán álló mátrixot és változókat újra definiáljuk:

$$\mathbf{x} := \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \quad \mathbf{c} := \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{A} := [\mathbf{A} \mid \mathbf{I}],$$

ahol $\mathbf{0}$ az m dimenziós null vektor, akkor a feladat a következő alakra hozható:

(5)

$$\boxed{\begin{aligned} &\min \mathbf{c}^T\mathbf{x} \\ &\text{s.t.} \ \ \mathbf{A}\mathbf{x} = \mathbf{b} \\ &\qquad \boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned}}$$

az $\boldsymbol{\ell}$ és $\mathbf{u}$ vektorok értelemszerű kiterjesztése után. $\mathbf{x}$ és $\mathbf{c}$ új dimenziója $n = \bar{n} + m$. (5)-öt az LP feladat primál alakjának, vagy primál LP feladatnak szokás nevezni.

(5) egy bázisát $\mathbf{B}$-vel jelöljük, míg a bázisváltozók indexhalmazát $\mathcal{B}$-vel, a többi változó indexhalmazát pedig $\mathcal{R}$-rel. Az összes változó indexhalmaza: $\mathcal{N} = \mathcal{B} \cup \mathcal{R}$ és

$|\mathcal{N}| = n$. Az általánosság megszorítása nélkül feltesszük, hogy $\mathbf{B}$ az $\mathbf{A}$ mátrix első m oszlopa. Ez $\mathbf{A}$ particionálását eredményezi $\mathbf{A} = [\mathbf{B} \mid \mathbf{R}]$ alakban, ahol $\mathbf{R}$ jelöli $\mathbf{A}$ nem-bázis részét. Ennek megfelelően particionáljuk $\mathbf{x}$-et és $\mathbf{c}$-t is. $\mathbf{A}$ mátrix j-edik oszlopát $\mathbf{a}_j$-vel, míg i-edik sorát $\mathbf{a}^i$-vel jelöljük. (5)-nek a $\mathbf{B}$-hez tartozó bázis megoldása

$$\mathbf{x}_B = \boldsymbol{\beta} = \mathbf{B}^{-1}\Big(\mathbf{b} - \sum_{j \in \mathcal{U}} u_j \mathbf{a}_j\Big),$$

ahol $\mathcal{U}$ jelöli azon bázison kívüli változók indexhalmazát, amelyek felső korláton vannak. Az i-edik bázisváltozót β_i-vel jelöljük. A d_j redukált költség definíciója $d_j = c_j - \boldsymbol{\pi}^T \mathbf{a}_j = c_j - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{a}_j$, ami tovább egyenlő $c_j - \mathbf{c}_B^T \boldsymbol{\alpha}_j$-vel, ha az $\boldsymbol{\alpha}_j = \mathbf{B}^{-1}\mathbf{a}_j$ jelölést használjuk.

2.2. A duál feladat

A duál feladathoz a primál dualizálásán keresztül jutunk el. Először a primál feladat azon változatát tekintjük, ahol minden változó 2-es típusú (nem-negatív):

Ha a primál

$$\begin{aligned} \text{(P1)} \quad & \min \ \mathbf{c}^T\mathbf{x} \\ & \text{s.t.} \ \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

alakban van megadva, akkor ennek a duálja

$$\begin{aligned} \text{(D1)} \quad & \max \ \mathbf{b}^T\mathbf{y} \\ & \text{s.t.} \ \mathbf{A}^T\mathbf{y} \leq \mathbf{c}. \end{aligned}$$

Megjegyzendő, hogy itt az $\mathbf{y}$ duál változók 3-as típusú (szabad) változók.

A $\mathbf{w} = [w_1, \ldots, w_n]^T$ duál logikai változók bevezetésével (D1) felírható egyenlőség formában:

$$\max \ \mathbf{b}^T\mathbf{y}$$

$$\text{s.t.} \ \mathbf{A}^T\mathbf{y} + \mathbf{w} = \mathbf{c}, \tag{6}$$

$$\mathbf{w} \geq \mathbf{0}. \tag{7}$$

Legyen $\mathbf{B}$ az $\mathbf{A}$ mátrix egy bázisa, aminek nem kell primál megengedettnek lenni. (6) átrendezésével a $\mathbf{w}^T = \mathbf{c}^T - \mathbf{y}^T\mathbf{A}$ alakot kapjuk, ami particionált formában:

$$\mathbf{w}_B^T = \mathbf{c}_B^T - \mathbf{y}^T\mathbf{B},$$

$$\mathbf{w}_R^T = \mathbf{c}_R^T - \mathbf{y}^T\mathbf{R}.$$

A (7) alatti, $\mathbf{w}$-re vonatkozó nem-negativitási feltétel particionálva: $[\mathbf{w}_{\mathcal{B}}^T, \mathbf{w}_{\mathcal{R}}^T]^T \geq \mathbf{0}$. Az $\mathbf{y}^T = \mathbf{c}_{\mathcal{B}}^T \mathbf{B}^{-1}$ választással azt kapjuk, hogy

$$\mathbf{w}_{\mathcal{B}}^T = \mathbf{c}_{\mathcal{B}}^T - \mathbf{c}_{\mathcal{B}}^T \mathbf{B}^{-1}\mathbf{B} = \mathbf{0}, \tag{8}$$

$$\mathbf{w}_{\mathcal{R}}^T = \mathbf{c}_{\mathcal{R}}^T - \mathbf{c}_{\mathcal{B}}^T \mathbf{B}^{-1}\mathbf{R} = \mathbf{d}_{\mathcal{R}}^T \geq \mathbf{0}, \tag{9}$$

ahol $\mathbf{d}_{\mathcal{R}}$ a primál redukált költségekből alkotott vektor bázison kívüli részét jelenti. Mivel (8) teljesül tetszőleges bázis esetén és $\mathbf{y}$ szabad változó, a $\mathbf{B}$ bázis duál megengedett, ha kielégíti (9)-et. Ez azonban nem más, mint a primál optimalitási feltétel. Vagyis a duál megengedettségi feltétel azonos a primál optimalitási feltétellel és a primál redukált költségek egyben a duál logikai változók.

A gyakorlatban a duál szimplex algoritmusok a primál feladaton dolgoznak, használva annak számítástechnikai eszköztárát, de a báziscserét a duál algoritmus szabályai szerint hajtják végre.

Nagyméretű LP feladatokat a gyakorlatban csak akkor lehet megoldani, ha kihasználjuk a feltételi mátrixok ritkásságát (kis kitöltöttségét). Ez a módosított szimplex módszer keretein belül valósítható meg a legjobban. Ilyen esetben csak az iterációhoz szükséges transzformált elemeket kell meghatározni. A duál szimplex algoritmusoknál mind első, mind második fázisban számítástechnikailag a „legdrágább" művelet a transzformált pivot sor előállítása. Ennek a sornak az esetleges többszörös felhasználása jelentősen motiválta a szerzőnek az általánosított duál első fázis eljárásra irányuló vizsgálatait.

A 3. szakaszban bemutatandó új algoritmus, amelyre GDPO (Generalized Dual Phase One) néven fogunk hivatkozni, egy iterációval képes annyi haladást elérni, mint a hagyományos módszer sok iterációval. Mindezt a drágán előállított transzformált pivot sor többszöri felhasználásával éri el. Ezen túlmenően, a GDPO-nak olyan további tulajdonságai is vannak, amelyek algoritmikusan és számítástechnikailag is egyaránt igen előnyösek. Ezeket a 4. és 5. szakaszok tárgyalják.

3. A GDPO algoritmus leírása

3.1. Elméleti alapok

A GDPO algoritmus részletesebb tárgyalása Maros eredeti cikkében (l. [6]) található.

Az LP (5) alatti általános esetére is igaz (lásd pl. [5]), hogy a primál redukált költségek azonosak a duál logikai változókkal. Ezért az (5) minimalizálási feladat duáljának a megengedett megoldásai kielégítik a következő feltételeket. Megjegyzendő, hogy a d_j duál logikai változók a primál strukturális változókkal állnak összefüggésben.

(10)

Type(x_j)	Érték	d_j	Megjegyzés
0	$x_j = 0$	Mindegy	
1	$x_j = 0$	≥ 0	
1	$x_j = u_j$	≤ 0	$j \in \mathcal{U}$
2	$x_j = 0$	≥ 0	
3	$x_j = 0$	$= 0$	

Ebből adódóan egy $(\mathcal{B},\mathcal{U})$ halmazpárhoz tartozó duál megoldás megengedett, ha teljesíti (10)-et.

Miután a fix (type-0) primál változók redukált költsége mindig egy megengedett d_j érték, ezeket a változókat kihagyhatjuk a duál megengedettség vizsgálatából. Hasonló a helyzet a korlátos (type-1) változókkal is. Ha ugyanis egy type-1 változóhoz tartozó d_j előjele nem teljesíti (10)-et, akkor a primál változó értékét az ellenkező korlátra állítva d_j duál megengedett lesz. Ez a lépés nem igényel bázischerét, mindössze a primál bázismegoldást kell transzformálni. Ugyanis a korlátos primál változókhoz tartozó d_j kétféleképpen lehet nem-megengedett (infízibilis). Formálisan: legyenek

$$\mathcal{T}^+ = \{j : \text{ type}(x_j) = 1,\, x_j = u_j,\, d_j > 0\}$$

$$\mathcal{T}^- = \{j : \text{ type}(x_j) = 1,\, x_j = 0,\, d_j < 0\}$$

a type-1 duál infízibilis indexek halmazai. Ekkor a primál változók korlátcseréje után a megoldás transzformációja:

$$\beta := \beta - \sum_{j\in\mathcal{T}^+} u_j\alpha_j + \sum_{j\in\mathcal{T}^-} u_j\alpha_j, \tag{11}$$

ahol $\alpha_j = \mathbf{B}^{-1}\mathbf{a}_j$, ':=' értékadást jelent, és a szumma értéke nulla, ha a vonatkozó indexhalmaz üres. α_j-k kiszámítása a (11)-ben szereplő összes j-re nagyon számításigényes lenne. Azonban az egész műveletet egyetlen lépésben el lehet végezni a következő módon:

$$\begin{aligned}\beta &:= \beta - \sum_{j\in\mathcal{T}^+} u_j\alpha_j + \sum_{j\in\mathcal{T}^-} u_j\alpha_j \\ &= \beta - \mathbf{B}^{-1}\Big(\sum_{j\in\mathcal{T}^+} u_j\mathbf{a}_j - \sum_{j\in\mathcal{T}^-} u_j\mathbf{a}_j\Big) \\ &= \beta - \mathbf{B}^{-1}\tilde{\mathbf{a}}\end{aligned} \tag{12}$$

$\tilde{\mathbf{a}}$ nyilvánvaló értelmezésével. $\tilde{\mathbf{a}}$ (egyszerű) meghatározása után csak egyetlen FTRAN műveletet (lásd pl. [5]) kell végrehajtani a bázis inverzével. (12)-t *duál megengedettségi korrekció*nak nevezzük. Ezt a műveletet elég akkor végrehajtani, amikor már sikerült (ha lehet) az összes type-2 és type-3 pozíció duál logikai változóját megengedett értékre hozni.

Fentiek alapján elegendő a type-2 és type-3 változókkal foglalkozni. Ezekre két infízibilitási indexhalmaz határozható meg:

$$\mathcal{M} = \{j : d_j < 0 \text{ és } \mathrm{type}(x_j) \geq 2\}, \tag{13}$$

$$\mathcal{P} = \{j : d_j > 0 \text{ és } \mathrm{type}(x_j) = 3\}. \tag{14}$$

Ezek segítségével a duál infízibilitások összegét a következőképpen definiáljuk:

$$f = \sum_{j \in \mathcal{M}} d_j - \sum_{j \in \mathcal{P}} d_j, \tag{15}$$

ahol bármely szumma egyenlő nullával, ha a vonatkozó indexhalmaz üres. Nyilvánvalóan mindig igaz, hogy $f \leq 0$.

Duál első fázisban a cél f maximalizálása. Ha $f = 0$-t el tudjuk érni, akkor a megoldás duál megengedett lesz egy esetleges megengedettségi korrekció után. Ha ez nem érhető el, akkor a feladatnak nincs duál megengedett megoldása.

A duál algoritmusok előbb a bázisból kilépő változót határozzák meg, ami egyben a pivot sort is definiálja a soronlévő iterációra. Tegyük fel, hogy a p-edik sort választottuk ki, mert egy később tárgyalandó kritérium alapján ez bizonyult a legjobbnak. A szimplex iteráció ennek a transzformált sornak bizonyos t többszörösét vonja le a duál logikai változók sorából: $\mathbf{d}_{\mathcal{R}}(t) = \mathbf{d}_{\mathcal{R}} - t\boldsymbol{\alpha}^p$. Így a duál logikaiak, mint a t függvényei:

$$d_j^{(p)}(t) = d_j - t\alpha_{pj}. \tag{16}$$

Ezzel a jelöléssel $d_j^{(p)}(0) = d_j$. Ha t kellően kicsi úgy, hogy az $\mathcal{M}$ és $\mathcal{P}$ halmazok változatlanok, az infízibilitások összege a t függvényében a következőképpen fejezhető ki:

$$f^{(p)}(t) = \sum_{j \in \mathcal{M}} d_j^{(p)}(t) - \sum_{j \in \mathcal{P}} d_j^{(p)}(t) = f^{(p)}(0) - t\left(\sum_{j \in \mathcal{M}} \alpha_{pj} - \sum_{j \in \mathcal{P}} \alpha_{pj} \right).$$

Látható, hogy a (15)-beli f-et $t = 0$-ra kapjuk meg: $f = f^{(p)}(0)$. Könnyen kimutatható, hogy a soron következő tárgyalás akkor is igaz, ha a halmazok csak a $t = 0$ esetben változatlanok (degeneráció).

A duál infízibilitások megváltozása, ha t elmozdul a nulla értékről:

$$\Delta f = f(t) - f(0) = -t\left(\sum_{j \in \mathcal{M}} \alpha_{pj} - \sum_{j \in \mathcal{P}} \alpha_{pj} \right). \tag{17}$$

Ha bevezetjük a

$$v_p = \sum_{j \in \mathcal{M}} \alpha_{pj} - \sum_{j \in \mathcal{P}} \alpha_{pj} \tag{18}$$

jelölést, (17) a $\Delta f = -tv_p$ alakban írható. Ennek következtében a duál infízibilitások javulásának a követelménye, $\Delta f > 0$, ekvivalens a

$$-tv_p > 0$$

követelménnyel. Ezt pedig kétféleképpen lehet elérni:

(19) Ha $v_p > 0$, akkor $t < 0$-nak kell teljesülni,

ha $v_p < 0$, akkor $t > 0$-nak kell teljesülni.

Mindaddig, amig van olyan i sor hogy type$(\beta_i) \neq 3$ és a hozzá tartozó (18) által definiált v értéke nem nulla, lehetőség van a duál első fázis célfüggvény javítására. A feltételek pontos kimunkálása a későbbiekben történik meg. Amennyiben több sor is potenciális javító, akkor ezek közül valamilyen egyszerű vagy összetett szabály alapján választunk („dual phase-1 pricing").

Jelöljük a p-edik bázisváltozó β_p eredeti indexét $\mathbf{A}$-ban μ-vel. Így $x_\mu = \beta_p$ a kiválasztott kilépő változó. Ezen a ponton kikötjük, hogy a kilépő változó redukált költsége, $\bar{d}_\mu$, duál megengedett legyen a bázisból való kilépés után. Ez nem feltétlenül szükséges, de így a duál megengedettség jobban kézben tartható.

Ha t elmozdul nulláról, egyes d_j-k elmozdulnak a nulla irányába, ami a megengedettségi tartományuk határa, és bizonyos t esetén el is érik azt. Ezek az értékek (16) alapján $d_j - t\alpha_{pj}$-ból határozhatók meg:

$$t_j = \frac{d_j}{\alpha_{pj}} \quad \text{bizonyos } j \in \mathcal{R} \text{ indexekre,}$$

és ezek lehetővé tesznek egy báziscserét, mivel itt $d_j(t)$ nullává válik. Ez azt is jelenti, hogy a j-edik duál feltétel ekkor egyenlőség formában teljesül. Ha a több lehetséges j közül kiválasztott indexet q-val jelöljük és ezt a változót vonjuk be a bázisba x_μ helyett, akkor (a szimplex transzformációs képlete alapján) azt kapjuk, hogy

$$\bar{d}_\mu = -\frac{d_q}{\alpha_{pq}} = -t_q,$$

aminek duál megengedettnek kell lenni a fentiek szerint. $\bar{d}_\mu$ előjelét az határozza meg, hogy x_μ hogyan (alsó vagy felső korláton) hagyja el a bázist. Ez rögtön szabályt is ad arra, hogy a belépő változó hogyan határozható meg, ha a kilépőt már megválasztottuk. Alább a szabály szóbeli formáját adjuk, a részletes bizonyítás [6]-ban található.

1. Ha $v_p > 0$, akkor $t_q < 0$ kell, hogy (19) teljesüljön. Ez maga után vonja, hogy a p-edik bázisváltozónak alsó korláton kell kilépni (ugyanis $\bar{d}_\mu \geq 0$ szükséges a duál megengedettséghez). Duál degeneráció esetétől eltekintve ez azt jelenti, hogy d_q-nak és α_{pq}-nak ellenkező előjelűnek kell lenni, vagyis a potenciális pivot pozícióknak teljesíteni kell ezt a követelményt.

2. Ha $v_p < 0$, akkor $t_q > 0$ szükséges, ami csak úgy lehetséges, ha a kilépő változó β_p ($\equiv x_\mu$) korlátos típusú és a felső korláton lép ki a bázisból. Degenerciótól eltekintve ez azt jelenti, hogy d_q-nak és α_{pq}-nak azonos előjelűnek kell lenni.

3. Ha $v_p \neq 0$ és a kilépő változó 0 típusú (fix), akkor $\bar{d}_\mu$ előjele érdektelen. Ezért ha $v_p > 0$, akkor $t_q < 0$ hányadosokat keresünk, és ha $v_p < 0$, akkor pozitív t-ket vizsgálunk.

Hátra van még annak a vizsgálata, hogy a $\mathbf{v} = [v_1, \ldots, v_m]^T$ vektort hogyan lehet meghatározni. $\mathbf{v}$-t duál *első fázis redukált költség* vektornak nevezzük.

Vektor alakban (18) a következőképpen írható:

$$\mathbf{v} = \sum_{j \in \mathcal{M}} \boldsymbol{\alpha}_j - \sum_{j \in \mathcal{P}} \boldsymbol{\alpha}_j = \mathbf{B}^{-1}\Big(\sum_{j \in \mathcal{M}} \mathbf{a}_j - \sum_{j \in \mathcal{P}} \mathbf{a}_j \Big) = \mathbf{B}^{-1}\tilde{\mathbf{a}} \tag{20}$$

$\tilde{\mathbf{a}}$ nyilvánvaló értelmezése mellett. Ez a számítás egy FTRAN művelet segítségével végezhető el, ami egy standard szimplex technika.

3.2. Az $f(t)$ függvény vizsgálata

Miután a kilépő változót meghatároztuk (20) segítségével, a duál infízibilitások összege t függvényében:

$$f(t) = f(0) - t\Big(\sum_{j \in \mathcal{M}} \alpha_{pj} - \sum_{j \in \mathcal{P}} \alpha_{pj} \Big) = f(0) - tv_p. \tag{21}$$

Látható, hogy $f(t)$ előállításához az $\boldsymbol{\alpha}^p$ transzformált pivot sorra van szükség, amiből kiolvashatók az α_{pj} együtthatók. Ez számítástechnikailag egy költséges művelet.

Az előzményekből következően t elmozdulása nulláról lehet pozitív, illetve negatív irányban is, a helyzettől függően. (21) nyilvánvalóan lineáris függvény t-ben, amíg a $-v_p$ meredekséget definiáló halmazok ($\mathcal{M}$ és $\mathcal{P}$) változatlanok. Bebizonyítható (lásd Maros [6, 5]), hogy

> *$f(t)$ egy szakaszonként lineáris konkáv törtvonal függvény, melynek töréspontjai a belépő változó különböző megválasztásainak felelnek meg és amely pontokban az infízibilitási halmazok ($\mathcal{M}$ és/vagy $\mathcal{P}$) megváltoznak. Ennek megfelelően $f(t)$ a globális maximumát akkor éri el, amikor a meredeksége előjelet vált. Az így definiált báziscsere eredményezi a duál infízibilitások maximális javulását, amit a kiválasztott kilépő változóval el lehet érni.*

Ugyancsak bebizonyítható (lásd Maros [6, 5]), hogy a töréspontokat a következő $j \in \mathcal{R}$ pozíciók definiálják:

1. *A* $t \geq 0$ *esetben*
   $d_j < 0$ *és* $\alpha_{pj} < 0$ *vagy*
   $d_j \geq 0$ *és* $\alpha_{pj} > 0$,
2. *a* $t \leq 0$ *esetben pedig*
   $d_j < 0$ *és* $\alpha_{pj} > 0$ *vagy*
   $d_j \geq 0$ *és* $\alpha_{pj} < 0$.

A második eset közvetlenül származtatható az elsőből úgy, hogy $-\alpha_{pj}$*-t használunk* α_{pj} *helyett. Mindkét esetben egy további lehetőség az, hogy ha* $\text{type}(x_j) = 3$ *(szabad változó) és* $d_j \neq 0$*, akkor a* d_j/α_{pj}, $(\alpha_{pj} \neq 0)$ *töréspont multiplicitása* 2.

Ha a definiált töréspontokat nagyság szerint sorba rendezzük, akkor könnyen követhető, hogy $f(t)$ meredeksége akkor változik, amikor t eléri a legkisebb definiált hányadost (töréspontot). A rendezett értékek a $t \geq 0$ esetben $0 \leq t_1 \leq \cdots \leq t_Q$, ha Q-val jelöljük az összes definiált töréspontok számát, míg a $t \leq 0$ esetben fordított a sorrend $t_Q \leq \cdots \leq t_1 \leq 0$, vagy ezzel egyenértékűen, $0 \leq -t_1 \leq \cdots \leq -t_Q$, illetve a közös alak $0 \leq |t_1| \leq \cdots \leq |t_Q|$.

Ha sikeres volt p megválasztása (van kilépő) változó, akkor $Q > 0$. Bebizonyítható (lásd Maros [6, 5]), hogy

Akár a $v_p < 0$ $(t > 0)$*, akár a* $v_p > 0$ $(t < 0)$ *esetről van szó,* $f(t)$ *kezdeti meredeksége* $s_p^0 = |v_p|$*, és a lineáris szakasz meredeksége a* k*-adik töréspont után*

$$s_p^k = s_p^{k-1} - |\alpha_{pj_k}|\,, \text{ for } k = 1, \ldots, Q.$$

Ezen elméleti alapokon nyugszik a GDPO algoritmus, amelyet a következő szakaszban mutatunk be részletesen.

$f(t)$ jellegzetes alakja az 1. ábrán látható.

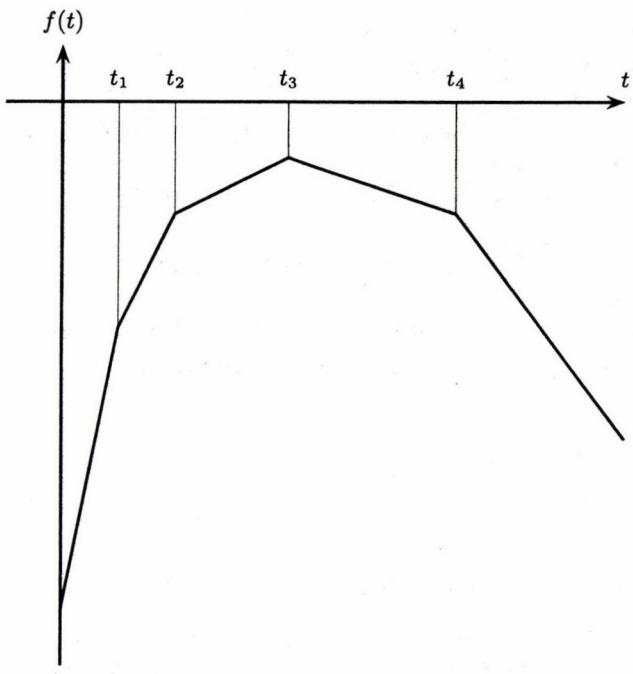
3.3. A GDPO algoritmus

Az alábbiakban a GDPO algoritmus egy iterációját definiáljuk a 3.1. és 3.2. szakaszokban tárgyaltakra alapozva.

Legyen $t_0 = 0$ és $f_i = f(t_i)$. Belátható, hogy a duál infízibiliások összege a töréspontokban a következő rekurzióval számítható ki: $f_k = f_{k-1} + s_p^{k-1}(t_k - t_{k-1})$, $k = 1, \ldots, Q$.

A duál első fázis egy iterációjának a lépései:

1. Lépés. Határozzuk meg az $\mathcal{M}$ és $\mathcal{P}$ halmazokat (13) és (14) alapján. Ha mindkettő üres, hajtsunk végre megengedettségi korrekciót. Ezután a bázis duál megengedett, az algoritmus befejeződik.



1. ábra. Duál infízibilitások mértéke (összege) t függvényében

2. Lépés. Állítsuk fel az $\tilde{\mathbf{a}} = \sum_{j \in \mathcal{M}} \mathbf{a}_j - \sum_{j \in \mathcal{P}} \mathbf{a}_j$ segédvektort (20) szerint.

3. Lépés. Határozzuk meg a duál redukált költségek vektorát: $\mathbf{v} = \mathbf{B}^{-1}\tilde{\mathbf{a}}$ (20) alapján.

4. Lépés. Kilépő vektor meghatározása: Válasszunk ki egy jelöltet a $\mathbf{v}$-ből valamilyen (egyszerű [pl. Dantzig], vagy összetett [pl. Devex]) szabály alapján. Jelöljük ezt a pozíciót p-vel. Ez lesz egyben a pivot sor indexe is.

Ha nem találunk jelöltet, akkor a duál feladatnak nincs megengedett megoldása, az algoritmus befejeződik.

5. Lépés. Határozzuk meg $\mathbf{B}^{-1}$ p-edik sorát: $\boldsymbol{\rho}^T = \mathbf{e}_p^T\mathbf{B}^{-1}$ és ennek segítségével $\mathbf{A}$ p-edik sorának transzformált alakját: $\alpha_{pj} = \boldsymbol{\rho}^T\mathbf{a}_j$ a $j \in \mathcal{R}$ pozíciókban.

6. Lépés. Duál hányadosteszt. Határozzuk meg és tároljuk a duál hányadosokat a 3.2. szakaszban leírtak alapján a $v_p > 0$, illetve $v_p < 0$ esetnek megfelelően. Rendezzük a hányadosokat (töréspontokat): $0 \leq |t_1| \leq \cdots \leq |t_Q|$.

7. Lépés. $f(t)$ maximalizálása.

Legyen $k = 0$, $t_0 = 0$, $f_0 = f(0)$, $s_p^0 = |v_p|$.
While $k < Q \wedge s_p^k > 0$ **do**
$k := k + 1$
Jelölje j_k azt az oszlopindexet, amelyik $|t_k|$-t, a k-adik legkisebb hányadost (töréspontot) határozta meg.
Noha algoritmikusan nem szükséges, a teljesség kedvéért számítsuk ki $f(t)$-t ebben a pontban: $f_k = f_{k-1} + s_p^{k-1}(t_k - t_{k-1})$,
Határozzuk meg $f(t)$ meredekségét ezután a pont után: $s_p^k = s_p^{k-1} - |\alpha_{pj_k}|$.
end while
Jelölje q az utolsó töréspontnak az indexét, amelyre a s_p^k meredekség még pozitív, $q = j_k$. $f(t)$ a maximumát ennél a töréspontnál éri el. A belépő változó x_q.

8. Lépés. Új megoldás meghatározása:

Ha x_μ felső korláton lép ki a bázisból ($v_p < 0$ eset), $\bar{x}_\mu = u_\mu$, különben ($v_p > 0$ eset) $\bar{x}_\mu = 0$.

Határozzuk meg a belépő oszlopvektor transzformált alakját: $\boldsymbol{\alpha}_q = \mathbf{B}^{-1}\mathbf{a}_q$.

Legyen $\theta_P = \beta_p/\alpha_{pq}$, ha $v_p > 0$, vagy $\theta_P = (\beta_p - \omega_p)/\alpha_{pq}$ (ω_p a p-edik bázisváltozó egyedi felső korlátja), ha $v_p < 0$. Transzformáljuk a megoldást:

$\bar{\beta}_p = x_q + \theta_P$ és $\bar{\beta}_i = \beta_i - \theta_P \alpha_{iq}$ minden $i \neq p$.

Transzformáljuk a bázison kívüli változókhoz tartozó duál logikai változókat:

$\bar{d}_\mu = -\dfrac{d_q}{\alpha_{pq}}$ és $\bar{d}_j = d_j + \bar{d}_\mu \alpha_{pj}$ minden $j \neq \mu$.

Módosítsuk a $\mathcal{B}$ és ha szükséges, az $\mathcal{U}$ halmazt, hogy tükrözzék a báziscserét.

Térjünk vissza az 1. Lépésre.

4. A GDPO algoritmus elméleti vizsgálata

Ebben a szakaszban először a GDPO algoritmus helyességét mutatjuk meg, majd néhány fontos tulajdonságát elemezzük.

4.1. Az algoritmus helyessége

Először is, a 3.2. szakaszban kiderült, hogy $f(t) = f(0) - t\Big(\sum_{j\in\mathcal{M}} \alpha_{pj} - \sum_{j\in\mathcal{P}} \alpha_{pj}\Big)$ egy szakaszonként lineáris konkáv törtvonal függvény, amit minden iteráció során újra definiálunk és maximalizálunk.

Másodszor, miután az $f(t)$ függvénynek 0 a felső korlátja, a GDPO során megoldott LP feladat *megoldása mindig korlátos.*

Harmadszor, $f(t)$ maximalizálása a duál megengedettségi feltételek mellett egy konvex probléma. Ezért ha nincs több javító sor és a megoldás még mindig nem duál megengedett, akkor feladatnak *nincs duál megengedett megoldása.*

Negyedszer, ha az algoritmus minden iterációban pozitív lépést tesz a duál megengedettség felé ($f(t)$ szigorúan monoton növekszik), akkor egyetlen bázis sem ismétlődhet, tehát ciklizálás nem fordulhat elő, így az *algoritmus véges számú lépés után befejeződik* vagy az 1., vagy a 4. Lépésben. Ha az iterációk során degeneráció lép fel, és a GDPO is csak degenerált lépést tudna végrehajtani (lásd még 4.2.4. szakaszt), akkor például az elméletileg garantált (és számítástechnikailag hatékony) Wolfe féle „ad hoc" módszert ([9]) lehet használni, amíg a degenerációból kikerül az algoritmus.

4.2. A GDPO néhány fontos tulajdonsága

4.2.1. A hagyományos eljárás általánosítása. A GDPO az $f(t)$ függvény maximalizálásával a lehető legnagyobb lépést teszi meg a duál megengedettség felé, amit a kiválasztott kilépő változó esetén el lehet érni. Könnyen látható, hogy a *hagyományos duál (HD) első fázis eljárás ennek speciális esete*, amikoris $f(t)$-nek csak az első töréspontjáig megyünk el. A másik oldalról nézve *GDPO a HD eljárás általánosításának* tekinthető.

4.2.2. A pivot sor többszörös hasznosítása. HD a „drágán" előállított transzformált pivot soron egyetlen iterációt végez el és utána eldobja azt. GDPO ezt a sort többszörösen hasznosítja, ami azt eredményezi, hogy egyetlen GDPO iterációval akár nagyon sok HD iterációnyi előrehaladást képes elérni. Mindez attól függ, hogy $f(t)$ maximumát hányadik töréspontnál éri el.

4.2.3. Nagyobb numerikus stabilitás. A szimplex implementációk numerikus problémáinak nagy része abból adódik, hogy abszolút értékben túl kicsinek adódik az α_{pq} pivot elem. GDPO esetén ezen könnyű segíteni, ha $f(t)$ maximuma nem az első töréspontón éretik el. Ekkor ugyanis lehetőség van az eggyel korábbi töréspontot venni. Ha az ehhez tartozó pivot elem még mindig nem megfelelő nagyságrendű, akkor fokozatosan lehet visszalépni a töréspontokon egy jó pivot elem megtalálásáig (ha egyáltalán van ilyen). Nagy valószínűséggel GDPO még ilyen visszalépéses esetben is nagyobb előrehaladást tud biztosítani, mint HD. Ezen túlmenően, ha az első töréspont pivot eleme „rossz", akkor HD nem is tud mit csinálni ezzel a pivot sorral, míg GDPO-nak van esélye egy jó iterációra, ha $f(t)$ a maximumát nem az első töréspontón éri el.

4.2.4. Jobb hatásfok degeneráció esetén. Duál degeneráció esetén egy vagy több bázison kívüli változóra $d_j = 0$. Ha ezek a pozíciók részt vesznek a duál hányados tesztben, akkor ezek 0 értékű hányadosokat definiálnak. Ez a (esetleg többszörös multiplicitással rendelkező) $t = 0$ töréspontot eredményezi. HD ezek közül választ egyet, és egy 0 lépéshosszú, nem-javító iterációt hajt végre.

Ezzel szemben GDPO esetén a helyzet a következő. Tegyük fel, hogy $t = 0$ multiplicitása ℓ, vagyis

$$0 = |t_1| = \cdots = |t_\ell| < |t_{\ell+1}| \leq \cdots \leq |t_Q|. \tag{22}$$

Jelöljék $\alpha_{j_1}, \ldots, \alpha_{j_\ell}, \ldots, \alpha_{j_Q}$ a megfelelő pozíciókban a transzformált pivot sor elemeit, az egyszerűség kedvéért elhagyva p-t a felső indexből. $f(t)$ maximumát definiáló töréspont k indexét az $s_k > 0$ és $s_{k+1} \leq 0$ relációk határozzák meg, ami részletesen:

$$s_k = s_0 - \sum_{i=1}^{k} |\alpha_{j_i}| > 0 \quad \text{és} \quad s_{k+1} = s_0 - \sum_{i=1}^{k+1} |\alpha_{j_i}| \leq 0.$$

Ha erre a k-ra $k > \ell$ teljesül, akkor (22) által azt kapjuk, hogy $|t_k| > 0$. Ez pedig azt jelenti, hogy a degeneráció ellenére *GDPO pozitív lépést tud tenni* a duál megengedettség felé. Ha $k \leq \ell$, akkor a lépés degenerált (0 lépéshosszú) lesz.

4.2.5. Egy iterációra eső számítási munka. GDPO egy iterációja valamivel, de általában nem sokkal több számítási munkát igényel, mint HD.

1. Hányados teszt: ugyanaz, mint a HD-nál. A különbség annyi, hogy GDPO esetén a hányadosokat ($f(t)$ töréspontjait) tárolni kell. Ez többlet memóriaigényt jelent, általában nem több, mint $n - m$ dupla pontosságú szám tárolását. Noha a type-3 pozíciók két töréspontot is meghatározhatnak, az ilyen változók száma kevés szokott lenni az LP modellekben.

2. Míg HD a legkisebb hányadost (első töréspontot) választja, addig GDPO számára a t_k töréspontokat nagyság szerint sorba kell rendezni. Ha sok töréspont definiálódik, akkor ez jelentős többletmunkát igényel, hiszen a rendezést minden iteráció során újból el kell végezni. Általában azonban nincs szükség az összes t_k-ra $f(t)$ maximumának a meghatározásánál. Így célszerű olyan rendezési módszert használni, amelyiknél az i-edik lépésben az első i elem helyes sorrendben van. A legmegfelelőbb rendezési módszer megválasztását nehezíti az, hogy a definiált és felhasznált töréspontok száma iterációról iterációra drasztikusan különböző lehet. Vizsgálataink azt mutatták, hogy a heap-sort módszer adaptációja megbízhatóan, jó hatékonysággal képes ezt a feladatot ellátni.

3. Tapasztalataink szerint a fenti többletmunka GDPO iterációs sebességét alig érzékelhetően befolyásolja HD-hez képest.

5. A GDPO algoritmus számítástechnikai vizsgálata

A GDPO tulajdonságainak elméleti vizsgálata során többször szerepelt olyan kitétel, hogy az igazán jó tulajdonságok akkor jönnek elő, ha valóban több töréspont definiálódik iterációnként és nem az elsőn éretik el $f(t)$ maximuma, mert így az algoritmusban rejlő nagyfokú rugalmasság jobban kihasználható. Az, hogy ilyen helyzet mennyire fordul elő, és ezáltal mekkora az algoritmus jelentősége, elsősorban gyakorlati tapasztalatok alapján derül ki. Ebben a szakaszban a szakmában elfogadott tesztfeladatokkal végzett számítások során nyert tapasztalatokról számolunk be. Miután GDPO egy lokális stratégiát valósít meg, elméleti úton nem lehet globális teljesítményt garantálni. Ezért is fontos a számítástechnikai vizsgálat elvégzése.

Az irodalomban nem ismeretes különféle duál szimplex algoritmusokkal kapcsolatos számítástechnikai tapasztalatok publikálása. Kommerciális LP rendszerekkel való összehasonlításnak nem lett volna értelme, mert azok algoritmikus háttere ismeretlen, publikálatlan üzleti titok, ezért algoritmikus összehasonlításra alkalmatlanok. Így GDPO-nak az első töréspontón alapuló hagyományos duál első fázissal való összehasonlításához folyamodtunk.

5.1. A tesztfeladatok jellemzői

A tesztelés célja GDPO algoritmikus *hatásosság*ának a vizsgálata. A tesztelésre a korábban a szerző által kifejlesztett, szimplex alapú MILP nevű kísérleti kódot használtuk. Noha MILP primál orientáltságú, tartalmazza azon eszközök nagy részét, amik a duál implementációjához szükségesek.

MILP a nagyméretű LP feladatok hatékony megoldására szolgáló algoritmikus és számítástechnikai elemek kipróbálására készült. Így ez egy kísérleti kód, amely tele van tűzdelve olyan utasításokkal, amelyek a működést figyelik és információt gyűjtenek. Ezáltal nagyon alkalmas új eljárások tulajdonságainak vizsgálatára, amikor is a hatásosság a kérdés. Mindezek mellett MILP működése igen hatékony, noha a „numerikus kernel"-je kb. 10–12 évvel ezelőtt készült (ami az akkori igényeket biztonságosan kielégítette). Mindezt azért szükséges megjegyezni, mert bár ez az esetek nagy részében jól ki tudta szolgálni a duál ismert nagyobb igényét a számítási pontosságra, de egy-két feladatnál ez nem bizonyult elegendőnek. Hangsúlyozni kell, hogy ez nem a GDPO algoritmikus hiányosságát jelenti, hiszen nem annak hibás működéséről volt szó, hanem pl. bizonyos mennyiségek kétféle kiszámítása közt adódó numerikus különbségről és az ebből esetleg bekövetkező kisebb visszaesésről. Noha MILP/Dual ilyenkor kijavítja a numerikus hibát néhány többlet iteráció árán, ez azonban eltorzítja az iterációs statisztikát. Az alább ismertetett tesztelésben olyan feladatok vettek részt, amelyek esetén MILP/Dual numerikus kernelje hibátlanul működött, mert így a különbségek feketén-fehéren a megoldó algoritmusok közti különbségből adódnak és ezáltal alkalmasak GDPO hatásosságának kiértékelésére. *A hatásosságot a duál első fázisban megtett iterációk számával*

mérjük. Ennek azért is van értelme, mert GDPO és HD iterációs sebessége alig mérhetően tér el egymástól.

Miután a type-0 és type-1 változók kezelése a duál első fázisban triviális (lásd duál megengedettségi korrekció), olyan feladatokat választottunk, amelyeknél a type-2 (nem-negatív) és type-3 (szabad) változók vannak túlsúlyban. Ezen túlmenően az is szempont, hogy a futási eredmények reprodukálhatók, illetve ellenőrizhetők legyenek. Ennek érdekében a használt feladatokat a szakmában elfogadott tesztfeladattárakból vettük. MILP két változatát használtuk a futások során. Az egyik (`m503d`) a GDPO, míg a másik (`m503d1`) a hagyományos duál (HD) implementálását tartalmazza. Az utóbbi valójában az előzőnek egy olyan változata, ahol az első töréspont alapján történik a báziscsere meghatározása (legkisebb duál hányados, $k = 1$). Összesen 48 feladat szerepel a vizsgálatokban. Köztük van kisebb, közepes és nagyméretű is. Ezek valós életből vett problémák, nem pedig generált feladatok.

Az 1. táblázat a tesztelésben résztvett feladatok főbb jellemzőit tartalmazza, amelyek: feltételek száma (m), strukturális változók száma ($\bar{n}$), nem-nullák száma **A**-ban, valamint a strukturális változók számának megoszlása típus szerint.

5.2. A tesztfutások eredményei és kiértékelése

Az eredményeket először összevont táblázatos formában mutatjuk be, majd a levont következtetéseknek megfelelő további táblázatok következnek.

A 2. táblázatban a duál első fázisban végzett iterációk számát mutatjuk be GDPO és HD esetén, továbbá a megoldás során használt stratégiát: volt-e normálás (általában igen), mi volt az induló bázis (logikai vagy crash), illetve a pivot sor meghatározása Devex technikával történt-e vagy sem. Nagy érdeklődésre tarthat számot az iteráció arányokat feltüntető két oszlop. A H/G jelentése az, hogy a HD algoritmus hányszor többet iterált, mint GDPO, a G/H oszlop ennek a reciproka. Látható, hogy nem minden feladat esetén voltak a futási paraméterek azonosak. Ennek több oka van. Először is, olyan induló bázist kellett választani, ami duál nem-megengedett. Bizonyos esetekben az egységvektorokból álló logikai bázisra ez teljesült, és amikor nem, akkor egy „crash" eljárással előállított bázis (lásd [5]) felelt meg a célnak. Másodszor, a nagyobb feladatok sorkiválasztására a duál Devex technikát használtuk a racionálisabb megoldási idő érdekében. Harmadszor, a feladatok legtöbbjét normálással oldottuk meg, hogy numerikus problémák ne lépjenek fel és az eredmények „tisztán" legyenek értelmezhetők az algoritmusok hatásossága szempontjából. Nagyon fontos megjegyezni, hogy minden konkrét feladat esetén ugyanazzal a stratégiával futott mindkét (`m503d` és `m503d1`) program. Ezért a megoldási stratégia gyanánt ezek a közös beállítások vannak feltüntetve.

A GDPO elvárt hatásos működésének az alapja a pivot sor többszörös felhasználása, ami abban nyilvánul meg, hogy $f(t)$ maximalizálására hány töréspontot használ fel. Ezt egy bizonyos értelemben *algoritmikus lépéshossz*nak lehet nevezni. A 3. táblázat erről ad képet. Miután nagyon nagy a variáció, az adatokat kissé tömöríteni kellett, hogy be lehessen mutatni. A táblázat egy sora azt mondja meg, hogy az összes duál első fázis iteráció során (ez a szám az utolsó oszlopban van)

Feladat	Sorok	Oszlopok	Nem-nullák	Változók száma típus szerint			
				Type-0	Type-1	Type-2	Type-3
25fv47	822	1571	11127	0	0	1571	0
80bau3b	2262	9799	29063	498	2986	6315	0
agg	488	163	2541	0	0	163	0
agg2	516	302	4515	0	0	302	0
agg3	516	302	4531	0	0	302	0
baxter	27441	15128	109823	0	1122	14006	0
bnl1	643	1175	6129	0	0	1175	0
bnl2	2325	3489	16124	0	0	3489	0
boeing1	351	384	3865	0	228	156	0
cre_a	3516	4067	19054	0	0	4067	0
cre_b	9648	72447	328542	0	0	72447	0
cre_c	3068	3678	16922	0	0	3678	0
cre_d	8926	69980	312626	0	0	69980	0
czprob	929	3523	14173	229	0	3294	0
d6cube	415	6184	43888	0	0	6184	0
dbir2	18906	27355	1148847	0	0	27355	0
degen2	444	534	4449	0	0	534	0
degen3	1504	1818	26230	0	0	1818	0
degen4	4420	6711	107375	0	0	6711	0
grow07	140	301	2633	0	280	21	0
grow15	300	645	5665	0	600	45	0
grow22	440	946	8318	0	880	66	0
israel	174	142	2358	0	0	142	0
maros	847	1443	10006	35	0	1408	0
mod011	4481	10958	37425	1	1596	9361	0
nsct2	23003	14981	686396	0	0	14981	0
osa-07	1118	23949	167643	0	0	23949	0
osa-14	2337	52460	367220	0	0	52460	0
osa-30	4350	100024	700160	0	0	100024	0
perold	625	1376	6026	64	266	958	88
pilot_we	723	2789	9218	78	294	2337	80
rentacar	6804	9557	42019	650	179	8728	0
scagr_2	32847	34580	141757	0	0	34580	0
scrs_3	16545	17420	71401	0	0	17420	0
scsd6	147	1350	5666	0	0	1350	0
scsd8	397	2750	11334	0	0	2750	0
sctap2	1090	1880	8124	0	0	1880	0
sctap3	1480	2480	19734	0	0	2480	0
ship08l	778	4283	17085	0	0	4283	0
ship12l	1151	5427	21597	0	0	5427	0
stair	357	467	3857	82	6	373	6
sto27	14441	34114	114973	0	0	34114	0
stocfor2	2157	2031	9492	0	0	2031	0
stocfor3	16676	15695	74004	0	0	15695	0
sws	14310	12465	105480	0	0	12465	0
unicolns	5421	45569	168220	2	1449	44118	0
woodlp	244	2594	70216	0	0	2594	0
woodw	1098	8405	37478	0	0	8405	0

1. táblázat

Feladat	Induló duál inf. száma	Duál Ph-1 it. száma		Iteráció arányok		Megoldási stratégia		
		GDPO	HD ($k=1$)	H/G	G/H	Normálás I/N	Ind. bázis CB/LB	Devex I/N
25fv47	41	238	1033	4.34	0.23	I	LB	N
80bau3b	208	736	997	1.35	0.74	I	LB	I
agg	95	11	107	9.73	0.10	I	LB	N
agg2	171	13	109	8.38	0.12	I	LB	N
agg3	171	13	109	8.38	0.12	I	LB	N
baxter	3259	2687	4482	1.67	0.60	I	CB	I
bnl1	57	4	27	6.75	0.15	I	LB	I
bnl2	156	49	134	2.73	0.36	I	LB	I
boeing1	164	9	102	11.33	0.09	I	LB	N
cre_a	1156	476	1655	3.48	0.29	N	CB	I
cre_b	14503	4604	12281	2.67	0.37	N	CB	I
cre_c	1056	532	1559	2.93	0.34	N	CB	I
cre_d	10409	3479	14798	4.25	0.23	N	CB	I
czprob	1521	191	1959	10.26	0.10	I	LB	N
d6cube	2637	1209	6500	5.38	0.19	I	CB	I
dbir2	9210	9631	9848	1.02	0.98	I	LB	I
degen2	425	143	574	4.01	0.25	I	LB	I
degen3	1249	571	1945	3.41	0.29	I	LB	I
degen4	2697	1177	6792	5.77	0.17	I	LB	I
grow07	21	1	14	14.00	0.07	I	CB	N
grow15	45	1	14	14.00	0.07	I	CB	N
grow22	66	1	14	14.00	0.07	I	CB	N
israel	24	1	24	24.00	0.04	I	LB	N
maros	162	666	799	1.20	0.83	I	LB	N
mod011	4343	602	3753	6.23	0.16	I	CB	I
nsct2	11240	11507	11636	1.01	0.99	I	LB	I
osa-07	9201	114	3456	30.32	0.03	I	CB	I
osa-14	19695	141	3616	25.65	0.04	I	CB	I
osa-30	37495	68	7785	114.49	0.01	I	CB	I
perold	7	725	587	0.81	1.24	I	LB	N
pilot_we	91	580	1065	1.84	0.54	I	LB	N
rentacar	2	1778	1629	0.92	1.09	I	LB	N
scagr_2	8645	16676	27473	1.65	0.61	I	LB	I
scrs_3	4355	11635	12765	1.10	0.91	I	LB	I
scsd6	218	46	147	3.20	0.31	I	CB	N
scsd8	353	6	30	5.00	0.20	I	CB	N
sctap2	238	442	578	1.31	0.76	I	CB	I
sctap3	315	532	714	1.34	0.75	I	CB	I
ship08l	581	39	587	15.05	0.07	I	CB	N
ship12l	708	51	958	18.78	0.05	I	CB	N
stair	1	180	152	0.84	1.18	I	LB	N
sto27	11541	4879	6844	1.40	0.71	I	CB	I
stocfor2	639	1366	1552	1.14	0.88	I	LB	N
stocfor3	5077	10620	11848	1.12	0.90	I	LB	N
sws	2190	988	1694	1.71	0.58	I	CB	I
unicolns	43914	4975	50841	10.22	0.10	I	CB	I
woodlp	1057	30	1288	42.93	0.02	I	CB	N
woodw	1738	60	2520	42.00	0.02	I	CB	N

2. táblázat

	Felhasznált töréspontok száma										Iterációk
Feladat	1	2	3	4	5	6–10	11–20	21–50	50+	Max	Ph1-ben
25fv47	80	76	38	16	7	17	4	–	–	18	238
80bau3b	389	165	66	41	29	41	5	–	–	14	736
agg	–	1	–	2	–	6	2	–	–	18	11
agg2	1	1	–	3	1	4	–	3	–	48	13
agg3	1	1	–	3	1	4	–	3	–	48	13
baxter	1381	331	157	142	74	201	188	178	35	194	2687
bnl1	–	–	–	–	–	–	4	–	–	17	4
bnl2	17	18	4	–	–	–	10	–	–	19	49
boeing1	1	–	–	–	2	3	2	–	1	136	9
cre_a	79	62	39	27	30	67	79	57	36	474	476
cre_b	103	179	169	203	171	702	287	875	1915	3538	4604
cre_c	102	93	62	49	20	89	49	47	21	397	532
cre_d	108	105	151	109	133	548	656	733	936	3948	3479
czprob	46	71	32	12	8	5	2	4	11	172	191
d6cube	103	90	83	81	92	300	231	139	90	821	1209
dbir2	8851	587	130	34	22	6	1	–	–	13	9631
degen2	19	34	64	7	5	11	–	1	2	58	143
degen3	128	131	247	23	11	14	11	3	3	91	571
degen4	312	165	189	109	111	185	90	10	6	162	1177
grow07	–	–	–	–	–	–	–	1	–	21	1
grow15	–	–	–	–	–	–	–	1	–	45	1
grow22	–	–	–	–	–	–	–	–	1	66	1
israel	–	–	–	–	–	–	–	1	–	24	1
maros	258	200	97	44	24	34	8	1	–	32	666
mod011	260	107	64	46	31	52	17	10	15	917	602
nsct2	10974	298	85	35	19	59	31	6	–	26	11507
osa-07	37	24	3	6	1	2	4	5	32	4503	114
osa-14	70	18	4	2	1	2	3	6	39	9454	145
osa-30	18	3	4	4	2	2	1	6	39	3654	79
perold	414	156	47	18	21	38	15	14	2	154	725
pilot_we	341	84	39	16	11	42	36	9	2	103	580
rentacar	1737	33	6	–	–	1	1	–	–	11	1778
scagr_2	10625	5186	–	865	–	–	–	–	–	4	16676
scrs_3	8311	2995	298	23	8	–	–	–	–	5	11635
scsd6	2	2	4	2	2	6	14	10	4	100	46
scsd8	–	–	–	1	–	1	–	1	3	260	6
sctap2	65	98	58	64	26	83	28	17	3	55	442
sctap3	103	124	46	59	39	87	39	32	3	90	532
ship08	16	2	–	1	12	–	–	–	8	73	39
ship12	13	6	9	9	1	1	–	–	12	62	51
stair	147	30	1	–	1	–	1	–	–	14	180
sto27	612	725	916	520	484	1006	424	191	1	56	4879
stocfor2	583	450	179	99	30	25	–	–	–	10	1366
stocfor3	3730	3491	1604	761	443	546	44	1	–	21	10620
sws	367	332	12	85	31	71	70	–	–	17	988
unicolns	441	627	225	359	96	348	2659	220	–	34	4975
wood1p	–	–	–	–	–	–	5	8	17	588	30
woodw	–	2	–	2	4	7	7	15	23	801	60

3. táblázat

hányszor maximalizálta $f(t)$-t az első, a második, ..., az ötödik töréspont, illetve hányszor volt a maximalizáló töréspont 6 és 10, 11 és 20, 21 és 50 között, illetve 50-nél több. Az összevont része a táblázatnak sok érdekes esetet eltakar (főleg az 50+ részben). Ennek részbeni ellensúlyozására szerepel a „Max" feliratú oszlop, amelyik azt tünteti fel, hogy az összes iteráció során mennyi volt az egy lépésben felhasznált töréspontok maximális száma. Ha például megnézzük a `mod011` sorát, akkor azt látjuk, hogy az első fázisban megtett 602 iterációból (utolsó oszlop) 31 esetben fordult elő, hogy $f(t)$ az ötödik töréspontnál érte el a maximumát, továbbá volt olyan eset (Max), amikor a 917. töréspontig kellett elmenni a maximum eléréséhez.

A kiértékelés első szempontja az algoritmus helyességének az igazolása. Noha ez elméleti úton már megtörtént a 4.1. szakaszban, a kísérletek során szerzett tapasztalatok is azt mutatják, hogy GDPO képes helyesen megoldani a feladatokat.

GDPO hatásosságnak a kiértékeléséhez a 2. táblázatból indulunk ki. Már az első átnézés alapján látszik, hogy GDPO három eset kivételével jobb, mint HD. A H/G oszlop azt mutatja, hogy HD hányszor több iterációt végzett a duál megengedettség eléréséig, mint GDPO. Az 1-nél nagyobb számok jelzik azokat az eseteket, amikor GDPO volt a jobb, és ez a szám egyben a hatásosság mérőszáma is. Három esetben ez a szám 1-nél valamivel kisebb, jelezve, hogy ekkor HD volt a hatásosabb. A jobb áttekinthetőség érdekében elkészítettük a 4. sz. (tömörített) táblázatot annak a bemutatására, hogy GDPO hány esetben és hányszorosan volt jobb HD-nél.

Optimalizáló algoritmusok esetén 25% javulást általában jelentősnek szoktak minősíteni. Ha GDPO esetén csak a legalább 50%-os javulást tekintjük, akkor azt látjuk, hogy a 48 feladatból 35 esik ebbe a kategóriába (lásd 4. sz. táblázat). Feltűnő, hogy ezen belül 13 esetben a hatásosság több mint 10-szeresére növekedett. Különösen az `osa` feladatcsalád megoldása látszik kiemelkedő teljesítménynek, ahol a legjobb esetben (`osa-30`) a javulás 114-szeres.

Az első töréspontot használó algoritmusok (mint amilyen HD is) a duál nem-megengedettségek számát általában csak egyesével tudják csökkenteni, kivéve amikor duál degeneráció miatt ennél többet sikerül egy lépésben elérni. Bár GDPO csak a nem-megengedettségek összegében monoton, mégis képes a nem-megengedettségek számát is igen gyorsan csökkenteni. Az ebből a szempontból kiemelkedően jó példákat az 5. sz. táblázatban foglaltuk össze (összesen 20 feladat).

A 3. sz. táblázat azt tanúsítja, hogy $f(t)$ aktívan használja a definiált töréspontokat. Sőt, ennél több is kiderül. Elméletileg a lehető legjobb teljesítmény az, ha az összes nem-megengedettséget egyetlen iterációval ki lehet küszöbölni. A táblázatból látható, hogy valódi feladatokon GDPO ezt el is tudja érni. Ezek a feladatok: a `grow` család (`grow07`, `grow15` és `grow22`), valamint `israel`. A `grow` feladatokban viszonylag kevés type-2 változó van (21, 45 és 66), azonban az ezekhez tartozó összes duál logikai változó duál nem-megengedett a crash bázis esetén. Az első iterációban definiált $f(t)$ maximumát az összes (21, 45 és 66 [annyi, mint a type-2 változók száma]) definiált töréspont felhasználásával érte el, és ezáltal mindegyik duál logikai változó megengedett értékre került egyetlen lépésben. `israel` esetén az összes változó 2-es típusú, de itt is a maximális hatásossággal működött GDPO.

Javulás	Hány esetben
1.0 – 1.5-szeres	10
1.6 – 3.0-szoros	7
3.1 – 5.0-szörös	7
5.1 – 10.0-szeres	8
Több mint 10-szeres	13
Romlás	
0.8 – 1.0-szeres	3
Összes	48

4. táblázat. GDPO hatásossága az iterációk számának arányában mérve

Feladat	Induló duál inf. száma	GDPO iterációk
agg	95	11
agg2	171	13
agg3	171	13
bnl1	57	4
boeing1	164	9
grow07	21	1
grow15	45	1
grow22	66	1
israel	24	1
mod011	4343	602
osa-07	9201	114
osa-14	19695	141
osa-30	37495	68
scsd6	218	46
scsd8	353	6
ship08l	581	39
ship12l	708	51
unicolns	43914	4975
wood1p	1057	30
woodw	1738	60

5. táblázat. Duál infízibilitások számának különösen gyors eliminálása

6. Következtetések

A cikk célja a szimplex módszerre kidolgozott és GDPO-nak nevezett duál első fázis algoritmus [6, 5] elméleti és számítástechnikai elemzése, az algoritmus tulajdonságainak vizsgálata volt. Először magát az algoritmust mutattuk be röviden, hogy az elméleti tulajdonságok tárgyalása önmagában is érthető legyen. Ezután tértünk rá a számítástechnikai vizsgálatra, amit intenzív kísérleti munka előzött meg. A kísérleteket a nemzetközileg elfogadott valódi életből vett tesztfeladatokon végeztük el. Ezek méretben és komplexitásban széles kört ölelnek fel. GDPO-t összehasonlítottuk egy tipikus hagyományos duál első fázis algoritmussal. Kommerciális LP rendszerekkel való összehasonlításnak nem lett volna értelme a korábban már említett okok miatt.

A GDPO-val kapcsolatos elméleti és számítástechnikai vizsgálatok tapasztalatait az alábbiakban lehet összefoglalni.

1. GDPO a hagyományos duál első fázis algoritmusokat speciális esetként tartalmazza, és így azok általánosításának tekinthető.

2. GDPO a transzformált pivot sort többszörösen képes használni, ami által egy iterációban sok hagyományos iterációnak megfelelő előrehaladást képes elérni.

3. GDPO csak az infízibilitások összegében monoton, ami nagy rugalmasságot biztosít és lehetővé teszi megfelelő nagyságrendű pivot elem kiválasztását. Ez pedig kedvező numerikus tulajdonságot eredményez.

4. Duál degeneráció esetén GDPO-nak sokkal nagyobb esélye van nem-degenerált iterációt végrehajtani, mint más duál algoritmusoknak.

5. GDPO jól implementálható, és az iterációs sebesség alig érzékelhetően változik HD-vel szemben, ha a számítástudomány korszerű módszereit használjuk.

6. GDPO előnyös elméleti tulajdonságai a gyakorlatban rendszeresen érvényesülnek.

7. GDPO hatásosság tekintetében szinte mindig jelentősen felülmúlja a hagyományos duál első fázis módszert. Több valós esetben képes az elméleti maximumot is nyújtani, vagyis duál megengedetté tenni a megoldást egyetlen nem triviális iterációval.

8. GDPO kedvező működésének egyértelműen az a magyarázata, hogy minden lépésben a maximális előrehaladást valósítja meg, ami egy kiválasztott kilépő változó esetén lehetséges, és ami csak (akár nagyon) sok normál duál szimplex iterációval lenne elérhető. Sok töréspont felhasználása esetén ez hatalmas különbséget jelent.

Fentiek alapján levonható az a következtetés, hogy GDPO elméleti és gyakorlati szempontból egyaránt jelentős algoritmus, aminek feltétlen helye van a duál szimplex algoritmus korszerű eszköztárában.

Hivatkozások

[1] Dantzig, G. B., Maximization of a linear function of variables subject to linear inequalities, in: Koopmans, T. C. (ed.), *Activity analysis of production and allocation*, 339–347 (Wiley, New York, 1951).

[2] Fourer, R., Notes on the Dual Simplex Method, unpublished (1994, March).

[3] Karmarkar, N., A New Polynomial-Time Algorithm for Linear Programming, *Combinatorica* **4** (1984), 373–395.

[4] Maros, I., A Piecewise Linear Dual Procedure in Mixed Integer Programming, in: Giannesi, F., Komlósi, S. and Rapcsák, T. (eds.), *New Trends in Mathematical Programming*, 159–170 (Kluwer Academic Publishers, Boston, 1998).

[5] Maros, I., *Computational Techniques of the Simplex Method*, International Series in Operations Research and Management 61 (Kluwer Academic Publishers, Boston, 2003).

[6] Maros, I., A Piecewise Linear Dual Phase-1 Algorithm for the Simplex Method, *Computational Optimization and Applications* **26** (2003), 63–81.

[7] Roos, C., Terlaky, T. and Vial, J.-Ph., *Theory and Algorithms for Linear Optimization*, Discrete Mathematics and Optimization (Wiley, 1997).

[8] Terlaky, T. (ed.), *Interior Point Methods of Mathematical Programming*, Applied Optimization 5 (Kluwer Academic Publishers, Boston, 1996).

[9] Wolfe, Ph., A technique for resolving degeneracy in linear programming, *SIAM Journal of Applied Mathematics*, **11** (1963), 205–211.

(Beérkezett: 2005. július 22.)

MAROS ISTVÁN
DEPARTMENT OF COMPUTING
IMPERIAL COLLEGE
LONDON
i.maros@imperial.ac.uk

INVESTIGATING PHASE 1 OF THE DUAL SIMPLEX

ISTVÁN MAROS

The paper performs a theoretical and computational analysis of a new dual simplex algorithm GDPO that is based on a piecewise linear phase 1 objective function. It concludes that it is able to considerably outperform the traditional dual phase 1 methods. It offers enhanced numerically stability and more effectiveness in coping with degeneracy. Tests on 48 real life problems indicate that the theoretically possible improvements are very likely to materialize in practice thus making this algorithm a prime candidate for inclusion in any modern simplex solver.

Alkalmazott Matematikai Lapok **23** (2006), 163–189.

FEJÉR LIPÓT 100 ÉVE HABILITÁLT KOLOZSVÁRON STABILITÁSELMÉLETBŐL

GARAY BARNABÁS, HATVANI LÁSZLÓ, KOLUMBÁN JÓZSEF

Budapest, Szeged, Kolozsvár

Fejér Lipót száz esztendeje, 1905. június 23-án habilitált a kolozsvári egyetemen. Habilitációs előadásának témája a közönséges differenciálegyenletek stabilitáselmélete volt, címe pedig *„Stabilitási és labilitási vizsgálatok a tömegpontrendszer mechanikájában"* [8].

Ebben a dolgozatunkban Fejér Lipóton kívül szeretnénk emléket állítani a kolozsvári iskolának is, amely a magyar matematika első, valóban nemzetközi rangú szakmai közössége volt. Fejér habilitációs előadásának ismertetését a stabilitáselmélet 1900 körüli általános helyzetét bemutató alfejezet, valamint a Ljapunov munkásságát röviden leíró Függelék foglalják keretbe. A hangsúlyt a habilitációs előadásban szereplő mechanikai példára helyezzük, amelyet Fejér – a topológia fogalmi és technikai apparátusának akkori viszonylagos fejletlensége miatt – csak részben tudott kielemezni [7], [9]; megmutatjuk, hogy Fejér gondolatmenetét az ötven évvel később felfedezett LaSalle-elv teszi teljessé.

1. A helyszín és a főszereplő

Gergely Jenő, a kolozsvári Bolyai Tudományegyetem egykori tanára, az első világháború idején Riesz Frigyes tanítványa volt. A társszerzők egyike tőle hallotta az alábbi történetet.

A Keleti Pályaudvaron a magyar tudományos élet számos kiválósága izgatottan várta a párizsi gyors érkezését. Henri Poincaré, a kor egyik legnagyobb tudósa jött Budapestre, hogy átvegye a Bolyai-díjat.[1] Az állomáson az üdvözlő szavak el-

[1] A Magyar Tudományos Akadémia Elnöksége 1902-ben Kolozsváron tartott ünnepi megemlékezést Bolyai János születésének századik évfordulója alkalmából. Ekkor jelentették be a Bolyai-díj létrehozását, amelyet – a Nobel-díjhoz hasonló feltételek mellett – ötévenként szándékoztak odaítélni a világ legjobb matematikusai egyikének. Elsőként ezt a díjat Henri Poincaré nyerte el 1905-ben. A második díjat David Hilbert kapta. Az 1915-ös díjkiosztás a háború miatt elmaradt. A Bolyai-díj történetének első szakasza a háborút követő inflációval (pontosabban az alapítványi

hangzása után megszólalt Poincaré is: – Hol van Fözsé? – kérdezte. A magyarok zavartan néztek össze. Ki lehet az a Fözsé? Hamarosan rájöttek, hogy Fejér Lipótról, a kolozsvári egyetem tanáráról van szó, aki 25 éves kora ellenére az akkori idők egyik legismertebb magyar matematikusa volt. Érthető tehát Poincaré óhaja, hogy magyarországi rövid látogatása alkalmával találkozhasson a fiatal tudóssal. Mit lehetett tenni? Hathatós közbenjárás után, néhány óra múlva egyetlen személykocsiból és mozdonyból álló különvonat robogott Fejérrel Kolozsvárról Budapest felé ...

Akár hiteles, akár nem, a történet azért is figyelemreméltó, mert Fejér Lipót nem az egyedüli matematikus, akinek tevékenysége lényegesen befolyásolta a XX. század tudományának fejlődését, és aki akkortájt hosszabb-rövidebb ideig a kolozsvári egyetemen dolgozott. Érdekes, hogy a XIX. század második felében a magyarországi tudományegyetemek közül nem a budapesti, hanem a kolozsvári vált fontos matematikai centrummá. A háborút megelőző másfél évtizedben a kolozsvári matematikai iskola a világ legjobbjai közé tartozott. Véleményünk szerint a XXI. században sem múlik el olyan munkanap, amelyen valahol a nagyvilágban Farkas Gyula, Fejér Lipót, Haar Alfréd vagy Riesz Frigyes nevét ne emlegetnék.

Hogyan jutott a XX. század elején a kolozsvári matematika ezekre a magaslatokra? Milyen személyi feltételek, társadalmi és gazdasági jelenségek, tudományművelési és oktatásfejlesztési stratégiák segítették a fejlődésben?[2]

1.1. A kolozsvári matematika a 19. század végén

1872-ben, a kolozsvári egyetem megnyitásakor a matematika szempontjából Erdélyben a helyzet nem volt kecsegtető. A két Bolyai tevékenységétől eltekintve a kor itteni matematikai irodalma egészen szegényes. A leginkább említésre méltó esemény Brassai Sámuel nevéhez fűződik, aki magyarra fordította Eukleidész *Elemek* című könyvét. Amikor a kolozsvári egyetemen a matematikai tanszékek megszervezésére sor került, az erdélyiek közül csak Brassai jöhetett szóba, aki akkor az Erdélyi Múzeum-Egylet igazgatója és a Természettár őre volt. Tudományos hírneve, tekintélye Kolozsváron olyan nagy volt, hogy az egyetem ügyével foglalkozó újságcikkek állandóan emlegették őt a jelöltek között. Már 1870-ben egyetemi tanárságra pályázott Pesten, és várta, hogy hívják meg a szanszkrit nyelv tanárának. Úgy tudjuk, Brassait báró Eötvös József tanügyminiszter megkérdezte, hajlandó volna-e az alapítandó kolozsvári egyetemen valamilyen tanszéket vállalni. A választást őrá bízták, de Brassai több tárgyat jelölt meg, ami utóbb a kinevezéseket eszközlő új tanügyminiszternek, Trefort Ágostonnak nem kis fejtörést okozott, „nem könnyen állapodhatván meg abban, melyik tanszékre nevezzék ki". Brassai azt ajánlotta, hogy válasszanak a filozófia, növénytan, pedagógia, művelődéstörténet, nyelvtudo-

pénzek általános elértéktelenedésével) együtt ért véget. A díjat a Magyar Tudományos Akadémia 1994-ben újraélesztette.

[2]Bizonyára az erdélyi magyarság mai gondjainak kezelése szempontjából is tanulságos lehet az ezekre a kérdésekre adott válasz.

mányok és a matematika között. Végül kinevezték nyilvános rendes tanárnak az elemi mennyiségtani tanszékre, amire maga Brassai is a legkevésbé számított.

A másik kinevezett matematikus nyilvános rendes tanár Martin Lajos volt a felsőbb mennyiségtan tanszékén, aki mérnökként lett 1859-ben az Akadémia levelező tagja, 1871-től pedig a kolozsvári távirda igazgatója. A kinevezés pillanatában, a szó szoros értelmében sem Brassai, sem Martin nem volt matematikus. Egyikük sem végzett matematikai kutatásokat, nem ez állt érdeklődésük középpontjában. Viszont mindketten érdeklődtek alkalmazott matematikai kérdések iránt: az előbbi csillagászattani, az utóbbi pedig repüléstechnikai tárgyú dolgozataival igazolta ezt.

A harmadik matematikai tanszékre, melynek neve mennyiségtani fizika volt, csak 1874-ben találtak megfelelő embert a fiatal Réthy Mór személyében.

Az egyetemnek nagyon keményen kellett küzdenie a kezdet nehézségeivel. Számos, többnyire célszerűtlen épületben folyt az oktatás. Az intézetek s az egyetemi könyvtár helyzete szinte kétségbeejtő volt. Igaz ugyan, hogy az Erdélyi Múzeum-Egylet összes gyűjteményét, mintegy 30 000 kötetből álló könyvtárával együtt, évi ötezer forintért 50 évre az egyetemnek bérbe adta, de az oktatás és a tudományos kutatás céljaira szolgáló szakkönytár alapját még ezután kellett lerakni. Az állam, a szükségletekhez képest, kezdetben csak csekély összeggel tudta segíteni az egyetemet. A kolozsvári társadalom maga is „kevéssé buzdult fel, s az első évben mindössze csupán négy alapítványt tett az ifjúság számára". A tanárok csak hosszas keresgélés után találtak maguknak megfelelő lakást, s a kisvárosban a diákság sem talált elég szórakozást, ezért „zajos mulatságokban keresett kárpótlást".

Ilyen előzmények után nem csoda, hogy az egyetem létesítése után néhány évig a kolozsvári matematika a felzárkózás kényszere alatt állt. A nehézségeket tetőzte, hogy azokban az években valójában az egész magyar matematika hasonló gondokkal küszködött. Az érdekeltek többsége tudta, hogy az egyetem csak úgy töltheti be igazi szerepét, ha ott az oktatás mellett magas szintű tudományos kutatás folyik. Ennek ellenére, úgy tűnik, az első években a tanárok között nem volt egyetértés az egyetemi tevékenység céljait és formáit illetően. A Ferenc József Tudományegyetem 1896-ban kiadott története és statisztikája szerint 1878. szeptember 8-án Imre Sándor rektor a tudományos szellem hiányát kifogásolta, hangsúlyozva, hogy az egyetem nem iskola, hanem tudományos intézet, s a kettőt nem szabad összetéveszteni. Ugyanakkor, „első és fődolognak a tudományok meggyökereztetését s terjesztését tartván, erélyesen sürgette a középiskolák reformját, hogy az ifjak igazán megfelelő előkészültséggel jöjjenek az egyetemre". A következő évi rektorváltás idején a régi rektor büszkeséggel utalt az utóbbi tanévben kétszer tartott pályázati ünnepségre s az egyetemi ifjúság tudományos szorgalmának, önállóságának jeleire, valamint a tanári karnak a nemzeti és egyetemes tudományosság előbbre vitelében tanúsított és külföldön is elismeréssel fogadott munkásságára. Ezzel szemben Brassai Sámuel, az új rektor „kikelt az ellen, hogy a tanár ne csak maga törekedjék új dologra a tudományban, hanem tanítványait is ez irányban vezesse. Az egyetemi tanárnak nem az az első és fő kötelessége, hogy újat találjon, s az efféle research iránt való követelőzés felesleges, jogtalan és képtelen. Annál feleslegesebb, jogtalanabb és képtelenebb magukkal a hallgatókkal szemben." Ezért kifogásolta a pályadíjakat,

„mint a melyek megtestesítései ama, már sokszor elutasított követelménynek, hogy a tanítványt újak fölfedezésére, a tudomány tovább-vitelére kell vezetni". Szerencsére az elkövetkező években a kolozsvári egyetemen nem ez a felfogás vált uralkodóvá. Az egyetem egyre nagyobb figyelmet fordított a legtehetségesebb hallgatók alapos felkészítésére és a kutatásba való bevezetésére. A hallgatók számára kiírt kutatási pályázatok évről évre népszerűbbekké váltak, a tananyag pedig fokozatosan közeledett a tudomány aktuális problémáihoz. A legjobbak továbbképzését ösztöndíjakkal segítették. A tanárképzés munkájától függetlenül folyt az egyetemen az alkotómunkára alkalmassá tevő tudósképzés. Ugyanakkor az egyetem vezetősége nagy gondot fordított olyan tanárok kinevezésére, akik híres külföldi egyetemeken doktorátust szereztek, és ott elismert tudományos eredményeket értek el. Ezek sok olyan speciális kollégiummal segítették a tehetséges fiatalok fejlődését, amelyek egy-egy divatos diszciplína legmélyére nyúltak, magukba foglalva előadójuk sokszor egészen alapvető eredményeit is. Kolozsváron matematikából a számelmélet, a differenciálegyenletek elmélete, a komplex függvénytan, a vektoralgebra és analízis, a kvaterniók és az elliptikus függvények tana, valamint a Bolyai-geometria volt előtérben. Az eredményességet és színvonalat nemcsak a megjelent néhány monográfia, a sok litografált jegyzet, de a bel- és külföldi folyóiratokban közzétett értekezések egyre növekvő száma is bizonyítja.

Az állam évről évre többet költött az egyetemre: az első 15 évben 160 000 forintról 279 000 forintra nőtt a költségvetési támogatás. A tanárok tudományos működéséről és az egész egyetemről elismerő szavak hangzottak el a sajtóban s az országgyűlésen is. Mindezen erőfeszítések ellenére 1876 és 1886 között a matematikus hallgatók száma 67-ről 23-ra csökkent. Az egyetem vezetősége abban látta az okot, hogy éveken át még a jelesebb tanárjelöltek is csak nehezen találtak maguknak megfelelő állást. Annak fő okát pedig, hogy az egyetemi hallgatók száma általánosságban véve is alacsony maradt, a város rendkívüli drágaságában kereste.

A XIX. század utolsó évtizedének elején az egyetem életében fontos változások történtek. A tanszékek száma 51-re, a tantestület tagjaié 68-ra, a tanársegédekkel és gyakornokokkal együtt pedig a tudományos alkalmazottak száma 119-re emelkedett. A hallgatóké az első negyedszázadban 233-ról 702-re ugrott. Hasonlóképpen növekedett a matematikusok száma is. Különösen nagy nyeresége volt a tudománynak Farkas Gyula és a valamivel később Kolozsvárra került Schlesinger Lajos, akiknek munkássága döntő módon kihatott az itteni matematikai életre. Nagy szaktekintélyüket és rátermettségüket arra használták, hogy a matematikai oktatás és kutatás feltételeit minél magasabb szinten biztosítsák, és a kolozsvári iskola eredményeit mindenhol elismertessék.

Az épületgondok megoldása végett, Meixner Károly tervei szerint, Reményik Károly kolozsvári építész 1893-ban megkezdte a központi épület felépítését. Így 1895-ben Martin Lajos rektor, aki székfoglalójában a repülőgépről értekezett, immár az egyetem új épületében üdvözölhette az ifjúságot. Ugyanabban az évben Kolozsvár városa az államnak ajándékozta a Bel- és a Külső-Torda utca sarkán levő telket, hogy rajta az egyetemnek egészen modern könyvtári palotát emeljen.

A századfordulóra az egyetem Erdély szellemi központja, a tudományoknak valóban olyan szentélye lett, ahol gróf Mikó Imre szavai szerint „az ismeret, a felvilágosodás és ezáltal a szabadság igéit hirdetik". Mindez, természetesen, elválaszthatalan attól a társadalmi, technikai és gazdasági fejlődéstől, amely abban az időben Magyarországra – és benne Erdélyre – jellemző volt. Az ekkor kialakuló polgári társadalom igényelte az oktatás és a tudomány fejlesztését. A polgári családokból származó fiatalok egyre nagyobb hányada vonzódott a tudományokhoz. Nem egy fiatalember mondott le biztos megélhetést ígérő karrierjéről a tudományos pálya kedvéért.

Külön ki kell emelnünk a magyar középiskolai reformok jótékony hatását. A '90-es évek elejére sikerült kiépíteni egy modern iskolahálózatot, amelyben igényes, hatékony oktatás folyt. Kitűnő középiskolai tanárok jelentek meg, mint a matematikus Arany Dániel és Rátz László, akik fáradságot nem ismerő munkájukkal a fiatalok hosszú seregét nyerték meg a tudománynak. A matematikus-képzés szempontjából fontos szerepet játszott az 1893-ban elindított „magyar csoda", a *Középiskolai Mathematikai Lapok*, amely a maga nemében a legelső volt az egész világon (s a két háború által okozott kényszerszüneteket leszámítva) azóta is folyamatosan jelenik meg. Mindenkori szerkesztői pontosan tudták, milyen hasznos, ha tehetséges fiatalok, megfelelő irányítás mellett, éveken át példák tucatjain törik a fejüket és írják le gondolataikat. A ma hivatalosan KöMaL-nak nevezett lapon nevelkedettek egy része tudós lett, mások „csak" nagyon jó szakemberek, tanárok. Megoldói közül kerültek ki a XX. század legjobb magyar matematikusai, így azok is, akik az első világháborút megelőző időben világszintre emelték a kolozsvári matematikát.

1.2. Fejér Lipót kolozsvári elődei és kortársai

BRASSAI SÁMUEL (1800–1897) Torockószentgyörgyön született. Huszonegy éves korában a kolozsvári unitárius kollégiumban végezte a filozófiai tanfolyamot. Néhány évig nevelő, majd 1837-ben a kolozsvári unitárius kollégiumban a földrajz és a történelem tanára, később a matematikát és a természettant tanította. 1841-ben németországi tanulmányúton vett részt, azután visszatért tanári állásába mint igazgató. A szabadságharc után Pesten, majd ismét Kolozsváron tanárkodott. Akit a magyar matematikai szaknyelv fejlődése érdekel, sokat tanulhat Brassai írásaiból. Sok tankönyve közül már csak az 1883-ban Budapesten kiadott *Algebrai gyakorlatok*, valamint az általa magyarra fordított *Elemek* találhatók meg a kolozsvári matematikai könyvtárban. Őt tartják az utolsó magyar polihisztornak (aki sajnálatos módon nem ismerte fel Bolyai János új korszakot nyitó felfedezésének jelentőségét).

MARTIN LAJOS (1827–1897) Budán született, felsőbb tanulmányait szülővárosában, a Műegyetem jogelődjén és a Genie-Akadémián végezte. 1859-ig tartó hadmérnöki pályafutása után 1861-ben budai főmérnök, 1863–68 között középiskolai tanár. Ebben a minőségében, a közoktatásügyi miniszter megbízása alapján, mennyiségtan-, mértan- és ábrázoló mértan tankönyvet írt. 1868-ban a pesti távirda gondnoka, 1869-ben pedig a debreceni távirda helyettes igazgatója. Innen 1871-ben a kolozsvári távirdához igazgatónak nevezték ki. 1872-től haláláig a kolozsvári

egyetem felsőbb mennyiségtani tanszékének nyilvános rendes tanára. Tudományos tevékenységének középpontjában a repülés kérdésének megoldása állt. A repülés magyar úttörője. Főbb művei: *A vízszintes szélkerék* (Budapest, 1874), *Az erőműtani csavarfelületek* (Budapest, 1874), *A csillagászat újabbkori haladásáról* (Kolozsvár, 1877), *Variatio Számítás* (Kolozsvár, 1879), *A madárrepülés általános elmélete* (Kolozsvár, 1890). Ezek közül az első két mű, egy kötetbe foglalva, valamint a *Variatio Számítás* a kolozsvári matematikai könyvtárban ma is megtalálható.

A németországi híres egyetemeken doktorátust szerzett, alapos felkészültséggel és rendkívüli tehetséggel rendelkező fiatal matematikusok, akik a kolozsvári egyetemre kerültek, új lendületet adtak az itteni tudományos életnek. Közülük időrendi sorrendben az első RÉTHY MÓR (1848–1925) volt. Nagykőrösön született, a bécsi, a göttingeni és a heidelbergi egyetem hallgatója volt. Ez utóbbi helyen szerzett doktorátust 1874-ben. Már ebben az évben a kolozsvári egyetem rendkívüli tanára, majd két év múlva a mennyiségtani fizika tanszékének nyilvános rendes tanára. 1884–1886 között az elemi mennyiségtani tanszék vezetője. Ezután a budapesti Műegyetemre ment át. Az elméleti fizika egyik első magyarországi professzora. Külföldön is elismert eredményei az inkompresszibilis folyadéksugár alakjára vonatkoznak. Jelentősek a mechanika elveire és a kémiában szereplő Ostwald-elvre vonatkozó kutatásai. A matematika terén különösen értékesek azon eredményei, amelyek a Bolyai Farkas által vizsgált végszerűen egyenlő területek kérdésére vonatkoznak. Réthy Mór rendezte sajtó alá Kőnig Gyulával együtt a *Tentamen* második kiadásának első kötetét. Elévülhetetlen érdemeket szerzett a Bolyai-geometria kutatása és elismertetése területén is.

Réthy leghíresebb tanítványa a Marosvásárhelyen született VÁLYI GYULA (1855–1913) volt, aki később a kolozsvári egyetem első erdélyi származású, markáns matematikusa lett. Egyetemi tanulmányait 1877-ben Kolozsváron végezte, majd az egyetem támogatásával négy féléven át Berlinben Weierstrass, Kirchhoff, Kronecker, Borchardt és Kummer előadásait hallgatta. *A másodrendű parciális differenciális egyenletek elméletéhez* című doktori értekezését 1880-ban Kolozsváron védte meg. 1881-ben kezdte meg magántanári működését a kolozsvári egyetemen. 1884-ben a mennyiségtani természettan, majd 1886-ban az elemi mathésis tanáraul választották meg. Több fontos publikációja a *Mathematikai és Természettudományi Értesítő*-ben jelent meg. Amint Réthy is megjegyzi, „értekezéseinek nem a száma és nagysága, hanem azok minősége imponál". Testi törékenysége, egyre jobban elhatalmasodó szembetegsége gátolta a munkában, ennek ellenére „lelke maradandók alkotására vitte".

A kolozsvári egyetemen a matematikusok és fizikusok közül senki sem játszott fontosabb szerepet, mint FARKAS GYULA (1847–1930). Pusztasárosdon született, egyetemi tanulmányait a pesti egyetemen végezte, ahol főleg Jedlik Ányos volt rá nagy hatással. Később Batthyány Géza gróf jóvoltából franciaországi tanulmányúton vett részt. Farkas Gyula fiatalkori matematikai eredményei közül itt csak Bolyai Farkas trinomegyenletekre vonatkozó, a *Tentamen*-ben röviden tárgyalt, gyökközelítő algoritmusával kapcsolatos vizsgálatait említjük meg. Ezáltal a Bolyai-

algoritmus igen ismertté vált; általánosításaival, alkalmazásaival, a vele kapcsolatos konvergenciaproblémák vizsgálatával – a legújabb időkben is – több magyar és külföldi matematikus foglalkozott. Farkas Gyula 1887-ig Pesten magántanár, amikor kinevezik a kolozsvári egyetem tanárának. 1888-ban az egyetem rendes tanára lett, és e minőségében 1915-ig működött. Ortvay Rudolf, egykori tanársegéde írta róla, hogy „mélyreható kritika, a hajthatatlan, mellékes szempontok által el nem téríthető keresése az igazságnak jellemezte úgy tudományos működését, mint egyetemi közügyekben kifejtett tevékenységét ... és épp mivel nem kereste a népszerűséget, igen nagy tekintélyt tudott magának szerezni, és áldásdús befolyást gyakorolni az egyetemi ügyek vezetésére".

Egyetemi tanárként főleg az elméleti fizikai kérdéseivel foglalkozott, de a vizsgált problémák matematikai hátterét olyan mélységben dolgozta ki, hogy azok között klasszikus matematikai eredmények is vannak. Különösképpen a Fourier-féle mechanikai elv foglalkoztatta a '90-es évektől. Dolgozataiban egyenlőtlenségekkel adott kötések esetén az egyensúly szükséges feltételét adja meg. Ehhez bebizonyítja a homogén lineáris egyenlőtlenségekre vonatkozó tételét, mely *Farkas-lemma* néven az egyik legismertebb magyar matematikai eredmény, sőt egyike a világ matematikai irodalmában a legtöbbet idézett tételeknek. Ezeknek a munkáknak az alapján ma világosan látjuk, hogy Farkas Gyula egyike a modern optimalizáláselmélet megalkotóinak. Az utóbbi évtizedekben oly alaposan tanulmányozott és sokfelé alkalmazott variációs egyenlőtlenségek elméletének szintén egyik előfutára. Egyetemi előadásait gondos kidolgozásban litografálva közreadta. A kolozsvári matematikai könyvtárban még ma is megtalálható egyetemi jegyzetei: *Analytikus mechanika* (1907–08), *Analitikus mechanika* (1913–14), *Erőtan* (1913–1914), *A mechanika alaptanai* (1913–14). Ugyancsak megtalálható a Kolozsváron megjelent *Vector-tan és az egyszerű inaequatiok tana* című könyve. Utóbbi egy nagyon jól megírt vektoranalízis tankönyv, mely tartalmazza saját kutatásainak legfontosabb eredményeit is.

KLUG LIPÓT (1854–1944) Gyöngyösön született. Budapesten ábrázoló geometriából és matematikából szerzett tanári oklevelet. Pozsonyban, majd Pesten tanított, 1897-től 1917-ig a kolozsvári egyetemen az ábrázoló geometria tanszék tanára. Eközben, nagyszámú értekezés mellett, öt népszerű tankönyve jelent meg: *A projektív geometria elemei* (Budapest, 1892), *Projektív geometria* (Budapest, 1903), *Az általános és négy különös Pascal-hatszög configuratiója* (Budapest, 1898), *Ábrázoló geometria* (Budapest, 1900) és *A harmadrendű térgörbék synthetikai tárgyalása.* Ezek közül az első négy Kolozsváron ma is megtalálható. Sajnos, egyetemi jegyzetei közül csak egy van meg: *Az egyszerű görbe felületek ábrázolása* (1909–10).

A századfordulón a kolozsvári egyetem matematikai intézetének külföldön legismertebb tanára SCHLESINGER LAJOS (1864–1933) volt. Nagyszombatban született, középiskolai tanulmányait Magyarországon, egyetemi tanulmányait pedig Heidelbergben és Berlinben végezte. Mint a berlini egyetem magántanára, 1890-ben egy félévet Kolozsváron tanított.

1897-ben, amikor nyilvános rendes tanárként az egyetemre került, már ismert tudós volt. *Handbuch der Theorie der linearen Differentialgleichungen* című híres könyvének első két kötetét mint a berlini egyetem magántanára, a lipcsei Teubner Kiadónál jelentette meg. A befejező, harmadik kötetet 1898-ban már kolozsvári professzorként jegyezte. A matematikus társadalom úgy ismerte őt, mint a differenciálegyenletek komplex függvénytanra épített elméletének egyik meghatározó szaktekintélyét. Kevés olyan része van ennek a jelentős elméletnek, amelyet Schlesinger ne gazdagított volna lényegbevágóan új eredményekkel.

Kolozsvárra érkezve, a kutatói munka folytatása mellett, nagy lelkesedéssel végezte az egyetemi oktatást. A kolozsvári matematikai könyvtárban ma is megtalálható 15 egyetemi jegyzete: *Elliptikus függvények elmélete és alkalmazásai* (1898–99), *Égi testek mechanikája* (1898–99), *A differentiál-számítás* (1900), *Riemann-féle felületek* (1900), *Elliptikus függvények* (1901), *Bevezetés a variatio számításba* (1902), *A tér abszolute igaz tudománya* (jubiláris előadássorozat Bolyai János születésének 100. évforulójára), *Differenciálszámítás* (190?), *Az abszolút sík eltolásaiból alkotott discontinuus csoportokról* (1905), *Fuchs-féle függvények* (1906–07), *Égitestek mechanikája* (190?), *Görbevonalak és felületek elmélete* (1907–08), *Válogatott fejezetek az infinitesimális geometriából* (1908), *Égi testek forgásáról* (1908–09), *Differenciál-egyenletek elmélete* (1909–10). Előadásai lenyűgöznek áttekinthető, világos stílusukkal és a tárgykörök akkori legújabb eredményeinek szabatos tárgyalásával. Kétségtelen, hogy a mindenkori kolozsvári matematikai oktatás csúcsteljesítményei közé tartoznak.

Schlesinger Lajos kolozsvári tudományos munkásságának fontos eredménye a Teubner Kiadónál 1908-ban megjelent *Vorlesungen über lineare Differentialgleichungen*. Ez nem a fent említett *Handbuch*-ban kifejtett elmélet átdolgozása, hanem a lineáris differenciálegyenlet-rendszerek teljesen új módszerekkel való tárgyalása. Ez az első monográfia, amelyben a Vito Volterra olasz matematikus által értelmezett szorzatintegrál segítségével kezelik a lineáris differenciálegyenlet-rendszereket. Schlesinger szaktekintélyének elismerését jelenti a Német Matematikai Társaság gondozásában – a Teubner Kiadónál 1909-ben – megjelent *Bericht über die Entwickelung der Theorie der linearen Differentialgleichungen seit 1865* című könyve is. *Automorphe Funktionen* című ismert könyvét már mint a giesseni egyetem professzora írta.

Kolozsváron töltött évei alatt igen sokat tett az itteni matematikai élet fellendítéséért. Farkas Gyulával és Vályi Gyulával együtt döntő szerepe volt abban, hogy az egyetemen kitűnő matematikai könyvtár jött létre. Schlesinger volt az, aki Kolozsváron Bolyai János szülőházát felkutatta. A Bolyai-geometria, amint egyetemi jegyzeteiből is kitűnik, szívügye volt. Bolyai János születésének centenáriumi ünnepségén tartott emlékbeszédében fontos érveket sorakoztatott fel a Bolyai-Lobacsevszkij-geometria prioritási vitájában, nagymértékben hozzájárulva Bolyai János lángelméjének elismertetéséhez.

Végül megemlítünk két nagy matematikust, akik ugyan Fejér távozása után, de az ő közreműködésével jöttek Kolozsvárra, és akikkel mindig szoros kapcsolatot tartott fenn.[3]

RIESZ FRIGYES (1880–1956) 1912 és 1919 között vezette a felsőbb mennyiségtan tanszéket. Győrött született, és igen gondos nevelésben részesült. A középiskolát a bencésrend győri gimnáziumában végezte, egyetemi tanulmányait pedig a zürichi műegyetemen kezdte. A tudományos hivatás utáni vágya azonban győzedelmeskedett a biztosabb megélhetést nyújtó mérnöki pálya vonzóerején, és tanulmányait 1899-től a budapesti tudományegyetemen folytatta, majd egy évet töltött Göttingenben. Budapesten Kőnig Gyula és Kürschák József, Göttingenben Hilbert és Minkowski előadásai voltak rá a legnagyobb hatással. 1902-ben Budapesten avatták doktorrá, majd Lőcsén lett gimnáziumi tanár. Fiatalkori felfedezései hamar világhírűvé tették, még mielőtt Kolozsvárra került volna.

Egyetemi jegyzetei közül Kolozsváron ma csak három található meg: *Függvénytan* (1911–12), *Fourier-féle sorok* (1913–14) és *Függvénytan* (1914–15). Ugyancsak megtalálható a Riesz kolozsvári évei alatt Párizsban megjelent *Les systemes d'équations linéaires a une infinité d'inconnues* című könyve, amely jelentős szerepet játszott a funkcionálanalízis fejlődésében.

A mai Magyarország területéről Kolozsvárra került kiváló fiatal matematikusok sorát *Haar Alfréd* (1885–1933) zárta. Budapesten született, és az ottani evangélikus gimnáziumban érettségizett, ahol Rátz László, a *Középiskolai Mathematikai Lapok* neves szerkesztője volt a matematika tanára. Ennek a lapnak Haar is szorgalmas munkatársa volt középiskolás korában. Az előző évben érettségizettek számára évenként tartott országos „Eötvös Loránd matematikai verseny"-en 1903 őszén Haar Alfréd nyerte az elő díjat. Egyetemi tanulmányait Budapesten és Göttingenben végezte. Budapesten Beke, Eötvös, Frölich, Kürschák, Rados, Scholtz, Göttingenben pedig Carathéodory, Hilbert, Klein, Minkowski, Prandtl, Runge, Schwarzschild, Voigt és Zermelo előadásain, illetve szemináriumain vett részt. Hilbertnél doktorált 1909-ben. Ezt követően magántanár a zürichi műegyetemen. 1912-ben a kolozsvári egyetem egyik matematika-fizika tanszékére nevezték ki, először nyilvános rendkívüli tanárrá, majd 1917-ben nyilvános rendes tanárrá.

Haar Alfrédtól könyv nem jelent meg, azonban több igen gondosan megírt jegyzetet készített, amelyek sok eredeti részletet tartalmaznak, s mint egyetemi tankönyvek ma is beválnának. Sajnos, ma ezek közül Kolozsváron csak négy található meg: *Differential-Gleichungen* (Göttingen, 1911), *Algebra* (1912–13), *Determinánsok és quadratikus formák* (1912–13), *Számelmélet* (1915–16).

1.3. Fejér Lipót Kolozsváron

Farkas Gyula – máig ható tudományos tevékenysége mellett – meghatározó szerepet játszott abban, hogy a kolozsvári matematikai intézet, szinte a semmiből

[3] Jelen írásunknak nem lehet célja Riesz Frigyes és Haar Alfréd matematikai munkásságának ismertetése. Életútjukat is csak kolozsvári tartózkodásuk végéig követjük. Ugyanez vonatkozik magára Fejér Lipótra is.

indulva, az egyetemalapítás után alig negyedszázaddal a Monarchia legkiválóbb tudományos műhelyei közé emelkedett. A tudomány iránti elhívatottsága, nagyszerű emberi kvalitásai következtében tanártársai és tanítványai körében egyaránt nagy tekintélynek örvendett, melyet az egyetemi ügyek intézésében érvényre is juttatott. Többször volt dékán és egyízben az egyetem rektora is. Szava mindenben döntő volt. Befolyását rendszerint arra használta, hogy az egyetemi tevékenység anyagi és személyi feltételei minél jobbak legyenek. Igényes volt, sokat kívánt magától és másoktól is. Az emberi értékeket igen nagyra becsülte, és sokat tett azokért, akiket arra érdemesnek tartott. Fejér Lipót esete is bizonyítja, hogy mennyire szívén viselte a honi matematika és fizika ügyét, ugyanis 1905 márciusában Fejér Lipót kifejezetten Farkas Gyula közbenjárására került a kolozsvári egyetemre. (Valószínűleg ő kezdeményezte Schlesinger Lajos (1897), Riesz Frigyes (1911) és Haar Alfréd (1912) kinevezését is.)

Miért éppen Fejér Lipót? Mit tudhattak róla kinevezésekor a kolozsvári egyetemen? Tudták, hogy Pécsett született 1880. február 9-én. Gimnazistaként egyik legsikeresebb megoldója volt a *Középiskolai Mathematikai Lapok* feladatainak. 1897-ben a Matematikai és Fizikai Társulat versenyén második helyezést ért el. Ugyanebben az évben beiratkozott a budapesti műegyetem gépészmérnöki szakosztályába, egy szemeszter után azonban átlépett az ún. egyetemes szakosztályba mint a matematika- és fizikatanári szak hallgatója. Itt főleg Kőnig Gyula, Kürschák József és Rados Gusztáv előadásait hallgatta, majd átkerült a budapesti tudományegyetemre. Az 1899/1900-as tanévet a berlini egyetemen töltötte, ahol Frobenius, L. I. Fuchs (aki egyébként Schlesinger apósa volt) és H. A. Schwarz előadásait látogatta. Hazatérve, az 1900/1901-es tanévet ismét a budapesti egyetemen töltötte. Ezalatt publikálta a párizsi *Comptes Rendus*-ben a Fourier-sorokkal kapcsolatos tételét, ami nevét világszerte egy csapásra ismertté tette. Tanári szakdolgozatát matematikából a Fourier-sorokról, fizikából pedig a fényelhajlásról írta. 1901. szeptember 1-től repetitor volt a budapesti egyetemen, majd ugyanott 1902 tavaszán megszerezte a bölcsészdoktorátust. Az 1902/1903-as tanév első felét Göttingenben, második szemeszterét pedig Párizsban töltötte. Göttingenben főleg Hilbert és Minkowski, Párizsban Picard és Hadamard előadásait hallgatta. 1905-ig kilenc közleménye jelent meg, melyből három a már említett *Comptes Rendus*-ben, egy pedig a *Mathematische Annalen*-ben. Ilyen előzmények után – Farkas Gyula szavait használva – senki sem szerette volna „elveszíteni őt a külföldnek". A kolozsvári egyetem lehetőséget biztosított számára az előlépésre. A mennyiségtani természettan tanszéken (melynek egyedüli tagja Farkas Gyula volt) repetitorként alkalmazták. Ebben ragyogó matematikai tehetségének tanúbizonysága mellett annak is szerepe lehetett, hogy korábban fizikát is tanult, és érdeklődéssel fordult az elméleti mechanika felé. Fejér Kolozsvárra kerülése után három hónappal, 1905 nyarán habilitált. Az 1905/1906-os tanév első felében Schlesinger Lajos tanársegédeként heti két órában gyakorlatokat vezetett a felsőbb és elemi mennyiségtan, valamint a mennyiségtani természettan tanszékei mellett. A második szemeszterben már az analízis s az analitikai mechanika magántanára. Ekkor a variációszámítás II. részét tanítja heti két órában, valamint az infinitezimalis calculus alkalmazásait a görbevonalak és

felületek elméletére, heti három órában. 1906 szeptemberében adjunktus lett, 1911-ben pedig nyilvános rendkivüli tanár. A kolozsvári egyetemen tevékenységét ez év nyarán befejezte, mert szeptembertől a budapesti egyetem nyilvános rendes tanára lett. Vele párhuzamosan Schlesinger Lajos is távozott Kolozsvárról.

Az 1911-es év őszén Farkas Gyula több olyan levelet írt Fejér Lipótnak, amelyek ma is megtalálhatók Fejér hagyatékában. A Fejér által írott levelek hollétéről nincs tudomásunk. Farkas Gyula Fejérhez intézett levelei közül néhányat Prékopa András közölt egyik tanulmányában [16]. Ezekből a levelekből arra következtethetünk, hogy Fejér feladott állásának betöltésére Riesz Frigyest javasolta.[4]

Kolozsvári évei alatt Fejér Lipót egyik fontos feladata volt a Matematikai Szeminárium ügyvezető igazgatása. Ezt a (mai szóhasználattal élve: összintézeti) szemináriumot, amelynek vezetői (akkori szóhasználattal: igazgatói) Farkas Gyula, Schlesinger Lajos és Vályi Gyula voltak, 1901-ben létesítették. A konkrét szervezői, „ügyvezető igazgatói" teendőket 1905-ig Schlesinger látta el, utána Fejért bízták meg vele.

Végül szóljunk arról is, milyen város volt Kolozsvár száz évvel ezelőtt és hogyan élt benne Fejér Lipót. Passuth László *Kutatóárok* című önéletrajzi regényében felvillant néhány emlékképet a város akkori hangulatából. *„... úgy idézem vissza a várost, mint amelyet egyik oldalon a Monostor, a másikon a Hóstát zöld övezete határol. A Hóstát két végtelenbe nyúló főutcája mellett a néhány ezer főnyi parasztság „corpus separtum"-ként élt. Elég sok vasutas tömörült telepekbe, kevés kisebb gyárüzem volt. Viszonylag sok volt a bank, az egyházi intézmény, de mindent felülmúltak az iskolák. A nem városi eredetű diákok özönlése szeptembertől júniusig tartott, s rengeteg kosztot adónak, szabónak, kvártélyosnak juttatott kenyeret. Szemináriumokon töltötték a bölcsességet mezei jogászok fejébe, s viszonylag színes volt a város „éjszakai" élete is. ... Ipari munkásság csekély számban élt, s nem emlékszem arra, hogy bérkövetelések, sztrájkok vagy munkanélküliség formájában éreztünk volna társadalmi feszültséget. A szociális eszmék elsősorban elméleti forrásokból közelítettek: egyetemi professzorok „radikalizálódtak", ... erős volt a Munkásbiztosító Pénztár hatása (... Kun Béla is itt dolgozott) ... A sokféle vallás együttélése feltételezett*

[4]Farkas Gyula Fejér Lipótnak (1911. október 1-én) ... *Kérem azonban, hogy szíveskedjék vélem közölni Riesz Frigyes lakcímét, hogy a helyettesítés ügyének eldöntése után lehető közvetlenséggel fordulhassak hozzája* ... (1911. október 3-án) ... *A bizottság tegnap javaslatomat magáévá tette, s a rendes tanszék helyettesítésére meg a szeminárium ügyvezető igazgatására Riesz Frigyes fölkérését javasolja. Kari ülés holnapután* ... (1911. október 20-án) ... *A mai napon Riesz Frigyes új társunk megérkezett; délben a dékáni hivatalban találkoztunk véle a dékán, Vályi meg én, s immár a matematikai szemináriumok vezetését is kezébe adtuk. Nyomban megírta a fekete táblánkon előadásainak és gyakorlatainak a hirdetését, 23-án megkezdi előadásait. Most már bizonyos jótékony megnyugvás szállotta meg árvaságunkat. Haar felől írt szíves értesítését vételekor azonnal közöltem a dékáni hivatallal* ... (1911. november 4-én) ... *íme Riesz Frigyes társunk már javában megkezdte itteni működését nagy lelkesedéssel a math. szeminárium körül is* ... *Tegnap levelet kaptam Haartól. Úgy látszik ebből, hogy meghívójában, mint a tanszéki bizottság kari referense meg lettem nevezve. Következőleg válaszomban jónak láttam kissé körvonalazni a lényeget. Egyébiránt Haar levele oly mély hazafias érzelmeket árul el, hogy most már nem is tartok attól, hogy ha egyszer haza került, elveszítsük a külföldnek* ... A megszólítás *Kiválóan Tisztelt Kedves Tanártársam*, illetve *Fölöttébb Tisztelt Kedves Tanártársam*, az aláírás pedig minden esetben *Igaz barátja, Farkas Gyula.*

bizonyos türelmességet, melyben nem volt sok helye a vaskalaposságnak. Mert talán sehol Magyarországon nem volt olyan tarka a lakosság vallási megoszlás szerinti térképe akkor, mint ebben a városban, ahol a főiskolák egy része is felekezeti formákban élt tovább. A többség katolikus volt, de a város törzsökös lakosságának komoly hányada – kálvinista. A lutheránusok inkább szász ajkúak. Az unitáriusok egyetlen püspöksége is itt volt. Két „görög" templom osztozott a román híveken: a görögkatolikus és a görögkeleti. A háború előtt patrícius zsidóság élt a városban, neológ életformák között. A biedermeier város a maga apró szellemi vulkánjaival alapjában provinciális pletykafészek volt, bár a városlakók távolról sem ismerték úgy egymást, mint más ötven-hatvanezer lakosú városban. Ennek oka az állandó népcserélődés volt, elsősorban a középosztályban: nagyszámú tisztviselő telepedett be, vagy vándorolt el, a hivatali lét mozgástörvényei szerint ...".

A New York Szálló nagykávéháza volt a társadalmi és szellemi élet központja. A teremben két hosszú különasztal volt, ahol egyetemi tanárok s mágnások jól megférték egymás mellett. Sűrűn ültek össze, poharaztak, közösen vonultak a „művészasztalokhoz". Ott a Nemzeti Színház színészei, írók, újságírók vertek hagyományszerű tanyát. „Van itt Kolozsvárt egy néhány nyakas, javíthatatlan idealista. Kifinomodott izlésű, művészi hajlandóságú emberek. Van közöttük piktor is, szobrász is, építész is, író ember is" – írja az *Ellenzék* 1905. január 3-án. A város sajátos, történelmi légköre, s ugyanakkor szellemi avantgardizmusa éreztette hatását ezeken az összejöveteleken. A szájhagyomány szerint Fejér Lipót törzsvendég volt a New York kávéházban. A szépirodalomban és zenében egyaránt tájékozott matematikus a társasági élet népszerű alakja volt. Így találkozott egy tarokkparti sodrában Passuth anyjával is, aki nagyjából egyidős volt vele. Passuth elbeszélése szerint, játék után az ifjú professzor megkérdezte a fiatal hölgyet, miért nem iratkozik be az egyetemre, s felajánlotta, hogy vállalja matematikai előkészítését ...

Fejér szellemi érdeklődése a matematikán messze túlterjedt. Szenvedélyesen szerette a zenét, és maga is jól zongorázott. Muzsikusok és írók éppúgy nagyra tartották ítéleteit, mint esztéták és jogfilozófusok. A New York kávéházban többször találkozott Ady Endrével is. Barátságukat Ady neki ajándékozott arcképén a dedikáció melegsége illusztrálja.

A Farkas utca volt a tudomány és a múzsák közös szentélye, gyönyörű gesztenyesorral. Egyik végében az egyetem főépülete, mellette a régi kőszínház, velük szemben a piarista gimnázium, a másikban pedig a református kollégium és a toronynélküli református templom. Ezek, a köréjük társult intézményekkel együtt, mintegy megszabták e nevezetes városrész hangulatát. Ebből az utcából tért be Fejér Lipót nap mint nap munkahelyére, az egyetem főépületébe.

2. Habilitáció, szóban-írásban

Fejér Lipót éppen száz esztendeje, 1905. június 23-án habilitált a kolozsvári Ferencz-József Tudományegyetem Matematikai és Természettudományi Karán. Ha-

bilitációs előadásának témája a közönséges differenciálegyenletek stabilitáselmélete volt, címe pedig „Stabilitási és labilitási vizsgálatok a tömegpontrendszer mechanikájában". A habilitációs előadás témaválasztását minden bizonnyal Farkas Gyula és Schlesinger Lajos kezdeményezték, összhangban a kolozsvári matematika hagyományaival és – amint arra már a korábbiakban utaltunk – az analitikai mechanika leendő magántanárának oktatási feladataival.[5] Fejér habilitációs előadásának írásos változata a *Mathematikai és Fizikai Lapok* 1906-os évfolyamában jelent meg [8]. Az ezt (logikailag) követő [7], [9] dolgozatok közvetlenül kapcsolódnak a habilitációs előadás témaköréhez.[6]

2.1. Stabilitáselmélet 1900 körül

A stabilitás fogalma a matematikába a mechanikából került, a mechanikába pedig a latin köznyelvből. *Stabilitas* állhatatosságot, szilárdságot, állandóságot, tartósságot, elmozdíthatatlanságot jelent. A görög hasonló értelemben a *hedraios* szót használja, amint az a magyar fül számára is visszacseng a poliéder, a „sokféleképpen letelepedni képes test" nevének hallatán. Jól állni (*sto* = *állni*) [latin], illetve jól ülni (*hedra* = *ülőhely*) [görög] – ez a stabilitás.

A stabilitás fogalmának a múlt század elején nem volt általánosan elfogadott matematikai definíciója. Magát az elnevezést akkor már vagy százötven éve egyre gyakrabban használták a legkülönfélébb mechanikai rendszerek egyensúlyi helyzeteinek, illetve általános megoldásainak vonatkozásában – magától értetődő természetességgel, de az egymáséitól gyakorta teljesen eltérő értelemben[7]. A stabilitás mint

[5]Az általában vett differenciálegyenletek elmélete egyébként fontos szerepet játszott Fejér tanulmányaiban, sőt az őt egy csapásra világhírűvé tevő szummációs tételt is az elliptikus differenciálegyenletek elméletének egy kérdése motiválta. A nevezetes eredmény megszületésének történetét (s benne az $\Omega = \{(x,y) \in \mathbf{R}^2 \mid x^2+y^2 < 1\}$ körlemezre vonatkozó $\Delta u = 0$, $u|_{\partial\Omega} = g$ peremfeladat megoldását leíró Poisson-féle integrálformula szerepét) Turán Pál írta meg a *Fejér Lipót Összegyűjtött Munkái - Leopold Fejér Gesammelte Arbeiten I–II, Akadémiai Kiadó, Budapest, 1970* kötetben, pontosabban az eredeti Comptes Rendus dolgozat ottani facsimile változatát követő kommentárban. Fejér egyébként kilenc differenciálegyenletek témájú cikket publikált, amelyek közül az első kettő [3], [4] még kolozsvári működése előtt jelent meg. Az Ostwald-féle mechanikai elvről írott két dolgozatát [5], [6] Réthy Mór [és minden bizonnyal a (leendő) kolozsvári állás], a habilitácós előadás utáni lineáris közönséges differenciálegyenletek dolgozatot [10] pedig Schlesinger Lajos hatása/késztetése inspirálta. Kolozsvári évei után Fejér egyetlen alkalommal tért vissza a differenciálelegyenletek témaköréhez [11]. Differenciálegyenletek témájú írásainak a hivatkozásjegyzékben történő felsorolása az *Összegyűjtött Munkái*-ban szereplő sorrendet (az egyes dolgozatok ottani sorszáma rendre 3, 7, 10, 11, 13, 14, 16, 18, 57) követi.

[6]Egészen 1918-ig általános szokás volt a hazai matematikában a magyar és a nem magyar nyelven párhuzamosan történő publikálás. A magyar és a német (esetenként francia) változatok megjelenési sorrendje a szerző szabad döntésétől függött, tartalmukat tekintve csaknem mindig a későbbi időpontban közlésre leadott változat a gazdagabb. Fejér esetében a [7] és a [9] dolgozatok egymás szó szerinti fordításai. Habilitációs előadásáról (legjobb tudomásunk szerint) csak 65 évvel később készült (német) fordítás, amely az *Összegyűjtött Munkái*-ban olvasható.

[7]Az első többé-kevésbé sikeres kísérletet egyensúlyi helyzet stabilitásának, pontosabban vízben úszó test felborulás elleni stabilitásának definiálására Euler tette, 1749-es „Hajózástudomány"-ának *De stabilitate, qua corpora aquae insidentia in situ aequilibrii persistunt* fejezetében. A mérnöki gyakorlat sokkal gyorsabban fejlődött, mint a rá vonatkozó absztrakció. Watt 1784-ben felfedezte a centrifugális regulátort, amellyel szabályozni és stabilizálni tudta a gőzgép által forgatott

olyan, konkrét mechanikai rendszerek tulajdonsága volt, nem pedig az őket leíró közönséges differenciálegyenletrendszereké. Stabil lehetett egy egyensúlyi helyzet, egy mozgás vagy éppen a mozgások összesége. Adott egyensúlyi helyzetek és mozgások stabilitása bizonyos zavaró hatásokra vonatkozott, amelyek a kezdeti állapot, illetve a mechanikai rendszer paramétereinek kismértékű megváltozását – ma úgy mondanánk, az $\dot{x} = f(x)$ egyenlethez tartozó kezdetiérték-feltétel, illetve az egyenlet jobb oldalának perturbációját – eredményezték. Ez a sokszínűség meghökkentő módon tükröződik Felix Klein ([14], 345. oldal) megjegyzésében: „Allgemein verbindet man nämlich mit dem Worte „instabil" die Auffassung eines ausnahmsweisen und turbulenten Vorganges". Instabilitás mint turbulencia? Stabilitás mint szokásos viselkedés? Teljesen egyet kell értenünk Lord Kelvin ([19], 282. oldal) Felix Klein által is idézett véleményével: „There is scarsely any question in dynamics more important for Natural Philosophy than the stability or instability of motion".[8]

A XIX. századi égi mechanika központi kérdése a Naprendszer (Lagrange) stabilitása volt, ami alatt azt értették, hogy a bolygók nem ütköznek össze sem egymással, sem a Nappal, de túlságosan el sem távolodhatnak tőle. A matematikai modell az úgynevezett n-test probléma, amikor is n darab pontszerű m_i tömegű test úgy mozog $\mathbf{R}^3$-ban, hogy rájuk csak az egymás általi tömegvonzás az egymástól vett mindenkori $|r_i - r_j|$ $(i \neq j)$ távolságok négyzetével fordítottan arányos erői hatnak. Vegyük észre, hogy az energiamegmaradás

$$(1) \qquad \frac{1}{2}\sum_{i=1}^{n} m_i \dot{r}_i^2 - \gamma\left(\sum\left\{\frac{m_i m_j}{|r_i - r_j|} \mid i \neq j;\ i,j = 1,2,\ldots,n\right\}\right) = \text{Const.}$$

formulája szerint $\max\{|\dot{r}_i| \mid i = 1,2,\ldots,n\}$ pontosan akkor nem korlátos, ha $\max\{|r_i - r_j|^{-1} \mid i \neq j;\ i,j = 1,2,\ldots,n\}$ sem az. Egyúttal az is látszik, hogy az n-test probléma pozitív időben korlátos megoldásai automatikusan a teljes $t \geq 0$ félegyenesen vannak értelmezve. Így a Nap és a kilenc bolygó (Lagrange) stabilitása – kihasználva a súlypont helybenmaradását is – matematikailag úgy fogalmazható meg, hogy a 10-test problémában szereplő valamennyi tömegpont mozgása hely és sebesség tekintetében $t \to \infty$ mellett korlátos marad[9].

tengely szögsebességét. Igazából csak ezzel lépte túl Alexandriai Heron majd 2000 évvel korábbi munkásságát, aki saját gőzgépét-gőzgépkezdeményét nem tudta jól „kordában tartani" (viszont sikerrel oldotta meg a lámpában égő olaj szintjének valamint a vízóra sebességének szabályozását).

[8] Az egyes idegen nyelvű idézeteket szándékosan nem fordítottuk le, hogy érzékeltessük a hazai szaknyelv (ki)alakításának a feladatát, amelyet egyébként, mint arra már utaltunk, Fejér és kortársai olyan derekasan felvállaltak.

[9] Általában is, jóllehet az égi mechanikán kívül ez az azonosítás teljesen önkényesnek tűnik, egy korlátos mozgást Lagrange értelemben stabilnak hívunk és vice versa. A (Lagrange) stabilitás = korlátosság szóhasználat védelmében meg kell jegyeznünk, hogy a közelítő eljárások Neumann János és Lax Péter által kiépített általános elméletének híres *konzisztencia & stabilitás* $\Leftrightarrow$ *konvergencia* tételében a stabilitás szintén korlátosságot jelent, lineáris feladatokra bizonyos lineáris operátorok egyenletes korlátosságát, nemlineáris feladatokra pedig bizonyos becslések egyenletességét. (Itt jegyezzük meg azt is, hogy a numerikus analízis egyes fejezetei a stabilitás kifejezést sok más egyéb értelemben is használják.)

Kihasználva az erőtér örvénymentességét, Poincaré igazolta, hogy amennyiben az n-test problémát leíró közönséges differenciálegyenlet $6n$ dimenziós fázisterének (testenként három hely- és három sebességkoordináta) egy korlátos V tartománya teljes trajektóriákból áll, akkor a V-ben mozgó n tömegpont kezdeti mozgásállapotának tetszőlegesen kicsiny környezetébe $t \to \infty$ mellett egy valószínűséggel végtelen sokszor tér vissza, azaz majdnem minden $x_0 \in V$ rendelkezik az alábbi, rekurrenciának nevezett tulajdonsággal: minden $\varepsilon > 0$ és $T > 0$ esetén létezik olyan $\tau > T$, hogy $|x(\tau; x_0) - x_0| < \varepsilon$, ahol $x(\tau; x_0)$ az $x = x_0$ kezdeti állapotból a $t = 0$ időpontban induló megoldásnak a $t = \tau$ időpontban felvett értékét jelöli. Ennek az eredménynek a fényében Poincaré egy másik stabilitásfogalmat is használ és az n test együttesének pontosan azokat a mozgásait nevezi (Poisson) stabilnak, amelyek a mai szóhasználat szerint rekurrensek.

Pontrendszerek egyensúlyi helyzetének stabilitását Lord Kelvin és Tait ([19], 202. oldal) így definiálják: „[... a material system ...], when displaced infinitely little in any direction from a particular position of equilibrium, and left to itself, it commences and continues vibrating, without ever experiencing more than infinitely small deviation in any of its parts, from the position of equilibrium, the equilibrium in this position is said to be stable". A huszadik században született matematikus azonnal felismeri ebben a stabilitás ma általánosan elfogadott definíciójának csíráját.

2.1. Definíció. Tekintsük az $\dot{x} = f(x)$ differenciálegyenletet, ahol $f : \mathbf{R}^n \to \mathbf{R}^n$ folytonos függvény. Az egyszerűség kedvéért tegyük fel, hogy minden $x_0 \in \mathbf{R}^n$ esetén az $\dot{x} = f(x)$, $x(0) = x_0$ kezdetiérték-problémának pontosan egy megoldása létezik és hogy ez az $x(\cdot; x_0)$-al jelölt megoldás értelmezve van a teljes $t \geq 0$ félegyenesen. Legyen továbbá $f(0) = 0$. Az $\dot{x} = f(x)$ egyenlet $x_0 = 0$ egyensúlyi helyzete STABIL, ha minden $\varepsilon > 0$ esetén van olyan $\delta > 0$, hogy tetszőleges $|x_0| \leq \delta$ és $t \geq 0$ mellett $|x(t; x_0)| < \varepsilon$.

2.2. Definíció (folytatás). Az $x_0 = 0$ egyensúlyi helyzet ASZIMPTOTIKUSAN STABIL, ha stabil és ezenkívül még létezik olyan $\eta > 0$ is, hogy tetszőleges $|x_0| \leq \eta$ és $t \to \infty$ esetén $|x(t; x_0)| \to 0$.

(Érdemes megjegyeznünk, hogy a stabilitás definíciójában szereplő feltételekből már következik a megoldásoknak a kezdeti feltételektől való folytonos függése. Így az $x_0 = 0$ egyensúlyi helyzet mindenképpen rendelkezik a következő tulajdonsággal: minden $\varepsilon > 0$ és $T > 0$ esetén van olyan $\delta > 0$, hogy tetszőleges $|x_0| \leq \delta$ és $t \in [0, T]$ mellett $|x(t; x_0)| < \varepsilon$. Egyensúlyi helyzet stabilitása tehát azt jelenti, hogy a kezdeti feltételektől való folytonos függés ε-δ megfogalmazása a $[0, T]$ helyett a teljes $[0, \infty)$ intervallumon érvényes.) Abban mindenki egyetértett, hogy egy lineáris rendszer egyensúlyi helyzetét pontosan akkor kell stabilnak hívni, amikor valamennyi megoldás korlátos a $t \geq 0$ időintervallumon. Mindez teljes összhangban van az összes eddigi megfontolással.

2.3. TÉTEL. *Tekintsük az $\dot{x} = Ax$ differenciálegyenletet, ahol A valós elemű n-edrendű négyzetes mátrix*[10]. *Az alábbi tulajdonságok egymással páronként ekvivalensek:*

$(i)_S$ *az $\dot{x} = Ax$ egyenlet $x_0 = 0$ egyensúlyi helyzete stabil;*

$(ii)_S$ *az $\dot{x} = Ax$ egyenlet megoldásai a $t \geq 0$ időintervallumon (egyenként) korlátosak;*

$(iii)_S$ *az A mátrix valamennyi sajátértékének valós része ≤ 0, a 0 valós részű sajátértékek mindegyikéhez pedig annyi dimenziós sajáltaltér tartozik, amennyi az illető sajátérték multiplicitása a karakterisztikus polinomban.*

2.4. TÉTEL (folytatás). *Egymással páronként ekvivalensek az alábbi tulajdonságok is:*

$(i)_{AS}$ *az $\dot{x} = Ax$ egyenlet $x_0 = 0$ egyensúlyi helyzete aszimptotikusan stabil;*

$(ii)_{AS}$ *az $\dot{x} = Ax$ egyenlet minden megoldása $t \to 0$ mellett az $x_0 = 0$ egyensúlyi helyzethez tart;*

$(iii)_{AS}$ *az A mátrix valamennyi sajátértékének valós része < 0.*

Az előző tétel vezet tovább Ljapunov „stabilitás linearizálással" tétele felé.[11]

2.5. TÉTEL (Ljapunov (1892)). *Tekintsük az $\dot{x} = f(x)$ differenciálegyenletet, ahol $f : \mathbf{R}^n \to \mathbf{R}^n$ sima*[12] *függvény. Tegyük fel, hogy $f(0) = 0$ és hogy az $A = f'(0)$ deriváltmátrixra teljesül a $(iii)_{AS}$ feltétel. Ekkor az $\dot{x} = f(x)$ egyenlet $x_0 = 0$ egyensúlyi helyzete aszimptotikusan stabil.*

A „stabilitás linearizálással" módszer általánosan ismert volt, azt azonban a Weierstrass-féle analitikus szigorúság talaján felnőtt matematikusok – közöttük Fejér is – nem tartották matematikailag megalapozottnak. Teljesen igazuk volt: nem tudtak és nem is tudhattak Ljapunov 1892-es korszakalkotó felfedezéséről. Az okokat röviden taglaljuk írásunk Függelékében, ahol egyúttal kitérünk Ljapunov munkásságának általános értékelésére is. Biztosak vagyunk abban, hogy ez a Függelék az olvasók többsége részére számos meglepetést tartalmaz.

[10] Az A, mint mátrix jelölés még Fejér idejében sem volt szokásos: a meghatározó elemeket kiírták a megfelelő kettős indexekkel, a közbülső sorok és oszlopok elemeit pedig egyszerűen kipontozták.

[11] A Watt-féle centrifugális regulátor differenciálegyenletrendszerét Maxwell írta fel és vizsgálta meg 1868-ban. A magasabbfokú tagok elhagyásával linearizált az ($\omega = \omega_0$ állandó szögsebességnek megfelelő) egyensúlyi helyzet körül és meghatározta, hogy a paraméterek mely értékeire teljesül az aszimptotikus stabilitás $(iii)_{AS}$ feltétele. Érdemes felidéznünk, hogy a $(iii)_{AS}$ tulajdonság megfogalmazható az A mátrix karakterisztikus polinomjának együtthatóira vonatkozó feltételrendszerként is. Ezt az úgynevezett Routh kritériumot a 19. század végének mérnökei egyre gyakrabban használták nemlineáris rendszerek egyensúlyi helyzeteinek stabilitásvizsgálatára is.

[12] Az $f \in C^2$ feltevés bőven elegendő. Ljapunov azt tette fel, hogy az f függvény analitikus. A nem-analitikus jobb-oldalú közönséges differenciálegyenletekre vonatkozó exisztencia- és unicitástételek csak a XIX. század utolsó évtizedében születtek meg. Akárcsak Ljapunov, Fejér is szinte egész életében hűséges maradt az analitikus függvényekhez. Absztrakt tereket is csak nagyritkán használtak.

2.2. A stabilitás fogalma Fejér olvasatában

Az n-test probléma egyszerre Lagrange és Poisson stabil megoldásait Fejér Poincaré nyomán „astronomiailag stabilis"-aknak nevezi. Habilitációs értekezése ezzel a definícióval kezdődik.

Az n-test probléma ma is hihetetlenül nehéz, és a bonyolult dinamikai viselkedések hosszú sorának megléte bizonyítható benne [1], [2]. Tanulmányozása meghatározó volt a a dinamikai rendszerek XX. századi elmélete legkeményebb módszereinek kialakulása során. Száz évvel ezelőtt még csak sejtették, mennyire összetett lehet annak a testnek/bolygónak a mozgása, amelynek a többiekhez viszonyított helykoordinátái nem maradnak korlátosak. Fejér – az $m_3 = 0$ választás által jellemzett (Nap, Jupiter, ideális kisbolygó) úgynevezett korlátozott 3-test problémában – Jacobi, Hill, Poincaré, Charlier, G. Darwin, Bohlin, Kobb és Levi-Civita gondolatait ismertetve érvel egy ilyen tulajdonságú „kéményszerű" (ma úgy mondanánk, a Hill-féle felület szingularitásán át divergáló) megoldás létezése mellett.

A stabilitás Fejér habilitációs előadásában szereplő második definíciója minimális változtatásokkal az akkorra már alighanem folklórnak számító „minden $\varepsilon > 0$ esetén van olyan $\delta > 0$, hogy tetszőleges $\tilde{x}_0 \in \mathbf{R}^n$, $|\tilde{x}_0 - x_0| \leq \delta$ és $t \geq 0$ mellett $|x(t;\tilde{x}_0) - x(t;x_0)| < \varepsilon$" formulára egyszerűsíthető. Csatlakozva Klein és Sommerfeld [14] véleményéhez[13], Fejér ezt a definíciót – amennyiben x_0 nem egyensúlyi helyzet – megengedhetetlenül szűknek tartja. Konkrét mechanikai példákon keresztül bemutatja, hogy a definíciót érdemes módosítani a parciális és az orbitális típusú stabilitásfogalmak irányába, de egyik változtatást sem érzi igazán meggyőzőnek. A bőség zavarával szembekerülve így foglal állást: *„A stabilitás fogalmának még a tömegpontrendszerek mechanikájának keretén belül is nagyon különböző tartalmat szokás adni. Azon vitatkozni, hogy ezen különböző definicziók között melyik a legjobb – nem lehet; a stabilitás ugyanis, mint populáris fogalom, néhány – tőle igazán elválaszthatatlan jegyen kívül annyira határozatlan, továbbá annyira relatív, hogy a fennforgó viszonyok különfélesége szerint egymástól lényegesen különböző definicziókat állíthatunk fel, a nélkül, hogy a stabilitás populáris fogalmával ellentmondásba jutnánk".*

Talán nem erőltetett párhuzamot vonnunk a „kaotikus" melléknév mostani valamint a „stabil" melléknév száz évvel ezelőtti használata között. Jóllehet a kaotikusság Devaney-féle definíciója egyre inkább kiszorítani látszik a többit, be kell vallanunk, hogy a matematikusok mindennapi szóhasználatában a „kaotikus" egy-

[13] A német algebrista Klein (aki egyéb munkáiban a differenciálegyenleteket csak úgy emlegeti, mint bizonyos transzformációcsoportok generátorait), monumentális esettanulmányt [14] készített a pörgettyűről, erről a John Herschel szép szavával élve, „philosophical instrument"-ről [talán bizony a cirkálók ágyúkamráit kellett stabilizálni ... igen, azokat] – a konkrétnak adva a prioritást az általánossal szemben, az angol empirizmus legjobb hagyományainak megfelelően. Fejér – habilitációs előadásának számos részlete rá a bizonyíték – nagyon jól ismerte a második kötet 342–374. oldal közötti részeit. Aligha járunk messze az igazságtól, ha feltételezzük, Fejér számára Felix Klein hatása és nézőpontja meghatározó volt a stabilitáselmélet szempontjából. A Lagrange–Dirichlet-tétellel kapcsolatban Fejér [7], [8], [9] Bohl, Hadamard, Hamel, A. Kneser és Painlevé munkái mellett egyébként Liapounoff egy francia nyelvű (nagy felfedezéseihez képest meglehetősen partikuláris) cikkére is hivatkozik.

szerűen a „roppant bonyolult" szinonimája. Felix Klein már idézett megjegyzésének értelmében a „stabil" annak idején egyszerűen a „rendezett", vagy legalábbis a „nem csúnya" biztonságérzetét sugározta.

2.3. Fejér példája tömegpont egyensúlyának stabilitásáról ellenálló közegben

Habilitációs előadásában [8] (akkor szép magyar kifejezéssel magántanári próbaelőadásnak nevezték) Fejér Lipót két problémáról beszélt: a már említett n-test-problémáról, és tömegpont egyensúlyának stabilitásáról ellenálló közegben. Ez utóbbi kérdéskörnek egy külön dolgozatot [7] is szentelt, amelyben leírja a habilitációs előadásban ismertetett saját eredményei (egy kisebb mértékű általánosításának) bizonyítását is. A most kezdődő alfejezetben végig ezzel a dolgozattal foglalkozunk.

Jelölje x, y, z egy egységnyi tömegű tömegpont koordinátáit az $\mathbf{R}^3$ térben. Tegyük fel, hogy a pontra egy konzervatív erő hat, amelynek $U : \mathbf{R}^3 \to \mathbf{R}$ a potenciálfüggvénye, és a mozgás ellenálló közegben történik. Az ellenállásból eredő erő

$$\left(-f(v)\frac{x'}{v}, -f(v)\frac{y'}{v}, -f(v)\frac{z'}{v}\right)$$

alakú, ahol (x', y', z') a sebességvektor (vagyis $(\cdot)'$ az idő szerinti differenciálhányadost jelöli), v a sebességvektor hossza, $f : [0, \infty) \to [0, \infty)$ folytonos függvény, és $f(v) > 0$, ha $v > 0$. A mozgást Newton II. axiómájának megfelelően az

$$x'' = \frac{\partial U}{\partial x}(x, y, z) - f(v)\frac{x'}{v} \tag{2}$$

$$y'' = \frac{\partial U}{\partial y}(x, y, z) - f(v)\frac{y'}{v}$$

$$z'' = \frac{\partial U}{\partial z}(x, y, z) - f(v)\frac{z'}{v}$$

egyenletek írják le. Arra vagyunk kíváncsiak, hogy adott (x_0, y_0, z_0) helyzetből adott (x_0', y_0', z_0') kezdősebességgel kiindulva hogyan mozog a pont időben és térben. A probléma megoldását az a $t \mapsto \big(x(t), y(t), z(t)\big)$ függvény adja, amely eleget tesz a differenciálegyenleteknek és az

$$x(0) = x_0, \quad y(0) = y_0, \quad z(0) = z_0,$$
$$x'(0) = x_0', \quad y'(0) = y_0', \quad z'(0) = z_0'$$

kezdeti feltételeknek. Ismeretes, hogy ez a rendszer nem integrálható típusú, ami durván szólva azt jelenti, hogy nem létezik formula a mozgást leíró függvényekre. Ezért sokszor megelégszünk a mozgások aszimptotikus viselkedésének, vagyis

az őket leíró függvények t nagy értékeire mutatott tulajdonságainak leírásával. Fejér sokkal szebben fogalmaz: *„... az ellenálló közegben történő mozgás lefolyására, vagy, mi több, annak véglefolyására nézve keresünk előleges tájékoztatást".*

Az aszimptotikus tulajdonságok közül mind elméleti, mind gyakorlati szempontból a legfontosabb a stabilitás. Az az időszak a stabilitás matematikai elméletének hőskora volt, akkor kezdtek letisztázódni maguk az alapfogalmak. A habilitációs előadásból [8] kiderül, hogy Fejér a ma Ljapunov nevéhez kötött stabilitásfogalommal dolgozott. Tegyük fel, hogy $\operatorname{grad} U(0,0,0) = \vec{0}$, vagyis hogy az $x = y = z = 0$ pont egyensúlyi helyzet, és alkalmazzuk a Ljapunov-féle 2.1. Definíciót az $x = y = z = 0$, $x' = y' = z' = 0$ egyensúlyi állapot stabilitásának meghatározására. Ehhez először vezessük be a ma használatos vektoros írásmódot. Jelölje $\vec{r} = (x, y, z) \in \mathbf{R}^3$ a mozgó pont helyvektorát, $\vec{v} = \vec{r}' = (x', y', z') \in \mathbf{R}^3$ pedig a sebességvektort. Jelölje továbbá r, illetve v az $\vec{r}$, illetve a $\vec{v}$ vektor hosszát:

$$r := (x^2 + y^2 + z^2)^{1/2}, \quad v := \left((x')^2 + (y')^2 + (z')^2\right)^{1/2}.$$

Így a (2) másodrendű differenciálegyenlet-rendszer a tömörebb

$$\vec{r}\,'' = \operatorname{grad} U(\vec{r}) - f(v)\frac{\vec{v}}{v} \tag{3}$$

alakot ölti, amely ekvivalens a hat egyenletből álló

$$\vec{r}\,' = \vec{v}, \quad \vec{v}\,' = \operatorname{grad} U(\vec{r}) - f(v)\frac{\vec{v}}{v} \tag{4}$$

elsőrendű rendszerrel. A 2.1. Definíció értelmében az $\vec{r} = \vec{0}$, $\vec{v} = \vec{0}$ egyensúlyi állapot *stabil*, ha bármely $\varepsilon > 0$ számhoz létezik olyan $\delta(\varepsilon) > 0$, hogy ha $r_0^2 + v_0^2 < \delta(\varepsilon)^2$, akkor $r(t; \vec{r}_0, \vec{v}_0)^2 + v(t; \vec{r}_0, \vec{v}_0)^2 < \varepsilon^2$ tetszőleges $t \geq 0$ esetén.

Fejér a [7] dolgozatot a híres Lagrange–Dirichlet-tétel ismertetésével kezdi, amely a konzervatív esetről (nincs ellenállás, vagyis $f(v) \equiv 0$) azt mondja, hogy *„valamely egységnyi tömegű P tömegpontnak G egyensúlyi helyzete* stabilis, *ha az $U(x, y, z)$ potenciálfüggvénynek a G pontban izolált maximuma van".*[14]

Fejér itt nem fejti ki, mit ért stabilitáson, de a kijelentést követő paragrafus szó szerint idézve a következőképpen szól: *„Ha ekkor a G pont körül, mint középpont körül, két elegendő kicsiny, de különben tetszőleges R_1 és R_2 ($R_1 < R_2$) sugarú gömböt írunk, akkor a következőt állíthatjuk:*

Ha a P tömegpont mozgása folyamán egyszer, valamely t_0 időpontban, a kisebbik, R_1 sugarú gömb belsejében van, akkor a tömegpont nemcsak minden későbbi *időben* marad *a nagyobb, R_2 sugarú gömb belsejében, hanem minden korábbi időben*

[14] A ma szokásos terminológia az U helyett a $-U$ függvényt használja és potenciális energiának nevezi; így a Lagrange–Dirichlet-tétellel kapcsolatban potenciálgödörről, izolált minimumról beszél. A továbbiakban is Fejér szóhasználatát és (amennyire csak lehet) jelölésrendszerét követjük.

is az R_2 sugarú gömb belsejében volt – *föltéve, hogy a t_0 időpontban a sebessége kisebb, mint egy az R_1, R_2 sugaraktól függő mennyiség. Az izolált maximum esetében tehát stabilitás van a »jövőre« és »múltra« vonatkozólag".*

A 2.1. Definíció értelmében vett stabilitás tényleg következik az izolált (vagyis szigorú) maximumra vonatkozó feltételből, ahogyan azt Dirichlet gyönyörű bizonyításában megmutatta. Érdemes viszont megjegyezni (és a zseniális Fejér Lipótnak ez az apró tévedése is jól mutatja a stabilitáselmélet akkori képlékeny voltát), hogy a helyes megállapítást követő kiegészítő „magyarázat" (definíció?) nem állja meg a helyét, ugyanis az abban az R_1 és R_2 gömbbel megfogalmazott tulajdonság *nem következik* az izolált maximum feltételéből. Erről könnyen meggyőződhetünk, ha olyan gömbszimmetrikus U potenciálfüggvényt tekintünk, amelynek a G pontban szigorú maximuma van ugyan, de a G tetszőlegesen kicsiny környezetében létezik olyan G körüli koncentrikus gömbgyűrű, ahol U állandó értéket vesz fel. (Érdemes elgondolkodni azon, hogy mely U függvények esetén teljesül a Fejér által megfogalmazott (a Ljapunov-félénél erősebb) stabilitási tulajdonság.)

Fejér megemlékezik a Lagrange–Dirichlet-tétel megfordításának problémakörében elért eredményekről és idézi az akkor ismert legáltalánosabb tételt[15]: *„Ha a potencziálfüggvénynek a G egyensúlyi helyzetben izolált minimuma van, és ha a potencziálfüggvénynek minimális volta a Taylor-féle sor legalacsonyabb rendű tagjai révén jut fölszínre, akkor az egyensúlyi helyzet labilis."* Ezután rátér a dolgozat fő témájára: milyen hatással van a közegellenállás az egyensúlyi helyzet stabilitására, instabilitására?

Általános természeti törvényként ismeretes az a tény, hogy a közegellenállás stabilizáló hatású. Ezt kicsit pontosítja a következő tétel:

2.6. TÉTEL (Fejér [7]). *Ha a potencziálfüggvénynek a G egyensúlyi helyzetben izolált maximuma van, akkor a jövőre nézve stabilitás van, míg a múltra vonatkozólag labilitás is lehetséges.*

Azt is észrevette, hogy a közegellenállás növeli a stabilitás fokát abban az értelemben, hogy a mozgás időnként nagyon lelassul, ahogyan ő fogalmaz, a sebesség abszolút értéke, *„v végtelen sokszor tetszőlegesen kicsinynyé lesz"*. Ezt a tulajdonságot ma a

$$\liminf_{t\to\infty} v(t) = 0 \tag{5}$$

formulával fejezhetjük ki.

Megállapítja továbbá: *„Az ellenállás a labilitás esélyét csökkenti."* Ezért érdekesebbnek tartotta az instabilitásra gyakorolt hatás tanulmányozását, nevezetesen annak a kérdésnek a megválaszolását, hogy ha olyan potenciálfüggvényből indulunk ki, amelynek esetén ellenállás hiányában az egyensúlyi helyzet instabil, akkor

[15] A probléma még ma sincs teljesen lezárva. A nagy áttörést Kozlov és Palamodov dolgozata [15] jelentette, amelyben a tétel megfordíthatóságát analitikus potenciálfüggvények egy igen általános osztályára bizonyították.

megmarad-e az instabilitás. Pontosabban, milyen potenciálfüggvény, milyen közegellenállás esetén marad meg az instabilitás?

2.7. TÉTEL (Fejér [7]). *Tegyük fel, hogy „a potenciálfüggvénynek az egyensúlyi helyzetben izolált minimuma van, mely tulajdonsága a potenciálfüggvénynek a Taylor-féle sor legalacsonyabb rendű tagjai által lesz nyilvánvalóvá.*

Szigorúan ki lehet most már mutatni, hogy ha $f(v)/v$ a v kicsiny értékei mellett egy véges felső határnál, M-nél kisebb, akkor az egyensúlyi helyzet labilis."

Kettős szándékkal tartottuk meg az eredeti fogalmazást. Egyrészt, be szeretnénk mutatni a Fejér által használt szép magyar matematikai műnyelvet, másrészt érzékeltetni a nagy tudósnak azt a megértést könnyítő képességét, amellyel bonyolult tételeket tudott formulák nélkül megfogalmazni.

Fejér a tétellel kapcsolatban felvet egy problémát: *„Ha tehát pl. $f(v)$ olyan erősen konvergál a zérushoz mint v^α, hol $\alpha \geq 1$ – a legtöbbször használatos esetek $\alpha = 1, 2$ ide tartoznak –, akkor az egyensúlyi helyzet labilitása még biztosítva van. Ha azonban az $f(v)$ végtelen kicsinynyé válásának mértéke v^α, hol $0 < \alpha < 1$, vagyis, ha az ellenállás – összehasonlítva az $\alpha \geq 1$ esethez tartozó ellenállással – a v kicsiny értékei mellett* nagy, *akkor a labilitást a most közlendő módszer segítségével már nem tudom kimutatni.* KÉNYTELEN VAGYOK[16] NYÍLTAN HAGYNI AZT A KÉRDÉST, VAJJON AZ ELLENÁLLÁS KÉPES-E EZEN ESETBEN A LABILITÁST TELJESEN MEGSZÜNTETNI?"

A 2.7. Tétel bizonyítása [7] az energia változását leíró *„elevenerő-egyenlet"* (Fejér a latin „vis viva" mintájára próbálkozik meg a „die Energie" magyarításával) és egy, Jacobi-tól származó azonosság ötletes kombinációján alapszik, amelyben $f(v)/v$ korlátossága lényegesen ki van használva.

Úgy gondoljuk, Fejér Lipót munkássága előtt akkor tisztelgünk méltóképpen, ha érzékeltetjük a jelen írásban azt is, hogy az eltelt száz év alatt az utódok hogyan fejlesztették tovább azt az elméletet, amelynek elindításában Ljapunovval, Poincaré-val és a többi más nagy tudóssal együtt ő is részt vett, és amit ma a DIFFERENCIÁLEGYENLETEK KVALITATÍV ELMÉLETE, vagy kicsit általánosabban DINAMIKUS RENDSZEREK néven szokás emlegetni. Fejér kiváló problémafelismerő érzékét dícséri, hogy ezeknek az elméleteknek a történetén végighúzódik az általa választott téma, az ellenállás hatásának vizsgálata. Most megadjuk Fejér két idézett tételének továbbfejlesztett változatát. A 2.9. Tétel egyben választ ad a Fejér által nyitva hagyott problémára is.

2.8. TÉTEL. (Salvadori [18]) *Tegyük fel, hogy*

(i) *az U potenciálfüggvénynek helyi maximuma van G-ben;*

(ii) *G izolált egyensúlyi helyzet.*

Ekkor a (3) rendszer G egyensúlyi helyzete aszimptotikusan stabil.

Ez a tétel azt fejezi ki, hogy a stabilitás megmaradásánál és a (5) tulajdonságnál jóval több teljesül: a rendszer aszimptotikusan visszatér az egyensúlyi állapotba,

[16] dőlt betűs kiemelés az eredetiben

azaz

$$\lim_{t\to\infty} \vec{r}(t) = G, \qquad \lim_{t\to\infty} v(t) = 0.$$

2.9. TÉTEL (Salvadori [18]). *Tegyük fel, hogy*

(i) *az U potenciálfüggvénynek nincs helyi maximuma G-ben;*

(ii) *G egy környezetében az $\{(x,y,z) \in \mathbf{R}^3 \mid U(x,y,z) < U(G)\}$ halmaz nem tartalmaz egyensúlyi helyzetet.*

Ekkor a (3) rendszer G egyensúlyi helyzete instabil.

Mindkét tétel bizonyítása ugyanazon a technikán alapszik, amelynek kulcsa a megoldások pozitív határhalmazának fogalma és az úgynevezett LaSalle-féle (1960) invariancia-elv. (Ez utóbbi lényegében véve nem más, mint az orosz nyelvű irodalomban a Barbasin-Kraszovszkij-tétel (1952) néven emlegetett állítás.) A bizonyítások – n szabadsági fokú rendszerekre – megtalálhatók a [17] monográfia III. fejezetének 5. pontjában. A módszernek létezik kiterjesztése olyan rendszerekre is, amelyekben a kinetikai energia és az erők explicit módon függhetnek az időtől is (nem-autonóm, avagy instacionárius rendszerek [12, 13]).

Mivel Fejér csak a 2.7. Tételre adott részletes bizonyítást és ezzel kapcsolatban vetett fel problémát, mi is csak a 2.9. Tételt bizonyítjuk. Olyan bizonyítást adunk, amely csak implicit módon használja a Fejér után kialakult fogalmakat — magát a bizonyítást akár Fejér is leírhatta volna ebben a formában.

A 2.9. Tétel bizonyítása. 1. Először vizsgáljuk meg, hogyan viselkedik a mozgások során az

$$E(\vec{r},\vec{v}) := \frac{1}{2}v^2 - U(\vec{r})$$

teljes mechanikai energia. Ehhez számoljuk ki az E iránymenti deriváltját a (3) rendszer szerint:

$$(6)\quad E'(\vec{r},\vec{v}) = \vec{v}\cdot\vec{v}\,' - \operatorname{grad}U(\vec{r})\cdot\vec{v} = \vec{v}\cdot\operatorname{grad}U(\vec{r}) - f(v)v - \operatorname{grad}U(\vec{r})\cdot\vec{v} =$$

$$= -f(v)v \le 0.$$

Tehát az energia csökken a mozgások mentén és

$$(7)\qquad E'(\vec{r},\vec{v}) = 0 \Leftrightarrow v = 0.$$

2. Feltehetjük, hogy $G = (0,0,0)$, és $U(0,0,0) = 0$. A továbbiakban indirekt utat választunk: tegyük fel, hogy az $\vec{r} = \vec{0}$ egyensúlyi helyzet stabil. Legyen $\varepsilon > 0$ tetszőlegesen rögzített, és vegyük a (4) rendszer egyensúlyi helyzetének stabilitási definíciójában az ε-hoz tartozó $\delta(\varepsilon) > 0$ számot. Az (i) feltétel következtében a

$$H := \{\vec{r} \in \mathbf{R}^3 \mid U(\vec{r}) < 0\}$$

halmazban van olyan $\vec{r}_0$ pont, amelyre $r_0 < \delta(\varepsilon)$ teljesül. Tekintsük az

$$\vec{r}(t) := \vec{r}(t;\vec{r}_0,\vec{0}), \quad \vec{v}(t) := \vec{v}(t;\vec{r}_0,\vec{0})$$

mozgást. Az ε és $\delta(\varepsilon)$ közötti összefüggés miatt

$$r^2(t) + v^2(t) < \varepsilon^2 \quad (t \geq 0). \tag{8}$$

Tehát – az 1. pontot figyelembe véve – az energia a tekintett mozgás mentén csökkenő és alulról korlátos, vagyis létezik a

$$\lim_{t\to\infty} E\big(\vec{r}(t), \vec{v}(t)\big) =: e_* \leq U(\vec{r}_0) < 0 \tag{9}$$

határérték. Megmutatjuk, hogy a (8)–(9) tulajdonságok ellentmondáshoz vezetnek.

3. Most konstruálunk egy r_* pontot a H halmazban, amelyről meg fogjuk mutatni, hogy egyensúlyi helyzet.

Tekintsünk egy tetszőleges $\{t_k\}_{k=1}^{\infty}$ ($\lim\limits_{k\to\infty} t_k = \infty$) sorozatot. Az $\mathbf{R}^6$ tér origó körüli ε-sugarú gömbje kompakt, tehát (8) következtében feltehetjük, hogy a

$$\lim_{k\to\infty} \vec{r}(t_k) =: \vec{r}_*, \quad \lim_{k\to\infty} \vec{v}(t_k) = \vec{v}_* \tag{10}$$

határértékek léteznek. (9) szerint

$$E(\vec{r}_*, \vec{v}_*) = \lim_{k\to\infty} E\big(\vec{r}(t_k), \vec{v}(t_k)\big) = e_* < 0,$$

és $v_*^2 \geq 0$, ezért $U(\vec{r}_*) < 0$, vagyis $\vec{r}_* \in H$. A következő pontban bebizonyítjuk, hogy $\vec{r}_*$ egyensúlyi helyzet, ami ellentmond az (ii) feltételnek.

4. Indítsunk mozgást az $\vec{r}(0) = \vec{r}_*$, $\vec{v}(0) = \vec{v}_*$ kezdeti feltételekkel. Be fogjuk látni, hogy $\vec{v}(t; \vec{r}_*, \vec{v}_*) \equiv 0$ $(t \geq 0)$. Ekkor $\vec{r}(t; \vec{r}_*, \vec{v}_*) = \vec{r}(t; \vec{r}_*, \vec{0}) \equiv \vec{r}_*$ $(t \geq 0)$, azaz $\vec{r}_*$ valóban egyensúlyi helyzet, és készen is vagyunk.

Legyen most $t > 0$ tetszőlegesen rögzített időpillanat. A megoldások a kezdeti értékektől folytonosan függnek, tehát (9) és (10) szerint

$$E\big(\vec{r}(t; \vec{r}_*, \vec{v}_*),\ \vec{v}(t; \vec{r}_*, \vec{v}_*)\big) = \lim_{k\to\infty} E\big(\vec{r}(t + t_k; \vec{r}_0, \vec{0}),\ \vec{v}(t + t_k; \vec{r}_0, \vec{0})\big) = e_*.$$

Mivel t tetszőleges volt, azt kaptuk, hogy az E mechanikai energia állandó az $(\vec{r}_*, \vec{v}_*)$ állapotból indított mozgás mentén. De ez azt jelenti, hogy ezen mozgás mentén minden időpillanatban teljesül az $E' = 0$ feltétel, ami (7) szerint csak úgy lehet, hogy $v(t; \vec{r}_*, \vec{v}_*) \equiv 0$.

Ezzel a bizonyítást befejeztük.

3. Epilógus

1919-ben a kolozsvári egyetem román fennhatóság alá került, ezért a tanárok nagy része – köztük Haar Alfréd és Riesz Frigyes – Szegedre ment, ahol egy új egyetem alapjait rakták le. Közben a magyar „matematikusokat gyártó szerkezet" tovább működött. Kevéssel a háború befejezése után a tudomány olyan művelői jelentek meg Magyarországon, mint Kármán Tódor, Pólya György, Szegő Gábor, Neumann János és mások. Ők viszont nem a Kárpát-medence keleti régiói felé vették útjukat, mint néhány évvel előttük járó társaik, hanem ... de ez már egy másik történet. A mostani véget ért.[17]

Hivatkozások

[1] Arnold, V. I., *A mechanika matematikai módszerei*, Műszaki Könyvkiadó, Budapest, 1985.

[2] Diacu, F. és Holmes, P., *Égi találkozások. A káosz és a stabilitás eredete*, Akkord, Budapest, 2003.

[3] Fejér, L., A Poisson-féle integrál elméletéhez, *Mat. Term. Ért.*, **19** (1901), 322–325.

[4] Fejér, L., Über zwei Randwertaufgaben, *Math. u. Naturwissenschaftliche Berichte aus Ungarn*, **19** (1903), 329–331.

[5] Fejér, L., Az Ostwald-féle mechanikai elvről, *Mat. Term. Ért.*, **23** (1905), 155–176.

[6] Fejér, L., Das Ostwaldsche Prinzip in der Mechanik, *Math. Ann.*, **59** (1906), 422–436.

[7] Fejér, L., Tömegpont egyensúlya ellenálló közegben, *Mat. Term. Ért.*, **24** (1906), 109–116.

[8] Fejér, L., Stabilitási és labilitási vizsgálatok a tömegpontrendszer mechanikájában, *Mat. és Fiz. Lapok*, **15** (1906), 152–172.

[9] Fejér, L., Über Stabilität und Labilität eines materiellen Punktes im widerstrebenden Mittel, *Journal für die reine und angew. Math.*, **131** (1906), 216–223.

[10] Fejér, L., Sur le calcul des limites, *Comptes Rendus*, **143** (1906), 957–959.

[11] Fejér, L., Über die Eindeutigkeit der Lösung der linearen partiellen Differentialgleichungen zweiter Ordnung, *Math. Zeitschr.*, **1** (1918), 70–79.

[12] Hatvani László, Nem-autonóm differenciálegyenlet-rendszerek megoldásainak stabilitása és parciális stabilitása, *Alk. Mat. Lapok*, **5** (1979), 1–48.

[13] Hatvani László, Közönséges differenciálegyenletek megoldásainak stabilitásáról mechanikai alkalmazásokkal, *Alk. Mat. Lapok*, **15** (1990), 1–90.

[14] F. Klein und A. Sommerfeld, *Über die Theorie des Kreisels I-IV*, Teubner, Leipzig, 1897–1910.

[15] Kozlov, V. V. and Palamodov, V. P., On asymptotic solutions of the equations of classical mechanics, *Soviet Math. Dokl.*, **25** (1982), 335–339.

[17] Elképzelhetetlennek tartjuk, hogy a legendás tanár hírében álló Fejér Lipót kolozsvári működése alatt ne készített volna az általa oktatott tárgyakhoz egyetemi jegyzeteket. Ezekből azonban – legalábbis a kolozsvári egyetem matematikai könyvtárában – egyetlen egy sem lelhető fel.

[16] Prékopa András, Farkas Gyula élete és munkásságának jelentősége az optimalizálás elméletében, *Új utak a magyar operációkutatásban. In memoriam Farkas Gyula* (szerk. Komlósi Sándor és Szántai Tamás), Dialóg Campus Kiadó, Pécs, 1999.

[17] Rouche, N., Habets, P. és Laloy, M., *Stabilitáselmélet. A Ljapunov-féle direkt módszer,* Műszaki Könyvkiadó, Budapest, 1984.

[18] Salvadori, L., Sull' estensione ai sistemi dissipativi del criterio di stabilitá del Routh, *Ricerche Mat.,* **15** (1966), 162–167.

[19] Thomson, W. (Lord Kelvin) and Tait, P. G., *Treatise on Natural Philosophy,* Clarendon Press, Oxford, 1867.

4. Függelék. Ljapunov szerepe a stabilitáselmélet létrehozásában

Ljapunov, amint arra a 2.1. alfejezetben már kitértünk, felzárkóztatta a stabilitás elméletét a matematika akkori fejlettségi szintjére. Ami legalább ennyire fontos, három területen a jövőt is előkészítette.

1) Az általa kidolgozott és később őróla elnevezett módszer[18] – amelynek felhasználásával sikerült a 2.5. Tételt is bizonyítania – minden további stabilitásvizsgálat (1930-tól a Szovjetunióban, 1960-tól kezdve Nyugaton is) leghatékonyabb eszközének bizonyult és számtalan alkalmazást nyert az irányítás- és a szabályozáselméletben is.

2) Befejezetlenül maradt írásaiban Ljapunov kísérletet tett a centrális sokaság tételének kimondására és bizonyítására. Viktor Pliss,. aki végülis a hatvanas évek közepén sikerrel járt, sokat profitált Ljapunov egy kéziratából, amely egy konkrét nemlineáris mechanikai feladat egyensúlyi helyzetének stabilitásvizsgálata kapcsán a három sajátérték a képzetes tengelyen problematikájával foglalkozott.

3) Ljapunov (1884) Poincaréval egyidejűleg, ugyanazzal a gondolatmenettel – egy térbeli integrációs tartomány határa szerinti sorfejtés első tagjainak vizsgálatával – kimutatta, hogy az egyre gyorsabban forgó bolygók/csillagok (folyadékok, amelyek részecskéit csak a gravitációs erő tartja össze) alakja „körte" is lehet. Visszatérve e témakörhöz, egy összességében több mint ezer oldalas (1906–1916) cikksorozatban azt is sikerült plauzibilissá tennie, hogy ez a – ma úgy mondanánk, hogy a forgási ellipszoidból az egyenlítő mentén szimmetriasértő bifurkációval keletkező – „körte" instabil[19].

[18] Természetesen ennek is voltak előzményei: az, hogy az energia bizonyos disszipatív rendszerekben a trajektóriák mentén csökkenve „Ljapunov-függvény"-ként viselkedik, már 50 évvel korábban, Jacobinál is fontos szerepet játszik.

[19] Poincaré és George Darwin (a biológus Darwin fia, a Királyi Csillagászati Társaság elnöke) ezzel ellentétes véleményt képviseltek, sőt a „körte" stabilitását (tanítványaik munkái által) bizonyítottnak vélték. A „körtés ág" stabilitásán alapult G. Darwin elmélete a kettőscsillagok keletkezésének magyarázatára. Végérvényesen csak jóval mindhármuk halála után derült ki teljes bizonyossággal, hogy a „körte" valóban instabil. A „körte" stabil vagy instabil voltának eldöntése – a funkcionálanalízis eszközeinek szinte teljes hiányában – alighanem még egy Poincaré erejét is meghaladta volna.

Jóllehet nemzetközi elszigeteltsége soha nem volt teljes, Ljapunov nagyfokú nyelvi és földrajzi elkülönültségben élt. „Körtés" dolgozatait leszámítva, stabilitáselméletből egyetlen nem-orosz nyelvű cikket publikált. Felix Klein [14] egy könyvelő pontosságával említi Ljapunov ezen cikkét, sőt egy lábjegyzetben még azt is megjegyzi, hogy ugyanennek a szerzőnek további, orosz nyelven írott dolgozatai is vannak a stabilitás témakörében.[20]

A Ljapunovra – csakúgy mint a stabilitás régi történetére – vonatkozó információk jelentős részét N.D. Mojszejev 1949-ben megjelent *Ocserki po razvityiji tyeoriji usztojcsivosztyi* című könyvéből vettük. Bátor és kiegyensúlyozott írás[21], amely beszámol Ljapunov halálának körülményeiről is. Ljapunov nem érezte jól magát a forradalom utáni Petrograd zavaros világában. Feleségével együtt Ogyesszába ment, de továbbjutniuk nem sikerült. Az asszony tüdőgyulladásban megbetegedett és meghalt. Alekszandr Mihajlovics még aznap, 1918. október 31-én fejbe lőtte magát és három napra rá meghalt.

GARAY BARNABÁS
MŰEGYETEM, MATEMATIKA INTÉZET
1521 BUDAPEST, MŰEGYETEM RKP. 3-9.
garay@math.bme.hu

HATVANI LÁSZLÓ
SZEGEDI EGYETEM, BOLYAI INTÉZET,
6720 SZEGED, ARADI VÉRTANÚK TERE 1.
hatvani@math.u-szeged.hu

KOLUMBÁN JÓZSEF
BABES-BOLYAI TUDOMÁNYEGYETEM,
MATEMATIKA ÉS INFORMATIKA KAR
STR. KOGALNICEANU NR. 1
RO-400084 CLUJ–NAPOCA
kolumban@math.ubbcluj.ro

[20]A groteszk az egészben az, hogy mindezt könyvének abban a fejezetében teszi, amelyben egyensúlyi helyzetek linearizálással történő stabilitásvizsgálatáról, mint matematikailag megalapozatlan módszerről értekezik. Pedig a lényeg – nevezetesen a Ljapunov által már korábban bizonyított 2.5. Tétel – ott rejtőzik a lábjegyzetben ([14], 374. oldal). Abban az időben még természetes volt, hogy egy göttingeni professzor nem érdeklődik egy harkovi docens cirillbetűs munkái iránt, Ljapunov eredményeit balti matematikusok (akik értettek oroszul) közvetíthették volna a Nyugat felé. Közülük Bohl és Adolf Kneser (a sokkal ismertebb H. Kneser apja) maguk is foglalkoztak stabilitáselmélettel. Hogy miért nem tették? [Minden oroszok cárjának alattvalóiként aligha érezték feladatuknak, hogy az orosz tudomány és kultúra értékeit Nyugaton képviseljék.] A matematikusok 1908-as római kongresszusán Poincaré és Ljapunov egyaránt részt vett. Poincarét itt is nagy udvartartás vette körül. Ljapunov, mint afféle vidéki ember, anélkül ment haza, hogy személyesen találkozhatott volna vele.

[21]A német forrásokat ugyan negligálja, de Sztálin-idézet sincsen benne. Lenintől is csak elvétve idéz ezt-azt. A legérdekesebb a következő: „Poincaré gyermeteg filozófus, de hatalmas fizikus". (Úgy látszik, ez a szavajárása lehetett. A társszerzők még legfiatalabbika is emlékszik egy másik Lenin-idézetre, középiskolai irodalomtanulmányainak idejéből: „Tolsztoj gyermeteg filozófus, de óriási művész".)

LIPÓT FEJÉR HABILITATED AT KOLOZSVÁR 100 YEARS AGO IN STABILITY THEORY

BARNABÁS GARAY, LÁSZLÓ HATVANI, JÓZSEF KOLUMBÁN

Lipót Fejér habilitated 100 years ago at June 23, 1905 at the University of Kolozsvár. His habilitation lecture was devoted to the stability theory of ordinary differential equations with title "A study on stability and instability of the mechanics of mass-point systems". (A German translation of the original Hungarian can be found in *Fejér Lipót Összegyűjtött munkái – Leopold Fejér Gesammelte Arbeiten* (ed. P. Turán), Akadémiai Kiadó, Budapest, 1970.)

In this paper we commemorate Lipót Fejér as well as the Kolozsvár School, being the first internationally recognized professional community of the Hungarian mathematics. Our review on Fejér's habilitation lecture ranges from an introductory chapter surveying the general state of stability theory around 1900, to a short Appendix on the work of A. M. Liapunov. The main emphasis here is put on the mechanical example in Fejér's lecture, which he could not analyze in full detail due to the relative immaturity of topological concepts and techniques that time: his original line of thought can be made complete by LaSalle's invariance principle discovered 50 years later.

Alkalmazott Matematikai Lapok **23** (2006), 191–201.

MEGEMLÉKEZÉS KŐNIG DÉNESRŐL

LIBOR JÓZSEFNÉ

Budapest

1884. 09. 21. - 1944. 10. 19.

Ezen megemlékezés aktualitását tekintve megkésett ugyan egy kicsit, hiszen 2004-ben volt nagy matematikusunk születésének 120. és halálának 60. évfordulója. Rövidebb előadás, illetve poszterbemutató ugyan már készült különböző konferenciákra, mégis úgy gondolom, hogy Kőnig – mint a gráfelmélet egyik megalkotója – megérdemli, hogy ne csak szóban, hanem írásban is megemlékezzünk életéről és munkásságáról. Nem ő volt a legelső, aki gráfelmélettel foglalkozott, hiszen pl. Petersen már 1891-ben megírja [37]-es cikkét és mások is foglalkoztak a témával, de mégis Kőnig Dénes nevéhez köthetjük a gráfelméletnek, mint önálló matematikai diszciplinának a megszületését. Az adatok, információk összegyűjtéséhez felhasználtam Gallai Tibor 1965-ben megjelent, hasonló témájú munkáját, H. W. Kuhn és Egerváry Jenő cikkeit, valamint Kőnig megjelent munkáit, melyeket ajánlok tanulmányozásra a téma iránt érdeklődők számára (lásd irodalomjegyzék).

1. Néhány fontosabb életrajzi adat

Kőnig Dénes 1884. szeptember 21-én született Budapesten, a Műegyetem világhírű matematika professzora, Kőnig Gyula kisebbik fiaként. Édesapja nevéhez fűződik többek között a matematikai logika első monográfiájának megírása. Oktatói és tudományos tevékenysége révén egyaránt az egyik legnagyobb hatású magyar matematikus volt, akinek a halmazelméletben alapvető a számosságokra vonatkozó munkássága. Kőnig Dénes matematikai tehetségét – édesapján kívül – középiskolai tanárai is hamar felismerték. A budapesti Gyakorló Főgimnáziumban Szijjártó Miklós mellett főleg Beke Manó volt rá nagy hatással és segítette tehetsége korai kibontakozását. Ennek köszönhetően első tudományos cikke már középiskolás korában, 1899-ben megjelenik a Matematikai és Fizikai Lapokban [1].

Kőnig Gyula
1849–1913

Beke Manó
1862–1946

Ezen első dolgozatában két szélsőérték-probléma elemi tárgyalását adja. Tehetségét dicséri az Eötvös Loránd matematikai tanulóversenyen 1902-ben elért első díja. Ugyanebben az évben megjelenik első kiadott műve, a *Matematikai mulatságok* első kötete, melyet 1905-ben követ a második rész [I, II]. A könyvek – vagy inkább füzetecskék – legnagyobb érdeme, hogy ez az első magyar nyelven megjelent, színvonalas, szórakoztató matematikai mű. Az előszóban Beke Manó ír méltató szavakat a munkáról.

Egyetemi tanulmányait a budapesti Műegyetemen kezdi, itt négy szemesztert hallgat, majd a göttingeni egyetemen az utolsó öt szemesztert. Göttingenben végighallgatta Minkowskinak az Analysis Situs előadásait az 1904/5-ös tanévben, melyek nagy hatással voltak Kőnig témaválasztására. Ezen előadásokon Minkowski a kétdimenziós felületek topológikus tulajdonságainak jellemzése, a különböző normáltípusok előállítása mellett a négyszín-sejtésnek egy Wernicke-től származó bizonyítását is közölni akarta. Azonban a bizonyítás előkészítő lépéseinek előadása során kiderült, hogy a bizonyítás hibás, így csak az ötszín-tétel került bizonyítás-

ra. Természetesen Kőnig nem ekkor találkozott először kombinatorikus topológikus problémákkal, hiszen a *Matematikai mulatságok* készítésekor már találkozott ilyen jellegű feladatokkal, mégis kutatásainak nagy lendületet adtak az említett előadások.

1907-ben, tanulmányai befejezése után bölcsészdoktori címet szerez geometriai tárgyú értekezésével, melyben az n-dimenziós tér valódi és nem valódi forgásait vizsgálja ([3]). Ugyanettől az évtől a budapesti Műegyetemre kerül gyakornokként és 1944-ig, haláláig az egyetemen tanít. 1908-ban mint tanársegéd, 1910-ben már adjunktusként és 1911-től magántanárként tart előadásokat az alábbi címeken: *Analysis situs*, *Nomographia*, *Valós számok*, *Halmazelmélet*, *Valós számok és függvények*, *Gráfelmélet*. A gráfelmélet különálló tárgyként csak 1927-től van meghirdetve, de Analysis situs előadásain már 1911-től szerepel ez a témakör. Az 1913/14-es tanévtől 1927-ig meghívott előadóként matematika előadásokat tart építész- és vegyészhallgatók számára. Előadásának anyaga könyvben is megjelenik 1920-ban [IV]. 1932-ben műegyetemi rendkívüli tanári címmel ruházzák fel, 1935-től pedig intézeti tanárrá nevezik ki.

Egyéniségével kapcsolatban meg kell említeni kiváló humorát, társaság- és emberszeretetét, mely majd a közéleti munkáját vizsgálva még jobban kirajzolódik előttünk. A fasizmus idején, főleg 1944-ben sokat tett az üldözött matematikusok segítése érdekében. Azonban a nyilas párt hatalomra kerülése, október 15-e megakadályozta tervei végrehajtásában: október 19-én az üldözések elől a halálba menekült, öngyilkos lett.

2. Gráfelméleti munkái

Legjelentősebb alkotásai ezen a területen születtek, több alapvető tétel fűződik a nevéhez. Tételeinél azonban nem kisebb jelentőségűek a gráfelmélet népszerűsítése, meg- és elismertetése terén elért eredményei. Az ő munkássága révén vált a szórakoztató matematika egyik ága, a gráfelmélet a matematikai tudományok elismert új fejezetévé. Gráfelméleti előadásait az alábbi mondattal kezdte: *„A gráfelmélet a matematika legérdekesebb diszciplináinak egyike."* Bár akkor még csak sejthette, mégis kitartóan hitt a gráfelmélet jövőjében. Ma már tudjuk, hogy mennyire igaza volt, hiszen nehezen találnánk olyan ágát a matematikának, melynek ne lenne gráfelméleti vonatkozása. Nap mint nap újabb területeken láthatjuk az alkalmazhatóságát. Csak példaként említem, hogy a jól ismert internetes keresőprogramok a találatok kilistázási sorrendjének összeállításához szintén gráfelméleti eredményeket használnak fel.

2.1. Topologikus gráfelméleti munkák

Első munkáiban Kőnig a gráfok felületekre való felrajzolhatóságával és ezek tulajdonságaival foglalkozott, mely problémákat relatív gráfelmélet néven is ismerünk. Első ilyen jellegű munkája 1905-ben jelenik meg [2], melyben az úgynevezett

háromszín-tételt ismerteti. A tétel azt mondja ki, hogy *ha egy ország síkra rajzolt térképét egyetlen összefüggő határvonal határolja, és az ország megyéi mind valamely vonalszakasz mentén határosak az országhatárral, akkor a megyék megszínezhetők három színnel oly módon, hogy a közös határúak mindig különböző színt kapjanak.*

1911-ben két dolgozatában is [8, 9] azt a kérdést vizsgálja, hogy egy megadott gráf mikor rajzolható fel egy adott nemszámú (genusú) irányítható felületre. Bebizonyítja, hogy *minden gráf felrajzolható egy elég magas nemszámú irányítható felületre*, ezenkívül bevezeti a gráf nemszámának fogalmát. (A nemszám vagy genus p, ha a gráf felrajzolható p nemszámú irányítható felületre, de alacsonyabb nemszámúra nem rajzolható fel.) Foglalkozik továbbá azzal a kérdéssel, hogy hogyan lehet a gráf belső, kombinatorikus tulajdonságaiból a nemszámát megállapítani. Ezen vizsgálatai újabb kutatásokat indukáltak, 1937-ben Vázsonyi Endre egészítette ki és fejlesztette tovább az elért eredményeket, valamint több amerikai matematikus kezdett nemszámokkal kapcsolatos kutatásokba.

2.2. Kombinatorikus (abszolút) gráfelméleti munkák

Kőnig legismertebb eredményei ehhez a területhez kapcsolódnak. Legelőször 1914-ben Párizsban az első matematikai-filozófiai kongresszuson számolt be ilyen tárgyú eredményeiről. Bár az előadásának teljes szövege csak 1923-ban jelenik meg [14], eredményeit determinánselméleti alkalmazásokkal bővítve 1916-ban jelenteti meg [18]. E dolgozatból kiderül, hogy eredményeihez egy halmazelméleti probléma révén jutott. Még 1906-ban édesapja a halmazelmélet ekvivalencia-tételére egy egyszerű, új bizonyítást adott, mely a gráfelmélet fogalmaival igen szemléletessé tehető ([V], 85. oldal). Ezt a módszert kísérelte meg alkalmazni Bernstein következő tételének bizonyítására:

Ha m és n tetszőleges számosságok és k természetes szám, akkor a $km = kn$ egyenlőségből következik az $m = n$ egyenlőség.

A bizonyítás részletesen szerepel a [18]-as irodalomban. Ezen dolgozat egyik fő érdeme, hogy ebben szerepelnek először végtelen gráfokra vonatkozó problémák. Későbbi kutatásai során is mindig vizsgálja, hogy az adott gráfelméleti tétel érvényessége kiterjeszthető-e végtelen gráfokra. Ezt olyan lényegesnek tartotta, hogy könyvének címe is kiemeli [V]. Az említett [18]-as munka fő eredménye az alábbi véges gráfokra vonatkozó tétel:

Minden véges k-adfokú (k természetes szám) reguláris páros gráfnak van elsőfokú faktora.

E nevezetes, talán egyik legszebb gráfelméleti tétel egyik változatát közérthető alakban így fogalmazza meg Kőnig ([V], 175. oldal):

Ha egy táncestélyen minden férfi k nőt és minden nő k férfit ismer, akkor létrehozható olyan párbeosztás, hogy minden párba ismerősök kerüljenek.

Az említett [18]-as dolgozatban ezen tételének végtelen gráfokra való kiterjesztését is kimondja, de a bizonyítást itt csak másodfokú gráfokra tudja elvégezni. Az említett tétel két fontos következményét is közli:

a) *Ha az $n \times n$-es M mátrix elemei nemnegatív egész számok, és minden sorhoz, illetve oszlophoz tartozó elemek összege ugyanaz a pozitív k szám, akkor M k számú $n \times n$-es permutációs mátrix összege.*

b) *Ha egy determináns elemei nemnegatív egész számok és minden sor és oszlop összege ugyanaz a pozitív érték, akkor a determinánsnak van zérustól különböző kifejtési tagja.*

(Ezen két következményt érdemes összevetni G. Birkhoff tételével: minden $n \times n$-es duplán sztochasztikus mátrix $n \times n$-es permutációs mátrixoknak olyan lineáris kombinációja, amelyben az együtthatók nemnegatív számok és az együtthatók összege 1.)

A determinánsok és páros gráfok kapcsolatát Kőnig úgy hozza létre, hogy a determináns minden sorának és oszlopának megfeleltet egy-egy szögpontot, és két szögpontot akkor és csak akkor köt össze éllel, ha a megfelelő sor és oszlop találkozási helyén zérustól különböző elem áll. Tételének alábbi általánosítása azért jelentős, mert elvezet a tétel végtelen gráfokra való kiterjesztéséhez:

Ha egy véges páros gráf minden szögpontjához k-nál nem több él illeszkedik, akkor a gráf élei megszínezhetők k színnel úgy, hogy egyik szögpontba se fusson két egyenlő színű él.

A bizonyításra az egyik leghatásosabb eljárást, az alternáló utak módszerét alkalmazza. Ugyancsak ezzel a módszerrel sikerül a [17]-ben Frobenius tételének egy egyszerű gráfelméleti (kombinatorikus) bizonyítását adnia. A [23]-as dolgozatban megkísérli tételének a) következményét többdimenziós mátrixokra kiterjeszteni. Ez az első olyan dolgozat, amelyben kétdimenziós gráfokkal kapcsolatos kérdések felmerültek. Az 1925–26-os években Valkó Istvánnal közösen sikerül tételének a végtelen gráfokra való kiterjesztését igazolni, melyet a [24]-es dolgozatban publikálnak. A bizonyítás alapgondolata egy olyan általános eljárást szolgáltat, melyet fel lehet használni tételeknek végesről végtelenre való kiterjesztésénél. 1926-ban a [27]-ben közli az eljárás alapját szolgáltató ún. végtelenségi lemmát:

Legyen $P_1, P_2, \ldots$ páronként idegen véges és nemüres ponthalmazoknak egy végtelen sorozata, G pedig egy olyan gráf, amelyben a szögpontok halmaza $\bigcup_{i=1}^{\infty} P_i$, és P_{n+1} $(n = 1, 2, 3, \ldots)$ minden pontját él köti össze P_n valamelyik pontjával. Ekkor létezik G-ben olyan $x_1 x_2 x_3 \ldots$ végtelen út, amelyre $x_i \in P_i$ $(i = 1, 2, 3, \ldots)$.

Ugyanebben a dolgozatban bemutatja két halmazelméleti alkalmazását is a lemmának, majd 1927-ben egy külön dolgozatot [28] szentel a lemma különböző alkalmazásainak.

Szintén 1927-ben megjelenik Mengernek az „n-Kettensatz" néven ismertté vált görbületelméleti cikke. Kőnig észrevette e tétel kombinatorikus vonatkozását, valamint hogy kimaradt a bizonyításból egy kombinatorikus szempontból különösen érdekes eset. Ezt a hiányt pótolta bizonyítással együtt Kőnig, mely eredményről az Eötvös Loránd Matematikai és Fizikai Társulat 1931. március 26-án tartott előadóülésén számolt be (l. [29]). Nevezetes tétele azóta is az ő nevét viseli, és a gráfelmélet egyik legtöbbet idézett tétele:

Páros körüljárású gráfban az éleket kimerítő szögpontok minimális száma megegyezik a páronként közös végpontot nem tartalmazó élek maximális számával.

Szintén Kőnigtől származik a talán még ismertebb mátrixelméleti megfogalmazás:

Bármely mátrixra az oly vonalak (sorok vagy oszlopok) *minimális száma, melyek összességükben az összes el nem tűnő elemeket tartalmazzák, megegyezik az oly el nem tűnő elemek maximális számával, melyek páronként nem feküsznek egy vonalban."*

A bizonyítás ebben az esetben is az alternáló utak módszerével történik. Még ugyanabban az évben Egerváry Jenő a tétel egy egyszerű, új bizonyítását adja, valamint egy érdekes általánosítást [34]. (Ezért fordul elő, hogy a tételt Egerváry–Kőnig néven is említik.) Egerváry munkásságáról részletesebben a [40]-esben olvashat az érdeklődő. A tétel számos további kutatás kiindulópontja lett. Maga Kőnig is, valamint a már említett Egerváry Jenőn kívül elsősorban R. Rado, Ore, Ford és Fulkerson foglalkoztak alaposabban a tétellel. A tétel gyakorlati alkalmazásai közül talán a legjelentősebb H. W. Kuhn amerikai matematikus által kidolgozott eljárás, mely a matematika gazdasági alkalmazásainál felmerülő ún. hozzárendelési problémára ad egy megoldási algoritmust. Kuhn, akit módszerének kialakítására az Egerváry-féle általánosítás bizonyítása inspirált, eljárását „magyar módszernek" nevezte el, és az eljárás ezen a néven vált közismertté. Maga Kuhn írja a [34]-es dolgozat bevezető részében: „One interesting aspect of the algorithm is the fact that it is latent in work of D. Kőnig and J. Egerváry that predates the birth of linear programming by more than 15 years (hence the name, the Hungarian Method)." A témával részletesen foglalkozik még a [38] és [39]-es irodalom.

Az 1933-ban megjelent [31]-es munkában visszatér a Frobenius tételével kapcsolatos vizsgálataira, valamint Menger tételének végtelen gráfokra való kiterjesztésére. Ez utóbbi tétel bizonyítása Erdős Páltól származik.

2.3. Fő művéről

Kőnig legjelentősebb művének az 1936-ban megjelent könyvét tekinthetjük, hiszen ez a mű az első valóban tudományos színvonalú könyv, amelynek egyedüli tárgya a gráfelmélet. Összegyűjtötte a kombinatorikus gráfelmélet majdnem minden lényeges eredményét és ezzel könnyen hozzáférhetővé tett sok, addig csak elszórtan fellelhető problémát. Anyagának tervszerű elrendezésével, tárgyalásmódjának pontosságával, teljesnek mondható irodalomjegyzékével a színvonalas matematikai monográfiák között biztosított helyet könyvének. Formalizmustól mentes, könnyen érthető stílusa és a felvetett érdekes problémák sok fiatal matematikus érdeklődését keltették fel a gráfelmélet iránt. Hosszú éveken át, nagy gonddal írta könyvét, nem is lehet pontosan tudni, hogy mikor kezdett hozzá. Annyi biztos, hogy 1930-ban a könyv jelentős része már készen volt. Az íráshoz a szétszórt irodalom minden fellelhető forrását felkutatta. Utalásai, megjegyzései és főleg lábjegyzetei a gráfelmélet egész történetét tartalmazzák. A mű terjedelmének jelentős hányada más területekkel való kapcsolatokkal, alkalmazásokkal foglalkozik. A szereplő bizonyí-

tások – köztük azok is, melyek más szerzők gondolataira épülnek – a könyvben közölt alakjukban kevés kivétellel Kőnigtől származnak. A végtelen gráfok vizsgálatát könyvében tervszerűen továbbfejlesztette. Igen sok érdekes kérdést vet fel, melyek aztán további kutatások megindítói lettek. A könyv 14 fejezete röviden az alábbi témákat tárgyalja:

1. fejezet: alapfogalmak és néhány fontos, utakra és körökre vonatkozó tétel,
2. fejezet: Euler- és Hamilton vonalak tárgyalása,
3. fejezet: labirintus-probléma különböző megoldásai,
4. és 5. fejezet: körnélküli gráfok,
6. fejezet: végtelen gráfok,
7. fejezet: irányított gráfok bázisproblémái,
8. fejezet: irányított gráfok logikai, játék- és csoportelméleti alkalmazásai,
9. fejezet: irányított gráfok ciklusai és irányított csillagjaihoz tartozó lineáris formák,
10. fejezet: 9.-ben szereplő lineáris formák mod 2 redukálása,
11. fejezet: reguláris véges gráfok faktorizációs kérdései,
12. fejezet: Petersen tételének bizonyítása,
13. fejezet: végtelen reguláris gráfok faktorizációja,
14. fejezet: majdnem megegyezik a [31]-es értekezéssel.

3. Egyéb, nem gráfelméleti munkái

3.1. Halmazelméleti problémák

Az első munkája ebben a témában az 1908-ban megjelent [4] dolgozata. Ebben az $m = 2m$ (m tetszőleges végtelen számosság) egyenlőségre ad új bizonyítást. Majd 1909-ben Haar Alfréddal ír közös munkát [5], melyben a lineáris ponthalmazokra vonatkozó főbb tételeket tetszőleges, rendezett halmazokra általánosítják. Az említett tételek: Heine–Borel, Cantor–Bendixon és a Bolzano–Weierstrass tételek. Végül itt kell megemlíteni a logikai ellentmondásokról szóló [7] értekezését is, amelyben a Russel- és a Richard-féle antinómiákkal foglalkozik.

3.2. Geometriai tárgyú munkák

Elsőként az 1907-ben megírt doktori értekezéséről [3] kell szólni. Ebben az n-dimenziós (ahol n legalább 3) euklideszi tér valódi és nemvalódi forgásait (fixponttal bíró távolságtartó transzformációit) és ezeknek olyan véges forgáscsoportjait vizsgálja, melyek a minden $n \geq 3$-ra létező három szabályos test valamelyikét önmagába viszik át. Egyik eredménye: *az n-dimenziós szimplex valamely forgástengelyének dimenziója 1-gyel kisebb, mint a forgást meghatározó csúcspont-permutáció ciklusainak száma.*

1913-ban Szűcs Adolffal írt közös munkában [15] egy kocka belsejében $\underline{v}_0$ kezdősebességgel magára hagyott pontnak a kocka lapjain való rugalmas visszaverődések következtében létrejövő pályáját vizsgálják. Módszerüket Egerváry Jenő és Turán

Pál is alkalmazzák az ideális gázok egy modelljét tárgyaló művükben: A kinetikus gázelmélet bizonyos kérdéseiről, *MTA III. Osztályának Közleményei* **1** (1951).

1922-ben a [22]-es dolgozatban a Helly-féle tételre ad új bizonyítást. Ez a bizonyítás megegyezik Helly eredeti bizonyításával, mely csak 1923-ban került publikálásra. A tétel maga 1921-ben Radon egy dolgozatában került nyilvánosságra, melyben a szerző egy a Helly–Kőnig félétől eltérő bizonyítást ad.

3.3. Kombinatorikus-topológiai munkák

Az 1912-ben írt [10] és [12] dolgozataiban igazolja, hogy az n-dimenziós projektív tér irányítható, illetve nem irányítható aszerint, hogy n páratlan vagy páros.

Legjelentősebb műve ebben a témában az 1918-ban megjelent *Az analysis situs elemei* című könyve [III], melyben a kétdimenziós irányítható felületek topológiájának a kezdők számára is könnyen elsajátítható tárgyalását adja. A mű nemcsak a hazai, hanem a nemzetközi matematikai irodalomban is az első ilyen irányú mű. Sok magyar matematikus Kőnig könyvéből ismerte meg a felületek topológiáját. Szemléletre támaszkodó tárgyalásmóddal igazolja a felületek topológikus jellemzését megadó főtételt, ismerteti a felületek különböző normáltípusait. A normáltípusok származtatására az eddig ismerteknél egyszerűbb eljárást ad. Bizonyításai hézagpótlóak a nemzetközi irodalomban is.

4. Közéleti tevékenysége

Édesapjához, Kőnig Gyulához hasonlóan ő is jelentős szerepet játszott matematikai közéletünkben, melynek legjelentősebb színtere az Eötvös Loránd Matematikai és Fizikai Társulat volt. Már 1907-től tagja a Társulatnak, és 1908-tól egészen haláláig tagja a matematikai tanulóversenyek dolgozatait elbíráló bizottságnak, melynek 1942-től az elnöke is lesz.

1933-tól, Fejér Lipót utódaként a Társulat titkára lesz, ezt a tisztséget is élete végéig viseli.

Szintén 1933-tól Pogány Bélával, majd 1940-től Ortvay Rudolffal közösen szerkeszti a Társulat lapját, a Matematikai és Fizikai Lapokat. A szerkesztési munkák mellett a fenntartáshoz szükséges pénz előteremtése is az önként vállalt feladatai közé tartozott, hiszen az állami támogatás nem volt elegendő a lap megjelentetéséhez. Fő feladatának mégis a fiatal tehetségek támogatását tartotta. Minden segítséget megadott a disszertációk, cikkek publikálásához. Szintén az ifjú matematikusok támogatását szolgálta a bátyjával, Györggyel létrehozott alapítvány is. Ezen alapítványt az 1913-ban elhunyt édesapjuk emlékére hozták létre 1918-ban és a kamatokból szándékozták jutalmazni a fiatal tehetségeket. A legjelentősebb támogatást azonban a fiataloknak mégiscsak tanári működése nyújtotta. Közvetve vagy közvetlenül az ő hatására kezdett el gráfelmélettel foglalkozni sok neves matematikusunk, mint Egerváry Jenő, Egyed László, Erdős Pál, Hajós György, Krausz József, Szele Tibor, Turán Pál, Vázsonyi Endre és még sokan mások.

Utolsó munkája a Társulat 50 éves fennállása alkalmából tartott jubiláris ülésen tartott beszéde [33], melyből kiérezni a szeretetet és törődést, mellyel a Társulat működését egész életén át kísérte. Bár a megemlékezés a háború legsötétebb napjaiban készült, befejezésében Kőnig bizalommal tekint a Társulat jövője elé. Sajnos reményeinek megvalósulását nem érhette meg, nem láthatta erőfeszítéseinek leginkább kívánt eredményét, a gráfelmélet rohamos fejlődését sem, melynek mi napjainkban is tanúi vagyunk.

A leírtak alapján mondhatjuk, hogy akár emberi tulajdonságait, akár tanári tevékenységét, tudományos munkásságát, vagy közéleti tevékenységét tekintjük, élete példaként szolgálhat mindannyiunk számára.

Felhasznált irodalom

Kőnig Dénes munkái

[I] *Matematikai mulatságok I* (Budapest, 1902).

[II] *Matematikai mulatságok II* (Budapest, 1905).

[III] *Az analysis situs elemei* (Budapest, 1918).

[IV] *Matematika (műegyetemi építész- és vegyészhallgatók számára)* (Budapest, 1920).

[V] *Theorie der endlichen und unendlichen Graphen* (Leipzig, 1936).

[1] Két maximum-minimum probléma elemi tárgyalása, *Mat. és Fiz. Lapok* **8** (1899).

[2] A térképszínezésről, *Mat. és Fiz. Lapok* **14** (1905).

[3] A többméretű tér forgásainak és véges forgáscsoportjainak elemi tárgyalása, *Mat. és Fiz. Lapok* **16** (1907).

[4] Zur Theorie der Mächtigkeiten, *Rendiconti del Circ. Mat. di Palermo* **26** (1908).

[5] Egyszerűen rendezett halmazokról, *Mat. és Term.-tud. Értesítő* **27** (1909).

[6] Über einfach geordnete Mengen, *Journal f. Math.* **139** (1911).

[7] *Logikai ellentmondások* (Budapest, 1910).

[8] Vonalrendszerek kétoldalú felületeken, *Mat. és Term.-tud. Értesítő* **29** (1911).

[9] A vonalrendszerek nemszámáról, *Mat. és Term.-tud. Értesítő* **29** (1911).

[10] Többméretű alakzatok egy és kétoldalúságáról, *Mat. és Fiz. Lapok* **22** (1913).

[11] Über Ein- und Zweiseitigkeit mehrdimensionaler Mannigfaltigkeiten, *Archiv der Math. und Physik* **19** (1912).

[12] Zur Analysis Situs Doppelmannigfaltigkeiten und der projektiven Raume, *Proceedings of the 5th Congress of Math.* **II** (1913).

[13] Kőnig Gyula utolsó művéről, *Mat. és Fiz. Lapok* **23** (1914).

[14] Sur un problème de la théorie générale des ensembles et la théorie des graphes, *Revue de Métaphysique et de Moral* **30** (1923).

[15] Magára hagyott pont mozgása egy kocka belsejében, *Mat. és Term.-tud. Értesítő* **31** (1913).

[16] Mouvement d'un point abandonné a l'intérieur d'un cube, *Rendiconti del Circ. Mat. di Palermo* **36** (1913).

[17] Vonalrendszerek és determinánsok, *Mat. és Term.-tud. Értesítő* **33** (1915).

[18] Gráfok és alkalmazásuk a determinánsok és halmazok elméletében, *Mat. és Term.-tud. Értesítő* **34** (1916).

[19] Über Graphen und ihre Anwendungen auf Determinantentheorie und Mengenlehre, *Math. Annalen* **77** (1916).

[20] Kétméretű számtáblázatokról, *Mat. és Fiz. Lapok* **29** (1922).

[21] Konvex testekről, *Mat. és Term.-tud. Értesítő* **38** (1921).

[22] Über konvexe Körper, *Math. Zeitschrift* **14** (1922).

[23] Sur les rapports topologiques d'un problème d'analyse combinatorique, *Acta Litt. Ac. Sci. Szeged* **2** (1924).

[24] Halmazok többértelmű leképezéséről, *Mat. és Term.-tud. Értesítő* **42** (1926).

[25] Über mehrdeutige Abbildungen von Mengen, *Math. Annalen* **95** (1926).

[26] A graphokról, *Középiskolai Mat. és Fiz. Lapok* **2** (1926).

[27] Sur les correspondences multivoques des ensembles, *Fund. Math.* **8** (1926).

[28] Über eine Schlussweise aus dem Endlichen ins Unendliche, *Acta Litt. Ac. Sci. Szeged* **3** (1927).

[29] Graphok és mátrixok, *Mat. és Fiz. Lapok* **38** (1931).

[30] Egy végességi tétel és alkalmazásai, *Mat. és Fiz. Lapok* **39** (1932).

[31] Über trennende Knotenpunkte in Graphen, *Acta Litt. Ac. Sci. Szeged* **6** (1933).

[32] Kürschák József, *Középiskolai Mat. és Fiz. Lapok* **9** (1933).

[33] Az Eötvös Loránd Matematikai és Fizikai Társulat első ötven éve, *Mat. és Fiz. Lapok* **48** (1941).

Egyéb

[34] H. W. Kuhn, The Hungarian method for the assignment problem, *Naval Res. Quarterly* **2** (1955) 83–97.

[35] Egerváry J., Mátrixok kombinatorius tulajdonságairól, *Matematikai és Fizikai Lapok* **38** (1931) 16–28.

[36] Gallai T., Kőnig Dénes, *Matematikai és Fizikai Lapok* **XV/4** (1965) 277–293.

[37] Petersen, Die Theorie der regulären graphen, *Acta Math.* **15** (1891) 193–220.

[38] H. W. Kuhn, A magyar módszer eredetéről (ford. Komlósi Sándor), *Szigma* **23** (1992) 113–118.

[39] Frank A., A magyar módszer és általánosításai, *Szigma* **XXXIII** (2002) 1–2, 13–44.

[40] Rapcsák T., Egerváry Jenő élete és munkássága, *Szigma* **XXXIII** (2002) 1–12.

LIBOR JÓZSEFNÉ
SZOLNOKI FŐISKOLA
GAZDASÁGELEMZÉSI, MÓDSZERTANI TANSZÉK
liborne@szolf.hu

IN COMMEMORATION OF DÉNES KŐNIG

JÓZSEFNÉ LIBOR

My presentation is about one of the famous Hungarian mathematicians, Dénes Kőnig. I've chosen him because he was the creator of graph-theory and his most famous theory is used in Operations Research nowadays too.

To most graph theorists there are two outstanding landmarks in the history of their subject. One is Euler's solution of the Königsberg Bridges Problem, dated 1736, and the other is the appearance of Dénes Kőnig's textbook in 1936.

But the honour of presenting Graph Theory to the mathematical world as a subject in its own right, with its own textbook, belongs to Dénes Kőnig.

In 2004 there were the anniversary of his 120th birthday and the 60th anniversary of his tragic death.

First of all I'd like to write about his biography and than about his scientific works and his activities in mathematical community.

His father, Gyula Kőnig, was an eminent professor of mathematics at the Technological University of Budapest. In 1899, while he was a high-school student, his first paper was published in Matematikai és Fizikai Lapok (Mathematical and Physical Journal of secondary schools). In this he gives an elementary discussion of two extreme-value problems.

He attended the first four semesters of his university classes at the University of Budapest, and the last five at the University of Göttingen.

After finishing his studies he obtained in 1907 the degree of doctor of philosophy, with a dissertation on a geometrical topic. In the same year he worked as Demonstrator at the Technical University of Budapest. From that time until his death in 1944 he remained attached to the Technical University.

Dénes Kőnig's most important contributions are in graph theory. His name is attached to a number of fundamental theorems. No less important than his results is his success in making graph theory widely known and appreciated.

His best-known results are in combinatorial (absolute) graph theory. Kőnig never failed to investigate whether the result under consideration might be extended to infinite graphs. He considered this sufficiently important to stress it in the title of his book: Theory of finite and infinite graphs.

Using Kőnig's theorem, the American mathematician H. W. Kuhn constructed a solution algorithm to the so-called assignment problem in mathematical economics. Kuhn, whose inspiration came from Egerváry's generalization, called the process the 'Hungarian method' and it has become known by that name. Kuhn said in the introduction of his paper: *The Hungarian Method for the assignment problem*: "One interesting aspect of the algorithm is the fact that it is latent in work of D. Kőnig and J. Egerváry that predates the birth of linear programming by more than 15 years (hence the name, the Hungarian Method)."

In 1907 he was already a member of the Loránd Eötvös Mathematical and Physical Society, in 1933 he was elected secretary of the Society as successor of Lipót Fejér, and he held this office until the end of his life.

In 1944 he expended great effort to help persecuted mathematicians, but 15th October – when the Hungarian National Socialist Party took the power over –, prevented him from fulfilling his plans. This sad turn of events had also become the cause of his tragedy. On 19th October 1944, he sought refuge from persecution in death.

Alkalmazott Matematikai Lapok **22** (2005), 203–204.

KÖNYVISMERTETÉS

Tóth János, Simon Péter: Differenciálegyenletek. Bevezetés az elméletbe és az alkalmazásokba (393 oldal; 4100 Frt) Typotex, Budapest, 2005

GARAY BARNA

A szerzőpáros sikerrel oldotta meg a közös tankönyvük alcímében felvállalt feladatot. Megismertetik az olvasót a differenciálegyenletek elméletének elemeivel, bemutatják a legfontosabb alkalmazásokat, s eközben kijelölik a továbbhaladás útját, mind az igényesebb elmélet, mind a bonyolultabb alkalmazások felé.

A rövid „1. Bevezetés" és az 50 oldalnyi „11. A feladatok megoldása" részeket leszámítva valamennyi fejezet az aktuális célkitűzések tézisszerű megfogalmazásával kezdődik, amelyet az adott témakör részletes tárgyalása követ. Az absztrakt eredményeket számos előkészítő és menet közbeni példa motiválja, majd mechanikai, fizikai, biológiai és kémiai alkalmazások jönnek, ahol is a hangsúly – a szerzőtársak saját kutatási tapasztalatának megfelelően – ez utóbbiakra, azon belül is a reakciókinetikai alkalmazásokra esik. A tényleges alkalmazások elképzelhetetlenek számítógép használata nélkül. Ennek megfelelően az egyes fejezeteket (a *Mathematica* programcsomag segítségével készített) felhasználói szintű, bemutató-demonstrációs programok zárják.

A könyv közönséges differenciálegyenletekből az alábbi témaköröket veszi sorra: kezdetiérték-feladatok, lineáris rendszerek, egyensúlyi helyzetek és periodikus megoldások stabilitása ($n \geq 1$ dimenzióban); lineáris peremérték-feladatok, általános kvalitatív elmélet ($n = 1$ és $n = 2$ dimenzióban). A szigorúan vett közönséges differenciálegyenletek rész mintegy 250 oldal terjedelmű, ezután 30-30 oldalon a parciális differenciálegyenleteknek és a variációszámításnak azok az alaptípusai-alapegyenletei következnek, amelyek a közönséges differenciálegyenletek imént bemutatott fejezetei alapján jól kezelhetők. A könyv olvasásához előzetes funkcionálanalízis tanulmányokra nincsen szükség, az elemi analízis és a lineáris algebra alapjainak ismerete teljesen elegendő. Mindazonáltal – utalásaiban, megjegyzéseiben, és ami különösen fontos, természetes általánosításokként felmerülő példáiban $\left(\left(\dot{x}(t)\right)^4 = x \circ x(t)\right.$ polinom alakú megoldásai, $\ddot{x} + \lambda x = 0$, $x(0) = x(1) = 0$ mint sajátérték-feladat, $\dot{x}(t) = \int_{-\infty}^{t} K(s,t)x(s)\,ds$ mint a teljes múlttól való lineáris füg-

gés etc.) – a könyv nyitott a matematika számos további fejezete felé; a szűkebb szakmán belül az olvasót a bifurkációelmélet és a centrális sokaság küszöbéig kíséri.

A szerzők szerencsés kézzel teremtenek egyensúlyt a matematikán belüli és az alkalmazások gyakorlati megfontolásaiból származó intuíció között – a kezdeti feltételektől való folytonos függés tételét például a „Mérési és modellhibák hatása a megoldásokra" alfejezetben tárgyalják. A kitűzött feladatok egyrészt a típuspéldák begyakorlását, másrészt az elmélet alapvető összefüggéseinek felidézését, önálló újragondolását szolgálják. Szükség esetén az olvasó előrelapozhat a megoldásokért, de az a cél, hogy ezt minél kevesebbszer kelljen tennie. Az „együttlakva ismerszik meg" bölcsessége erre is vonatkozik: az olvasó akkor ismeri meg igazán a differenciálegyenleteket, akkor érti meg őket, ha bánni is tud velük – papíron, ceruzával, számítógéppel, matematikus és nem-matematikus végzettséggel, a saját szakmájában. Simon Péter és Tóth János egyetemi előadásokból kinőtt, a magyar nyelvű szakirodalomban hiánypótló munkája ehhez kínál segítséget.

Tóth János tervezi, hogy a könyvvel kapcsolatban folyamatos kiegészítéseket jelentet meg saját `http://www.math.bme.hu/~jtoth` honlapján.

A kiadásért felelős a BJMT főtitkára
Szedte és tördelte az Egyenes Bt.

Nyomta az MSZH Nyomda és Kiadó Kft., Budapest
Felelős vezető: Nagy László

Budapest, 2006
Megjelent 18 (A/5) ív terjedelemben
250 példányban
HU ISSN 0133-3399

ÚTMUTATÁS A SZERZŐKNEK

Az Alkalmazott Matematikai Lapok csak magyar nyelvű dolgozatokat közöl. A kéziratok gépelését olyan formában kérjük, hogy minden gépelt oldal 25, egyenként átlag 50 betűhelyes sort tartalmazzon. A közlésre szánt dolgozatokat e-mailen az aml@math.elte.hu címre kérjük elküldeni az ábrákat tartalmazó fájlokkal együtt. Előnyben részesülnek a TEX-ben elkészített dolgozatok.

A kéziratok szerkezeti felépítésének a következő követelményeket kell kielégíteni. A fejlécnek tartalmaznia kell a dolgozat címét, a szerző teljes nevét, valamint annak a városnak a nevét, ahol a szerző dolgozik. A fejléc után egy, képletet nem tartalmazó, legfeljebb 200 szóból álló kivonatot kell minden esetben megadni. A dolgozatot címmel ellátott szakaszokra kell bontani, és az egyes szakaszokat arab sorszámozással kell ellátni. Az esetleges bevezetésnek mindig az első szakaszt kell alkotnia. Az irodalomjegyzék után, a kézirat befejezéseképpen fel kell tüntetni a szerző teljes nevét és a munkahelye (illetve lakása) pontos címét. A dolgozatban előforduló képleteket szakaszonként újrakezdődően, a képlet előtt két zárójel közé írt kettős számozással kell azonosítani. Természetesen nem szükséges minden képletet számozással ellátni. Az esetleges definíciókat és tételeket (segédtételeket és lemmákat) ugyancsak szakaszonként újrakezdődő, kettős számozással kell ellátni. Kérjük a szerzőket, hogy ezeket, valamint a tételek bizonyítását a szövegben kellő módon emeljék ki. Minden dolgozathoz csatolni kell egy angol, német francia vagy orosz nyelvű, külön oldalra gépelt összefoglalót.

Mind az ábrákat, mind a lábjegyzeteket a dolgozat szakaszokra bontásától független, folytatólagos arab sorszámozással kel ellátni. Az ábrák elhelyezését a dolgozat megfelelő helyén, széljegyzetként feltüntetett, ábraazonosító sorszámokkal kell megadni. A lábjegyzetekre a dolgozaton belül az azonosító sorszám felső indexkénti használatával lehet hivatkozni.

Az irodalmi hivatkozások formája a következő. Minden hivatkozást fel kell sorolni a dolgozat végén található irodalomjegyzékben, a szerzők, illetve a társszerzők esetén az első szerző neve szerint alfabetikus sorrendben úgy, hogy a cirill betűs szerzők nevét a Mathematical Reviews átírási szabályai szerint latin betűsre kell átírni. A folyóiratban megjelent cikkekre [1], a könyvekre [5], a kötetben megjelent dolgozatokra [4], a disszertációkra [3] és a gépi program leírásokra [2] a következő minta szerint kell hivatkozni:

[1] Farkas, J., Über die Theorie der einfachen Ungleichungen, *Journal für die reine und angewandte. Mathematik* **124** (1902) 1–27.

[2] Kéri, G., „DUALSIMP", rutin a CDC 3300-ás gépekre (Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete, CDC 3300 felhasználói ismertetők 2. 1973. május) 19–20.

[3] Prékopa, A. „Sztochasztikus rendszerek optimalizálási problémáiról", doktori értekezés. Magyar Tudományos Akadémia, Budapest, 1970.

[4] Prabhu, N. U. „Recent research on the ruin problem of collective risk theory", in: *Inventory Control and Water Storage* Ed. A. Prékopa (János Bolyai Mathematical Society and North-Holland Publishing Company, Amsterdam–London, (1973) 221–228.

[5] Zoutendijk, G. *Methods of Feasible Directions* (Elsevier Publishing Company, Amsterdam and New York, 1960).

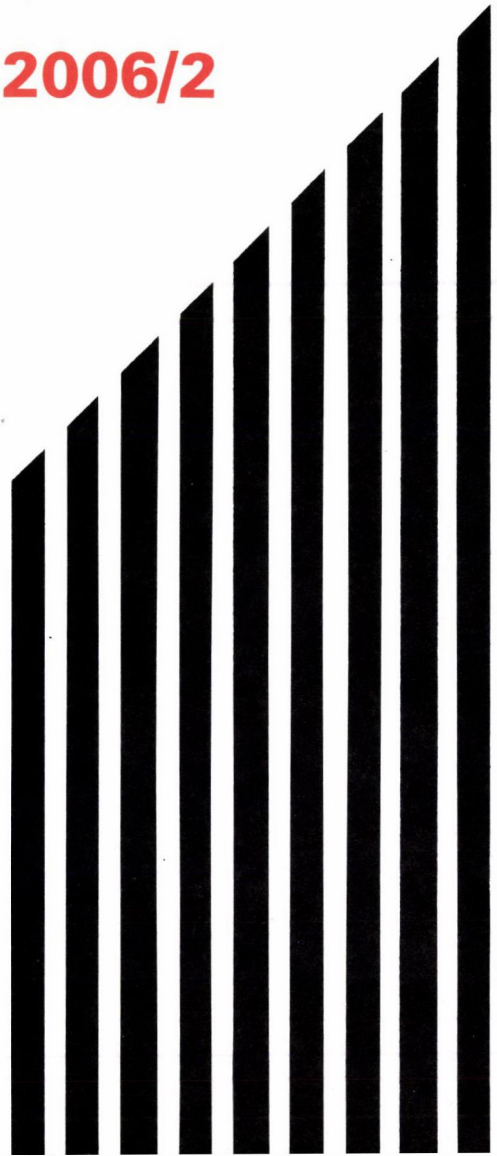
A dolgozatok szövegében az irodalmi hivatkozás számait szögletes zárójelben kell megadni, mint például [5] vagy [4, 76–78]. A szerzők a dolgozatukról 50 darab ingyenes különlenyomatot kapnak. A dolgozatok után szerzői díjat az Alkalmazott Matematikai Lapok nem fizet.

TARTALOMJEGYZÉK

INDEX

Alkalmazott matematikai lapok

2006/2



A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI TUDOMÁNYOK
OSZTÁLYÁNAK KÖZLEMÉNYEI

23.

KÖTET

ALKALMAZOTT MATEMATIKAI LAPOK

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

ALAPÍTOTTÁK
KALMÁR LÁSZLÓ, TANDORI KÁROLY, PRÉKOPA ANDRÁS, ARATÓ MÁTYÁS

FŐSZERKESZTŐ
PÁLES ZSOLT

FŐSZERKESZTŐ-HELYETTESEK
BENCZÚR ANDRÁS, SZÁNTAI TAMÁS

FELELŐS SZERKESZTŐ
VIZVÁRI BÉLA

TECHNIKAI SZERKESZTŐ
KOVÁCS GERGELY

23. kötet

Szerkesztőség és kiadóhivatal: 1027 Budapest, Fő u. 68.

Az Alkalmazott Matematikai Lapok változó terjedelmű füzetekben jelenik meg, és olyan eredeti tudományos cikkeket publikál, amelyek a gyakorlatban, vagy más tudományokban közvetlenül felhasználható új matematikai eredményt tartalmaznak, illetve már ismert, de színvonalas matematikai apparátus újszerű és jelentős alkalmazását mutatják be. A folyóirat közöl cikk formájában megírt, új tudományos eredménynek számító programokat, és olyan, külföldi folyóiratban már publikált dolgozatokat, amelyek magyar nyelven történő megjelentetése elősegítheti az elért eredmények minél előbbi, széles körű hazai felhasználását. A szerkesztőbizottság bizonyos időnként lehetővé kívánja tenni, hogy a legjobb cikkek nemzetközi folyóiratok különszámaként angol nyelven is megjelenhessenek.

A folyóirat feladata a Magyar Tudományos Akadémia III. (Matematikai) Osztályának munkájára vonatkozó közlemények, könyvismertetések stb. publikálása is.

A kéziratok a főszerkesztőhöz, vagy a szerkesztőbizottság bármely tagjához beküldhetők. A főszerkesztő címe:

Páles Zsolt, főszerkesztő
1027 Budapest, Fő u. 68.
A folyóirat e-mail címe: aml@math.elte.hu

Közlésre el nem fogadott kéziratokat a szerkesztőség lehetőleg visszajuttat a szerzőhöz, de a beküldött kéziratok megőrzéséért vagy továbbításáért felelősséget nem vállal.

Az Alkalmazott Matematikai Lapok előfizetési ára kötetenként 850 forint. Megrendelések a szerkesztőség címén lehetségesek.

A Magyar Tudományos Akadémia III. (Matematikai) Osztálya a következő idegen nyelvű folyóiratokat adja ki:

1. Acta Mathematica Hungarica,
2. Studia Scientiarum Mathematicarum Hungarica.

A Kriptográfia különszámot Nemetz Tibor vendégszerkesztő hozta létre, aki sajnálatos módon a szám előkészítő munkálatai alatt elhunyt. Egy későbbi számunkban emlékezünk meg Róla részletesen.

A szerkesztők

Alkalmazott Matematikai Lapok **23** (2006), 207–294.

A KRIPTOGRÁFIAI BIZTONSÁG MEGKÖZELÍTÉSI MÓDJAI – ALKALMAZÁSOK, KRIPTOGRÁFIAI STRUKTÚRÁK

PAPP PÁL, SZABÓ ISTVÁN

1. Bevezetés

Jelen Tanulmány elsősorban a kriptográfiai biztonság elérésének módjait elemzi, különös tekintettel a nyilvános kulcsú rendszerek (PKI) kriptográfiai biztonságára. Ehhez áttekintésre kerülnek:

- a biztonságot veszélyeztető tényezők, a szakirodalomból ismert támadási módszerek osztályozása;
- a kriptográfiai biztonság különböző megközelítései:
  - információelméleti megközelítések, bizonyított biztonság,
  - redukciós-bonyolultságelméleti biztonság,
  - kalkulációs biztonság,
  - a kvantumkriptográfiai eredmények hatása a kriptográfiai biztonság klasszikus eredményeire,
- a kriptográfiai rendszerek struktúrái, részosztályai:
  - kriptográfiai primitívek:
    - kulcsnélküli primitívek,
    - titkos kulcsú primitívek (folyamrejtjelzők, blokkrejtjelzők),
    - nyilvános kulcsú primitívek,
  - kriptográfiai sémák,
  - kriptográfiai protokollok,
  - kriptográfiai alkalmazások,
- kriptográfiai szabványok.

A téma rendkívül sokrétű, és szinte áttekinthetetlenül hatalmas szakirodalma van. Számtalan könyv, interneten elérhető publikáció, az új eredményeket évente áttekintő – hatalmas kutatói létszámmal megrendezett – konferencia (pl. EUROCRYPT, CRYPTO, ASISACRYPT, FSE /Fast Software Encryption/, SHARCS /Special Purpose Hardware for Attacking Cryptographic Systems/, stb. kiadványai mutatják a kriptográfia dinamikus fejlődését. Jelen áttekintés elsősorban a kriptográfiai alkalmazás rendszerszemléletét, a biztonság különböző megközelítéseit emeli ki (néhány, a konferenciákon elhangzó, nehezen hozzáférhető eredményre figyelemfelhívással). Akit a téma részletesebben érdekel, a hatalmas irodalomból külön is ajánljuk a magyar nyelven is elérhető, a klasszikus eredményeket (DES, LFSR, KnapSack, ...) áttekintő [6] könyvet, a modern biztonsági algoritmustervezés követelményeit részletező [2] könyvet, az átfogó ismeretterjesztést megvalósító [10] könyvet, illetve [5] interneten szabadon hozzáférhető könyvét: Handbook of applied cryptography, `http://www.cacr.math.uwaterloo.ca/hac/`.

Kevés olyan alkalmazotti szakterület van, ahol annyira szerteágazó matematikai részterületek eredményei kerülnek komplex alkalmazásra, mint a kriptográfia területén. Néhány matematikai terület, melynek eredményeire épít a kriptográfia: algebra (csoportelmélet, Galois-testek ...), számelmélet (prímszámelmélet, faktorizáció, kongruenciák ...), bonyolultságelmélet (NP elmélet, NPC osztályok, redukció bizonyítottan nehéz problémákra ...), valószínűségszámítás, matematikai statisztika, információelmélet (entrópia, Shannon-féle tökéletes rejtjelrendszer, véletlenszám-generálások ellenőrzése ...), számítógéptudomány (algoritmusok gyorsítása, párhuzamosítása), és így tovább.

Ugyanakkor a hatás nem egyirányú, a felvetődő gyakorlati problémák az elméleti kutatásokra is jelentős visszahatást eredményeztek, melyek alapján jelentős új matematikai eredmények születtek. Gondoljunk itt az információelméletre gyakorolt hatáson túl pl. az RSA algoritmus által indukált fejlődésre a faktorizációs módszerekben, a stream-cipherek területén bevezetett lineáris ekvivalens fogalma alapján kidolgozott új statisztikai próbákra, a kriptográfiai célú véletlenszám generálás követelményei alapján kifejlesztett statisztikai programrendszerekre,[1] továbbá az NP elmélet kriptográfiai indíttatású továbbfejlesztéseire (pl. az egyirányú függvény, keménybit, véletlentől megkülönböztethetőség NP technikát felhasználó, továbbfejlesztő bizonyításaira, melyekről részletesen olvashatnunk a [2] könyvben) vagy a kriptográfiai szakirodalomban az 1990-es évektől új kutatási irányként megjelenő (NP elméleti módszerek felhasználásával) bizonyított biztonságú primitívek gyakorlati számításigény-becslési megközelítésére, és a sor még hosszan folytatható.

A titkosírás több mint kétezer éves története során megszámlálhatatlan mennyiségű algoritmus (eljárás) került kidolgozásra és (rengeteg) feltörésre (ld. például David Kahn híres könyvét: The Coedebreakers, New York, (1996)). Ez a folyamat a legutóbbi néhány évtizedben felgyorsult (ld. pl. a hatalmas mennyiségű újkeletű

[1] (*ld. pl. NIST Special Publication 800-2: A Statistical Test Suite for Random and Pseudorandom Number Generators, May 2001;* Maurer Ueli M.: „A universal statistical test for random number generators", Journal of Cryptology, vol.5, no. 2., 1992, pp. 89–105.);

algoritmusok egy részének összefoglalását Bruce Schneier: Applied Cryptography: Protokols, Algorithms, and Source Code in C, John Wiley & Sons, New York, 2nd edition, (1996) könyvében).

Ahogy az információ-technológiai rendszerek fejlődésében is nemzetközi szinten az egységesítésre, a szabványosításra való törekvések térnyerése figyelhető meg, ugyanúgy az információ védelme, és ennek fontos részterülete, a kriptográfiai eljárások területén is – a korábbi egyedi törekvésekkel szemben – egységes követelmények, szabványos védelmi megoldások (és szabványok) jelentek meg.

A kriptográfia nem öncélú eszköz, hanem az informatikai (kommunikációs) rendszerek biztonságához tud nélkülözhetetlen eszközeivel hozzájárulni, mind a bizalmasság (Confidentiality, Secrecy) megőrzése, az integritás (Integrity) biztosítása (pl. MAC), mind a hitelesség (Authenticity) biztosítása (pl. elektronikus aláírás, hitelesítési protokollok) területén – de csak kellő erősségű, bizonyításokkal alátámasztott eljárások, rendszerek alkalmazása esetén.

2. A kriptográfia helye az információvédelemben

Az informatikai biztonság témaköre rendkívül szerteágazó terület, melyből a kriptográfia alkalmazása több kiemelt területen elősegíti az informatikai biztonság növelését.

Az informatikai biztonság két nagy ága:

- A *megbízható működés* biztosítása, mely terület többek között magában foglalja az alábbi veszélyek kivédését:
  - működési (SW, HW) hibák okozta károk;
  - a rendszer fizikai sérülései;
  - szolgáltatások megbénítása, megbénulása (pl. DOS, DDOS: Distributed Denial of Services), támadások;
  - számítási kapacitás lopása; stb.
- *Információvédelem*, melynek leggyakrabban említett részterületei:
  - bizalmasság (Confidentiality, Secrecy) megőrzése;
  - integritás (Integrity) biztosítása: a rendszerek (programok) és adatok integritásának, konzisztenciájának fenntartása;
  - hitelesség (Authenticity), beleértve az eredet (küldő), és a tartalom hitelességének /pl. elektronikus aláírással történő/ igazolását.
- Kiegészítő informatikai biztonsági *elvárás:*
  - Tranzakció utólagos *letagadhatatlanság*ának garantálása (Non-repudiation): megakadályozni a felhasználókat abban, hogy utólag letagadjanak valamilyen általuk elvégzett tevékenységet.

A fenti biztonsági elvárások teljesítéséhez a kriptográfiai eljárások több területen nyújthatnak magas színvonalú, (környezeti feltételek biztosítása mellett) bizonyítható biztonságú módszereket, például a bizalmasság megőrzése területén, de ezen kívül a hitelesség, integritás megőrzése, a rendszerekhez való illetéktelen hozzáférés kizárása (ld. jelszóképek egyirányú függvények segítségével való tárolása, egyszeri jelszóképzések, stb.) területein is.

A kriptográfia alkalmazásával kapcsolatos szabályozás nemzetközi szinten nem egységes. A legtöbb országban, ahol a kriptográfiával törvényi szinten is foglalkoznak, a kriptográfiai jogalkotás többnyire csak a titkosításra használt kriptográfiára vonatkozik (a sértetlenség, hitelesség, letagadhatatlanság céljából használt kriptográfiára nem). Magyarországon a titkosítással kapcsolatos jogszabályok hatóköre csak az ún. minősített információk (állam- és szolgálati titok) védelmére vonatkozó hatósági felügyeletet szabályozzák. A hitelesség kérdéseivel – az elektronikus aláírások szabályozásaival – külön jogszabályok foglalkoznak.

Korábban a kriptográfiai eszközök exportját a Wassenaar Egyezmény (Wassenaar Arrangement – WA) értelmében 32 ország együttesen korlátozta. Néhány évtizede a COCOM (Coordinating Committee for Multilateral Export Controls) egy nemzetközi szervezet volt, amely a tagországok stratégiai termékeinek és technikai adatainak közös szabályozását, védelmét biztosította meghatározott célországok vonatkozásában.

A Wassenaar Megállapodást 1998 decemberében felülvizsgálták, könnyítéseket vezettek be. Ennek alapján a következő termékek szabadon exportálhatók lettek: az összes szimmetrikus kriptográfiai termék 56 bit-ig, az összes aszimmetrikus kriptográfiai termék 512 bit-ig és az összes alcsoporton alapuló kriptográfiai termék (beleértve az elliptikus görbét) 112 bit-ig.

A 2000. november 30-án megtartott ülés alkalmával a Wassenaar egyezményhez csatlakozott államok *feloldották* az 56 bit-es export szabályozási korlátot a tömegpiaci kriptográfiai szoftverek és hardverek vonatkozásában.

Ma elfogadott – sőt követelmény – a kriptográfiai megoldások alkalmazása, szabványosítása – így nyilvánossága – az információvédelem több területén: a bizalmasság és a hitelesség biztosítása, a sértetlenség segítése, az utólagos letagadhatatlanság garantálása területén.

A kriptográfia (alapvetően elméleti matematikára támaszkodó) eredményei döntő hozzájárulást adhatnak az informatikai rendszerek biztonságának garantálásához. Ugyanakkor az informatikai rendszerek biztonsága (benne a kriptográfiai eljárások biztonsága is) több más szakterülethez kapcsolódik. Ilyen, a matematikai diszciplínától eltérő területek:

- *jogtudomány* (ld. jogszabályi követelmények az adatvédelemre, konkrét matematikai eljárások megjelenése jogszabályokban – pl. az elektronikus aláírási törvény, illetve végrehajtási rendeletei, ahol meghatározták a jogilag releváns aláíró algoritmusokat, ajánlásokat fogalmaztak meg paraméterválasztásra);
- *minőségbiztosítási, szervezetirányítási szakterületek:* pl. termékek és folyamatok minőségbiztosítása, informatikai rendszereket üzemeltető szervezetek mű-

ködtetése, szabályozása, emberi tényezők kezelése: a legbiztonságosabb rendszerek is veszélybe kerülhetnek emberi mulasztásokból, ezt használja ki az ún. „social engineering" féle támadás: ld. pl. Kevin Mitnick: „A legendás hacker" című könyvét.

A kriptográfiai rendszerek alapját a kriptográfiai algoritmusok, alapelemek biztosítják, melyek egzakt leírását a szakirodalomban ***kriptográfiai primitívek***nek nevezik. Erre épülnek a ***kriptográfiai sémák***, azokra a ***kriptográfiai protokollok***. Ezeket használják fel a ***kriptográfiai alkalmazások***ban, melyeket komplex informatikai rendszerekben működtetnek. A biztonsági elvárások teljesüléséhez minden szinten biztosítani kell a követelményeket teljesítő biztonságos eljárásokat, valamint ezek konzisztenciáját (későbbiekben példát látunk arra, hogy erős titkosítás, jó kriptográfiai protokoll mellett is gyenge konstrukciót valósíthat meg, ld. 10.1. fejezet).

3. A szteganográfia fogalmai, osztályozásai, felhasználási területei

A bizalmasság biztosítása a kommunikáció során két – módszereiben elkülönülő – módon (illetve ezek együttes alkalmazásával) is biztosítható /kellő erősségű eljárásokkal/:

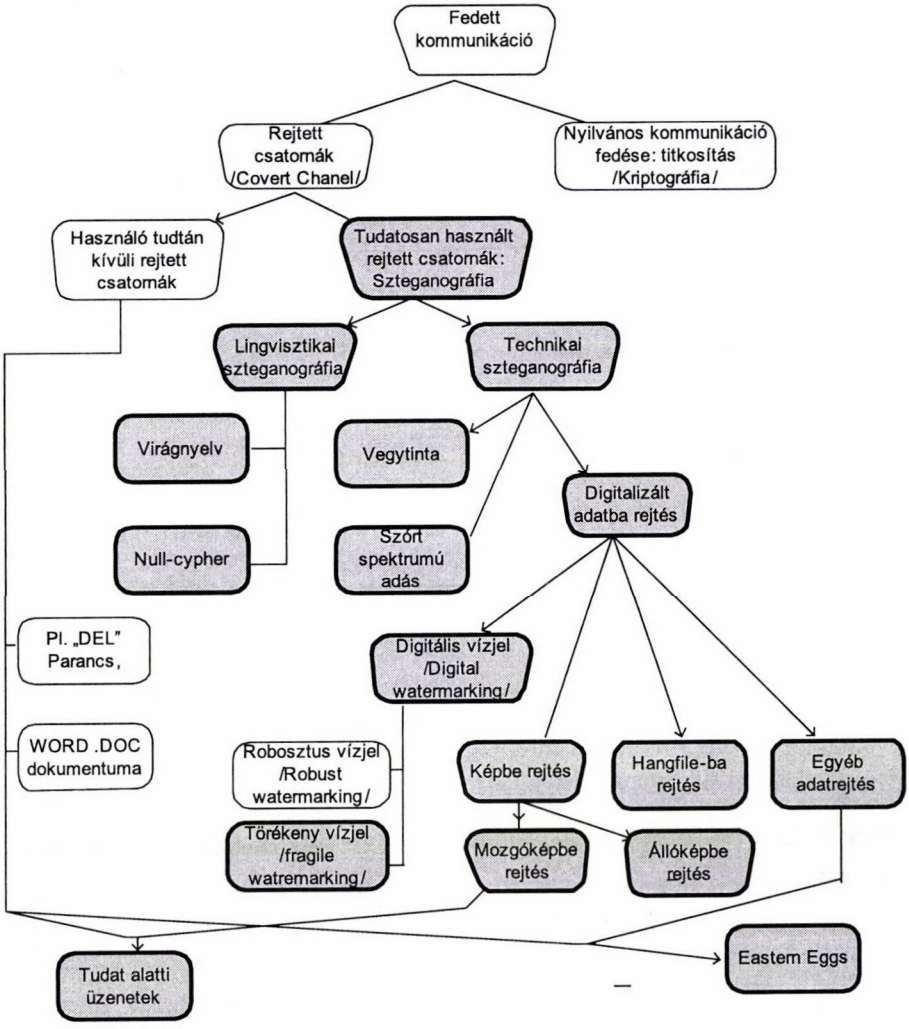
- a kommunikáció tényének /létének/ elfedésével,
- a kommunikáció tartalmának elfedésével.

Az első eljárást a szakirodalom a szteganográfiai módszerekhez, míg a másodikat kriptográfiai módszerekhez sorolja.

Jelen tanulmány tárgya elsősorban a kriptográfia, de a bizalmasság biztosításában a görögöktől napjainkig (sőt ma ismét egyre növekvő arányban) jelentős szerepe volt (van és lesz) a szteganográfiának. A szteganográfia rejtett írást jelent. A kommunikáció védelme tárgyában olyan kommunikációt értenek alatta, amikor a kommunikáció (információ közlés) ténye is rejtve marad egy illetéktelen figyelő előtt. Sok esetben ez is a biztonságot növelő fontos szempont. (Szokás még a „DATA HIDING", katonai szakirodalomban a „TRANSEC" /transmission security/ kifejezések analóg használata.)

A szteganográfia két nagy ága: a lingvisztikai szteganográfia és a technikai szteganográfia (mely utóbbiak közé tartozik például az ún. „szórt spektrumú" adás, amikor a kommunikáció – illetékteleneknek – „fehérzajként" fogható, de adó-vevő oldal által ismert eljárással az üzenet kivehető a zajból, vagy amikor digitalizált adatfolyamba – pl. képekbe – illesztik a védendő üzeneteket).

Az 1. ábra összefoglalja a szteganográfia néhány nagyobb csoportját (terjedelmi okokból a teljesség igénye nélkül):


Fedett kommunikáció
Rejtett csatornák /Covert Chanel/
Nyilvános kommunikáció fedése: titkosítás /Kriptográfia/
Használó tudtán kívüli rejtett csatornák
Tudatosan használt rejtett csatornák: Szteganográfia
Lingvisztikai szteganográfia
Technikai szteganográfia
Virágnyelv
Vegytinta
Digitalizált adatba rejtés
Null-cypher
Szórt spektrumú adás
Pl. „DEL" Parancs,
Digitális vízjel /Digital watermarking/
WORD .DOC dokumentuma
Robosztus vízjel /Robust watermarking/
Képbe rejtés
Hangfile-ba rejtés
Egyéb adatrejtés
Törékeny vízjel /fragile watremarking/
Mozgóképbe rejtés
Állóképbe rejtés
Tudat alatti üzenetek
Eastern Eggs

1. ábra

4. A kriptográfia fogalmai, osztályozásai, felhasználási területei

A **kriptológia** két nagy ága: a **kriptográfia**, mely alatt elsősorban a rejtjelrendszerek tervezését és használatát értik; míg **kriptoanalízis** alatt egyrészt a rejtjelrendszerek erősségének vizsgálatát, másrészt az illetéktelen támadó módszereinek tárházát, amely a rejtjeles üzenetek feltörésével foglalkozik, vagyis azzal, hogy hogyan lehet visszanyerni a nyílt szöveget a megfelelő kulcs ismerete nélkül. (A következőkben a szakirodalomban is összevontan használt fogalomként kriptográfiáról beszélünk.)

A kriptográfiában az alábbi fontosabb elnevezések használatosak:

- a védendő üzenet: nyílt szöveg /**Plaintext**/;
- az üzenet tartalmának az olyan kódolása, transzformálása, amely elrejti annak tartalmát a kívülállók elől: rejtjelzés /**Encryption**/;
- a transzformálás eljárása: rejtjelző algoritmus;
- míg ennek eredménye, a transzformált üzenet: rejtjeles szöveg /**Ciphertext**/;
- az a folyamat, amelynek során a nyílt szöveget visszanyerjük a rejtjelesből: megoldás /**Decryption**/;
- a rejtjelzés során általában egy kulcsot /**Key**/ használnak úgy, hogy a megoldást a címzett a kulcs ismeretében elvégezheti.

Az algoritmusok általában nyilvánosak, sőt a legújabb algoritmusok esetén (pl. AES, NESSI projektek) az algoritmusok tervezési, tesztelési szempontjai is nyilvánosak. Elsődleges cél – ez csak kellő erősségű algoritmus és a rendszerkörnyezetre vonatkozó feltételek biztosítása esetén teljesül –, hogy a kulcs ismeretének hiányában illetéktelen fél a nyílt szöveget ne tudja visszaállítani.

A kriptográfia biztonsági mechanizmusokat biztosít a biztonságos üzenetküldési, hitelesítési, a digitális aláírási és egyéb információvédelmi problémák megoldásához is.

A kulcsokat felhasználó kriptográfiai algoritmusok két nagyobb csoportba sorolhatók:

- a szimmetrikus (titkos) kulcsú és
- az aszimmetrikus (nyilvános) kulcsú algoritmusok.

A **szimmetrikus kulcsú algoritmusok** ugyanazt a kulcsot használják a rejtjelzés és a megoldás során (vagy a megoldó /dekódoló/ kulcs könnyen származtatható a rejtjel kulcsból).

Az **aszimmetrikus kulcsú** (vagy nyilvános kulcsú: PKI) algoritmusoknál kicsit félreérthető a „nyilvános" kulcsú megjelölés, mivel külön kulcsot használnak a rejtjelzéshez és a megoldáshoz, és csak a rejtjelző kulcs nyilvános, míg a megoldó kulcs számítása a rejtjelző kulcsból – jó rendszerek esetén – nem kivitelezhető.

Általában a szimmetrikus kulcsú algoritmusok sokkal gyorsabbak, mint az aszimmetrikus kulcsú algoritmusok, ezért a gyakorlatban ezeket gyakran együttesen használják úgy, hogy egy nyilvános kulcsú algoritmussal egyeztetnek (viszonylag rövid) titkos rejtjelkulcsot a hosszabb üzenet – szimmetrikus kulcsú, gyors – rejtjelzéséhez.

5. A kriptográfiai rendszerek struktúrái, részosztályai

A kriptográfiai rendszerek sokféle célból (különböző elvárások teljesítésére) számos algoritmust, algoritmusok együttesét alkalmazzák, melyek egy részének biztonsága (és tervezési elve) közös matematikai problémákra vezethető vissza.

A tervezés során biztonsági szempontból megkülönböztethetők

- intuitív módszerek (legyen minél „bonyolultabb");
- jól struktúrált, klasszikus matematikai problémákra visszavezethető algoritmusok.

Az első típusra példák a DES (ahol utólag dolgoztak ki tervezési elveket, pl. a véletlen permutációk, S-dobozok tervezésére), vagy AES blokkrejtjelzők.

A második típus inkább a nyilvános kulcsú rendszerekre jellemző, a felhasznált klasszikus matematikai problémák, melyekre próbálják visszavezetni a kriptográfiai biztonságot: a faktorizáció (IFP: Integer factorization Problem), diszkrét logaritmus probléma (DLP), hátizsák probléma (KSP), stb., melyekről részletesebben szólunk majd a kriptográfiai rendszerek biztonságával foglakozó fejezetben.

A kriptográfiai rendszerek alapját a kriptográfiai algoritmusok biztosítják, melyek egzakt leírását a szakirodalomban *kriptográfiai primitívek*nek nevezik, a kriptográfiai primitívek tipikus – klasszikus matematikai problémákkal kapcsolatos – *kriptográfiai függvények*et használhatnak fel (ld. lentebb).

A kriptográfiai primitívekre épülnek a *kriptográfiai sémák*, azokra a *kriptográfiai protokollok*. Ezeket használják fel a *kriptográfiai alkalmazások*ban, melyeket komplex informatikai rendszerekben működtetnek. A biztonsági elvárások teljesüléséhez minden szinten biztosítani kell a követelményeknek biztonságos eljárásokat, valamint ezek konzisztenciáját.

A kriptográfiai rendszer hierarchiáját mutatja az alábbi összegzés (melynek részletezése későbbi fejezet tárgya):

- a kriptográfiai primitívek (kriptográfiai algoritmusok),
  melyek tipikus – klasszikus matematikai problémákkal kapcsolatos – kriptográfiai függvényeket használhatnak fel (pl. véletlenszám generálás, véletlen permutációk alkalmazása, véges testbeli műveletek ...).
- kriptográfiai sémák,
  melyek a primitívek kritográfiai alkalmazását vezérlik:
  - meghatározzák, hogyan bontsuk blokkokra a rejtjelzendő üzenetet;
  - hogyan kell kiegészíteni a csonka blokkokat;
  - hogyan kapcsolódjanak egymáshoz a blokkok az üzenet rejtjelzése folyamán;
  - hogyan alakítsuk ki a rejtjelzéshez a kulcsokat (a kulcsok egyeztetése már a protokollok területe);
- kriptográfiai protokollok:

  melyet a partnerek közötti kapcsolat határoz meg. A protokoll a résztvevők közötti egyértelműen meghatározott lépések sorozata, amely két, vagy több

résztvevő között zajlik le a biztonsági elvárások maradéktalan teljesülésének érdekében.

A protokolloknak illeszkedniük kell a kommunikációs protokollokhoz. Alkalmazásukat meghatározzák a biztonsági elvárások: például a kulcskialakítási és szétosztási lehetőségek és követelmények (ld. kulcscsere protokollok), vagy speciális elvárások (mint a partnerhitelesítési, résztvevői kiegészítő védelmi elvárások, pl. zero-knowledge, secret sharing ...).

- kriptográfiai alkalmazások:

  A kriptográfiai rendszer fenti elemei nem öncélúak, hanem egy általános informatikai alkalmazást segítő komplex feladat megoldásához szükséges elemek.

  Komplex kriptográfiai rendszert valósít meg egy védett levelező rendszer, a mobil kommunikáció magánszférába tartozó (privacy) információinak védelmét megoldó GSM rendszer kriptográfiai alrendszere vagy az elektronikus fizetés védelmére kidolgozott SET (Secure Electronic Transaction) ...

6. A kriptográfiai biztonságot veszélyeztető tényezők, a főbb támadási módszerek osztályozása

Az alábbi osztályozás természetesen nem diszjunkt csoportokra bontja a lehetséges eseteket, ráadásul nem is törekedhet teljességre az esetszámok hihetetlen nagy száma (sok esetben ismeretlen alesetek) miatt sem, csak a legfőbb típusokat szándékoztunk kiemelni, amelyeket a védelmi módszerek tervezésénél szükséges figyelembe venni.

6.1. A támadás megcélzott eredménye szerint (cél lehet):

- a rejtjelzett üzenet visszaállítása;
- megoldókulcs megszerzése;
- hamisítás elkövetése (pl. üzenetblokkok kihagyása, felcserélése, módosítása ...), elektronikus aláírás hamisítása;
- véletlentől való megkülönböztetés felismerése /distinguish attack/, mely egy algoritmikus támadás kiindulópontja lehet;
- forgalomanalízisból értékes információk szerzése ...

6.2. A támadó aktivitása szerint

Passzív módszerek (passive attack):

Amikor a támadó „csak" a hozzáférhető titkosított szöveget analizálja (interception, eavesdropping, wiretapping /pl. switch-eknél/), megfigyeli a kommunikációs – nyílt – csatornát vagy elektronikusan tárolt rejtjeles szöveget. Beletartozik a lehetőségébe:

- a nyílt szöveg valamilyen *a*-priori feltételezése, kipróbálása (azaz összetartozó nyílt-rejtjeles pár/ok/ vizsgálata), így az összetartozó nyílt-rejtjeles párokból próbálja a támadó meghatározni a titkos kulcsot, mellyel azután más rejtjelzett szövegek védett üzenetét is visszaállíthatja;
- known-key attack: néhány kulcselőzményből következtetnek a következő (új) kulcsra.

Hasonlóan ide tartoznak a lehallgatás egyéb módszerei, pl. forgalomanalizálás, az eszközök működése közbeni elektromágneses kisugárzás vételéből származó technikai információk felhasználása (melyekhez nem kell „hozzányúlni" a kriptográfiai algoritmust végrehajtó hardverhez, távolból lehet levenni, analizálni a támadáshoz szükséges információkat).

Aktív módszerek (active attack):

Amikor a támadó megkísérelheti a rejtjelszöveget módosítani (törölni, hozzáírni, blokksorrendet változtatni), rákényszeríteni a legális alkalmazót valamilyen számára nem tervezett műveletre (valamilyen szöveget titkosíttatni/aláíratni), a kommunikációba harmadik félként belépni (ld. pl. az „*intruder in the middle attack*", vagy ennek speciális esetét a „*grandmaster chess*" támadást ...).

Tipikus példái ennek az úgynevezett visszajátszás (*replay attack*), megszemélyesítés (*impersonation attack*), összefésülés (*interleaving attack*) ...

Beletartozik a „fegyvertárba" valamilyen „side information" aktív megszerzése (pl. a titkosító kulcs megszerzése, részinformáció szerzése a titkosítás folyamatából az eszköz aktív támadásával, pl. *power analysis attack*, *timing analysis attack* ...).

Ilyen támadásokat tipikusan Smart Cardokra dolgoztak ki, amikor a kártyán külön van a processzor és ennek tápellátása, s a kettő közötti vezetéken mérhetők a fizikai jelek. Ilyen lehetőségeket bizonyítottak és prezentáltak például az ASIACRYPT 2000 konferencián Mehdi-Laurent Akkar, Régis Bevan, Paul Dischamp, and Didier Moyart „Power Analysis, What Is Now Possible" c. előadásukban (megjelent Springer-Verlag kiadónál).

A DES chipek energiafelvételből történő támadását is részletesen elemezték több kutatásban, például a „Public Key Cryptography: 5th International Workshop on Practice and Theory in Public Key Cryptosystems, Paris, France, February 2002. konferencián hangzott el „**Differential Power Analysis**" címmel Paul Kocher, Joshua Jaffe, and Benjamin Jun előadása, melyben ábrákkal szemléltették a power analysis attack működését a DES működése során. Az RSA hasonló támadásáról később lesz szó.

Másfajta aktív támadást mutatott be a **CRYPTO 2000 konferencián**, „*Differential Fault Attacks* on Elliptic Curve Cryptosystems" címmel, ahol még bon-

tásvédelemmel ellátott eszközökben is hibabit beszúrásával (bejuttatásával) olyan működési hibát lehetett előidézni, mely az elliptikus görbéken alapuló kriptorendszer támadását teszi lehetővé. (A módszer a korábban RSA-ra kidolgozott „differential fault attack" továbbfejlesztése: ld. pl. D. Boneh, R.A. DeMillo, and R.J. Lipton: On the Importance of Checking Cryptographic Protocols for Faults, Lectures Notes of Computer Science 1233, Proceedings of EUROCRYPT'97, Springer, pp. 37–51.)

6.3. Támadási technikák szerint

A támadási technikákat is sokféleképp lehet csoportosítani.

Algoritmikus:

- Ezen belül számítási „erőfölényt" kihasználó (ld. „*brute force attack*", „*exhaustive attack*" pl. a DES-re: EFF Cracking DES projekt /The Electronic Frontier Foundation projekt, ld.: `http://www.eff.org/descracker/` /.
- A rendszerek strukturális összefüggéseit kihasználó, matematikai eszköztárat igénybevevő támadások (statisztikai, algebrai, számelméleti ... módszerekkel)

A felhasználó közreműködését kiváltó támadások:

- A támadót ráveszik, valamilyen üzenet rejtjelzésére/aláírására, melyből a támadó sikeres támadást hajthat végre (pl. az I. világháború előtt az osztrák-magyar monarchia rejtjelfejtői úgy fejtették meg a bevezetett olasz katonai kódkönyvet, hogy érdeklődésre számot tartó „fal" üzenetet jelentettek meg egy konstantinápolyi újságban, melyet az olasz katonai attasé rejtjelezve hazaküldött). Ennek egyik mai alkalmazása, amikor elektronikus aláírási rendszerekben egy „közjegyzőt" rávesznek tetszőleges üzenet aláírására, mely után hamisítani lehet egy másik üzenet aláírását.
- „Social engineering attack" támadások, amikor a felhasználót a támadó ráveszi valamilyen, a rendszer biztonságát gyengítő tevékenységre (Nemrég jelent meg magyarul Kevin Mitnick – A legendás hacker c. könyve, melyben sok példa olvasható, amikor a befolyásolás és rábeszélés eszközével megtévesztik a felhasználókat, meggyőzik őket, a rendszer gyengítését okozó tevékenységek végzésére).

Fizikai eszközt igénybevevő aktív támadási módszerek:

pl. üzenetátirányítás, hamisítás, intruder in the middle attack, power analysis attack, timing analysis attack ...

6.4. A támadás inputja szerint

a) Pusztán rejtjelszövegből végrehajtott támadás /***Ciphertext only attack***/

Klasszikus passzív támadás:
itt a támadó ismerheti a nyílt szöveg statisztikai tulajdonságait (ezzel ellenőrzi a támadás sikerességét), pl.

- a nyelvszerű statisztikát;
- vagy a DES FFT támadásánál csak azt feltételezik, hogy a nyelvszöveg rejtjelzett karakterei az összes lehetséges byte érték kb. negyedét vehetik fel (kis- és nagybetűk, számok, írásjelek);
- vagy esetleg kihasználhatják, ha azonos nyílt blokkok kerültek – különböző kulcsokkal, vagy eltérő nyílt szöveg azonos kulcsokkal rejtjelzésre, pl. OTP (One-time-pad rejtjelzés) esetén, illetve egyes protokollok támadása esetén, ahol sok üzenethez azonos protokollépesek rejtjelződnek le, stb.

b) Nyílt szövegből /nyílt töredékből/ végrehajtott támadás /***plaintext and corresponding ciphertext attack***/.

Itt a támadó feltételezhet nyílt szöveget:

- Pl. a file rejtjelzése során az utolsó blokk vége 8 db. H00 érték, hasonló feltételezés működik Hellman első DES fejtő ötleténél,
- vagy JPEG képek első blokkja jellegzetes struktúrájú,
- az EXE file-ok első két byte-ja: „MZ", azaz H4D, H5A, decimálisan: 77,90, stb.

Érdekes side information attackról szól John Kelsey: Compression and Information Leakage of Plaintext c. cikke, mely elhangzott a Fast Software Encryption 9th International Workshop, FSE 2002 konferencián. Ebben a tömörített adat (az input és output méretének eltérése) tömörítési aránya ad side-információt.

(Az inputról valami információ kiszivárgása valószínűbb, mint a kulcsról.

Például, ha tudjuk, hogy egy 1 MB-os file 1 KB-osra tömörödött, tudjuk, hogy nagyon redundáns volt.

Triviális, ha van néhány valószínűsített üzenet, ezek között a támadó tud választani.)

c) Választott rejtjelszövegű – és hozzátartozó nyílt szövegű – támadás /***Selected ciphertext and corresponding plaintext attack***/. Ide tartozó gyakorlati módszer pl. a *differential cryptoanalysis.*

Ennek alesete az adaptív választott rejtjelszövegű támadás, amikor a menetközbeni eredmények szerint választják a következő rejtjeles blokkot. /*Adaptive chosen-ciphertext attack*/

Példa erre Lars R. Knudsen and John Erik Mathiassen „A *Chosen-Plaintext Linear Attack* on DES" c. cikke, mely elhangzott az EUROCRYPT 2001 konferencián.

Speciális algoritmusokra egyre újabb módszereket dolgoznak ki, példa erre John Kelsey, Tadayoshi Kohno, and Bruce Schneier *Amplified Boomerang Attacks* Against Reduced-Round MARS and Serpent, EUROCRYPT 2001-es konferencián elhangzott előadása. A bumeráng attack egy adaptív választott nyílt-rejtjelszövegpár alapú támadás.

További algoritmusra Jongsung Kim, Dukjae Moon, Wonil Lee, Seokhie Hong, Sangjin Lee, and Seokwon Jung szerzőktől az „Amplified Boomerang Attack against Reduced-Round SHACAL" c., ASIACRYPT 2002-es konferencián elhangzott ismertetőt említjük.
Figyelemre méltó speciális vizsgálati módszereket fejlesztettek ki például a Clinton adminisztráció által 1993 áprilisában javasolt Skipjack algoritmusra, melynek nyilvánosságra kerülése után ezt tudományos körökben is intenzíven analizálták. Új módszerrel való támadás hangzott el a Fast Software Encryption 9th International Workshop, FSE 2002 konferencián „Saturation Attacks on Reduced Round Skipjack" címmel Kyungdeok Hwang, Wonil Lee, Sungjae Lee, Sangjin Lee, and Jongin Lim szerzőktől. Szintén ennek vizsgálatáról hangzott el előadás „Flaws in Differential Cryptanalysis of Skipjack" címmel Louis Granboulan-től az FSE 2001 konferencián. A 2002-es Fast Software Encryption 9th International Workshop konferencián „New Results on Boomerang and *Rectangle Attacks*" címmel Eli Biham, Orr Dunkelman, and Nathan Keller szerzőktől hangzott el előadás a differenciális és lineáris kriptoanalízis kombinációs támadásáról, ahol a differenciák lineárisan közelíthetők.

d) Választott nyílt – és hozzátartozó rejtjeles – szövegű támadás /***selected plaintext and corresponding ciphertext attack***/. (Például a blind signature eljárást lehet így hatékonyan támadni.)
Ennek a módszernek alesete az adaptív választott nyíltszövegű támadás, amikor a menet közbeni eredmények szerint választják a következő nyílt blokkot. /***Adaptive chosen-plaintext attack***/

e) A ***rejtjelrendszerről nyilvánosságra került információkból*** történő támadás.
Erre jó példa az RSA modulus faktorizálása a nyilvános kulcsokból. Ehhez nem kell rejtjelszöveg, viszont pl. sikeres faktorizálás esetén az RSA alapú rejtjelzés/aláírás fejthető/hamisítható. (Hasonló nyilvános információkból történő támadás más PKI rendszereknél is elvileg lehetséges.)

6.5. A támadás célpontja szerint

- Kriptográfiai algoritmus feltörése, kulcs megszerzése.
- Kriptográfia protokoll támadása.
- Kriptográfiai rendszer támadása (kulcsmenedzsment, kulcsfelhasználás, algoritmus felhívása ...).

(Pl. Bruce Schneier, Adam Shostack „Breaking Up Is Hard To Do: Modeling Security Threats for Smart Cards, `http://www.counterpane.com/smart-card-threats.html`) című cikkben külön elemzi az egyes – rosszindulatú – résztvevők lehetséges támadásait SmartCard-okat használó rendszerekben a többi résztvevővel szemben, kiemelve az alábbi kockázatokat:

- Jogosult résztvevők támadásai más résztvevőkkel szemben:
  - terminál tulajdonos támadásai
    - a kártyabirtokos vagy adattulajdonos ellen;
    - a kártyakibocsátó ellen;
  - a kártyabirtokos támadásai
    - a terminál ellen;
    - az adattulajdonos ellen;
    - a kártya kibocsátója ellen;
    - a szoftvergyártó ellen;
  - a kártyakibocsátó támadásai a kártyabirtokos ellen;
  - a gyártó támadásai az adatok tulajdonosa ellen.
- Nem jogosult résztvevők támadásai más résztvevőkkel szemben:
  - kívülállók támadásai lopott kártyákkal.
- Kooperatív támadások.

Vizsgálják különböző kriptográfiai sémák viselkedését véletlenszerű hardver hibák kihasználása szempontjából (látens hibák, tranziens hibák, indukált /gerjesztett/ hibák), például ilyen vizsgálat szerepel D. Boneh, R. A. DeMillo, R. J. Lipton, On the Importance of Eliminating Errors in Cryptograpic Computations cikkében.

Röviden, csak a lényeget kiemelve elmondhatjuk, hogy a titkos kulcsú algoritmusokkal szemben az alábbi minimálisan elvárt követelménycsoportokat emelik ki, melyek teljesítése elengedhetetlen, ugyanakkor ezek még nem biztosítják pl. a kalkulációs biztonságot. Ilyen minimális követelmények (a fogalmak pontosítása nélkül):

Bitsoros algoritmusokkal szembeni követelmények:

- legyen hosszú periódus, nagy kulcstér;
- legyen a generált sorozat (bit, byte ...) egyenletes;
- lineáris és ugrás-komplexitása véletlenszerű legyen;
- Lempel–Ziv komplexitása véletlenszerű legyen;
- a generált sorozat differenciasorozata egyenletes legyen;
- állapottér elemei közötti korrelációk véletlenszerűek legyenek.

Blokkos algoritmusokkal szembeni követelmények, alaptámadási módok:

- Elegendően *nagy kulcstér:* a kulcstér teljes kipróbálása ellen („brute force attack").
- *Lavinahatás* teljesülése: hiba, ha a bemenet csak kis mértékben változtatja meg a kimenetet, elég valamilyen közelítő bemenet megtalálása, mind a kulcs, mind a nyílt bemenet lavinahatása szükséges).
- *Statisztikai egyenletesség,* statisztikai összefüggések kizárása:
  Statisztikai egyenetlenségek strukturális hibákra mutatnak, melyeket esetleg ki lehet használni. Az egyenetlenség csökkenti a keresett ismeretlen entrópiáját, így csökkenti a sikeres teljes kipróbálás esetszámának várható értékét.

- *Lineáris kriptoanalízissel szembeni ellenállóképesség:*
  A kódoló leképezések által meghatározott ANF (algebrai normál forma) egyenletek közelíthetőek lineáris formulával, az így kapott közelítő lineáris egyenletrendszerek Gauss-eliminációval könnyen megoldhatók. Sok ilyen lineáris egyenletet megoldva közelítik az ismeretlen kulcsokat.
- *Differenciál kriptoanalízissel szembeni ellenállóképesség:*
  Input-output közötti differencia-sorozatok egyenetlenségéből a felhasznált kulcsokra lehet következtetni (szűkíteni).
- *Találkozunk középen támadás kivédése:*
  Ismert bemenet és kimenet-pár esetén véletlenszerű kulcsokkal rejtjelezve a bemenetet és véletlenszerű kulcsokkal megoldva a kimenetet a két halmaz közös elemeit keressük; ha találunk, akkor az ismeretlen kulcsú egyszeri rejtjelzést a két ismert kulcsú rejtjelzés/megoldás kompozíciójára vezettük vissza.
- *Láncolt alkalmazásnál ne legyen rövid periódus* (pl. CBC üzemmódú blokkrejtjelző csupa nulla sorozatra).
  Stream cipher alkalmazásnál ez a tulajdonság mint kulcsismétlés közvetlenül támadható.

A titkos kulcsú algoritmusokkal szemben támasztott, fent nagyon vázlatosan ismertetett minimálisan elvárt követelménycsoportok teljesítése elengedhetetlen, ugyanakkor ezek még teljes körűen nem garantálják a biztonságot, melynek sok megközelítése van (ld. később).

7. Kriptográfiai rendszerek elemei

7.1. Kriptográfiai primitívek

A kriptográfiai primitívek meghatározása elemeik pontos meghatározását jelenti:

- jelölések és *input* pontos meghatározása (kulccsal rendelkező primitívek esetén a nyílt bemenő adatok és a kulcs /plain text- key/ formátumának, méretválasztékának megadása);
- *output* pontos meghatározása;
- *algoritmus* egzakt leírása (teszteredményekkel adott input-output párokra);
- esetleges *feltételezések* megadása.

A kriptográfiai primitívek három nagyobb osztálya:

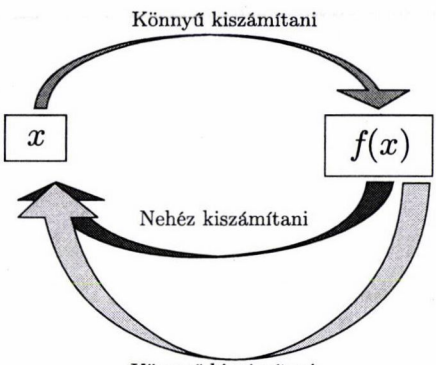
- kulcsnélküli primitívek;
- titkos kulcsú primitívek;
- nyilvános kulcsú primitívek.

7.1.1. Kulcsnélküli primitívek

7.1.1.a. Egyirányú leképezés

A leképezés nyilvános, mindenki megvalósíthatja, és könnyű kiszámítani (gyorsan: a bemenő paraméterek méretének függvényében polinomiális időben számítható), viszont az inverz leképezésre nem ismert polinomiális idejű algoritmus (ld. pontosabban a 8.2. fejezetben).

Csapda egyirányú irányú leképezés (trapdoor one way function)



A fordított irányú leképezés egy rejtett (*magán*) kiegészítő információ birtokában hatékonyan számítható. (Ld. például az RSA nyilvános kulcsú rendszerben a két prím, melyből számítható a visszafelé irány, vagy a hátizsák algoritmusoknál a csapda információt egy szupernövekedő sorozat jelenti.)
Jelentős az egyirányú leképezések felhasználása a jelszóképzések területén. Jelszóképzésre vonatkozó legfontosabb szabványok:
FIPS 112, Password Usage May 1985.
FIPS 181, Automated Password Generator October 1993.

7.1.1.b. Hash függvények

A (bináris) hash függvények: $M \to H(M)/\{0,1\}^n \to \{0,1\}^m$ leképezést valósítanak meg egy tetszőleges n ($\geq n_0$) hosszúságú bináris sorozatot egy rögzített m ($\geq m_0$) hosszúságú sorozatra képezve le. Ennek felhasználási területe lehet az üzenet integritásának (sértetlenségének) igazolása, ilyen értelemben a hibajelző kódok egy változatának is tekinthetők (pl. CRC: Cyclic Redundancy Check), mely kódok megfelelőek voltak a kommunikáció /tárolás/ véletlen hibáinak kimutatására, azonban a szándékos módosítások okozta integritás-megsértést nem tudják kivédeni. Egyik legfontosabb felhasználási területe az elektronikus aláírási rendszerekben van, ahol nem a teljes dokumentumot írják alá (aláírási primitívvel), hanem csak egy lenyomatát, amit hash függvénnyel állítanak elő. Ehhez elengedhetetlen, hogy adott lenyomathoz ne lehessen másik olyan dokumentumot találni, amely azonos lenyomatot (így azonos aláírást) képez. Ilyen tulajdonságokkal a hagyományos hibajavító kódolók nem rendelkeznek (mint alább látni fogjuk, nem is egyszerű megfelelő hash függvényt konstruálni).

Hash függvényekre nagyon sok algoritmust javasoltak, ilyenek például: az MD család (Message Digest rövidítéséből: MD2, MD4, MD5), az SHA család (SHA-0,

SHA-1, SHA-256, HSA-384, SHA-512), RIPMD-160, és egyebek (HAVAL, N-HAsh, Snefru, Tiger, Whirpool).

Szabvány leírás Hash függvényekre:

FIPS 180-2, Secure Hash Standard (SHS), August 2002. (Ebben definiálták az SHA-1, SHA-256, SHA-384 és SHA-512, valamint 2004. február 25-én kiegészítették az SHA-224-gyel.)

7.1.1.c. Kriptográfiai célú véletlenszám generálás

A kriptográfiai véletlenszám generátorok kriptográfiai alkalmazásokhoz – például kulcsok generálásához, protokollok működéséhez, esetleg véletlen feltöltésekhez /random padding/ – állítanak elő véletlen számokat.

A valódi véletlen számok valamilyen véletlen fizikai forráson alapulnak, amelynek outputja nem megjósolható. Ilyen forrás lehet például egy félvezetőből származó zaj, egy hang input legkevésbé szignifikáns bitje, vagy billentyűzet leütések közötti időtartamok. A fizikai forrásból származó zajt ezután „feljavítják", tesztelik, amely olyan outputot eredményez, amelyben a kívánt statisztikai tulajdonságok, valamint az előzményekből „jósolhatatlansági" elvárások garantálhatóak.

Legtöbb alkalmazáshoz fizikai véletlen generátor nem áll rendelkezésre, pszeudo-véletlen számokat alkalmaznak.

A hagyományos véletlen-szám generátorok, amelyek a legtöbb programozási nyelvben rendelkezésre állnak, nem alkalmasak kriptográfiai célokra (ezek valamilyen statisztikai véletlenszerűségre vannak tervezve, és nem arra, hogy ellenálljanak a kriptoanalízisnek).

Például C-nyelvben a véletlenszám generátor alapja egy kongruencia-elvű számítás:

s_0 initial vector,

$s_i \equiv as_{i-1} + b$ (modulo m),

ahol a, b és m fix konstansok, általában $m = 2^{24}$, vagy $m = 2^{32}$.

A véletlenszám generátorokra is irányt mutatnak szabványok, például **ANSI** /American National Standard/ **X9.17** (1991): Key Management; **FIPS** /Federal Information Processing Standard Publication/**186–2**, Digital Signature Standard /DSS/ (2000), Appendix 3-ban szerepel két véletlenszám generátorspecifikációja; tesztelésükre **NIST** /National Institute of Standards and Technology/ Special Publication **800–22** (2001): Statisztikai próbák a kriptográfiai alkalmazásoknál használt véletlen- és pszeudo-véletlen szám generátorokra ...

A kriptográfiai célú véletlenszám generálás követelményeiről, leggyakrabban alkalmazott eljárásairól részletesen olvashatunk Papp Pál ebben a folyóiratban megtalálható cikkében.

7.1.2. Titkos kulcsú primitívek

7.1.2.a. Egyirányú fv-ek

Az általános elnevezés szerint MAC (Message Authentication Code) eljárásaival lehet kulcshoz kötötten egy üzenet olyan tömörített leképezését előállítani, mely

a vételi hibák mellett, a szándékos módosítások kivédését is – titkos kulcstól függően – detektálja. (Ez hasonló a HASH képzéshez, csak ott nincs titkos kulcs). A legismertebb ilyen eljárás a DES láncolt üzemmódjához kapcsolódik, ahol a teljes üzenet láncolt rejtjelzése után még egy blokk rejtjelzésével, az utolsó blokk 4 byte-ja adja a MAC képet, melyet a rejtjeles blokkokhoz fűzve detektálható minden módosítás, blokkelhagyás, blokk-sorrend csere stb.

MAC-ra vonatkozó szabványok:

MAC (DAC)	*FIPS 113*,Computer Data Authentication May 1985
HMAC	*FIPS 198*,The Keyed-Hash Message Authentication Code (HMAC) March 2002.

7.1.2.b. Folyam-rejtjelzők /Stream ciphers/

A kulcsfüggő – rejtjelző – primitívek hatalmas osztályát jelentik az ún. folyamrejtjelzők, melyek karakterenkénti (bitenként, betünkénti, byte-onkénti) rejtjelzést valósítanak meg, míg a blokkrejtjelzők egyszerre több karaktert (pl. 64, vagy 128 bitet, illetve ennek megfelelő byte-ot) alakítanak át a kulcs függvényében.

A folyamrejtjelzők a karaktereket véletlen (valódi-, vagy pszeudo véletlen) kulcselemekkel módosítják bitről bitre (byte-ról byte-ra). Szokták még a biteken operáló folyamrejtjelzőt bitsoros rejtjelzőnek is nevezni.

A bitsoros rejtjelzések szerepe a blokkrejtjelzések terjedésével is jelentős, melynek sebességi, kommunikációs és biztonsági okai is vannak.

Bizonyos alkalmazásokban a blokkrejtjelzésekhez szükséges karakterek bevárása rejtjelzés előtt (kiegészülve a rejtjelzés idejével) a kommunikációban nem elfogadható, vagy a blokkos rejtjelézek esetén fellépő nagyobb hibaterjedés hátrányos (ezen okok miatt használnak a GSM rendszerben stream cipher kódolást, ld. később az A5 algoritmust). További előnye a folyam-rejtjelzőknek, hogy nagyon gyors működéssel implementálhatók célhardverben is.

Nagy számú kripto-analitikai módszert dolgoztak ki folyam-rejtjelzőkre, élesítettek, melyeket azután hatékonyan használhatnak blokk-rejtjelzők esetében is (pl. a lineáris kriptoanalízist, vagy a blokkrejtjelzők egyik tesztje az Output feedback üzemmódú stream cipher felhasználás tesztelése).

A folyam-rejtjelzők legnagyobb jelentőségét mégis az adja, hogy az egyetlen bizonyítottan megfejthetetlen rejtjelzés, az **OTP** *(one-time-pad)* is folyam-rejtjelzés. Ennél – megfelelő tulajdonságokkal generált, adó, vevő oldalra védetten szétosztott, ott védetten tárolt – valódi véletlen elemek módosítják a nyílt szöveg karaktereit.

Az eljárás a Vernan rejtjelezésen (1926) alapul. Shannon tárgyalta az információ-elméleti hátterét (1949), leírása szerint – tökéletes rejtjelezésnek nevezve – az eljárás:

- üzenet: $m_0, m_1, m_2, \ldots, m_i \in \{0, 1, \ldots, N-1\}$
- kulcs: $k_0, k_1, k_2, \ldots, k_i \in \{0, 1, \ldots, N-1\}$ (az üzenet karaktereivel azonos jelkészletű) valódi véletlen számok

- rejtjelszöveg: $c_0, c_1, c_2, \ldots$
  ahol $c_i = m_i + k_i \bmod N$

Feltétel: a küldő és fogadó ismeri a kulcsot, de a támadó nem.

Ekkor – Shannon bizonyította – a támadó a rejtjelest szövegből semmilyen következtetésre nem tud jutni a nyílt üzenetről (annak hosszán kívül).

Precíz matematikai leírás, bizonyítás olvasható [6], valamint [2] könyvekben.

A fentiek alapján megkülönböztetünk

- végtelen kulcsterű folyam-rejtjelzőt (OTP)
- és véges kulcsterű folyam-rejtjelzőket.

A véges kulcsterű folyam-rejtjelzők, mint álvéletlen generátorok legtöbbje leírható véges automata modellel, annak állapot és kimeneti függvényeivel.

A folyam-rejtjelzők legelterjedtebb osztályát a lineáris visszacsatolású shift regiszterek (jelölésben az angol betűkezdetekből: LFSR) kombinációin alapuló véletlenszám generátorok alkotják, amikor a „feedback function" (visszacsatoló függvény) $f(x_1, x_2, \ldots, x_n)$ lineáris, azaz felírható a következő formulával:

$$f(x_1, x_2, \ldots, x_n) = c_1 x_1 \oplus c_2 x_2 \oplus, \ldots, \oplus c_n x_n.$$

Általános esetben a műveletek $GF(q)$ felett értendők. Ha a c_i konstansok 0 vagy 1 értéket vehetnek fel, a művelet moduló 2 /azaz $GF(2)$/, akkor bináris lineáris visszacsatolású shift regiszterekről beszélünk.

A generált véletlen számok az output bitjei, amelyek néhány órajel (léptetés) sebességgel generálódhatnak (azaz sok GHz sebességgel az órajel függvényében, mely nagyságrendekkel nagyobb sebesség, mint a PC-ken megvalósított véletlenszám-generálás sebessége).

Lineáris shift-regiszterek előnyei:

Nagyon jól kidolgozott elméleti háttér van,

- nagy periódus: m állapot esetén a maximális periódus $2^m - 1$ ($GF(q)$ feletti LFSR esetén $q^m - 1$);
- jó tulajdonságok (Golomb postulátumok)
  egyenletesség, autokorreláció ...
- hardver-megvalósítási hatékonyság (szoftver is).

Lineáris shift-regiszterek hátrányai:

- megjósolhatóság;
  - a sorozat lineáris komplexitása kicsi (regiszter-hossznyi);
  - néhány ismert nyílt és hozzátartozó rejtjeles karakter ($2m$ bit) elegendő az állapot visszaállításához a Berlekamp–Massey algoritmussal;
- lineáris módszerekkel támadható /azaz akár a rejtjeles sorozatból is visszaállíthatóak a kezdeti induló állapotok (a kulcs)/.

Shift regiszteren alapuló konstrukciók a bonyolultság növelésére:

- filter generátorok, melyeknél az LFSR kimenete bonyolult Bool-függvénye az állapotoknak;
- kombinációs generátorok, melyeknél több lineáris visszacsatolású shift regiszter outputját – egy nem lineáris függvénnyel – kombinálják, és így a több outputból együttesen számítható a generátor kimenete.

A kombinációs generátorok legismertebb típusa az ún. *Geffe generátor*, ahol a z_i kimenet három LFSR (a_i, b_i, c_i) kimenetéből kapható: $z_i = a_i b_i + c_i(b_i + 1)$. A képlet szerint (ez az eredeti hardver konstrukcióból adódik), a három LFSR közül a középső (b_i) vezérli, hogy a kimenetet az első (a_i), vagy a harmadik (c_i) LFSR outputja legyen-e.

Egyéb LFSR-eken alapuló ismert generátorok:

- clock control /alternatív léptetés/ generátor;
- Stop-and-go generátorok;
- Gollmann cascade;
  - egymást léptető shiftregiszterek;
  - multiplexor (Jennings): az egyik LFSR mondja meg, hogy melyik állapotbitet vegyük ki a másik generátorból;
- FCSR – Feedback with carry (nemlinearitás a carry bit segítségével, például Klapper, Goresky 1995).

Léteznek nemlineáris visszacsatolású shift regiszterek is, például a De Bruijn sorozatok bizonyítottan megfelelő statisztikai tulajdonságokkal (maximális $2L$ periódussal).

Néhány gyakorlati alkalmazás:

A legáltalánosabban használt, lineáris visszacsatolású shift regisztereken alapuló alkalmazás a GSM kommunikáció rejtjelző algoritmusa, az A5, melynek két verziója van: az A5/1 és A5/2.

Ez az algoritmus egyszerre *kombinációs* (három LFSR-t használva) *és clock-control* generátor is (az A5/1 az adott három regiszterből kivett bitek alapján, míg az A5/2 algoritmus egy negyedik shift regiszterből kivett bitek alapján, ezek által vezérelve szabálytalanul lépnek az egyes regiszterek).

86 kulcsbit betöltés után 100 léptetés történik a rejtjelzés megkezdése előtt /lavinahatáshoz/, mellyel „eléggé" behatnak a nem lineáris struktúrák.

Leírását ld. pl.: `http://calliope.uwaterloo.ca~ggong/ECE710T4/lec8-ch6b.pdf` vagy `http://cryptome.org/gsm-a512.htm`.

Mind az A5/1, mind az A5/2 algoritmusára, kihasználva a protokoll hibáit is, léteznek támadási algoritmusok (ld. pl.: Alex Biryukov- Adi Shamir: Real Time Cryptanalysis of the Alleged A5/1 on a PC, 1999.)

Más, nem LFSR alapú folyam-rejtjelzéshez használt algoritmusok:

- RC4 (Rivest, RSA Security):

  Az egyszerű, bájt orientált rejtjelezést Rivest tervezte 1992-ben, mely kódolás 256 bájtos önmagát módosító permutációs táblát használ:

  $S[i]$: 256 byte elemű tömb inicializálása után:

  $i = i + 1$, $j = j + S[i] \bmod 256$

  $S[i]$, $S[j]$ felcserélése

  Kimenet $z = S\big[S[i] + S[j] \bmod 256\big]$

  Széles körben használják (pl. SSL, WEP, Windows alkalmazások).

- PKZIP (Schaffely)

  3 db 32 bites regiszter

  - 2 db lineáris shift-regiszter,
  - 1 db kongruencia generátor.

Léteznek blokkos rejtjelzésen alapuló bitgenerátorok is, melyek generálási sebessége lényegesen lassabb a fenti konstrukcióknál.

Jelentős kutatások folynak arra vonatkozóan, hogy mikor nem tudja megkülönböztetni egy feltételezett támadó a pszeudo-véletlen (kis kulcsból generált) sorozatot a valódi véletlen sorozatoktól, mert ekkor a támadó számára olyan nehézséget jelentene feltörni a kódolást, mint a valódi véletlen sorozatok esetében – azaz lehetetlen – (ld. pl. [2]).

7.1.2.c. Blokk-rejtjelző algoritmusok

1973 májusában tűzték ki egy nyilvános, széles körben alkalmazható algoritmus kidolgozását, melynek neve ***DES*** (Data Encryption Standard) lesz. Először egy LUCIFER elnevezésű jelölt algoritmust publikáltak 1975. márciusában, majd széleskörű viták után 1977. január 15-én fogadták el hivatalosan a „Feistel" struktúrán alapuló, ma DES-ként ismert algoritmust.

A *National Bureau of Standards* a FIPS No. 46 sz. publikációban szabványosította DES „Nemzeti Adatfeldolgozási Szabványt".

A blokkméret 64 bit, a nyílt üzeneteket először 64 bites blokkokra bontja (ha az utolsó blokk csonka, akkor azt kiegészíti). Ezeket ugyanazzal a kóddal képezi le ugyancsak 64-bites rejtjeles blokkokba, amelyeket egymás után írva kapjuk a rejtjeles üzeneteket. A megválasztható kódok száma 2^{56} (az 56 szabadon választható kulcsbiten kívül további 8 bit ellenőrzési célokat szolgál), tehát az effektív kulcsméret nagysága 56 bit.

A DES megfejthetőségére utaló publikációk sorát Martin Hellman egy 1977-ben tartott előadása nyitotta meg, aki a teljes kipróbálást is kivitelezhetőnek tartotta megfelelően épített hardware segítségével (akkoriban ennek költségét 20 millió dollárra becsülte).

Javításként először a Hellman ötletén alapuló fejtést gyakorlatilag kivitelezhetetlennek beállító eljárásként a DES kétszeres alkalmazását javasolták. Ehhez

az üzenetet kétszer kellett egymás után rejtjelezni két, egymástól függetlenül választott kulccsal, ami a kulcs méretét immár 112 bit hosszúságúvá tette. Ezzel kapcsolatban már 1992-ben Merkle és Hellman nyilvánvalóvá tette azonban, hogy ha a „simple DES" fejthető, akkor a „meet in the middle attack" (egy „középen találkozunk" elnevezésű módszer) lehetővé teszi a „Double DES" fejtését is. A hírek szerint több helyen építettek olyan célgépet, amellyel a Double DES-t fejteni lehet.

A DES jelenleg elterjedt változata a „Triple DES", /más jelölésekben "T-DES", „3-DES"/. Ez vagy kettő, vagy három 56 bites kulccsal dolgozik. Az üzenetet először az első kulccsal rejtjelzik normál DES módban, majd a második kulccsal a megoldó algoritmust alkalmazzák. Az így nyert közbülső szövegre alkalmazzák ismét az első, három kulcsos rendszerben a harmadik kulcsot.

A TDES struktúra algoritmusát egyes leírásokban TDEA algoritmussal jelölik:

A TDES modellje:

Rejtjelzés képlete: $C = E_{K3}(D_{K2}(E_{K1}(M)))$.

Megoldás képlete: $M = D_{K1}(E_{K2}(D_{K3}(C)))$.

A TDES-hez 3 kulcsra van szükség (K_1, K_2, K_3), melyeket az egyes standard leírásokban az alábbiak szerint specifikálnak:

1. Opció: K_1, K_2, és K_3 független kulcsok.
2. Opció: K_1 és K_2 független kulcsok, és $K_3 = K_1$.
3. Opció: $K_1 = K_2 = K_3$.

(A 3. változat előnye, hogy egyszeres DES-sel rendelkező célhardverekkel is képes kommunikálni – természetesen a biztonság rovására.)

A DES kriptoanalízisének sikerei, az amerikai hozzáállás miatti bizalmatlanság alapján az európaiak is törekedtek egy szabvány (megbízható – azaz saját) rejtjelző algoritmus elterjesztésére.

1990-ben Lai és Massey javasoltak a DES kiváltására egy **PES** (Proposed Encryption Standard) elnevezésű algoritmust, majd a differenciál kriptoanalízis nevű támadási módszer által felfedett gyengeség miatt ezt módosították, kezdeti elnevezésben a módosított algoritmus neve IPES /Improved Proposed Encryption Standard/ volt, melynek végső elnevezése – ahogy a szakirodalomban ismert, a legtöbb kriptoprotokoll rejtjelző készletében szerepel – **IDEA** /International Data Encryption Algorithm/ lett.

Bár az IDEA algoritmust a szakemberek kellő erősségűnek tartják, több protokollba is implementálták, mégsem váltotta fel a DES-t világszabványként.

A nemzetközi bizalom megrendülése a DES biztonságában inspirálta a National Institute of Standards and Technology (NIST) azon döntését, hogy ki kell fejleszteni a DES utódját, amely az Advanced Encryption Standard (AES) nevet kapta.

A NIST döntése alapján az AES algoritmust nyilvános pályázat során választották ki. Ehhez 1997 szeptemberében közzétették azon elvárásoknak a listáját, amelynek az AES algoritmusnak meg kell felelni. Ugyanakkor deklarálták,

hogy a benyújtott rejtjelzési algoritmusok nyilvánosak, szabadon felhasználhatók lesznek. A kiírás szerint blokkos (128, 196, 256 biten operáló), 128, 196 és 256 bites kulcsméret opcionálisan egyaránt választható algoritmusnak kell lenni, mely ellenáll minden ismert fejtési támadásnak. Követelmény volt továbbá, hogy hatékonyan implementálható (azaz gyors, bármely plattformon kis memóriával is megvalósítható) legyen.

A tervek szerint az elvárásoknak megfelelő rejtjelzési algoritmus hosszú távra, akár 20-25 évre is megoldja a polgári életben keletkező adatok biztonságos védelmének a kérdését.

A beérkezett pályaművek közül 15 felelt meg a formai elvárásoknak. A benyújtott algoritmusok mögött komoly multinacionális cégek sorakoznak fel (*IBM, Microsoft, RSA Laboratories, Deutsche Telekom AG, Nippon Telegraph and Telephone Corporation, Centre National pour la Recherche Scientifique, stb.*) Ezek elemzése a legszélesebb nyilvánosság bevonásával folyt. Maguk a szerzők, de más kriptográfusok is elemezték az algoritmusokat, s több, kifejezetten e témának szentelt konferenciát is tartottak.

A NIST *1999-ben Rómában* rendezett konferenciáján 15 algoritmust értékeltek. A konferencia megrendezésével a NIST-nek az volt a célja, hogy az itt elhangzott értékelések segítséget nyújtsanak annak az öt algoritmusnak a kiválasztásához, amelyek továbbra is versenyben maradnak. Ezután már csak ezeknek az algoritmusoknak a vizsgálata folyt.

Az öt, a végső versenyben maradt algoritmus ábécé sorrendben:

- **Mars** algoritmus: szerzői: az IBM[2] több mint tízfős csapata (USA),
- **RC6** algoritmus: szerzői: Ron Rivest[3] és csapata, RSA Laboratories (USA),
- **Rijndael** algoritmus: szerzői: Daemen, Rijmen, belga kriptográfusok,
- **Serpent** algoritmus: szerzői: Anderson, Biham[4], Knudsen nemzetközi csapat,
- **Twofish** algoritmus: szerzői: Bruce Schneier[5] és csapata (USA).

Az algoritmusok analizálása, elemzése rendkívül nagy erőket kötött le. A viszonylag rövid, jó kétéves vizsgálati idő ellenére állítható, hogy a DES után ez az öt algoritmus a világ legmélyebben elemzett szimmetrikus kulcsú algoritmusa. A győztes algoritmust 2000. október másodikán jelentették be. A versenyt a RIJNDAEL algoritmus nyerte. Szerkezete alapján az algoritmus nem hasonlít a legtöbb blokkos algoritmusra, nem követi a DES Feistel-struktúráját. Ez is számos iterációs lépésben valósul meg. Minden iteráció három rétegből áll, ezek szerepe különböző.

[2] közülük több kriptográfus már a DES kifejlesztésében is jelentős szerepet játszott
[3] az RSA algoritmus egyik feltatálója
[4] a differenciális kriptoanalízis kidolgozója
[5] az „Applied Cryptography" c. könyv és több jelentős hatást kiváltó módszer szerzője

- *ByteSub:* nem-lineáris keverés réteg (S-dobozok),
- *ShiftRow:* pozíció-keverő réteg,
- *MixColoumn:* oszlop-keverő réteg,
- *Round Key Addition:* kulcs-addíciós réteg.

Az algoritmus több technikát alkalmaz (pl. byte-szintű operációk a 256 elemű véges test fölött).

7.1.3. Nyilvános kulcsú primitívek

7.1.3.a. PKI kódolók

RSA kódoló:

Az RSA rendszer olyan ismert, hogy részletes leírását itt mellőzzük.

- Előnyei:

  könnyen megérthető algoritmus;
  biztonsága klasszikus (egész számok faktorizálása, IFP) problémára vezethető vissza.

- Hátrányai:

  ha $m(= p * q)$ nagy, nagyon számításigényes (lassú).

Hátizsák (knapsack) probléma, Merkle-Hellman algoritmus

Legyenek $a_1, a_2, \ldots, a_n$ természetes számok, ún. „súlyok", $\underline{x_i} \in \{0,1\}^n$ bitvektor

$$S = \Sigma x_i a_i$$

Ekkor S, $a_i - k$ ismeretében „nehéz" x_i-ket meghatározni, ugyanis a KnapSack (hátizsák) probléma NP-teljes, ha a_i-k véletlenszerűek.

Viszont könnyű a meghatározás, ha a_i-k „szupernövekedő" sorozatot alkotnak (pl. 2 hatványai).

Az ezen alapuló rejtjelzés leírása szerepel a 8.2. redukciós biztonságot tárgyaló fejezetben.

Elliptikus görbék pontjain értelmezett DLP (Elliptic Curve Discreet Logarithm Problem, ECDLP)

Néhány ECDLP-n alapuló algoritmus:

- *Elliptic Curve Diffie–Hellmann Key Agreement (ECDH),*
- *Elliptic Curve Menezes-Qu-Vanstone (ECMQV),*
- *Elliptic Curve Integrated Encryption Scheme (ECIEC),*
- *Elliptic Curve Digital Signature Algorithm (ECDSA),*
- *Elliptic Curve Nyberg-Rueppel (ECNR).*

Az EC-n alapuló rejtjelző algoritmus rövid leírása:

A $P(x,y) : y^2 = x^3 + ax + b \pmod p$ görbe pontjain értelmezünk egy algebrai struktúrát az alábbi műveletekkel:
Összeg: $P(x_1,y_1) + P(x_2,y_2) = P(x_3,y_3)$:

- $x_3 = k^2 - x_1 - x_2$
- $y_3 = k(x_1 - x_3) - y_1$, ahol
  $k = (y_2 - y_1)/(x_2 - x_1)$, ha $P(x_1,y_1) \neq P(x_2,y_2)$ egész;
  $k = (3x_1 + a)/(2y_1)$, ha $P(x_1,y_1) = P(x_2,y_2)$.

Ezen a struktúrán is értelmezhető az úgynevezett diszkrét logaritmus probléma: $e * P = Q$ probléma megoldása e-re. A felsorolt algoritmusok mutatják, hogy ebben a struktúrában is létrehozhatók a nyilvános kulcsú algoritmusok megfelelői. A rendszer előnye az RSA-hoz képest, hogy lényegesen kisebb számokkal is elérhető a biztonság, mert ebben a struktúrában nem tudunk olyan hatékonyan diszkrét logaritmust számolni.

Több más nyilvános kulcsú primitív is ismert (pl. Rabin, ElGamal, McEliece, Chor-Rivest knapsack ..., ld. pl. [5]), itt csak szemléltettünk néhányat.

7.1.3.b. Aláíró primitívek

RSA alapú aláírás algoritmusai:

Ismert az aláíró előtt	Ismert az ellenőrző előtt	
Prímek: p, q $n = p * q$ Kulcsok: d, e $d * e = 1 \bmod (p-1) * (q-1)$	n e y	Bár Európában az RSA a legelterjedtebb, legismertebb aláíró primitív, de ismertek (Amerikában inkább használatosak) a **DSA**, és az **elliptikus görbéken** alapuló primitívek is, ld. **FIPS 186-2** szabványt.
Az x üzenet aláírása: $y = \text{hash}(x)^d \bmod n \Rightarrow$	Ellenőrzés: $\Rightarrow z = y^d \bmod n$ $Z = \text{hash}(x)$?	

További, kevésbé ismert primitívek:

XTR (Efficient Compact Subgroup Trace representation)

ld. `www.ecstr.com`

- Egyesíti az RSA, ECL előnyeit (gyorsaság, kis memória igény, kis méretek).
- Erőssége a DLP-n alapul (Diszkrét Logaritmus Probléma).

NTRU

ld. `www.ntru.com`

- Rácsredukción alapul.
- Még rövidebb programkód, paraméterméretek.

7.1.4. Törekvések új primitívek, szabványok kialakítására – A NESSIE projekt

A NESSIE (New European Schemes for Signatures, Integrity and Encryption) projekt 2000 januárjától 2003 márciusáig tartott. A projektről angol nyelven a `www.cryptonessie.org` honlapon található részletes információ.

A projekt – ellentétben az AES projekttel – a bizalmasságot, adat sértetlenséget és hitelesítést szolgáltató primitívek széles készletének szabványosítását tűzte ki célul: ezek a primitívek blokkos rejtjelezéseket, bitsoros rejtjelezéseket, hash függvényeket, MAC (üzenet hitelesítő kód) algoritmusokat, digitális aláírás sémákat és nyilvános kulcsú rejtjelzési rendszereket tartalmaznak. A végső cél a kriptográfia területén az európai kutatás erős pozíciójának megtartása és az európai ipar pozíciójának erősítése volt.

A meghirdetett kategóriák nagy részében elfogadtak egy vagy több kriptográfiai primitívet. A következő táblázat azt mutatja, hogy az ajánlott primitívek között kategóriánként mindig akadt egy európai fejlesztésű is, de a többi mind tengerentúli (amerikai vagy japán) volt. A bitsoros rejtjelző algoritmusok kategóriájában nem fogadtak el ajánlásra egyetlen pályázatot sem. Ez részben annak volt köszönhető, hogy nagyon erősen határozták meg a biztonsági szintet, elvárták a véletlentől való megkülönböztethetetlenséget is, amit adott szinten egyik algoritmus sem tudott teljesíteni. A kulcs teljes kipróbálásánál kevesebb lépésben mindegyik algoritmust meg lehetett különböztetni a valódi véletlen sorozattól. Emiatt a bitsoros rejtjelző algoritmusokra egy új projektet írtak ki a 2005–2007 időszakra, amelynél csak azt a minimális követelményt fogalmazzák meg, hogy a valódi véletlentől való megkülönböztethetőséghez legalább 2^{64} lépésre legyen szükség.

A következő táblázat azt mutatja, hogy a pályázatokat mely kategóriákban hirdették meg, az egyes kategóriákban hány pályázat érkezett, a végső fázisban hányat ajánlottak közülük és melyek ezek. Az utolsó oszlopban dőlt betűvel szereplő elemek az USA-ban használatos szabványok, melyeket szintén ajánlanak európai használatra is.

Típus	Érkezett	Ajánlott
Blokkos rejtjelző algoritmus	17 db	MISTY-1[6] (Mitsubishi) Camellia (Nippon) SHACAL-2 (Gemplus,F) *AES (USA FIPS 197)*
Bitsoros rejtjelző algoritmus	6	–
Önszinkronizáló bitsoros rejtjelző algoritmus	0	–
Üzenet hitelesítő kód (MAC)	2	Two-Track-MAC (B,D) UMAC (USA) *CBC-MAC (ISO/IEC 9797-1)* *HMAC (ISO/IEC 9797-1)*
Hash függvény	1	Whirlpool (B, Brazilia) *SHA-xxx (USA FIPS 180-2)*
Nyilvános kulcsú rejtjelző algoritmus	5	ACE Encrypt (IBM Zürich) PSEC-KEM (Nippon) *RSA-KEM (ISO/IEC 18033-2)*
Digitális aláírás séma	7	SFLASH (F) ECDSA (USA, Kanada) RSA-PSS (USA)
Aszimmetrikus azonosítási séma	1	GPS (F)

IEEE P1363:

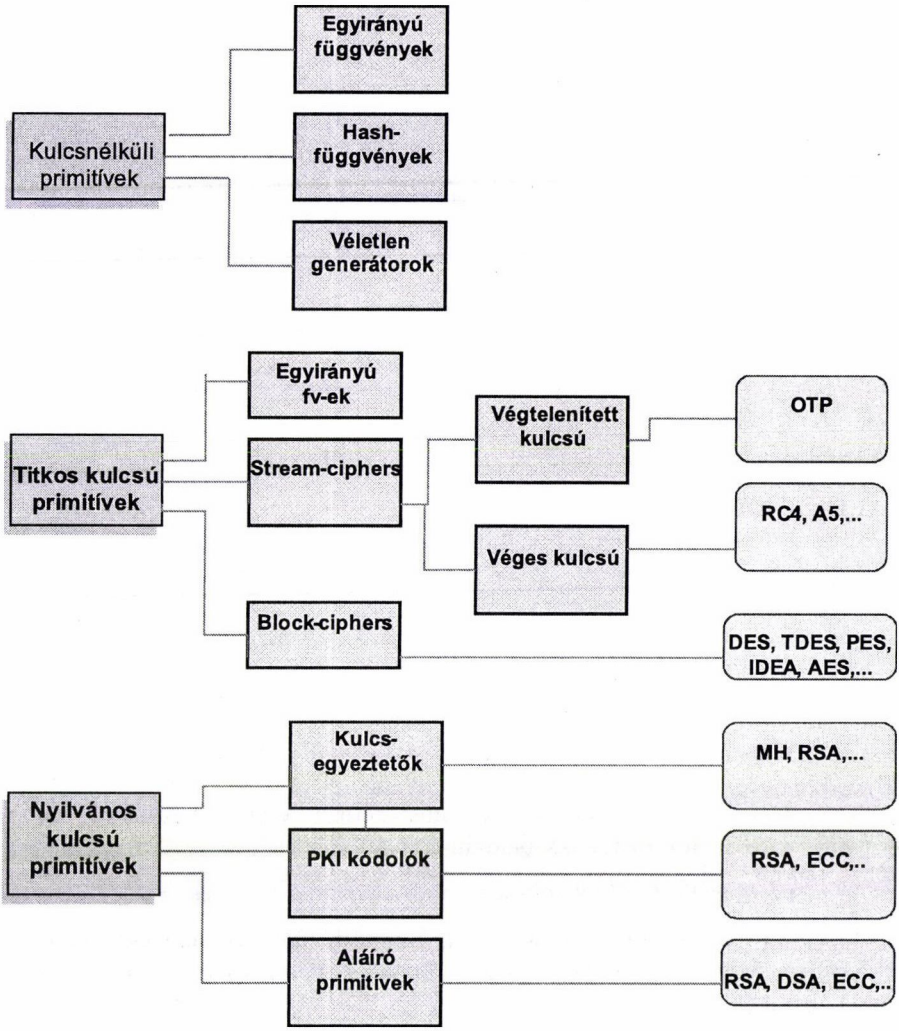
„Standard Specifications for Public-Key Cryptography" céljára indított projektről részletek olvashatók az interneten.

7.2. Kriptográfiai sémák

A kriptográfiai sémák feladata annak meghatározása, hogy a kriptográfiai primitívek hogyan rejtjelezzék a teljes rejtjeles üzenetet, azaz mit kezdjenek a rejtjelzőblokknál hosszabb üzenetekkel (darabolás), ezeket hogyan láncolják össze, mit kezdjenek az utolsó, nem teljesen kitöltött blokkal, milyen kulcsokat használjanak, hogyan generálják a kulcsokat, stb. Közbülső réteget képeznek a primitívek és a protokollok között. Ide tartoznak például az alábbiak:

- üzenet feldarabolása blokkokra;
- blokkfeltöltés (pl. utolsó blokk esetén), feltöltés jelölése; az elektronikus aláírás területén szabványként használandók az „emsa-pkcs1-v1_5", „emsa-pss" blokkfeltöltő eljárások;
- blokk-rejtjelzés üzemmódjai;
- véletlen választás, prímteszt;
- KAS: Key Agreement Sceme;

[6] A MISTY blokkos algoritmuson alapuló bitsoros rejtjelző algoritmus lett a 3. generációs mobil telefonok rejtjelző algoritmusa.



2. ábra. A primitívek összefoglalása

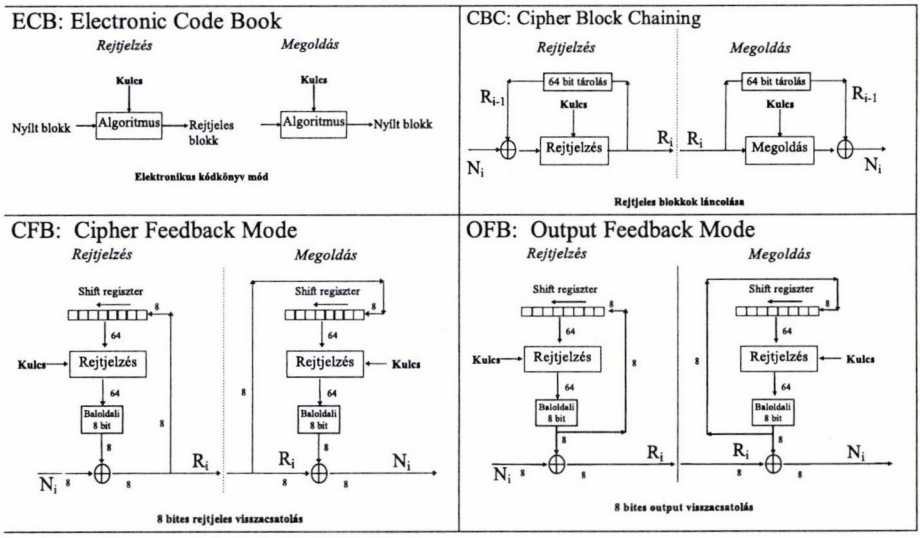
– SSA: Signature Scheme with Appendix; stb.

7.2.a. Blokkrejtjelző üzemmódok:

Közismertek a blokkos rejtjelzések különböző „üzemmódjai", melyek akár DES, TDES, AES, IDEA esetén használhatóak.

A blokkos algoritmusok fix méretű adatblokkokkal operálnak, a valóságban azonban egy üzenet bármilyen méretű lehet. Esetenként olyan adatfolyamot kell rejtjelezni, amelynek a hossza sem ismert. Ehhez hasonló problémák megoldására a blokkos algoritmust különböző üzemmódokban használhatjuk. Ezek a szabványosított[7] működési módok a gyakorlatban nagyon hasznos segítséget nyújtanak.

Például az elektronikus kódkönyv mód (ECB), melynél a 8 byte méretű blokkokra bontott nyílt információ az algoritmus inputja (DES esetén), s az output lesz a rejtjeles. A rejtjeles blokkjai egymástól függetlenek, vagyis egy potenciális támadó törölheti, beszúrhatja, átrendezheti a rejtjeles blokkokat. A legnagyobb veszélyt ebben a módban az jelenti, ha olyan dokumentumokat rejtjeleznek így, amelyeknek csak egy kicsi része változik, az egyéb struktúrák változatlanok maradnak. Pl. egy szerződés, ahol mindig csak az összeg változik. Ekkor egy kevés oldalinformáció (side information) segítségével összeállítható egy táblázat, hogy melyik rejtjeles blokkhoz milyen összeg tartozik. Használata rövid (egy-két blokk) üzenetek esetén ajánlott.



[7] Information processing – modes for operation for a 64-bit block cipher algorithm-ISO 8372

A következő szabványok a visszacsatolási módok pontos leírását adják:
FIPS 81, December 1980.
SP 800–38A 2001 ED, December 2001.
SP 800–38C, May 2004.

7.2.b.

Az RSA rendszernél (de több más nyilvános kulcsú primitívnél) fontos nagyon nagy *prímszámok generálása*, mely az előkészítő fázishoz tartozik. Ilyen sémához tartozó eljárások a nagy véletlen számokra vonatkozó *valószínűségi tesztek* (pl. Miller-Rabin teszt, Lucas prímteszt, Solovay-Strassen prímteszt), melyek *gyorsak*, de *nem döntik el teljes biztonsággal*, hogy a kérdéses szám prím-e. Azonban a tévedés valószínűsége a teszt többszöri végrehajtásával – ha mindig pozitív a válasz – *tetszőleges küszöbérték alá csökkenthető*. Így ezek a módszerek kriptográfiai célokra – például RSA kulcsgenerálásra – megfelelőek. Az algoritmushoz szükséges paraméterek választási szempontjai, eljárásai már a sémákhoz tartoznak. (Pl. az RSA esetén milyen kiegészítő feltételeknek kell a prímszámoknak eleget tenni. Egyes rendszerekben megkötik, hogy $(p-1)$-nek és $(q-1)$-nek is legyenek nagy prímosztói, sőt ezen nagy prímosztókat eggyel csökkentve ennek is legyenek nagy prímosztói. Egyes gyorsítási ötletekkel szemben az e nyilvános, d titkos kulcsra is különböző alsó korlátokat szokás kikötni, ld. az RSA kalkulációs biztonságáról szóló fejezetet.)

7.2.c.

A kriptográfiai sémákra (az algoritmusok és környezetük egységes szerepének kezelésére) jó példát nyújt az **EESSI-SG** (European Electronic Signature Standardisation Initiative Steering Group) felügyelete alatt dolgozó Algoritmus csoport (**ALGO**) javaslata az elektronikus aláírási készletekre.

Az *elektronikus aláírások* megbízhatósága (sértetlenséget, letagadhatatlanságot biztosító tulajdonságai) alapvetően támaszkodik az egész technológia alapját képező kriptográfiai algoritmusok (és azok paraméterei) megbízhatóságára.

Az elektronikus aláírás biztonságát érintő lehetséges kölcsönhatások miatt az algoritmusok és paraméterek csak előre meghatározott kombinációkban használhatók, melyeket együttesen **aláírás-készletnek** neveznek.

Egy *aláírás-készlet* a következő *elemekből* áll:

- aláíró algoritmus a paramétereivel

  Az aláíró algoritmus az aláírandó dokumentum hash-értékéből képezi az aláírást az aláírás-létrehozó adat (magánkulcs) felhasználásával.

- kulcsgeneráló algoritmus

  A véletlenszám generálásnak az aláírás-létrehozó adat (magánkulcs) generálásánál, illetve bizonyos kriptográfiai algoritmusok (pl. DSA) véletlen paramétereinek generálásakor van jelentősége. Bizonyos esetekben a

hash-függvény feltöltésénél is fontos lehet. Ezért a véletlenszám generálásra vonatkozó kritériumokat mindig a feltöltési módszerekkel és a kulcsgeneráló algoritmusokkal összefüggésben szükséges megadni.

Egy kriptográfiai kulcs megtalálásához, valamint a generátor belső állapotáról bármilyen információ megtalálásához szükséges kipróbálás várható értékének legalább annyinak kell lennie, mint egy **EntropyBits** hosszú véletlen érték megtalálásához.

Egy pszeudo véletlenszám generátort valódi véletlen számmal kell inicializálni.

Ezt a számot kezdőértéknek hívjuk, hossza **SeedLen.**

Négy – az aláíró algoritmushoz rendelt – kulcsgeneráló algoritmust fogadtak el szabványosnak: Rsagen1, Dsagen1, Ecgen1, Ecgen2, melyek valódi véletlen vagy pszeudo véletlen számokból generálnak kulcsokat. Mindegyiknél meghatározott a kulcsgenerálás minimális szabadsági foka: EntropyBits$\geq$128, vagy SeedLen$\geq$128[8].

- feltöltési eljárás

  Bizonyos aláíró algoritmusoknál szükség van a hash-érték kiegészítésére az algoritmus által meghatározott blokk-hosszúságra (pl. RSA modulus hossza). Amennyiben egy aláíró algoritmusnak szüksége van feltöltési eljárásra, akkor ennek ki kell elégítenie a *normatív referenciák*ban található követelményeket.

- kriptográfiai lenyomatolási (hash) függvény.

Ha az aláírás-készlet bármely eleme megbízhatatlan, az egész készlet használata kérdőjeleződik meg.

Ha az aláírás-készlet bármely elemét érvénytelenítik, akkor maga a készlet is érvénytelenné válik. Ha a készlet bármely elemét frissítik, a készletet is frissíteni kell.

Az EU ETSI (ALGO Group) szakbizottsága által biztonságosnak elfogadott aláírás-készletek:

Az elfogadott aláírás-készletek listája az alábbi táblázatban található.

[8] Egy ilyen – a követelményeknek eleget tevő – véletlenszám generátor leírása található [Blum, M and Micali, S: „*How to generate cryptographically strong sequences of pseudo-random bits.*" SIAM Journal on Computing, vol. 4, No. 13, pp. 850–863, 1984]-ben

Készlet sorszáma	Aláíró algoritmus	Aláíró algoritmus paraméterei	Kulcsgeneráló algoritmus	Feltöltő eljárás	Hash-függvény
001-004	Rsa	MinModLen=1020	Rsagen1	emsa-pkcs1-v1_5, vagy emsa-pss	sha1, vagy ripemd160
005	Dsa	PminLen=1024 QminLen=160	Dsagen1	–	sha1
006	Ecdsa-Fp	QminLen=160 r0Min=10^4 MinClass=200	Ecgen1	–	sha1
007	Ecdsa-F2m	QminLen=160 r0Min=10^4 MinClass=200	Ecgen2	–	sha1
008, 009	Ecgdsa-Fp	QminLen=160 r0Min=10^4 MinClass=200	Ecgen1	–	sha1, ripemd160
010, 011	Ecgdsa-F2m	QMinLen=160 r0Min=10^4 MinClass=200	Ecgen2	–	sha1, ripemd160

Megjegyzés: Az RSA algoritmus modulus hosszának minimuma 1020 bit a megszokottabb 1024 bit helyett. Ez lehetővé tesz olyan megvalósításokat, amelyek nem tudják használni a legfelső bit(ek)et.

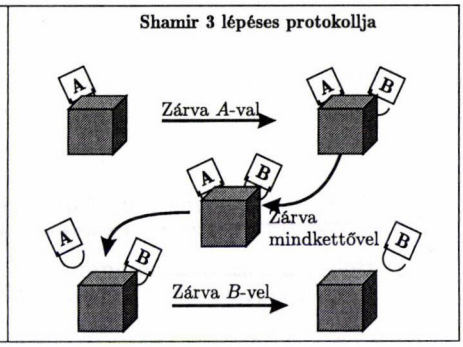
7.3. Kriptográfiai protokollok

Ahogy már említettük, a kriptográfiai protokollokat a partnerek közötti kapcsolat határozza meg. A protokoll a résztvevők közötti egyértelműen meghatározott lépések sorozata, mely két, vagy több résztvevő között zajlik le a biztonsági elvárások maradéktalan teljesülésének érdekében.

A protokolloknak illeszkedniük kell a (megfelelő OSI rétegben aktivizálódó) kommunikációs protokollokhoz. Ezeket meghatározzák a biztonsági elvárások: például a kulcskialakítási és szétosztási lehetőségek és követelmények (ld. kulcscsere protokollok), vagy speciális elvárások (mint a partnerhitelesítési, résztvevői kiegészítő védelmi elvárások, pl. zero-knowledge, secret sharing ...).

7.3.1. Kulcsokkal kapcsolatos protokollok

A kulcsok egyeztetése a legősibb, protokollt igénylő feladatok közé tartozik. Ennek legegyszerűbb formája Shamir híres, három lépéses protokollja:



A nyilvános kulcsú rendszerekkel egyidős a Diffie–Hellmann kulcsegyeztető protokoll, amely először mutatott rá a nyilvános kulcsú kriptográfia lehetőségeire.

Diffie–Hellman féle kulcsegyeztető protokoll:

Legyen a két szereplő $U1$ és $U2$.

Választanak egy csoportban (pl. $GF(p)$) felett egy g primitív (nyilvános) elemet:

Titkos kulcsaik k_{U1} és k_{U2} véletlenül választott pozitív egész számok; ekkor *elküldik egymásnak:* a $P_{U1} = g^{kU1}$ és $P_{U2} = g^{kU2}$. (Azt a DLP megoldásának nehézsége garantálja, hogy g^{kUi} – vagyis a nyilvános kulcsrész – ismeretében egy harmadik fél nem tudja meghatározni k_{Ui}-t, a titkos kulcsot.)

A kulcsegyeztetés:

A Diffie–Hellman protokoll szerint a g^{kU1kU2} érték a közös titok.

Speciális protokollt valósítanak meg a kulcskezeléssel kapcsolatos egyedi elvárások, pl. az alábbiak:

Titkosító kulcsok kezelése

Kulcs letét (*key escrow*): egy titkosító nyilvános kulcs magánkulcs párjának megőrzése a kulcs visszaállítás támogatása érdekében. (Ezt szimmetrikus kulcsú rendszerekben is használják, elsősorban a rendszer működtető szervezet titkosított adatvagyonának visszaállíthatósága érdekében. Az USA-ban Clinton elnök direktívát adott ki a Skipjack algoritmust megvalósító „Clipper Chip" készülékekhez rendelt titkos részkulcsainak két állami szervben megosztott tárolásáról, mely nagy vihart váltott ki, a rendszer nem vált működőképessé. Ld. még **FIPS 185**, Escowed Encryption Standard.)

Kulcs visszaállítás (*key recovery*): egy letétbe helyezett kulcsról másolat készítése, és ezen kulcsmásolat átadása egy erre jogosult kérelmezőnek. A kulcskezelés követelményeiről részletesen olvashatunk a **FIPS 140-2** szabványban: Security Requirements for Cryptographic Modules.

7.3.2. Egyéb, funkcionális protokollok

Mivel a biztonsággal kapcsolatos sokrétű igények a protokollok szintjén jelennek meg, ezért ez a terület rendkívül sokrétű, szerteágazó. Ezért csak néhány példát tudunk mutatni különböző biztonsági feladatot kezelő protokollokra:

- üzenethitelesítő protokoll,
- partnerhitelesítés,
- elektronikus aláírás protokolljai (XAdES, X509v3, CRL lista-kezelés ...),
- zero-knowledge protokoll,
- blind signature protokoll,
- Secret Sharing (titokmegosztási) protokoll.

Terjedelmi okokból ezeket itt nem részletezzük, csak ez utóbbiról példaként adunk rövid áttekintést:

A titokmegosztási protokollok hatékony segítséget nyújtanak informatikai rendszerek kritikus pontjainak védelméhez. A kritikus információ (kulcs, jelszó) szétosztása csökkenti a rendszer függőségét az üzemeltető személyektől, illetve véletlen adatvesztés ellen is védelmet adhat.

A titkos információt olyan módon kell felosztani n személy között, hogy tetszőleges m ($m < n$) személy együttesen rekonstruálni tudja azt, de m-nél kevesebb személy ne legyen képes rekonstruálni a titkot semmilyen esetben sem. A fenti feladatot nevezzük ezután (m, n)-titokmegosztási protokollnak.

Blakley szellemes és szemléletes konstrukciója szerint legyen a titok egy pont (ennek koordinátái) a háromdimenziós térben. Legyenek a titokdarabok a pont vetületei a (x, y), (x, z), (y, z) tengely által meghatározott síkra. (Ez az *eredete az árnyék elnevezésnek.*) Minden ilyen (titok)pont egy egyenest határoz meg, amelyen a titoknak rajta kell lennie. Tehát bármely két titokbirtokos vissza tudja állítani a titkot, ez egy $(2, 3)$ küszöb séma. Természetesen egyéb protokollok is megvalósíthatók a tér és az alterek dimenzióinak megfelelő megválasztásával. Ez a protokoll nem tökéletes, mivel elvárás, hogy nem kellő számú titokbirtokos együttese ne rendelkezzen a titokról szűkítő információval, mely ebben a protokollban nem teljesül, hiszen a titokdarab birtokosa tudja, hogy a titok melyik egyenesen (altérben) van, míg egy kívülálló nem. Megjegyezzük még, hogy itt általában nem euklideszi, hanem véges projektív térben dolgozunk.

A titokmegosztási feladat másik (m, m) modelljét egy p elemű véges testben definiálták. Véges testek felett értelmezett polinomok valamennyi együtthatója a véges test eleme. A titokmegosztási feladatban használt polinom együtthatói véletlenszerűen generáltak és $a_k < p (k = 1, \ldots, m-1)$, illetve a nulladfokú tag együtthatója maga az M titok.

A titok szétosztásához $k_i = (a_{m-1}x_i^{m-1} + a_{m-2}x_i^{m-2} + \ldots + a_1 x_i + M) \pmod p$ értékeket kell kiszámolni az x_i előre meghatározott alappontokban. Célszerű az $1, 2, \ldots, n$ számokat alappontnak kijelölni. A szétosztás során minden résztvevő két releváns adatot kap. A polinom helyettesítési értéke mellett szükség van annak megjegyzésére is, hogy az adott személy mely alappontban kiszámított polinomértéket őrzi. Mindkét adat szükséges a titok visszaállításához.

Már a fenti Shamir-séma is figyelembe tudja venni azt, ha a titokdarabhordozók nem egyformán fontosak. Osszuk az árnyékok hordozóit két csoportra, az első csoport tagjai kapjanak egy titokdarabot, míg a második csoporthoz tartozók kettőt. Ha négy titokdarab szükséges a visszaállításhoz, akkor a második csoportból bármely kettő, az első csoportból bármely négy személy állíthatja vissza a titkot, illetve a harmadik lehetőség az, hogy az

első csoportból kettő, a második csoportból egy titokdarab-birtokosra van szükség a visszaállításhoz.

Általában igaz az, hogy ha egyesével meghatározzuk azokat a részhalmazokat, amelyek jogosultak a titok visszaállítására, akkor konstruálható olyan titokmegosztási séma, amely ezt teljesíti.

7.3.3. Interneten használt komplex (bizalmasságot, integritást, hitelességet biztosító) protokollok

Elterjedtségük miatt külön figyelmet érdemelnek a kriptográfiai primitíveket, sémákat alkalmazó, ***internetes*** kommunikáció biztonságát segítő ***protokollok***.

Az első széles körben elterjedt, biztonságos kommunikációs csatornát megvalósító protokoll a *Netscape* által 1994–95-ben kifejlesztett *SSL (Secure Socket Layer)* volt. 1999-ben az *IETF (Internet Engineering Task Force)* elfogadott szabvány szintre emelte az SSL-t *RFC-t 2246-os számmal*, az SSL továbbfejlesztése a TLS (Transport Layer Security) is internetes szabvány. Ezzel párhuzamosan 1998-ban a *WAP Forum* megjelentette a *WTLS protokoll*t (Wireless Transport Layer Security) tervezetét is, amely a drótnélküli kommunikáció (*WAP, Wireless Application Protocoll*) szabványa. (A WTLS gyakorlatilag a TLS mobil környezetre adoptált változata.)

A főbb követelmények ezekkel a protokollokkal szemben:

- titkosság (secrecy),
- együttműködési képesség (interoperability),
- továbbfejleszthetőség (extensibility),
- relatív hatékonyság (relative efficiency): minél gyorsabb megvalósíthatóság, kevesebb tárolási igény.

7.3.3.a. Az SSH protokoll

A kriptográfiai protokollok a kriptográfia alkalmazásának kiemelt fontosságát mutatják. A kriptográfiai alapelemek ezen protokollok részeként szolgálnak ki egy adott biztonsági elváráshalmazt. A legrégebbi ilyen protokoll az SSH /Secure SHell/, de rendkívül elterjedt az SSL/TLS protokoll is. Az SSH-t részletesebben bemutatjuk, beletekintve a protokoll szerkezetébe, a kriptográfiai alapelemek (primitívek, sémák) beillesztéséhez használt formalizmusba is. Ez a legtöbb protokoll esetében hasonló, ezért a további protokollokat már csak vázlatosabban ismertetjük.

Az SSH protokoll 3 fő részből áll:

A *Transport Layer Protocol* végzi a szerver hitelesítését, és egységes, titkos felületet nyújt. Tartalmazhat tömörítést is. Általában TCP/IP kapcsolaton fut, de bármilyen más megbízható adatfolyamon futhat.

A *User Authentication Protocol* végzi a kliens oldali felhasználó hitelesítését. A Transport Layer Protocol-on fut.

A *Connection Protocol* osztja fel a titkosított csatornát több logikai csatornára. A User Authentication Protocol-on fut.

Két formátum használatos az algoritmus elnevezésekhez:

Az olyan nevek, amelyek nem tartalmaznak „kukacot" (@) azokat az IANA (Internet Assigned Numbers Authority) jelöli ki. Ilyenek például az '3des-cbc', sha-1', 'hmac-sha1' és zlib', az idézőjelek nem részei a neveknek. Ilyen formátumú nevek csak az IANA regisztrációja után használhatók, és nem lehet bennük vessző (,) és kukac (@).

További algoritmusokat bárki elnevezhet a név@domainnév formátum alapján, pl. „ourcipher-cbc@ssh.fi". A formátum @ előtti része tetszőleges, de US-ASCII karakterekből áll és nincs benne vessző és @. A másik rész valós Internet domain név kell, hogy legyen [RFC-1034], amit az adott cég, illetve személy felügyel, aki a nevet definiálta. Az a domaintől függ, hogyan használja a lokális nevet.

Kódolás

A kódoló algoritmust és a kulcsot a kulcs-egyeztetés alatt választja meg a két fél. A kulcs mérete minimum 128 bit kell, hogy legyen. Minden csomag az adott irányban egy adatfolyamnak tekinthető, vagyis az inicializációs vektorok az egyik csomag végéről a másik csomag elejére kell, hogy mutassanak.

A kódolásnak a két irányban egymástól függetlenül kell működnie, és mindkét irányban lehetőséget kell adni különböző kódoló algoritmusok használatára.

Jelenleg az alábbi algoritmusok vannak definiálva:

3des-cbc	kötelező	3 kulcsos DES CBC módban
blowfish-cbc	javasolt	blowfish CBC módban
twofish-cbc	javasolt	twofish CBC módban
aes256-cbc	javasolt	AES (Rijndael) CBC módban, 256 bites kulccsal
aes192-cbc	opcionális	AES 192 bites kulccsal
aes128-cbc	opcionális	AES 128 bites kulccsal
serpent256-cbc	opcionális	serpent CBC módban, 256 bites kulccsal
serpent192-cbc	opcionális	serpent CBC módban, 192 bites kulccsal
serpent128-cbc	opcionális	serpent CBC módban, 128 bites kulccsal
arcfour	opcionális	az ARCFOUR adatfolyam kódoló
idea-cbc	opcionális	IDEA CBC módban
cast128-cbc	opcionális	CAST-128 CBC módban
none	opcionális	nincs kódolás; NEM JAVASOLT

A „3des-cbc" 3 kulcsos DES (kódolás–dekódolás–kódolás), ahol a kulcs első 8 byte-ját használják az első kódolásra, a következő 8-at a dekódolásra és az utolsó 8-at a végső kódolásra. Ehhez tehát 24 byte-nyi kulcs kell (amiből 168 bit kell a kódoláshoz). A CBC mód a külső láncolást jelenti (vagyis hogy csak egy inicializációs vektor van). Ez blokkos kódolás, 8 byte-os blokkokkal.

A „blowfish-cbc" szintén blokkos kódolás, 8 byte-os blokkokkal, CBC módban, 128 bites kulccsal; részletesen megtalálható [7]-ban.

A „twofish-cbc" 256 bites, CBC módban, 16 byte-os blokk-kódolással dolgozik, részletesen tárgyalja a Twofish AES beadvány.

Az „aes256-cbc" az „Advanced Encryption Standard", ami a Rijndael néven algoritmus lesz, CBC módban, 256 bites kulccsal.

Az „aes192-cbc" mint fent, de 192 bites kulccsal.

Az „aes128-cbc" mint fent, de 128 bites kulccsal.

A „serpent256-cbc" CBC módban, 256 bittel dolgozik, részletesen kifejtve a Serpent AES beadványban.

Az „serpent192-cbc" mint fent, de 192 bites kulccsal.

Az „serpent128-cbc" mint fent, de 128 bites kulccsal.

Az „arcfour" az Arcfour 128 bites kulcsú adatfolyam kódoló, elvileg kompatibilis az RC4 kódolással. Az RC4 az RSA Security Inc. védjegye. Az Arcfour (és az RC4) nem biztonságos a gyenge kulcsokkal, ezért figyelmesen kell használni.

A „cast128-cbc" a CAST-128 kódolás CBC módban [RFC-2144].

A „none" azt az algoritmus jelenti, hogy a továbbítás alatt nincs használva semmiféle kódolás. Ebben az esetben az adatok nincsenek védve illetéktelenek elől, ezért lehetőleg kerülni kell a használatát. Bizonyos funkciók ki lehetnek iktatva biztonsági szempontból e kódolás esetén (például a jelszavas hitelesítés).

Egyéb algoritmusokat az előbbiekben meghatározott módszer alapján lehet specifikálni.

Adatintegritás

Minden adatcsomag tartalmaz MAC-ot, ez védi az adat integritását. A MAC-ot egy megosztott, közös csomagsorszámból és a csomag tartalmából számolja ki az algoritmus. Az üzenethitelesítő algoritmust és kulcsot a kulcs-egyeztetés alatt dönti el a két fél. Kezdetben nincs MAC, a hossza így nulla. A kulcs-egyeztetés után, kódolás előtt a MAC-ot a csomag tartalmából kiszámolja az algoritmus:

$$\text{mac} = (\text{MAC kulcs, sorszám}|| \text{ kódolatlan\_csomag}),$$

ahol a kódolatlan_csomag az egész csomagot jelenti MAC nélkül (a hosszmezők, payload, padding), a sorszám egy implicit csomagsorszám uint32-ben ábrázolva. A sorszám 0 az első csomagnál, és minden csomagnál emelkedik (függetlenül attól, hogy MAC vagy kódolás használatban van-e). Nincs sose visszaállítva, még kulcs/algoritmus újraegyeztetéskor se, viszont minden

2^32-dik csomag után újra nulla lesz. A csomag sorszám maga sose továbbítódik, mivel nincs benne a csomagban. A MAC algoritmusok mindkét irányban függetlenül futnak, minden implementációnak meg kell engednie, hogy különböző legyen az algoritmus a két félnél. A MAC algoritmus által előállított MAC byte-ok kódolás nélkül, a csomag végéhez fűzve továbbítódnak. A MAC byte-ok száma a választott algoritmustól függ.

Jelenleg az alábbi algoritmusok vannak definiálva:

hmac-sha1	kötelező	HMAC-SHA1 (digest hossz = kulcshossz = 20)
hmac-sha1-96	javasolt	a HMAC-SHA1 első 96 bitje (digest hossz = 12, kulcs hossz = 20)
hmac-md5	opcionális	HMAC-MD5 (digest hossz = kulcshossz = 16)
hmac-md5-96	opcionális	a HMAC-MD5 első 96 bitje (digest hossz = 12, kulcs hossz = 16)
none	opcionális	nincs MAC; NEM JAVASOLT

A „$hmac-*$" algoritmusok leírása megtalálható: [RFC-2104]. A „$*-n$" MAC algoritmusok az eredmény első n bitjét használják csak.

A hash algoritmusok leírása megtalálható a [7]-ben.

A „none" módszer NEM JAVASOLT, ugyanis ebben az esetben egy aktív támadó módosítani tudja az átvitel alatt lévő csomagot.

Egyéb módszerek definiálhatók az adott specifikációk szerint.

Nyilvános kulcsú algoritmusok

A protokoll képes együttműködni a legtöbb (signature és/vagy kódolás alapú) nyilvános kulcsú formátummal, kódolással és algoritmussal.

Különböző aspektusok alapján lehet definiálni a nyilvános kulcs típusát:

Kulcs formátum: hogyan van kódolva a kulcs, hogyan van reprezentálva a bizonyítvány (certificate).

Signature és/vagy kódolás algoritmusok. Néhány kulcstípus nem támogatja mindkét módszert. A kulcshasználat ugyancsak korlátozva lehet a bizonyítvány alapján. A különféle irányelvekhez különböző kulcstípusokat kell definiálni.

A signature és/vagy a védett adat kódolása.

Az alábbi nyilvános kulcs és/vagy bizonyítvány formátumok vannak jelenleg definiálva:

ssh-dss	kötelező	egyszerű DSS
ssh-rsa	javasolt	egyszerű RSA
x509v3	javasolt	X.509 bizonyítvány
spki	opcionális	SPKI bizonyítvány
pgp	opcionális	OpenPGP bizonyítvány

Egyéb kulcstípusokat az ismert módon lehet definiálni.

A kulcstípusnak explicit módon ismertnek kell lennie (algoritmus egyeztetés vagy egyéb forrás alapján). Általában nem a kulcs részben van.

A bizonyítványok és nyilvános kulcsok az alábbi módon vannak kódolva:

string	bizonyítvány vagy nyilvános kulcs formátum azonosító,
byte[n]	kulcs/bizonyítvány adat.

A bizonyítvány rész lehet 0 hosszúságú string, de a nyilvános kulcs kötelező. Ez a nyilvános kulcs lesz használva a hitelesítéshez.

Ezt a kulcs formátumot használva a jelölés és ellenőrzés a Digital Signature Standard [FIPS-186] alapján történik, az SHA-1-et használva. Leírása megtalálható a [7]-ben.

A kulcs-egyeztetés eredménye

A kulcs-egyeztetés két értéket ad vissza: egy közös titkos K, és egy egyeztetett hash H értékét. A kódoló és hitelesítő kulcsok ezekből származnak. A H, ami az első kulcs-egyeztetésből származik, használatos továbbá még mint session azonosító, amely egy egyedi azonosítója a kapcsolatnak. A hitelesítő eljárások használják az adatnak olyan szignált részeként, ami a privát kulcs létezésének a bizonyítéka. Ha előállt, a session azonosító nem változhat, még akkor se, ha később újabb kulcs-csere történik.

Minden kulcs-egyeztetési módszer meghatároz egy hash függvényt, ami a kulcs-egyeztetés alatt használatos. Ugyanezt a hash algoritmust kell használni a kulcs előállításánál. Jelen esetben HASH néven fog szerepelni.

A kódoló kulcsokat a HASH függvénnyel kell előállítani egy ismert értékből és a K-ból a következőképpen:

– Kezdeti IV kliens ⇒ szerver: HASH(K || H || „A" || session_id) (itt K mpint, „A" byte és a session_id mint sima adat. „A" a sima A karakter, ASCII 65).

– Kezdeti IV szerver ⇒ kliens: HASH(K || H || „B" || session_id).

– Kódoló kulcs kliens ⇒ szerver: HASH(K || H || „C" || session_id).

– Kódoló kulcs szerver ⇒ kliens: HASH(K || H || „D" || session_id).

– Integritás kulcs kliens ⇒ szerver: HASH(K || H || „E" || session_id).

– Integritás kulcs szerver ⇒ kliens: HASH(K || H || „F" || session_id).

A kulcs adatot a hash kimenetének elejéből kell venni. 128 bitet (16 byte-ot) kell használni a változó hosszúságú kódú algoritmusokhoz. Egyéb algoritmusokhoz annyit kell venni az elejéről, amennyire szükség van. Ha a kulcs mérete hosszabb, mint a HASH eredménye, a kulcsot ki kell egészíteni a K, a H és a kulcs összefűzésének HASH értékével. Ezt addig kell ismételni, míg nem lesz elég adat a kulcshoz; a kulcs ennek az eleje lesz. Vagyis:

$K1 = \text{HASH}(K \,||\, H \,||\, X \,||\, \text{session\_id})$ (X mint pl. „A")
$K2 = \text{HASH}(K \,||\, H \,||\, K1)$
$K3 = \text{HASH}(K \,||\, H \,||\, K1 \,||\, K2)$

...
kulcs $= K1 \,||\, K2 \,||\, K3 \,||\, \ldots$

Biztonsági jellemzők

Az SSH protokoll biztonságos csatornát alakít ki egy gyenge védelmű hálózaton. Tartalmaz hitelesítést, kulcs-egyeztetést, kódolást, integritás védelmet. Egyedi session azonosítót generál, amit magasabb szintű protokollok használhatnak.

Előfordulhat, hogy a protokoll úgy van használatban, hogy a host neve és a host kulcsa közti megfeleltetés nem garantált biztonságú. Így nem annyira biztonságos, de megfelelő lehet a nem kritikus biztonság-igényű alkalmazások esetében, és passzív támadások ellen így is védelmet nyújt. Ezt mindenképp figyelembe kell venni implementáláskor.

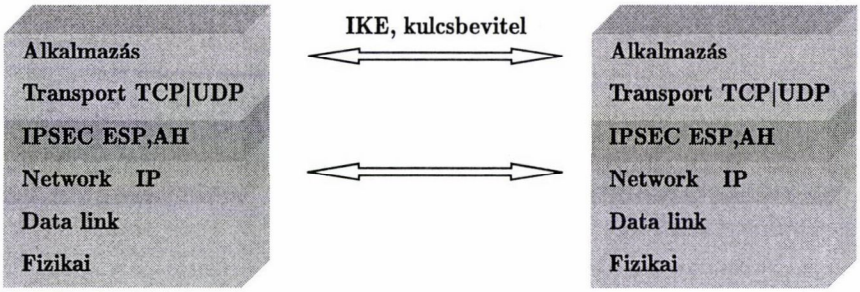
A protokollt átviteli szempontból megbízható hálózaton való használatra tervezték. Ha átviteli hibák vannak, vagy manipulálják az üzeneteket, a kapcsolat lezárul. Ilyenkor a kapcsolatnak újra kell épülnie. Az ilyen típusú Denial-of-Service támadásokat nem védi ki.

A protokoll tartalmaz potenciális rejtett csatornákat. Például a padding, az SSH_MSG_IGNORE üzenetek, és még egyéb helyek a protokollban használhatók rejtett információk átvitelére, és a fogadó fél nem tud megbízható módon meggyőződni arról, hogy ilyen információt fogad.

7.3.3.b. Egyéb fontos protokollok

A hálózati biztonság terén használt de facto standard protokollmegoldások között a két legelterjedtebb a TLS (Transport Layer Security) protokoll és az IPsec (Internet Protocol SECurity). A TLS/SSL protokollt használja a legtöbb web böngésző a http-kapcsolatok védelmére, ezért nagyon jelentős a szerepe pl. az elektronikus kereskedelemmel kapcsolatos kezdeményezésekben. Gyakorlatilag a TLS a világ legtöbbet használt biztonsági protokollja. Ezzel szemben az ***IPSEC*** hostok közötti tetszőleges adatforgalom védelmére szolgál, a virtuális magánhálózatok létrehozásának standard eszköze.

A két protokoll a protokoll stack különböző szintjein dolgozik. A TLS protokoll a TCP réteg fölött helyezkedik el. A gyakorlatban úgy működik, hogy nyit egy új portkapcsolatot, s az eredeti kommunikáció ezen a védett csatornán folyik. Ezzel szemben az IPSEC protokoll a TCP alatt, az IP csomagok szintjén helyezkedik el.



Alapvetően ebből adódnak a két protokoll eltérő tulajdonságai.

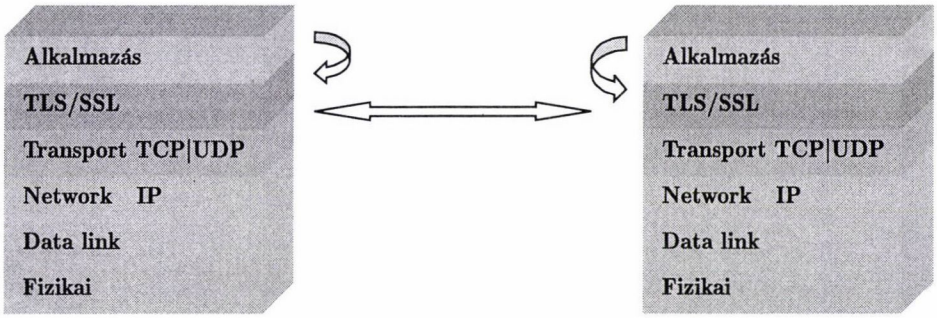
Amiatt, hogy a TLS egy extra protokollként jelenik meg a TCP fölött, a TLS egy TCP session-t képes védeni. Több session védelméhez a protokoll több példányát kell futtatni.

Az IPSEC az IP kiterjesztése, az IP csomagokat védi, ezért egyrészt teljesen transzparens módon minden fölötte folyó TCP (és UDP) kommunikációt véd, másrészt ehhez elég egy IPSEC példányt futtatni.

Mivel az IPSEC az IP szintjén fut, nem tud semmit a fölötte működő alkalmazásokról. Ezért a gyakorlatban nincsenek olyan megoldások, amelyek a felhasználó hitelesítését lehetővé tenné az IPsec alapján, csupán a két kommunikáló host hitelesítése oldható meg.

A TLS ezzel szemben képes a kommunikáló felek kölcsönös hitelesítésére, ami a legtöbb esetben elengedhetetlen (felhasználó azonosítása, jogok hozzárendelése, stb.)

Összegezve elmondható, hogy az IPSEC jelentős térnyerése ellenére ott, ahol a felhasználó hitelesítésére is szükség van, ott az IPSEC helyett a TLS (vagy más hasonló protokoll) használatát kell támogatni.



A fenti protokollokról – több más forrás mellett – a fentitől eltérő szemléletű áttekintés olvasható [2]-ben is.

A Kerberos hitelesítési protokoll

A Kerberos hitelesítési protokollt szintén nagy hálózatok központi hitelesítési mechanizmusaként használják kliens szerver viszonylatban. A Kerberos szerver központilag tárolja mind a kliensek, mind a szerverek titkos jelszavát (itt a titkos jelszó nem az aszimmetrikus kulcsú rendszerek szerint értendő). A Kerberors működése során a DES (Data Encryption Standard) rejtjelező algoritmust használja. A kliens bejelentkezik a Kerberos szerverre, majd közli azonosítóját, és az elérni kívánt szolgáltatás azonosítóját is. A Kerberos szerver véletlenszerűen generál egy ún. session key-t, a továbbiakban majd ezzel rejtjelezve zajlik a kliens és szerver közti kommunikáció. A Kerberos szerver a session key-t és a kliens azonosítóját az elérni kívánt szolgáltatáshoz tartozó titkos kulccsal lerejtjelezi, ez az ún. ticket, amit időpecséttel is ellát. A ticketeknek bizonyos érvényességi idejük van, aminek lejártakor új ticketet kell generálni. Továbbá a session key-t a kliens titkos jelszavával is lerejtjelezi, majd ebből és a ticketből összeállítja a hitelesítési tokent, amelyet a kliensnek visszaküld. A kliens saját titkos jelszava ismeretében a tokenből előállítja a session key-t, majd a ticketet az elérni kívánt szolgáltatást futtató szerverhez küldi. A szerver saját titkos jelszava ismeretében szintén előállítja a session key-t, így mindkét oldalon létrejön a kommunikáció rejtjelezéséhez használandó közös kulcs. A továbbiakban a kliens és szerver között ezzel rejtjelezve folyik a kommunikáció (opcionális), valamint a közös kulcs ismerete bizonyítja a partnerek hitelességét is. A Kerberos támogatja a realm-ok használatát is, ez a Windows NT domain fogalmához hasonló fogalom.

Az S/Key hitelesítési protokoll

Szintén a hálózatok lehallgatásával megszerezhető jelszavak problémája ellen véd az S/Key protokoll. Lényegét tekintve egy alapjelszóból egyszer használatos jelszavakat generál oly módon, hogy a hálózatot figyelő támadó a már megfigyelt jelszavakból nem szerez használható információt a következő esetben használt jelszóra. Az S/KEY jelszavak alapján történő hitelesítés a következőképpen zajlik:

– **Inicializálás**

- A kliens választ egy jelszót és egy véletlen elemet (seed vagy salt.) A véletlen elem szerepe az, hogy a kliens ugyanazt a jelszót több rendszerben is használhassa.
- A jelszót és a seed-et egymáshoz fűzi, s erre alkalmazza az MD4 hash függvényt. Az outputként kapott 128 bit első és második felét modulo 2 összeadva kapja az első, 64 bites jelszót.
- A kliens elkészíti az első száz, esetleg ezer jelszót úgy, hogy az előző jelszóra alkalmazza az MD4 függvényt, s a 128 bit output két felét összeadja.
- Az ezredik jelszót és a seed értéket átadja a szervernek, ezzel az inicializálás lezárult.

– **Hitelesítés**

- A szerver elküldi a seed értékét és a 999 számot a kliensnek, jelezve, hogy a 999. jelszó beírását kéri.
- A kliens megvalósítástól függően vagy kikeresi, vagy kiszámítja a 999. jelszót és elküldi a szervernek.
- A szerver a kapott jelszóra elvégzi az MD4 műveletet, s ha a kapott érték megegyezik a tárolt értékkel, a hitelesítést elfogadja, és a 999. érték váltja fel az ezredik értéket az adatbázisában. Legközelebb a 998. jelszót fogja kérni.

A módszer ereje a hash függvény tulajdonságaiban rejlik. Az ellenőrzés könnyű, hiszen a hash képzés gyors, viszont a következő jelszó megjóslásához fel kell törni a hash függvényt. A módszert az rfc 1760 specifikálja.

A protokoll problémái:

Az S/Key protokoll szótáras támadással támadható. Mivel a módszer nyílt, egy támadó, aki a vonalról leolvassa az üzenetváltást, próbálgatással ellenőrizheti a kliens jelszavára vonatkozó tippjét. Ezen a módon a szokásos, gyakori jelszavak kipróbálásán alapuló módszerek sikerrel alkalmazhatók gyengén választott jelszavak ellen.

A szerveren tárolt adatok védelme. A szerveren tárolt adatok (felhasználónév, számláló, seed, számlálóhoz tartozó jelszó) közvetlenül nem kompromittálják a jelszót, de megismerésük az előzőleg megismert támadási módszert teszi végrehajthatóvá, vagyis a fájlhoz való hozzáférést megfelelően kell beállítani.

Megszemélyesítés. Ha a támadó megszemélyesíti a szervert, és a következő, pl. 998. jelszó helyett a századikat kéri be, akkor ez veszélyes lehet. Ha ugyanis a kliens beírja a századik jelszót, akkor abból a támadó ki tudja számolni az összes száznál nagyobb sorszámú jelszót.

Belépésverseny. Ha a támadó nemcsak lehallgatni, hanem módosítani is képes a vonali forgalmat, elképzelhető, hogy módosítja a kliens válaszát, s így az nem tud belépni, viszont ezután a támadó be tud lépni, mert a rendszer (a sikertelen kísérlet miatt) ugyanazt a sorszámú jelszót kéri ismét.

Hash függvény támadása. A legtöbb rendszer az MD4 hash függvényt használja, amely azonban feltörhetőnek bizonyult (Hans Dobbertin: Cryptoanalysis of MD4, Fast Software Encryption, 1996), így a rendszer MD4 használata esetén alapjaiban rendült meg. Néhány rendszerben az MD5-öt használják, amely lényegesen biztonságosabb, azonban azt is érték már támadások Dobbertin részéről.

A protokoll egyszerű, hatékony, a felvetett problémák a sérülékeny hash kivételével kezelhetők. Megfelelő hash függvény használatát kell elérni az implementációban (pl. SHA-1, HMAC).

7.4. Kriptográfiai alkalmazások

A kriptográfiai alkalmazások adott biztonsági cél megvalósítása érdekében kidolgozott komplex rendszerek.

Ide tartoznak – több más mellett – pl. a következők:

- önálló biztonsági funkciókat megvalósító rendszerek (pl. elektronikus aláírási rendszerek; PGP),
- elektronikus fizetési rendszerek (pl. SET) védelmi alrendszerei,
- kommunikációs rendszerek (pl. GSM rendszer, elektronikus levelezés ...) védelmi alrendszerei.

7.4.1. Önálló biztonsági funkciókat megvalósító rendszerek

A kriptográfiai alkalmazások széleskörű csoportját alkotják azon rendszerek, melyek fő funkciója valamilyen védelmi megoldás felkínálása a felhasználóknak. Csak példaként említjük a jól ismert ***PGP*** /Pretty Good Privacy/ rendszert, mely aszinkron kommunikáció, valamint adattárolás védelmére is szolgál. A védelem nyilvános kulcsú kriptográfiára épül, de ellentétben az elektronikus aláírási rendszerekkel a nyilvános kulcsokat nem ún. CA-k (Certificate Authority-k) hitelesítik, hanem elsősorban az ún. bizalmi háló elvén épül fel a tanúsítványok elfogadása.

Az ***elektronikus aláírási rendszerek*** a magas szintű hitelesítést célul kitűző nagyon komplex alkalmazások, melyek az információs társadalom fejlődésének bizonyos szintjéhez már meghatározott jogszabályi, intézményi és technológiai rendszerben működtetendők. Magyarországon (az EU Direktíva[9] alapján) a 2001. évi XXXV. Törvény (valamint kiegészítő módosítása a 2004. évi LV törvény) az elektronikus aláírásra teremti meg a jogszabályi alapokat, melyekre végrehajtási rendeletek sokasága épült (pl. 1515/2001. Korm. Rendelet, 16/2001 MeHVM Rendelet stb.). Az elektronikus aláírási rendszer komplexitását már a résztvevők és funkciók sokasága is mutatja:

- az aláíró (küldő) és ellenőrző (címzett) közvetlen résztvevők (akiknek a törvény által nevesített ún. minősített aláíráshoz biztonságos aláírás-létrehozó eszközzel /SSCD: Secure Signature Creation Device/, valamint ellenőrzött, nemzetközi követelményeknek /CEN CWA 14170, CWA 14171/ megfelelő aláíró alkalmazásokkal kell rendelkezniük;
- a Hitelesítés-szolgáltatók CA-k, akik a nyilvános kulcsok hitelességét és érvényességét elektronikus tanúsítvány kiállításával (X509v3 tanúsítvány) tanúsítják, bonyolult és drága informatikai és szervezeti rendszerelőírásokkal; a résztvevőkre vonatkozó speciális regisztrációs eljárásokkal, az érvényességi listakezelés szigorú eléjárásaival (OCSP, vagy CRL), az időbélyegzési funkcióra vonatkozó előírásokkal ...

[9] „Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures," December 1999.

- a termékek, rendszerek, szolgáltatások megfelelőségét tanúsító szervezetek (Nemzeti Hírközlési Hatóság, Tanúsító Szervezetek), ezek feladatira jogszabályi előírások ...

7.4.2. Elektronikus fizetési rendszerek védelmi alrendszerei

Az internet szolgáltatások terjedésével egyre nagyobb szerepet kapnak olyan szolgáltatások, melyek kiteljesedéséhez pénzátutalásoknak is kapcsolódni kell. A bonyolult informatikai rendszer része az elektronikus fizetési alrendszer (EPS), mely lehet interneten keresztüli fizetés (pl. SET), vagy ún. elektronikus pénztárca, digitális készpénz[10].

Internet alapú fizetésre iKP néven került kifejlesztésre a – PKI alapú – Internet Keyed Payments Protocol[11]. Mondex néven került kifejlesztésre egy SmartCard alapú fizetési rendszer[12].

SET: Secure Electronic Transaction

A SET kifejlesztését a 90-es években a VISA és a MasterCard végezte azért, hogy elektronikus fizetéshez általánosan elfogadott, biztonságos fizetési módszer álljon rendelkezésre.

A SET résztvevői:

Kártyatulajdonos az elektronikus kereskedelemben a ***fogyasztó***, illetve a ***vevő*** a kereskedőkkel a számítógépeiken keresztül kerülnek kapcsolatba. A kártyatulajdonos egy, a kibocsátótól származó fizetési kártyát használ. A SET biztosítja, hogy a kártyatulajdonos és a kereskedő kapcsolata során a fizetési kártya számlájára vonatkozó információk titkosak maradnak.

Kibocsátó egy olyan pénzügyi intézmény, amely egy kártyatulajdonos számára számlát hoz létre. A kibocsátó garanciát vállal a kártya-használat szabályainak megfelelő tranzakciók tényleges kifizetéséért.

Kereskedő termékeket vagy szolgáltatásokat nyújt fizetség fejében. A SET segítségével a kereskedő a kártyatulajdonosoknak biztonságos elektronikus együttműködést tud nyújtani.

Számlagyűjtő egy olyan pénzügyi intézmény, amely számlát tart fenn egy kereskedő részére, és gondoskodik a fizetési kártya felhatalmazásokról és a kifizetésekről.

Fizetésszervező egy a Számlagyűjtő által működtetett eszköz vagy egy kijelölt harmadik partner, amely a kereskedők fizetéssel kapcsolatos értesítéseit elvégzi, beleértve a kártyatulajdonosok fizetési instrukcióit.

A SET az alábbi kiemelkedően fontos üzleti követelményeket célozta meg:

[10] A rendkívül szerteágazó területről összefoglalás olvasható pl. `http://wwww.w3.org/ECommerce/roadmap.html` címen.

[11] M. Bellare, J. A. Garay, R. Hauser,...: iKP – A Family of Secure Electronic Payment Protocols, 1995., ld. még `http://www.zurich.ibm.com/security/past-projects/ecommmerce/iKP.html`.

[12] `http://www.mondex.com`.

1. A fizetési információk bizalmasságának és a fizetési információkkal együtt továbbításra kerülő rendelési információk bizalmasságának megőrzése.
2. Minden továbbításra kerülő adat integritásának biztosítása.
3. Hitelesítés biztosítása, amely egy kártyatulajdonost a fizetési kártya számlához tartozó legitim felhasználóként hitelesíti.
4. A kereskedő hitelesítésének biztosítása, miszerint ő egy fizetési kártyás tranzakciókat elfogadó személy vagy cég, aki kapcsolatban áll egy számlagyűjtő pénzügyi intézménnyel.
5. Megvédeni az elektronikus kereskedelem tranzakcióinak minden résztvevője érdekeit.
6. Egy protokoll létrehozása, amely nem függ az átvitel biztonsági mechanizmusától, és nem akadályozza azok használatát.

7.4.3. Kommunikációs rendszerek védelmi alrendszerei

Nagyon sokféle kommunikációs rendszerhez fejlesztettek védelmi alrendszereket. Ide sorolhatjuk az internet használat kommunikációs védelmi alrendszereit (melyek közül a SSH, SSL, TLS, IPSEC a protokollokról szóló fejezetben szerepeltek és másokat, pl. HTTPS), a fizetős televíziózás kódolását, az elektronikus levelezés biztonsági alrendszereit, a Wireless technológia védelmi alrendszerét, stb.

7.4.3.a.

A nagyvállalati belső hálózatokban nagyon elterjedt ***Lotus Notes levelező rendszer***.
A rendszer védelmi szintjei:

- az azonosítás eszközei,
- szerver szintű biztonsági eszközök,
- adatbázis szintű biztonsági eszközök,
- adat szintű biztonsági eszközök.

Az *azonosítás* eszköze a Notes ID.
A Notes ID (azonosító) azonosítja a felhasználókat és szervereket. Az ID a felhasználó vagy a szerver regisztrálásakor keletkezik.

Szerver szinten

- port szintű hozzáférés-védelem és port szintű titkosítás; valamint
- szerver dokumentumvédelem (hitelesítés, elérési korlátozások, és felhasználói, „ügynöki") futtatási szabályok vannak.

Az adatbázis védelem titkosítással és hozzáférési listákkal (ACL) megoldott.

Titkosítás a Notes-ban

A Notes az adatok titkosítására az RC2 és RC4 algoritmusokat, a kulcsok kezelésére az RSA algoritmust használja.

Az RSA módszernél minden felhasználóhoz egy egyedi titkosító kulcspárt rendel: egy saját kulcsot, amely csak a felhasználói ID fájlba kerül be, és egy nyilvános kulcsot, amely a felhasználó ID fájlján kívül a közös címjegyzékbe is bekerül, ahol nyilvánosan elérhető.

Ha a felhasználók elfelejtik a jelszavukat vagy elvesztik az ID fájljukat, és nincs róla back-up másolatuk, az összes levél, amit a saját kulcsukkal titkosítottak, örökre elveszett.

Titkosítani lehet hálózati portokat, adatbázisokat, levélfájlokat, dokumentumokat és mezőket.

Aláíráskor a Notes elkészíti a dokumentum 128 bites „ujjlenyomatát", és ezt titkosítja a felhasználó privát kulcsával és hozzácsatolja a dokumentumhoz.

Egy aláírt dokumentum olvasása nincs korlátozva, de ha valaki módosítja a dokumentumot, akkor már nem fog egyezni a titkosított „ujjlenyomat" a dokumentummal, és a Notes nem fogadja el hitelesnek az aláírást.

Amikor egy felhasználó titkosított levelet küld, akkor a Notes generál egy véletlen kulcsot és ezzel titkosítja a levelet. Ezután a Notes a címzett publikus kulcsával titkosítja a véletlen kulcsot. Végül elküldi a titkosított kulcsot a titkosított levéllel együtt. A címzett a saját privát kulcsával tudja ezt a levelet elolvasni.

Adatbiztonság

Az adatok védelmére a Notes egy kombinált rejtjelező rendszert alkalmaz. A különösen érzékeny információkat, az elektronikus pecséteket és aláírásokat, valamint a használat közben generált egyszerűbb kulcsokat az RSA rejtjelező rendszer védi.

RSA alkalmazási területek a Lotus Notes-ban

- Levél és dokumentum kulcs titkosítás, 630 bit.
- Adatbázis kulcs titkosítás, 630 bit.
- Levél, dokumentum és mező digitális aláírás, 630 bit.
- Felhasználó azonosítás, 630 bit.

RC2 és RC4 titkosítás

A Lotus Notes az amerikai és kanadai felhasználók részére 64 bites kulccsal végzi az RC algoritmusú rejtjelzést, míg a nemzetközi felhasználók részére 40 bites rejtjelzést biztosít. Valójában ez utóbbi is 64 bites rejtjelzéssel történik, de a kulcs egy részét (20 bitet) az Amerikai Egyesült Államok importkorlátozó törvényének eleget téve az NSA rendelkezésére bocsájtja. Információink

szerint lehetőség van az erősebb amerikai változat beszerzésére külföldi ügyfeleknek is külön kérés és elbírálás alapján.

RC2 és RC4 felhasználási területek

- ID fájl titkosítása.
- Levél, dokumentum és mező titkosítás, RC2 (64 bit).
- Adatbázis titkosítás, RC2 (64 bit).
- Kapcsolat felépítés, RC2 (64 bit).
- Adatkapcsolati kulcs, RC2 (40 bit).
- Egyéb titkosítási kulcsok, RC2 (64 bit).

Az alábbiakban rövid áttekintést adunk a mobil kommunikáció védelméről:

7.4.3.b. A GSM-hálózatok alkalmazás-szintű biztonsági megoldásai

A GSM egyik kriptográfiai primitívjéről (A5/1 algoritmusról) már írtunk. Több más primitív (A3, A8) is szerepet játszik, de az egész rendszer sokkal komplexebb az algoritmusoknál. A biztonság elemzése is kitekintést igényel a teljes kriptográfiai alkalmazásrendszerre (a gyengeség is ott mutatkozik).

A GSM hálózatban az előfizetőt az úgynevezett IMSI-szám (International Mobile Subscriber Identity – nemzetközi mobilelőfizető-azonosító) azonosítja. A rendszer a telefonszám helyett ezt használja azonosításra a kommunikáció során. Az IMSI a telefonban levő smart kártyán – SIM előfizetői kártyán – tárolódik. Az IMSI-t a telefon bekapcsoláskor elküldi a központnak, s onnan egy TMSI-nek nevezett TMSI (Temporary MSI – ideiglenes MSI) számot kap vissza, s a továbbiakban ezt használják az előfizető azonosítására. Ezt rendszeresen cseréli a rendszer. A fentiek miatt az esetleges lehallgatónak a teljes folyamatot végig kell kísérnie az előfizető azonosításához.

Az azonosítás mellett híváskezdeményezéskor az előfizető hitelesíti magát, majd egy közösen kialakított egyszeri kulccsal védve rejtjelezik is az adatcsatornát, pontosabban annak a telefon és a bázisállomás közötti kapcsolatát. Mindennek kriptográfiai alapját a SIM kártyán, a kártya operációs rendszere által védve tárolt KI kulcs adja.

A hitelesítéshez a telefon a központtól kap egy 128 bites véletlen elemet. (RND) Ebből és a KI kulcsból a kártya az A3 algoritmus segítségével számol ki egy választ, s annak első 32 bitjét küldi vissza. A központ is ismeri a KI kulcsot, ezért ugyanazt a számítást elvégezve ellenőrizni tudja a kapott értéket.

Ugyanebből az RND számból és a KI kulcsból egy A8 nevű algoritmussal készít a kártya egy K 64 bites kulcsot. Sajnos annakidején COCOM előírások miatt ennek a kulcsnak tíz bitjét nullára állították be, s ez a kulcs ezért azóta is 54 bites. Ez a kulcs lesz a kommunikáció titkosításához felhasznált kulcs. A titkosítás csak a „levegőben" működik, kicsit egzaktabbul: a telefon és a

bázisállomás között. A kommunikációt az A5 algoritmussal titkosítják. Az A5 algoritmust a mobiltelefonban implementálják. Az A5-nek két verziója van, az egyiket a fejlett országokban – A5/1, ez az erősebb –, míg a másikat Ázsiában használják.

Az A5 algoritmussal szemben az A3 és A8 algoritmusok a kártyán kerülnek implementálásra. Az A3 és A8 algoritmusok a szolgáltatóhoz kapcsolódnak, azaz akár minden egyes szolgáltatónál más-más algoritmust használhatnak az A3 és az A8 helyén, ez a roamingot nem zavarja.

Az A3 és A8 szerepében sok szolgáltató használja a COMP128 algoritmust.

A GSM és a harmadik generációs UTSM rendszer, valamint a mobil telefonról történő internet elérést segítő WAP /Wireless Application Protocol/ szolgáltatás biztonsági architektúrájáról ajánljuk még a [2]-ben található áttekintést.

7.4.3.c. Wireless technológia

A számítástechnikai hálózatok vezeték nélküli kommunikációja sok felhasználási környezetben nagyon kényelmes, gyorsan terjed. Több más védelem mellett (SSID Bradcasting tiltása, hálózati azonosító /MAC cím/ szerinti szűrés) lényeges kriptográfiai elem a hálózati forgalom titkosítása.

A kezdeti rendszerekben használt titkosítás a WEP (Wired Equivalent Privacy) RC4 eljárást használ, legtöbb esetben 40 bites kulccsal. A kulcs és az ebből képzett inicializáló vektor segítségével kódolnak, de az IV inicializáló vektor csak 24 bites. A kulcs statikus, az IV változik, de rövid, ez az egyik alapja a WEP támadásának.[13]

A gyengeségek kiküszöbölésére fejlesztették ki a WPA (WI_FI Protected Access) eljárást, mely változó kulcs, megnövelt inicializáló vektor mellett több más védelmi elemet is tartalmaz.

A legújabb továbbfejlesztés a WPA2. Ez AES algoritmust használ kódolásra, nagyobb kulcsméretekkel.

7.5. Kriptográfiai alkalmazásokat futtató informatikai rendszerek védelmi alrendszerei

Természetesen a komplex biztonsági (kriptográfiai megoldások hatalmas együttesét) tartalmazó rendszerek (pl. LotusNotes, PGP, elektronikus aláírási rendszer ...) futtatása is általában informatikai rendszerkörnyezetben történik.

A teljes kriptográfiai rendszer biztonsága függhet az általános informatikai rendszer biztonságától (az operációs rendszerek, adatbázis-kezelők, levelező rendszerek, internet és egyéb informatikai szolgáltatók gyenge biztonsága veszélyeztetheti a jól megalapozott védelmet is, pl., ha a gépemben már egy feltételezett támadó az én jogosultságaimmal kezelheti kriptográfiai primitíveimet, kulcsaimat, a sémákat, protokollokat, kriptográfiai alkalmazásokat).

[13] A WEP támadásáról, valamint Budapesten használt rendszerek felméréséről olvashatunk a `www.saveas.hu` honlapon cikkeket.

Ez a terület nagyon szerteágazó: a technológiai (operációs rendszerek, adatbázis-kezelő rendszerek, alkalmazások) biztonsági kérdéseken túl a szervezeti, humán biztonsági kérdések széles skálája is beletartozik.

8. A kriptográfiai primitívek különböző biztonsági megközelítései

A kriptográfiai primitívek biztonságát többféleképpen csoportosíthatjuk:

- *Bizonyítottan nem biztonságos rendszerek.* Ennek két osztályát lehet megkülönböztetni:
  - valamilyen reális feltétel mellett – feltörhető a rejtjelrendszer /ld. pl. David Kahn: The Codebreakers, New York, 1996 c. könyvét, vagy a DES-re vonatkozó támadások leírásait/;
  - publikált támadási technikák vannak, amelyekkel a kulcstér teljes kipróbálásánál hatékonyabban támadható a rendszer, ugyanakkor a módszer használata a legtöbb esetben reálisan nem kivitelezhető (pl. összetartozó 2^{47} db nyílt-rejtjeles blokkpár ismerte a DES kulcs meghatározásához differenciál kripto-analízis módszerével, mely messze van az átlagos eszközökkel kivitelezhető támadástól, de szakmailag a rendszer gyengeségére utal).

- Eddig még nem törték fel az adott (paraméterű) rejtjelrendszert, de *nem kizárható az elmélet és a támadási technikák olyan fejlődése, mellyel az a gyakorlatban is kivitelezhetővé válik.*

  Ezen biztonsági osztály között megkülönböztethetőek az alábbiak:
  - Titkos, vagy nyilvánosságra került, de nem kellően bevizsgált rendszerek.
  - A kriptográfiai szakmai tudományos közösség által különböző támadási technikákkal alaposan vizsgált rendszerek, melyek eddig ellenálltak az ismert technikáknak (pl. TripleDES, IDEA, AES). Ennél az osztálynál érdemes megjegyezni, hogy mit értünk azon a kifejezésen, hogy *egy algoritmus ellenáll az adott támadási technikának.* Például az AES algoritmus is támadható a differenciál kripto-analízis (vagy lineáris kripto-analízis) módszerével, de a támadás igényelt lépésszáma lényegesen nem csökkenti a teljes kulcskipróbáláshoz szükséges lépésszámot. Ekkor azt mondjuk, hogy az algoritmus ellenáll ennek a támadásnak. (Nem hatékonyabb ez a támadás, mintha kipróbálnánk az összes kulcsot. Ez a kijelentés például nem igaz a DES algoritmusra.)
  - Klasszikus, ismert matematikai problémákra visszavezethető kriptográfiai rendszerek, mely problémákra mind a kriptográfiai, mind a matematikai szakmai tudományos közösség még nem talált hatékony megoldó

algoritmusokat (ezek külön osztályát alkotják azok az algoritmusok, melyekre visszavezetve a kriptográfiai rendszert, annak feltörése egyszerre az összes nehéznek tartott (NPC) matematikai probléma megoldását is eredményezné). Ez esetben a kriptográfiai paraméterek méretétől függ a rendszer feltörhetősége. Megadható olyan mérete a paramétereknek, melyek mellett a fenti értelemben biztonságosnak tekinthetőek az algoritmusok (pl. RSA rendszer).

- *Bizonyítottan nem lehetséges a rendszer hatékony – algoritmikai – támadása*, azaz nem várható olyan tudományos eredmény és számítástechnikai fejlődés, melyek lehetővé tennék a támadást (pl. OTP).

8.1. Információelméleti megközelítések, bizonyított biztonság

Shannon adta eddig az egyetlen teljes, információelméleti megközelítésen alapuló bizonyítási módszert az ún. One-Time-Pad /OTP/ rejtjelzésre, mellyel bizonyította, hogy egy esetleges támadó korlátlan erőforrásai mellett sem képes visszaállítani a titkosított üzenetet.

Elméletileg fejthetetlen algoritmus *(tökéletes rejtjelező)*, amelyre teljesül, hogy a kódolt üzenet (y) és az eredeti üzenet (x) közötti kölcsönös információ 0, vagyis $I(x, y) = 0$.

Shannon a fentieknek megfelelő (OTP) kódolás alatt egy valódi véletlen kulcssorozat moduló (jelkészlet) additív felhasználását értette. A One-Time Pad kifejezés onnan származik, hogy régebben a véletlen sorozatokat noteszlapokra írták le („pad"), és minden lapot (kulcselemet) csak egyszer lehetett felhasználni („one-time").

Kiemelt jelentősége van ennek az eredménynek az alábbiak miatt

- A bizonyítási módszer lehetővé teszi az algoritmus fejthetetlenségére vonatkozó állítás feltételrendszerének pontos meghatározását:
  - A kulcssorozat egyenletes eloszlású valódi véletlen, legalább a rejtjelzendő üzenet hosszúságú, karakterkészletével megegyező karakterkészletű (ha pl. a titkosítandó karaktereket bitsorozattá konvertálják titkosítás előtt, akkor a kulcssorozat is bit-sorozat, az additív művelet a mod2-es összeadás; ez a leggyakoribb felhasználási mód).
  - A támadó semmilyen információval nem rendelkezhet a kulcssorozatról (pl. „side-information" nem állhat rendelkezésére, például a rejtjelzés folyamatából).
  - A titkos kulcsok nem juthatnak egy támadó birtokába (őrizni kell).
  - Minden titkos kulcselemet csak egyetlen egyszer szabad felhasználni titkosításra.
- A bizonyítási módszer lehetővé teszi annak megfogalmazását, hogy mit is jelent a támadó bizonyított sikertelensége:

- A támadó az elfogott titkosított üzenetből – a titkosított nyílt üzenet hosszán kívül – semmilyen következtetést nem tud levonni arra vonatkozóan, hogy melyik üzenet a rejtjelszöveg ősképe (bármelyik nyílt szöveg, vagy a szöveg részlete egyenlő valószínűségű).
- Vonatkozik a fejthetetlenség bármilyen inputból kiinduló támadási módszerre /Ciphertext only attack, plaintext and corresponding ciphertext attack, Selected ciphertext and corresponding plaintext attack, Adaptive chosen-ciphertext attack, Selected plaintext and corresponding ciphertext attack/ és bármely algoritmikus módszerre /Brute force attack, Differential criptanalysis attack, Linear criptanalysis attack, .../, abban az értelemben, hogy feltételezett nyílt szöveget /a hosszán kívül/ sem megerősíteni, sem további nyílt szövegekre következtetni nem lehet. De nem feltétlenül vonatkozik a titkosítás műveleteinek kimérését megvalósító támadásokra /Power analysis attack, .../, melyek az előző pontban említett feltételek között kell kizárni.

- Kiemelkedő gyakorlati alkalmazásokban használják, ahol az alkalmazás egyrészt megköveteli a legerősebb titkosítás használatát, másrészt az alkalmazási nehézségek (hatalmas mennyiségű véletlen kulcs gyártása, szétosztása, védett tárolása és felhasználása) elfogadhatóak.

- Az OTP titkosítás gyakorlati problémáinak kezelésére próbálkoztak „pszeudovéletlen" sorozatok felhasználásával. Erre természetesen nem igaz az információelméleti biztonság bizonyítása. Azonban ha egyéb módszerekkel valószínűsíthető, hogy az esetleges támadó nem tudja a generált sorozatot megkülönböztetni a valódi véletlen sorozattól /distinguish attack), akkor már igaz lenne, hogy a nyílt szöveget sem tudja visszaállítani. Ekkor a biztonság mértékét a megkülönböztethetőség garanciális szintje adja, mely nem információelméleti, hanem a következő pontokban tárgyalt redukciós megközelítésen alapul.

- Ez az egyetlen hagyományos titkosítási eljárás, melyre teljesül, hogy ha a támadó eszköztárába kerülnek hatékony kvantumszámítógépek, akkor is ellenáll, a fent részletezett „fejthetetlenségi" fogalomnak megfelelően.

8.2. Redukciós-bonyolultságelméleti biztonság

A kriptográfiai primitívekre számos algoritmust, algoritmusok együttesét alkalmazzák, melyek egy részének biztonsága (és tervezési elve) közös matematikai problémákra vezethető vissza.

A tervezés során biztonsági szempontból megkülönböztethetők

- Intuitív módszerek (legyen minél „bonyolultabb");
- jól struktúrált, klasszikus matematikai problémákra visszavezethető algoritmusok.

Az első típusra példa a DES (Data Enctrition Standard) blokkrejtjelző, ahol utólag dolgoztak ki tervezési elveket (pl. a véletlen permutációk, S-dobozok tervezésére, ezekről részletesen olvashatunk [2]-ben).

A második típusra említhetjük az LFSR-ekből felépített rendszereket, de ez az elv leginkább a nyilvános kulcsú rendszerekre jellemző, a felhasznált klasszikus matematikai problémák, melyekkel próbálják visszavezetni a kriptográfiai biztonságot:

- **IFP** (Integer factoring problem);
  - **RSA_IFP:** A probléma külön osztályaként kezelik az RSA struktúrán alapuló problémát, mely két, ismeretlen nagy prím szorzatának faktorizációját jelenti;
- **SQROOT** (Square roots modulo n): adott egy n összetett szám, meghatározandó x:

$$x^2 \equiv a \pmod{n}$$

- **DLP** (Discret logarithm problem):

Adott egy p prím és α generátor eleme Z_p^* -nak, valamint $\beta \in Z_p^*$, meghatározandó x, $0 \leq x \leq p-2$, melyre $\alpha^x \equiv \beta \pmod{p}$

- **GDLP** (Generalzized discrete logarithm problem): olyan DLP, melynél a mod p osztály helyett egy ciklikus csoportból indulunk ki: Adott egy G véges ciklikus csoport, rendje: n, és α generátor eleme G-nek, valamint $\beta \in G$, meghatározandó x, $0 \leq x \leq n-1$, melyre $\alpha^x \equiv \beta$
  - **DHP, GDHP:** A fenti problémák külön osztályaként kezelik a Diffie–Hellman protokollon alapuló problémákat:
- **SSP, (KSP):** Subset-sum (Knapksacks) problem: adottak $\{a_1, a_2, \ldots, a_n\}$ súlyok, S pozitív egész, meghatározandó (ha létezik) olyan részhalmaza a súlyoknak, melyek összege: S.

(A felosztást részletesebben ld.: [5] 3. Number-Theoretic Reference Problem c. fejezetében.)

A fenti problémáknál a kriptográfiai algoritmust egy nehéznek gondolt feladatra vezetik vissza. A visszavezetés alatt azt kell érteni, hogy olyan lépések sorozatát kell igazolni, hogy amennyiben a titkosító algoritmust valaki fel tudná törni, a leírt lépésekkel a nehéznek vélt matematikai probléma megoldását is megkapná. Olyan nehéz feladatot szoktak választani, amelyre nem ismert hatékony – azaz polinomiális időben végrehajtható – megoldás. Ebből következik, hogy kellő méretű (kulcsterű) probléma mellett gyakorlatilag kivitelezhetetlen lesz a támadás.

Polinom idejű algoritmus (a bemenő paraméterek méretének függvényében):

> Létezik olyan c pozitív egész szám, hogy a legfeljebb n bites egészekkel végzett bitműveletek száma $O(n^c)$, $f(n)$ és $g(n)$ két pozitív értékeket felvevő $fv.$, $f = O(g)$, ha létezik olyan k konstans, hogy elegendően nagy n-re $f(n) \leq k^* g(n)$.

A kutatások az ún. egyirányú függvényekből indultak ki:

Az $f\{0,1\}^* \to \{0,1\}^*$ *fv.* egyirányú, ha a következő két feltétel teljesül:

1. Könnyű kiszámítani: létezik olyan hatékonyan (polinomiális időben) kiszámítható A algoritmus, mely x bemenetre az $f(x)$ kimenetet adja $(A : x \to f(x))$.
2. Nehéz invertálni: tetszőleges hatékony A' algoritmus $(A' : f(x) \to x)$, és tetszőleges $p(n)$ polinom, valamint elegendően nagy n esetén, véletlenszerűen választott x ősképek esetén:

$$Pr\Big\{f\big(A'[f(x), 1^n]\big) = f(x)\Big\} < 1/p(n).$$

Az egyirányú függvények egy osztályát alkotják a „csapda egyirányú függvények", melyeknél valamilyen ún. csapda információ birtokában az invertálás mégis könnyű (polinomiális időben végrehajtható). Ez a csapda tulajdonság szükséges ahhoz, hogy az üzenet címzettje (egy csapda-információ birtokában) a titkosított üzenetből visszakaphassa az eredeti üzenetet.

Tipikus példája ennek az RSA eljárás, ahol a csapda információ a két prímszám, melynek szorzata adja a nyilvános modulust. Amennyiben egy támadó nem ismeri a két prímszámot (csapdát), akkor az üzenet visszafejtése neki „nehéz" probléma, míg a címzettnek könnyű (a modulus méretében polinomiális (köbös) időben végrehajtható).

A nehéz (NP: non-polinomial) problémák külön osztályát alkotják az ún. NP-teljes problémák, amelyek közé azon problémák tartoznak, amelyekről bizonyították, hogy ha megoldhatók lennének polinomiális időben, az összes nehéz probléma megoldható lenne.

A faktorizáció problémája (IFP, RSA_IFP) bár nem NP-teljes probléma, mégis oly sokan vizsgálták évezredek óta, hogy a nehézsége elfogadott.

Még jobbnak tűnne NP-teljes problémát választani. Sikerült is ehhez az osztályhoz tartozó „csapda" konstrukciót választani, mely az SSP problémához csatlakozó **KSP:** Knapksack titkosító eljárás. (Ebben a csapda információt egy szupernövekedő sorozat biztosítja):

Legyenek $a_1, a_2, \ldots, a_n$ „súlyok"

$\underline{x_i} \in \{0,1\}^n$ bitvektor

$$S = \sum_1^n x_i a_i$$

Ekkor S, a_i-k ismeretében „nehéz" x_i-ket meghatározni, ugyanis a

KnapSack (hátizsák) probléma NP-teljes, ha a_i-k véletlenszerűek.

Viszont könnyű a meghatározás, ha a_i-k „szupernövekedő" sorozatot alkotnak (pl. kettes számrendszer).

Az ennek megfelelő kriptográfiai primitív:

Legyen $c_1, c_2, \ldots, c_n$ szupernövekedő sorozat:
és legyen $M > \Sigma c_i$, $(M, U) = 1$ (csapda-paraméterek).
Legyen $a_i \equiv U\ {}^*c_i \pmod{M}$, a_i-**k a nyilvános kulcsok.**

Titkosítás: $x = (x_1, x_2, \ldots x_n)$ az üzenet, a rejtjelzés: $C = \sum_{k=1}^{n} x_k\ {}^*a_k$.

Megoldás (a C rejtjeles sorozatból): Legyen $W : W\ {}^*U \equiv 1 \pmod{M}$-et kielégítő konstans, és

$$C\ {}^*W = \sum_{k=1}^{n} x_k\ {}^*(a_k\ {}^*W) = \sum_{k=1}^{n} x_k\ {}^*c_k,$$

ahonnan az x meghatározható.

Ez az eljárás mutat rá az NP-elméleti megközelítés biztonsági kockázataira, mivel a fenti problémát – az elméleti megközelítés ellenére – lehet támadni. Ennek okai a következőek:

- Az NP-elmélet „csak" a legrosszabb eset megoldásával foglalkozik, ugyanakkor az adatvédelemben nyilvánvalóan nem elegendő, ha sok titkosított üzenetből a támadó néhányat (a számára legrosszabb eseteket) nem tudja visszaállítani;
- Adatvédelemben nem megfelelő, ha közelítő megoldásokat, vagy a megoldás szempontjából ekvivalens megoldások közül egyet meg lehet hatékonyan – polinomiális időben – keresni.

A fenti problémák miatt titkosításra a hátizsák probléma nem elégséges biztonságú. A gyengeség (támadás) részletesen olvasható pl. [6] c. könyvben (itt a módszer javítási kísérleteinek támadásairól is olvashatunk). A kritikai megjegyzések nem az egész NP elmélet használhatóságára, hanem konkrétan a hátizsák algoritmusra vonatkoznak, ugyanakkor rámutatnak az alkalmazással kapcsolatos óvatos (átgondolt) megközelítés fontosságára.

Az NP elmélet kriptográfiai felhasználásáról, a bizonyítási technikákról részletesen olvashatunk [2] könyv IV. részében.

Itt olvashatunk pl. részletes meghatározásokat, eredményeket az alábbi témákról:

- a keménybit fogalma (mely a részleges kulcsfejtés kizárásához szükséges);
- a véletlen algoritmikus megkülönböztetetőség vizsgálata (szükségességét ld. a OTP-nél kifejtett meggondolásnál);
- az ún. szemantikai biztonság;
- üzenet-megkülönböztethetetlenség biztonság;
- rejtjeles szöveg módosíthatatlanság biztonság;
- a biztonságos kriptográfiai függvény-tervezés elvei; ...

fogalmakról, ezekkel kapcsolatos eredményekről.

8.3. Kalkulációs biztonság

A kalkulációs biztonság nem aszimptotikus megközelítését adja a biztonságnak, hanem, hogy ismereteink (gyakorlati vizsgálataink, jobb esetben bizonyítási módszereink) szerint nincs olyan algoritmus és hozzá rendelhető reális (létező eszközzel a futási idő életünkben bevárható) számítási kapacitás, amellyel adott támadás sikerrel elvégezhető lenne.

Ilyen értelmű kalkulációs biztonságot nyújt például az AES algoritmus az ismert támadásokkal szemben (de csak ezekkel szemben!), amelynél bizonyították, hogy pl. a differenciál kripto-analízis, illetve a lineáris kripto-analízis nem lényegesen hatékonyabb a teljes kipróbálás módszerénél, melynél például 128 bites kulcsot feltételezve (ez a minimális használható kulcsméret) a szükséges fejtési lépésszám: c^*2^{128}, ahol $c > 1$ egy kulcsra végzett AES művelet.

Az ismert támadási algoritmusok élesítésével konkrét becsléseket keresünk az algoritmusok szükséges lépésszámára. Így kapjuk, hogy például ez a biztonságfogalom nem teljesül a DES algoritmusra, sőt adott számítási kapacitásokat (memória-műveleti igény párt tekintve) a kétszeresen használt DES-re sem.

A kalkulációs biztonság elemzéséhez két tényező vizsgálata szükséges:

- Strukturális vizsgálatok (a teljes kipróbálás műveletigényét lényegesen lecsökkentő eljárások kutatása);
- Ezek műveletszámának (vagy a teljes kipróbálás műveletigényének) összevetése a számítástechnikai lehetőségekkel.

A fentiekre részletes elemzéseket mutatunk a 8.3.3. pontban az RSA kalkulációs biztonságának áttekintésekor, míg a DES-re vonatkozó megállapítások a második szempont szerint is kijelenthetők (a kétszeres DES nem megfelelő biztonságához szükséges kripto-analitikai megállapításokat is tenni).

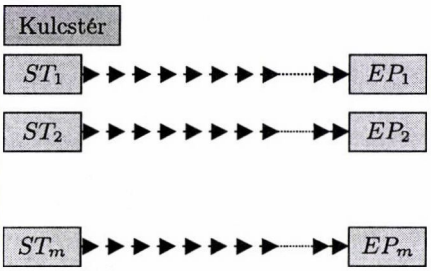
Az alábbiakban példaként néhány olyan eljárást szemléltetünk, amelyek nem felelnek meg a kalkulációs biztonság követelményének, azaz annak, hogy az ismert összes algoritmusra a reálisan létező számítástechnikai kapacitásokkal ne lehessen belátható idő alatt a rejtjelrendszert feltörni.

8.3.1. A DES és a kalkulációs biztonság

A DES-sel kapcsolatban már publikálása után nem sokkal Hellman egy speciális kimerítő támadás lehetőségét vázolta fel (akkoriban olyan technikai eszközparkkal, amelynek költségét akkor 20 millió dollárra becsülte), de kellő (hosszadalmas) prekondicionálási fázis után bármilyen, a feltételeknek eleget tévő rejtjelzett üzenetet hatékonyan – hamar vissza tudott állítani. (Támadási módszere a choosen plaintext attack elvéből indult ki.) Egy blokkra kellett ismerni az összetartozó nyílt-rejtjeles párt (azaz 8 byte-ot), és eszközével az összes olyan kódolt üzenet gyors fejtését lehetővé tette, amelyben az adott nyílt blokk szerepel, és tudjuk (vagy kipróbáljuk), hogy melyik a neki megfelelő rejtjeles blokk.

Hellman módszerének továbbfejlesztése – az azóta nagyon sok rejtjelrendszerre élesített – ún. ***Time-Memory Trade-Off Attack*** (azaz a rendelkezésre álló memória és a használható idő olyan megosztása, mely mellett egy prekondicionálási

szakaszban a kulcstér jelentős részét kipróbálva, ügyesen az eredményeket letárolva, a gyakorlati fejtés gyorsan végrehajtható.



A módszer lényege:
A kulcstérben véletlenszerűen választva „m" db. belépési pontot (ST), „ügyesen" megtéve „t" lépést, le kell tárolni a belépési – (ST) és végpontokat (EP), a végpontok (EP) szerint rendezve az m^*t^*2 (blokkméretnyi) tárban.
Ismeretlen kulcsú rejtjelzés kulcsát (mely valamelyik soron van) átlag $c^*t/2$ rejtjelző, majd $\log_2$m kereső és újból $c^*t/2$ rejtjelző lépésben megtalálhatjuk P valószínűséggel. (P függ a (t, m) pártól, ajánlott választás a $t^*m^2 = N^2$, ahol N a kulcstér mérete).

Például megemlítünk két ezzel foglalkozó előadást:

Philippe Oechslin: *Making a Faster Cryptanalytic Time-Memory Trade-Off,* **CRYPTO 2003 konferencián elhangzott előadásában** a DES törésre 1980-ban Martin Hellman által leírt (ismert P_0, C) párból induló, a kulcstéren lépegető, letároló eljárását gyorsította meg.

A cikkben példát mutat az MS Windows hash jelszótörésére 140 GByte tárhely felhasználásával, az összes alfanumerikus lehetséges hash értékre néhány másodperc alatt.

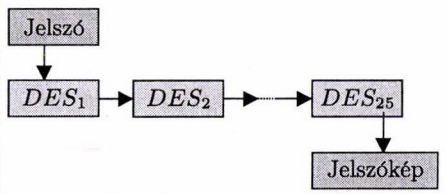
Alex Biryukov and Adi Shamir **Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers** c. ASIACRYPT 2000 konferencián elhangzott előadásában stream-cipher-ekre terjesztik ki ezt a hatékony támadási módszert. Eredménye szerint, ha a támadónak elégséges „d" számú output adat kipróbálása a döntéshez, akkor a $t^*m^{2*}d^2 = N^2$ paraméterválasztás mellett működik a támadás, ahol $d^2 \leq t \leq N$.

Külön említésre érdemes az EFF csoport DES-fejtő projektje (ld: `http://www.eff.org/descracker/`.) Az Electronic Frontier Foundation 1997-ben kezdte a projektet. Speciális FPGA (Field Programmable Gate Array) alapon épített rendszer keresi a DES kulcsokat. Ebben a szükséges hardver modulokról és az eredményes futtatás várható idejéről a következő táblázat szerepel:

Egység	Egységek száma a következő egységben	Másodpercenként kipróbálható kulcsok m száma	Az átlagos keresési idő napokban
Alapegység	24	2,500,000	166,800
Chip	64	60,000,000	6,950
lap	12	3,840,000,000	109
doboz	2	46,080,000,000	9,05
EFF DES Cracker		92,160,000,000	4,524

Azaz a speciális fejlesztésű célhardver minden chipje 24 kereső egységet tartalmaz. 64 ilyen chipet helyeznek el 12 panelen, melyből kettővel számolva másodpercenként 92,2 millió kulcs kipróbálására nyílik lehetősége, így az átlagos megoldási idő 4,5 nap lesz.

A DES támadását olyan szintre emelték, hogy több DES alapú más alkalmazás támadásánál sikerrel jártak. Példa erre a UNIX által használt password egyirányú képének képzése.
Az algoritmust a jobboldali ábra szemlélteti:



Pl. A SHARCS /Special Purpose Hardware for Attacking Cryptographic Systems konferencián 2005 februárjában Párizsban N. Mentes, L. Batina, B. Prencel, I. Verbauewhede: Cracking Unix Password using FPGA Platforms c. előadásukban hatékony célhardvert mutattak be a 25-szörös DES jelszókép ősképének visszaállítására.

A fentiek alapján megállapítható, hogy – bár egy átlagos felhasználó nem rendelkezik elegendő kapacitásokkal, azonban – a *DES nem felel meg a kalkulációs biztonság követelményeinek.* Ugyancsak nem megfelelő a kétszeres DES használat annak ellenére sem, hogy a felhasznált 112 bites kulcshossz önmagában elegendő lenne a kalkulációs biztonság kimondásához (nincs jelenleg elegendő számítástechnikai kapacitás 2^{112} eset végigfuttatásához), azonban a „találkozunk középen módszerrel, két összetartozó nyílt-rejtjeles blokkpár ismeretében a rendszer támadható:

A rejtjelzés képlete:

$$C = E_{K2}\big(E_{K1}(M)\big),$$

ezért

$$D_{K2}(C) = E_{K1}(M).$$

Ismert (M, C) párra kipróbálva – és letárolva – az összes $K1$ értékre az $E_{K1}(M)$ blokkokat, majd egyezést keresve az összes $K2$ kulcsra számolt $D_{K2}(C)$ blokkokkal,

leszűkíthető a $(K1, K2)$ esetszám: $2^{112}/\ 2^{64} = 2^{48}$-ra, amelyből a második összetartozó (M_2, C_2) blokkból nagy valószínűséggel kiválasztható az igazi $(K1, K2)$ kulcspár ($2^{48}/\ 2^{64} = 2^{-16}$ a tévedés valószínűsége).

Természetesen az eljárás során nagy számú technikai (pl. tárolási: 2^{56}* *8 byte, rendezési*) problémával kell megküzdeni, melyhez a Time-Memory Trade-Off módszerhez hasonló technika nyújt segítséget, de a kétszeres DES használat sem felel meg a kalkulációs biztonság követelményének.

8.3.2. Klasszikus stream-cipherek és a kalkulációs biztonság

Szintén nem nyújt megfelelő kalkulációs biztonságot (messze a teljes kipróbálás műveletigénye alatt marad) egy másik klasszikus – több millió készülékben kódolásra használt algoritmus, a GSM mobil telefonban kódoló A5 algoritmus sem.

A sokáig titokban tartott algoritmus internetre kerülése után indult vizsgálatok jelentős támadási eredményeket mutattak. A sok publikáció közül kiemeljük az alábbiakat:

Jovan Dj. Golic: **Cryptanalysis of Alleged A5 Stream Cipher**, EUROCRYPT '97; Alex Biryukov, Adi Shamir, and David Wagner: **Real Time Cryptanalysis of A5/1 on a PC**, mely a Fast Software Encryption, 7th International Workshop, FSE 2000. konferencián hangzott el; Alex Biryukov and Adi Shamir **Time/Memory/Data Tradeoffs for Stream Ciphers** címmel az ASIACRYPT 2000 konferencián, továbbá

Elad Barkan, Eli Biham, and Nathan Keller: **Instant Ciphertext – Only Cryptanalysis of GSM Encrypted Communication** címmel a CRYPTO, 2000 konferencián számoltak be további eredményekről.

Ez utóbbi cikkben foglalkoztak az A5/2 algoritmus hatékony törésével, valamint az A5/1 algoritmusnak pusztán rejtjelszövegből való támadásával, cikkükben az alábbi táblázat szerepel az eredményekről:

Támadáshoz szükséges adat	Felvételi percben	Szükséges 200 Gbyte-os tárolókapacitású HDD-k száma:	Szükséges PC-k száma egy éves előszámítási idővel számolva:	A támadás „reális" időben elvégzéséhez szükséges PC-k száma
2^{12}	5	22	140	1
$2^{6,7}$	8	176	5000	1000
$2^{6,7}$	8	350	5000	200
2^{14}	20	3	35	1

Néhány további példa a gyakorlatban használt, de a kalkulációs biztonsági követelményeket nem teljesítő stream-cipher-es rendszerre. (LFSR-ekből építkező rendszerek; a Bluetooth; több protokollban, pl WEP protokollban használt RC4; a PKZIP algoritmusa; stb.)

Sokszor alkalmazott rendszer a lineáris visszacsatolású shift regiszterekből modulszerűen építkező rejtjelrendszer. Ennek kritikus elemeire mutatnak példát az alábbiak:

- Az EUROCRYPT 2001 konferencián Geffe generátorára Siegenthaler támadási módszerének továbbfejlesztését mutatták be: Anne Canteaut and Eric Filiol „**Ciphertext Only Reconstruction of Stream Ciphers Based on Combination Generators**" c. cikkükben.
- Jovan Dj. Golic and Guglielmo Morgari **„On the Resynchronization Attack"** Fast Software Encryption 2003-as konferencián elhangzott előadásukban a lineáris következő állapotú-függvénnyel és nemlineáris kivételi függvénnyel bíró stream-cipherek egy támadása van leírva.
- AZ EUROCRYPT 2002-es konferencián **„Linear Cryptanalysis of *Bluetooth* Stream Cipher"** címmel Jovan Dj. Golić, Vittorio Bagini, and Guglielmo Morgari tartott előadást.
- **„Statistical Analysis of the Alleged RC4 Keystream Generator"** címmel Scott R. Fluhrer and David A. McGrew tartottak előadást a EUROCRYPT 2001 konferencián.
- **„ZIP Attacks with Reduced Known Plaintext"** címmel Michael Stay tartott előadást a Fast Software Encryption, 2001-es konferencián a PKZIP töréséről.

A fenti példák azt mutatták, hogy a kalkulációs biztonság nagyon sok, széles körben alkalmazott eljárásra nem biztosított. Érdemes azonban megjegyezni, hogy egy „átlagos" felhasználó kockázati szintjének lehet, hogy megfelel valamelyik standard eljárás, csak nagyobb kockázati szint esetén törhető nagyon jelentős erőforrással. Erre különösen jó példákat láthatunk az RSA biztonságával foglalkozó következő fejezetben.

8.3.3. Az RSA és a kalkulációs biztonság

Az RSA kalkulációs biztonságának elemzéséhez az alábbi mindkét tényezőt vizsgálni kell:

- Strukturális támadási vizsgálatok, ezen belül:
  - faktorizáció nélküli támadási irányok;
  - faktorizációs eljárások.
- Ezek műveletszámának (vagy a teljes kipróbálás műveletigényének) összevetése a számítástechnikai lehetőségekkel.

8.3.3.a. Az RSA elleni – nem faktorizácón alapuló – támadási módszerek

Az RSA kezdete óta nagyon sok publikáció jelent meg különböző, a modulus faktorizációját nem igénylő támadási módszerekről (ezekről áttekintés olvasható pl. [6], vagy D. Boneh, **Twenty years of attack on the RSA cryptosystem:** [1]. `http://crypto.stanford.edu/~dabo/abstract/ RSAattack-survey.html`). A támadások kiinduló-pontja hol rosszul megválasztott paraméterek, hol rossz rendszerhasználat; a támadások egyes esetekben az $\langle N, e\rangle$ nyilvános kulcsból a d magán

(titkos) kulcs visszaállítását, egyes esetekben a rejtjelzett üzenetek visszaállítását célozzák meg.

Néhány ismert, speciális esetekben hatékony támadási módszer látható a következő dupla oldalakon.

8.3.3.b. Az RSA faktorizációs támadása

A faktorizációs támadás fejlődése a számítástechnikai fejlődés (gyorsabb processzorok, ezek hatalmas rendszereinek egy feladatra koncentrálása) eredményeképpen, valamint (nagyobb részben) az elméleti módszerek fejlődésének eredményeképpen álltak elő.

Számítástechnikai segítség a faktorizációban

A nagy számok faktorizációja ősidők óta foglalkoztatja a matematikával foglalkozókat. Az algoritmikus módszerek és technikai lehetőségek egyaránt hatalmasat fejlődtek az utóbbi időben.

1874-ben úgy gondolták (Jevons), hogy senki sem tudja majd faktorizálni a 10 decimális jegyű ($10D$) 8616460799 számot. Persze akkoriban nem láthatták, a számítástechnika majd milyen segítséget ad a faktorizációhoz, de ez is mutatja az ilyen „jóslások" kockázatát.

A faktorizációs rekordok (D: a decimális jegyek számában)[14]

1964	1974	1984	1994
$20D$	$45D$	$71D$	$129D$

Kezdetben csak egy-egy gépen folyt a faktorizálás, 1984-ben a faktorizációs feladatra lehetett fordítani az akkori legerősebb Cray X-MP gép 9.5 CPU óra idejét.

A $129D$ faktorizálásában 1600 számítógép dolgozott 8 hónapig. Bár ez utóbbi nagyon sok gépnek számít, mégis pl. a Silicon Graphics kb. 5.000 munkatársa 10.000 workstation erőforrás felett rendelkezik. Mindez 10^5 MY (MY (MIPSYear): másodpercenként millió műveletet végző gép műveletszáma 1 év alatt), azaz 10-szer nagyobb erőforrás, mint az RSA129 projekthez szükséges volt.

A faktorizáció biztonságát befolyásoló, prognosztizált lehetőségeket lehet becsülni a sok átlagos gép erőforrásának együtteséből, illetve céleszközök kapacitásából.

A jelenlegi számítástechnikai lehetőségeket (maximális kooperativitást feltételezve) becsülhetjük az internetre kötött gépek számából (10^9 körül lehet a gépek száma), illetve a „Moore törvény"-ből, mely szerint a számítástechnikai teljesítmény (megfogalmazása szerint az integrált áramköri lapkákon elhelyezett tranzisztorok száma) 18 hónaponként megduplázódik, nagyjából lehet egy becslést adni az általános célú számítógépek teljesítménynövekedésére.

[14] A táblázat nem teljesen precíz, mivel – mint később látni fogjuk – a rekordok függnek a faktorizálandó számok tulajdonságaitól is(pl. 1994-ben faktorizálták a $162D$-ből álló $(12^{151} - 1)/11$ számot); ezért nem folytattuk a táblázatot 2004-re.

Támadás jellemzője	Támadás bemenő adatai	Feltétel	Támadás eredménye	Megjegyzés
Az üzenet rejtjelzés után is olvasható marad	Rejtjeles	M fixpontja az RSA rendszernek	M olvasható	Blakley-Borosh bizonyítása szerint az RSA rendszernek $(e,(p-1)) * (e,(q-1))$ fix pontja van, mely nagy szám is lehet, speciális esetekben (de elenyésző számú, ha $(p-1)$-nek és $(q-1)$-nek van nagy prímosztója. $(1+(e-1,$ $p-1)) * (1+(e-1,q-1))$
Ciklikus Attack	$\langle N,e\rangle$ nyilvános kulcs, C rejtjeles		Az M üzenet: $C^e, C^{e^2}, C^{e^3}, \ldots$ sorozatban megtalálva a $C = C^{e^k}$ egyezést, $M = C^{e^{k-1}}$	A Simmons és Morris által vázolt támadás nem működik, ha $(p-1)$-nek és $(q-1)$-nek van nagy prímfaktora.
Magán kulcs dekonspirálódása	$\langle N,e\rangle$ nyilvános kulcs és a d magán kulcs		N faktorizációja	
Közösen használt magán kulcsok	Több résztvevő által használt közös modulusok: $\langle N_1,e_1\rangle$, $\langle N_2,e_2\rangle$		Az előző állítás szerint meghatározhatják egymás magán kulcsait.	Támadó a rendszer egyik résztvevője
Kicsi nyilvános exponens használata	$\langle N_i,e\rangle$; $i=1,2,\ldots,k$; $C_i = M^e \bmod (N_i)$	Kicsi közös exponenseket használnak; $k \geq e$; $M < \min(N_i)$	Az M üzenet meghatározható	„Hastad's Broadcast Attack" a tétel további erősítését adja
Kicsi nyilvános exponens használata kapcsolódó üzenetekre	$\langle N,3\rangle$; $C_1 = M_1^3$ $C_2 = M_2^3$; f fv.	$e=3$, $f=ax+b$, $M_2 = f(M_1)$	M_1, M_2 meghatározható	Franklin-Reiter Related Message Attack

Kétszer ugyanazt az üzenetet rejtjelzik különböző blokkfeltöltéssel	$\langle N, e\rangle$; $C_1 = M_1^e$, $C_2 = M_2^e$	N : n bit hosszú, M : $n-m$ bit hosszú üzenet, ahol $m = \lfloor n/e^2 \rfloor$; $M_1 = 2^m M + r_1$; $M_2 = 2^m M + r_2$ $r_1 \neq r_2$ $0 \leq r_1$, $r_2 < 2^m$	M meghatározható	Copperrsmith' Short Pad Attack. A támadás hatékony $e = 3$ esetben, ha a közös üzenet hossza a modulus hosszának 9-ed részénél kisebb. Nem hatékony a gyakran használt $e = 65537$ esetben.
Kicsi nyilvános exponens használata	$\langle N, e\rangle$	$N = p * q$, $q < p < 2q$, $d < \frac{1}{3}\sqrt[4]{N}$	d hatékonyan meghatározható	A bizonyítást kiterjesztették $d < N^{0,292}$-re, továbbá sejtés: $d < \sqrt{N}$ esetén is d meghatározható
Részleges titkos kulcs ismeret	$\langle N, e\rangle$; d legkisebb $\lceil n/4 \rceil$ bitje, ahol N n bit hosszú modulus		D meghatározható $e * \log_2 e$-ben lineáris időben.	A tétel általánosítható p $\lceil n/4 \rceil$ számú legkisebb, vagy legnagyobb helyértékű bitjeire is
Kikényszerített elektronikus aláírás	$\langle N, e\rangle$ nyilvános ellenőrző kulcsok? A támadó által összeállított M' üzenet S' aláírása $(S' = (M')^d$	$M' \equiv r^e * M$ mod N, ahol r véletlen szám.	Támadó alá tudja írni az M üzenetét.	
Protokoll elleni támadás	PKCS-1 (Public Key Cryptography Standard-1) blokk, melyet a támadó módosít, és vizsgálja, a címzett elfogadja-e a blokkot.	Vonali módosítás lehetősége, a válasz figyelése	PKCS-1 blokkfelépítése: „02-RANDOM-00-M" Ha a címzettnél nem OK a „02" adat (16 bit), akkor ismétlést kér. Ha a támadó többször módosítja a kódolt üzenetet $C' = r * C$-re, látja, mikor fogadja el a címzett ugyanazt az üzenetet, melyből decryptálni tudja C-t!	Bleichenbacher'Attack on PKCS-1, 1998

Az egy feladatra koncentrálható erőforrásokat lehet az általános célú számítástechnikai eszközök rendszeréből (sok ilyen gép együttes alkalmazásából) becsülni (mint fent, ún. GRID-es technológiával), illetve célgépek felhasználási lehetőségeivel. Erre a jelenlegi lehetőségeket jól mutatják a www.top500.org lapon olvasható erős gépek (vektorszámítógépek) adatai, melyek közül (2005 júliusi adat): pl. a BlueGene/L-eSrever 2005-ben installált gépben 65536 processzor található (440700 MHz-es órajellel /2,8 GFlops/), melynek átlagos teljesítménye 136800 GFlops (136800 milliárd lebegőpontos művelet végrehajtására képes másodpercenként).

A faktorizációs módszerek fejlődése

A faktorizációs algoritmusok hatékonysága függ az n faktorizálandó szám szerkezetétől, valamint méretétől.

- Ha az n szám nem túl nagy, akkor az n szám négyzetgyökéig történő próbálgatás egy lehetséges módszer, melynek műveletigénye: $O(p + \lg n)$, ahol p a második legnagyobb prímosztót jelöli (alulról eddig kell elmenni a próbálgatással). Ismeretes, hogy az $[1, x]$ intervallumban véletlenül választott n számra kb. 0,9 (0,1) a valószínűsége annak, hogy n-nek p_2 második legnagyobb prímfaktorra $p_2 \leq x^{0,359} (p_2 \leq x^{0,0056})$ legyen, mely nagy számokra nagy valószínűséggel hatalmas szám, ráadásul az RSA számokra garantáltan nagy a második prím is.
- Ha N-nek csupa kis prímosztója van (ekkor az egyik leghatékonyabb módszer az ún. „$p - 1$ method").
- Az ún. SNFS (Special Number Field Sieve) eljárás jelenleg a leghatékonyabb az $a^k \pm b$ típusú (pl. Fermat) számokra (például ezzel a módszerrel faktorizálták F_9-et, a 9. Fermat számot, és 1999-ben a $10^{211} - 1$-et).
- Külön figyelmet érdemelnek az RSA számok (melyeknek nincs kis prímosztójuk, két nagy prím szorzatából állnak).
- $120D$ alatt a QS: kavadratikus szita (vagy MPQS: Multiple Polynomial Quadratic Sieve), míg felette a GNSF: General Number Field Sieve) eljárások az ismert leghatékonyabb módszerek.

Az RSA rendszer megalkotása után nem sokkal nyilvánosságra hoztak ún. RSA számokat, melyek két nagy, titokban tartott prímszám szorzatai voltak (ezek faktorizációjára pénzdíjakat is kitűztek, de nagyobb érték volt a hírnév). Mindezt azért tették, hogy lehessen követni az új faktorizációs eljárásokból (illetve a technikai fejlődésből) adódóan az RSA paraméterekkel szemben milyen nagyságrendi követelményeket kell támasztani. Az alábbi táblázat ennek a faktorizációs versenynek néhány állomását mutatja (jelölve a faktorizációs algoritmust, és a szükséges számítási kapacitást MIPS-évben).

Az alábbi táblázatban MY: MIPS-Year-t (Millio Interception Per Year), QS: a Quadratikus szita faktorizációs módszerét, NFS: a Number Field Sieve módszerét jelöli.

Az RSA 155 (1999-ben történt) faktorizációját a EUROCRYPT 2000 konferencián mutatták be „Factorization of a 512-Bit RSA Modulus" címmel, szerzők:

RSA szám:	Decimálisan	Faktorizálás		Műveletigénye	Kitűzött díj
		Dátuma	Módszere		
RSA-100	100	1991.04.	MPQS	7 MY	
RSA-110	110	1992.04.	MPQS	75 MY	
RSA-120	120	1993.06.	MPQS	835 MY	
RSA-129	129	1994.04.	MPQS	5000 MY	
RSA-130	130	1996.04.	NFS	1000 MY	
RSA-140	140	1999.02.	NFS	2000 MY	
RSA-150	150	2004.04.	NFS		
RSA-155	155 (**512** bit)	1999.08.22.	NFS	8400 MY	
RSA-160	160	2003.04.01.	NFS		
RSA-576	174	2003.12.03.	NFS		10.000 $
RSA-200	200	2005.05.09.	NFS		
RSA-640	193				20.000 $
RSA-704	212				30.000 $
RSA-**768**	232				50.000 $
RSA-896	270				75.000 $
RSA-**1024**	309				100.000 $
RSA-1536	463				150.000 $
RSA-**2048**	617				200.000 $

Stefania Cavallar, Bruce Dodson, Arjen K. Lenstra, Walter Lioen, Peter L. Montgomery, Brian Murphy, Herman te Riele, Karen Aardal, Jeff Gilchrist, Gérard Guillerm, Paul Leyland, Joël Marchand, François Morain, Alec Muffett, Chris and Craig Putnam és Paul Zimmermann.

Ez a hosszú sorban azért bizonyult nagy áttörésnek, mivel az 512 bites modulust nagyon sok implementációban használt(j)ák. Szemléltetésül az eredmény:

RSA-155 =
109417386415705274218097073220403576120037329454492059909 13842131476349
984288934784717997257891267332497625752899781833797 0765372440271 467435
31593354333897
Szám faktorizációja, mely az alábbi két prímszám szorzata:
p = 102639592829741105772054196573991675900716567808038066803341933521790711307779
q = 106603488380168454820927220360012878679207958575989291522270608237193062808643

2005 közepén a csúcs a 2005. május 9-én bejelentett, RSA200 faktorizálása GNFS módszerrel, melynél a megfelelő egyenletek keresését (szita lépésről ld. később) 2003 karácsonya előtt kezdték és 2004 októberéig tartott (melyek összességében 55 év CPUidőt vettek volna igénybe egy átlagos 2,2 GHz-es gépen), míg a hatalmas mátrix megoldása 2004 decemberében vette kezdetét. Az eredmény két szám:

353246193440277012127260497819846436867119740019762502364930346877612125367942320005854795652808834 9
és
792586995447833303334708584148005968773797585736421996073433034145576787281815213538140930474018546 7

Faktorizáció Fermat ötletével

Az RSA modulus (két nagy prím szorzata) hatékony faktorizációja meglepően Fermat alapötletén alapul:

Amennyiben találunk olyan a, b számokat, melyekre $a^2 \equiv b^2 \bmod n$, és $a \neq +- b \bmod n$, akkor n osztható $a^2 - b^2 = (a-b) * (a+b)$ kifejezéssel, azaz $gcd(a-b, n)$ n-nek egy osztója.

Meglepő, hogy miért könnyebb találni adott tulajdonságú a, b számokat, melyek négyezetei mod$n$ megegyeznek, mint n-et faktorizálni.

Ehhez dolgozták ki a nagyon hatékonynak bizonyult ún. **kvadratikus szita** módszerét, melynek lényege:

Keresünk sok olyan $r^2 (> n)$ számot, melynek mod$n$ vett értékét tudjuk faktorizálni, és ezek prímfaktorai egy korlátos halmazból valók (viszonylag kicsik).

Sok ilyen

$$(*) \qquad r^2 \equiv y = \prod p_j^{\alpha_j} \bmod n$$

kongruenciából kiválasztjuk – lineáris egyenletrendszert megoldva – azokat, amelyeket összeszorozva a jobb oldalon minden (a korlát alatt lévő prím) páros hatványon szerepel ez adja b^2-et, míg a baloldalak (eleve négyzetszámok) szorzata adja a^2-et.

A fent keresett y számokra használják az ún. ***B-smooth*** fogalmát (B pozitív egész szám), mely szerint egy y szám B-smooth, ha minden prímtényezője kisebb B-nél.

A fenti szellemes eljárás gyengéje, hogy nagyon nagy n-ekre nagyon kicsi a valószínűsége, hogy y-nak csak B-nél kisebb prímosztói legyenek.

D. J. Bernstein: Bounding SmoothIntegers c. Extended Abstract-jában vizsgálta ezen esélyeket.

$\Psi(x, B) = \{\#m : 1 \leq m \leq x\}$ és m B-smooth jelölés mellett leírta $\Psi(x, B)$ nagyságrendjét. Például azt kapta, hogy $1{,}16 \cdot 10^{45} < \Psi(10^{54}, 10^6) < 1{,}19 \cdot 10^{45}$, azaz már $y = 10^{54}$ esetén is $\Psi(x, B)/y \sim 10^{-9}$ nagyon kis szám, sokkal nagyobb y-ra még sokkal kisebb az esély (hogy B-smooth legyen).

A módszert úgy kellett fejleszteni, hogy a (*) kongruencia jobb oldalán (y) nagyobb valószínűséggel legyen B-smooth, melyet úgy értek el, hogy megpróbálták y-t minél kisebb számra kihozni úgy, hogy nem tetszőleges r^2 alakú számként próbálgatunk, hanem olyat, ahol y „kicsi" lesz.

A fent leírt ötlet: *A kvadratikus szita (QS) módszere*

n gyökének alsó egész részéből $\left(m = \lfloor \sqrt{n} \rfloor\right)$ kiindulva számolja a $q(x) = (x+m)^2 - n = x^2 + 2mx + m^2 - n \approx x^2 + 2mx$ számot (mely kicsi n-hez ké-

pest, ha x kis szám); $x = \pm 1, \pm 2, \ldots, \pm S$, ahol S a „Sieve" intervallum paramétere. Mivel $q(x)$ kicsi n-hez képest (kis S-re), ezért nagyobb valószínűséggel ennek csak kis prímfaktorai vannak.

Választva egy B számot, tesztelik, hogy a $\left[q(x) \bmod n\right]$ számok B-smooth-ok-e, azaz ($\left[(x+m)^2 \bmod n\right]$ prím faktorai kisebbek B-nél) – különböző x értékekre (s-ig).

Keresve sok x_i értékhez B-smooth $\left[q(x_i) \bmod n\right]$ értékeket, ezekre

$$(x_i + m)^2 \equiv q(x_i) \mod n \quad (K1)$$

Mivel a $\left[q(x_i) \bmod n\right]$ számok B-smooth-ok, azaz csak „kis" prímfaktorai vannak, ezért olyan i indexű kongruenciákat vesznek, melyek szorzatában minden „kis" prímfaktor páros hatványon szerepel a fenti $K1$ kongruenciák jobb oldalainak összeszorzása után.

A $K1$ kongruenciák szorzatának bal oldalain eleve négyzetszámok szerepelnek, bal oldalon a fenti kiválasztás alapján kapunk négyzetszámokat, így kapják meg a kívánt tulajdonságokkal rendelkező a és b számokat, melyekből n faktorizálható.

(Gyorsítja az eljárást, hogy mivel egy $p(< B)$ prím ha osztja $q(x)$-et, akkor $(x+m)^2 \equiv n \bmod p$, ezért n kvadratikus maradék moduló p. Így csak azokkal a faktor-bázis elemekkel kell foglalkozni, amelyekre a Legrende szimbólum értéke 1.)

Példaként tekintsük az $n = 11111$ számot, legyen $S = 10$, $B = 15$, ezért $m = 105$, a prímek bázishalmaza $= \{-1, 2, 3, 5, 7, 11, 13\}$ (a -1 kiegészítés azért kell, mert $q(x_i)$ lehet negatív.

Ekkor a szita fázisban a következő megfelelő értékeket kapjuk:

$$\begin{aligned}
Q(-4) &= 101^2 - 11111 = -910 = -1 \cdot 2 \cdot 5 \cdot 7 \cdot 13 \\
Q(\ 1) &= 106^2 - 11111 = 125 = 5^3 \\
Q(\ 2) &= 107^2 - 11111 = 338 = 2 \cdot 13^2 \\
Q(\ 4) &= 109^2 - 11111 = 770 = 2 \cdot 5 \cdot 7 \cdot 11 \\
Q(\ 6) &= 111^2 - 11111 = 1210 = 2 \cdot 5 \cdot 11^2
\end{aligned}$$

Meg kell oldani az alábbi kongruenciát:
$= (-1 \cdot 2 \cdot 5 \cdot 7 \cdot 13)^x \cdot (5^3)^y \cdot (2 \cdot 13^2)^u \cdot (2 \cdot 5 \cdot 7 \cdot 11)^v \cdot (2 \cdot 5 \cdot 11^2)^w$
úgy, hogy minden prím páros hatványon szerepeljen:

$$\begin{aligned}
-1 &: \quad x \equiv 0 \pmod 2 \\
2 &: \quad x + u + v + w \equiv 0 \pmod 2 \\
5 &: x + 3 \cdot y + v + w \equiv 0 \pmod 2 \\
7 &: \quad x + v \equiv 0 \pmod 2 \\
11 &: \quad v + 2 \cdot w \equiv 0 \pmod 2 \\
13 &: \quad x + 2 \cdot u \equiv 0 \pmod 2
\end{aligned}$$

A fenti kongruencia-rendszer egy megoldása:

$x = 0$, $y = 1$, $u = 1$, $v = 0$, $w = 1$; ezért

$(106 \cdot 107 \cdot 111)^2 = (2 \cdot 5^2 \cdot 11 \cdot 13)^2$

ezért $a = 106 \cdot 107 \cdot 111$ és $b = 2 \cdot 5^2 \cdot 11 \cdot 13$ esetén

$a^2 \equiv b^2 \bmod n$, így $\text{lnko}\,(a - b, n) = 41$ n osztója.

A fenti kis mintapélda csak az elvet mutatja, lényegesen nagyobb számokra érzékeltetésül megadjuk az eredményes futtatáshoz várható szükséges paraméterek (S, B) méretét:

N jegyei decimálisan	Faktor bázis	Szita (Sieving) intervallum (S)
50	3.000	200.000
60	4.000	2.000.000
70	7.000	5.000.000
80	15.000	6.000.000
90	30.000	8.000.000
100	51.000	14.000.000
110	120.000	16.000.000
120	245.000	26.000.000

(A fenti táblázat Johannes A. Buchmann: Introduction to Cryptography, Springer, 1999 könyvében szerepel.)

A módszer használatában (a B és S paraméterek meghatározása után) két jelentős műveletigényű fázis van:

- találni $(x_i + m)^2 \equiv q(x_i) \bmod n = \prod p_i^{\alpha i} (p_{i<}B)$ kongruenciákat „Sieve Step";
- megoldani egy GF(2) feletti egyenletrendszert, annak meghatározására, hogy mely kongruenciákat kell összeszorozni ahhoz, hogy a baloldalon minden prím páros hatványon szerepeljen „Matrix Reduction Step".

Míg az első feladat jól párhuzamosítható (sok gépre szétoszthető a keresés), a második egy nagy teljesítményű gépet igényel.

Mind a Kvadratikus szita (**QS**), mind továbbfejlesztése, a 120 decimális számnál nagyobb számokra hatékonyabb Számelméleti szita (Number Field Sieve: **NFS**, Generalized Number Field Sieve: **GNFS**) módszerek a Fermat ötletnek megfelelő (mod n négyzetükben kongruens) a és b számokat keresnek. Mindegyiknél van sok gépre szétoszthető (párhuzamosítható) feladatrész: speciális kongruenciák keresése) és hatalmas mátrixot megoldó rész.

A módszerről *olvasható* például az alábbi hivatkozásokban:

A. K. Lenstra and H. W. Lenstra, Jr., *The Development of the Number Field Sieve.* Berlin: Springer-Verlag, 1993.

C. Pomerance, „A Tale of Two Sieves." *Not. Amer. Math. Soc.* **43**, 1473–1485, 1996.

Eric W. Weisstein, „Number Field Sieve." From MathWorld-A Wolfram Web Resource. http://mathworld.wolfram.com/NumberFieldSieve.html

A. K. Lenstra, Bellcore, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard, The number field sieve: http://www.std.org/~msm/common/nfspaper.pdf

D. Coppersmith, „Modifications to the Number Field Sieve." *J. Cryptology* **6**, 169–180, 1993.

J. Cowie, B. Dodson, R. M. Elkenbracht-Huizing, A. K. Lenstra, P. L. Montgomery and J. A. Zayer, „World Wide Number Field Sieve Factoring Record: On to 512 Bits." In *Advances in Cryptology-ASIACRYPT '96 (Kyongju)* (Ed. K. Kim and T. Matsumoto.) New York: Springer-Verlag, pp. 382–394, 1996.

Mint említettük az N összetett szám faktorizálására vannak speciális tulajdonságokkal rendelkező számokra hatékony és ilyen tulajdonságra nem kihegyezett módszerek.

Például ilyen, speciális tulajdonságú számokra hatékony módszerek:

- az osztásos próbálkozás;
- Pollard „rho" algoritmusa;
- Pollard „p-1" algoritmusa;
- ECM: Lenstra elliptikus görbéket felhasználó algoritmusa;
- Fermat faktorizációs módszere;
- a „Special Number Field Sieve" módszer.

A módszerek többsége akkor hatékony, ha N-nek sok kis prímosztója van, az utolsó akkor, ha $N = a^k \pm b$ alakú, ahol a, b kis számok (k lehet nagy).

Általános célú faktorizációs módszerek:

- Dixion algoritmusa;
- CFRAC: Continued Fraction Factorization;
- MPQS: Multiple Polynomial Quadratic Sieve;
- GNFS: General Numver Field Sieve.

Az N összetett szám faktorizációhoz szükséges műveletigény egyszerűbb leírásához használják a következő formulát (u, v pozitív egész számok, $N \geq e$):

$$L_N[u, v] = e^{v*(\ln N)^u*(\ln\ln N^{(1-u)})}.$$

A fenti képlet jól jellemzi a futási időt, mivel

- $L_N[0, v] = e^{v*(\ln N)^0*(\ln\ln N^1)} = (\ln N)^v$, amely polinomiális futási időt eredményez, mivel N bináris hossza $\lfloor \log_2 N \rfloor + 1$;
- $L_N[1, v] = e^{v*(\ln N)^1*(\ln\ln N^0)} = e^{v*\ln N}$, amely exponenciális futási időt mutat;
- $0 < u < 1$ esetén a futási idő ún. szubexponenciális.

A fenti jelölésekkel a szakirodalomban található levezetések alapján:
A CFRAC algoritmus műveletigénye: $L_N\left[1/2, c + O(1)\right]$, ahol $c = \sqrt{2} \approx 1{,}4142$.
A MPQS algoritmus műveletigénye: $L_N\left[1/2, 1 + O(1)\right]$.
A CFRAC algoritmus műveletigénye: $L_N\left[1/3, c + O(1)\right]$, ahol

$$\begin{array}{ll} \text{GNFS esetén} & c \approx 1{,}9229, \\ \text{SNFS esetén} & c \approx 1{,}5262. \end{array}$$

Képlettel felírva a szükséges műveletszámokat az alábbi táblázat ad egy összefoglalást, ahol N a faktorizálandó szám, melynek egy p ($\leq \sqrt{N}$) prímfaktorát kell megtalálni. A kétféle módszer, az elsőben a módszer műveletszáma függ p-től, a másodikban nem (pl. az RSA számokra a második esetet kell használni):

Módszer elnevezése	**Függ-e a meghatározandó prímfaktortól**	**Számításigény**
Prímosztásos próbálkozás	Függ	$O\left(p * (\ln N)^2\right)$
Pollard „rho" algoritmusa	Függ	$O\left(p^{1/2} * (\ln N)^2\right)$
ECM: Lenstra Elliptikus görbéken alapuló algoritmusa	Függ	$O\left(e^{\sqrt{c * \ln p * \ln \ln p}} (\ln N)^2\right)$, ahol $c \approx 2$
Lehman algoritmusa (legrosszabb esetre)	Nem függ	$O(N^{1/3})$
MPQS (Multiple Polynomial Quadratic Sieve)	Nem függ	$O\left(e^{\sqrt{c * \ln N * \ln \ln N}}\right)$, ahol $c \approx 1$
NFS (Number Field Sieve) SNFS: Special NFS: $N = a^k \pm b$, a, b kicsi GNFS: General Number Field Sieve	Nem függ	$O\left(e^c * (\ln N)^{1/3} * (\ln \ln N)^{2/3}\right)$, ahol $C = (32/9)^{1/3} \approx 1{,}5262$ $C = (64/9)^{1/3} \approx 1{,}93$

Megjegyzés: Ha $p \approx \sqrt{N}$, akkor az ECM és az MPQS műveletigénye aszimptotikusan közel megegyezik.

A fenti táblázatot Francois Morain: „Thirty Years of Integer Factorization", 2001. febr. 5. cikkéből, valamint Richard P. Brent „Some parallel Algorithms for Integer Factorization", LNCS Vol. 1685 (1999), 1–22 c. cikkéből állítottuk össze.

Az NFS módszer eredményességéhez szükséges faktor bázisok (prekondicionált prímek), a szita módszerhez felhasznált előkészítő kongruenciák, illetve a lineáris kongruencia-rendszer mátrixának méretére Shamir a következő becslést adta különböző méretű RSA számok esetére:

Key-size	Total Time	Factor Base	Sieve Memory	Matrix Memory
428	$5.5 * 10^{17}$	600 K	24 Mbytes	128 Mbytes
465	$2.5 * 10^{18}$	1.2 M	64 Mbytes	825 Mbytes
512	$1.7 * 10^{19} (\sim 2^{64})$	3 M	128 Mbytes	2 Gbytes
768	$1.1 * 10^{23}$	240 M	10 Gbytes	160 Gbytes
1024	$1.3 * 10^{26}$	7.5 G	256 Gbytes	10 Tbytes

Az RSA140-re és RSA155-re faktorizálásánál a mátrix megoldásához szükséges kapacitásadatokat mutatja az alábbi táblázat:

	MIPS igény	Sorok száma a mátrixban	Nem nulla elemek számának átlaga egy sorban	A mátrix megoldási ideje Cray C916-tal
RSA140	2000	$4{,}7 * 10^6$	32	100 óra
RSA155	8000	$6{,}7 * 10^6$	62	224 óra

Az utóbbi időben a kutatások fő iránya az 1024 bites RSA feltörésére alkalmas módszerek és célhardverek előállítása irányában felerősödtek.

Az ilyen kutatások sikerének kockázata, hogy a legtöbb RSA-t használó kriptográfiai rendszerben (pl. **HTTPS, SSH, IPSec, S/MIME, PGP**), de a legtöbb elektronikus aláírási rendszerek ma 1024 bites RSA-t használnak. Vagyis az esetleges sikernek közvetlen, a gyakorlatban használt módszerek biztonságát kompromittáló következménye lenne.

Az ASIACRYPT 2003 konferencián Arjen Lenstra, Eran Tromer, Adi Shamir, Wil Kortsmit, Bruce Dodson, James Hughes, and Paul Leyland adtak elő „Factoring Estimates for a 1024-Bit RSA Modulus" címmel.

A CRYPTO 2003 konferencián Adi Shamir and Eran Tromer adtak elő „Factoring Large Numbers with the TWIRL Device" címmel.

A módszer alapja az NFS (Number Sieve Field) eljárás, 1999-ben a szita része többszáz gépen több hónapig futott. Akkor úgy gondolták, hogy az 1024 bit még 15–20 évre biztos elegendő nagyságú lesz. A cikk bemutat egy 1 GHZ-es VLSI technológián alapuló hardware implementációt, mely 3–4 nagyságrenddel hatékonyabb az összes korábbinál (elnevezése TWINKLE), mely az 512 bites modulus szita lépését 10 perc alatt elvégzi, és az eszköz megvalósítható 10.000 USD-ből. Ugyanez 1024 bitre – egy év alatti szita-lépésre – 10 millió USD-ből jönne ki.

Nyilvánvaló, hogy a jelenlegi matematikai módszerekkel általános célú számítógépekkel az 1024 bites RSA nem törhető. Ezért a kutatások nagyon sok processzor, speciális vezérléssel szervezett célhardverek irányában is intenzíven folynak. Itt megemlítjük a SHARCS (Special PurposeHardware for Attacking Cryptographic Systems) 2005, Párizs konferencián elhangzott két előadást:

- Shamir és E. Tromer: Special-Purpose Hardware for Factoring: the step NFS Sieving Step, melyben a TWINKLE és a TWIRL célhardverek tervezéséről volt szó;
- J. Franke, T. Kleinjung, C. Paar, J. Pelzl, C. Priplata, C. Stahlke: Shark – A Realizable Special Hardware Sieving Device for Factoring 1024-bit integers.

Összefoglalva az RSA kalkulációs biztonságáról szóló részfejezetet elmondhatjuk, hogy rendszerszintű hibát nem elkövetve, az 1000 bit feletti modulusú RSA jelenleg általános célú felhasználásra biztonságosnak tekinthető.

A korábbi, prímekre vonatkozó kikötések (miszerint „$p-1$"-nek, „$q-1$"-nek is legyen nagy prímosztója, ma már nem követelmény. Ennek oka, hogy a korábbi elvárásokhoz képest jelentősen megnövelt modulus miatt, a kapcsolódó támadási módszerek csak nagyon kis valószínűséggel hajthatók végre. Ezt támasztják alá az első és második prímosztóra vonatkozó alábbi eredmények:

Az $[1,x]$ intervallumban véletlenszerűen választott N szám legnagyobb prímosztója: $p_1 \leq x^\alpha$ egy $F(\alpha)$ valószínűség-eloszláshoz tart, ahol

$$F(\alpha) = \int_0^\alpha F\left(\frac{t}{1-t}\right)\frac{dt}{t}, \quad \text{ha} \quad 0 \leq \alpha \leq 1; \quad \text{és} \quad F(\alpha) = 1, \quad \text{ha} \quad \alpha \geq 1.$$

Hasonló integrállal fejezhető ki a második legnagyobb prímosztó $G(\beta)$ eloszlása is (mely pl. a próbálgatásos faktorizáció műveletigényének becslésekor kerül elő). Példaként megadjuk ennek a két sűrűségbecslésnek néhány értékét, mely mutatja, hogy egy x-nél nem nagyobb véletlen szám első, ill. második legnagyobb prímosztója $F(\alpha)$, ill. $G(\beta)$ valószínűséggel lesz x^α, x^B-nél kisebb, ahol:

$F(\alpha), G(\beta)$:	0,9	0,8	0,5	0,35	0,2	0,1	0,01
1. legn. α:	0,9048	0,8187	0,6065	0,5220	0,4430	0,3785	0,2697
2. legn. β:	0,3590	0,3104	0,2117	0,1611	0,1003	0,0531	0,0056

azaz például ha $x = 2^{1024}$, akkor a legnagyobb, második legnagyobb prímosztó 50% valószínűséggel kisebb, mint $x^{0,6065} \approx 2^{621}$, ill. $x^{0,2117} \approx 2^{216}$.

Az EESSI-SG (European Electronic Signature Standardisation Initiative Steering Group) felügyelete alatt dolgozó Algoritmus csoport (ALGO) által meghatározott követelmények RSA algoritmus esetén:

- a modulus $n = pq$ bithossza (ModLen) legalább 1020 bit legyen,
- p és q megközelítőleg azonos hosszú: $0{,}5 < |\log_2 p - \log_2 q| < 30$,
- megfelelően nagyszámú prímszám közül függetlenül kell a két prímet választani, és ezek eloszlása megfelelően egyenletes kell legyen: a véletlen választás, ill. egy véletlen generátor induló értékének legalább 128 bit szabadsági fokúnak kell lennie,
- a prímtesztnél a hiba valószínűsége (tehát annak valószínűsége, hogy p, vagy q mégis összetett szám) kisebb mint 2^{-60} legyen.

8.3.4. Az elektronikus aláíráshoz szabványosított hash eljárások és a kalkulációs biztonság

A gyakorlatban alkalmazott nem megfelelő biztonsági szintű primitívek nem csak a titkos kulcsú rejtjelző primitívek között ismertek, hanem például a hash számoló primitívek között is:

- A Fast Software Encryption, 2003 konferencián „**Cryptanalysis of Block Ciphers Based on SHA-1 and MD5**" címmel Markku-Juhani O. Saarinen tartott előadást, melyben bevezetett „related key attack" módszerével ún. „slid-pairs"-t kerestek (találtak).
- Ugyanezen a konferencián „**New Attacks against Standardized MACs**" címmel Antoine Joux, Guillaume Poupard, and Jacques Stern előadásukban legfeljebb 2^{33} összetartozó „choosen messages"-re támadják a ISO/IEC 9797-1 szabvány szerinti DES-t használó MAC eljárást.
- A CRYPTO 2000 konferencián „**Key Recovery and Forgery Attacks on the MacDES MAC Algorithm**" c. előadásukban Don Coppersmith, Lars R. Knudsen, and Chris J. Mitchell foglakoztak a DES alapú MAC számítás problematikájával.

Az elektronikus aláírási rendszerekben – számítási kapacitás korlátok, valamint a dokumentumokhoz fűzött kiegészítő információk csökkentése okán – nem a teljes dokumentumot írják alá (aláírási primitívvel), hanem csak egy lenyomatát, amit a hash függvénnyel állítanak elő. (A bináris hash függvények: $M \to H(M)\{0,1\}^n \to \{0,1\}^m$ leképezést valósítanak meg egy tetszőleges n ($\geq n_0$) hosszúságú bináris sorozatot egy rögzített m ($\geq m_0$) hosszúságú sorozatra képezve le.) Ekkor – több más követelmény mellett – elengedhetetlen, hogy adott lenyomathoz ne lehessen másik olyan dokumentumot találni, amely azonos lenyomatot (így azonos aláírást) képez. Ilyen tulajdonságokkal már a hagyományos hibajavító kódolók nem rendelkeznek (mint alább látni fogjuk, nem is egyszerű megfelelő Hash függvényt konstruálni).

A szakirodalomban a követelmények többféle megfogalmazása szerepel (és néha nehéz jó magyar kifejezést találni ezekre), melyek három *tulajdonságra* vezethetők vissza:

- *őskép-ellenállóság* (preimage resistance): adott y értékhez „nehéz" találni olyan M üzenetet, amelyre $H(M) = y$;
- *második őskép-ellenállóság* (2-nd preimage resistance): adott M üzenethez (melyre „könnyű" kiszámolni $y = H(M)$-et), nehéz találni olyan M'-t, melyre $H(M) = H(M')$;
- *ütközés-ellenállóság* (collision resistance): ne lehessen két különböző M, M' üzenetet találni, melyekre $H(M) = H(M')$.

(Az egyes tulajdonságok közötti összefüggéseket részletesen vizsgálták. Például az ütközés-ellenállóság nem garantálja az őskép-ellenállóságot.)

A Hash függvényekre használt *követelményeket* a fenti tulajdonságok alapján két nagyobb csoportra szokták osztani:

- **OWHF** /One Way Hash Function/, mely az őskép-ellenállóság és a második őskép-ellenállóság tulajdonságának teljesítését jelenti; valamint
- **CRHF** /Collision Resistant Hash Function/ az ütközésmentesség követelményei, mely a második őskép-elleneállóság és az ütközés-ellenállóság tulajdonságának teljesítését jelenti.

Megjegyzések:

Egyes definíciók az OWHF-re a „weak one-way hash function" kifejezést, míg a CRHF-re a „weak one-way hash function" kifejezést használják.

A „könnyű" kiszámítani fogalom alatt a legtöbb szakirodalom a bemenő paraméterek (M mérete) függvényében polinomiális időben kiszámíthatóságot érti.

Egyes speciális feladatokhoz szokás még más tulajdonságok teljesülését is előírni, ilyenek pl:

- *Közeli ütközés-ellenállóság* (near-collision resistance), amikor nehéz két különböző M, M' üzenetet találni, melyekre $H(M)$, $H(M')$ csak kevés számú bitben különbözik.
- *Részleges őskép-ellenállóság* (partial-preimage resistance), amikor az input egy része és $y = H(M)$ ismert, akkor is „nehéz" találni olyan M üzenetet, amelynek része az ismert input-töredék és $H(M) = y$; (pontosabban, ha az inputból t bit ismert, átlagosan 2^{t-1} hash-művelet szükséges az M input megtalálásához).
- *Korreláció-mentesség (non correlation):* rövid üzeneteknél feltétel (amikor az egyirányú függvényre az input és output nagyságrendje közeli (ilyen felhasználás például a jelszavak egyirányú képének letárolásakor, vagy blokkos rejtjelzéseknek, mint egyirányú függvényeknek a vizsgálatakor jelentkezik).
- *Univerzális egyirányú függvény (UOWHF):* néhány alkalmazásban – az egyszerűbb konstrukciók miatt – a CRHF alternatívájaként jelenik meg az UOWHF, melynél a lenyomatképző függvények egy indexelt halmaza áll rendelkezésre (azonos ősképtérrel és képtérrel), és az üzenetválasztás után választanak a lenyomatképző halmazból egyirányú függvényt.
- *Szabad kezdőérték melletti őskép-ellenállóság (pseudo-primage* resistance), amikor az egymást követő input blokkokon operáló iteratív lenyomatképző algoritmus (ahol a következő blokk képzésébe az előző blokk eredménye behat), véletlen input blokkal kezdődik (ez hat be az első blokk lenyomatképzésébe). Nyilván valamivel könnyebb feladat, ha a támadó is szabadon választhatja meg az első input blokkot, ezért az ilyen támadás kizárását is elő kell írni.

Léteznek ajánlások a képtér méretére [ld. Menezes–Oorschot–Vanstone: Handbook of Applied Cryptography c. könyvének 9.3. fejezet alapján]:

$$\begin{array}{ll} \text{OWHF-re} & n \geq 80, \\ \text{CRHF-re} & n \geq 160. \end{array}$$

Hash függvényekre nagyon sok algoritmust javasoltak, ilyenek például: az MD család (Message Digest rövidítéséből: MD2, MD4, MD5), az SHA család (SHA-0, SHA-1, SHA-256, HSA-384, SHA-512), RIPMD-160, és egyebek (HAVAL, N-HAsh, Snefru, Tiger, Whirpool).

A szakirodalom sokat foglalkozik ezek biztonságával, ld. például az alábbi cikkeket:

B. den Boer, Antoon Bosselaers, Collisions for the Compression Function of MD5, Eurocrypt, 93.

H. Dobbertin, Cryptanalysis of MD4, Fast Software Encryption, LNCS 1039, D., Springer-Verlag, 1996.

H. Dobbertin, Cryptanalysis of MD5 compress, presented at the rump session of EurocrZpt'96.

Hans Dobbertin, RIPEMD with Two-round Compress Function is Not Collision-Free, *J. Cryptology* **10(1)**, 1997.

H. Dobbertin, A. Bosselaers, B. Preneel, „RIPMEMD-160: A Strengthened Version of RIPMMD," Fast

P. R. Kasselman, W. T. Penzhorn, Cryptananlysis od reduced version of HAVAL, Vol. 36, No. 1, Electronic Letters, 2000.

Xiaoyoun Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu „Collisions for Hash Functions, MD4, MD-5, HAVAL-128 and RIPMD" című cikkükben a címben meghatározott hash függvények mindegyikére párokat adtak meg, melyeknek ugyanaz a hash képe, azaz nem teljesül az ütközés-állóság fontos követelménye.

Dan Kaminsky elkészített egy olyan perl scriptet („proof-of-concept" kódot), mellyel az MD5 algoritmust be lehet csapni. Meg is adott 2 file-t (fire.bin és ice.bin néven), melyek MD5 kódja megegyezik, de SHA1 kódja különbözik (ezzel is bizonyítva a tartalmak különbözőségét és az SHA1 „megfelelőségét". Bizonyítása:

fire.bin: md5 = 2fda7b311ce4d4f05256284e41843d87,
ice.bin: md5 = 2fda7b311ce4d4f05256284e41843d87,
fire.bin: sha1 = 5ee5f5edc5744a4803baaf0b4c869d65d6001888,
ice.bin: sha1 = 072c421c6e43db03204665c59ade2eb45827ae6c.

Ekkor még úgy gondolták, hogy az SHA1 megfelelő (ez szerepel az ETSI ALGO munkacsoportja ajánlásai között is).

Nemrég azonban a Shandong egyetem munkatársai (Xiaoyun Wang, Yiqun Lisa Yin, és Hongbo Yu) bejelentették az SHA1 feltörését.

(A kínaiak az összes lehetséges kombináció végigpróbálásához szükséges 2^{80} támadási próbálkozás helyett 2005 elején már 2^{69} művelettel sikerrel jártak, 2005 nyarán a CRYPTO' 2005 konferencián tett bejelentés szerint ezt csökkentették 2^{63}-ra, ami jelentős a 2^{64} határ áttörése miatt.)

8.3.5. Összegzés a kalkulációs biztonsághoz:

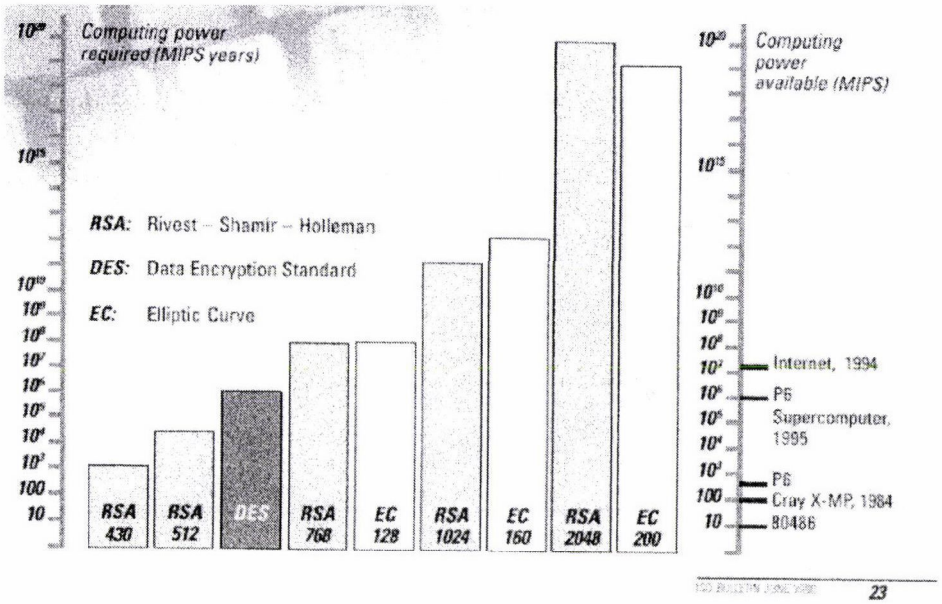
A kalkulációs biztonság elemzéséhez három tényező összevetése szükséges:

- a leghatékonyabb törési algoritmusok keresése;

- az algoritmus futtatásához használt (cél)gépek kapacitása (sebesség, tárhely);
- a feladathoz igénybe vehető gépek száma.

A kalkulációs biztonság megközelítéseihez a töréshez szükséges műveletszámok becslése szükséges.

Az alábbi táblázat az ISO Bulletin 2000. évi számából mutatja az egyes kriptográfiai primitívek (adott paraméterek melletti) támadási időszükségletét:



A táblázat bal oldala logaritmikus skálán mutatja a MIPS year szükségletet, jobb oldala az adott (jelzett) időben rendelkezésre álló kapacitásokat, például az összes internetre kötött számítógép számát jelzi.

Burt Kaliski, RSA Laboratories, IPA/TAO Cryptography Symposium, October 20, 2000, „Cryptography Trends: A US-Based Perspective" c előadásában vázolta az ekvivalens kulcsmérteket (hasonló táblázat több forrásban fellelhető):

Szimmetrikus kulcsú rejtjelzés	ECC	RSA (műveletben)
80	161	1024
96	192	1500
112	225	2048
128	257	3072

Az elektronikus aláírási rendszerekre az intenzív kutatások eredményei megközelítik az ETSI ALGO munkacsoportja által előírt 1020 bites modulus törési lehetőségét. A jelenlegi információk szerint a még (hosszú) kutatási fázisban lévő

TWINKLE és SHARK berendezések nincsenek elérhető közelségben (átlag támadóknak), így a jelenlegi 1020 bit körüli RSA-t használható rendszerek a közeljövőben még használhatók, de az új rendszereknél ajánlatos a nagyobb (például 2048 bites) modulus használata.

Az ETSI munkacsoport – nem hivatalos, informális (2005-ös) – tájékoztatása szerint a szabványos hash függvények (pl. SHA1) még használhatók. A bemutatott eredmények még nem indokolják az ajánlások érvénytelenítését, mivel a kutatási eredmények csak rendkívül számításigényesen használhatók két, aláírási szabványformátum szerint strukturált üzenet generálására, melyek lenyomata (hash képe) megegyezik.

A fentiek mutatják az igényt a kalkulációs biztonság pontosabb meghatározására, az ennek megfelelő garanciák elérésére. A vizsgálatok alapján indokolható, hogy 80 bites kulcsméret (2^{80} esetszám) már akár elegendő is lehet szimmetrikus kulcsú rejtjelzés esetén, de az is érthető hogy ma miért tartják minimumnak a kb. 2^{120} eset kipróbálási igényét.

Az 1990-es évektől új kutatási irányként jelent meg a kriptográfiai szakirodalomban a (NP elméleti módszerek felhasználásával) bizonyított biztonságú primitívek gyakorlati számításigény-becslési megközelítése.

M. Bellare, P. Rogaway: „Random oracles are practical: A paradigm for designing efficient protocols", In Proceedings of the First Annual ACM Conference on Computer and Communications Security, 1993; M. Bellare, P. Rogaway: „Optimal asymetric encryption – How to encrypt with RSA", EUROCRYPT'94; M. Bellare, P. Rogaway: „The exact security of digital signatures: How to sign with RSA and Rabin", EUROCRYPT'96. munkáikban nem aszimptotikus, hanem rögzített támadási erőforrás mellett vizsgálták a módszerek biztonságát.

9. Kvantum-kriptográfia és kriptográfiai biztonság

A klasszikus kriptográfia a makrovilág fogalmaival, a makrovilág lehetőségeit felhasználva tesz kísérletet az információvédelem problémáinak megoldására.

Ebből a keretrendszerből kilépve egy esetleges támadónak olyan lehetőségek állhatnak rendelkezésére, amelyek érvénytelenítik az addigi biztonsági megfontolásokat, illetve a védelem teljesen új dimenziói nyílnak ki.

Alapgondolat: a szubatomikus részecskék bizonyos körülmények között hullámként viselkednek, pl. az elektronok nem szigorúan egyetlen pontban, hanem a térben „elkenődve" helyezkednek el. Pozíciói hullámfüggvénnyel jellemezhetők, mely leírja, hogy a részecske milyen valószínűséggel található a tér különböző pontjaiban, a hullámfüggvények szuperpozíciói pedig egyszerre sok állapotot (műveletet) reprezentálhatnak.

A kvantumelmélet az adatvédelmet (ezen belül elsősorban a kriptográfia módszereit) a szakirodalom szerint három területen segítheti (a jövőben):

- kvantum számítások (quantum computation), kvantum számítógéppel;

- kvantum-kriptográfia (quantum cryptography);
- teleportálás és biztonságos kommunikációs csatorna kialakítása (quantum teleportation and communication).

9.1. Kvantum-számítógépek (számítások)

segítségével a hagyományos komputerekkel exponenciális idő alatt elvégzett számítások polinomiális időben elvégezhetőkké válnak.

A kvantum számítógép alapgondolata:

Az elemi információs egység – a hagyományos bit („binary digit") mintájára – a *qbit* („quantum bit"). Ezt elemi részecskék testesítik meg, amelyeknek bizonyos állapotát – például a forgásuk irányát – feleltetik meg az „1" illetve a „0" állapotnak. Ezek az elemi részecskék a kvantumelmélet törvényeit kihasználva *szuperpozíciós állapotba hozhatók*. Ilyenkor egyszerre veszik fel az „1" és a „0" értéket.

Az ilyen qbiteken végzett műveletek ekkor tulajdonképpen egyszerre minden lehetséges értéken végrehajtódnak. Ez egy qubit esetében 2 művelet egyidejű elvégzését jelenti, míg n qbit esetén már 2^n művelet egy lépésben való elvégzéséről van szó. A kvantumszámítógépekben a qubitek számának lineáris növelése exponenciális mértékben növeli a párhuzamosan elvégezhető műveletek számát.

Papíron sikerült már olyan gépet tervezni (1994, Peter W. Shor – AT&T Bell Labs), ami kvantummechanikai segédlettel képes arra, hogy egy **szám prímfelbontását** a számjegyek számának négyzetével arányos időben elvégezze. Hagyományos módon ez csak exponenciális idejű algoritmussal valósítható meg.

15 prímtényezőkre bontásához egy legalább hét kvantumbites kvantumszámítógép szükséges.

Az IBM kémikusai ezért olyan molekulát hoztak létre, amelyben a hét kvantumbitet hét magspin – öt fluor- és két szénatommag mágneses momentuma – alkotja. A kutatók kis üvegfiolában milliárdszor milliárd (10^{18}) ilyen molekulát tartalmazó oldatban futtatták le a Shor-féle algoritmust. A kvantumszámítógép működése során az információ beírása (a spinek beállítása) rádiófrekvenciás impulzusokkal, az eredmény kiolvasása pedig magmágneses rezonancia módszerrel (NMR) történt. Meg is kapták a helyes eredményt, miszerint 15 prímtényezői: 3 és 5.

9.2. Kvantum-kriptográfia (kulcsegyeztetés)

elsősorban a hagyományos titkosítás kulcsegyeztetése területén hoz teljesen új lehetőségeket:

- lehetséges olyan csatorna, ahol bármilyen mérési kísérlet szükségszerűen, észlelhető módon megzavarja a jelet;
- bizonyos fizikai jellemzők komplementer módon léteznek: az egyik mérése megzavarja a másikat.

Az egyik – egyelőre a kísérletekben – általánosan használt eljárás, a BB84 algoritmus menete:

1. Alice, a küldő – általa ismert – polarizációjú fotonokat generál és küld Bobnak;

2. Bob előre meghatározott méréssorozatot végez a kapott fotonokon;
3. Bob a sikeres mérések típusát nyilvános csatornán elküldi küldőnek;
4. Alice elküldi, melyik mérés típusa volt megfelelő;
5. ezek alapján a megfelelő típusú mérések eredményéből a közös kulcsbitsorozat úgy előállítható, hogy egy illetéktelen belehallgatás detektálható, mivel a csatorna támadója nem tudja egyszerre mérni a kétféle polarizációtípust, így nem minden esetben küld „ugyanolyan" fotont tovább (címzett felé), mint amit kapott.

Ehhez biztonsági szempontból fontos, hogy:

- küldőnek szűrőkkel le kell csökkenteni a fény intenzitását villanásonként átlagosan kevesebb mint egy fotonra (mivel az intenzitás-csökkentéssel négyzetesen csökken a lehallgatás lehetősége);
- csökkenteni kell az ún. „sötét számolás" kockázatát, mivel a jelenleg használatos detektorok néha jeleznek akkor is, amikor valójában nem is érkezett be foton, így az ehhez hasonló hibák „lehallgatást" jeleznek, mely megzavarja a kommunikációt.

A fenti problémák mellett is jelentős gyakorlati eredmények vannak a kvantum kulcsküldés területén:

- 2002. Los Alamos National Laboratory: 10 km, levegőben,
- 2002. Svájc 60 km, optikai szálon,
- 2002. QnetiQ Lab és a müncheni Ludwig Maximilian egyetem 23 km, levegőben.

Az Európai Unió 11 millió eurót különít el négy évre, hogy kifejlesszenek egy biztonságos kommunikációs rendszert, amely a kvantum-kriptográfiára épül. Kereskedelmi forgalomban is kaphatóak már olyan eszközök, amelyek ezen az elven valósítják meg a kulcsegyeztetést (www. idquantique.com).

9.3. Kvantum teleportálás (kommunikáció)

egyik módszere az EPR-hatáson (Einstein-Podolsky-Rosen) alapul:

Léteznek olyan részecskepárok, amelyek mindaddig elválaszthatatlan kötelékben (korrelációban, szuperpozícióban) maradnak, amíg – egy méréssel – „drasztikusan" szét nem választják őket. Az EPR elve:

- egy középpontosan szimmetrikus atom két fotont bocsát ki ellenkező irányba a két megfigyelő felé ismeretlen polarizációval;
- ha küldő és címzett ugyanolyan típusú mérést (+) hajt végre rajtuk, akkor egyforma valószínűséggel kapják az egyik eredményt (−), azonban ekkor a másik pont az ellenkezőjét (|) kapja és viszont;
- mindkét foton polarizációja csak akkor határozódik meg, amikor valamelyikét megmérjük, a távolságtól függetlenül.

- Első próbálkozások: 1982–84-ben voltak.

- Működő prototípus: 1989. IBM.

A módszert *Charles Bennett* 1989-ben Montreálban, az IBM laboratóriumában négy kollégájával együtt a gyakorlatban is megvalósította. Az első prototípusban zöld lézert használtak, és a fotonokat kb. 30 cm távolságra tudták továbbítani.

- 1991-ben az Oxfordi Egyetemen Arthur Ekert az „összetartozó" kvantum rendszerek – mint például az egyszerre létrehozott foton-párok – azon speciális tulajdonságát használta ki, hogy az egyes kvantumok megőrzik bizonyos közös tulajdonságaikat a szétválásuk után is. 1994-ben sikerült egy ezen az elven működő rendszert a gyakorlatban is megvalósítani.

Néhány web-cím, ahol további részletek olvashatók:
`http://www.qubit.org/library/intros/comp/comp.html`
`http://www.qubit.org/library/intros/compSteane/qcintro.html`
`http://www.qubit.org/library/introductions.html`
`http://www.titoktan.hu/_raktar/honlapok.htm`

10. Rendszerszemléletű biztonság

Az eddigiekben elsősorban a kriptográfiai primitívekre vonatkozó biztonsági fogalmakat tekintettük át. Ahogy a korábbi fejezetekben szerepelt, a felhasználói rendszerek kriptográfiai alrendszerének hierarchikusan egymásra épülő elemei:

- a kriptográfiai primitívek;
- kriptográfiai sémák;
- kriptográfiai protokollok;
- kriptográfiai rendszer;
- informatikai rendszer.

Nem elégséges önmagában vizsgálni a primitíveket, hanem azok biztonsága sokszor alkalmazás-függő.

10.1. Sémák és protokollok

Jó példa a primitív és protokoll illesztetlenségéből adódó veszélyekre a „legerősebb" primitív, a OTP és az ún. kétkulcsos láda együttes használata:

Legyen a két kommunikáló fél: A, B.

Tegyük fel, hogy A egy M üzenetet kíván titkosan küldeni B-nek, melynek hossza N.

Mindkét fél generál magának nagy mennyiségű véletlen elemet, mely véletlen sorozatokat kizárólag a generáló fél (A, vagy B) ismer.

Jelölje: Va és Vb a két oldalon generált sorozatból kivett N hosszú véletlen sorozatot.

Ha A küld egy M üzenetet titkosan B-nek, akkor a protokoll szerint a következő lépéseket kell végrehajtaniuk:

Lépés	A számol	irány	Vonalon utazik	B számol
1.	$M + Va$			
2.		$\rightarrow$	$M + Va$	
3.				$M + Va + Vb$
4.		$\leftarrow$	$M + Va + Vb$	
5.	$M + Va + Vb - Va$			
6.		$\rightarrow$	$M + Vb$	
7.				$M + Vb - Va = M$

A protokoll (leszámítva középen aktív támadás módszerét) megfelelő, mégis a jó protokoll és a jó primitív együtt passzív (lefigyeléses) módszerrel is támadható. A 4. és 2. lépés adatainak kivonásával a támadó is megszerezheti Vb-t, amiből a 6. lépésben utazó információból kiszámíthatja M-et!

A nyilvános kulcsú rendszerekkel a legnagyobb probléma az ún. „rosszindulatú postás", az „intruder in the middle", vagy más elnevezéssel az „attack in the middle" támadás kivédhetetlensége. (Ezért olyan bonyolultak az aláírási rendszerek, a két kommunikáló fél mellett be kell vonni egy ún. TTP (Trusted Third Party) résztvevőt, amelyet az aláírási rendszerekben CA-nak (Certificate Authority-nek) neveznek.

A kétkulcsos láda fenti modelljében B-t helyettesítheti a postás (egy intruder), aki először elvégzi a protokoll-lépéseket (B-nek hazudva magát) A-val, majd utána B-vel is. Erről részletesebben majd a későbbiekben az ún. „grand master chess attack"-nál.

Az alábbiakban – példaszerűen – néhány protokolltámadási kutatási irányt említünk meg:

- „New Attacks on PKCS#1 v1.5 Encryption" című, a EUROCRYPT 2000 konferencián elhangzott előadásukban Jean-Sébastien Coron, Marc Joye, David Naccache, and Pascal Paillier foglalkoztak a címben jelzett szabvány eljárás támadási lehetőségével.
- A CRYPTO 2003 konferencián Brice Canvel, Alain Hiltgen, Serge Vaudenay, and Martin Vuagnoux „Password Interception in a SSL/TLS Channel" címmel a MAC használatának CBC módban lévő gyengesége javításával foglalkozik.
- A CRYPTO 2002 konferencián Jakob Jonsson and Burton S. Kaliski Jr. „On the Security of RSA Encryption in TLS" című előadásukban vizsgálták a TLS-ben alkalmazott RSA protokollrészét, a hand-shake alkalmazásának biztonságát.
- Az ASIACRYPT 2001 konferencián Phong Q. Nguyen and Igor E. Shparlinski „On the Insecurity of a Server-Aided RSA Protocol" c előadásukban a SASC protokoll támadásáról szóltak (passiv attack-kal)
- A CRYPTO 2001 konferencián James Manger „A Chosen Ciphertext Attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as Standardized in PKCS #1 v2.0" címmel tartott előadást.

- A szakmai közvéleményt foglalkoztatja a kvantum-kriptográfia jövője, ezen belül az egyre inkább gyakorlati alkalmazási közelségbe kerülő kvantum-kulcscsere biztonsága. Ennek problematikájáról szólt Donald Beaver előadása a EUROCRYPT 2002 konferencián „On Deniability in Quantum Key Exchange" címmel.

10.2. Kriptográfiai alkalmazásokat futtató informatikai rendszerek biztonsága

Nyilvánvaló, hogy biztonságos primitívek, sémák, protokollok sem teljesíthetik védelmi feladataikat nem biztonságos rendszerkörnyezetben.

10.3. Egyéb támadások elleni ellenállóképesség

Az utóbbi időben nagyon sokan vizsgálják az RSA támadását fizikai eszközök igénybevételével is, az ún. power attack, vagy a timing attack segítségével. Ezzel a két támadási típussal hatékonyan támadhatók pl. megfelelő paraméterekkel rendelkező, algoritmikus támadások ellen biztonságos RSA rendszerek is.

Például a „Public Key Cryptography: 5th International Workshop on Practice and Theory in Public Key Cryptosystems, Paris, France, February 2002. konferencián elhangzott *Side-channels attacks* szekcióban ***Roman Novak*** előadó részéről az **„SPA-Based Adaptive Chosen-Ciphertext Attack on RSA Implementation"** c. előadásban szerepelt a jobboldalon látható, a SmartCard energiafelvételének méréséből az RSA működésére, titkos paramétereire következtethető ábra.

Szintén a PKC,2002-es konferencián hangzott el **„A Combined Timing and Power Attack"** címmel ***Werner Schindler*** előadása a két támadási módszer hatékony kombinációjáról.

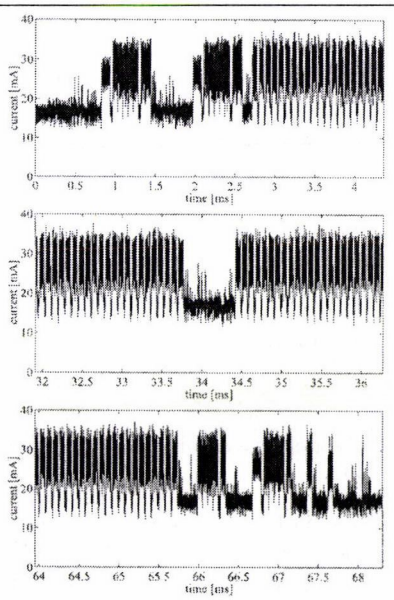


Fig. 1. Typical power consumption patterns during RSA decryption

Ezen támadások ellen is lehet algoritmikus eszközökkel védekezni (a számítási idő rovására), ha kellő programlépéseket kiegészítenek „fal" utasításokkal, melyek az áramfelvételeket, illetve időráfordításokat kiegyensúlyozzák (függetlenítik a kulcsoktól).

AZ ETSI CEN munkacsoportja az elektronikus aláírási követelmények között megjelentette a CWA (Cen Workshop Agreement) 14890–1:2004 előírást, melynek A.5 mellékletében hívja fel a figyelmet a kommunikáló felek közötti protokoll ún. „Grandmaster Chess Attack" sok rendszerben kivédhetetlen módszerére. Ez a módszer közbeékelődő támadással szimulálja mindkét fél felé a lépéseket (mint ha két nagymesterrel egyszerre sakkoznánk, akik nem tudnak egymásról, és az ő lépéseikkel játszunk). A modellt a mellékelt, CWA 14890–1-ből vett ábrával szemléltetjük:

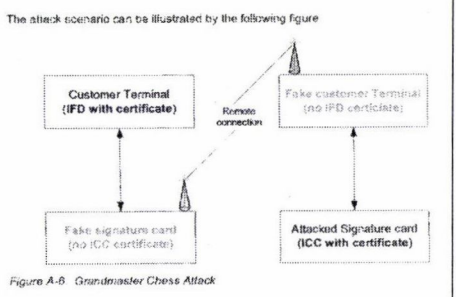


Figure A.6 Grandmaster Chess Attack

Ennek kivédése nagyon bonyolult rendszert igényel. Erre látunk példát I. Zs. Berta, L. Buttyán, I. Vajda: Mitigating The Untrasted Terminal Problem Using Conditional Signatures, IEEE, 2004, april. cikkében, ahol a nembiztonságos terminálok biometriás kiegészítésével, és sűrű időbélyegzéssel nehezítik (teszik lehetetlenné) a közbeékelődő támadó tevékenységét.

Biztonságosnak kell lennie annak a platformnak, ahol a kriptográfiai rendszer működik.

Kritográfiai modulokra a FIPS 140–2 szabvány ír elő követelményeket, általános informatikai környezetre pedig a magyar, európai és világszabvány MSZ ISO/IEC 15408 (Common Criteria: Közös szempontok az informatikai biztonság értékelésére) határozza meg a biztonság elbírálásának szempontjait.

A biztonságos informatikai rendszer (mely nélkül nem működtethető biztonságosan a kriptográfiai alrendszer) a szakirodalom szerint több tényező kezelését igényli:

- *Adminisztratív biztonság*
  - Szervezeti biztonság
  - Személyi biztonság
  - Üzletviteli biztonság

  Kapcsolódó szabványok:

  ISO 9001 (Quality Management systems)

  NIST 800-12 /Kapcsolódó fejezetek: Computer Security Policy, Computer Security Risk Management; Security and Planning in the Computer system Life Cycle; Personal/User issues; Computer Security Incident handling; Awareness, training and Education, Preparing for Contingencies and Disasters/;

  ISO/IEC 17799 / Kapcsolódó fejezetek: Information Security Management: Security Policy; Security Organization; Personal Security; Business Continuity Management/;

- *Fizikai biztonság*

  NIST 800–12 / Kapcsolódó fejezetek:An Introduction to Computer Security; Physical and Enviromental Securiy/

  ISO/IEC 17799 / Kapcsolódó fejezetek: Information Security Management; Physical and Enviromental Securiy/

- *IT Biztonság*

  NIST 800-14

  NIST 800-12 / Kapcsolódó fejezetek: An Introduction to Computer Security/ 16. fejezet Identification and Authentication; 17: Logical Access Control; 18: Audit Trails; 19: Cryptography/

  ISO/IEC 17799 / Kapcsolódó fejezetek: Information Security Management, Communication and Operations Management; Access Control; System Development and Support process/

  Legmeghatározóbb szabvány: MSZ ISO/IEC 15408

Az 1970-es években a DoD és az IBM 40 millió USD-t költött az ún. „tiger team"-ekre, a rendszerek gyenge pontjainak megtalálására.

Ennek alapján állítottak fel biztonsági modelleket, melyek közül a leghíresebb a **Bell–LaPadula modell** (D. E. Bell and L. J. LaPadula, Secure Computer Systems: Mathematical Foundations and Model, Mitrte Corp., Bedford (MA), 1973.), melyben matematikai formalizmussal kezelték a biztonsági elvárásokat.

Ez a modell volt az alapja a **UNIX** biztonsági rendszerének, valamint erős hatással volt a biztonsági követelmények, kiértékelési rendszerek kidolgozására is, és így a TCSEC (Trusted Computer System Evaulation Criteria)-Orange Book sorozata kidolgozására.

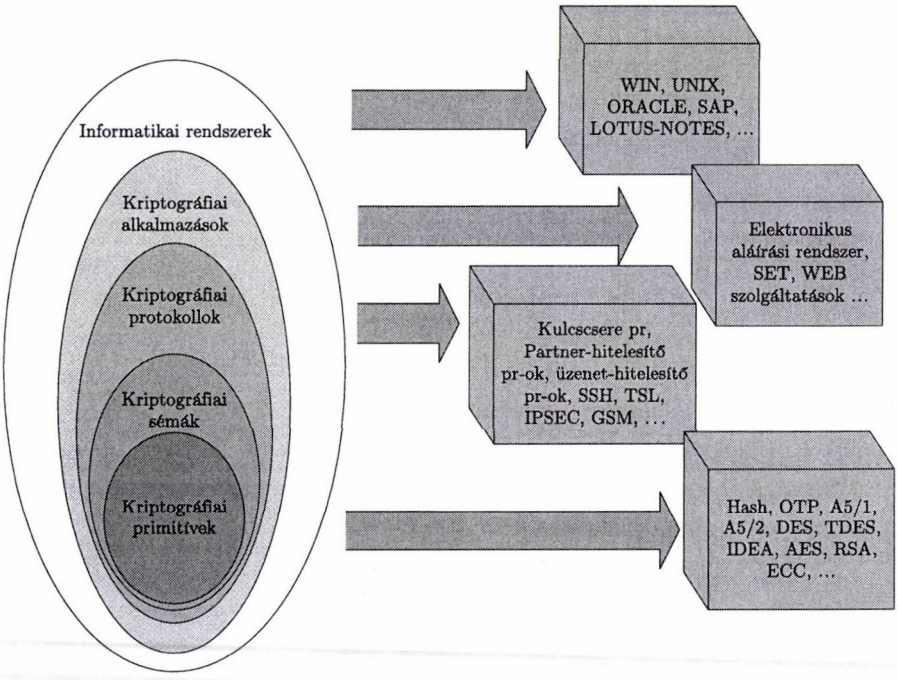
A TCSEC további nemzeti, illetve egyéb közösségi biztonsági módszertanok kidolgozását inspirálta, pl

- **ITSEC:** Information Technology Security Evaulation Criteria (Európai módszertan)
- **FC:** Federal Criteria for Information Technology Security
- **CTCPEC:** Canadian Trusted Computer Product Evaulation Criteria

A fentiekből a nemzetközi testületek összefogásával alakították ki a Common Criteria (röviden CC) követelmény-rendszert, mely magyar, európai és ISO szabvány is lett:

MSZ-ISO/IEC 15408 2002–2003.

11. Összefoglaló ábra



Mellékletek

Kriptográfiai szabványok

Kriptográfiával kapcsolatos szabványokat több szervezet kiadott, ezek között vannak olyan nemzeti szervezetek, melyek szabványait nemzetközi szinten is elfogadják (pl. FIPS), és vannak nemzetközi szervezetek.

Az egyes szervezetek szabványai között jelentős átfedések is vannak, és többségük az interneten ingyenesen hozzáférhető. Ezért itt csak példaként, az eligazodás segítésére említünk néhány szabványt, az érdeklődők a szervezetek honlapjain a teljes listát letölthetik.

ISO: International Organization for Standardization

- ISO 8372 Modes of Operation for 64-bit Cipher
- ISO 9796 Data Integrity Mechanism (MAC)
- ISO 9798 Entity Authetication
- ISO 9779 Register of Cryptographic Algorithm
- ISO 10118 Hash Functions
- ISO 11770 Key Management

- ISO 13888 Non-repudiation
- ISO 14888 Signatures with Appendix

Federal Information Processing Standards Publication

- FIPS PUB 31 Guidelines to ADP Physical Security and Risk Management.
- FIPS PUB 39 Glossary for Computer Systems Security.
- FIPS 46–2 DES
- FIPS 46–3 3DES üzemmódok
- FIPS PUB 73 Guidelines for Security of Computer Applications.
- FIPS PUB 74 Guidelines for Using DES
- FIPS PUB 81 DES Modes of Operation.
- FIPS PUB 87 Guidelines for ADP Contingency Planning.
- FIPS PUB 112 Password Usage.
- FIPS PUB 113 Computer Data Authentication. (CBC MAC)
- FIPS PUB 140–1,–2 Security Requirements for Cryptographic Modules.
- FIPS PUB 185 Key Escrow (Clipper and Skipjack)
- FIPS 186 Secure Hash Algorithm ANSI X9.42, Agreement of Symmetric Keys on Using Diffie–Hellman and MQV Algorithms
- FIPS 196 Entity Authentication (asymetric)

American National Standards Institute

- ANSI X.92 Data Encryption Algorithm (DEA)
- ANSI X3.106 Data Encryption Algorithm (DEA) Modes
- ANSI X9.8 PIN Management and Security
- ANSI X9.9 Message Authentication
- ANSI X9.23 Encription of Messages
- ANSI X9.28 Multi-center Key Management
- ANSI X9.30–1 Digital Signature Algorithm (DSA)
- ANSI X9.30–2 Secure Hash Algorithm (SHA) for DSA
- ANSI X9.31–1 RSA Signature Algorithm
- ANSI X9.31–2 Hashing Algorithm for RSA
- ANSI X9.42 Agreement of Symmetric Keys on Using Diffie–Hellman and MQV Algorithms
- ANSI X9.52 Triple Data Encryption Algorithm Modes of Operation
- ANSI X9.57 Certificate Management

Public Key Criptographic Standards

- RSA PKCS#1 RSA Encription Standard

- RSA PKCS#3 Diffie–Hellman Key-aggreement Szandard
- RSA PKCS#5 Pasword-based Encryption Standard
- RSA PKCS#7 Cryptographic Message Syntax Standard
- RSA PKCS#11 Cryptographic Token Interface Szandard

Request for Comments /Internet „szabványok"/

- RFC 1321 MD5 Hash Function
- RFC 1421 PEM Encription, autjentication
- RFC 1938 One-time Pasword System
- RFC 2246 Tranport Layer Security, TLS V1.0
- RFC 2808 HTTP over TLS
- RFC 2510 Internet X.509 Public Key Infrastructure Certificate Management Protocols
- RFC 2104 HMAC: Keyed Hashing for Message Authentication
- RFC 2402 IPSEC Cryptography
- RFC 2406 IPSEC Authentication

Néhány fontos magyar és egyúttal ISO szabvány az IT rendszerek biztonságához kapcsolódva

- *MSZ ISO/IEC 13888-1:2001* Információtechnika. Biztonságtechnika
- *MSZ ISO/IEC 17799:2002* Informatika. Az informatikai biztonság menedzselésének eljárásrendje
- *MSZ ISO/IEC 9594-8 Informatika. Nyílt rendszerek összekapcsolása. Névtár: Nyilvános kulcs és attribútum tanúsítási keretrendszer*
- *MSZ ISO/IEC TR 13335-1* Informatika. Irányelvek az IT-biztonság menedzseléséhez
- *MSZ ISO/IEC 15408-1:2002 Informatika. Biztonságtechnika. Az informatikai biztonságértékelés közös szempontjai*
- *MSZ EN 45011:1999 Terméktanúsítási rendszereket működtető szervezetre vonatkozó általános követelmények,* és az ezt kiváltó *MSZ ISO/IEC 2700:2006*
- *MSZ EN 17025:2001* Vizsgálólaboratóriumok működésének általános feltételei

Néhány fontosabb hivatkozás

[1] Dan Boneh, Twenty Years of Attack on the RSA Cryptosystem, *Notices of the AMS,* **46(2)** (1999), 2003–213, `http://crypto.stanford.edu/~dabo/abstract/RSAattack-survey.html`

[2] Buttyán Levente, Vajda István, *Kritográfia és alkalmazásai,* Typotex Elektronikus Kiadó Kft. (2004).

[3] Kahn, D., *The Coedebreakers,* New York (1996).

[4] Knuth, D. E., *A számítógép-programozás Művészete 2. Szeminumerikus algoritmusok,* Műszaki Kiadó (Budapest, 1987).

[5] Alfred J. Menezes, Paul C. van Oorshchor, Scott A. Vanstone, *Handbook of Applied Cryptography,* CRC Press (1997), http://www.cacr.math.uwaterloo.ca/hac/

[6] Nemetz Tibor, Vajda István, *Algoritmusos adatvédelem,* Akadémiai Kiadó (Budapest, 1991).

[7] Bruce Schneier, *Applied Cryptography: Protokols, Algorithms, and Source Code in C,* John Wiley & Sons (New York, 2nd edition, 1996).

[8] Bruce Schneier, Adam Shostack, *Breaking Up Is Hard To Do: Modeling Security Threats for Smart Cards,* http://www.counterpane.com/smart-card-threats.html

[9] Shannon, C. E., Communication theory of secrecy systems, *Bell Syst. Techn. J.,* **28** (1949), 656–715.

[10] Virrasztó Tamás, *Titkosítás és adatrejtés, Biztonságos kommunikáció és algoritmikus adatvédelem,* NetAkadémia Oktatóközpont (Budapest, 2004).

DIFFERENT APPROACHES TO THE SECURITY OF CRYPTOGRAPHY – BASED SECURITY MECHANISMS

PÁL PAPP, ISTVÁN SZABÓ

The goal of this paper is to give a broad overview of the main security problems of cryptography, and the different tools we have to handle these problems. In this paper we analyze the security of different crypto mechanisms focusing ont he security of Public Key Infrastructure and its elements. We overview and classify the threats, and the well-known attacking method, we deal with the different ways to measure cryptographic security. We also examine the information-theoretic approaches, and provable security. We also classify different sets of cryptographic primitives. After the theoretic approach, the topics of crypto standards, application, protocols are also dicsussed.

Alkalmazott Matematikai Lapok **23** (2006), 295–312.

MATEMATIKA A KRIPTOGRÁFIÁBAN: ÍZELÍTŐ

NEMETZ TIBOR

Budapest

A *kriptográfia* általános értelemben mindazoknak az eljárásoknak, algoritmusoknak, biztonsági rendszabályoknak kutatását, alkalmazását jelenti, amelyek információnak illetéktelenek elől elrejtését és az elektronikus aláírást hivatottak megvalósítani. Szokásos a *kriptológiá*nak és a kriptográfiának a megkülönböztetése. Ez utóbbi a fentieken kívül magában foglalja az illetéktelen megfejtés algoritmikus és hírszerzési vonatkozásait is. Szűkebb értelemben szokás a kriptográfiát a rejtjelző algoritmusok világára korlátozni. Ez a szűkebb terület lényegében véve matematika, amely a matematika számos részterületét öleli fel. Ez a megfigyelés indokolja, hogy egy általános, ízelítő jellegű ismertetést bocsássunk az érdeklődő olvasók rendelkezésére. Ismertetésünk korántsem lehet teljes. Egy népszerűsítő szinten megírt kriptográfia történelem ezer oldalnyi anyagot jelent, amint azt a „kriptográfia bibliája"-ként idézett Kahn könyv [8] mutatja.

Információk titkosítására vonatkozó *igény gyakorlatilag egyidejű az írásbeliség megjelenésével.* Az igény elsősorban a hatalom világi és egyházi urai részéről jelentkezett. Hadvezérek, diplomaták részéről az igény természetes és alapvető jelentőségű volt. Gyakorlatilag kisajátították maguknak az elméletet és a gyakorlatot egyaránt. Ez az ősállapot a 70-es évek elejéig tartott, amikor a hírközlés robbanásszerű fejlődése azt az igényt is megteremtette, hogy akár ismeretlen emberek is biztonságos, mások számára érthetetlen levelezést folytathassanak egymással anélkül, hogy erre vonatkozóan korábban bármilyen közös megállapodást kellett volna tenniük. A korábbi *zárt hálózatok* helyébe *nyílt hálózatok* léptek, ami lényeges szemléletváltást tett szükségessé.

Zárt hálózat: olyan hírközlő hálózat, amelyben minden felhasználó előre ismert, nyilvántartott. Senki sem léphet be új felhasználóként önkényesen a rendszerbe. A rendszeren belüli forgalmazás csak a hálózat által elfogadott azonosítás megtörténte után lehetséges.

Nyílt hálózat: nyilvánosan közzétett szabályok szerint bárki csatlakozhat a hálózathoz.

A központosított zárt (hivatali titoktartással működő) hálózatokkal szemben a nyílt hálózatok digitális világában a magánélet védelme, valamint az adatok biztonsága érdekében másféle intézkedésekre van szükség. Két szokásos meghatározás:

Adatbiztonság: digitális adatok sértetetlenségét, illetéktelen személyek által történő megismerését megakadályozó módszerek összesége.

Adatvédelem: azoknak a módszereknek az összesége, amelyek megakadályozzák az adatok alapján a személyiségi jogok megsértését, például hozzáférésvédelem biztosításával.

Ennek eszköze a nyilvános kulcsú kriptográfia.

Nyilvános kulcsú kriptográfia: olyan kriptográfiai rendszer, amelynek a résztvevői közös algoritmust használnak rejtjelezésre, és az algoritmusnak két – a használótól függő – kulcsa van. Ezek egyikét (nyilvános kulcs) nevükkel együtt nyilvánosságra hozzák, a másikat titokban tartják (magánkulcs). A kulcsok egyikét a rejtjelzésre, a másikat a (jogosult) megoldásra használják. Az adatvédelmi és adatbiztonsági feladatok megoldása az elektronikus társadalomba való átmenet legaktuálisabb kihívását jelentik. Ez a téma felöleli az elektronikus üzletvitel (kereskedelmi és pénzügyi szolgáltatások) általánossá válását, ennek *adatvédelmi és adatbiztonsági vonzatait*. Ezek között kiemelkedő szerepet játszik a *digitális aláírás,* és ennek egy *digitális közjegyző által történő hitelesítése,* törvényes elfogadhatósága.

Szóhasználat

Digitális aláírás: Olyan titkosított karaktersorozat, melyet igen nagy valószínűséggel csak az aláíró kódolhatott, s ez magából a kódolásból következik. Keltezést (dátumot, pontos időpontot), sorszámot, a küldött teljes üzenetből képezett ellenőrző számot tartalmaz.

Digitális közjegyző: (CA = Certification Authority) Olyan szakosodott szervezet vagy cég, amely tanúsítványokat adhat ki kliensek és szerverek számára. Igazolja, hogy egy adott azonosítóval rendelkező felhasználó az, akinek vallja magát.

A nyilvános hálózatok „munkaanyaga" egy *digitális dokumentum.* Ez a 40/1998. III.6. sz. Korm. Rendelet szerint „Számítástechnikai program felhasználásával – elektronikus formában rögzített – elektronikus úton érkezett, illetve továbbított irat, amelyet számítástechnikai adathordozón tárolnak. Számunkra ennél egyszerűbben, digitális dokumentum egy véges ábécé betűiből álló tetszőleges sorozat.

A fenti kihívásnak hatékonyan megfelelni csak gyorsan lehet: A távközlési kutatások felgyorsulására jellemző, hogy az eredmények (gyakorlati) bevezetését ma már szinte a(z alap)kutatással párhuzamosan kell megkezdeni." (Boda, 1999)

Matematikai modell

A *kriptográfiai algoritmus*ok egy véges X input ábécé véges $x \in X^\infty$ sorozataihoz rendelnek hozzá egy véges Y output ábécé betűiből álló $y \in Y^\infty$ sorozatot. A hozzárendelést az információelméletben szokásos módon értelmezett *kódok egy paraméteres családja* valósítja meg. Minden digitális dokumentumhoz ezek egyikét

kell kiválasztani. A kiválasztott paramétert *kulcs*nak nevezik, a lehetséges paraméterek halmazát *kulcstér*nek.

Az alkalmazott kódok tipikusan blokk-kódok. Felidézzük, hogy egy $n \to k$ egyértelműen dekódolható kódot egy $f : X^n \to Y^k$ és egy $\varphi : Y^k \to X^n$ függvény definiál, amelyre teljesül, hogy minden $x \in X^n$ esetén $\varphi\big(f(x)\big) = x$. A gyakorlatban általánosan $X \equiv Y$ (és így $n = k$). Ha $n = 1$, akkor betűnkénti kódolásról (stream chipher-ről) beszélünk. Ez akkor is így van, ha a kód betűről-betűre haladva változik (idő-függő, azaz függ a kódolandó betű pozíciójától).

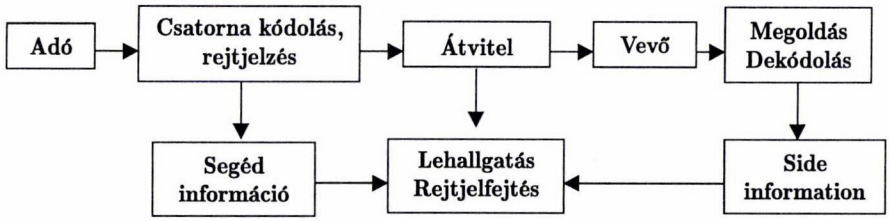
1. Példa. Történelmileg az első betű-kódot Caesar római császárnak tulajdonítják. Ő egy írott latin nyelvű szövegben minden betűt az alfabetikus sorrendbe írt 25 betűs latin ábécében a rá következő negyedik betűvel helyettesített, miközben a szóközt mindig megtartotta. Ma Caesar eljárás alatt értjük az output ábécé rögzített sorrendje mellett a betűk helyettesítését az alfabetikusan rájuk következő c-edik betűvel. Ekkor tehát c megválasztása jelenti a kulcs rögzítését, a kulcstér mérete pedig azonos az ábécé betűszámával.

A Caesar eljárás lehetővé teszi, hogy az algoritmikus rejtjelzés egyik alapvető fogalmát, az *egyértelműségi pontot* megvilágítsuk. Egy rejtjelző algoritmus egyértelműségi pontja az a minimális távolság, amely lehetővé teszi a kulcs azonosítását. Ha ilyen nincs, akkor a távolságot végtelennek tekintjük. Maga a távolság függ a forrás által generált sorozattól, tehát ez egy valószínűségi változó. Szokás ennek a várható értékét tekinteni egyértelműségi pontnak, de a fogalom precizírozásától itt eltekintünk. Megjegyezzük azonban, hogy a gyakorlatban a véletlentől függő „valószínűségi változó közel van a várható értékéhez". Európai nyelvek esetén a Caesar eljárás egyértelműségi pontja 4-5 betű. Állításunk szabatos bizonyítása helyett javasoljuk, hogy az olvasó erről „empirikusan" győződjön meg. Válasszon egy könyvből (napilapból) egy 4 (5) betűs blokkot, és próbálja azt elölről-hátulról hosszabb értelmes szöveggé kiegészíteni. Tapasztalataink szerint az esetek döntő többségében ez csak egyféleképpen tehető meg. További részletekre vonatkozóan utalunk Nemetz–Vajda [17] könyv 3.1. fejezetére.

A kriptográfia matematikai modelljét az információelmélet megalapozásával egyidőben C. E. Shannon (1949) írta le a II. világháborús rejtjelfejtői tevékenységének tapasztalatait felhasználva. A modell blokkdiagrammját az 1. ábra mutatja. Ezen az ábrán az illetéktelen fejtő bekapcsolódásának a lehetősége is fel van tüntetve, utalva az illetéktelen információszerzés lehetőségére is. Az átviteli csatorna, csatorna kódolás is olyan régi, mint maga a Caesar eljárás, akkor az információ táv-átvitelét Polübiosz „fáklyatávírója" valósította meg, lásd Révay [23].

Egy kriptográfiai eljárás bevizsgálása során alapvető feltétel, hogy az ellenfél (potenciális illetéktelen megfejtő) a paraméteres családot ismeri (például megvette egy alkalmazottól).

Ilyenkor a titkosságot csak a kulcs megválasztása jelenti, így azt úgy kell megválasztani, hogy illetéktelenek számára maximális bizonytalanságot tartalmazzon. Ezt úgy lehet biztosítani, hogy minden dokumentumhoz egy új kulcsot választunk, mégpedig minden előzőtől függetlenül, egyenlő valószínűséggel. Megérkeztünk is-



1. ábra. Rejtjelzéssel kombinált információátviteli modell

mertetésünk első matematikai részterületére: a *véletlenszám generálás, véletlen választás* témakörébe.

A véletlenszerű választás feltétele, hogy ismerjük a kulcsteret. Ez nem mindig triviális, ugyanis bizonyos eljárások a fejtés szempontjából ekvivalensek lehetnek. Ennek bemutatására szolgál a következő példa, melyben egy formális bizonyítást is elvégzünk.

2. Példa. Egyszerű helyettesítés. Ennek alkalmazása szerint a dokumentumban az input ábécé valamennyi betűjét helyettesítjük az output ábécé valamelyik betűjével, ugyanazt a betűt mindig ugyanazzal, különböző input betűket különböző output betűkkel. Mivel a fantázia rendkívül sokféle output ábécét tud elővarázsolni, így a kulcstér is végtelen nagynak tűnhet. Valójában azonban ezek nagy része egymással ekvivalens. Nevezetesen igaz a következő

1. TÉTEL. *Tetszőleges output ábécé melletti tetszőleges helyettesítés ekvivalens az output ábécé = input ábécé melletti valamelyik helyettesítéssel abban az értelemben, hogy az egyik akkor és csak akkor fejthető meg, ha a másik is megfejthető.*

Bizonyítás. Tekintsük az output ábécének azt a megszorítását, amely csak olyan betűkből áll, amelyek a rejtjeles szövegben előfordulnak, és a továbbiakban csak ezt tekintsük output ábécének. Ennek mérete nem lehet nagyobb, mint az input ábécéé. Megadunk egy kölcsönösen egyértelmű megfeleltetést a redukált output ábécé és az input ábécé ugyanennyi betűt tartalmazó részhalmaza között.

Tekintsük a rejtjeles szövegben előforduló első output betűt, és feleltessük meg ezt a rendezett input ábécé első betűjének. Folytassuk ezt az eljárást lépésről-lépésre. Tekintsük a rejtjeles szöveg következő olyan output betűjét, amelyhez még nem rendeltünk egyetlen input betűt sem, és rendeljük ehhez az első, még semmihez sem rendelt input betűt. Mivel a dokumentum véges, így a hozzárendelés is véges idő alatt sikeresen befejeződik. Ezzel a „közvetett" hozzárendeléssel az input ábécét önmagába képeztük le. Ha vannak még össze nem rendelt betűk, akkor azokat önkényesen párosíthatjuk: a konkrét feladat számára ez lényegtelen. Innen a tétel állítása azonal következik.

KÖVETKEZMÉNY. *Az egyszerű helyettesítéses eljárásoknál mindig feltehető, hogy az output ábécé azonos az input ábécével, és maga a helyettesítés ekvivalens az input ábécé egy permutációjával. Ezzel* elérkeztünk a kombinatorika területére. *Itt a véletlen kulcsválasztás egy véletlen permutáció megválasztását jelenti. Vonatkozó irodalom:* Knuth [9], Nemetz [19].

Még az ókorból ismert, hogy az egyszerű helyettesítéses eljárás *statisztikai módszerrel* megfejthető. A *valószínűségszámítás* nagy számok törvénye kimondja, hogy ergodikus, stacionárius sorozatok esetén nagy minta relatív gyakoriságai megbízható becslést adnak a valószínűségekre. Ha tehát egy nagy minta alapján megállapítjuk a betűk gyakorisági sorrendjét, akkor a rejtjeles szöveg gyakoriságairól is feltételezhető, hogy ezt a sorrendet követik. Ez persze csak korlátozottan érvényes, hiszen egy rejtjeles szöveg ehhez általában nem elég hosszú. Feltehető a kérdés, hogy mekkora az a hosszúság, amely elegendő az eredeti dokumentum visszaállításához. Erre elméleti választ az egyértelműségi pont ad meg, amelynek a meghatározásához csak a kulcstér méretét és a forrás *entrópiáját* kell ismernünk

Az entrópia az *információelmélet* egyik alapfogalma. Írott nyelvek entrópiájának meghatározására különböző módszereket dolgoztak ki, amelyek közül az elsőt szintén Shannon publikálta, lásd Shannon [26]. Európai írott nyelvek entrópiája 1,3 körül van (optimális kód esetén ennyi bit kell átlagosan egy betű kódolásához). Magyar nyelvre vonatkozóan lásd Nemetz–Simon [16]. Ebből az egyértelműségi pontra egyszerű helyettesítés esetén 22-24 betű adódik, ami teljes összhangban van gyakorlott rejtjelfejtők tapasztalataival.

Ugyancsak permutációkat alkalmaz az ókor másik gyakori rejtjelző eljárása a *pozíciócserés eljárás* (keverés, angolul: transpozition). Ennek során a dokumentumot adott hosszúságú blokkokra osztják fel, majd a blokkokon belül az egyes pozíciókban levő betűket a megválasztott permutáció szerint összekeverik. Pusztán érdekességként említjük meg, hogy Verne „Sándor Mátyás" című regényében alkalmazott „rácsos" rejtjelzés is ebbe a kategóriába tartozik. Népszerűsítő szintű leírást ad meg Nemetz [19].

Ebben az esetben az egyszerű betűstatisztika nem nyújt segítséget az illetéktelen fejtőnek, hiszen az a keverés után nem változik meg. Eggyel magasabb rendű, betűpárokra vonatkozó statisztika azonban most is alkalmazható. Ez az alkalmazás a *statisztikai hipotézis vizsgálatok* területére vezet el bennünket. További részleteket árul el a fejtésről Nemetz–Vajda [17].

3. Példa. A múlt század rejtjelzési algoritmusa; a Vigenére eljárás. Az egyszerű helyettesítés fejthetőségének egyik oka az, hogy mindig ugyanazt az egyetlen helyettesítő kódot alkalmazzák. Ennek megszüntetésére javasolta Vigenére azt, hogy pozíciónként változzon meg az alkalmazott egyszerű helyettesítés. A kor technikai szintjének megfelelően az alkalmazott helyettesítések ciklikusan megismétlődtek, nem túl hosszú periódussal. A könnyű memórizálhatóság miatt valamennyi helyettesítés Caesar típusú volt, ahol az eltolás mértékét az ábécé valamelyik betűje adta meg. Ha a periodúsban szereplő „vezérlő" betűk értelmes szöveget képeztek, akkor annak megjegyzése is egyszerű volt. Egy szokványos magyar megvalósítás

segítségével mutatjuk be magát az eljárást. Állapodjunk meg abban, hogy a dokumentumban nem teszünk különbséget a kis és nagy betűk között, csak betűket használunk (számokat és írásjeleket nem). A hosszú magánhangzókat rövid párjukkal helyettesítjük, viszont a szóközt mindig jelöljük (erre most a láthatóság kedvéért a + jelet alkalmazzuk). A magyar rejtjelzési gyakorlatban ez az átírás nagyon sokáig uralkodó volt. Ekkor a dokumentum által használt 31 betűs (input) ábécé a hagyományos alfabetikus sorrendbe írva:

AÁBCD EÉFGH IJKLM NOÖPQ RSTUÜ VWXYZ+

Jelölje o(betű) a betű sorszámát az input ábécében, a számozást 0-tól kezdve. Állapodjunk meg a „VIGENÉRE" jelszóban. Mivel $o(V) = 24$, $o(I) = 10$, $o(G) = 8$, $\ldots$, $o(E) = 5$, így a kulcs a

$$k_1 = 24,\ k_2 = 10,\ k_3 = 8,\ k_4 = 5,\ k_5 = 15,\ k_6 = 6,\ k_7 = 20,\ k_8 = 5,\ k_{i+8} = k_i$$

sorozat lesz. Ha a dokumentum i-edik betűjét d_i jelöli, akkor a rejtjeles szöveg i-edik betűje

$$r_i \equiv k_i + d_i \pmod{31}$$

lesz. Természetesen ez a rejtjelző eljárás is megfejthető. Megfejtéséhez először a periódus hosszát kell megállapítani. A periódus ismeretében össze lehet gyűjteni az azonos Caesarral rejtjelzett pozíciók betűit, és ezeken belül a betűgyakoriságok alapján azonosítható az eltolás nagysága. Ma a periódushossz meghatározását akár teljes kipróbálással is megkísérelhetjük. A kor akkori szinvonalán azonban ez elképzelhetetlen volt. Működött viszont egy *egyszerű statisztikai teszt, a koincidencia teszt.*

A *koincidencia teszt* hatékony eljárás annak megállapítására, hogy két dokumentum ugyanazzal a betűnkénti eljárással lett-e rejtjelezve, még akkor is, ha magát a rejtjelzési algoritmust nem ismerjük. Alapötlete is rendkívül kézenfekvő: a két egymás alá írt rejtjeles szövegben akkor és csak akkor áll egymás alatt azonos betű, ha a megfelelő nyílt betűk is megegyeztek (koincidencia történt). A dokumentumok azonos sorszámú betűinek egybeesésének valószínűségét jól lehet becsülni azon reális feltétel mellett, hogy a két dokumentum független.

Tegyük fel, hogy az ábécének r betűje van, és ezek valószínűségei rendre $P_0, P_1, \ldots, P_{r-1}$. Ekkor az egybeesés valószínűsége

$$\chi_{\text{nyelv}} = \sum_{i=0}^{r-1} P_i^2,$$

ami az adott nyelvre jellemző mennyiség („nyelvállandó").

Ha a két kulcs is független egymástól, akkor az egybeesés az egyenleteshez közeli valószínűséggel következik be. Ha Vigenére eljárás kulcsainak periódusa különbözik, akkor az egybeesés valószínűségét az eredeti eloszlás különböző ciklikus eltolásokhoz tartozó keveréke adja meg. Ez a valószínűség a *Jensen-egyenlőtlenség*

alapján mindig kisebb lesz a χ_{nyelv} nyelvállandónál. A nagy számok törvénye szerint ezért hosszabb dokumentumok esetén azonos kulcs magasabb koincidencia gyakoriságot eredményez. A teszt alkalmazása során a rejtjeles szöveget önmagához képest R betűvel eltoljuk, ahol R rendre az $1, 2, \ldots,$ (nem túl nagy) egész értékeken fut át, és minden eltolás esetén megszámoljuk a koincidenciák gyakoriságát. A nagy gyakoriságot mutató eltolások közös osztói közül kerül ki a valódi periódus. A fejtési eljárást szemlélteti a Springer Newsletter egyik számában közölt titkos szöveg megfejtése, Nemetz [18].

4. Példa. Egy elméletileg fejthetetlen rejtjelzés; végtelen átkulcsolás (one time pad). A Vigenére eljárás mindenekelőtt azért fejthető meg, mert az alkalmazott helyettesítés ciklikusan ismétlődik. Ezt megakadályozandó kézenfekvő olyan kulcssorozatot alkalmazni, amely soha nem ismétlődik meg, végtelen. (Persze a végtelen ebben az esetben mindössze annyit jelent, hogy legalább olyan hosszú, mint a dokumentum.) Egy másik célszerű elvárás az, hogy *a rejtjeles szöveg a kulcs ismeretének hiányában semmilyen információt ne tartalmazzon a dokumentumra vonatkozóan.* Ez biztosítható, ha úgy tudjuk *a kulcsot megválasztani, hogy bármilyen rögzített dokumentum esetén a rejtjeles szöveg tőle független, egyenletes eloszlású sorozat legyen.* A Vigenére eljárás sugall egy ilyen kulcsválasztást.

Legyenek a kulcssorozat karakterei egymástól (teljesen) független, egyenletes eloszlású változók az input ábécé felett. Legyen az átkulcsolás ugyanaz a Caesar eljárás, mint a Vigenére eljárásé. Ekkor egyetemi gyakorlat szintjén bizonyítható a következő tétel:

2. TÉTEL. *Jelölje D_i, K_i, R_i a dokumentum, a kulcssorozat, a rejtjeles szöveg i-edik betűjét, mint valószínűségi változót, d_i, k_i, r_i e változók konkrét értékeit. Tegyük fel, hogy a $P(K_i = k_i)$ valószínűségek egyenlők és a $\{K_i\}$ valószínűségi változók teljesen függetlenek. Ekkor*

$$P\{R_{i1} = r_{i1}, \ldots, R_{ik} = r_{ik} \mid \text{Dokumentum} = \text{bármi}\} = \{1/\text{ábécé méret}\}^k,$$

$$P\{D_{i1} = d_{i1}, \ldots, D_{ik} = d_{ik} \mid \text{Rejtjeles szöveg} = \text{adott}\} = \\ = P\{D_{i1} = d_{i1}, \ldots, D_{ik} = d_{ik}\}.$$

Ez a tétel bizonyítja, hogy egy végtelen átkulcsolással rejtjelzett dokumentum elméletileg is fejthetetlen. A technikai kivitelezés a modulo összeadástól eltérő összeadást is lehetővé tesz. Átírhatjuk például az ábécét bináris blokkokká. Ekkor a dokumentum is bináris sorozatként értelmezhető. Ilyenkor a kulcssorozat is lehet bináris, és az összeadás a szokásos XOR művelet lehet. Ez a TELEX gépek világában volt egy gyakori kivitelezés. Az összeadás bármilyen *csoportművelettel* megadható. Bonyolult csoportműveletek alkalmazhatók table-look-up eljárással, ha például az összeadást *Latin-négyzetek* segítségével adjuk meg, lásd Dénes–Keedwell [3].

A gyakorlat számára nem egyetlen, hanem sok dokumentum rejtjelzéséről kell gondoskodni. Ebből a szempontból rendkívül fontos leszögezni, hogy egyetlen véletlen kulcs csak egyszer használható fel: az ismételt kulcsfelhasználást ugyanaz a

koincidencia teszt kimutatja, amelyiket az előbb ismertettünk. Ha a kulcsismétlést felismertük, akkor a két rejtjeles szöveget egymásból kivonva eltüntethetjük a kulcsot. A rejtjelesek különbsége megegyezik a nyílt dokumentumok különbségével. Már a múlt század végén ismert volt, hogy két európai nyelven írt szöveg különbségéből a két szöveg visszaállítható. Formálisan olyan eljárás alkalmazható, mint az *információelmélet szekvenciális dekódolása.* Az ismételt kulcsfelhasználást tehát el kell kerülni ahhoz, hogy a rejtjelzés valóban fejthetetlen legyen. Ezt hangsúlyozza az eljárás angol neve is: *One Time Pad.*

Természetesen a feladó és a legális vevő mindegyikének ismernie kell a kulcssorozatot. Ezért azt a partnerhez abszolút megbízható úton el kell juttatni, és abszolút megbízható módon kell őrizni, majd a felhasználás után meg kell semmisíteni. Ez egy nehezen megoldható feladat. Ezért többféle megoldást javasoltak az egyszeri használat megkerülésére. Ezek egyik legnegatívabb példája a szovjetek második világháború alatti gyakorlata: a véletlen számsort bizonyos eltolásokban újra használták. Ez a hiba azt eredményezte, hogy elméletileg is megfejthetetlennek tartott táviratuk az USA rejtjelfejtői számára olvashatókká váltak.

Rövid megjegyzésünk mutatja, hogy a szigorúan matematikai analízis, és az ennek következtében létrehozott szabályokon túl további szigorú biztonsági rendszabályok kialakítása és betartatása szükséges. Ezekkel jelen dolgozatban nem foglalkozunk. Pusztán két hivatkozást adunk meg az érdeklődők számára: Vasvári [30], MEH [10].

Egy másik megkerülési mód, hogy valódi véletlen sorozatok helyett determinisztikus, de véletlenszerűen viselkedő sorozatokat alkalmaznak. Gyors hardver megoldást tesznek lehetővé a hosszú periódust lehetővé tevő Lineáris Visszacsatolt Shift Regiszterek. A maximális periódusú regiszterek megkeresése, elemzése a *Galoa testek elméletében* való elmélyülést igényel. Vonatkozó kulcsreferencia: Golomb [6], magyar témaismertetést ad meg Nemetz–Katona [14].

Végül szeretnénk rámutatni egy triviális tényre és a belőle fakadó feladatra. Ha sikerül egy dokumentumot az eredeti hosszának egy részére egyértelműen dekódolható módon tömöríteni, akkor a felhasználandó véletlen kulcssorozat hossza is ugyanígy rövidül. Így még az alkalmazás előtt célszerű adattömörítési algoritmust alkalmazni. Az adattömörítés lehetőségei és módszerei vonatkozásában utalunk a Salomon [24] könyvre.

A nyilvános hálózatok adatbiztonságának kérdései

A nyilvános hálózatok dokumentumainak világában ugyanazokat az elvárásokat kell teljesíteni, mint amit a papír-világban megszoktunk.

Hozzáférés védelem: Az adathordozókon tárolt információkhoz, programokhoz és az ezekhez kapcsolódó eszközökhöz csak jogosult személy juthasson hozzá.

Személyazonosítás: Meg lehessen győződni arról, hogy ki küldött egy adott dokumentumot, illetve arról, hogy a szándékolt fogadófél valóban megkapta az elküldött dokumentumot.

Integritás követelménye: Meg kell találni annak a lehetőségét, hogy a címzett meggyőződhessen arról, hogy valóban az a dokumentum jutott el hozzá, amit küldeni akartak. Ugyanígy, a feladónak is biztosnak kell lennie abban, hogy senki sem piszkált bele az üzenetébe, az változatlanul jutott el a címzetthez.

Datálás: A dokumentumokat el kell tudni látni dátummal.

Aláírás: A digitális dokumentumokat is alá lehessen írni.

Hitelesítés, tanúsítás: A közjegyző szerepéhez hasonlóan valakinek hitelt érdemlően bizonyítani kell tudni, hogy egy adott entitáshoz tartozó nyilvános kulcsot valóban az használ, akinek vallja magát.

Törvény előtti elismerés: A tanúsított digitális aláírásnak a jogi következményei ugyanolyanok legyenek, mint a hagyományos aláírásnak.

Az elektronikus üzletvitelnek (elektronikus kereskedelemnek, elektronikus pénznek, elektronikusan kötött szerződéseknek, banki garanciáknak) biztonságosnak kell lennie.

A hozzáférést leginkább jelszóval szabályozzák. Elterjedőben vannak biometriás azonosítók is. Ez a témakör a jelenlegi ismertető területén kívül esik. Ugyancsak nem foglalkozunk a törvénykezési vonatkozásokkal sem. A maradék követelmények kielégítésére elsősorban a nyilvános kulcsú kriptográfia módszereit használják.

Nyilvános kulcsú rejtjelzés: Az RSA algoritmus

A potenciális felhasználók két kulcsot választanak maguknak, és a két kulcs együttesen képezi a kulcsukat. Az egyik kulcsot nyilvánosságra hozzák, a másikat szigorúan titokban tartják. A nyilvánosságra hozott kulcsot a feladó használja a rejtjelezésre, a titkos kulcsot a címzett a megfejtésre. Olyan közös rejtjelzési algoritmust használnak, amelyben a rejtjelzést a nyilvános kulcs birtokában könnyű elvégezni, de pusztán ezzel a kulccsal a dekódolás gyakorlatilag nem kivitelezhető. A magánkulcs segítségével azonban a dekódolás is gyors művelet. Ezt a filozófiát megvalósító rendszerek gyüjtőneve: nyilvános kulcsú rendszerek (public key cryptosystems).

Maga az elképzelés Hellmantól származik, de a leggyakrabban idézett mű Rivest–Shamir–Adleman [22], az első és máig az egyetlen megbízhatónak látszó technikailag kivitelezhető eljárást adja meg. Matematikai alapja a *számelmélet Fermat tétele*, és az a tény, hogy nagy számok osztóinak meghatározása rendkívül bonyolult feladat. Az első 10 év eredményeit Diffie [5] foglalja össze, amely egyben a várakozásokat és a próbálkozásokat is bemutatja. Magyar nyelvű témaismertetést ad meg Nemetz–Vajda [17].

Az RSA algoritmus a *modulo aritmetikában* az ismeretlent hatványban tartalmazó egyenletek megoldásának nagyfokú bonyolultságát használja ki, így megfelelő nagyságú modulus esetén a megoldás technikai kivitelezhetetlensége szolgáltatja a biztonságot. A „megfelelő nagyság" itt kritikus szerepet játszik: a kezdetben biztonságosnak ítélt 40 bit hosszú kulcsok helyett ma nem nevezhető biztonságosnak 1024-nél rövidebb kulcs.

A nyilvánosságra hozott kulcs egy (E, M) egészekből álló számpár. A titkosítás ezek segítségével történik. Először a dokumentum adott hosszúságú blokkjait az M modulusnál kisebb egész számmá alakítják, majd ezt a számot M modulusban felemelik az E-edik hatványra. Ez a szám, illetve ennek az átviteli csatornára elfogadható sorozattá kódolt változata lesz a titkosított üzenet. A titkos kulcs hasonlóan egy (D, M) számpár, ahol M azonos az előzővel, míg a D dekódoló exponens úgy van megválasztva, hogy a titkosított üzenetnek megfelelő modulo-M számot D-edik hatványra emelve az eredeti üzenet adódik. Megbízható algoritmushoz M-et két nagyon-nagy prímszám szorzatának, E-t véletlenszerűen választják. Megjegyezzük, hogy a rejtjelzést a (D, M) titkos kulccsal is el lehet végezni. Ekkor a megoldó kulcs a nyilvános (E, M) kulcs lesz.

A hatékony alkalmazáshoz gyorsan kell elvégezni a megkívánt műveleteket, így például a hatványozást. Ehhez a matematika különböző területéről lehet módszereket importálni. Meglepőnek látszik, de olyan absztrakt tételek is segítenek, mint a *számelmélet kínai maradéktétele.*

Nyilvános kulcsú kriptográfia alkalmazásai

Az első alkalmazások között szerepel az azonosság-igazolás, digitális aláírás és az üzenethitelesítés. A témába jó bevezetést ad Akl [1], egy összefoglaló helyzetképet pedig Simmons [27]. Ebben a pontban pusztán azt akarjuk megmutatni, hogy a nyilvános kulcsú kriptográfia valódi lehetőséget teremt a nyilvános hálózatok biztonsági feladatainak a megoldásában.

A *küldő azonosságának igazolására* szolgálhat, ha az üzenetetet a küldő saját magánkulcsával rejtjelzi. Ekkor a megoldás a nyilvános kulcsával történhet, anélkül lehetetlen. Ha tehát a nyilvános kulccsal az üzenet elolvasható, akkor valóban az küldte, aki a nyilvános kulcsot a nyilvántartásban elhelyezte. Sajnos csak erre alkalmas, arra nem, hogy azt is igazolja, hogy valóban az küldte, akinek vallja magát. Ezért erről másként kell gondoskodni. Ennek egyik módja lehet egy tanúsítvány csatolása. Ezzel a kérdéssel nem itt nem foglalkozunk.

Az is egyszerűen megoldható, hogy *az üzenetet csak a szándékolt címzett tudja elolvasni*, az, aki az adott magánkulccsal feltehetően egyedül rendelkezik, és ő ugyanaz, akinek az érdekében az adott nyilvános kulcsot a nyilvántartásban elhelyezték. Az üzenetet az ő nyilvános kulcsával rejtjelezve a megoldás a hozzá tartozó titkos kulccsal lehetséges.

A két lépés együttes alkalmazása *a két kihívást egyszerre* oldja meg. A feladó kétszer rejtjelez: egyszer a saját magánkulcsával, majd a címzett nyilvános kulcsával. A vevő is kétszer oldja meg a rejtjeles üzenetet: először a saját titkos kulcsával, majd a feladó nyilvános kulcsával.

Mindkét fél számára megnyugtató, ha egy *megbízható harmadik fél* igazolja számukra, hogy a nyilvántartásban elhelyezett nyilvános kulcsok valóban azoktól származnak, akiknek vallják magukat. Erre a digitális világban igazolásukért pénzt kérő és anyagi felelősséget vállaló *Hitelesítő Központok* (angolul: Certification Authorities, *CA*-k) szakosodnak. Nekik is van nyilvános-titkos kulcspárjuk, amiket az

elektronikus igazolás kiadására lehet alkalmazni. Az általuk kibocsátott tanúsítvány (Certificate) tartalmazza az adott entitáshoz tartozó nyilvános kulcsot, az entitás nevét (személyazonosítóját), az érvényesség (lejárat) idejét. Ezt írja alá titkos kulcsával a Hitelesítő Központ, s ezzel az adott entitás és a nyilvános kulcs összetartozását mindenki számára ellenőrizhető módon hitelesíti.

Az üzenethez a feladó hozzárendhet egy ún. *digitális aláírást,* amely tartalmazza az üzenet egyirányú képét (lenyomatát), s egyéb adatokat. Az aláírás jellemző a létrehozójára és az üzenetre is, ellenőrizni viszont bárki tudja, aki a megfelelő infrastruktúrához hozzáfér. Az aláírás egyben lehetőséget ad az üzenet sértetlenségének ellenőrzésére is. A digitális aláírás két részből áll: a személyhez kötött részből, s az ellenőrzést bárki számára lehetővé tevő részből.

A nyilvános kulcsú rejtjelzés időigényes, ezért nem célszerű nagy dokumentumokra alkalmazni. E helyett a dokumentumnak egy alkalmas *sűrítményét* készítjük el, és erre alkalmazzuk a nyilvános kulcsú módszereket. Az alkalmas sűrítmények elkészítésére szolgálnak a Hash eljárások.

Hash eljárások

A digitális aláírás, hitelesítés során a hitelesítendő hosszú dokumentumokhoz egy rövid sűrítményt akarunk hozzárendelni úgy, hogy különböző dokumentumokhoz különböző sűrítmények tartozzanak, miközben a sűrítmények előállítása könnyű feladat. Ezeket a sűrítményeket akarjuk azonosításra használni. Azt a leképezést, amely ezt a feladatot megoldja, hash függvénynek nevezik.

A Hash algoritmus lényegében véve egy olyan transzformáció, amely egy tetszőleges hosszú szöveg digitális sűrítményét készíti el. Az elkészítés alatt álló MSZ ISO/Iec 11770 szabványsorozatban ezt a sűrítményt lenyomatnak nevezik.A lenyomat fix hosszú bitsorozat, amely jellemző az adott szövegre. Angolul *message digestnek* is nevezik. Ez *a Digitális aláírás protokoll*nak szerves alkotórésze. Alkalmazása során a dokumentumot adott hosszuságú blokkokra osztják fel, ezek mindegyikére alkalmaznak egy tömörítő, ismét adott hosszuságú Hash függvényt. Az eredményeket egymás után írva új (átmeneti) dokumentum keletkezik. Erre iterálva alkalmazzák a Hash függvényeket, mindaddig, amíg csak egyetlen blokknyi dokumentum marad. Ennek kimenetele lesz az eredeti dokumentum lenyomata. A szabvánnyal összhangban van a következő angol-magyar kifejezés gyűjtemény:

Hash	lenyomat
hash algorithm	lenyomatképző algoritmus
hashed octets	lenyomat oktettek
hash function	lenyomatképző függvény
hash results	lenyomatképzés eredménye
hash value	lenyomati érték
generate to	előállít, kelt

A Hash függvényeket eredetileg adattárolásra és adatbankokban való keresésre dolgozták ki. Lényegében az ott elért eredményeket használják fel a kriptográfiá-

ban is, ahol azonban további feltételekre is figyelemmel kell lenni. Az elvárásokat a következő követelmények összegezik:

1. Gyakorlatilag lehetetlen egy adott lenyomathoz olyan dokumentumot konstruálni, amelyikhez a hash függvény ugyanazt az lenyomatot rendeli.
2. Gyakorlatilag lehetetlen két olyan dokumentumot konstruálni, amelyek azonos hash értéket eredményeznek.
3. Ha legalább egy bitet egy dokumentumban megváltoztatunk, akkor a megfelelő hash értékek több bitben különböznek.
4. A Hash algoritmusok számos alkalmazásában fontos, hogy a lenyomat véletlenszerűen viselkedjen.

A negyedik tulajdonság egy speciális esetben triviálisan biztosítható:

3. TÉTEL. *Legyenek* $X_1, X_2, \ldots, X_n$ *független, azonos eloszlású valószínűségi változók, amelyeknek az értékei* $0, 1, \ldots, m-1$, *ahol* 0 *és* 1 *valószínűsége pozitív. Akkor az*

$$S(n) = X_1 + X_2 + \ldots + X_n \bmod m$$

részletösszeg határértékben egyenletes eloszlású.

A tétel ebben a formájában ma már egyetemi gyakorlat szintjén áll. Számunkra lényegesebb ennek egy enyhébb változata, amelyben a függetlenséget ergodicitás, az azonos eloszlást stacionaritás helyettesíti. Az írott nyelvekről feltételezik, hogy megfelelnek ezeknek a követelményeknek, így a tétel lehetőséget teremt egyetlen, m értékű „sűrítmény" létrehozására.

SHA-szabvány

A nyilvános kulcsú kriptográfiában (ideértve a hitelesítést) több Hash-eljárást publikáltak.

Ezek közül a leggyakrabban alkalmazott az amerikai szabvány, a Standard Hash Algoritmus, SHA. Az algoritmus inputja egy tetszőleges hosszúságú (maximum 2^64 bit) tetszőleges dokumentum, az outputja pedig egy 160 bit hosszúságú füzér (hash érték = message digest).

Az SHA függvényt az amerikai *Federal Information Processing Standard* sorozat 180-as számú dokumentuma szabványosítja. A szabványon 1996-ban apróbb változásokat hajtottak végre, az új változatot SHA-1 jelöli. Az új szabvány a *FIPS PUB 180-1* jelet kapta. Az algoritmus számítástechnikailag sokféle módon valósítható meg, amelyeknek azonban ugyanazon bementi sorozat esetén ugyanazt a lenyomatot kell eredményezni.

Ahogyan az előző tétel szerint a részletösszegek véletlenszerűen viselkednek, ugyanúgy most is elegendő hosszúságú bemeneti dokumentum esetén a lenyomat véletlenszerűen viselkedik. 5-6 darab 32 bites szóból álló bemenet már „elegendően" hosszúnak tekinthető. Ezt pusztán empirikusan sikerült megmutatni, formális bizonyításról a szerzőnek nincs tudomása. A nehézséget itt a képlet bonyolultsága jelenti. A képletet most megadjuk, már csak azért is, mert maga az algoritmus

olyan elemekből építkezik, amelyek egyre gyakoribbak a rejtjelző algoritmusoknál is. Az ismertetés során a következő terminológiát alkalmazzuk.

- **Füzér:** bitekből álló véges sorozat.
- **Hexadecimális jegy** (digit): A $0, 1, \ldots, 9, a, b, c, d, e, f$ jelek bármelyike 4-bites sorozatok megkülönböztetésére szolgál, mégpedig azok alfabetikus sorrendjében (tehát $0 \to 0000$, $f \to 1111$).
- **Szó:** 32-bites füzér. Szokásos látható tömörített leírása 8 hexadecimális jegygyel történik. Példa: A103FE23 = 1010 0001 0000 0011 1111 1110 0010 0011.
- (Előjel nélküli) **egész szám:** 0 és 232-1 közti egész szám, amely egyetlen szóval megadható. Például a hexadecimális 103fe23 sorozat a decimális formájú 2701393443 egész számot jelöli.
- **Blokk:** 512 bit hosszúságú füzér. Egy blokkot 16 szó képezhet.

Az SHA által szavakkal végzett **műveletek:**

- AND = bitenkénti logikai és
- OR = bitenkénti logikai inkluziv vagy
- XOR = bitenkénti logikai exkluziv vagy
- $\sim x$ = az x bitenkénti komplementere
- **Összeadás:** Az A és B szavak $A + B$ összege: Legyen az A és B szavaknak megfelelő előjel nélküli egész szám x és y, ahol tehát $0 \leq x < 232$. Kiszámítjuk a $z = (x + y) \bmod 232$ egészet. Legyen a z-nek megfelelő szó C. Ez lesz az $A + B$ összeg.
- **Ciklikus eltolás:** Adott X szó és adott $0 \leq n < 32$ egész szám mellett $S(n, X)$ jelöli az X szó jegyeinek balra n jeggyel történő ciklikus eltolását.
- **Feltöltés:** A sűrítendő füzéret kiegészítjük 1 darab „1" bittel. Így a sorozat L hossza legfeljebb 264 bit lesz. Ha az új sorozat $L + 1$ hossza nem kongruens 448-cal moduló 512, akkor annyi „0" bittel egészítjük ki, hogy ilyen legyen. Ekkor a kiegészített füzér valamilyen nemnegatív s egészre $16s + 14$ szót tartalmaz. Ezt még két szóval egészítjük ki: Felírjuk az eredeti L hosszat mint két szót, ahol az első szó tartalmazza az értékesebb (nagyobb helyiértékű) biteket, és ezt a két szót hozzáfűzzük a már meghosszabbított sorozathoz. Az így *előkészített sorozat mindig* 16*-tal osztható számú szót tartalmaz.* Ezek 16-16 szavanként rendre egy-egy blokkot alkotnak. A blokkok jelölésére M(sorszám) használatos.

Az SHA algoritmus *80 függvényt* használ, amelyeknek a bemenetele 3 darab 32 bites szó. Az $f(0, x, y, z), \ldots, f(79, x, y, z)$ kimenetelek is 32 bites szavak. Az $f(t, x, y, z)$ függvények formális definíciója:

$$\begin{aligned}
f(t, x, y, z) &= (x \text{ AND } y) \text{ OR } (\sim x \text{ AND } z) && (0 \leq t \leq 19) \\
f(t, x, y, z) &= x \text{ XOR } y \text{ XOR } z && (20 \leq t \leq 39) \\
f(t, x, y, z) &= (x \text{ AND } y) \text{ OR } (x \text{ AND } z) \text{ OR } (y \text{ AND } z) && (40 \leq t \leq 59) \\
f(t, x, y, z) &= x \text{ XOR } y \text{ XOR } z && (60 \leq t \leq 79).
\end{aligned}$$

Az algoritmus ugyancsak *80 konstanst* használ. Ezek hexadecimális formában felírva:

$$\begin{aligned} K(t) &= 5a827999 \qquad (0 \le t \le 19) \\ K(t) &= 6ed9eba1 \qquad (20 \le t \le 39) \\ K(t) &= 8f1bbcdc \qquad (40 \le t \le 59) \\ K(t) &= ca62c1d6 \qquad (60 \le t \le 79). \end{aligned}$$

Az SHA algoritmus a lenyomatot a kiegészített dokumentum (füzér) blokkjain blokkról blokkra haladva végzi. Minden blokkon ugyanazt a 80 lépést ismételi meg. Leírásához segédtárolókat is használunk. Ezek:

- Egy ötszavas (5...32 bites) puffer. A szavakat A, B, C, D, E jelöli.
- Egy másik ötszavas puffer. Ennek a szavait $h0$, $h1$, $h2$, $h3$, $h4$ jelöli.
- 80 darab szóból álló sorozat: $W(0), W(1), \ldots, W(79)$.
- Egy kitüntetett TEMP nevű szó.

Az algoritmus a $\{hj\}$ *puffer inicializálásával* kezdődik:

$$\begin{aligned} h0 &= 67452301 \qquad h1 = efcdab89 \qquad h2 = 98badcfe \\ h3 &= 10325476 \qquad h4 = c3d2e1f0 \end{aligned}$$

Valamennyi blokk feldolgozása a W(sorszám) szavak inicializálásával kezdődik. A soron következő feldolgozandó blokk tartalmát 16 szóba töltjük át. Ezek $W(0), W(1), \ldots, W(15)$, ahol $W(0)$ a baloldali első szót (32 bitet) jelöli. Ezek segítségével adjuk meg a többi W(sorszám) szó értékét:

$$W(t) = W(t-3) \text{ XOR } W(t-8) \text{ XOR } W(t-14) \text{ XOR } W(t-16), \qquad 15 < t < 80$$

Ugyancsak felfrissítjük az első puffert is:

$$A = h0, \quad B = h1, \quad C = h2, \quad D = h3, \quad E = h4.$$

Az egyes blokkokon 80 lépéses ciklusban módosítjuk az első puffert:

For $t = 0$ to 79 do
TEMP $= S(5, A) + f(t, B, C, D) + E + W(t) + K(t)$;
$E = D$; $D = C$; $C = S(30, B)$; $B = A$; $A =$ TEMP;

A ciklus után módosítjuk a második puffert is:

$$h0 = h0 + A, \quad h1 = h1 + B, \quad h2 = h2 + C, \quad h3 = h3 + D, \quad h4 = h4 + E.$$

A legutolsó blokk feldolgozása után adódó puffer alkotja a 160 bites, 5 szóval megadott lenyomatot:

$$h0 \quad h1 \quad h2 \quad h3 \quad h4.$$

A vizsgálatokhoz láthatóan kirándulni kell a *Boole-függvények* világába.

CRC-32: Gyakran használják a ANSI X3.66 szabványt is, a CCITT által javasolt 32 bites CRC algoritmust. (A CRC az angol Cyclic Redundancy Check rövidítése.) Ennek Hash függvénye egy 64 elemű $\mathbf{b} = (b_0, b_1, \ldots, b_{63})$ bináris blokkhoz rendel hozzá egy 32 bitből álló $\mathbf{r} = (r_0, r_1, \ldots, r_{31})$ bináris blokkot, tehát a sorozatot a felére sűríti össze. Az algoritmus $GF(2^k)$ *Galoa testek* területéről építkezik. A Hash függvény bináris generátor függvénye:

$$\mathrm{CRC}_{32}(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + \\ + x^4 + x^2 + x^1 + 1.$$

Osszuk el a **b** dokumentum-blokkhoz készített

$$\boxed{P(x) = \sum_{i=0}^{63} B_i \cdot x^i}$$

polinomot maradékos osztással a $\mathrm{CRC}_{32}(x)$ polinommal. A maradék polinom,

$$R(x) = \sum_{i=0}^{31} r_i \cdot x^i,$$

$r_0, r_1, \ldots, r_{31}$ együtthatói adják meg a Hash függvény outputját. A CRC-vel összefüggő számítástechnikai kérdésekkel kapcsolatban utalunk a Ramabadran–Sunil [21] dolgozatra. A részletösszegekre vonatkozó tételhez hasonló állításra bizonyítást nem publikáltak, bizonyítása érdekes matematikai eredmény lehet.

A kriptográfiai protokollok területén is találhatunk több olyan problémát, amelynek kezeléséhez mélyebb matematikai ismeretek is szükségesek. A kriptográfiai protokoll kriptográfiai algoritmusokból, alapelemekből épül föl, és egy összetett feladatot hajt végre. Leggyakrabban hitelesítő és kulcs-csere protokollokat használunk. A legismertebb komplex protokoll az interneten két gép közötti bizalmasság és hitelesség biztosítására használt SSL/TLS protokoll. Manapság egyre ismertebbé válnak a különböző digitális pénzt kezelő (Ecah, Digicash, Micromint) protokollok is, bár ezek némelyike annyira összetett, hogy több részprotokollra bontható. Ezekre a séma elnevezés is használatos.

Nyilvánvaló, hogy biztonságos alapelemekből építkező protokoll is lehet hibás. A protokollok formális elemzése a kriptográfia egyik nagy aktuális kihívása. A tapasztalatok azt mutatják, hogy a „józan paraszti ész" itt nem elegendő, sokszor több éve használt, széles körben vizsgált protokollban is találnak hibát. A terület kapcsolatokat ápol a *véges állapotú automaták* elméletével.

Egy másik kapcsolódási területre adnak példát a titokmegosztási protokollok. Adi Shamir használt először véges test fölötti *Langrange-interpolációt* titokmegosztási séma létrehozásához. A titokmegosztási séma feladata legegyszerűbb formájában az, hogy egy titkot n résztvevő között felosszon úgy, hogy közülük bármely k vissza tudja állítani a titkot, de bármely $k-1$ résztvevő által együtt birtokolt adat

nem ad információt a titokról. Ha például n résztvevő között akarjuk elosztani a C titkot úgy, hogy bármely három rekonstruálni tudja azt, akkor válasszunk egy nagy p prímet, és egy másodfokú polinomot:

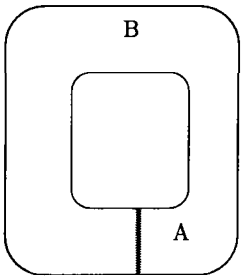
$$ax^2 + bx + C \bmod p,$$

ahol a, b tetszőlegesek, C a titok, és p olyan, hogy mindannyiuknál nagyobb. A résztvevők titokrészei (árnyék) a polinom adott pontokban (pl. 1, 2, 3) vett értékei. Nyilvánvaló, hogy bármely három résztvevő vissza tudja állítani a polinomot, s így a titkot jelentő konstans tagot is.

Ugyanerről a területről megemlíthető Blakley sémája is, ahol a titok egy pont az n-dimenziós térben, az árnyékok pedig olyan hipersíkok(egyenletei), amelyek tartalmazzák a pontot. n hipersík ismerete megadja a pontot.

Rendkívül érdekesek az úgynevezett zero-knowledge protokollok is. Az alaphelyzet az, hogy András ismer egy információt, s be akarja bizonyítani ezt Bélának úgy, hogy Béla ne ismerje meg az információ egyetlen bitjét sem. Első pillanatra az sem világos, hogy ez lehetséges.

J. J. Quisquatertől származik a következő példa. Tegyük fel, hogy András egy, a 2. ábrán látható barlang mélyén áll, és van ott egy titkos ajtó, amelynek a jelszava a szóban forgó titok.



2. ábra

Először bemegy András valamelyik irányba, majd Béla is megáll a bejáratnál, és felszólítja Andrást, hogy jöjjön elő az általa megadott irányból. Ezt sokszor ismételve, nyilván elenyésző az esélye annak, hogy András a jelszó ismerete nélkül hajtotta végre a produkciót.

A valódi zero-knowledge protokollok egy matematikai értelemben nehéz problémán alapulnak, pontosabban a rendszert úgy választjuk meg, hogy András tudása egy nehéz probléma megoldását reprezentálja. Egy lehetséges probléma a *gráfelmélethez* visz bennünket. Zero-knowledge protokollok alapjaként gyakran használják két gráf izomorfiájának megállapítását, amely NP-teljes probléma, de Hamilton-kör keresése is használható ugyanerre a célra.

Hivatkozások

[1] Akl, S. G., Digital signature: A tutorial survey, *IEEE Trans. on Computers,* **C-34** (1983), 15–24.

[2] Boda, M., Új irányzatok a távközlési kutatásban, *Akadémia,* **III/4** (1999. TÉL) (1999).

[3] Dénes, J., Keedwell, A. D., *Latin squares and their applications,* Academic Press (New York, 1974).

[4] Diffie, W., The first ten years of public-key cryptography, *Proc. of the IEEE,* **76** (1988), 560–577.

[5] Diffie, W., Hellman, M. E., New directions in cryptography, *IEEE Trans. on Info. Theory,* **IT-22** (1977), 644–654.

[6] Golomb, S. W., *Shift Register Sequences,* Holden Day (San Francisco, 1960)

[7] Hellman, M. E., An Extension of the Shannon Theory Approach to Cryptography, *IEEE-IT,* **23** (1977), 289–294.

[8] Kahn, D., *The codebreakers,* MacMillan (New York, 1967).

[9] Knuth, E. D., *A számítógépes programozás művészete 2. Szeminumerikus algoritmusok,* Műszaki Kiadó (Budapest, 1987).

[10] MEH, *Informatikai rendszerek biztonsági követelményei,* Miniszterelnőki Hivatal, Informatikai Koordinációs Iroda (1996)

[11] Merkle, R. C., Protocols for public-key cryptosystems, *Proc. of the IEEE Symp. on Security and Privacy* (Oakland, 1980).

[12] Muha L. (szerk), *Az informatikai biztonság kézikönyve,* 10. Javított kiadás, Verlag Dashöfer (Budapest, 2004).

[13] National Bureau of Standards, *Data Encryption Standard,* Washington, D.C. (1977).

[14] Nemetz, T., Katona, Gy., Néhány megjegyzés a shift regiszter generátorokról, *MTA Számítástechnikai Központ Közl.,* **5** (1969), 1–10.

[15] Nemetz, T., Permutációk szisztematikus és véletlenszerű generálása, *MTA III. Oszt. Közl.,* **XIX** (1969), 235–245.

[16] Nemetz, T., Simon, J., Hiányos szövegek rekonstrukciója és a magyar nyelv entrópiája, *Magyar Nyelv,* **LXXXV.** No. 4 (1989), 427–438.

[17] Nemetz, T., Vajda I., *Algoritmikus adatvédelem,* Akadémiai Kiadó (1991).

[18] Nemetz, T., A Springer rejtjeles levele, *Matematikai Lapok,* **3** (1991), 7–18.

[19] Nemetz, T., Milyen matematikát rejt Verne postagalambja, *Módszertani Lapok,* Matematika, **3.** évfolyam, 3. szám (1996–97).

[20] Papp, Pál, *Hitelesítés nyílt hálózatokban, avagy egy hatékony pénzkímélő rendszer,* Előadás a DAT'96 konferencián (1996).

[21] Ramabadran, T. V., Sunil, S. G., A Tutorial on CRC Computations, *IEEE Micro,* August (1988), 62–75.

[22] Rivest, R. L., Shamir, A., Adleman, L., A method for obtaining digital signatures and public-key cryptosystems, *Comm ACM,* **21** (1978), 120–126.

[23] Révay Zoltán, *Titkosírások,* Zrínyi Katonai Kiadó (Budapest, 1978).

[24] Salomon, D., *Data Compression,* Springer (1997)

[25] Shannon, C. E., Communication Theory of Secrecy Systems, *Bell Syst. Techn. J.*, **28** (1949), 656–715.

[26] Shannon, C. E., Prediction and Entropy of Printed English, *Bell Syst. Techn. J.*, **30** (1951), 50–64.

[27] Simmons, G. J., A survey of information authentication, *Proc. of the IEEE*, **76** (1988), 603–620.

[28] Storer, J. A., Sszymanski, A., Data Compression via Textual Substitution, *Journal of the ACM*, **29** (1982), 928–951.

[29] Svékus Olivér, *Titkosírások*, Móra Köyvkiadó (Budapest, 1964).

[30] Vasvári György, *Biztonsági rendszerek szervezése*, PRO-SEC Kft. (Budapest, 1997).

MATHEMATICS IN CRYPTOGRAPHY

TIBOR NEMETZ

Alkalmazott Matematikai Lapok **23** (2006), 313–333.

KRIPTOGRÁFIAI CÉLÚ VÉLETLENSZÁM GENERÁLÁS ÉS ELLENŐRZÉS

BORBÁS GERGELY, NEMETZ TIBOR, PAPP PÁL

1. Bevezetés

Üzenetek titkosításához egyértelműen dekódolható kódok egy halmazát, a kódhalmazt szokták megválasztani. Kerckhoff 19. századi munkássága óta ezt a halmazt ismertnek tételezik fel a rejtjelfejtés számára. Szimmetrikus kulcsú rendszerek esetében egyetlen titokban tartandó elem az, hogy a sok kód közül melyiket használjuk egy konkrét üzenet titkosítására. A felhasznált kód megadására annak egy azonosítóját használják fel. Ezt az azonosítót **az üzenet** (a kiválasztott kód) **kulcsának** nevezik. A kulcsot az üzenet továbbítója és címzettje az üzenetváltás előtt úgy egyeztetik egymással, hogy egy harmadik személy a kiválasztott kulcsról semmilyen információt ne tudjon szerezni.

Egy harmadik személy számára tehát a kulcs választása *maximális bizonytalanság*ot kell, hogy tartalmazzon. Ezért az aktuális **kulcsot véletlenszerűen kell kiválasztani** az összes lehetséges kulcs halmazából, a **kulcstérből**. Nem jelent semmilyen megszorítást, ha a kulcsteret egymás után következő, nem-negatív egész számok halmazának tekintjük. Ennek a véges halmaznak kell egy elemét véletlenül (véletlenszerűen) kiválasztani.

Véletlenszerűen választott számokat a mindennapi élet, a számítástechnika más területein is régóta alkalmaznak, utalunk Knuth (1987) számítógépes bibliájára. Vannak olyan területek is, ahol véletlenszámok nélkül az alkalmazás hatékonyságáról nem is lehetne beszélni. A véletlen kulcsválasztás fontossá válása új lendületet adott a véletlenszám-generálásnak, új utakat nyitott meg.

A kriptográfiában véges sok értéket felvevő egyenletes eloszlású véletlenszámokkal dolgoznak. A felhasználandó véletlenszámok legalapvetőbb tulajdonsága a kiismerhetetlenség. Ez azt jelenti, hogy hiába ismerjük meg pontosan a korábban

már használt véletlenszámokat, a pillanatnyilag előállított (előállítandó) véletlenszámokról semmit sem tudunk, azok maximális bizonytalanságot tartalmaznak.

Heurisztikusan ezt úgy fogalmazhatjuk meg, hogy a változók a lehetséges értékek bármelyikét felvehetik, mégpedig egyenlő valószínűséggel (ezért nevezzük őket **egyenletes**nek), és teljesen függetlenül attól, hogy korábban milyen értékek jelentek meg (ezért hívjuk őket **független**nek). Etalonként gondolhatunk egy szabályos érme egymás utáni dobálása során nyert fej-írás sorozatokra (amelyekből a szokásos átírás után keletkezhetnek a **0–1 értékű bináris sorozatok**), vagy egy szabályos kocka dobásával nyert $1, 2, \ldots, 6$ értékeket felvevő sorozatokra. Ezek a sorozatok egy fizikai jelenség egymás utáni végrehajtása során adódnak, innen származik az elnevezésük is: **fizikailag generált** vagy **valódi véletlenszámok**. A fogalmat a következő definíció teszi matematikailag precízzé:

Definíció. Legyen $X = \{0, 1, \ldots, k-1\}$ az $X_1, X_2, \ldots, X_n$ valószínűségi változók közös értékkészlete. A változókat *teljesen független* (röviden független) *egyenletes eloszlású valószínűségi változó sorozatnak*, röviden **véletlennek** nevezzük, ha a lehetséges értékek bármilyen $x_1, x_2, \ldots, x_n$ választása mellett fennáll a

$$\mathbf{Prob}\{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\} = (1/k)^n$$

azonosság.

A kulcsválasztás feladatának megoldásához viszonylag kisszámú véletlenszám elegendő volt. Az elméletileg is fejthetetlen „one time pad" rejtjelző algoritmus (lásd később) azonban ugyanolyan hosszú véletlenbetű sorozatot igényelt, mint maga a nyílt szöveg (a titkosítandó üzenet), tehát általában nagyon hosszút. Ezt még ráadásul rendkívül biztonságosan szállítani és őrizni is kellett. Így természetes, hogy a felhasználók „takarékosan" igyekeztek felhasználni a véletlen betűket. Ezért az egy üzenet titkosításához már felhasznált véletlen sorozatot kisebb-nagyobb eltolásokkal ismételten felhasználták. Az ismételt kulcsfelhasználás azonban betűnkénti helyettesítés esetén már az I. Világháború idején is felismerhető volt a koincidencia teszt segítségével, lásd Friedman (1920). Kulcsismétléses táviratok fejtésére viszont még korábbról létezett megoldás, a „kerchkoffozás", lásd Kerchkoff (1883).

Az ismételt felhasználás a Szovjetunió esetében súlyos következményekkel járt. Már a II. világháború alatt az USA egy erre a célra betanított csapata folyamatosan fejtette a szövetséges Szovjetunió legtitkosabb üzeneteit. Ez a tény ugyan csak a 90-es években vált publikussá, de szakberkekben ismertté vált a veszély. Így az ismételt felhasználást más takarékos módszerrel igyekeztek kiváltani. Ennek a módszernek a lényege az volt, hogy a valódi véletlen sorozatokat olyan algoritmikusan generálható sorozatokkal helyettesítettek, amelyek „úgy viselkedtek, mintha véletlenek lennének." Ilyen sorozatok iránti igény találkozott a lényegében ugyanebben az időben keletkezett „polgári" igényekkel.

A számítógépek hőskorában rendkívül fontos számítástechnikai követelmény volt, hogy véletlenszerű számsorozatokat ismételten elő lehessen állítani, reprodukálni. A véletlen számsorozatok szerepében használt reprodukálható sorozatoktól

azt várták el, hogy *statisztikailag ugyanúgy viselkedjenek, mintha valódi véletlen sorozatok lennének* Ezért helyettük determinisztikus számsorozatot állítottak elő. Történelmileg először az atombomba gyártásánál merült fel nagyon nagy mennyiségű, reprodukálható véletlenszám előállításának az igénye. A magyar származású Neumann János 1944-ben azt javasolta, hogy erre a célra álvéletlen számokat használjanak.

Az algoritmikusan előállítható, véletlenszerűen viselkedő **számsorozatokat *álvéletlen* (pszeudovéletlen) sorozatoknak** nevezik. A számítógépek jelenlegi gyakorlata ilyen véletlenszámokkal dolgozik, csak imitálja a valódi véletlenszámok generálását, lásd Knuth (1987).

Definíció. Az $y_1, y_2, \ldots, y_n$ számsorozatot determinisztikusnak nevezzük, ha a sorozat bármely tagja meghatározható a megelőzők determinisztikus függvényeként, tehát létezik olyan f_n függvénysorozat, melyre

$$y_n = f_n(y_1, y_2, \ldots, y_{n-1}).$$

A véletlenszámok gyakorlatában (és irodalmában) nem szokták matematikai precízséggel definiálni az álvéletlen fogalmát. Mi ezt később, a statisztikai tesztek kapcsán fogjuk megtenni.

A kriptográfiai alkalmazásokban a tökéletes titkosságot csak **fizikai forrás alapján** generált véletlenszámok lehet biztosítani. Ennek megfelelően a nyilvános kulcsú kriptográfia terjedésével párhuzamosan a piacon az utóbbi évtizedekben számos fizikai generátor jelent meg. Ugyanakkor megnőtt az igény megbízható kész sorozatok iránt is. Egy ilyen helyzet mindig kompromisszumot eredményez, és ez most is így van. A hőskori táblázatok mintájára a fizikai forrás alapján generált sorozatokat **CD-ROM-ra** (DVD-re, vagy bármilyen más, nagy kapacitású adathordozóra) **írva** is be lehet szerezni, miközben *minden CD-ROM egy-egy új táblázatot jeleníthet meg.* Az ilyen típusú felhasználások vezérfonala az, hogy nagy mennyiségű valódi véletlenszámot biztonságos körülmények között generálnak, azokkal egy CD-ROM-ot teleírva tárolják őket, majd „jól összekutyult", *egyszeri kivételi algoritmussal* akkora részt vesznek ki belőle, amekkorára éppen szükség van, lásd Nemetz–Papp (1998).

Ahogyan arra már fentebb utaltunk, meg kell fogalmazni, mit jelent a kriptográfia számára az a kifejezés, hogy *véletlenszerűen viselkedik* egy számsorozat. Ehhez összegyűjtötték azokat a lényeges tulajdonságokat, amelyekkel a valódi véletlen sorozatok rendelkeznek. Ezeket a tulajdonságokat csoportosították, és meglétük ellenőrzésére a sorozatoktól függő függvényeket, „statisztikákat" határoztak meg. A statisztikák értékeiről kiszámították, milyen eloszlás szerint viselkednek, ha a sorozat valóban véletlenszerű. Ezt használják fel a véletlenszerűség ellenőrzésére, lásd **„Statisztikai próbák a kriptográfiai alkalmazásoknál használt véletlen- és pszeudo-véletlen szám generátorokra"**, NIST: Special Publication 800-22 (2001).

Definíció. Egy determinisztikus sorozatot relatív pszeudo véletlen sorozatnak tekintünk, ha kielégíti egy adott, a véletlen sorozatokra definiált statisztikai próbák rendszerét a véletlen sorozatoktól megkívánt szinten.

Meg kívánjuk jegyezni, hogy a pszeudo-véletlen fogalom valamennyi értelmes felhasználásban kimondva-kimondatlanul szerepel a háttérben egy ilyen statisztikai teszt-csomag.

Tanulmányunk következő fejezeteiben továbbra is a kriptográfiai igényekhez igazodva ismertetjük a véletlenszámok előállításának és ellenőrzésének módjait. Mindezt kiegészítjük **két új véletlenszám generátorral**, és egy olyan **programcsomaggal**, amely az USA szabványoknak megfelelően a felhasználásra számot tartó **sorozatok véletlenszerűségét ellenőrizni tudja.**

Mindhárom megoldás *jogtiszta alkalmazást* tesz lehetővé, megszabadítva az alkalmazót az Internetről letölthető, vagy egyéb úton beszerezhető megoldásokban esetleges rejlő tulajdonjogi buktatóktól.

2. Véletlenszámok generálása: helyzetkép

Felidézzük, hogy a kriptográfiában véletlenszámok sorozata alatt heurisztikusan a következőt szokás érteni. Valamilyen k egész szám mellett tekintsük a $\{0, 1, \ldots, k-1\}$ értékkészletet. Tekintsük ebbe a halmazba tartozó számok olyan sorozatát, amelyeknek mindegyike lényegében ugyanannyiszor fordul elő a sorozatban, mégpedig bármelyik helyen bármelyik szám **egyenlő eséllyel** fordul elő (*egyenletesség*), **függetlenül attól**, hogy mi volt előtte, vagy mik jönnek utána (függetlenség). Az értékkészlet k számosságának szokásos megválasztása $k = 2$, amikor **bináris véletlenszámok**ról beszélünk. Ezek központi jelentőségét az információelméletben figyelhetjük meg. Ugyancsak gyakori a $k = 10$ választás, amikor **decimális véletlenszámok**ról beszélünk. A $\{0, 1, \ldots, k-1\}$ értékkészlet helyett egy nyelv ábécé-jét is lehet értékkészletnek tekinteni. Ekkor is véletlenszámokról, vagy véletlenbetűkről beszélnek.

Véletlenszámok manuális előállításához kockát, pénzérmét, kártyát, sorsolást szoktak használni. Ez azonban egy rendkívül lassú folyamat. Ezért keresnek más lehetőségeket is. Századunk első felében a statisztikai következtetésekhez a véletlenszámokat manuálisan állították elő és táblázatolták. Az első „nagy" táblázatot az angol Tippet készítette 1927-ben: 40 ezer véletlen számjegyet állított elő népszámlálási adatokból. 1939-ben 100 000 véletlen számjegyet állítottak elő egy erre a célra épített célgéppel. 1955-ben a RAND CORPORATION 1 millió véletlen számjegyet publikált.

2.1. Véletlenszámok fizikai generálása

Kriptográfiai kulcsok generálásakor a pszeudó-véletlenszám generátorok előnyei rendkívüli hátrányokká válnak. Ilyenkor különösen fontos, hogy

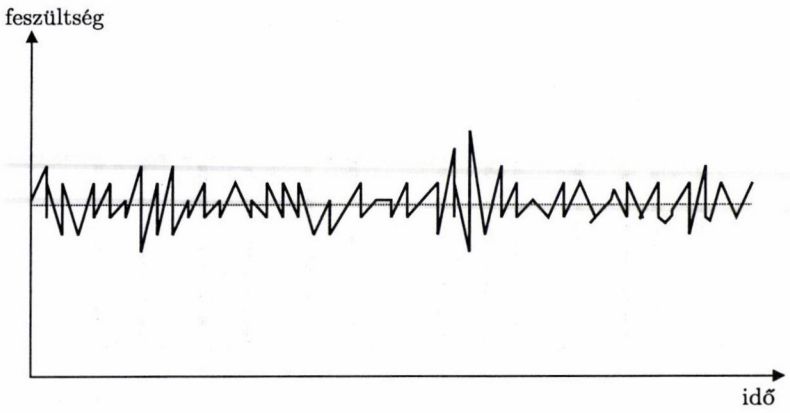
- az elkészült sorozat ne legyen reprodukálható,

- „információtartalma" megegyezzen a méretével.

Fentiek miatt pusztán fizikai véletlenszám generátorokat használnak a rejtjelkulcsok készítésére. ***A fizikai véletlenszám generátorok alapjául véletlen jeleket adó természeti jelenség szolgál.***

A gyakorlatban legtöbbször valamilyen rádiótechnikai eszköz a jelforrás. Segítségével építhető olyan számítógépes bővítőkártya, amelyben a fizikai folyamatot erősítik, mintavételezik, és amely a kapott digitális jeleket átadja egy számítógépnek. A generált sorozatot folyamatosan ellenőrizni kell, mert a fizikai folyamatot befolyásolja a környezete, amely enyhébb esetben az eloszlás megváltozását, súlyosabb esetben degenerációt okoz.

A következőkben leírjuk egy PC-be illeszthető (vagy azzal más módon – pl. USB – összekapcsolható) fizikai generátorkártya felépítését. Ennek a fizikai véletlenszám generátornak az alapja, a zajforrás egy úgynevezett Zener dióda. Az ezen átfolyó áram feszültsége alapvető fizikai törvényszerűségek miatt kismértékben ugyan, de véletlenszerűen ingadozik. Az ingadozás „frekvenciája" 0 és néhány száz (200) kHz között változik (1. ábra).



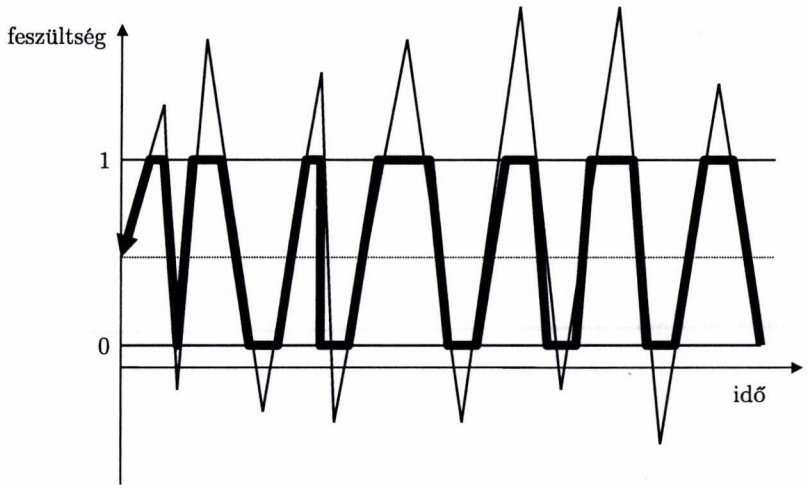
1. ábra

A kártyára épített elektronika ezt az ingadozást több fokozatban erősíti, majd a jel „közepén" szimmetrikusan kijelöl egy intervallumot, amelynek két szélső értéke lesz a 0 és az 1 érték (2. ábra).

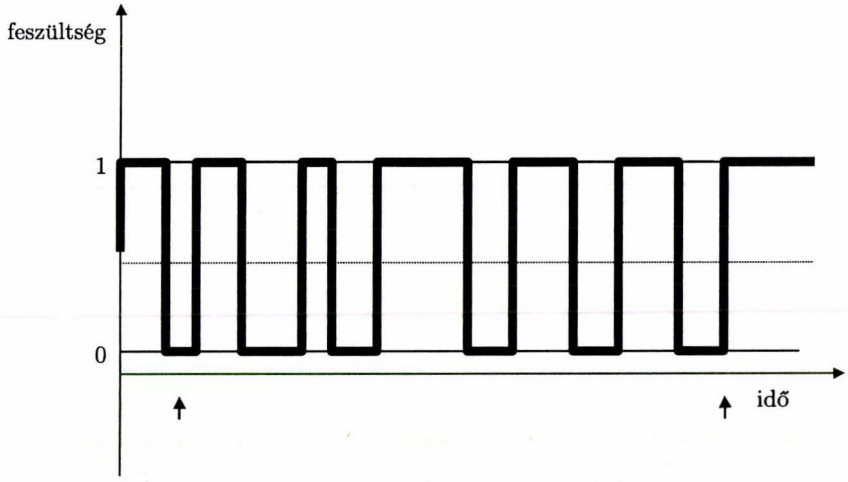
A jelet ezután négyszögesítjük. Ezután kerül sor a mintavételezésre, amelynek frekvenciája néhány ezer bit másodpercenként, vagyis a mintavételezés frekvenciája nagyságrendekkel kisebb a kiinduló jel frekvenciájánál, ami a szomszédos minták függetlenségét biztosítja (3. ábra).

A generátorkártyán két független jelforrás és mintavevő rendszer működik, a két forrásból származó biteket modulo 2 (xor) összeadva kapjuk azt a bitet, amely a generátor outputja lesz.

A PC-be illesztett kártyához egy, a PC operációs rendszerén futó driver tartozik, amely statisztikai ellenőrzéseket is végez az outputra, s szélsőséges esetben



2. ábra



3. ábra

riaszt. Szükséges még egy applikáció, amely segíti a kártya installálás utáni kalibrálását, és statisztikai teszteket végez, amelyek eredményét a felhasználó számára meg is jeleníti. Többszörösen hangsúlyozzuk **statisztikai tesztek folyamatos elvégzésének** szükségességét.

A legjobb beállítás mellett is eltér kissé a generált 0 és 1 bitek gyakorisága. Ezen ritkítással szokás javítani. Ennek során a sorozatot párokba osztják, és az azonos biteket tartalmazó párokat elvetik. Ha a pár bitjei különböznek, akkor az első bitet megtartják, a másodikat szintén elvetik. Jelölje p a ritkítás előtti sorozatban a 0 bit gyakoriságát, és tegyük fel, hogy a fizikai generátor bitjei független soroza-

tot képeznek. Ekkor annak a valószínűsége, hogy egy bitpár első bitjét megtartjuk a ritkítás után $= 2p \cdot (1-p)$. Így páronként a megtartott bitek számának várható értéke is $2p \cdot (1-p)$. Egy bitpár mindkét bitjét $p^2 + (1-p)^2 = 1 - \big(2p(1-p)\big)$ vetik el. Egy bitpár elvetése geometriai eloszlású ezzel a paraméterrel. Nyertük a következő tételt:

TÉTEL. *A fenti meghatározás és a függetlenség feltevése mellett az elvetett bitpárok számának várható értéke n bitpár esetén:*

$$\boxed{n\big[p^2 + (1-p)^2\big].}$$

Ez a várható érték szemléletesebb alakot nyer, ha p helyett $p = 1/2 + d$ összefüggést használjuk, ahol a d előjeles szám a várt 1/2-től való eltérést méri, tehát egy kis mennyiség.

$$p^2 + (1-p)^2 = \frac{1}{4} + 2\frac{1}{2} \cdot d + d^2 + \frac{1}{4} - 2 \cdot \frac{1}{2} \cdot d + d^2 = \frac{1}{2} + 2 \cdot d^2.$$

2.2. Pszeudo véletlenszámok generálása

Mint már említettük, az atombomba gyártásánál felmerült nagyon nagy mennyiségű véletlenszám előállításának az igénye. Fából vaskarika kívánalomként az is felmerült, hogy ezek reprodukálhatók legyenek. Neumann János 1944-ben javasolta azt, hogy erre a célra álvéletlen számokat használjanak. Valódi véletlenszámok helyett determinisztikus számsorozatot állítsanak elő, amelyek *statisztikailag ugyanúgy viselkednek, mintha valódi véletlen sorozatok lennének.* Ezeket a számsorozatokat ismételten elő lehet állítani, reprodukálni, ami a számítógépek hőskorában rendkívül fontos követelmény volt. Amint azt már mondtuk, az algoritmikusan előállítható, véletlenszerűen viselkedő determinisztikus számsorozatokat **pszeudo-véletlen** *(álvéletlen)* sorozatoknak nevezik.

A Neumann János által javasolt eljárás nagyon egyszerű volt. Kiindulásként választott egy n-jegyű (páros) számot kezdőértéknek. A pszeudo-véletlen sorozat következő számának előállításához az utolsónak generált n jegyű számot négyzetre emelte, majd az így nyert négyzet középső n jegyéből álló szám lett a sorozat új száma. (Az n jegy szükség esetén nullákkal kezdődhet.)

A négyzetközép módszernek ma már csak történelmi jelentősége van. Pszeudo véletlen számsorozatok előállítására a leggyakrabban a *lineáris kongruencia generátorokat* használják. Ezt a módszert D. A. Lehmer javasolta 1949-ben, lásd Lehmer (1951). Ehhez választanak egy $R(0)$ **egész kezdőértéket, egy** A **multiplikatív és egy** B **additív konstanst**, valamint egy nagy M **modulust**. Ezek segítségével az utolsónak előállított $R(N)$ számból kiszámítják az $R(N+1) = A \cdot R(N) + B \pmod{M}$ egész számot. Ebből egy további $U(N)$ álvéletlen valós számot M-mel való osztással kapunk. Ez az osztás egy bizonyos fajta standardizálást szolgál: az $U(N)$ számok mindig 0 és 1 közé esnek. A számítógépek RND generátora ilyen

számok sorozatát szolgáltatja. Innen a $\{0, 1, \ldots, k-1\}$ halmazba eső egyenletes eloszlású egész számokat az

$$X(N) = \lfloor k \cdot U(N) \rfloor$$

képlet ad meg (adja vissza).

A paraméterek megválasztásával és az előállított sorozat statisztikai vizsgálatával kapcsolatos kérdésekről D. Knuth (1987) 2. kötete ad kimerítő tájékoztatást.

Az algoritmikus eljárások gyors végrehajtására különböző célgépeket szerkesztettek, melyek valódi véletlen sorozatok helyett determinisztikus, de véletlenszerűen viselkedő sorozatokat alkalmaznak. Gyors hardver megoldást tesznek lehetővé a maximális periódusú Lineáris Visszacsatolt Shift Regiszterek (angol rövidítéssel LFSR). Nagy periódusú regiszterek megkeresése, elemzése a *Galois testek elméletében* való elmélyülést igényel. Maximális periódusú regiszterek kereséséhez kulcsreferencia: Golomb (1967), magyar témaismertetést ad meg Nemetz–Katona (1969).

2.3. Hibrid generátorok

Ahogy azt a 2. fejezet elején említettük, a véletlenszámok tömeges előállításának hőskorában megszületett az az ötlet, hogy bármilyen nagy munkával is, de célszerű *egyszer* nagymennyiségű véletlenszámot előállítani, és ezeket közkinccsé téve belőlük bárki, bármikor tetszőlegesen sokat fel tudjon használni. Természetesen ez a megoldás nem egy megbízható megoldás a kriptográfusok számára. Hiszen éppen a legfontosabb követelmény, a kiismerhetetlenség sérül meg. Ezt kell orvosolni ahhoz, hogy ilyen típusú megoldás számukra is használhatóvá váljon. Ezt több lehetőség is segítheti:

- A CD-ROM tárolók megjelenése az általában egyetlen alkalmazáshoz szükséges véletlenszám mennyiség sokszorosának megőrzését teszi lehetővé.
- A CD-ROM-okat abszolút biztonságos körülmények között lehet teleírni valódi véletlen (fizikai) generátorok segítségével, miközben a teleírás történhet szubjektív (személyi) beavatkozás nélkül.
- Véletlenszerű kezdettel kis blokkokat úgy lehet kiemelni a CD-ROM-ról, hogy még a kezelő sem képes felismerni, honnan származnak, tehát védelmet lehet biztosítani a saját kezelő személyzettel szemben.
- A kivételi program speciális, egyedi paramétereket tartalmazhat, testre szabható. Rövid, valódi véletlen kezdőértékeket használhat, melyek manuálisan is előállíthatók (például klaviatúra segítségével, egérmozgatásos eljárással).
- Nincs szükség gyors generálásra, tehát a generálást ellenőrző statisztikai tesztek halmaza a szokásosnál nagyobb lehet.
- Rendkívül gyors felhasználást tesz lehetővé.
- Hírközléses pár-kapcsolat esetén „egyszeri átkulcsolásra" is alkalmazható.

A hibrid generátorokról egy részletesebb analízis érhető el Nemetz–Papp (1998) cikkében. A szerzők CD-RANDOM generálási módszerrel meg is valósították a fenti elképzeléseket.

3. Véletlen kulcsgenerálás

3.1. Kriptográfiai háttér-minimum

Napjainkban a kriptográfia két elkülönült világa él egymás mellett:

- Hagyományos, *egyetlen* titkos *kulccsal* működő rejtjelzés, a **titkos kulcsú rejtjelzés**.
- *Két kulccsal* működő, **nyilvános kulcsú rejtjelzés**

Közös bennük, hogy mindkettő a leendő felhasználásnak megfelelően választott egyértelműen dekódolható *kódok nagy halmazát* választja ki. Egy továbbítandó (tárolandó) *nyílt üzenet*et mindig ezek egyikével kódolnak (rejtjeleznek). A lehetséges kódok halmazára mint kódhalmazra hivatkoznak, és elemeit valamilyen számszerű paraméterrel (sorszámmal vagy vektorral) azonosítják. Ezt az azonosítót a kriptográfiában kulcsnak nevezik. Szokásos azonban az is, hogy kulcsról csak egy titkosítandó szöveg esetén beszélnek. Egy **üzenet kulcsának** *nevezik annak a paraméternek az értékét, amellyel az adott üzenetet titkosítják.*

Mivel a kódhalmazt a támadó előtt ismertnek kell feltételezni, így az egyetlen bizonytalansági faktort a kulcs megválasztása jelenti. Ezt tehát a legnagyobb bizonytalanság biztosítása mellett, véletlenszerűen kell kiválasztani. Számunkra ebben tér el leginkább egymástól a két kriptográfiai rendszer.

A két rendszer különbözőségének igazi oka a lényegesen eltérő felhasználás.

3.1.1. Titkos kulcsú kriptográfia. A klasszikus kriptográfia módszereit elsősorban a **zárt hálózatok használják.** Ilyenek a katonai hírközlés, a diplomácia területén fordulnak elő. Fő jellemzőjük, hogy új felhasználó ilyen rendszerbe nem tud önkényesen, saját akaratából belépni. A kulcsokat többnyire egy központ generálja és osztja ki. A kapcsolatba lépő párok előre ismerik a felhasználható kódokat (transzformációkat), előre meg tudnak állapodni az egymás után alkalmazandó kulcsokban, vagy azok megválasztására alkalmazott algoritmusokban. Ezeket egymás között abszolút biztonságosan ki tudják cserélni, és elvben gondoskodhatnak abszolút biztonságos őrzésükről is. A titkos kulcsú kriptográfiában ennek megfelelően a kulcs megválasztásához *egyetlen véletlenszám* generálására van szükség, bár ez a szám akár egy végtelen számsor is lehet. Röviden felidézzük a ma leggyakrabban használt titkos kulcsú rejtjelző algoritmusokat.

DES: 1976 decemberében a National Bureau of Standards, USA, bejelentett egy új „Nemzeti Adattitkosítási Szabványt" (FIPS No. 46), amelyben a Data Encryption Standard, DES, **rejtjelző algoritmust** szabványosították. Leírása megtalálható az USA szabványok gyűjteményében. A DES 64 bites (8 byte-os) nyílt üzenetel et képez le ugyancsak 64-bites rejtjeles üzenetekbe 56 bit nagyságú kulcsméret mellett. Az IBM által **kifejlesztett blokk algoritmus** *Shannon keverő transzformációját* használva biztosítja, hogy a blokkon belül a kimenet minden bitje függ a bemenet minden bitjétől. A szabvány első változata tartalmazta azt a kikötést is, hogy csak hardverrel implementált változata használható, és az USA kormányzat megtiltotta, hogy ezt a hardver kivitelezést exportálják. Magát az algoritmust

5 évenként biztonsági vizsgálatnak vetették alá. Ez utoljára 1994-ben történt meg, amikor 1998-at jelölték meg a felhasználhatóság utolsó határának.

A DES jelenlegi legfejlettebb változata a „Triple Des, 3-DES". Ez vagy kettő, vagy három 56 bites kulccsal dolgozik. Az üzenetet először az első kulccsal rejtjelzik normál DES módban, majd a második kulccsal a megoldó algoritmust alkalmazzák. Az így nyert közbülső szövegre alkalmazzák ismét az első, három kulcsos rendszerben a harmadik kulcsot.

A DES megfejthetőségére utaló publikációk sorát Hellman egy 1977-ben tartott előadása nyitotta meg, aki a teljes kipróbálást is kivitelezhetőnek tartotta megfelelően épített hardver segítségével. A DES halálához a döntő csapást Biham és Shamir munkássága adta meg, akik egy újonnan kifejlesztett módszer, a „Differential Cryptanalysis" segítségével adtak meg egy fejtési eljárást. 1994-ben Matsui egy más típusú, ún. lineáris kriptanalízis módszert adott meg, amely már gyakorlatilag kivitelezett fejtésről számol be.

A véletlenszám generálás számára a lényeg: **egy üzenet rejtjelzéséhez 56 bit valódi véletlen bit szükséges.**

AES: A DES haldoklása láttán az amerikai *National Institute of Standards and Technology (NIST)* pályázatot hirdetett egy új amerikai (és világ) rejtjelzési algoritmusra, amely **Advanced Encryption Standard (AES)** néven volt hivatott átvenni a korábbi uralkodó DES szerepét.

Az NIST kidolgozott egy olyan követelményrendszert, amelyet a pályázatoknak ki kell elégíteni. A követelményrendszer blokk-rejtjelző algoritmust ír elő *128 bites blokkmérettel és 128-256 bites kulcsmérettel.*

A NIST *1999. március 22–23. között Rómában* egy szakértői konferenciát rendezett a beérkezett „legjobb" 15 pályamű értékelésére. Itt a legjobbnak ítélt öt algoritmust választották ki. 2001 szeptemberében hirdették ki a győztest, amely a Rijndael algoritmus lett. Részletesebben: Nemetz–Papp (2001).

A véletlenszám generálás számára a lényeg: **egy üzenet rejtjelzéséhez maximálisan 256 bit valódi véletlen bit szükséges.**

Véletlen átkulcsolás (one time pad). Ez az eljárás bizonyítottan, elméletileg fejthetetlen rejtjelezési eljárás. Ennek során a kulcssorozat minden elemét egymástól függetlenül választják ki egyenletes eloszlással abból az ábécéből, amelynek a jeleit a dokumentum is használja. A sorozat ugyanolyan hosszú, mint a dokumentum. A rejtjelzés abból áll, hogy egy adott összeadó tábla szerint a dokumentum és a kulcssorozat egymás utáni jegyeit összeadják. Mivel a partnerek ismerik egymást, képesek biztonságos csatornán a kulcssorozatot előre egymással kicserélni.

A véletlenszám generálás számára a lényeg: **egy üzenet rejtjelzéséhez ugyanannyi valódi véletlen betű szükséges, ahány betűből a rejtjelezendő szöveg áll.**

3.1.2. Nyilvános kulcsú kriptográfia. Nyilvános hálózatok: Az információs társadalomban egymást nem ismerő partnereknek kell megbízható elektronikus kapcsolatot teremtve titkos üzeneteket váltani. Erre a lehetőséget a nyilvános kulcsú kriptográfia adja meg.

Felhasználói két egymással kapcsolatban álló kulcsot választanak maguknak. Az egyik kulcsot nyilvánosságra hozzák, a másikat szigorúan titokban tartják. Az egyik kulcsot a feladó használja a rejtjelezésre, a másik kulcsot a címzett a megfejtésre. Közös rejtjelzési algoritmust használnak, amelyben könnyű kódolni, és a titkos kulcs birtokában a dekódolás is gyors. Az illetéktelen dekódolás azonban gyakorlatilag nem kivitelezhető. Ha a rendszer egy résztvevője egy másik résztvevőnek akar titkos üzenetet küldeni, akkor kikeresi a nyilvántartásból a címzett nyilvános kulcsát, és az ismert kódolási rendszerben ennek a paraméternek megfelelő kulccsal rejtjelzi az üzenetet. Ezt az üzenetet csak a hozzátartozó kulccsal, a címzett titkos kulcsával lehet visszaállítani (dekódolni), tehát ezt csak az illetékes címzett tudja elolvasni.

Megismételjük, hogy a nyilvános kulcsú rendszerek olyan közös rejtjelzési algoritmust használnak, amelyben a rejtjelzést a nyilvános kulcs birtokában könnyű elvégezni, de pusztán ezzel a kulccsal a dekódolás gyakorlatilag nem kivitelezhető. A titkos kulcs segítségével azonban a dekódolás is gyors művelet. A transzformáció végrehajtása gyors, az inverzének a meghatározása azonban rendkívül bonyolult. Az ilyen transzformációkat egyirányú transzformációnak nevezik. Ilyenek keresése a matematika nehéz feladata.

Az RSA algoritmus a modulo aritmetikában az ismeretlent hatványban tartalmazó egyenletek megoldásának nagyfokú bonyolultságát használja ki, így megfelelő nagyságú modulus esetén a megoldás technikai kivitelezhetetlensége szolgáltatja a biztonságot. A „megfelelő nagyság" itt kritikus szerepet játszik: a kezdetben biztonságosnak ítélt 512 bit hosszú kulcsok helyett ma a gyakorlatban 1024-nél rövidebb kulcsot (modulust) csak elvétve használnak.

Az RSA első alkalmazásai között szerepel a hozzáférés-védelem, digitális aláírás és az üzenethitelesítés. A témába jó bevezetést ad Akl (1983), egy összefoglaló helyzetképet pedig Simmons (1988).

A véletlenszám generálás számára a lényeg: **egy üzenet rejtjelzéséhez**

- **két nagy véletlen prímszám,**
- **egy nagy véletlen exponens szükséges.**

Az, hogy „nagy", azt jelenti, hogy a két prímszám szorzatának nagyságrendje 1024 bit, és általában ugyanekkora az E és D exponensek nagyságrendje is, de legalább az egyiké.

3.2. Véletlen kulcsgenerálás a klasszikus kriptográfiában

A klasszikus kriptográfia rövid ismertetéséből látszik, hogy két rendkívül eltérő nagyságú kulcsmérettel állunk szemben. Az egyik típus véges automatákkal dolgozik, ε nyílt szöveg rövid blokkjain hajt végre transzformációkat. Ennek megfelelően a kulcsai viszonylag rövid bitsorozatok. Előállításuk központilag, szakemberek irányítása mellett történik.

A másik eset a végtelen, véletlen átkulcsolás esete. Ehhez folyamatosan működő fizikai generátorokat használnak. Az így generált sorozatokat valahogyan tárolni és továbbítani kell. Ennek egyik módja a CD-ROM-on, DVD-ROM-on történő tárolás

és továbbítás. A generálás módját fentebb elfogadható részletességgel ismertettük, így most áttérünk a nyilvános kulcsú rendszerekre.

3.3. Nyilvános kulcsok generálása

A véletlen kulcsgenerálás egyik feladata a véletlen exponens megválasztása. Ez egy egyszerű feladat: adott nagyságrendű véletlenszámot kell pusztán generálni. Ennek a nagyságrendjét számítástechnikai, forgalmi, tehát nem matematikai meggondolások határozzák meg.

Az izgalmas kérdés a véletlen prímszámok megválasztása. Az általánosan elfogadott gyakorlat szerint a megadott nagyságrendben választanak egy véletlenszámot, majd algoritmikusan megkeresik a legkisebb, nála nagyobb prímszámot.

Az RSA titkos és nyilvános kulcspárában szereplő modulus ma javasolt minimális mérete 1024 bit, de mesterkulcsokhoz, hosszabb idejű védelem esetén 2048 bitet javasol az RSA cég. A generálás véletlenszámok alapján folyik. A kulcspárhoz két prímet kell generálni. A generálás legtöbbször úgy történik, hogy a véletlen input alapján az algoritmus egy-egy 512 bites páratlan számból indul. A véletlenszámot kettesével növelik, és a kapott számokat egy vizsgálatnak vetik alá, hogy az úgynevezett „pszeudo-prím" szám-e (potenciális prím-e)? Ez egy nem determinisztikus prímteszt. Az ellenőrzésre a Rabin-Miller tesztet használjuk, hússzoros iterációval. Korábban, amikor még 512 bites modulust is elfogadhatónak tartottak, a két véletlenszám nagyságrendjére különböző előírásokat ellenőriztek, hogy azok elég messze vannak-e egymástól. A következő gondolatsor mutatja, hogy erre 512 bites véletlenszámok esetén nincs szükség.

LEMMA. *Egy X valószínűségi változó akkor és csak akkor egyenletes a $\{0, 1, \ldots, 2^r - 1\}$ halmazon, ha X bináris kifejtésében minden bit egymástól teljesen függetlenül egyenlő $(1/2)$ valószínűséggel veszi fel a 0 és 1 értéket.*

LEMMA. *Legyen $x_i, y_i \in \{0, 1\}$*

$$X = x_1 \cdot 2^{r-1} + x_2 \cdot 2^{r-2} + \ldots + x_{r-1} \cdot 2^1 + x_r$$

és

$$Y = y_1 \cdot 2^{r-1} + y_2 \cdot 2^{r-2} + \ldots + y_{r-1} \cdot 2^1 + y_r$$

két egyenletes eloszlású valószínűségi változó a $\{0, 1, \ldots, 2^r - 1\}$ halmazon. Legyen továbbá d az a legkisebb index, amelyre $x_d \neq y_d$. Akkor $2^d \leq |X - Y| < 2^{d+1}$.

LEMMA. *Legyen X és Y két egyenletes eloszlású valószínűségi változó a $\{0, 1, \ldots, 2^r - 1\}$ halmazon. Ekkor eltérésük várható értékére fennáll, hogy*

$$\frac{1}{2} \cdot 2^{r-1} < E\{|X - Y|\}.$$

Ez $r = 512$ esetén nagy eltérést jelent: $1/2 \cdot 2^{r-1} \approx 10^{153}$, tehát a várható különbség külön „rendszabály" nélkül is nagy.

3.3.1. Billentyűs véletlenszám generátor. A generálás alapötlete az, hogy a billentyűzet gombjainak egymás utáni leütései között eltelt idő felhasználásával egy hexadecimális sorozatot állítunk elő. A billentyű-leütések közti idő kihasználása nem új ötlet. Létezik a PGP programoknak olyan változata, amely ezt a módszert használja. Ezek a programok azt kérik a felhasználótól, hogy üssön le (önkényesen választott) billentyűket, mondjuk ötvenszer.

Ami mégis újjá teszi módszerünket, az az, hogy a leütendő billentyűk sorozatában nem hagyunk helyet az önkényeskedésnek. Ezt a sorozatot a PC beépített pszeudo véletlenszám generátorának véletlenszerű indításával a leütendő sorozatot kiíratjuk a képernyőre, és a kliensnek ezt a sorozatot kell reprodukálnia. Hibás leütést a gép nem fogad el. Az eltelt időt (milisec) 16-tal osztva a maradék lesz az aktuálisan generált véletlen hexadecimális jegy.

Az ellenőrzéshez készítettünk egy C++ programot, és a vele generált véletlen outputot statisztikailag teszteltük. A tesztek megnyugtató eredményre vezettek. A program lépései:

Algoritmus

1. **Lépés:** Megadunk egy nyomtatható (képernyőre írható) karakterekből álló ABC$ stringet (vagy mátrixot). Legyen a stringben (mátrixban) szereplő jelek száma M, ami a program paramétere. A stringben szereplő karakterek ismeretén kívül bizonytalansági faktort jelenthet a benne szereplő jelek sorrendje is.
2. **Lépés:** A program közli, hogy a generált véletlen számokat mindig az [rndout.doc] file-ba menti. Felszólítja a felhasználót, hogy az exe file futtatása előtt gondoskodjon az esetleg ilyen nevű file tartalmának a mentéséről, mert ezt a program felülírja.
3. **Lépés:** A program rákérdez a generálandó hexadecimális jegyek N számára. Ez a szám 1 és 500 közti tetszőleges egész lehet.
4. **Lépés:** Állítsuk be a PC pszeudo-véletlen generátorát egy véletlen kezdőértékre, és generáljunk egymás után N pszeudo-véletlen számot az $1, 2, \ldots, M$ egész számok közül.
5. **Lépés:** Minden egyes generált pszeudo-véletlen egész esetén írjuk ki a képernyőre az ABC$ string ennyiedik betűjét. (Ez szervezhető $N = 280$ esetén például 4 darab 70 betűs sorba). Ezzel kezdődik a véletlen-hexa generálás.
6. **Lépés:** Üssük le a billentyűzeten egymás után a képernyőn levő N karaktert, miközben a PC méri és kódolja az egymás utáni leütések közti időt, és leírja (feljegyzi, tárolja) a generált értéket. A kiírás megint szervezhető 4×70-es mátrixba, ahogy ezt a konkrét példánk teszi is.

Algoritmus vége.

Teljesen lényegtelen, hogy a számítógép milyen pszeudo-véletlenszám generátort alkalmaz. A program működésének bemutatására $N = 4 \times 70$ paraméter mellett megadunk egy **leütendő sorozat**ot, majd a kiírt sorozat leütésekor létrejövő véletlenszám sorozatot. Megadunk egy példát:

$N = 280$. A leütendő sorozat:

```
3gkmh%g7wl   g%zi3$6e=,   wl3jz-bpfm   t6ibcbp8nt   2=a,%p4=ma
24hrcrcp08   ym7yms5x59   ecx4d-tpzn   3$@h5tpa60   37pfakprwu
a34-yx@0pv   0jh68$#e7!   bbizueg,kw   cynu.%r6$8   mesw%wxubn
w6b3phq9pz   d!a$%ny,x0   kwkzl!%=rf   03ap4xz9%3   za4yrct3o@
xmg0bu@d!1   su=a%cyv!2   .,9uton6yx   oo9@x$zcph   gmazktp3q3
t1zhw%!t-j   ok,t$dp8,v   r4wzg8%,xo
```

A generált sorozat:

8 2E5F 34 3C5	6687D4A 1574	F 2D60BE3 2 1	5B 1 0 5D9595	70 55 1 0 49C7
65D2EF56BB	9D 1A8E27A7	7 264AA 6BC 8	5E906AD0 0C	C6BF05 30 66
70 12CC70 2A	C0B 1 1 8E67 1	A35D4EDED6	C0EB221A4C	9BBBE0 043F
1 3 27 2E40B 2	878 1BCCBC9	AF 3 2 94E75 D	3 2 0 86A97B 2	1 45 276 8 6 0C
65EB2B9 1 6 1	9 6CE0 7F 28 2	4 2 7 8 8 2DB7 6	DBC9E59E1 1	DCB0956BF8
CBF5559E5C	37CFB0CE5D	6ABB 36 6 8E 0		

A generált sorozat gyakoriság-eloszlása:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
22	17	22	11	12	24	23	19	14	15	12	25	21	14	19	10

A 15 szabadsági fokú χ^2-statisztika értéke: 21.485, tehát a sorozat megbízhatóan véletlen.

Hat további 4×70-es sorozat: gyakoriság-eloszlása rendre:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
12	16	16	20	17	15	19	26	22	17	25	15	16	14	18	12
20	13	18	22	13	16	16	21	10	14	23	22	19	13	25	15
10	19	19	22	22	15	13	20	17	22	18	16	23	17	12	15
15	12	17	21	14	11	15	23	24	22	15	12	16	23	16	24
16	20	18	17	17	16	19	17	19	18	11	12	23	17	21	19
22	18	24	17	19	13	17	21	15	17	19	14	23	17	10	14

A hat 15 szabadsági fokú χ^2 **próba-statisztika értékei:**

Minta száma	1	2	3	4	5	6
Khi-értékek	14,28	16,46	12,80	18,06	7,66	12,46
Megbízhatóság	0,49	0,31	0,62	0,16	0,89	0,58

Ezen adatok alapján nem lehet megcáfolni a véletlenszerűség hipotézisét.

Bemutatunk egy másik, nagyon egyszerű tesztet is.

2. teszt: A statisztika a 4×70-es mátrix azon oszlopainak a számával egyenlő, amelyekben mind a 4 hexa-szám különböző, tehát oszloponként bináris változó:

$$P(\text{mind különböző}) = \frac{(15 \cdot 14 \cdot 13)}{(16 \cdot 16 \cdot 16)} = 0{,}6665,$$

$$P(\text{nem mind különböző}) = 0{,}3335.$$

A minta gyakoriságai: mind különböző: 46,
nem mind különböző: 24.

Gyakoriság: 0,6286, illetve 0,3714; az adódó χ^2 próba értéke: 0,453038. Az 1 szabadsági fokú kritikus értékek:

P	0,1	0,05	0,01
χ^2 érték	2,71	3,84	6,63

Tehát legalább 0,99 annak a valószínűsége, hogy a 70 elemű véletlen minta ekkora vagy ennél nagyobb χ^2 értéket eredményez.

Mindkét esetben a leggyakrabban használt statisztikai próbát, a χ^2 **próbát** alkalmaztuk.

3.3.2. Internetes véletlenszám generátor. Kriptográfiában olyan véletlenszámokra van szükség, amelyek kiismerhetetlenek, az előzményekből nem jósolhatók meg. Ezért javasoljuk olyan interneten elérhető fájlok használatát, amelyek azonosítását az ellenfél várhatóan nem tudja elvégezni. A módszer alkalmazhatóságát elméletileg a stacionárius, ergodikus sztochasztikus folyamatok egyik határeloszlástétele garantálja.

A program inputja egy, az internetről letöltött file. A program egy <input.txt> nevű file-t vár bemenetként, tehát a levett file-t erre a névre át kell nevezni. Ha nem talál ilyen nevű file-t a program, akkor hibát jelez.

A generált véletlen outputot egy <randout.txt> file-ba írja ki. Ha ilyen file létezett, azt felülírja, ha nem létezett, akkor újat hoz létre.

Az algoritmus vázlatos menete a következő:

1. lépés: Letöltünk egy szövegfile-t, például újságcikket az internetről. Ezt a továbbiakban egy **karakter sorozatnak (byte sorozatnak) tekintjük.**

2. lépés: A letöltött sorozatból blokkokat képezünk, felváltva a hosszú aktív, illetve b hosszú passzív blokkokat. Tekintsük a-t és b-t az algoritmus paramétereinek, amelyek ideális értékét statisztikai tesztek eredményeinek felhasználásával lehet meghatározni.

3. lépés: Tekintsük az aktív blokkok elemeit egész számoknak. Ezt megtehetjük úgy, hogy a byte-okat a bináris alakjuknak megfelelő számoknak tekintjük, de paraméterként egy egyszerű helyettesítést jelentő táblázattal is átértékelhetjük őket. A véletlenszámok generálása során a helyettesítő tábla semmilyen befolyásoló szerepet nem játszik, bármikor tetszőlegesen megváltoztatható. Megváltoztathatósága növeli viszont a bizonytalanságot. „Nagyjából" véletlenszerűen, automatikusan változtatva, a generálás konkrét megvalósítását is elrejthetjük a kezelőszemélyzet elől.

4. lépés: Választunk egy a hosszúságú $\mathbf{c}$ vektort az $\{1, 3, 5, 7, 11, 13\}$ halmaz elemeiből, lehetőleg valamilyen pszeudorandom generátor segítségével.

5. lépés: Az aktuális aktív blokk és a **c** vektor modulo 16 vett szorzata adja a következő véletlen számot.

6. lépés: A b hosszúságú passzív blokkot figyelmen kívül hagyva a következő aktív blokkal dolgozunk tovább.

Az algoritmus-tervezet megbízhatóságának kiértékeléséhez készítettünk egy C++ programot, amely mindkét számot paraméterként használja. A program rákérdez, hogy a felhasználó milyen hosszú blokkokkal kíván dolgozni. Különböző a és b paraméterek mellett számos interneten elérhető (U1–U10) cikket dolgoztunk fel, és a készített „véletlen" sorozatot statisztikai tesztek segítségével vizsgáltuk.

cikk	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
0	60	24	24	49	35	19	36	49	52	68
1	45	9	15	48	25	36	31	57	49	61
2	58	21	14	51	22	24	36	52	41	60
3	73	20	20	53	24	24	34	47	41	45
4	61	21	25	64	31	31	41	47	45	55
5	35	22	23	53	30	30	32	36	43	50
6	61	23	23	63	18	24	24	44	50	50
7	48	15	22	49	21	31	40	49	43	40
8	63	27	26	68	22	24	38	54	53	48
9	53	24	19	51	20	20	26	46	39	53
A	61	21	20	63	21	23	32	53	62	45
B	55	28	23	51	22	20	33	55	44	52
C	59	20	20	60	24	32	39	51	45	48
D	58	16	22	67	17	33	37	43	53	39
E	57	19	26	61	29	38	36	52	56	63
F	42	35	27	61	19	33	34	41	51	45
χ^2	22,96	23,65	9,464	12,842	16,295	19,828	9,543	9,691	12,577	19,966

1. táblázat: Generált véletlen számok gyakoriságeloszlásai.

Az első két cikk χ^2 értéke a $P = 0{,}1$-es elfogadási határ fölé csúszik, a többié megfelelően alacsony, így a véletlenszerűség feltételezése elfogadható.

A statisztikai tesztek a Knuth (1987) könyv tesztjeinek adaptációi voltak, $p = 0{,}05$ hibavalószínűség mellett, khi-négyzet kiértékeléssel. Nagyobb aktív-blokk hosszúság egyenletesebb eloszlást eredményezett. A passzív blokkok szerepe az, hogy a program outputjaként előálló sorozat elemei kellő mértékben függetlenek legyenek. Mindkét blokk hosszának növelésével az eredmények véletlenségét növelhetjük, ám egyúttal csökken a fileból kinyerhető sorozat hossza is. A statisztikai tesztek eredményeként a 10 hosszúságú „aktív" és 3 hosszúságú „tétlen" blokkok használatát ajánljuk.

Az 1. táblázat 10 cikkből nyert gyakoriság eloszlásokat mutatja az $a = 10$, $b = 3$ paraméterek esetén. Az utolsó sor a kapott „véletlen-hexa" sorozatnak az egyenletes eloszlástól való khi-négyzet eltérését adja meg.

A cikkek, illetve egyéb dokumentumok Internetről történő elmentésekor vigyázni kell arra, hogy lehetőleg csak természetes nyelvű szöveg kerüljön a program inputjára. A weboldalakon jelentős részt kitevő html utasítások, gyakran ismétlődő sablonok (például tartalomjegyzék) felhasználása nagyban rontja az eredmény véletlenszerűségét. Így a *mentés másként (text típus)* illetve a *kijelöli mindet* menüpontok használata helyett biztonságosabb, ha manuálisan jelöljük ki a kívánt részt és a vágólapon keresztül bármilyen egyszerű szövegszerkesztő segítségével mentjük el a program számára.

4. A véletlenszerűség statisztikai ellenőrzése

Az eddigiekben többször hangsúlyoztuk, hogy szükség van olyan tesztekre, amelyek segítségével eldönthető, hogy egy adott sorozat tekinthető-e véletlenszerűnek. Ez a feladat része egy általános statisztikai feladatnak. A kérdés az, hogy egy adott megfigyelés sorozat származhat-e egy adott valószínűség eloszlásból. Ezt a kérdést valamennyi statisztikai tankönyv tárgyalja, így nem lenne célszerű erre az általános kérdésre időt/teret vesztegetni. A továbbiakban cél-orientált tesztekkel foglalkozunk.

A véletlenszerűséget *vizsgáló kriptográfiai hátterű statisztikai próbák*nak egy gyűjteményét írja le a **National Institute of Standards and Technology (2001)**. Ezek az eljárások jól használhatóak arra, hogy egy *bináris sorozatnak* a véletlenszerűtől való eltérését kimutassák. A tesztelőnek azonban figyelembe kell vennie, hogy a véletlenszerűtől való eltérések lehetnek egy gyengén tervezett generátor következményei, de felléphetnek a tesztelt bináris szekvenciában lévő anomáliák miatt is (vagyis, bizonyos számú hiba fellépése valószínű a véletlen sorozatokban, amelyeket egy meghatározott generátor állít elő). A tesztelőn múlik, hogy a teszt eredményeket helyesen értelmezze.

4.1. A NIST által támasztott követelmények

Megadott bináris sorozatok véletlenszerűségére, megjósolhatatlanságára és a tesztelésre vonatkozó statisztikai tesztek összeállításánál három alapvető feltételt kell ellenőrizni. Ezek:

- **Egyenletesség**: Egy véletlen vagy pszeudo-véletlen bitsorozat előállításának bármely időpontján egy nulla vagy egy érték előfordulása egyformán valószínű, vagyis mindegyiknek a valószínűsége pontosan $1/2$. A nullák (vagy egyesek) száma várhatóan $n/2$, ahol n a sorozat hossza.
- **Skálázhatóság**: Egy sorozatra alkalmazható bármely próba alkalmazható egy véletlenszerűen kiragadott részsorozatra is. Ha a sorozat véletlen, akkor minden kiragadott részsorozatnak is véletlennek kell lennie. Így bármely kiragadott részsorozatnak meg kell felelnie minden véletlenszerűségre vonatkozó próbának. (Természetesen a vizsgált részsorozatok száma csak elenyésző kis töredéke lehet az összes lehetséges részsorozatnak: Kolmogorov bizonyította,

hogy ha a részsorozat választó algoritmusok száma legfeljebb $\frac{1}{2}e^{2n^2(1-\varepsilon)}$, akkor mindig létezik (n, ε) véletlen bináris sorozat, ahol az n-nél rövidebb részsorozatok egyenletesek legfeljebb ε eltéréssel (ld. Knuth (1981) 2. kötet, 174. o.))

- **Konzisztencia:** egy pszeudo véletlenszám generátor viselkedésének konzisztensnek kell lennie a kiinduló értékek mindegyikére. Nem kielégítő, ha egy PRNG-t csak egyetlen kiinduló értékből származó output alapján, vagy egyetlen másik paramétere alapján tesztelünk, vagy ha egy RNG-t egy olyan output alapján tesztelünk, amely egyetlen fizikai outputból lett származtatva.

A NIST által összeállított követelményrendszerben javasolt tesztekre 2001-ben a Hungard Kft egy RNDtests.exe nevű programcsomagot készített. A programcsomag ismertetésével rendkívül célratörően lehet magát a NIST javaslatot is ismertetni. Ezért az ismertetésnek ezt a módját választottuk.

4.2. Az RNGtests.exe program tartalma

A program indításakor a képernyőn megjelenik egy „startlap", amelyet a 4. ábra mutat.

A lap első kérdésként rákérdez az ellenőrizendő file nevére, majd a browserrel ezt megkeresi. Ugyancsak rákérdez az input formátumára.

A program input formátumai

A program háromféle input-formátumot tud fogadni. Ezek:

- Egy bájtban egy bit. (Bit formátumban előállított sorozatok. Ilyenek például a véletlenszerűen generált hexadecimális sorozatok.)
- Egy bájtban 8 bit. (Hagyományosan készített file, mesterségesen bit-értékekre korlátozva.)
- 0,1-et tartalmazó ASCII sorozat. (Ez a fajta file a kifejezetten erre a célra készített, véletlen bitsorozatot tartalmazó file.)

A futtatás elején meg kell jelölni, hogy ezek melyike az alkalmazni kívánt sorozat.

Bemeneti adat annak a file-nak a neve, amely az ellenőrizendő adatsort tartalmazza. Meg kell jelölni azokat a teszteket, amelyeket a felhasználó a megadott file-on el akar végezni. A Maurer-, Linear Complexity-, Random excursion teszteket fizikai generátorok bevizsgálásánál ajánlott elvégezni, „normál" napi munka számára kevésbé ajánljuk. Ezeket a teszteket a gép gyorsan hajtja végre, az eredményt a tudakozó lap jobb alsó részében írja ki.

A program minden tesztnél kijelzi, hogy mekkora a teszt megbízható működéséhez szükséges mintanagyság. A tesztek lefutása után a képernyőn megjelennek a tesztek eredményei, és egy összefoglalt ítélet arról, hogy a sorozat véletlenszerű-e.

A döntés minden teszt esetén $\alpha = 0{,}01$ megbízhatósági szinten történik. Ez azt jelenti, hogy várhatóan 100 sorozatból egy sorozatot utasít el a próba úgy, hogy a sorozat véletlen volt.

4.3. Tesztfajták

Frekvencia teszt. Az ellenőrizendő sorozatot adott hosszúságú konzekutív blokkokra osztjuk, ezeknek a blokkoknak a valószínűségeit az egyenletesség és függetlenség feltételezése mellett kiszámítjuk. Az adódó valószínűség-eloszlásra alkalmazzuk a khi-négyzet próbát. A teszt feladata, hogy kimutassa, ha a sorozatban túl sok nulla vagy egyes van.

Futamhossz teszt. k hosszú futamnak nevezünk egy pontosan k darab egymás után következő, azonos bitekből álló részsorozatot, amelynek mindkét végén az ellentétes érték található. A teszt célja, hogy a különböző hosszú futamok statisztikailag várható számát összehasonlítsa a sorozatban megfigyelt értékekkel. A futamhosszak túl nagy vagy túl alacsony száma arra utal, hogy a sorozatban túl sok az oszcilláció.

Spektrális Diszkrét Fourier Transzformáció teszt. A bitsorozatban esetlegesen fellelhető periodikus jelekre érzékeny. A sorozatban vizsgálat előtt -1-re cseréljük a nulla értékeket, s a módosított sorozatra elvégezzük a diszkrét Fourier transzformációt. A kapott változók sorozata reprezentálja az eredeti sorozat hosszabb-rövidebb periodikusan előforduló mintáit. Az ezután készülő statisztika a normális eloszlást használja referenciaként.

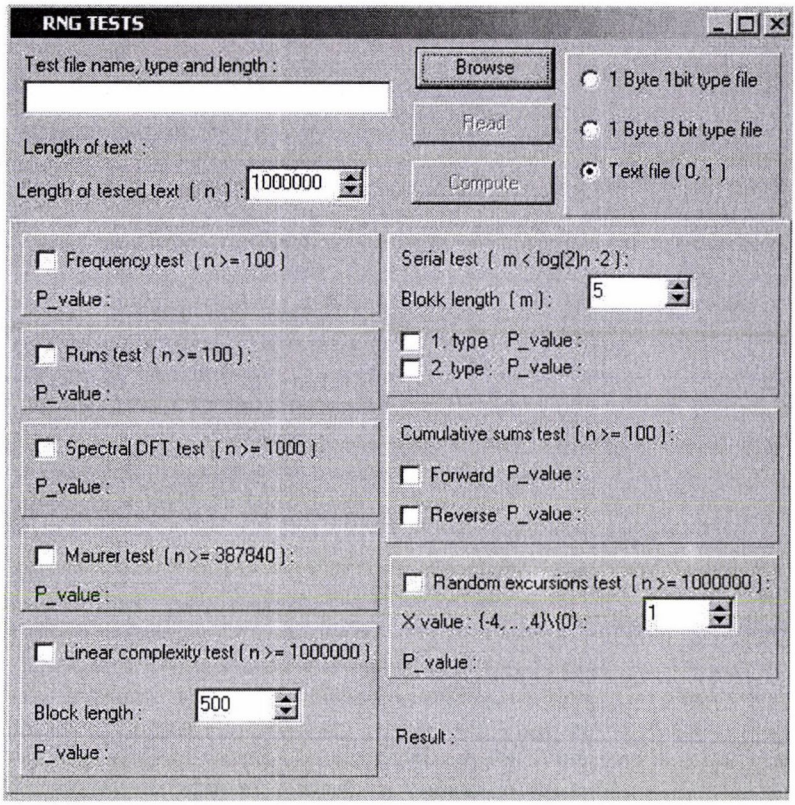
Maurer-féle univerzális entrópia teszt. A teszt fókusza az azonos minták (részsorozatok) közötti bitek számán van. A teszt azt méri, hogy a sorozat mennyire tömöríthető információvesztés nélkül. A szignifikánsan tömöríthető sorozatot *nemvéletlennek* tekintjük. A módszer alapja, hogy a sorozatot egy kezdeti inicializációs és egy maradék tesztrészre bontjuk. A teszt azt méri, hogy az inicializációs sorozat fix, k hosszú mintáival hogyan írható le a sorozat második, tesztszakasza. A statisztikához kapcsolódó referencia eloszlás a fél-normális eloszlás.

Lineáris komplexitás teszt. A teszt a lineáris shiftregiszterrel történő modellezhetőséget méri. A sorozatot k hosszú részblokkokra bontjuk, és minden blokkra végrehajtjuk a Berlekamp-Massey algoritmust, hogy meghatározzuk azt a legrövidebb lineáris shiftregisztert, amely az adott blokkot generálni képes. Véletlen esetben a regiszter hossza $k/2$ körüli érték. A kapott hosszakat minden részblokkra értékeljük.

Sorozat teszt. A teszt célja, hogy megvizsgálja az összes lehetséges 2^m darab egymást átfedő m-bites minta eloszlását a vizsgált sorozatban. Az adódó valószínűség-eloszlásra alkalmazzuk a khi-négyzet próbát. A teszt $m = 1$ esetben megegyezik a frekvencia teszttel. A teszt során két, részben különböző módon számított statisztikát is beállíthatunk.

Kumulatív összeg teszt. A 0/1 arány 0-tól való lokális eltérését méri. A teszt során a 0 értékeket -1-re módosítjuk és képezzük a részletösszegek s_i sorozatait, tehát az első i elemek összegeit. A statisztika az s_i értékek maximumát vizsgálva dönt a véletlenszerűségről. Beállítható, hogy a bejárást a sorozat melyik végéről kezdjük.

Véletlen bejárás teszt. Az előző tesztre épül. Megfigyeli, hogy az előző tesztben kiszámított s_i értékek hányszor egyeznek meg egy paraméterként beállítható



4. ábra. Az RNDtests.exe program tudakozó lapja

X értékkel. Minden X mellett független tesztnek tekinthető. A referenciaeloszlás a khi-négyzet eloszlás.

Hivatkozások

[1] Benson, R. L., Warner, M. (eds), Venona, Aegean Park Press (1996).

[2] Buszlenko, N. P.-Golenko, D. I., *Monte Carlo módszerek,* Műszaki Kiadó (1966)

[3] Friedman, *The Index of Coincidence and Its Applications in Cryptography,* Riverbank Publication, No. 22 (1920)

[4] Golomb, S. W., *Shift Register Sequences,* Holden-Day, San Francisco (1967)

[5] Jahnsson, B., *Random number generators,* Petterson, Stockholm(1966)

[6] Kahn, D. (1967), *The codebreakers,* MacMillan Co., New York

[7] Kerckhoffs, A., (1883) *La cryptographie militaire,* republished as „A Mathematical Theory of Cryptograhy", Bell System Techn. J. (1949)

[8] Knuth, D. E., *A számítógép-programozás művészete* 2., Szeminumerikus algoritmusok, Műszaki Kiadó, Budapest (1987)

[9] Lehmer, D. H., Linear Congruence Generators, in: *Proc.* 2nd *Symposium on Large-Scale Digital Calculating Machinery,* Harward Univ. Press, Cambridge (1951), 141–146

[10] Muha L. (szerk), Az informatikai biztonság kézikönyve, 10. Javított kiadás, Verlag Dashöfer, Budapest (2004)

[11] Nemetz, T. – Katona, G., Néhány megjegyzés a shift-regiszter generátorokról, *MTA Számítástechnikai Központ Közleményei,* No. 5 (június) (1969), 1–10.

[12] Nemetz, T. – Papp, P., Hybrid Random Byte Generators, in: *Global IT Security,* Papp, Gy.-Posch, R. (ed), Proc. of the 14th Int. Conf. on Info. Security, Wien (1998), 366–380

[13] Nemetz, T. – Papp, P., Belga Kriptográfusok Amerikában, *BYTE Magyarország,* V., No. 3. (2001), 38–40.

[14] Nemetz, T. – Wintsche, G., *Valószínűségszámítás és statisztika mindenkinek,* Polygon, Szeged (1999)

[15] NIST: Special Publication 800–22 (2001), *Statisztikai próbák a kriptográfiai alkalmazásoknál használt véletlen- és pszeudo-véletlen szám generátorokra* (javított kiadás: 2001. május 15.).

[16] Shannon, C. E., Communication theory of secrecy systems, *Bell Syst. Techn. J.,* **28** (1949), 656–715.

[17] Vincze, I., *Matematikai statisztika ipari alkalmazásokkal,* Műszaki Könyvkiadó, Budapest (1968).

[18] Yule, G. U. – Kendall, M. G., *Bevezetés a statisztika elméletébe,* Közgazdasági és Jogi Könyvkiadó, Budapest (1964).

GENERATING AND TESTING CRYPTOGRAPHICALLY SECURE RANDOM NUMBERS

Gergely Borbás, Tibor Nemetz, Pál Papp

In this paper we review some methods for generating real random numbers and pseudo-random numbers from a cryptographer's point of view. After summarizing the necessary crypto background, we examine how much random numbers are needed for different encryption algorithms.

We have developed two new methods for generating cryptographically secure pseudorandom numbers.Finally we deal with the methods known in the literature for testing the quality of a random number generator

Alkalmazott Matematikai Lapok **23** (2006), 335–348.

INTERNETES SZOLGÁLTATÁS-MEGTAGADÁSOS TÁMADÁSOK JÁTÉKELMÉLETI MODELLBEN

BENCSÁTH BOLDIZSÁR, VAJDA ISTVÁN

Budapest

Cikkünk kriptográfiai protokollok szolgáltatás-megtagadásos (Denial of Service – DoS) támadások elleni védelméről szól. A DoS támadások modellezésére a folyamatot stratégiai játékként értelmezzük. Ebben a modellben a támadó maximalizálni kívánja a kiszolgáló elhasznált kapacitásait, míg a kiszolgáló minimalizálni próbálja az elpazarolt erőforrásokat, és megpróbálja továbbra is kiszolgálni a legitim klienseket. A játékelméleti szemléletmódot részleteiben mutatjuk be, és felhasználjuk azt a kliens oldali rejtvény technika (client-side puzzle) optimalizálására. A cikkben analizáljuk azt az esetet is, amikor a szerver optimális kevert stratégiát választ a védekezéshez.

1. A szolgáltatás-megtagadásos támadás

A szolgáltatás-megtagadásos (Denial of Service – DoS) támadások kiemelt helyet foglalnak el a legveszélyesebb internetes fenyegetettségek között. A DoS támadás során a célpont egy internetes kiszolgáló, a támadó célja pedig az, hogy megbénítsa a kiszolgálót. A támadó megpróbálja felemészteni a kiszolgáló erőforrásait annak érdekében, hogy a kiszolgáló ne tudja ellátni feladatát, ne tudja biztosítani az általa nyújtott szolgáltatásokat, vagy a szolgáltatás minősége jelentősen csökkenjen. Noha számos megoldást és nyújtandó módszert javasoltak már a DoS elleni védekezésre, a védelmi módszerek többsége valamilyen gyengeséget tartalmaz, és nem alkalmazható speciális körülmények között. Valójában DoS támadással szemben a számítógépes rendszerek többsége ma többnyire védtelen. Megfelelő tudással és akarattal szinte mindig kivitelezhető sikeres DoS támadás. Az, hogy a DoS támadást sérülékenységnek, hibának tartjuk-e, jelentősen függ attól, hogy a DoS támadást egész pontosan miként definiáljuk, hiszen ilyen támadásnak tekinthető a sávszélesség elfoglalása is, vagy például a biztonsági protokollok túlterhelése.

A *protokoll átsorrendezés* (protocol reordering) és *protocol finomítás (protocol enhancement)* módszerek a biztonsági protokollok robosztusabbá, erősebbé tételét célozzák meg az erőforrások elhasználását célzó támadások ellen ([7], [6]). Ezek a megoldási javaslatok a protokollt erősebbé teszik azokkal a támadásokkal szemben, amelyeket egyetlen forrásból indítanak. Fontos problémája azonban a megoldásnak, hogy az elosztott támadásokkal szemben a megoldás nem mindig eléggé hatásos.

Az ún. állapot nélküli (*stateless*) protokollok a memóriafoglalásból, illetve a memória túlhasználatból eredő DoS támadások ellen próbálnak fellépni. Ebben az esetben a memória-túlhasználat kérdése valójában egy másik problémára kerül áthárításra, a hálózati forgalom túltöltésére, mivel a memóriahasználat csökkentését a hálózati forgalom növelésével érik el a megoldások. Az állapot nélküli protokollok tehát megoldanak bizonyos problémákat, de nem tekinthetők általános megoldásnak. (Bővebben az állapot nélküli protokollok használatáról DoS ellen az [1] cikkben.)

A DoS támadások elleni védelem jó kiindulópontja a támadók felismerése. A támadás forrásának felderítésére több megoldást javasoltak már, ilyen például az ún. IP Traceback ([10], [3]). Sajnálatos módon ez a megoldási javaslat jelentős ellentmondásban áll az internet egyik legfőbb tulajdonságával, az anonimitással. Az olyan megoldások, mint az onioun routing, illetve például annak egyik implementációja, a freenet teljes anonimitást nyújthat a támadó számára. Ilyen esetben a Traceback és az ahhoz hasonló algoritmusok teljes mértékben haszontalanná válhatnak. Másik oldalról vizsgálva az is látható, hogy az ilyen felderítő megoldások többnyire a csomagok forrását próbálják felderíteni, nem a támadás forrását. A mai internetes támadások többségét ún. zombi számítógépek hatják végre, olyan gépek, amelyek a támadó irányítása alatt állnak, de nincsenek a támadó tulajdonában. Ilyen módon a támadó gépek felderítése önmagában nem elégséges a támadás kivédésére.

Több eljárás, mint az ún. *ingress filtering*, *rate control*, *distributed rate control* az elárasztásos jellegű támadásokat tudja nehezebbé tenni. Ez azonban nem mindig működik: amennyiben a MULTOPS ([4]) széles körben használttá válna, úgy túl nagy erőforrásra lenne szükség a nyert adatok feldolgozásához. Amennyiben viszont a megoldás nincs széles körben bevezetve, úgy a rendszer igen kis mértékben használható. Az újabb és újabb megoldások bevezetésével ráadásul a támadók is módosítják a támadási stratégiájukat oly módon, hogy az a hasonló hálózati forgalmat figyelő és gátló eszközökkel szemben jobban ellenálló legyen.

A kliens oldali rejtvény (client side puzzle) és más ún. díjazás alapú (pricing) alapú algoritmusok ([2], [5]) arra kínálnak lehetőséget, hogy a protokollok kevésbé legyenek sérülékenyek a számítási kapacitásokat elfoglaló támadásokkal szemben. Ezen eszközök használata, valódi alkalmazhatósága elosztott rendszerekben (többezer támadóval szemben) azonban még kérdéses. Jelen cikkünkben példát mutatunk ilyen kliens oldali rejtvényre, és alkalmazzuk azt a játékelméleti modellezésünkben.

Számos protokoll, még a fejlettebbek is védtelenek az elosztott DoS *(Distributed Denial of Service – DDoS)* támadásokkal szemben. A jelenlegi internetes infrastruktúra strukturális gyengeségei teszik lehetővé a protokollok sebezhetőségét, és ez a sebezhetőség tovább növekedhet az elosztott támadások miatt. A jelenlegi eszköze-

inkkel az ilyen elosztott támadások ellen egyszerűen nem lehetséges védekezni és a sebezhetőséget megszüntetni. Néhány megoldás, mint a már említett kliens oldali rejtvény megoldást nyújthat a DDoS bizonyos típusaira, azonban ezen megoldások sem mindig használhatóak ([5]). Annak értelmezése, hogy egy módszer sikeres-e, jelentős mértékben függ attól a ténytől is, hogy a sikert hogyan modellezzük. Ennek megfelelően az ilyen védekezési módszerek adott modellen belül sikeresek lehetnek, de nem mindig alkalmasak valós életbeli használatra.

A kriptográfiai protokollok (melyek alatt most olyan internetes kommunikációs protokollokat értünk, amelyek kriptográfiai transzformációkat alkalmaznak) többnyire igen sérülékenyek DoS támadásokra, mivel komplex számításokat hajtanak végre, mint a nyilvános kulcsú rejtjelezés. A DoS támadást végrehajtó támadó lehetőségei és céljai lényegesen eltérnek egy „hagyományos" támadástól, amint az kriptográfiai protokollt céloz meg. A DoS támadó arra próbálja rábírni az áldozatot, hogy az sok felesleges műveletet végezzen el feleslegesen egy viszonylag rövidnek tekinthető idő alatt, és ezen műveletek lehetőség szerint minél nagyobb számítási igényt jelentsenek a kiszolgáló számára. Természetesen a kiszolgáló célja ekkor az lehet, hogy minimalizálja az így felmerült számítási költségeket. Mind a kliens, mind a szerver optimálisan próbálja felhasználni erőforrásait.

Kliens oldali rejtvény bevezetése a protokollba

A játék modelljét egy kihívás-válasz alapú kézfogás (handshake) jelenti a kiszolgáló (S) és a kliens (C) között. A kézfogás lépései a következők:

1.	$C \rightarrow S$:	szolgáltatás kérésének elküldése
2.	S :	rejtvény (challange) elkészítése
3.	$S \rightarrow C$:	a rejtvény feladványának elküldése
4.	C :	a rejtvény megfejtése
5.	$C \rightarrow S$:	a megfejtése elküldése
6.	S :	a megfejtés ellenőrzése
Ha a megfejtés helyes:		
7.	S :	a szolgáltás további lépéseinek elvégzése

A kliens oldali rejtvény lényege, hogy megkérjük a klienst, számítási kapacitásának felhasználásával bizonyítsa be, hogy kérése valóban komoly. Amennyiben valaki sok kérést kíván a szerver felé végrehajtani, maga is sok számítás végrehajtására kényszerül, így a feldolgozási szükséglet esetleges aránytalansága csökkenhet.

A kliens oldali rejtvény kihívása, azaz a rejtvény maga egy olyan feladat, amit a kliensnek kell megoldania, és a helyes választ visszaküldenie az 5. protokollépésben. A kiszolgáló a feladvány nehézségét, komplexitását az aktuálisan érzékelt veszélyhez igazítja hozzá. Ha a rejtvény erőssége megfelelően van beállítva, úgy a rejtvény le tudja lassítani a támadót, és így akár rá tudja bírni arra is, hogy abbahagyja DoS támadási kísérletét.

A kézfogás sikeres lefutása után történik meg a protokoll lényegi (belső) részének végrehajtása. A kézfogás tehát valójában egy protokoll fejlécnek tekinthető, amely algoritmikus védelmet nyújt DoS támadás ellen. Az igen rövid protokoll fej-

léc használata a sokkal hosszabb protokoll módosítása helyett azért célszerű, mert a protokoll lényegi (belső) része lépések tucatjait vagy százait is tartalmazhatja, és jelenleg nem rendelkezünk kellő tudással hosszabb protokollok megfelelő DoS elleni védelmének kifejlesztésére.

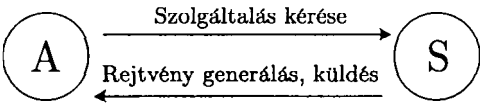
A ma használt kriptográfiai protokollokban általában hasonló kézfogást használnak fel a kommunikációban részt vevő partnerek hitelesítésére.

Megállapítható tehát, hogy a DoS támadás során a támadó maximalizálja a kiszolgáló erőforrás-veszteségét, míg a kiszolgáló minimalizálja a veszteséget, hogy elérhető maradjon a jogosult kliensek számára. Ez a helyzet felveti a játékelméleti ([9]) megközelítést. Két „természetes" szereplőnk van: a kiszolgáló és a támadó, akik egy játékban vesznek részt.

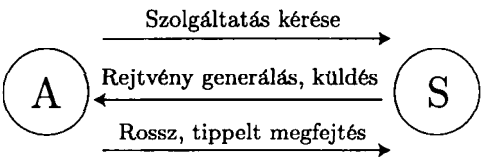
2. A játék modell

2.1. Játékosok, stratégiák, költségek, a játék mátrixa

Két szereplőt különböztetünk meg a játék során, ezek a kiszolgáló (S) és a támadó (A). A támadó célja az, hogy kiválassza a legjobb stratégiát, amely maximalizálja a szerver elpazarolt erőforrásait, míg a szerver megpróbálja minimalizálni, mint veszteséget, annak érdekében, hogy továbbra is biztosítani tudja szolgáltatásait a jogosult felhasználók számára. Egy protokoll-lépés költségét annak végrehajtása során elhasznált erőforrással definiáljuk, ezt a későbbieknek csak költségnek fogjuk jelölni. A támadó stratégiáiból hármat illusztrálunk az 1., 2., 3. ábrákon.



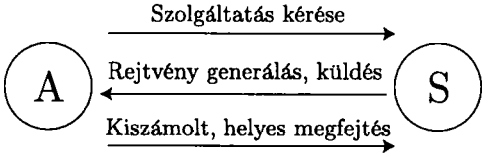
1. ábra. Csak a szolgáltatás kérése kerül elküldésre a támadótól



2. ábra. A támadó válaszol a rejtvényre, de csak egy véletlenszerű, rossz válasszal

Az első támadási stratégia esetén a támadó csak a protokoll nyitólépését, a szolgáltatás kérését hajtja végre. Ez a stratégia azt jelenti, hogy a támadó nem akarja elpazarolni erőforrásait a megfelelő válasz kiszámítására, de az is elképzelhető, hogy nem képes kétirányú csatorna nyitására a kiszolgáló irányában.[1] Ha a

[1] Cím hamisítása esetén például igen gyakori, hogy a támadó csak küldeni tudja a hamis adatcsomagokat, fogadni nem.



3. ábra. A támadó végrehajtja a teljes protokoll-fejlécet

támadó a második stratégiát választja, úgy egy hamis, nem kiszámolt, véletlenszerű megfejtést küld a kiszolgálónak annak érdekében, hogy rávegye a kiszolgálót a válasz ellenőrzésére és így az előző stratégiához képest további számításokra (2). A harmadik stratégia esetén a támadó megfelelően végrehajtja a kézfogás összes lépését (3). A harmadik stratégiát kell alkalmazni a támadónak akkor, ha el akarja érni, hogy a kézfogás utáni költséges részeket (például a digitális aláírás készítése) is végrehajtsa a szerver. A kliens oldali rejtvény, DoS kézfogás célját úgy is megfogalmazhatjuk, hogy azt kívánjuk elérni, hogy a támadó ritkán válassza ezt a stratégiát a támadás alatt, azaz a költséges rész ritkán kerüljön végrehajtásra. A támadó fenti stratégiáit jelöljük rendre A_1, A_2 és A_3-mal.

Tételezzük fel, hogy a szerver védekezési stratégiái a feladott rejtvény nehézségével, komplexitásával kerülnek megadásra. A kiszolgáló a rejtvény algoritmikus bonyolultságát különböző szintekre tudja beállítani. A nehézségi szint kiválasztásának alapja a kiszolgálón folyamatosan nyilvántartott és követett támadási jelzők, statisztikák alapján történik. A vizsgált támadási jelzőket és a rejtvényeket is megfelelő gondossággal kell előkészíteni, el kell kerülni például egy túl bonyolult rejtvény feladását is, hisz ez esetlegesen egy nagyobb számítási kapacitást igénylő ellenőrző lépéssel járhat.

Az egyszerűség kedvéért tételezzünk fel két kiszolgálós stratégiát, jelölje S_1 egy kisebb bonyolultságú rejtvény, míg S_2 egy nagyobb bonyolultságú rejtvény feladását.

A fentiek következtében a $G(A_j, S_k)$ egy olyan játékot jelöl, ahol a támadó az A_j, míg a kiszolgáló az S_k stratégiát választja.

Feltételezhetjük, hogy amennyiben a kliens oldali rejtvény használatra kerül, úgy a kliens oldali rejtvény megfejtéséhez szükséges algoritmust a szolgáltatás biztosítója, azaz a kiszolgáló már eljuttatta az előfizetőknek, klienseknek. Hasonlóképpen feltételezhetjük, hogy mindkét játékos képes jó közelítéssel megbecsülni a protokoll-lépések költségeit mindkét oldalon (természetesen feltehető a mai kereskedelemben megtalálható fejlett számítógépek használata mindkét oldalon). A felmerült költségek mindkét oldalon a számítási és tárolási kapacitások felhasználásából erednek. A költségek pontos felosztása és értékszerűsítése számos tényezőtől függ, mint a rejtvény algoritmusa, annak implementációja, a használt hardver és szoftverkörnyezet (pl. operációs rendszer, esetleges segédprocesszor), vagy a hálózat. A számításainkat absztrakt költség értékkel végezzük: egyetlen pozitív valós számmal jellemezzük a költségösszetevők összességét.

Az 1. táblázatban bevezetjük a protokoll lépések költségeit a különböző stratégiáknak megfelelően.

Lépés	Költség a támadónál	Költség a kiszolgálónál
Szolgáltatás kérés	c_r	
Rejtvény elkészítése és küldése		$c_c(k)$
Hibás válasz küldése	c_g	
Helyes válasz küldése	$c_a(k)$	
Válasz ellenőrzése		$c_v(k)$
$p1$. Protokoll kérés	c_p	
$p2$. Protokoll válasz		c_e

1. táblázat. A protokoll egyes lépéseinek költsége (k a szerver stratégiáját jelöli, azaz $k = 1, 2$)

Az 1. táblázatban jelölt $p1$ és $p2$ lépésének költsége a támadó A_3 stratégiához tartozik, amikor a protokoll kézfogás helyes lefutása után maga a védett protokoll is végrehajtásra kerül. A támadó ilyen esetben eléri, hogy a szerver egy bonyolultabb feladatot hajtson végre. Ennek megfelelően várhatóan a c_p egy kis költséget jelent a kliens oldalán, míg a c_e egy nagyobb mértékű kiszolgálási költséget jelöl. A 2. táblázatban foglaltuk össze a tiszta stratégiák alkalmazása esetén felmerülő teljes költségeket. Azt a stratégiát nevezzük tiszta stratégiának, ahol a játékos egy bizonyos állandó stratégia szerint játszik. Ha a játékos a stratégiáját egy valószínűségi eloszlás szerint választja meg, azt kevert stratégiának hívjuk.

	S_k	
	támadó költsége	kiszolgáló költsége
A_1	c_r	$c_c(k)$
A_2	$c_r + c_g$	$c_c(k) + c_v(k)$
A_3	$c_r + c_a(k) + c_p$	$c_c(k) + c_v(k) + c_e$

2. táblázat. A protokoll lefutásának költségei ($k = 1, 2$ szerver stratégia)

A 2. táblázat értelmezése a következő:

$G(A_1, S_k)$: A támadó egy szolgáltatás kérés üzenet segítségével megpróbálja rábírni a kiszolgálót arra, hogy az egy rejtvényt elkészítsen, és elküldje neki. A támadó költsége c_r, míg a kiszolgálóé $c_c(k)$, ahol a k paraméter a készített rejtvény szerver által kiválasztott bonyolultságát jelenti.

$G(A_2, S_k)$: Miután a támadó fogadta a rejtvényt a kiszolgálótól, a támadó nem küldi el a helyes választ, mert ez sok számítási kapacitásába kerülne. Ehelyett elküld a kiszolgálónak egy formailag helyes, véletlenszerű választ, ami csak egy kicsi c_g költségbe kerül. A válasz természetesen egy rossz megfejtés, de a kiszolgálónak ezt le kell ellenőriznie, ami $c_v(k)$ költséget jelent a számára. A teljes költség így a kiszolgáló oldalán $c_c(k) + c_v(k)$.

$G(A_3, S_k)$: A támadó ebben az esetben kiszámolja a helyes választ a rejtvényre, amely a k bonyolultság függvényében $c_a(k)$ költséget jelent számára. A támadó célja

az, hogy olyan mélységben tudjon belépni a védett protokollba, amennyire lehet. Jelen modellben feltételezzük, hogy a DoS kézfogás utáni első protokoll-lépés kerül végrehajtásra. Ez tipikusan egy rövid, egyszerű kérés, amelyre a szerver jelentős számítási kapacitás felhasználása után válaszol. Ha ez a lépés gyakran bekövetkezhet, az a DoS kézfogás hibáját jelenti, hisz nem tudtuk megvédeni a szervert, és elfogyhat a számítási kapacitás. A szerver teljes költsége $c_c(k) + c_v + c_e$.

Hasonlóan egyszerű modellt használunk fel a játékosok erőforrásának vizsgálatára. Feltételezzük, hogy létezik egy teljes költség mindkét félnél, amelyet fel tud használni, legyen ez egy R, pozitív valós szám. Ha a $G(A_j, S_k)$ játék $c_{int}(j,k)$ költséget jelent a támadónak, akkor arra számíthatunk, hogy $R/c_{int}(j,k)$ alkalommal fogja végrehajtani azt. Tegyük fel, hogy a $G(A_1, S_1)$ játék zajlik. A 2. táblázat szerint $c_{int}(j,k) = c_r$, így R/c_r darabszámú futásra számítunk, amely $(R/c_r)c_c(1)$ költséget okoz a kiszolgáló oldalán. Ezt folytatva juthatunk el a 3. táblázathoz. Ez a táblázat azt mutatja meg, hogy mik a költségek a szerver oldalán a különböző tiszta stratégiák szerint. A 3. táblázatot szokásosan a játék mátrixának hívjuk. A mátrixot jelen példánkban M-mel jelöljük, 3 sora és 2 oszlopa van.

		S_1	S_2
		x_1	x_2
A_1	y_1	$M_{11} = c_c(1) \cdot \frac{R}{c_r}$	$M_{12} = c_c(2) \cdot \frac{R}{c_r}$
A_2	y_2	$M_{21} = \big(c_c(1) + c_v(1)\big) \cdot \frac{R}{c_r + c_g}$	$M_{22} = \big(c_c(2) + c_v(2)\big) \cdot \frac{R}{c_r + c_g}$
A_3	y_3	$M_{31} = \big(c_c(1) + c_v(1) + c_e\big) \cdot \frac{R}{c_r + c_a(1) + c_p}$	$M_{32} = \big(c_c(2) + c_v(2) + c_e\big) \cdot \frac{R}{c_r + c_a(2) + c_p}$

3. táblázat. A játék mátrixa

2.2. Kevert stratégiák

Kevert stratégiák esetében a játékosok saját stratégiájukat egy valószínűségeloszlással adják meg, melyet a kiszolgálónál $X = \{x_1, x_2\}$, míg a támadónál $Y\{y_1, y_2, y_3\}$ segítségével jelölünk. Ha bármelyik valószínűség 1 értéket vesz fel, úgy tiszta stratégiáról beszélünk, minden egyéb eset kevert stratégia.

A játék résztvevői a játékot a következő módon kívánják optimalizálni: A kiszolgáló megpróbálja minimalizálni az átlagos veszteség maximumát, ahol a maximum a támadó stratégiájától függ (minimax szabály).

Ha a kiszolgáló kevert $X = \{x_1, x_2\}$ stratégiát választ, míg a támadó a j. tiszta stratégiát választja, úgy a kiszolgálónál okozott veszteség, költség a következő módon adható meg:

$$M_{j1} \cdot x_1 + M_{j2} \cdot x_2, \qquad j = 1, 2, 3.$$

Hasonlóképpen, ha a támadó kevert stratégiát választ, ahol a valószínűségeloszlást $Y = \{y_1, y_2, y_3\}$ adja meg, a kiszolgáló tiszta stratégiája pedig k, úgy az okozott költség

$$M_{1k} \cdot y_1 + M_{2k} \cdot y_2 + M_{3k} \cdot y_3, \qquad k = 1, 2.$$

A minimax stratégiát úgy tudjuk megállapítani, ha megkeressük azokat az $X^* = \{x_1^*, x_2^*\}$, és $Y^* = \{y_1^*, y_2^*, y_3^*\}$ valószínűségeloszlásokat, valamint azt a v^* értéket (a játék értéke) amelyek a következő két egyenlőtlenségrendszer megoldását jelentik:

A kiszolgáló számára:

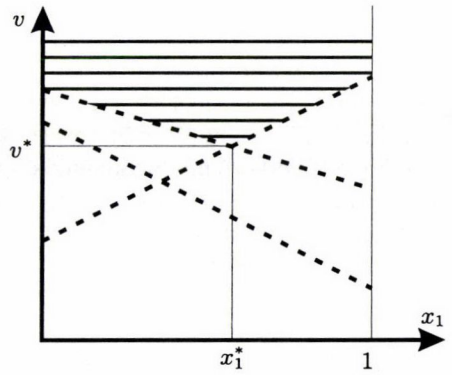
$$\begin{aligned} M_{11} \cdot x_1 + M_{12} \cdot x_2 &\leq v \\ M_{21} \cdot x_1 + M_{22} \cdot x_2 &\leq v \\ M_{31} \cdot x_1 + M_{32} \cdot x_2 &\leq v \\ x_1,\ x_2 &\geq 0 \\ x_1 + x_2 &= 1 \end{aligned}$$

A támadó számára:

$$\begin{aligned} M_{11} \cdot y_1 + M_{21} \cdot y_2 + M_{31} \cdot y_3 &\geq v \\ M_{12} \cdot y_1 + M_{22} \cdot y_2 + M_{32} \cdot y_3 &\geq v \\ y_1,\ y_2,\ y_3 &\geq 0 \\ y_1 + y_2 + y_3 &= 1 \end{aligned}$$

Az optimális stratégia kiválasztása

A kiszolgálóra vonatkozó egyenletrendszert a 4. számú ábrán szemléltetjük, ahol a szaggatott vonalak a rendszer három egyenlőtlenségéhez tartoznak. A minimax szabály szerint az optimális megoldáshoz a szerver kiválasztja a minimális átlagos veszteséget, azaz a legalacsonyabb pontot az árnyékolt területen, amely $[x_1^*, v^*]$. A támadó optimalizációs lépése hasonlóképpen adható meg.



4. ábra. A kiszolgáló optimális stratégiájának meghatározása

Neumann klasszikus tétele ([8]) szerint létezik egy közös optimum a támadó és a kiszolgáló számára. Az ehhez tartozó x_1^*, x_2^*, y_1^*, y_2^*, y_3^*, v^* paramétereket a játék megoldásának hívjuk.

Ha $M_{21} > M_{11}$ és $M_{22} > M_{12}$, akkor a mátrix második sora dominálja az első sort. Ez utóbbi akkor történhet meg, ha

$$\frac{c_c(l) + c_v(l)}{c_r + c_g} > \frac{c_c(l)}{c_r}, \qquad l = 1, 2. \tag{1}$$

A (1) egyenlőtlenség egyszerűsíthető:

$$\frac{c_v(l)}{c_g} > \frac{c_c(l)}{c_r}.$$

Annak a költsége, hogy egy szolgáltatás kérést küldünk a kiszolgálónak, illetve a helytelen megfejtés elküldése egy átlagos kommunikációs lépés költségének feleltethető meg, azaz gyakorlatilag nincs számítási és tárolási költség. Ennek megfelelően a $c_g = c_r$ közelítés alkalmazható. Az előzőekben említett dominancia akkor következik be tehát, ha a rejtvény ellenőrzésének költsége nagyobb, mint a rejtvény elküldésének költsége. Ha a rejtvény ellenőrzésének költsége kisebb, mint a rejtvény elküldésének költsége, úgy $M_{11} > M_{21}$ és $M_{12} > M_{22}$ és a mátrix első sora dominálja a másodikat. A továbbiakban az egyszerű leírás kedvéért ezt a tipikus dominancia esetet feltételezzük.

A dominanciát a 4. ábrán is illusztráljuk. Legyen a dominanciában levő sor a j. sor. Az $M_{j1} \cdot x_1 + M_{j2} \cdot (1 - x_1)$ és $M_{31} \cdot x_1 + M_{32} \cdot (1 - x_1)$ egyenesek egymást az $x_1 = 1/(1 + w)$ pontban metszik, ahol

$$w = \frac{M_{j1} - M_{31}}{M_{32} - M_{j2}}.$$

Így a következő eredményre juthatunk:

A játék megoldásai

Ha $w > 0$, akkor a kiszolgáló optimális stratégiája egy kevert stratégia, ahol a valószínűségeloszlás

$$\left[x_1^* = \frac{1}{1 + w}, x_2^* = 1 - x_1^*\right]$$

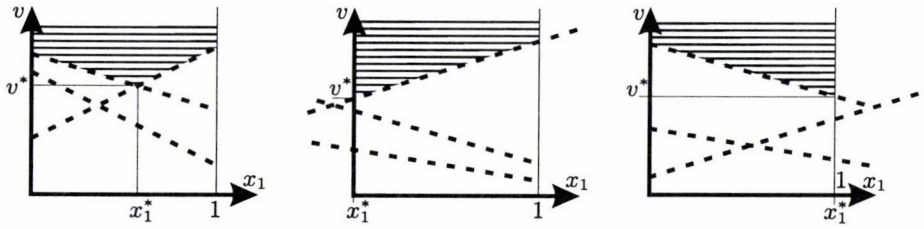
Ha $w \leq 0$, akkor a kiszolgáló a következő szabályok szerint választja ki tiszta stratégiáját:

- Ha $-1 < w \leq 0$, akkor $x_1^* = 1$, a kiszolgáló optimális stratégiája S_1,
- Ha $w \leq -1$, akkor $x_1^* = 0$, azaz az S_2 stratégia kerül kiválasztásra.

A kiszolgáló költsége nem haladhatja meg a

$$v^* = \frac{M_{21} - M_{22} \cdot w}{1 + w}$$

értéket a támadó stratégiájától függetlenül.



5. ábra. A játék megoldásainak szemléltetése

A játék megoldásának különböző eseteit az 5. ábrán szemléltetjük.

Ha a kiszolgáló tiszta stratégiát követ, úgy a támadó is tiszta stratégiát fog követni. Ez formálisan abból következik, hogy ha a kiszolgálóra felírt egyenlőtlenségrendszerben található egyenlőtlenségek egyike szigorú egyenlőtlenséggé válik (azaz $<$ a $\leq$ helyett) az X^* és v^* behelyettesítésekor, úgy az ehhez tartozó támadási stratégia valószínűségi súlya nullává válik. Ha $x_1^* = 1$ és $M_{21} = v^*$ és $M_{31} < v^*$, akkor $Y^* = \{0, 1, 0\}$, és ha $M_{21} < v^*$ és $M_{31} = v^*$, akkor $Y^* = \{0, 0, 1\}$. Hasonlóképpen, ha $x_1^* = 0$, $M_{22} = v^*$ és $M_{32} < v^*$, akkor $Y^* = \{0, 1, 0\}$ és ha $M_{22} < v^*$ és $M_{32} = v^*$, akkor $Y^* = \{0, 0, 1\}$.

Ha a kiszolgáló optimális kevert stratégiát folytat, akkor a támadó optimális stratégiája is kevert lesz. A támadó véletlenszerűen választ az A_2 és A_3 stratégiák közül. y_1^* értéke 0 lesz, mivel a második egyenlőtlenség -miként ezt már fentebb feltételeztük- dominálni fogja az elsőt.

Általánosságban beszélve, több stratégia esetén egyenlőtlenségek komplex rendszerét kell megoldani az optimális stratégia megkereséséhez. Szerencsére a megoldáshoz felhasználhatóak a lineáris programozás eszközei.

2.3. A játék érzékenysége

Az érzékenység analízist (másként stabilitási analízist) használhatjuk fel arra, hogy megállapítsuk, egy bemeneti változó megváltoztatása mennyire van hatással a kimeneti változó értékére, miközben minden más változatlan marad. Természetesen a vizsgálat eredménye jelentős mértékben függ a paraméterek kiválasztásától.

Jelen vizsgálatunkat a következő paraméter-értékekkel végezzük:

$c_r = 1c_c(1) = 14c_c(2) = 16c_g = 1c_a(1) = 40c_a(2) = 400c_v(1) = 10c_v(2) = 12c_p = 10c_e = 3000R = 100000$. A költségek modellezésére a „0.01 Ghz processzor másodperc" mértékegységet választottuk.

Ez az elméleti rendszer a következő játékra vezet:

$$\begin{aligned} x_1 + x_2 &= 1 \\ y_1 + y_2 + y_3 &= 1 \\ 14000 \cdot y_1 + 12000 \cdot y_2 + 59294 \cdot y_3 &\geq v \\ 16000 \cdot y_1 + 14000 \cdot y_2 + 7367 \cdot y_3 &\geq v \\ 14000 \cdot x_1 + 16000 \cdot x_2 &\leq v \end{aligned}$$

$$12000 \cdot x_1 + 14000 \cdot x_2 \leq v$$
$$59294 \cdot x_1 + 7367 \cdot x_2 \leq v$$

A játék megoldása pedig $v^* = 15679.8$, $x_1^* = 0.16$, $x_2^* = 0.84$, $y_1^* = 0.96$, $y_2^* = 0$, $y_3^* = 0.04$. Az optimális stratégia itt tehát kevert stratégia. Megvizsgáltuk a költségváltozók módosítását -2% és $+2\%$ között. A rendszer magas érzékenységet mutatott R értékére $(\pm 3,08\%)$, továbbá c_r $(\pm 3,14\%)$ és $c_c(2)$ $(2,6\%)$ értékekre. Alacsony viszont az érzékenység $c_c(1)$ $(0,43\%)$ és c_e $(0,12\%)$ paraméterekre.

3. Példa kliens oldali rejtvényre

Legyen $T = \{p_1, p_2, \ldots, p_N\}$ egy minden fél által ismert halmaza N prímszámnak. A kiszolgáló ebből a halmazból kiválaszt egy S részhalmazt, mely k prímet tartalmaz. A kiválasztás visszatétel nélkül történik, de a megoldás kiterjeszthető visszatételes esetre is. A kiszolgáló kiszámolja a kiválasztott elemek szorzatát, jelöljük ezt m segítségével.

$$m = p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_k}.$$

A rejtvényt a következő két változatban definiáljuk:

- *1. rejtvény:* A rejtvény tartalma m értéke.
- *2. rejtvény:* Legyen m' az m szorzat módosítása oly módon, hogy ℓ egymás után következő bitet, azaz $m_r, m_{r+1}, \ldots, m_{r+\ell-1}$ biteket 0-ra cserélésük m bináris reprezentációjában. A rejtvényt ez esetben a kapott m' és a hozzá tartozó r pozícióval adjuk meg.

Minkét rejtvény esetén a feladat az m (vagy m') prímfaktorainak meghatározása, és a faktorok megfelelő indexeinek elküldése a szerver részére. (Feltételezzük, hogy T rendezett és eszerint indexelünk.)

Az 1. rejtvény esetén a kliens a következő módon tudja kiszámítani a választ: Legyen μ egy változó, melyet minden számítási lépésben frissítünk. Legyen $\mu = m$ a kezdeti érték. Az i. lépésben a kliens ellenőrzi, hogy p_i faktora-e μ-nek (és így m-nek is). Ha igen, úgy μ módosításra kerül, értéke $\frac{\mu}{p_i}$ lesz, a megtalált indexet eltároljuk; máskülönben μ értéke nem változik. Ez az eljárás ismételhető, míg m minden faktorát meg nem találjuk (azaz μ eléri az 1 értéket).

A fenti számítás esetén $k-1$ darab osztás kerül elvégzésre[2], az osztók n bit méretű prímszámok, az osztandó pedig fokozatosan csökken kn méretről $2n$ méretig. Az osztások átlagos számára a következő képlet írható fel

$$D(N,k) = \sum_{i=k}^{N} q_i \cdot (i-1),$$

[2]Az utolsó prímfaktort már nem kell ellenőrizni osztással.

ahol q_i annak valószínűsége, hogy S legnagyobb indexe i, amelyet a következő módon kaphatunk meg:

$$q_i = \frac{\binom{i-1}{k-1}}{\binom{N}{k}}.$$

Ha k növekszik, úgy $D(N,k)$ gyorsan nő N felé, $D(N,k)$ már viszonylag kis k értékek esetén is közel kerül N értékhez. A 4. táblázatban feltüntetjük $D(N,k)$ értékeit $N = 100$ esetén.

k	4	8	16	32	64
$D(100,k)$	80,8	89,8	95,1	97,9	99,4

4. táblázat. $D(N,k)$ értékei, ha $N = 100$

A 2. rejtvény esetén a klienst arra kényszerítjük, hogy több számítást végezzen el. A kliens a következő két számítási eljárás közül tud választani:

- A kliens kipróbálja a kitörölt bitek összes helyettesítését. Ha egy helyettesítés nem megfelelő, úgy valószínűleg olyan prímfaktorokat fog kapni, melyek nem T részei. Ilyen esetben a kliens egy újabb helyettesítést tud elvégezni. Az osztások átlagos száma kb. $N2^{\ell-1}$, tehát a rejtvény megfejtésének nehézsége átlagosan $2^{\ell-1}$ faktorral növekedett. Ha például $N = 100$ és $\ell = 6$, úgy az átlagosan szükséges osztások száma legalább 3200.
- A kliens úgy is eljárhat, hogy kiszámolja T halmaznak k véletlenszerűen kiválasztott elemét és ellenőrzi szorzatukat, hogy az m'-t adja-e. Ezt m' megkapásáig folytatja. A szükséges szorzások átlagos száma

$$\frac{1}{2}\binom{N}{k}(k-1).$$

Példaként, ha $N = 100$ és $k = 8$, akkor megközelítően $6{,}51 \cdot 10^{11}$ szorzásra van szükség.

Ha a rejtvény állapot nélküli módon kerül megvalósításra ([1]), akkor védett lehet a memória elfoglalását célzó támadások ellen is. Amennyiben állapottal rendelkező módon valósítjuk meg a rejtvényt, akkor a kiszolgáló memóriájában a következő elemeket kell tárolni: Kliens IP, időbélyeg, S elemeinek indexei. Ez nagyságrendileg 300-400 bitet vehet igénybe kapcsolatonként, ami akár 20000 kapcsolat kezelését teszi lehetővé egy 1 megabájt méretű tároló használatával.

A kiszolgáló a költségparamétereket ismeri, így be tudja állítani a rejtvény nehézségét egy megfelelőnek vélt értékre. Ezáltal a kiszolgáló befolyásolni tudja az optimális stratégiát. Természetesen a nehézség növelésével a legitim klienseknek is egyre több számítást kell elvégezniük a szolgáltatás igénybevételéhez, ami korlátot jelent a rejtvény nehézségének növelésekor.

4. Összegzés

Jelen cikkben a kliens oldali rejtvény (client side puzzle) megoldást vizsgáltuk, erőforrásokat elfoglaló szolgáltatás-megtagadásos támadások ellen. A támadást a DoS támadó és a kiszolgáló közötti kétszereplős játékként modelleztük. Analizáltuk a játékot és betekintést nyújtottunk a kliens oldali rejtvény működési mechanizmusába. Fő hozzájárulásunk a témához az optimális stratégia kiválasztásának bemutatása a különböző esetekben. Legjobb tudásunk szerint a kliens oldali rejtvényeknek még nem készült hasonló játékelméleti vizsgálata. Bemutattunk és analizáltunk két konkrét, alkalmazható kliens oldali rejtvényt.

Hivatkozások

[1] Aura, T. and Nikander, P., Stateless protocols, in *Proceedings of the ICICS'97 Conference,* Springer-Verlag, LNCS volume 1334 (1997).

[2] Dwork, C. and Naor, M., Pricing via processing or combatting junk mail, in: *Advances in Cryptology - Crypto '92,* Springer-Verlag, LNCS volume 740 (August 1992), pp. 139–147.

[3] Eronen, P., Denial of service in public key protocols, in: *Proceedings of the Helsinki University of Technology Seminar on Network Security* (Fall 2000), December 2000.

[4] Gil, T. M., MULTOPS: A data structure for denial-of-service attack detection, Technical Report, Vrije Universiteit (2000).

[5] Juels, A. and Brainard, J., Client puzzles: A cryptographic countermeasure against connection depletion attacks, in: *Proceedings of the IEEE Network and Distributed System Security Symposium* (NDSS'99) (February 1999), pp. 151–165.

[6] Leiwo, J., Aura, T. and Nikander, P., Towards network denial of service resistant protocols, in: *Proceedins of the IFIP SEC 2000 Conference* (August 2000).

[7] Matsuura, K. and Imai, H., Protection of authenticated key-agreement protocol against a denial-of-service attack, in: *Proceedings of the International Symposium on Information Theory and Its Applications* (ISITA'98) (October 1998), pp. 466–470.

[8] von Neumann, J. and Morgenstern, O., *Theory of Games and Economic Behavior* (1924).

[9] Osborne, M. and Rubenstein, A., *A Course on Game Theory,* MIT Press (1994).

[10] Park, K. and Lee, H., On the effectiveness of probabilistic packet marking for IP Traceback under denial of service attack, Technical Report No. CSD-00-013, Department of Computer Sciences, Purdue University (June 2000).

BENCSÁTH BOLDIZSÁR, VAJDA ISTVÁN
LABORATORY OF CRYPTOGRAPHY AND SYSTEMS SECURITY (CRYSYS)
HÍRADÁSTECHNIKAI TANSZÉK
BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
e-mail: {bencsath, vajda}@crysys.hu

A GAME THEORETICAL MODEL OF DENIAL OF SERVICE ATTACKS ON INTERNET

BOLDIZSÁR BENCSÁTH, ISTVÁN VAJDA

Protection of cryptographic protocols against Denial of Service (DoS) penetration attack is considered. DoS attacks are modelled as strategic games. In this model the attacker tries to maximize the loss of resources suffered by the server, while the server wants to minimize this loss and (keep the quality of service for its clients) remain available for clients. This approach is presented in detail and applied for optimization of the client side puzzle technique. The optimal mixed strategy is shown.

Alkalmazott Matematikai Lapok **23** (2006), 349–361.

GEOMETRIAI KÓDOK

CSIRMAZ LÁSZLÓ, KATONA GYULA O. H.

Budapest

Az elmúlt évek egyik fontos kutatási területe a kriptográfiai műveleteket végző, illetve azok végrehajtását segítő fizikai objektumok vizsgálata. Például bizonyos fizikai objektumok speciális tulajdonságait kihasználva azokkal egyirányú függvényeket lehet számoltatni. A cikkben a lehető legegyszerűbb esetet tekintjük, nevezetesen, amikor egy kriptográfiai kulcsot (bitsorozatot) kívánunk előállítani a mérési eredményből. A fizikai objektum ebben az esetben úgy működik, mint egy valódi kulcs, azzal az igen hasznos további tulajdonsággal, hogy lehetetlen még egy példányban előállítani. Különböző alkalmakkor mérve persze az eredmények különbözők lesznek, a mérésből előálló bitsorozatnak azonban mindig ugyanannak kell lennie. Egy matematikai modellt állítunk fel, és a modell segítségével megvizsgáljuk, mik a fontos paraméterek és a paraméterek értékei között milyen összefüggésnek kell fennállnia, hogy a megfelelő mennyiségű és minőségű kriptográfiai kulcsot ki tudjuk nyerni.

1. Bevezetés

Különböző fizikai objektumok speciális tulajdonságait ügyesen használva azok segítségével kriptográfiai műveletek tudunk elvégeztetni. Ilyen módszereket alaposan vizsgáltak [4, 9], sőt az egyik eredmény még a Science folyóiratba is bekerült [10]. Mi arra az egyszerű esetre koncentrálunk, amikor egy fizikai objektum – amit kriptográfiai *bélyeg*-nek is szokás nevezni – azonosításra szolgál, egyediséget, hitelességet garantál. Ilyen bélyeget például egy DVD-re lehet ragasztani úgy, hogy a DVD tartalmát a bélyegből kivonható kulccsal titkosítják. Mivel a bélyeg előállítása fillérekbe kerül, másolása pedig csak hihetetlenül költséges eljárásokkal lehetséges, ha lehetséges egyáltalán, egy ilyen megoldás megoldaná a DVD-k másolásával kapcsolatos gondokat. Papír alapú dokumentumok eredetiségét lehet igazolni a papírra erősített bélyeggel, ha a bélyegből kivonható kulcsot és a dokumentum tartalmát a kibocsátó például digitálisan aláírja. Bélyeget bankkártyára téve meg lehet aka-

dályozni annak másolását, feltéve, ha a kártya csak a bélyegből kivont kulccsal tud dolgozni.

Kriptográfiai bélyeg szinte bármilyen fizikai objektum lehet, amit ismételten meg lehet mérni. Jó néhány szabadalom is létezik. Az [1] szabadalom mágneses szálakat használ, amiket egy vékony rétegre fröcsköltek fel. A méréséhez a bélyeget egy közönséges mágnesszalag olvasó előtt kell elhúzni. Az [5] szabadalom azt használja ki, hogy amikor egy papírlapot nagy fényerejű lámpa előtt elhúzunk, különböző sötétebb-világosabb foltok jelennek meg. A [6] apró vezető részecskéket használ, amik egy szigetelő rétegben vannak eloszlatva, és mikrohullámú berendezést használ az információ kiolvasásához. A szabadalmak áttekintése megtalálható a [9]-ben, ahol azt a lehetőséget vizsgálják, amikor a bélyeg átlátszó epoxigyantába ágyazott üveggömbökből áll. Papír alapú tárgyaknál a bélyeg állhat a papírba ágyazott speciális szálakból, a mérés történhet például ultraibolya fényben, amiben ezek a szálak világítanak. És persze a közönséges ujjlenyomat, íriszkép, tenyér-, illetve hanglenyomat is tekinthető kriptográfiai bélyegnek.

A mérés eredménye egy sor digitalizált adat, amit egy szkenner vagy egyéb speciális készülék szolgáltat. Az adat különböző szűrőkön, simításon, levágáson megy keresztül, esetleg további ravasz adatfeldolgozási technikákat is használnak. Ez a feldolgozás lényegesen csökkenti az adat mennyiségét, de feltehetően megtartja az objektum fontos fizikai jellemzőit. Végezetül az előfeldolgozott adatból származtatják majd a kriptográfiai kulcsot.

A cikkben ennek a folyamatnak egyetlen részére koncentrálunk; nevezetesen arra, hogyan lehet a kriptográfiai kulcsot kinyerni. Ebből a szempontból az előfeldolgozást tekinthetjük akár a mérés részének, akár a kulcs kinyerésére szolgáló eljárás részének. Magát a fizikai objektumot azonosítjuk egy tökéletes mérés eredményével. Minden más (valódi) mérés a tökéletes méréstől különböző, de hozzá „elegendően közeli" eredményt ad, bármit is jelentsen ez. Várakozásainknak megfelelően ugyanannak az objektumnak a mérései ugyanazt a kulcsot kell adják, vagyis elegendően közeli mérések eredményei ugyanazt a kivonatot kell eredményezzék.

Képzeljük el, hogy egy mérés eredménye mondjuk egy egymillió képpontból álló digitalizált szürke kép. Minden pixelben ismerjük az intenzitást, ami egy 0 (fekete) és 1 (fehér) közötti valós szám. Ekkor az egész kép tekinthető az 1 millió dimenziós Euklideszi tér egy vektorának. Két kép „elegendően közel van" egymáshoz, ha mondjuk pixelenként az intenzitások átlagos eltérése kisebb mint 1 százalék. Más szavakkal a vektorok közötti távolságot L_1 normában mérjük, és a még megengedett távolság 0,01-szer 1000000, vagyis 10000. Pixelek helyett a képet átranszformálhatjuk a frekvenciatartományba. Képek hasonlóságát jobban méri az, hogy mennyire van közel a frekvenciaképük. Mind az apró, mind a nagy méretű, teljes képre kiterjedő szisztematikus hibák könnyen kiszűrhetők, ha a magas, illetve alacsony frekvenciájú összetevőket kis súllyal vesszük figyelembe. Diszkrét frekvenciaértékeket választva a képet megint valós számok sorozataként – tehát vektorként – írhatjuk le, és a képek távolsága ebben a térben is valamilyen norma.

Modellünkben feltesszük, hogy a mérés valamilyen M *fázistér* egy pontját adja vissza. A mérések „közelségét" egy távolságfüggvény adja meg, tehát M-ben van metrika. A fizikai objektumokat (bélyegeket) az ideális mérés eredményével azonosítjuk, tehát a fizikai objektumokat is M elemeinek tekintjük. Mivel egy bélyeg valamilyen értelemben véletlen objektum, ennek modellezéséhez a fázistérben egy μ mértékre is szükség van: a lehetséges bélyegek egy A részhalmazára $\mu(A)$ adja meg annak a valószínűségét, hogy egy véletlenszerűen választott bélyeg A-ba esik.

Ha ismerjük egy mérés eredményét, vagyis M egy p pontját, akkor valamilyen tovább nem részletezett eljárás megmondja, mi a hozzá tartozó kriptográfiai kulcs. Minden egyes k kulcsra legyen A_k azoknak a pontoknak a halmaza, amikhez a k kulcs tartozik. Amikor a $p \in M$ objektumot megmérjük, akkor az eredmény egy $p' \in M$, ami elegendően közel van p-hez, vagyis p és p' távolsága kisebb vagy egyenlő mint valamilyen pozitív ε. A p'-höz tartozó kulcsnak persze ilyenkor ugyanannak kell lennie mint a p-hez tartozó kulcsnak.

Az A_k halmazok egyértelműen meghatározzák, hogyan kell a mérésekhez a kulcsokat hozzárendelni. Egy ilyen rendszer tulajdonságai szoros kapcsolatban vannak az A_k halmazok geometriai tulajdonságaival, ezért az A_k halmazok családját, amint k végigfut a lehetséges kulcsokon, *geometriai kód*-nak nevezzük.

A geometriai kódoknak három fontos paramétere van: az *előny*, a *biztonság* és a *hibatűrés*. Az *előny* egy $\alpha \geq 1$ valós szám, azt adja meg, hogy átlagosan hány véletlen bélyeget kell választanunk amíg egy érvényeset találunk, vagyis amíg a választott p pont beleesik valamelyik A_k halmazba. Ha az előny nagy, akkor várhatóan sok bélyeget kell eldobnunk amíg egy használhatót kapunk. Ha az előny közel van 1-hez, akkor majdnem az összes bélyeg használható. Az ideális esetben az előny értéke 1, amikor is mindegyik bélyeg (1 valószínűséggel) jó. A *hibatűrés* azt mondja meg, hogy egy méréskor mekkora hibát véthetünk. Ha ez az értéke ε és $p \in A_k$, akkor p-t megmérve az eredményül kapott p' és p távolsága legfeljebb ε lehet. Mivel p'-ből is ugyanazt a k kódot kell előállítanunk mint p-ből, ez azt jelenti, hogy az A_k halmazok ε sugarú környezetei diszjunktak kell legyenek. Végül a *biztonság* paraméter az A_k halmazok méretét korlátozza. Ez a σ paraméter mondja meg, hogyha véletlenszerűen választunk egy p érvényes bélyeget, akkor a belőle előállítható k kulcsot legfeljebb σ valószínűséggel kaphatjuk meg.

Nézzük meg, hogy például ujjlenyomatok esetén mi felel meg a modellben található fogalmaknak. Az [8] összefoglaló műben ismertettek szerint az ujjlenyomatról elsőként egy nagyjából 300-szor 300-as szürke skálájú raszterkép készül. Ezt egy kisméretű optikai leolvasóval, vagy közvetlen kapacitásméréssel készíthetik el. A raszterképen elsőként meghatározzák az ujjlenyomatot meghatározó vonalakat, majd ezeket a vonalakat *vékonyítják*, amíg csak egy pixel szélesek nem lesznek. Ezután a *finom jellemzőket* keresik meg. Ezek azok a pontok, ahol egy vonal véget ér, vagy ahol egy vonal kettéágazik (Y-pont). Egy ujjlenyomaton ezek száma néhányszor tíz és száz között változik. Minden ilyen jellemzőről a következő információkat tárolják: a pont típusa (végződés vagy elágazás), a pont helye, illetve a vonal iránya. Az így kapott jellemzőket még szűrik (például a túl közeli, hasonló irányú

végződéseket törlik), hogy csökkentsék az olvasás és a feldolgozás során elkövetett hibák hatását.

Mondjuk az ujjlenyomat alapján egy legalább 80 bites fix számot szeretnénk előállítani, és ehhez az ujjlenyomat első húsz jellemzőjét használjuk. Az M *fázistér* a húsz jellemző megadásához szükséges adatokat tartalmazza:

$$\left(\{\mathrm{I},\mathrm{Y}\}\times[0,1]^2\times[0,\pi]\right)^{20}.$$

Az első tényező a jellemző típusa (végződés vagy elágazás), a következő a pont két koordinátája, az utolsó pedig az irány; mindezt hússzor ismételjük meg. A *mérték* azt mondja meg, hogy egy véletlen ujjlenyomat mekkora valószínűséggel esik az adott részbe. Az egyszerűség kedvéért a teljes téren a szorzat mértéket választjuk. (Ez annak a feltételezésnek felel meg, hogy az egyes jellemzők függetlenek – ami természetesen nem igaz, hiszen az egyes jellemzők nem kerülhetnek túlságosan közel egymáshoz.) Mivel a jellemzőket mindig elhelyezkedésük alapján sorba rakjuk, azért az M térnek csak egy T részét használjuk fel. A mértéket úgy kell normálni, hogy ennek a T-nek éppen 1 legyen a mértéke.

Az egyes tényezőkben mind a hely, mind az irány bármi lehet, azaz ezekre a szokásos Lebesgue mértéket használjuk. A két elemű $\{\mathrm{I},\mathrm{Y}\}$ halmazon a mértéket annak megfelelően állapítjuk meg, hogy tipikusan egy ujjlenyomaton a jellemzők között hány százalék a végződés, és hány százalék az elágazás.

A *távolság* azt fejezi ki, hogy ugyanannak az ujjnak két különböző méréséből származó adat mennyire van közel. Ha valamelyik tényezőben a típus különbözik, akkor a távolság nagyon nagy – mondjuk 1000 –, egyébként mind a hely, mind az irány különbsége kicsi legyen. A teljes távolság az egyes tényezőkben mért távolságok maximuma lehet.

Ha figyelembe akarjuk venni, hogy a kép elfordulhat, akkor az M fázistér helyett vehetjük azt a faktorteret, melyben M két elemét azonosítjuk, ha azok a négyzet egy elforgatásával/eltolásával egymásba vihetők. Mértékként használhatjuk a faktormértéket (ami persze most már nem lesz egyenletes), a távolság pedig lehet az inverz képek távolságának infimuma. Hasonlóan kezelhetjük azt az esetet amikor egy jellemző kimaradása esetén is „közelinek" szeretnénk elfogadni a két mérési eredményt.

A σ *biztonságot* 2^{-80}-ra választjuk. Ezzel biztosítjuk, hogy a generált számok közül mindegyiket legfeljebb ekkora valószínűséggel kapjuk meg (feltéve hogy a mérték nagyjából egyenletes). Az α *előny* reciproka azt mondja meg, hogy az összes lehetséges mérési eredmény közül legfeljebb mekkora mértékű lehet az, amihez nem tartozik eredmény. Ha valakinek az ujjlenyomata ebbe a kivételes halmazba esne, akkor annak nem tud a rendszer kódszámot generálni. Azt reméljük, hogy az fázistér viszonylag kis része áll elő valódi ujjlenyomatból, így $\alpha = 2$ reális választásnak tűnik.

A *hibatűrés* értékét a legnehezebb előre megmondani. Miután definiáltuk a távolságot, több méréssorozat alapján megbecsülhetjük hogy ugyanarról az ujjról kapott jellemzők sorozata mekkora távolságra kerülhet el.

A 2. részben a pontos definíciókat adjuk meg, kimondjuk és bizonyítjuk a fő eredményünket, ami a geometriai kódok három paramétere ad szükséges feltételt. A 3. részben példákat adunk geometriai kódokra, amik megmutatják hogy a tételben szereplő feltétel konstans szorzó erejéig szükséges is bizonyos fontos esetekben. Megvizsgáljuk, hogy a fázistér milyen tulajdonságai szükségesek. Végül az utolsó részben összefoglaljuk az eredményeket.

2. A geometriai kód

Az M fázistér egy μ mértékkel ellátott tér, amin még egy $d(x,y)$ távolság is van. A p pont körül ϱ sugarú (nyílt) gömb azoknak az M-beli pontoknak a halmaza, melyek ϱ-nál közelebb vannak p-hez. Ezt a gömböt $p+\varrho$-val fogjuk jelölni:

$$p+\varrho \stackrel{\text{def}}{=} \{x \in M : d(x,p) < \varrho\}.$$

Az M egy tetszőleges A részhalmazára A-nak ϱ *sugarú környezete* a $p+\varrho$ gömbök uniója, ahol p végigfut A elemein:

$$A+\varrho \stackrel{\text{def}}{=} \bigcup\{p+\varrho : p \in A\}.$$

Ez nem más, mint az M olyan pontjainak halmaza, amik A valamelyik eleméhez ϱ-nál közelebb vannak. $A+\varrho$ az eredmény, ha A-t ϱ-val *meghízlaljuk*. A háromszögegyenlőtlenség miatt ha $A+\varrho_1$-et meghízlaljuk ϱ_2-vel, akkor az eredmény biztosan benne van $A+(\varrho_1+\varrho_2)$-ben. Azt mondjuk, hogy a metrika *lapos*, ha ez a tartalmazás pozitív ϱ_1 és ϱ_2 esetén soha sem valódi, vagyis ha minden pozitív ϱ_1 és ϱ_2 mellett

$$(A+\varrho_1)+\varrho_2 = A+(\varrho_1+\varrho_2). \tag{1}$$

Természetesen fel kell tennünk, hogy minden gömb mérhető a μ mérték szerint. Ennél azonban többet követelünk meg, nevezetesen azt, hogy a mérték *homogén* legyen, vagyis az azonos sugarú gömbök mind ugyanakkor mértékűek. A ϱ sugarú gömb mértékét (vagyis térfogatát) $V(\varrho)$-val fogjuk jelölni. Még azt is feltesszük, hogy ez a V függvény folytonos, szigorúan monoton nő és minden értékkészlete a nem-negatív valós számok halmaza. Ha $A \subseteq M$ egy mérhető halmaz, akkor $r(A)$-val jelöljük annak a gömbnek a sugarát, aminek térfogata megegyezik A mértékével, vagyis

$$\mu(A) = V\big(r(A)\big).$$

Ezen kívül rögzítjük a fázistér egy egységnyi mértékű T részét is. A T elemei lesznek a mérések lehetséges kimenetelei. Ha $A \subseteq T$, akkor a $\mu(A)$ mértéket úgy értelmezzük, mint annak a valószínűségét, hogy a mérés eredménye A-ba esik.

Például M lehet az R^n-nek jelölt n dimenziós Euklideszi tér a szokásos távolsággal, T a térben az egységkocka, μ pedig a Lebesgue mérték. Ez a mérték annak felel meg, amikor a fázis tér pontjait egyenletes eloszlással választjuk ki.

1. Definíció. Egy m méretű *geometriai kód* a T-nek m darab mérhető részéből álló $\{A_k : 1 \le k \le m\}$ sorozat. A kód
biztonsága σ, ha minden k-ra $\mu(A_k) \le \sigma$;
előnye α, ha $\mu(\bigcup A_k) \ge 1/\alpha$; és
hibatűrése ε, ha A_k-nak az ε-környezete még mindig része T-nek és ezek az ε-környezetek páronként diszjunktak, vagyis $A_k + \varepsilon \subseteq T$, valamint $(A_i + \varepsilon) \cap (A_j + \varepsilon) = \emptyset$ ha i és j különbözőek.

A *biztonság* azt garantálja, hogy mindegyik kulcsot legfeljebb σ valószínűséggel kaphatjuk meg, feltéve hogy a bélyeget véletlenszerűen választjuk. A paraméter értékét 2^{-50} vagy annál kisebbre kell választani. Az *előny* az az érték, ahány bélyeget várhatóan generálnunk kell ahhoz, amíg egy használhatót (vagyis valamelyik A_k-ba esőt) kapunk. Ennek tipikus értéke 1,5 és 100 között kell legyen. Végül a *hibatűrés* azt mondja meg, hogy mekkora hibát követhetünk el a méréskor: ha egy bélyeg mondjuk A_k-ba esik, és megmérjük, akkor a mérési eredmény A_k-nak ε sugarú környezetébe esik. Azt akarjuk, hogy ebből a mérésből egyértelműen megállapíthassuk a bélyeghez tartozó kulcsot. Ezért ezeknek a környezeteknek diszjunktaknak kell lenniük.

A modellünkből azonnal adódik, hogy egy geometriai kód előnye csak akkor lehet 1, ha hibatűrése 0. Ha a mérés eredményeképpen bármiféle hibát megengedünk, akkor kell lennie T-ben olyan pontnak, amihez nem tartozik kulcs. Ez különösen problematikus, ha T valamilyen biometriai mérték – például ujjlenyomat, vagy íriszkép –, hiszen nem lehet valaki a rendszerből kizárni azért, mert „nem elég jó az ujjlenyomata."

Mivel a mérési eredményekhez a kriptográfiai kulcsokat az A_k halmazok segítségével rendeljük hozzá, az A_k halmazoktól még további tulajdonságokat is megkövetelhetünk. Például, hogy az összes A_k ugyanakkora, vagy majdnem ugyanakkora valószínűségű legyen, ami azt jelenti, hogy a generált kulcsot nagyjából egyenletesen eloszlásban kapjuk meg. Azután az sem árt, ha az A_k halmazok elég „egyszerűek", vagyis a mérési eredményből a kulcs előállítása ne legyen túlságosan nehéz.

A geometriai kódok definiáló tulajdonságai természetesen adódnak a modellezendő feladatból. Az egyetlen kivétel talán az a megszorítás, hogy az A_k halmazok ε környezete még mindig része legyen T-nek. Ez az egyszerűsítő feltétel áttekinthetőbbé teszi az eredményeinket és a bizonyítást is. Bizonyos esetekben lényeges különbséget jelenthet ez a megszorítás, erre is fogunk látni példát.

2. Definíció. Az M fázistér *Brunn–Minkowski tulajdonságú*, ha minden mérhető A halmazra

$$\mu(A + \varepsilon) \ge \mu\Big((p + r(A)) + \varepsilon\Big).$$

Más szavakkal egy A halmaz akkor hízik a legkevesebbet, ha az egy gömb. A bal oldalon az A halmazt hízlaltuk meg ε-nal, jobboldalon pedig az A-val megegyező térfogatú gömböt. Ha a metrika lapos, akkor az egyenlőtlenség a következő egyszerűbb alakba írható:

$$r(A + \varepsilon) \ge r(A) + \varepsilon.$$

A nevezetes *Brunn-Minkowski egyenlőtlenség* azt mondja ki, hogy az n-dimenziós Euklideszi tér a szokásos távolsággal és a Lebesgue metrikával Brunn–Minkowski tulajdonságú [2]. A tételnek számos általánosítása van például hiperbolikus terekre is.

Minden definíció a rendelkezésünkre áll, hogy kimondjuk és bizonyítsuk a fő tételt, természetesen nem a lehető legáltalánosabb formában.

1. TÉTEL. *Tegyük fel, hogy az M fázistér lapos, Brunn–Minkowski tulajdonságú, és a ϱ sugarú gömb térfogatát megadó $V(\varrho)$ függvény log-konkáv. Ekkor egy geometriai kód α, σ, ε paraméterei kielégítik az alábbi egyenlőtlenséget:*

$$\varepsilon \le V^{-1}(\alpha\sigma) - V^{-1}(\sigma). \tag{2}$$

Bizonyítás. Tegyük fel, hogy a geometriai kód az $A_1, \dots, A_m$ halmazokból áll. Legyen $r(A_k) = a_k$, vagyis a_k annak a gömbnek a sugara, aminek ugyanakkor a térfogata, mint A_k-nak. Mivel M lapos, azért a Brunn–Minkowski egyenlőtlenség második alakját alkalmazhatjuk, ami szerint $r(A_k + \varepsilon) \ge r(A_k) + \varepsilon = a_k + \varepsilon$. Más szavakkal, $A_k + \varepsilon$ legalább akkora térfogatú, mint az $a_k + \varepsilon$ sugarú gömb:

$$\mu(A_k + \varepsilon) \ge V(a_k + \varepsilon). \tag{3}$$

Legyen még a annak a gömbnek a sugara, aminek térfogata éppen σ, vagyis $a = V^{-1}(\sigma)$. Mivel a kód biztonsága σ, ez azt jelenti hogy $\mu(A_k) \le \sigma$, ahonnan $a_k \le a$ minden k-ra.

Mind $a_k + \varepsilon$, mint a eleme az $(a_k, a + \varepsilon)$ intervallumnak, továbbá ha $a_k + \varepsilon$ az intervallumot $\lambda : 1 - \lambda$ arányban osztja, akkor a ugyanezt az intervallumot $1 - \lambda : \lambda$ arányban osztja. Feltételünk szerint a $\log V$ függvény konkáv, tehát

$$\log V(a_k + \varepsilon) \ge (1 - \lambda) \log V(a_k) + \lambda \log V(a + \varepsilon),$$

$$\log V(a) \ge \lambda \log V(a_k) + (1 - \lambda) \log V(a + \varepsilon).$$

Ezeket összeadva és átalakítva kapjuk, hogy

$$V(a_k + \varepsilon) \ge \frac{V(a + \varepsilon)}{V(a)} \cdot V(a_k).$$

A (3) szerint a bal oldal értéke legfeljebb $\mu(A_k + \varepsilon)$. Az $A_k + \varepsilon$ halmazok a T diszjunkt részei, tehát mértékük összege nem haladhatja meg T mértékét, ami 1:

$$1 = \mu(T) \ge \sum_k \frac{V(a + \varepsilon)}{V(a)} \cdot V(a_k) = \frac{V(a + \varepsilon)}{V(a)} \sum_k V(a_k) \ge \frac{V(a + \varepsilon)}{V(a)} \cdot \frac{1}{\alpha},$$

itt kihasználtuk hogy $V(a_k)$ az A_k halmaz mértéke, ezért $\sum_k V(a_k) = \mu(\bigcup A_k) \ge 1/\alpha$. Ha még felhasználjuk, hogy $V(a) = \sigma$, éppen (2)-t kapjuk.

3. Példák

Első példánkban M az n dimenziós Euklideszi tér a szokásos távolsággal és a Lebesgue mértékkel. A kódok lehetséges T halmaza lehet például az egységkocka, vagy az 1 térfogatú gömb. A ϱ sugarú n-dimenziós gömb térfogata

$$V(\varrho) = \gamma_n \varrho^n \quad \text{ahol} \quad \gamma_n = \frac{\pi^{n/2}}{(n/2)!}.$$

Világos, hogy $V(\varrho)$ log-konkáv (hiszen $\log V(\varrho)$ lineáris függvény), és hogy R^n Brunn–Minkowski tulajdonságú (lásd [2]). A $k!$ értékét $(k/e)^k$-nal közelítve az 1. tétel egyenlőtlensége a következőképpen alakul:

$$\varepsilon \leq \sqrt{\frac{2\pi e}{n}}\, \sigma^{1/n} \left(\alpha^{1/n} - 1\right). \tag{4}$$

Ha n legalább tízszer akkora, mint $\log \alpha$, akkor a jobb oldal utolsó tényezője helyettesíthető a $(\log \alpha)/n$ kifejezéssel. Ha most α-t és σ-t fixen tartjuk, akkor a jobb oldal a legnagyobb értékét az $n \approx 0{,}66 \log(1/\sigma)$ helyen veszi fel, függetlenül α értékétől. Ezt (4)-be helyettesítve kapjuk, hogy

$$\varepsilon \leq \frac{2}{3} e^{2/3} \sqrt{2\pi e}\, \frac{\log \alpha}{\log(1/\sigma)} \approx 5{,}37\, \frac{\log \alpha}{\log(1/\sigma)}.$$

Ha például $\sigma = 2^{-50}$ és $\alpha = 1{,}5$, akkor az innen adódó korlát $\varepsilon \leq 0{,}01$, és n értékét 23 körül kell megválasztani.

Ebben a fázistérben tudunk olyan geometriai kódokat konstruálni, melyek a (4) korlátot jól megközelítik. A konstrukció a következőképpen működik. Vágjunk ki diszjunkt $\sigma^{1/n} + 2\varepsilon$ élhosszúságú kockákat T-ből, és mindegyik kis kockát zsugorítsuk össze a középpontjából annyira, hogy az élhossza $\sigma^{1/n}$ legyen. Az így kapott zsugorított kockák egy geometriai kódot alkotnak, melynek biztonsága σ (hiszen mindegyik ilyen kis kocka térfogata éppen σ), és hibatűrése ε. A kód előnye attól függ, hogy hány kis kockát tudunk T-ből kivágni. Az egyszerűbb számítás érdekében tegyük fel hogy T az egységkocka. Mivel a kis kockák élhossza $\sigma^{1/n} + 2\varepsilon$, T-ből legalább

$$\left[\frac{1}{\sigma^{1/n} + 2\varepsilon}\right]^n \geq \left(\frac{1}{\sigma^{1/n} + 2\varepsilon} - 1\right)^n$$

ilyen kockát tudunk kivágni. A kód előnye α, ha

$$\left(\frac{1}{\sigma^{1/n} + 2\varepsilon} - 1\right)^n \sigma \geq \frac{1}{\alpha},$$

ami fennáll, hogyha

$$\varepsilon \leq 0{,}5\, \sigma^{1/n} \left(\alpha^{1/n}\, \frac{1}{1 + (\alpha\sigma)^{1/n}} - 1\right). \tag{5}$$

Ha $(\alpha\sigma)^{1/n}$ elegendően kicsi, akkor ez a (4)-ből adódó elméleti korlát $10\sqrt{n}$-szerese.

Általában ha ∂T jelöli a T felszínét, akkor T-ből legalább

$$\frac{1-\eta\sqrt{n}\,\partial T}{\eta^n}$$

darab η élű kockát tudunk kivágni. Ha még feltesszük, hogy $x=\eta\sqrt{n}\,\partial T$ kisebb félnél, akkor $(1-x)^{1/n}$-t közelíthetjük $1-x/n$-nel, ami mutatja, hogy létezik α előnnyel geometriai kód, ha

$$\varepsilon \le 0{,}5\,\sigma^{1/n}\left(\alpha^{1/n}\frac{1}{1+\dfrac{\partial T}{\sqrt{n}}(\alpha\sigma)^{1/n}}-1\right). \tag{6}$$

A konstansszor $(\alpha\sigma)^{1/n}$ hibatag a *kerületi hiba*, ez abból adódik, hogy a T határához közeli pontokat nem tudjuk felhasználni. Ez az érték arányos T felszínével, és nullához tart, ha σ tart nullához. Az itt álló 0,5 konstans és a (4)-beli $\sqrt{2\varphi e/n}$ konstans közti eltérés a *pakolási hiba*. Ez abból adódik, hogy az n-dimenziós gömböket nem lehet szorosan pakolni.

Ha T éppen az egység térfogatú gömb, akkor a kerületi hiba $1+\sqrt{2\pi e}\,(\alpha\sigma)^{1/n}$. Ez például (6)-ból vagy közvetlenül is kiszámolható.

A Lebesgue mérték megtartásával más távolságfüggvényt is használhatunk R^n-ben. A távolságot legegyszerűbben egy *norma* segítségével definiálhatunk: az x és y pontok (mint vektorok) távolsága az $x-y$ különbség normája:

$$d(x,y) \stackrel{\text{def}}{=} \|x-y\|.$$

Az így definiált metrikus tér mindig lapos, és a „gömbök" konvex halmazok. Következésképp ezekben a fázisterekben szintén teljesül a Brunn–Minkowski tulajdonság, lásd [2]. A ϱ sugarú „gömb" térfogata $V(\varrho)=c\cdot\varrho^n$, ahol c a $B_1=\{x\in R^n : \|x\|<1\}$ egységgömb térfogata. Mivel a $V(\varrho)$ függvény most is log-konkáv, alkalmazhatjuk az 1. tételt, ami a következő elméleti korlátot adja:

$$\varepsilon \le \frac{1}{c^{1/n}}\,\sigma^{1/n}\bigl(\alpha^{1/n}-1\bigr). \tag{7}$$

Ha a maximum, vagyis L^∞-normát használjuk, akkor a „gömbök" éppen az n-dimenziós kockák lesznek. Az egység sugarú gömb ebben az esetben a kettő élhosszúságú kocka, tehát $c=2^n$. A (7) korlátban a konstans értéke éppen 0,5. Mivel kockákkal a tér hézag nélkül kitölthető, azt várjuk hogy létezzen olyan geometriai kód, melyben nincs pakolási hiba. A számítások mutatják, hogy ténylegesen is ez a helyzet: az (5) határ ugyanazzal a konstrukcióval most is elérhető.

Általában tetszőleges normából származó távolságfüggvény esetén is léteznek az elméleti (7) korlátot elég jól megközelítő konstrukciók. A B_1 „egységgömb" ebben az esetben szükségszerűen konvex. Mint minden konvex test, B_1 is befoglalható egy olyan téglatestbe, melynek térfogata legfeljebb $n!$-szorosa B_1 térfogatának. Legyen φ az az affin transzformáció, ami ugyanakkor térfogatú kockát csinál ebből a téglatestből. Ez a transzformáció a T alakzatot φT-be viszi. A T belsejében csupa kis téglalapból álló $(\alpha, \sigma, \varepsilon)$ paraméterű geometriai kódot tudunk konstruálni ha

$$\varepsilon \leq \frac{1}{(n!)^{1/n} c^{1/n}} \sigma^{1/n} \left(\alpha^{1/n} \frac{1}{1 + \frac{\partial \varphi T}{\sqrt{n}} (\alpha\sigma)^{1/n}} - 1 \right).$$

A kerületi hibától eltekintve a két korlát az $(n!)^{1/n} \approx n/e$ konstans erejéig megegyezik.

Láttunk példákat, ahol a fázistér pontjai az n-dimenziós Euklideszi tér vektorai voltak, és a távolságot tetszőleges normából definiáltuk. Ezek a terek rendelkeznek néhány természetes, elvárható tulajdonsággal; ezek közül négyet külön fölsorolunk:

(i) a távolság eltolásinvariáns, vagyis a és b között ugyanaz a távolság, mint $a + x$ és $b + x$ között minden x vektorra;
(ii) a tér lapos;
(iii) a távolság a szokásos Euklideszi topológiát generálja;
(iv) a mérték homogén, vagyis egyenlő sugarú gömbök ugyanolyan mértékűek.

További eredményünk, hogy az 1. tétel állítása igaz az összes, az n-dimenziós Euklideszi tér pontjain definiált fázistérben, melyek ezekkel a tulajdonságokkal rendelkeznek.

2. TÉTEL. *Tegyük fel, hogy az M fázistér az n-dimenziós Euklideszi tér pontjain van definiálva, és teljesíti a fenti* (i)–(iv) *tulajdonságokat. Ekkor ebben a térben minden geometriai kód paraméterei teljesítik az 1. Tételben felírt feltételt.*

A tétel bizonyításához elegendő megmutatni, hogy az (i)–(iv) feltételek biztosítják, hogy a mérték csak a Lebesgue-mérték lehet, a távolság pedig valamilyen normából származik. Ezekre a fázisterekre az 1. Tétel közvetlenül alkalmazható.

Tudunk példát mutatni arra, hogy a (ii) feltétel szükséges. Ebben a teret a kétdimenziós Euklideszi tér pontjain definiáljuk. A mérték a szokásos, a távolságot viszont speciálisan kellett megválasztani. Ez a tér nem lehet Brunn–Minkowski tulajdonságú.

4. Összefoglalás

A cikkben egy gyakorlati feladat során felmerült probléma absztrakt matematikai modelljét adtuk meg. A feladat az volt, hogy véletlen tulajdonságokkal

rendelkező fizikai objektumhoz rendeljünk fix kriptográfiai kulcsot, melyet az objektum megmérésével elő tudunk állítani. A modellnek három fontos paraméter volt: a *biztonság*, az *előny* és a *hibatűrés*. Igazoltuk e három paraméter között fennálló egyenlőtlenséget, ami a geometriai kód mögött található fázistér geometriai tulajdonságaiból következett.

Részletesen megvizsgáltuk azt az esetet, amikor a fázistér az n-dimenziós Euklideszi tér a szokásos Lebesgue mértékkel, de nem feltétlenül a szokásos L^2 távolsággal. Konstrukciókat adtunk meg, amik az elméleti korlátot bizonyos hibákkal érték el. Két típusú hibatagot azonosítottunk. A *kerületi hiba* abból adódik, hogy a geometriai kód nem tudja kihasználni a fázistér megengedett részének a határát. A *pakolási hiba* pedig azt méri, mennyire sűrűn lehet a teret „gömbökkel" kitölteni.

A konstrukcióknak még további szép tulajdonságai is voltak. Mindegyik kódban a halmazok ugyanakkora térfogatúak voltak, és mindegyik vagy kocka vagy téglatest volt. Az egyszerű struktúra miatt a kulcs kiszámítása hatékonyan történhet.

Megvizsgáltuk azokat a feltételeket, melyeket a fázistérről tettünk. Az egyik a nevezetes Brunn–Minkowski egyenlőtlenség általánosítása; a másik egy különös feltétel a távolságfüggvényre, nevezetesen hogy a térnek laposnak kell lennie. Igazoltuk, hogy ez utóbbi tulajdonságnak több érdekes következménye van. Az egyik, hogy az r sugarú gömb térfogata legfeljebb exponenciálisan nőhet. Egy másik, hogy az n-dimenziós Euklideszi téren definiált fázistereken a metrika laposssága bizonyos további homogenitási tulajdonságokkal együtt már biztosítja, hogy a geometriai kódokra vonatkozó (2) egyenlőtlenség fennálljon. Példát konstruáltunk olyan kellemes tulajdonságokkal rendelkező fázistérre, amelyik nem lapos, tehát ez egy nem triviális tulajdonság.

A kettő és többdimenziós hiperbolikus terek szintén teljesítik az 1. tétel feltételeit. Érdekes volna különböző konstrukciókat látni, hogy azok mennyire közelítik meg a (2) alatti elméleti határt. A hiperbolikus térben T határa összemérhető T területével, ezért jelentős különbségekre számítunk, ha előírjuk, hogy az A_i-k ε sugarú környezete része legyen T-nek, illetve ha ezt nem követeljük meg.

Nagyon sok tér rendelkezik a Brunn–Minkowski tulajdonsággal. Érdekes kérdés, hogy ez a tulajdonság megőrződik-e a szorzatra. Könnyű látni, hogy a valós számegyenes és mondjuk egy kétdimenziós hiperbolikus tér szorzata a maximum normával és a szokásos szorzat metrikával ellátva már nem Brunn–Minkowski tulajdonságú. Sikerült azt is bizonyítanunk, hogy sima M-beli térfogatfüggvény esetén az $R \times M$ szorzat pontosan akkor Brunn–Minkowski tulajdonságú, ha az M-beli $V(r)$ hatványfüggvény.

Egy másik természetes módon adódó modellben az objektum n különböző jellemzőjét mérjük meg. Ezeknek a mennyiségeknek az értékei alkotják azt az n dimenziós x vektort, amivel magát az objektumot azonosítjuk. A mérésnél minden egyes koordináta a valódi értéktől egy s szórású 0 várható értékű normális eloszlású hibával tér el, a hiba az egyes koordinátáknál független. Ha x' a méréskor kapott vektor, akkor az $x - x'$ különbség L^2 normája szintén egy s szórású, nulla várható

értékű normális eloszlású változó. Ha tehát azt akarjuk elérni, hogy az x' mérésből a k kulcsot p valószínűséggel egyértelműen azonosítani tudjuk, ennek feltétele pontosan az, hogy az A_k halmazok ε sugarú környezetei diszjunktak legyenek, ahol ε olyan, hogy egy s szórású, nulla várható értékű normális eloszlású véletlen változó $-\varepsilon$ és ε közé eső értéket éppen p valószínűséggel vegyen fel.

Vizsgálatainkban mindig arra az esetre koncentráltunk, amikor a kód biztonsága nullához tartott. Egészen más típusú, véges bináris kódokat használó konstrukciókra van szükség, ha $\log(1/\sigma)$ kisebb a tér dimenziójánál. Ha a tér dimenzióját növeljük, akkor a felszíntől legalább ε távolságra lévő pontok mértéke exponenciálisan tart a nullához, tehát egy idő után már nem létezhet rögzített σ biztonságú kód. Ha nem követeljük meg, hogy a kódhalmazok ε sugarú környezete is része legyen T-nek (csak azt, hogy páronként diszjunktak legyenek), akkor akármilyen nagy dimenzióban létezik ε hibatűrésű és σ biztonságú kód – feltéve persze, hogy ilyen kód egyáltalán létezik. Az intuíció azt sugallja, hogy a dimenzió növelésével egyre „sűrűbb" ilyen kódot tudunk gyártani, vagyis a kódok α előnye tart az 1-hez. Meglepő módon azt sejtjük, hogy ez nem így van, nevezetesen $2^{-n} \leq \sigma$ esetén n dimenzióban a legkisebb előnnyel rendelkező kódok a teljes tér egy legfeljebb $\log_2(1/\sigma)$ dimenziós alterében vannak. Így például, ha 50 bites kriptográfiai kulcsra van szükségünk, vagyis $\sigma = 2^{-50}$, akkor nem érdemes 50-nél több független paraméterét megmérni az objektumnak.

További érdekes kérdés, ha a folytonos terek helyett diszkrét tereket tekintünk. A [3] cikkben az n hosszúságú 0–1 sorozatok terén néztek több különböző távolságfüggvényt. Az alkalmazott modellben nem egy, hanem egyszerre 2^k különböző geometriai kódot konstruáltak egyszerre úgy, hogy a tér minden pontja valamelyik kódban benne legyen. Ezzel a módszerrel egyrészt elérték, hogy ne legyenek „selejtek," vagyis olyan elemei a fázistérnek, melyek egyik kódban sincsenek benne, másrészt olyan kódokat kellett konstruálni, melyekben az előny 1-hez közeli érték helyett 2^k körüli kellett legyen. Hasonló konstrukciók a folytonos esetben is vizsgálhatók.

Hivatkozások

[1] J. Brosow: Method and system for verifying authenticity safe against forgery, US patent no. 4218674, 1980.

[2] Yu. D. Burago, V. A. Zalgaller: *Geometric Inequalities*, Springer, 1988.

[3] Y. Dodis, L. Reyzin, A. Smith, *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data*, Lecture Notes in Computer Science, Vol. 3027, pp. 523–540, Springer, 2004.

[4] B. Gassend, D. Clarke, M. van Dijk, S. Devadas, *Controlled Physical Unknown Functions*, MIT LCS TR-845, 2002.

[5] R. Goldman: Verification system for document substance and content, US patent no. 4568936, 1986.

[6] G. Lippner, personal communication, 2002.

[7] A. J. Menezes, P. C. van Oorshot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[8] L. O'Gorman, *Overview of fingerprint verification technologies*, Elsevier Information Security Technical Report, Vol. 3, No. 1, 1998.

[9] Ravi Pappu, *Physical one-Way Functions*, PhD Thesis, March 2001.

[10] Ravi Pappu, Ben Recht, Jason Taylor, Neil Gershenfeld, *Physical One-Way Functions*, Science, 2002, September 20, 297, pp. 2026–2030.

[11] J. Samyn, Method and apparatus for checking the authenticity of documents, US patent no. 4820912, 1989.

CSIRMAZ LÁSZLÓ
KÖZÉP EURÓPAI EGYETEM
e-mail: csirmaz@renyi.hu

KATONA GYULA O. H.
RÉNYI INTÉZET
e-mail: ohkatona@renyi.hu

GEOMETRIC CODES

László Csirmaz and Gyula O. H. Katona

The last couple of years saw a huge increase in research of physical devices performing, or helping in, cryptographic operations. For example one-way functions can be defined and evaluated using special properties of physical objects. This paper considers the simplest possible scenario, namely, when a cryptographic key is to be extracted from a measurement. In this case the object to be measured behaves like a real key with the very useful extra property that it is physically impossible to make an exact copy of it. The key will, and can, exist in a single copy only. When performed repeatedly the measurements give different results. The extracted key, however, must remain the same. We define a mathematical model about how this goal can be achieved. We determine the important parameters of the model and establish a fundamental inequality between the parameters limiting the quality of the extracted cryptographic key. We also investigate how tight is our limit by giving several constructions. The paper concludes with an outline of other approaches, and a list of interesting problems left open.

Alkalmazott Matematikai Lapok **23** (2006), 363–373.

A NAGYMÉRTÉKBEN NEMLINEÁRIS FÜGGVÉNYEK SZÁMÁNAK ALSÓ KORLÁTJA

LICSKÓ ILDIKÓ

Budapest

A jelen dolgozatban megkíséreljük a páratlan dimenziójú, nagymértékben nemlineáris függvények és az eggyel magasabb dimenziójú maximálisan nemlineáris függvények számát alulról megbecsülni. A becslésre a nevezett függvénycsoportok közötti összefüggés ad lehetőséget.

1. Bevezetés

A kriptográfia különböző területein alkalmazzák a Boole-függvényeket. A visszacsatolt siftregiszterekkel működő folyamrejtjelező algoritmusok nemlineáris Boole-függvényeket alkalmaznak a kulcsfolyam és a rejtendő adatok egyesítésére. A blokkos típusú rejtjelező algoritmusok az S-dobozok belsejében valamint az egyes iterációs lépésekben a kulcs és a rejtendő szöveg egyesítésére alkalmaznak nemlineáris Boole-függvényeket. A modern rejtjelfejtési módszerekkel, a lineáris és differenciális analízissel szemben csak meghatározott tulajdonságú függvények képesek ellenállni, a függvényeknek ilyen megkívánt matematikai tulajdonságai a lehető legnagyobb mértékű nemlinearitás, a korrelációimmunitás, a kiegyensúlyozottság, stb.

2. Előzetes ismeretek

Az adatokat, információkat általában bináris formában tároljuk, ezért az adatkezeléssel illetve információkezeléssel kapcsolatos feladatokat általában $\{0,1\}^n$ halmazon valósítjuk meg. $\{0,1\}^n$ elemei vektorteret alkotnak, amelyet Boole-térnek nevezünk. $\{0,1\}^n$-beli vektorok komponensei $\{0,1\}$ értékűek, tehát egy-egy vektort tekinthetünk úgy is, mint egy egész szám bináris leírását. A Boole-tér i-edik

vektorának komponensei az i egész szám bináris reprezentációját adják. Hivatkozhatunk a vektorokra egy indexelt névvel is, ebben az esetben az indexben megjelenő szám bináris előállítását mutatják a vektor koordinátái. A Boole-tér elemei között a szokásos műveleteket értelmezhetjük: az $\mathbf{a}, \mathbf{b} \in \{0,1\}^n$ elemek koordinátákkal felírt alakja $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ és $\mathbf{b} = (b_0, b_1, \ldots b_{n-1})$, ekkor az összeadást komponensenként elvégezve az

$$\mathbf{a} + \mathbf{b} = (a_0 + b_0, a_1 + b_1, \ldots, a_{n-1} + b_{n-1})$$

eredményt kapjuk, és a skalárszorzatot az

$$\mathbf{ab} = \bigoplus_{i=0}^{n-1} a_i b_i$$

kifejezés definiálja.

Az $f : \{0,1\}^n \to \{0,1\}$ leképezést Boole-függvénynek nevezzük. A Boole-függvényeket megadhatjuk formulákkal vagy igazságtáblájukkal. Az n-dimenziós (n változós) függvény igazságtáblája 2^n sorból áll és $n+1$ oszlopot tartalmaz, ahol az első n oszlop a független változók értékét, az $n+1$-edik oszlop a hozzájuk tartozó függvényértéket mutatja. Az igazságtábla 2^n sora az adott Boole-tér valamennyi pontját és a hozzájuk tartozó függvényértéket megadja. Az igazságtábla megnevezést a táblázat $n+1$-edik oszlopára is használják. Gyakran alkalmazzák a Boole-függvények helyett az általuk generált $\{-1,1\}$ sorozatot, amelynek jelölése $\overline{f} : \{0,1\}^n \to \{-1,1\}$. A két leképezés kapcsolata:

$$\overline{f}(\mathbf{x}) = (-1)^{f(\mathbf{x})}$$

vagy

$$\overline{f}(\mathbf{x}) = 1 - 2f(\mathbf{x}).$$

A továbbiakban az $\overline{f}(x)$ jelölés mindig az f által generált 2^n elemű $\{-1,1\}$ sorozatot jelenti.

Az $f(x)$ függvény súlya az igazságtáblájában lévő 1-ek számát jelenti:

$$w(f) = \sum_{x \in \{0,1\}^n} f(x).$$

Az $f(x)$ függvényről azt mondjuk, hogy kiegyensúlyozott, ha igazságtáblájában az 1-ek és a 0-ák száma megegyezik, vagyis $f(x) = 1$ pontosan ugyanannyi helyen, mint $f(\mathbf{x}) = 0$. Ekkor

$$\sum_{x \in \{0,1\}^n} \overline{f}(\mathbf{x}) = 0,$$

és az $f(\mathbf{x})$ súlya: $w(f) = 2^{n-1}$.

Egy f függvényt affin függvénynek neveznek, ha Zsegalkin-polinomja minden $a_i \in \{0,1\}$ $0 \leq i \leq n-1$ esetén

$$f(x_0, x_1, \ldots, x_{n-1}) = a_0 x_0 \oplus a_1 x_1 \oplus \ldots \oplus a_{n-1} x_{n-1} \oplus c$$

alakú. Ha $c = 0$ akkor lineáris függvényről beszélünk, és ekkor az $\mathbf{ax}$ lineáris függvényt tekinthetjük az $\mathbf{a}$ konstans vektor és az $\mathbf{x}$ változó skalárszorzataként.

Az f és g függvények távolságát a

$$d(f,g) = w(f \oplus g) = \sum_{\mathbf{x} \in \{0,1\}^n} \big(f(\mathbf{x}) \oplus g(\mathbf{x})\big),$$

míg kettőjük korrelációját a

$$c(f,g) = \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{f(\mathbf{x}) \oplus g(\mathbf{x})} = \overline{f}(\mathbf{x})\overline{g}(\mathbf{x})$$

kifejezés definiálja. A két függvény közötti távolság és a két függvény közötti korreláció kapcsolata:

$$c(f,g) = \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{f(\mathbf{x}) \oplus g(\mathbf{x})} = 2^n - 2d(f,g).$$

Legyen H egy $m \times m$ méretű mátrix, amelynek elemeire igaz, hogy $h_{ij} \in \{-1,1\}$ minden $i,j = 0,1,\ldots,m-1$ esetén. H-t Hadamard-mátrixnak nevezzük, ha teljesül, hogy $HH^T = mI$, ahol H^T a H mátrix transzponáltja, és I az m-edrendű egységmátrix. A 2^n-edrendű Hadamard-mátrixot H_n-nel jelöljük, és ezt a mátrixot egyebek között a következő rekurzív eljárás segítségével is előállíthatjuk:

$$H_0 = 1, H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}, n = 1,2,\ldots.$$

A H_n mátrix sorait l_i-vel jelöljük $i = 0,1,\ldots,2^n-1$. Az i-edik sor l_i, az $\mathbf{ix}$ lineáris függvény által generált $\{-1,1\}$ sorozat, és ílymódon a H_n mátrix sorai az n-dimenziós tér minden lineáris függvénye által generált $\{-1,1\}$ sorozatot megjelenítenek.

Az f függvény Walsh-transzformáltja az $\mathbf{a} \in \{0,1\}^n$, $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ pontban:

$$F(\mathbf{a}) = \sum_{\mathbf{x} \in \{0,1\}^n} f(\mathbf{x})(-1)^{\mathbf{ax}},$$

míg az $\overline{f}$ Walsh-transzformáltja

$$\overline{F}(\mathbf{a}) = \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{f(\mathbf{x}) \oplus \mathbf{ax}}.$$

$F(\mathbf{a})$ és $\overline{F}(\mathbf{a})$ definiciójában a szummázást a szokásos módon értelmezzük, tehát nem *modulo* 2 összeadásként. Ennek következményeként $F(\mathbf{a})$ és $\overline{F}(\mathbf{a})$ értéke tetszőleges -2^n és 2^n közötti egész értéket vehet fel. Az $\{\overline{F}(\mathbf{a})\}$ egész értékekből álló 2^n elemű halmazt szokás a függvény Walsh-spektrumának nevezni. Az $\overline{F}(\mathbf{a})$ érték tulajdonképpen nem más, mint az f függvénynek az $\mathbf{ax}$ lineáris függvénnyel mutatott korrelációja, amelyet szintén írhatunk egy skalárszorzat alakjába:

$$\left(\overline{f}(\mathbf{x})l_a\right).$$

Az f függvényt maximálisan nemlineárisnak nevezzük*, ha minden $\mathbf{a} \in \{0,1\}^n$-re

$$\overline{F}(\mathbf{a}) = \sum_{x \in \{0,1\}^n} (-1)^{f(\mathbf{x}) \oplus \mathbf{ax}} = \pm 2^{\frac{n}{2}},$$

vagyis a függvény Walsh-spektruma azonos abszolút értékű elemekből áll.

Az f függvény nemlinearitásán az f függvénynek az affin függvények halmazától mért távolságát értjük, és N_f-el jelöljük:

$$N_f = 2^{n-1} - \frac{1}{2} \max_{i=1,2,\ldots,2^n} \left(\overline{f}(\mathbf{x})l_i\right).$$

A nemlinearitás legkisebb értéke 0, ez az affin függvények nemlinearitása. Bár a maximálisan nemlineáris függvények a nemlinearitást tekintve igen kedvező tulajdonságúak, mégsem használják ezeket kriptográfiai célra, mert egyéb tulajdonságaik nem megfelelőek. Ezek a függvények például nem kiegyensúlyozottak, és nem is tehetők azzá, nem korrelációimmunisak, csak páros dimenziójú térben léteznek stb.

Az $f : \{0,1\}^n \to \{0,1\}$ függvényt nagymértékben nemlineárisnak nevezzük, ha n páratlan, és az f függvény Walsh-spektrumában levő elemek csak 0 és $\pm C$ értékűek és az elemeknek pontosan a fele nem zérus. A kriptográfiai alkalmazásokban a nagymértékben nemlineáris függvényeket előnyben részesítik a maximálisan nemlineáris függvényekkel szemben, tekintettel arra, hogy egyéb tulajdonságaik is megfelelőek. Például kiegyensúlyozottak, vagy könnyen azzá tehetők, korreláció immunitásuk is jó stb. [3] alapján a nagymértékben nemlineáris függvényekből diszjunkt párokat képezhetünk, amely párok jellemző tulajdonsága, hogy a pár elemei egy adott lineáris függvénnyel sohasem mutatnak azonos abszolút értékű korrelációt. Pontosabban fogalmazva: g_1 és $g_2 : \{0,1\}^n \to \{0,1\}$ nagymértékben nemlineári függvények egy diszjunkt párt alkotnak, ha $\left(\overline{g}_1(\mathbf{x})l_i\right) = 0$-ból következik $\left(\overline{g}_2(\mathbf{x})l_i\right) \neq 0$ illetve fordítva. Ilyen párok létezésére szükséges és elégséges feltétel, hogy

$$\left(\overline{g}_1(\mathbf{x})l_i\right)^2 + \left(\overline{g}_2(\mathbf{x})l_i\right)^2 = 2^{n+1}$$

*Az angol nyelvű irodalomban szokásos megnevezésük: Bent functions.

teljesül minden $i = 0, 1, \ldots, 2^n - 1$-re. Diszjunkt függvénypárok alapján további hasonló tulajdonságú függvénypárok szerkeszthetők az n-dimenziós lineáris függvények segítségével az alábbi módon:

$$\overline{h}_1(i) = \frac{(\overline{g}_1(\mathbf{x})l_i) + (\overline{g}_2(\mathbf{x})l_i)}{2^{n-m}}$$

és

$$\overline{h}_2(i) = \frac{(\overline{g}_1(\mathbf{x})l_i) - (\overline{g}_2(\mathbf{x})l_i)}{2^{n-m}}.$$

Az eredményként előálló $\overline{h}_1$ és $\overline{h}_2$ sorozat két függvény által generált $\{-1, 1\}$ sorozatnak tekinthető, és ez a két függvény kielégíti a

$$(\overline{h}_1(\mathbf{x})l_k)^2 + (\overline{h}_2(\mathbf{x})l_k)^2 = 2^{n+1}$$

feltételt minden $k = 0, 1, \ldots, 2^n - 1$-re.

Az előzőekben említett, nagymértékben nemlineáris függvénypárok mutatják meg a maximálisan nemlineáris és a nagymértékben nemlineáris függvények közötti szoros kapcsolatot. Páros n esetén az $f : \{0,1\}^n \to \{0,1\}$ függvény akkor és csak akkor maximálisan nemlineáris függvény, ha igazságtáblája előállítható két 1-gyel alacsonyabb változós diszjunkt nagymértékben nemlineáris függvénypár igazságtáblájának összefűzésével.

3. Nagymértékben nemlineáris függvények száma

A függvények számának megbecsüléséhez vizsgáljuk először a 3-dimenziós teret. Ha egy függvénynek egy lineáris függvénnyel vett korrelációja zérus, akkor ez azt jelenti, hogy a függvény igazságtáblájában a függvényértékek fele megegyezik az adott lineáris függvény függvényértékeivel, a pontok másik fele viszont különbözik. Ha a függvény korrelációja egy lineáris függvénnyel 4 vagy -4 értékű, akkor ez azt jelenti, hogy a függvény igazságtáblájában 2 pontban a függvényérték megegyezik a lineáris függvény értékével, 2 pontban különbözik attól, és a maradék 4 pontban vagy megegyezik a lineáris függvénnyel, vagy különbözik attól. Tekintsük a 3-dimenziós tér lineáris függvényeit, és változtassuk meg egy lineáris függvény igazságtáblájában a megfelelő számú függvényértéket.

A háromdimenziós lineáris függvények igazságtáblái:

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0x	0	0	0	0	0	0	0	0
1x	0	1	0	1	0	1	0	1
2x	0	0	1	1	0	0	1	1
3x	0	1	1	0	0	1	1	0
4x	0	0	0	0	1	1	1	1
5x	0	1	0	1	1	0	1	0
6x	0	0	1	1	1	1	0	0
7x	0	1	1	0	1	0	0	1

Induljunk ki a legegyszerűbb igazságtáblából, amely a $0x$ lineáris függvényhez tartozik. A lehetséges módosítások:

1. két pont értékét megváltoztatjuk és hat pont változatlan,
2. hat pont értékét megváltoztatjuk és két pont változatlan,
3. négy pont értékét megváltoztatjuk, négy pont változatlan.

Az első két esetben a módosított függvények mindegyike nagymértékben nemlineáris függvény, a harmadik módszerrel lineáris függvényeket is előállítunk.

Eredeti lin.fgv.	Módosított függvény								Korreláció a lin.függv-ekkel							
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7
0x	1	1	0	0	0	0	0	0	4	0	−4	0	−4	0	−4	0
0x	1	0	1	0	0	0	0	0	4	−4	0	0	−4	−4	0	0
0x	1	0	0	1	0	0	0	0	4	0	0	−4	−4	0	0	−4
0x	1	0	0	0	1	0	0	0	4	−4	−4	−4	0	0	0	0
0x	1	0	0	0	0	1	0	0	4	0	−4	0	0	−4	0	−4
0x	1	0	0	0	0	0	1	0	4	−4	0	0	0	0	−4	−4
0x	1	0	0	0	0	0	0	1	4	0	0	−4	0	−4	−4	0
0x	0	1	1	0	0	0	0	0	4	0	0	4	−4	0	0	4
0x	0	0	1	1	1	1	1	1	−4	0	4	0	4	0	4	0
0x	0	1	0	1	1	1	1	1	−4	4	0	0	4	4	0	0
0x	0	1	1	0	1	1	1	1	−4	0	0	4	4	0	0	4
0x	0	1	1	1	0	1	1	1	−4	4	4	4	0	0	0	0
0x	0	1	1	1	1	0	1	1	−4	0	4	0	0	4	0	4
0x	0	1	1	1	1	1	0	1	−4	4	0	0	0	0	4	4
0x	0	1	1	1	1	1	1	0	−4	0	0	4	0	4	4	0
0x	1	0	0	1	1	1	1	1	−4	0	0	−4	4	0	0	−4
0x	0	0	0	0	1	1	1	1	0	0	0	0	8	0	0	0
0x	0	0	0	1	0	1	1	1	0	4	4	0	4	0	0	−4
0x	0	0	0	1	1	0	1	1	0	0	4	−4	4	4	0	0
0x	0	0	0	1	1	1	0	1	0	4	0	−4	4	0	4	0
0x	0	0	0	1	1	1	1	0	0	0	0	0	4	4	4	−4
0x	0	0	1	0	0	1	1	1	0	0	4	4	4	−4	0	0

Ha minden lineáris függvényre az összes lehetséges módosításokat elvégezzük, akkor minden lehetséges nagymértékben nemlineáris függvényt előállítunk, de egy-egy nagymértékben nemlineáris és lineáris függvényt többször is generálunk. Kiválogatva a különböző függvényeket, arra az eredményre jutunk hogy a 3-dimenziós térben a nagymértékben nemlineáris függvények száma 112. Tovább vizsgálódva azt látjuk, hogy ez a 112 függvény 14 függvénycsoportba sorolható úgy, hogy mindegyik csoportban a tagok azonos korrelációs tulajdonságokat mutatnak a lineáris függvényekkel, vagyis egy-egy csoport elemei ugyanazokkal a lineáris függvényekkel adják a nem zérus korrelációt. A 14 függvényhalmazból 7 párt képezhetünk. A függvényhalmaz párokra jellemző, hogy e halmazpár egyik eleméből vett függvény és a halmazpár másik eleméből vett függvény, nevezzük ezeket g_1 és g_2-nek, kielégítik a

$$\left(\overline{g}_1(\mathbf{x})l_i\right)^2 + \left(\overline{g}_2(\mathbf{x})l_i\right)^2 = 2^{n+1}$$

feltételt. Vagyis nem találunk olyan nagymértékben nemlineáris függvényt, amelynek ne lenne párja a fenti értelemben.

Bemutatjuk az egyik függvényhalmaz-párt.

A függvényhalmaz-pár első tagja:

Eredeti lin.fgv.	Módosított függvény								Korreláció a lin.függv-ekkel							
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7
0x	0	0	1	1	1	1	1	1	−4	0	4	0	4	0	4	0
0x	1	1	0	0	1	1	1	1	−4	0	−4	0	4	0	−4	0
0x	1	1	1	1	0	0	1	1	−4	0	4	0	−4	0	−4	0
0x	1	1	1	1	1	1	0	0	−4	0	−4	0	−4	0	4	0
0x	1	1	0	0	0	0	0	0	4	0	−4	0	−4	0	−4	0
0x	0	0	1	1	0	0	0	0	4	0	4	0	−4	0	4	0
0x	0	0	0	0	1	1	0	0	4	0	−4	0	4	0	4	0
0x	0	0	0	0	0	0	1	1	4	0	4	0	4	0	−4	0

A függvényhalmaz-pár második tagja:

Eredeti lin.fgv.	Módosított függvény								Korreláció a lin.függv-ekkel							
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7
1x	0	1	1	0	1	0	1	0	0	−4	0	4	0	4	0	4
1x	1	0	0	1	1	0	1	0	0	−4	0	−4	0	4	0	−4
1x	1	0	1	0	0	1	1	0	0	−4	0	4	0	−4	0	−4
1x	1	0	1	0	1	0	0	1	0	−4	0	−4	0	−4	0	4
1x	1	0	0	1	0	1	0	1	0	4	0	−4	0	−4	0	−4
1x	0	1	1	0	0	1	0	1	0	4	0	4	0	−4	0	4
1x	0	1	0	1	1	0	0	1	0	4	0	−4	0	4	0	4
1x	0	1	0	1	0	1	1	0	0	4	0	4	0	4	0	−4

Mivel ezek a párok egyértelmű kapcsolatban vannak a maximálisan nemlineáris függvényekkel, megbecsülhetjük azok számát is. A 112 függvényből 448 függvénypárt képezhettünk, mindegyik függvénypárból 2 darab 1-el magasabb dimenziójú

maximálisan nemlineáris függvény szerkeszthető, így adódik, hogy a 4-dimenziós, maximálisan nemlineáris függvények száma 896.

Sajnos ezt a konstrukciót magasabb dimenzióban nem alkalmazhatjuk, mivel a lineáris függvények igazságtáblájában végrehajtott módosítások nem csak lineáris vagy nagymértékben nemlineáris függvényeket eredményeznek, hanem egyéb függvényeket is. A szabályosság, ami végül is elvezetne a nagymértékben nemlineáris függvényekhez nem látszik. A további becslési lehetőség érdekében tekintsünk egy példát [7]-ből. Legyen l_i a H_k Hadamard-mátrix i-edik sora. Az $l_0, l_1, \ldots, l_{2^k-1}$ $\{-1,1\}$ sorozatok az összes $\{0,1\}^k$-beli, 2^k darab lineáris függvény által generált $\{-1,1\}$ sorozatok, amelyek mindegyike 2^k elemből áll. Belátható, hogy bármely 2^{k-1} darab, 2^k hosszúságú különböző lineáris függvény által generált $\{-1,1\}$ sorozat összefűzése (konkatenálása) egy olyan $\{-1,1\}$ sorozatot eredményez, amelynek generáló függvénye f a $\{0,1\}^{2k-1}$ téren értelmezett. Az így létrejött $\{-1,1\}$ sorozathoz tartozó függvényre teljesül, hogy Walsh-transzformáltja a 0 vagy $\pm 2^k$ értéket veszi fel, és ezért nemlinearitása $N_f = 2^{2k-1-1} - 2^{\frac{1}{2}(2k-1-1)}$.

3.1. LEMMA. *Jelölje l_i a 2^n-edrendű H_n, és L_j a 2^{2n-1}-edrendű H_{2n-1} Hadamard-mátrix sorait, ahol $i = 0,1,2,\ldots,2^n-1$ és $j = 0,1,2,\ldots,2^{2n-1}-1$. A $g_i : \{0,1\}^n \to \{0,1\}$ és a $G_j : \{0,1\}^{2n-1} \to \{0,1\}$ lineáris függvények $\{-1,1\}$ sorozatai rendre megfelelnek az l_i és L_j soroknak, azaz $\overline{g}_i = l_i$ és $\overline{G}_j = L_j$ minden i-re és j-re. Az $f : \{0,1\}^{2n-1} \to \{0,1\}$ függvényhez tartozó $\overline{f}$ legyen olyan, amelyet 2^{n-1} darab különböző l_i összefűzésével állítottunk elő. Ekkor az f nagymértékben nemlineáris függvény azzal a G_j lineáris függvénnyel mutat nem zérus korrelációt, amelynek L_j sorát olyan l_i sor generálja, amely előfordul $\overline{f}$-ban.*

Bizonyítás. L_j a j-edik sor H_{2n-1}-ben.Vizsgáljuk meg, milyen komponensek alkotják. Induljunk ki H_n-ből, és tekintsük a rekurzív eljárást:

$$H_n \to H_{n+1} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \to \cdots H_{2n-1} = \begin{bmatrix} H_{2n-2} & H_{2n-2} \\ H_{2n-2} & -H_{2n-2} \end{bmatrix}$$

$$H_n = \begin{bmatrix} l_0 \\ \vdots \\ l_{2^n-1} \end{bmatrix}, H_{2n-1} = \begin{bmatrix} L_0 \\ \vdots \\ L_{2^{2n-1}-1} \end{bmatrix}$$

Az első lépésben a H_{n+1} mátrixot generáljuk. Az eljárásban észrevehetjük, hogy a kiindulási mátrix sorai nem keverednek. Ez azt jelenti, hogy ha tekintjük a H_n mátrix egy l_i sorát, akkor az eljárásban l_i mellé csak l_i vagy $-l_i$ sor kerül. Folytatva az eljárást, H_{2n-1}-ről elmondhatjuk, hogy az L_j sor csak az l_p vagy $-l_p$ $p = j$ *modulo* 2^n sort tartalmazza H_n-ből. Ez azt jelenti, hogy az L_j sorozat felírható mint l_p vagy $-l_p$ valamilyen sorrendbeli konkatenáltja azaz $L_j = (m_1 l_p, m_2 l_p, \ldots m_{2^{n-1}} l_p,)$ és $m_r \in \{-1,1\}$ minden $r = 1,2,\ldots,2^{n-1}$-re. Az f függvényről pedig azt tudjuk, hogy $\overline{f}(\mathbf{x}) = (l_{i_1}, l_{i_2}, \ldots, l_{i_{2^{n-1}}})$ $\{i_1, i_2, \ldots, i_{2^{n-1}}\} \subset$

$\{0,1,\ldots,2^n-1\}$ teljesül. A G_j lineáris függvény és az f korrelációja a két általuk generált $\{-1,1\}$ sorozat skalárszorzatával számítható:

$$\overline{f}(\mathbf{x})L_j = \sum_{r=1}^{2^{n-1}} \left(l_{i_r}(l_p m_r)\right) = \sum_{r=1}^{2^{n-1}} \left(l_{i_r} l_p\right) m_r.$$

Az Hadamard-mátrixok tulajdonsága, hogy két sor skalárszorzata zérus, ha a két sor különböző, és 2^n, ha a két sor azonos. Ebből adódik, hogy $\overline{f}(\mathbf{x})L_j = 0$, ha $\overline{f}(\mathbf{x})$ nem tartalmazza azt az l_p sort, amelyik L_j-t generálja, és $\overline{f}(\mathbf{x})L_j = \pm 2^n$, ha l_p előfordul $\overline{f}(\mathbf{x})$-ben. A H_{2n-1}-et generáló rekurzív eljárás alapján állíthatjuk, hogy egy adott l_p sor (2^{n-1})-szer szerepel a mátrix sorai között, $\overline{f}(\mathbf{x})$-et 2^{n-1} különböző sorból állítottuk elő, tehát pontosan $2^{n-1}2^{n-1} = 2^{2n-2} = 2^{(2n-1)-1}$ azon G_j lineáris függvények száma, amelyekkel az f korrelációja nem zérus. □

3.2. TÉTEL. *Az $f : \{0,1\}^{2n-1} \to \{0,1\}$ nagymértékben nemlineáris függvények száma legalább*

$$2^{2^{n-1}} \frac{(2^n)!}{(2^{n-1})!}.$$

Bizonyítás. A H_n mátrix 2^n sorából kell 2^{n-1} különbözőt kiválasztani. A lehetséges választások száma $\binom{2^n}{2^{n-1}}$. Mivel a negálás nem változtatja a korreláció abszolút értékét, ezért nem csak a kiválasztott sorokat, hanem azok negáltjait is fel lehet használni. Vigyázni kell azonban arra, hogy egy sorozatban nem szerepelhet együtt valamely sor a negáltjával. Ha így tekintjük a sorokat és azok negáltjait, akkor az elkészíthető sorozatok száma egy választásból a negáltakat is figyelembe véve:

$$\binom{2^{n-1}}{0}(2^{n-1})! + \binom{2^{n-1}}{1}(2^{n-1})! + \ldots + \binom{2^{n-1}}{2^{n-1}}(2^{n-1})! = (2^{n-1})!2^{2^{n-1}}.$$

Szorozzuk meg az egy választásból előállítható sorozatok számát a lehetséges választások számával:

$$\binom{2^n}{2^{n-1}}(2^{n-1})!2^{2^{n-1}} = \frac{(2^n)!(2^{n-1})!2^{2^{n-1}}}{\left((2^{n-1})!\right)^2} = 2^{2^{n-1}} \frac{(2^n)!}{(2^{n-1})!}.$$ □

3.3. TÉTEL. *Az $f : \{0,1\}^{2n} \to \{0,1\}$ maximálisan nemlineáris függvények száma legalább:*

$$\frac{(2^n)!2^{2^n}}{(2^{n-1})!}.$$

Bizonyítás. Minden esetben, amikor kiválasztunk 2^{n-1} sort a H_n mátrixból, tulajdonképpen két halmazt képezünk a sorokból, mégpedig két diszjunkt halmazt, amelyeknek nincs közös eleme. A kiválasztott sorokhoz tartozó nagymértékben nemlineáris függvények korrelációja egy lineáris függvénnyel sohasem veszi

fel ugyanazt az abszolút értéket, mint a megmaradó sorokból képzett függvények korrelációja. Tekintsünk egy függvényt a kiválasztott sorokhoz tartozó függvényhalmazból, legyen ez $g_1 : \{0,1\}^n \to \{0,1\}$, és egy függvényt a meghagyott sorokhoz tartozó függvényekből, legyen ez $g_2 : \{0,1\}^n \to \{0,1\}$. Ekkor a két függvény kielégíti a

$$\left(\overline{g}_1(\mathbf{x})L_j\right)^2 + \left(\overline{g}_2(\mathbf{x})L_j\right)^2 = 2^{2n}$$

feltételt, vagyis alkalmasak 1-gyel magasabb dimenziós maximálisan nemlineáris függvények szerkesztésére. Mindkét függvényhalmaz elemeinek száma $(2^{n-1})!2^{2^{n-1}}$, és az ezekből készíthető függvénypárok száma $\left((2^{n-1})!2^{2^{n-1}}\right)^2$. A

$$\overline{h}_1(i) = \frac{\left(\overline{g}_1(\mathbf{x})L_i\right) + \left(\overline{g}_2(\mathbf{x})L_i\right)}{2^n}$$

és

$$\overline{h}_2(i) = \frac{\left(\overline{g}_1(\mathbf{x})L_i\right) - \left(\overline{g}_2(\mathbf{x})L_i\right)}{2^n}$$

eljárással minden függvénypárhoz egy további párt készíthetünk, és ezzel az egy függvényhalmaz-párból készíthető függvénypárok száma: $2\left((2^{n-1})!2^{2^{n-1}}\right)^2$. Egy függvénypárból 1 maximálisan nemlineáris függvény készíthető a $2n$-dimenziós térben. A negált verziókkal külön nem kell törődni, hiszen a generált nagymértékben nemlineáris függvények negált alakját külön függvényként állítottuk elő. A H_n sorait $\binom{2^n}{2^{n-1}}$ féleképpen választhatjuk ki, tehát

$$\frac{\binom{2^n}{2^{n-1}}}{2}$$

darab függvényhalmaz-párt képezhetünk. Ezzel a szerkeszthető maximálisan nemlineáris függvények száma legalább:

$$\frac{(2^n!)2^{2^n}}{(2^{n-1})!}.$$

□

Összefoglalás

A dolgozatban bemutatott eredmény a páratlan dimenziós nagymértékben nemlineáris függvények és az 1-gyel magasabb dimenziós maximálisan nemlineáris függvények minimális számára ad alsó becslést.

Köszönetnyilvánítás. Köszönetemet szeretném kifejezni Dr. Nemetz Tibor Professzor úrnak, és Pásztorné Varga Katalin egyetemi docens asszonynak, akik értékes szakmai tanácsaikkal segítették munkámat.

Hivatkozások

[1] Licskó, I., On Highly Nonlinear Functions, *Annales Univ. Sci. Budapest, Sect. Comp.*, **21** (2002), 165–175.

[2] Licskó, I., Nagymértékben nemlineáris függvények vizsgálata, *Alkalmazott Matematikai Lapok*, **21** (2004), 3–22.

[3] Licskó, I., Characteristics of Highly Nonlinear Functions, *Periodica Matematica Hungarica*, Vol. **47 (1–2)** (2003), 135–149.

[4] Pásztor-Varga, K., Boole függvények Boole algebrájának strukturális tulajdonságait felhasználó Boole függvény optimizációs módszer, *Alkalmazott Matematikai Lapok*, **13** (1987–88), 69–76.

[5] Seberry, J., Zhang, X. M. and Zheng, Y., Nonlinearity and propagation characteristics of balanced Boolean functions, *Information and Computation*, **119(1)** (1995), 1–13.

[6] Vajda, I., Buttyán, L., Szekeres, B., *Az algoritmikus adatvédelem módszerei*, Technical University of Budapest, nem publikált kézirat.

[7] Zhang, X. M. and Zheng, Y., New lower bounds on Nonlinearity and class of highly nonlinear functions, in: *Information Security and Privacy*, Second Australian Conference Sidney, Australia, July 1997, Lecture Notes in Computer Science **1270**, 147–158, **1(1)**, 1–13, Springer-Verlag (Berlin, Heidelberg, New York, 1997).

LICSKÓ ILDIKÓ
BUDAPESTI GAZDASÁGI FŐISKOLA
KERESKEDELMI, VENDÉGLÁTÓIPARI ÉS IDEGENFORGALMI KAR
INFORMATIKAI INTÉZET
1054 BUDAPEST,
V. ALKOTMÁNY UTCA 9–11
e-mail: licsko_ildiko@t-online.hu

ESTIMATION FOR THE LOWER BOUND ON THE NUMBER OF HIGHLY NONLINEAR FUNCTIONS

ILDIKÓ LICSKÓ

This study provides an estimation for the lower bound of the number of highly nonlinear n-dimensional functions if n is odd. The basis of the estimation is the relationship between highly nonlinear odd n-dimensional functions and bent functions in the $(n+1)$-dimension.

Alkalmazott Matematikai Lapok **23** (2006), 375–388.

SZAKMAI SZERVEZETEN BELÜLI BIZTONSÁGOS SZAVAZÁSI RENDSZER EGY ALKALOMRA

LUKOVICS JÓZSEF

Budapest

Egy választás megszervezése egy szakmai szervezetben nem egy egyszerű feladat, mert a nagyobb szervezet már nem képes arra, hogy a szavazásra jogosult tagjait egy helyen és egy időben összehívja. Ehelyett általában levélben oldják meg a szavazást. Elvileg gondot fordítanak a szavazás titkosságára, a szavazók hitelesítésére és arra, hogy a legális szavazók csak egyszer szavazhassanak. Ebben a cikkben egy olyan protokollt ajánlunk, amely gyakorlatban is teljesíti a szavazásra vonatkozó előbbi feltételeket. A protokoll a postai levelezés helyett az internet e-mail szolgáltatását használja és lényegében a jelenleg is meglévő szavazási módot szimulálja. Ebben a sémában a szavazónak csak egy saját titkos jelszóra (és nem titkos kulcsra) van szüksége. A szavazók egy időben több kérdésről dönthetnek.

I. Bevezetés

A szakmai szervezeteknél általában még ma is kézi módszerrel történik a szavazás, ami többnyire a vezető tisztségviselők kiválasztására irányul. A dupla borítékos módszert használják, azaz a szavazó a szavazatát egy üres borítékba teszi be, amire nem ír semmit, és amit lezár. Ezt a borítékot rakja be egy felcímkézett borítékba, amelyre ráírja a saját adatait is. Ha eleget tesznek a titkosságra vonatkozó feltételnek, akkor a postai levelet felbontó részleg (authority) különbözik a belső zárt borítékot felnyító részlegtől, és csak azt ellenőrzi le, hogy a feladó személy valóban jogosult-e a szavazásra. Ezenkívül semmi más feladata nincs, minthogy a zárt borítékot átadja a szavazást értékelő részlegnek. Az értékelő részleg számolja ki a szavazás eredményét és közli a szavazás eredményét.

A szakmai szervezeteknél a jelenleg alkalmazott kézi módszerrel való szavazásnál a gyakorlatban három fő követelményt támasztanak a szavazással szemben:

- csak a jogosult személyek szavazhatnak,
- minden személy csak egyszer szavazhat (a duplázás lehetetlen),

- a szavazat legyen titkos (titkosság).

Az internet korszakában lehetségessé vált egy titkos szavazás megszervezése az e-mail szolgáltatáson keresztül. Az itt ajánlott séma teljesíti ezen követelményeket, egy nyilvános kulcsú rendszer (PKS = Public Key System), egy EF egyirányú függvény és egy valódi vagy pszeudovéletlen byte-sorozat generátor felhasználásával. A PKS rendszer két függvényből áll: egy $y = E(K, x)$ rejtjelző függvényből és egy $x = D(MK, y)$ megoldó függvényből. A nyilvános K rejtjelző kulcs és titkos MK megoldó kulcs különböző.

A szakirodalom már nem foglalkozik a gyakorlatban manapság használt szavazási sémákkal, mivel ezek elméleti szempontból már kidolgozott sémák. Helyette a szavazás újabb elméleti követelményeit vizsgálják, [1], [5]-ben a szerzők többnyire az alábbi egyéb elvárásokat próbálják megkövetelni egy választási sémától:

- általános ellenőrizhetőség: bárki ellenőrizheti, hogy a szavazat helyes;
- néhány megbízhatatlan hatóság, részleg esetén, titokmegosztási séma alkalmazásával a szavazás normális végrehajtása garantálható;
- a rendszer akkor is működik, ha néhány hatóság, részleg megtagadja a saját feladatának teljesítését;
- a szavazó el tudja adni a szavazatát úgy, hogy a vásárló nem tudja ellenőrizni, hogy mit tartalmaz a szavazat;
- a szavazó a szavazás után ne tudja bizonyítani, hogy mire szavazott.

A gyakorlatban jelenleg használt sémák nem foglalkoznak ezen problémákkal. Az általános ellenőrizhetőség helyett néhány hivatalos döntőbíró van, akik ellenőrizhetik, hogy a szavazás rendben zajlott. A hatóságokról feltételezzük, hogy megbízhatóak, így feladataikat pontosan elvégzik.

A szavazat eladás egy különös probléma. A mostani rendszerben, ahol a szavazó egyedül tartózkodik a szavazóhelységben, senki sem tudja ellenőrizni, hogy mire szavaz. Sokkal fontosabb, hogy senki se fogadjon el pénzt a szavazatáért, mint az, hogy valójában úgy szavaz-e, ahogy azt megígérte, és hogy az igéretének teljesítését hogyan lehet ellenőrizni. A szavazás utáni bizonyítás szintén egy erkölcsi probléma, a jelenlegi gyakorlati rendszerekben bárki tehet egy sajátos ismertetőjelet a szavazócédula sarkára, a szavazat attól még érvényes marad.

Az általunk javasolt séma csak a jelenleg használt séma elektronikus adaptációja kíván lenni.

II. A javasolt szavazási séma alkotó elemei és leírása

A sémában résztvevő szereplők a következők:

- Az A megbízható részleg, amelyik felelős a tagok regisztrálásáért, ellenőrzi a tagdíj fizetési eljárásokat és tárolja a tagok adatait, beleértve az e-mail címüket és a jelszó ellenőrző kódot. Az A hatóság gyűjti össze az e-mail üzeneteket a hozzájuk csatolt rejtjelzett állománnyal együtt és csak a hozzácsatolt megoldott állományt (attached file) adja át a B hatóságnak.
- A B megbízható részleg, amelyik felelős a szavazatok összegyűjtéséért, tárolja a regisztrált szavazók lexikografikusan sorba rendezett ellenőrző kódjait, és összegyűjti a megoldott hozzácsatolt állományokat és a belőlük nyert rejtjelzett szavazatot. A B részleg csak az érvényes szavazatokat adja át a C részlegnek.
- A megbízható C részleg, amelyik felelős a szavazás értékeléséért, tárolja a rejtjeles szavazatokat és kiszámítja a szavazás eredményét.
- A döntőbíró, aki a szavazás után megerősíti, hogy
  - az A és B részleg egyetért a regisztrált szavazók listájában,
  - az A és B részleg egyetért a hozzácsatolt állományok listájában,
  - a B és C részleg egyetért a rejtjeles szavazatok listájában,
  - a C hatóság helyesen számolta ki az eredményt.
- A szervezet tagja, aki vagy
  - regisztrált tag, aki részt vesz a szavazásban, vagy
  - regisztrált tag, aki nem vesz részt a szavazásban, vagy
  - olyan tag, aki nem regisztráltatta magát ennél a szavazásnál.

A sémában használt rejtjelzési algoritmusok:

- nyilvános kulcsú rendszer PKS három különböző kulccsal (K_1, K_2, K_3 nyilvános, MK_1, MK_2 és MK_3 titkos megoldó kulcsok), ahol az A hatóság csak az MK_3, a B hatóság csak az MK_2 és a C hatóság csak az MK_1 titkos kulcsot ismeri,
- egy egyirányú függvény (EF) a jelszó ellenőrző kód (JEK) generálásához,
- egy véletlen vagy pszeudovéletlen byte-sorozat generátor.

A sémában használt adatbázisok:

- tagok adatbázisa (A részleg), ahol egy rekord többek között tartalmazza:
  - a tag azonosító számát és személyi adatait;
  - e-mail címét;
  - a tagdíjfizetés módját;
  - a tagsági viszony aktuális érvényességének jelzését;
  - a tag jelszavának ellenőrző kódját, $JEK = EF(J)$, ahol J a tag titkos jelszava;

- egy lexikografikusan rendezett regisztrált jelszó ellenőrző kód adatbázis (ezt a B részleg tárolja), ahol egy rekord az alábbiakat tartalmazza:
  - a jelszó ellenőrzési kódot, $JEK = EF(J)$;
  - a kódhoz tartozó J jelszót;
- a rejtjeles szavazatok rendezett adatbázisa (a C részleghez tartozik), ahol egy rekord csak egy rejtjeles szavazatot tartalmaz.

A sémában használt programok:

- egy rejtjelzett csomagot készítő program, ami $E(K_3,.)$ segítségével a tagok adatbázisának egy rekordját titkosítja;
- az előző csomagot $D(MK_3,.)$ segítségével megoldó program, ami a tagok adatbázisában kitölti a megfelelő adatokat a csomag tartalma alapján;
- a tagok adatbázisában az éppen érvényes tagsággal rendelkezők nem üres jelszó ellenőrző kódjait összegyűjtő és kimásoló program;
- az összegyűjtött ellenőrző kódokat lexikografikusan sorba rendező és a rendezett halmazt egy új adatbázisba szervező program;
- a csatolt állományt elkészítő szavazóprogram, amelyik $E(K_1,.)$ felhasználásával lerejtjelzi a szavazatot, hozzáfűzi a jelszót, az egészet lerejtjelzi $E(K_2,.)$-vel, majd a kapott blokkot lerejtjelzi $E(K_3,.)$ segítségével;
- a csatolt állományt megoldó és leválasztó program, ami $D(MK_3,.)$ révén megoldott állományt elküldi érvényességi vizsgálatra a B hatóságnak;
- a szavazat érvényességét ellenőrző program, ami $D(MK_2,.)$ felhasználásával megoldja a kapott állományt, és leellenőrzi, hogy a megadott jelszó helyes-e vagy sem, azaz van-e olyan $JEK = EF$(jelszó) érték a rendezett jelszó ellenőrző kód adatbázisban, amelynél a jelszó rovat még nincs kitöltve. Ha van ilyen ,akkor kitölti a jelszó rovatot és a rejtjeles szavazatot elküldi a C hatóságnak;
- a szavazat összesítő program, ami $D(MK_1,.)$ segítségével megoldja a szavazatokat és kiszámolja a szavazás eredményét.

A szavazás megszervezéséhez az A hatóságnak el kell készítenie a lehetséges szavazók adatbázisát. A szavazónak a jelszó ellenőrző kód megadásával regisztráltatnia kell magát.

A regisztrációs határidő lezárása után az A hatóság másolatot készít az ellenőrző kódokról, és lexikografikusan rendezve átküldi azokat a B hatóságnak. Emellett az A hatóság összegyűjti a különböző posztokra a lehetséges jelölteket, és elkészíti a szavazóprogramot. A szavazó az internetről letölti a szavazóprogramot, lefuttatja, és a létrehozott állományt egy e-mail üzenethez hozzácsatolva elküldi az A hatóságnak. A B hatóság ellenőrzi a szavazat érvényességét és gyűjti az érvényes szavazatokat. A szavazás határidejének lejárta után a C hatóság számolja ki a szavazás végeredményét.

A továbbiakban bemutatjuk a szavazás főbb eljárásait. A III. fejezetben részletezzük a főbb lépéseket. Az A, B, C és D melléklet segít a részletek megértésében.

A IV. fejezetben megvizsgáljuk, hogy milyen problémák léphetnek fel a szavazásnál. Az V. fejezetben néhány olyan algoritmust javasolunk, amit ehhez a sémához fel lehetne használni.

Az e-mail révén megvalósított titkos szavazás folyamata az alábbi lépésekből áll:

a. az A hatóság regisztrálja a lehetséges szavazókat és elkészíti a regisztrációs programot;

b. a regisztrációs folyamat lezárása után az A hatóság előállítja a regisztrált jelszó ellenőrzési kódok rendezett adatbázisát, amit később átad a B hatóságnak;

c. a különböző posztokra a jelöltek összegyűjtése (jelölő eljárás);

d. a posztok és jelöltek alapján a szavazóprogram elkészítése;

e. a szavazóprogram letöltése az interneten keresztül, vagy elküldése egy e-mail üzenethez hozzácsatolva az összes regisztrált tag e-mail címére;

f. a szavazóprogram által létrehozott állomány elküldése az A hatóságnak egy e-mail üzenethez hozzácsatolva;

g. az A hatóság megoldja a hozzácsatolt állományt, és az eredményt félreteszi a B hatóság számára;

h. a határidő lejárta után a B hatóság kicsomagolja a szavazatot, ellenőrzi annak érvényességét, és az érvényes szavazatokat elküldi a C hatóságnak;

i. a C hatóság kiértékeli a szavazatokat;

j. a szavazás helyességét igazoló eljárás.

III. A fő lépések részletes leírása

1. A szavazók A hatóság általi regisztrációja

- Létezik egy aktuális tagsági lista. Kezdetben a tagok adatbázisában az összes jelszó ellenőrzési kód mezője üres.
- A regisztrációhoz a szavazónak választania kell egy titkos jelszót. (Ez néhány megszorítás mellett lehet egy 16 hosszú karaktersorozat, pl. egy karakter maximum 4-szer fordul elő, legalább 2 nem betű karaktert tartalmaz, nincs periódusa, stb.)
- Az egyirányú függvény segítségével a szavazó előállítja jelszavának ellenőrzési kódját és megadja az adatbázis számára a rávonatkozó többi adatot is.
- Az $E(K_3, .)$ felhasználásával a szavazó lerejtjelzi az adatcsomagot és elküldi az A hatóságnak vagy az interneten keresztül egy e-mail üzenethez csatolva, vagy egy konferencia alatt közvetlen hálózati kapcsolaton keresztül.
- Az A hatóság megoldja a csomagot a $D(MK_3, .)$ segítségével, és kitölti a megfelelő értékeket a tagok adatbázisában.

Megjegyzés: a szavazó speciális szoftverrel ellátott számítógépek segítségével az A hatósággal személyes kapcsolatba kerülve (pl. egy konferencia alatt) is végrehajthatja a regisztrációt (az a gép, amelyiken titkosan beüti a jelszavát összeköttetésben van a másikkal, ahol az A hatóság az adatbázist tárolja). A szavazó végrehajthatja a regisztrációt a regisztrációs program letöltésével is. A programot használva létrehozza azt az adatcsomagot, amit egy e-mail üzenethez csatolva visszaküld az A hatóságnak. Az A hatóság felel azért, hogy a tagok adatbázisának adatai helyesek legyenek. Az A melléklet az 1. lépés részleteit mutatja.

Amikor lejárt a regisztrációs határidő, az A hatóság másolatot készít az érvényes jelszó ellenőrzési kódokról, és elküldi azt a B hatóságnak és a döntőbírónak. A B hatóság létrehozza a jelszó ellenőrzési kódok sorba rendezett adatbázisát. Kezdetben minden jelszó mező üres.

2. A jelöltek összegyűjtése a különböző posztokra
Ezt a szokásos módon lehet megtenni e-mail vagy levélposta útján. A jelöléseket egy hatóságnak kell összegyűjteni. Ez lehet az A hatóság is.

3. A szavazóprogram elkészítése az összegyűjtött jelöltek alapján
A szavazóprogram az alábbi lépéseket hajtja végre:

- a különböző posztokra megengedi a jelöltek közül megfelelő számú személy kiválasztását, és létrehoz egy M üzenetet, ami az adott kiválasztásokat kódolja;
- kiegészíti ezt az M kódot véletlen byte-okkal egy fix n hosszra, tegyük fel, hogy a kiegészített üzenet $M_1 = (M, \text{véletlen byte-ok})$;
- lerejtjelzi az M_1 üzenetet a PKS algoritmussal a K_1 nyilvános kulcs felhasználásával;

  Legyen $X = E(K_1, M_1)$ az új rejtjelzett üzenet, amit egy rejtjelzett szavazatnak tekinthetünk. A véletlen feltöltés miatt, ez még azonos kiválasztások esetében is különböző lesz.
- hozzáfűzi a szavazó jelszavát az X üzenethez;

  Legyen $Y = (X, J)$ az új üzenet, ahol J a titkos jelszó nyílt megadása.
- lerejtjelzi az Y üzenetet a PKS algoritmussal a K_2 nyilvános kulcs felhasználásával;

  Legyen $U = E(K_2, Y)$ a következő üzenet.
- lerejtjelzi az U üzenetet a PKS algoritmussal a K_3 nyilvános kulcs felhasználásával.

  Legyen $Z = E(K_3, U)$ a szavazóprogram kimeneti állománya.

Megjegyzés: a szavazó letöltheti a szavazóprogramot, és jelszava és választásai alapján létrehozhatja a Z kimeneti állományt. Ezután egy e-mail üzenethez csatolva elküldi a Z állományt az A hatóságnak. A B melléklet a 3. lépés részleteit mutatja.

4. A csatolt állomány leválasztása az e-mail üzenetről az A hatóság által
   - amikor az A hatóság megkapja az e-mail üzenetet a csatolt Z állománnyal együtt, akkor a döntőbíró számára megőrizve az üzeneteket a saját MK_3 titkos kulcsával megoldja a csatolt állományt megkapva az $U = D(MK_3, Z)$ blokkot. Csak az U állományt küldi el a B hatóságnak.

5. A szavazatok kicsomagolása az U csomagból a B hatóság által
   - A B hatóság megoldja az A hatóságtól kapott U állományt a saját MK_2 titkos kulcsával és megkapja az $Y = D(MK_2, U)$ üzenetet.
   - Mivel $Y = (X, \text{jelszó})$, a B hatóság az egyirányú függvény segítségével ki tudja számolni a jelszó ellenőrző kódot.
   - Megkeresi, hogy az adatbázisában a kapott kód szerepel-e azok között, melyeknél a jelszó mező még kitöltetlen. Ha szerepel, akkor a megfelelő helyre beírja a jelszót, érvényesnek tekinti az X szavazatot, és egy különálló helyen tárolja ezen összegyűjtött X szavazatokat, amiket elküld a C hatóságnak.

   Megjegyzés: ilyen módon egy jelszó csak egyszer használható, tehát egy szavazó csak egyszer szavazhat. A C melléklet mutatja az 5. lépés részleteit.

6. A szavazás értékelése a határidő lejárta után a C hatóság által
   - A C hatóság megoldja az X szavazatokat a saját MK_1 titkos kulcsával, és megkapja azt az M_1 üzenetet, melyre $M_1 = D(MK_1, X_1)$.
   - Mivel $M_1 = (M, \text{véletlen feltöltés})$, a C hatóság hozzájut az M valódi szavazathoz. Ezekből ki tudja számolni, hogy mely jelöltek győztek ezen a választáson.

   Megjegyzés: amikor a C hatóság hozzájut az M szavazathoz, nem tudja, hogy azt ki adta le.

   A D melléklet mutatja részletesen a 6. pont lépéseit.

7. A választás helyességét igazoló eljárás
   - az A hatóság ismét előállítja a regisztrációs csomagokból a tagok éppen aktuális adatbázisát;
   - a döntőbíró ellenőrzi, hogy az aktuális adatbázist helyesen állították elő;
   - a döntőbíró ellenőrzi az EF segítségével a B adatbázisban a jelszavak helyességét;
   - a B hatóság megismétli a döntőbírótól kapott jelszó ellenőrző kódok másolata alapján a kezdeti adatbázis létrehozását;
   - a bíró ellenőrzi, hogy az aktuális jelszó ellenőrző kódok megegyeznek-e a kezdeti értékekkel;
   - a B hatóság megismétli az akcióját a kezdeti adatbázisra és az A hatóság által megőrzött csatolt állományokra vonatkozóan előállítva a C hatóság számára az érvényesnek talált rejtjeles szavazatokat;
   - a bíró ellenőrzi, hogy a jelenlegi B adatbázis megegyezik-e a generálttal;

- a bíró ellenőrzi, hogy az érvényesnek talált szavazatok halmaza megegyezik-e azzal, amivel jelenleg a C hatóság rendelkezik;
- a C hatóság megismétli a szavazatok értékelését;
- a bíró ellenőrzi, hogy az eredmény ugyanaz-e.

Miután a szavazás helyességét megerősítették, a C hatóság megsemmisítheti az összes rejtjeles szavazatot, a B hatóság eltörölheti a jelszó ellenőrzési kód adatbázist és a csatolt állományokat és az A hatóság megsemmisítheti az e-mail üzeneteket. Az A hatóságnak el kell törölnie az összes jelszó ellenőrzési kódra vonatkozó bejegyzést a tagok adatbázisában.

Megjegyzés: az igazoló eljárásban a döntőbíró adja a programokat az A, B és C hatóság számára. A hatóságok csak saját titkos adatukat használják. Ezen programok nyilvánosságra hozhatók.

Az 1. táblázat áttekintést nyújt a sémáról:

IV. A kitűzött célok teljesítésének igazolása

A kitűzött célok az alábbi esetek valamelyikének bekövetkezése esetén nem teljesülnek:

1. Egy nem regisztrált szavazó szavazatát elfogadják.
   - Bárki letöltheti a szavazó programot és létrehozhatja a Z állományt. De az elfogadáshoz kell egy helyes jelszó ismerete, amit véletlen találgatással kitalálni nagyon valószínűtlen, mivel a jelszó 16 karakterből áll és van némi bonyolító tulajdonsága.
   - Ha az egyirányú függvény valóban egyirányú, akkor nem lehet kitalálni a helyes jelszót az egyirányú képének ismeretében.

   Ezért elhanyagolható az a valószínűség, hogy egy nem regisztrált szavazó szavazatát elfogadják.

2. Egy regisztrált szavazó többször szavaz.
   - Egy jelszót csak egyszer fogadnak el, így ha egy regisztrált szavazó kétszer akar szavazni, akkor szüksége van egy másik érvényes jelszóra.

   Annak valószínűsége, hogy egy regisztrált szavazó többször szavaz elhanyagolható.

3. Valaki meg tudja mondani, hogy egy szavazó mire szavazott.
   - Az A, B, C hatóságok együtt meg tudják mondani, hogy egy szavazó mire szavazott. De ha közülük bármelyik megőrzi a saját titkát, akkor senki sem tudja megmondani, hogy mire szavazott a szavazó.

   Ez csak akkor tehető meg, ha az A, B és C hatóságok egyike sem őrzi meg titkát.

Fázis	Szavazó	A részleg	B részleg	C részleg
Regisztráció	$JEK = EF(J)$ generálása,	JEK begyűjtése és beírása a személyi	–	–
	JEK elküldése	adatbázisba		
Regisztráció végén	–	JEK-rendezett lista elkészítése	–	–
Jelölés	Jelöltek támogatása	Jelöltek begyűjtése		
Előkészület	–	Szavazóprogram elkészítése	–	–
Szavazás	Szavazóprogram letöltése	(Z) Szavazatok összegyűjtése	–	–
	(Z) Szavazat generálása a Jelszó megadásával			
	(Z) Szavazat elküldése $\rightarrow A$			
Szavazás végén	–	JEK rendezett lista átadása $\rightarrow B$	A JEK rendezett lista alapján az	–
		(U) Kicsomagolt szavazatok átadása	(U) szavazat érvényességének megállapítása	–
Értékelés	–	–	(X) Érvényes szavazatok átadása $\rightarrow C$	(X) Kapott szavazatok kiértékelése
				Eredmény közzététele
Szavazás után	Végeredmény elfogadása vagy kétségbe vonása			
Ha az eredményt kétségbe vonták, akkor döntőbíró előtt	–	Rendezett JEK lista generálás	Rendezett JEK lista ellenőrzés	
		(U) Szavazatok átadása	(U) Szavazatok ellenőrzése	
			(X) Érvényes szavazatok átadása	(X) Érvényes szavazatok ellenőrzése
			–	(M) Eredmény ellenőrzése
Végeredmény elfogadása után	–	Szavazatok, JEK adatok törlése	Adatok törlése	Adatok törlése

1. táblázat

V. Lehetséges jelöltek az algoritmusokra

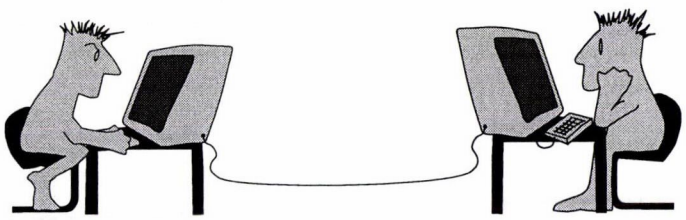
A kívánt algoritmusokra sok jelölt jöhet szóba. Itt egy javaslat, ami a PGP-ben is használt algoritmusokat ajánlja [2]:

a.
A PKS rendszer lehet az RSA-rendszer [4]. Válasszuk az első (K_1, MK_1) paramétereknél a modulust n bit hosszúnak (pl. $n = 1024$), a második (K_2, MK_2) paramétereket legalább $n +$ (jelszó bithossz) pl. 2048 bit hosszúnak. A (K_3, MK_3) paraméterek ugyanolyan hosszúak lehetnek, mint a (K_2, MK_2).

b.
A JEK előállításához felhasznált egyirányú függvény lehet egy szabvány hash függvény pl. $SHA - 256$ [6] vagy kiszámolhatjuk a JEK értéket egy blokkos algoritmus pl. az AES [3] felhasználásával. Legyen $S =$ (Jelszó,feltöltés) egy 128 bites vektor. Az első esetben legyen $T = SHA - 256(S, S)$, míg a második esetben $T = AES(K, S)$, ahol a K egy fix kulcs. Mindkét esetnél a JEK legyen a T első néhány bitje. A JEK javasolt mérete 64 bit. Ez az érték már elég kicsi valószínűségű eseménnyé teszi azt, hogy egy adott JEK értékhez találunk egy másik „értelmes" jelszót.

c.
A pszeudovéletlen generátort képezhetjük az AES blokkos rejtjelzéssel fix K kulcs mellett, ha azt egy fix hosszú karaktersorozatra alkalmazzuk. Ezt a karaktersorozatot csak egyszer használjuk, ezért nem szükséges megjegyezni. Legyen S egy 16 hosszú karakter sorozat. Generáljuk le a $T_i = AES(K, T_{i-1})$ 128 bites blokkokból álló sorozatot, $T_0 = S$. Legyen 64 bites pszeudovéletlen sorozat $W = W_0, W_{1,\ldots}$, ahol mindegyik W_i 64 bitből áll és $W_i = a(T_i)$ első fele.

A melléklet: Egy szavazó regisztrációja



lehetséges szavazó — a megbízható A hatóság

Rejtjeles csomag készítése:	A tagok adatbázisa (A hatóság)	Rendezett jelszó ellenőrző kód adatbázis (B hatóság)
Azonosító szám	*Id. Number*	*Jelszó ellenőrző kód JEK*
e-mail cím	*e-mail cím*	*Jelszó J*
Személyi adatok	*Személyi adatok*	
Jelszó ellenőrző kód JEK	*Jelszó ellenőrző kód JEK*	

A regisztrációs protokoll:

1.
A szavazó egy speciális program segítségével összeállít egy csomagot, ami tartalmazza a személyi adatait és jelszavának JEK ellenőrző kódját. A szavazó által választott J titkos jelszó csak erre a szavazásra érvényes, és az ellenőrzés miatt legalább kétszer egymás után adja meg. Végül az egész csomagot lerejtjelzi a $E(K_3,.)$ algoritmus segítségével.

($JEK = EF(J)$, ahol EF egy egyirányú függvény és J a titkos jelszó).

2.
Az A megbízható hatóság ellenőrzi a szavazó személyi adatait és eldönti, hogy szavazásra jogosult-e. Ha igen, akkor a jelszó ellenőrzési kódját felviszi a tagok adatbázisába.

3.
Amikor a regisztráció határideje lejár, az A hatóság másolatot készít a kitöltött jelszó ellenőrzési kódokról, lexikografikusan rendezett formában később átadja azt a B hatóságnak és a döntőbírónak.

B melléklet: A szavazó program (a szavazó használja)

1. A szavazó választ a jelöltek között:

Az 1. posztra (j_1 jelöltből legfeljebb $L1$ név választható)		A 2. posztra (j_2 jelöltből legfeljebb $L2$ név választható)		Az i. posztra (legfeljebb Li név j_i név közül)	
1.	1. jelölt neve	1.	1. jelölt neve	1.	1. jelölt neve
2.	2. jelölt neve	2.	2. jelölt neve	2.	2. jelölt neve
...		...		...	
j_1.	j_1. jelölt neve	j_2.	j_2. jelölt neve	j_i.	j_i. jelölt neve

Legyen

$$M = (V_1^1, V_2^1, \ldots, V_{L1}^1), (V_1^2, V_2^2, \ldots, V_{L2}^2), \ldots, (V_1^i, V_2^i, \ldots, V_{Li}^i)$$

a kódolt szavazat, ahol V_i^1 a V_i^1. névre utal az 1. poszt listájában, ($V_i^1 = 0$ azt jelenti, hogy arra a posztra senki sem volt kiválasztva). Például, ha $L1 = 2$ és $j_1 = 8$, akkor

$$M = (V_1^1, V_2^1), (V_1^2, V_2^2, \ldots, V_{L2}^2), \ldots, (V_1^i, V_2^i, \ldots, V_{Li}^i)$$

és ha

$$(V_1^1, V_2^1) = (2{,}8),$$

az azt jelenti, hogy az 1. posztra a lista 2. név és 8. név alatt megjelölt személyeket választották ki.

2. Egészítsük ki a kódolt szavazatot véletlen bitekkel egy fix N_1 hosszra:

$$M_1 = (M, \text{ véletlen vagy pszeudovéletlen bitek}).$$

3. Rejtjelezzük le az M_1 üzenetet a K_1 nyilvános kulccsal:

$$X = E(K_1, M_1) \text{ egy } N_1 \text{ hosszú rejtjeles szavazat.}$$

4. Adja meg a szavazó a titkos J jelszavát (az ellenőrzés miatt legalább kétszer) és fűzzük hozzá ezt a rejtjeles szavazathoz.

$$Y = (X, J) \text{ egy fix } N_2 = (N_1 + \text{ fix jelszó hossz}) \text{ hosszú üzenet.}$$

5. Rejtjelezzük le az Yüzenetet a K_2 nyilvános kulccsal:

$$U = E(K_2, Y) \text{ egy } N_2 \text{ hosszú üzenet.}$$

6. Rejtjelezzük le az U üzenetet a K_3 nyilvános kulccsal:

$$Z = E(K_3, U) \text{ egy } N_3 \text{ hosszú üzenet.}$$

a szavazó program által létrehozott állomány.

7. A szavazó elküldi a Z üzenetet, mint a saját szavazatát, az A megbízható hatóságnak az internet e-mail szolgáltatása segítségével.

C melléklet: Gyűjtőprogramok (az A és B hatóság számára)

1.
Az A hatóság kap az interneten keresztül egy e-mail üzenetet a szavazótól, amelyhez szavazatként hozzá van csatolva a Z állomány.

2.
Az A hatóság megoldja az ő MK_3 titkos kulcsával a csatolt Z állományt:

$$U = D(MK_3, Z) \text{ egy fix } N_2 \text{ hosszú blokk.}$$

3.
Az A hatóság gyűjti az U üzeneteket, majd a szavazási határidő lejárta után elküldi azokat a B hatóságnak, amely a saját MK_2 titkos kulcsával megoldja azt.

$$Y = D(MK_2, U) \text{ egy fix } N_2 \text{ hosszú blokk.}$$

Mivel $Y = (X, J)$, a B hatóság kiszámítja a $JEK = EF(J)$ értékét és megkeresi a JEK értéket a jelszó ellenőrző kód adatbázisban. Ha van ilyen kód és a hozzátartozó jelszó mező is üres, akkor a B hatóság beírja a J jelszó értékét ide, és az X rejtjeles szavazatot elküldi a C hatóságnak. Ha a jelszó mező nem üres, vagy nincs ilyen JEK érték az elfogadott jelszó ellenőrzési kódok között, akkor a B hatóság érvénytelennek minősíti a szavazatot, és nem küldi tovább a C hatóság részére.

Lexikografikusan rendezett jelszó ellenőrző kód adatbázis (B hatóság)	Titkos szavazatok adatbázisa (C hatóság)
jelszó ellenőrző kód JEK	*X titkos szavazat*
Jelszó J	

D melléklet: Kiértékelő program (a határidő lejárta után a C hatóság használja)

1. A C hatóság megoldja az összes rejtjeles X szavazatot a MK_1 titkos kulccsal:

Titkos szavazatok adatbázisa
X rejtjeles szavazat

A C hatóság azt kapja, hogy

$$M_1 = D(MK_1, X) \text{ egy fix } N_1 \text{ hosszú blokk.}$$

Mivel $M_1 = (M, \text{véletlen feltöltés})$ a C hatóság ismeri az M szavazatot:

$$M = (V_1^1, V_2^1, \ldots, V_{L1}^1), (V_1^2, V_2^2, \ldots, V_{L2}^2), \ldots, (V_1^i, V_2^i, \ldots, V_{Li}^i),$$

ahol V_2^1 az 1. poszt listájában a V_2^1. névre utal ($V_2^1 = 0$ azt jelenti, hogy senki sincs kiválasztva az 1. posztra).

2. A C hatóság kiszámítja a különböző posztokra az egyes indexek előfordulásának gyakoriságát, és kinyomtatja az 1. posztnál a leggyakrabban előfordult $L1$ számú nevet, a 2. posztnál a leggyakrabban előfordult $L2$ számú nevet, stb. Ez a lista mutatja a választás eredményét.

Irodalomjegyzék

[1] R. Cramer, M. Franklin, B. Schoenmakers, Moti Yung, Multi-authority secret-ballot election with linear work, in: *Advances in Cryptology – EUROCRYPT'96,* LNCS 1070, Springer (1996), pp. 72–83.

[2] Simson Garfinkel, *PGP: Pretty Good Privacy,* O'Reilly & Associates, Inc. (1995), ISBN 1-56592-098-8.

[3] FIPS PUB 197, *Advanced Encryption Standard (AES),* US Department of Commerse, National Bureau of Standards (2001).

[4] R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public key cryptosystems,* CACM 21 (1978).

[5] K. Sako and J. Kilian, Receipt-Free Mix type Voting Scheme – a practical solution to the implementation of a voting booth, in: *Advances in Cryptology – EUROCRYPT'95,* LNCS 921, Springer (1995), pp. 393–402.

[6] FIPS PUB 180-2, *Secure hash standard,* http://csrc.nist.gov/publications (2002).

LUKOVICS JÓZSEF
HUNGUARD KFT.
BUDAPEST
e-mail: luk8139@helka.iif.hu

A SECURE VOTING SCHEME FOR PROFESSIONAL ORGANIZATIONS FOR ONE OCCASION

JÓZSEF LUKOVICS

To organize an election in a professional organization is not a simple task, because it is impossible to call all the members to one place at one time. Usually the voting is realized by post-service. The organization takes care about the secrecy of the voting, the authentication of voters and that authenticated voters vote only once. In this article a voting scheme are supposed, which is fulfill these requirements. The protocol uses the Internet service instead of the post service and mainly simulate the currently used voting scheme. In this scheme the voters needs only a secret password (not a secret key). The voters can decide in more questions at the same time.

Alkalmazott Matematikai Lapok **23** (2006), 389–403.

KRIPTOGRÁFIAI KULCSVISSZAÁLLÍTÓ RENDSZEREK

PAPP PÁL

A kriptográfiai biztonság egyik alapeleme, hogy a véletlenszerűen választott üzenetkulcsot illetékteleneknek ne lehessen visszaállítani, megfejteni. Maguknak a felhasználóknak, vagy a törvény alkalmazóinak azonban szükségük lehet arra, hogy a rendelkezésre nem álló kulcsot rekonstruálják. Jelen dolgozatban ennek az ellentmondónak látszó feladatnak a megoldására kidolgozott eljárásokat ismertetjük, felvillantva a mögöttük rejlő matematikai ismereteket. A cikk második részében egy konkrét feladat kapcsán áttekintjük a felmerülő gyakorlati problémákat, és egy új, szimmetrikus kriptográfián alapuló megoldást mutatunk be egy részproblémára.

1. Bevezetés

Digitális üzenetek, dokumentumok biztonságának, hozzáférhetetlenségének egyik legfőbb eszköze egyértelműen dekódolható kódok alkalmazása. Az eljáráshoz lerögzítik kódoknak egy paraméteres halmazát, és a rejtjelzés ezek egyikének alkalmazásából áll. Az aktuális kód paraméterét véletlenszerűen választják az összes lehetséges paraméter közül. A kiválasztott paraméter értékét kulcsnak nevezik. A dekódoláshoz ezt a kulcsot kell ismerni, ennek ismeretében a dekódolás gyorsan végrehajtható művelet, ismeretének hiányában történő fejtés (rejtjelfejtés) viszont gyakorlatilag kivihetetlen. A kulcsok biztonságos kezelése adja meg az elvárható biztonságot.

Mind a hírközlés, mind az adattárolás esetében alapvető biztonsági rendszabály, hogy a rejtjelzett dokumentumokat és a rejtjelezésükhöz használt kulcsokat szigorúan elkülönítik egymástól. Így előállhat az a helyzet, hogy a kulcs(-hordozó) megsemmisül, míg a rejtjelezett dokumentum épségben megmarad. Természetesen merül fel tehát az igény arra, hogy ilyen esetekben is vissza lehessen állítani az eredeti nyílt szöveget. Mivel ez a kulcs nélkül az adatbiztonság alapvető követelménye

miatt gyakorlatilag lehetetlen, így az egyetlen megoldás olyan kulcsrendszer tervezését, alkalmazását jelenti, amelyben a legális felhasználó képes a kulcsot visszaállítani. A dolgozat címe ezt az igényt tükrözi, ennek a feladatnak a megoldására összpontosít.

Feladatunk nem merő fantázia kérdése. Az elmúlt néhány évben gyakorlatunkban több esetben fordult elő, hogy a kulcs megsemmisült. Találkoztunk olyan esetekkel is, amikor egy cég számítástechnikai alkalmazottai összehangolt zsarolást alkalmaztak a cég vezetőségével szemben: a teljes adatállományt lerejtjelezték, majd a visszaállításhoz szükséges kulcs kiadását sok milliós „váltságdíj" kifizetéséhez kötötték. Ezekben az esetekben magánszemély, vagy vállalat érdeke volt a kulcs visszaállítása. (A konkrét eseteinkben a kriptográfiai rendszer hibás megtervezése miatt ez gyorsan megtörténhetett.)

Van egy harmadik terület, ahol a kulcsvisszaállítás igénye országos jelentőségű. Jogi, ítélkezési, biztonsági követelmények kielégítése szükségessé teheti, hogy a kulcsot birtokló entitás hozzájárulása nélkül is elérhető legyen a kriptográfiai operáció által védett nyílt szöveg. Ezen a területen emlékezetes az akkoriban nagy felzúdulást váltott ki az USA kormányzatának Clipper chip projektje (1993). (1. Függelék)

Egy ilyen rendszer célkitűzése az, hogy a rejtjelzéssel védett kommunikációt éppúgy lehallgathatóvá tegye az államhatalom számára a megfelelő engedélyek birtokában, mint a nyílt kommunikációt. Ilyen rendszer tervezésénél több, ma megoldhatatlannak látszó matematikai és hírközlési probléma is jelentkezik. Problémát jelent a lehallgatható kommunikációs rendszer kikerülhetetlenségének biztosítása, a szükséges rendszer hatalmas mérete éppúgy, mint az emberi jogok betartásának biztosítására adandó, informatikai jellegű garanciák beépítése.

A kulcs esetleges, speciális helyzetben megtehető visszaállíthatósága természetes igényként jelentkezik az üzemeltető részéről a kriptográfiai rendszer működtetése során. Távolabbról nézve, vagy a rendszer biztonságáért felelősséget viselő pozícióból pedig minden ilyen igény ésszerűtlen, és bármilyen, ezt a célt szolgáló megoldás aláássa a rendszer biztonságát. A kompromisszumos megoldás szükségessége akkor válik nyilvánvalóvá, amikor kulcsvisszaállító rendszer működtetése nélkül az egyéni felhasználó a végleges adatvesztés miatti félelmében inkább nem is használja a kriptográfiát, illetve az államhatalom csak általa kezelhető erősségű kriptográfiai megoldásokat engedélyez azért, hogy szükség esetén gyakorolni tudja az információhoz való hozzáférésének jogát.

A kulcsvisszaállítás többféle célokat szolgálhat, s ennek megfelelően az alkalmazandó módszerek is jelentősen különböznek. Mi a három nagy részterületnek megfelelően foglaljuk össze őket. Az általános esetek felvázolás után egy konkrét gyakorlati problémát és annak megoldására kidolgozott módszerünket ismertetjük.

2. Kulcsvisszaállító rendszerek állami környezetben

Elvi szintű problémát jelent az a meggondolás, hogy míg a rendszer alapcélja a bűnözők lehallgatása, a társadalmat érő fenyegetések csökkentése, addig maga a kulcsvisszaállító rendszer visz be új támadási lehetőségeket az informatikai rendszerbe. A problémára adott legismertebb elvi szintű válasz S. Micali rendszere. (2. Függelék)

Magyarországon nem érzékelhetőek törekvések kulcsvisszaállító rendszer felállítására. Annyira nem, hogy a legutóbbi időkig a polgári szféra híján volt mindennemű informatika biztonsági jellegű hatóságnak. Ez a 2001-ben elfogadott Elektronikus aláírás törvénnyel részben megváltozott, mert a törvény hatósági, felügyeleti jogokat adott a Hírközlési Felügyeletnek (ma már Nemzeti Hírközlési Hatóság) az elektronikus aláírás termékekkel kapcsolatban.

A távközlési vonalak lehallgatása érzékeny kérdés hazánkban is. A lehallgatást minden esetben az igazságügyminiszter engedélyezheti, korlátozott időre. A törvény célja, hogy közvetlen politikai kontroll legyen a lehallgatást igénylő szervezetek (rendőrség, nemzetbiztonsági szolgálatok, pénzügyőrség, stb.) fölött.

3. Kulcsvisszaállító rendszer vállalati környezetben

A kulcsvisszaállító rendszerek alkalmazásának optimális területe a vállalati szféra, abban az értelemben, hogy ebben a nagyságrendben a feladat gyakorlati szempontból jól kezelhető, az irányítás egységes, centralizált, a rendszer elemei azonos biztonságpolitika alatt működnek. Ebben a körben is jelentős az igény ilyen megoldásokra. Ennek oka az, hogy a vállalati rendszerekben kezelt adatok nyilván a vállalat tulajdonát képezik, ezek az adatok fontosak, esetleg kritikusak lehetnek a cég működtetése szempontjából. Ezért biztosítani kell azt, hogy a cég akkor is hozzá tudjon férni a védett adatokhoz, ha a kulcsot birtokló alkalmazott ebben nem tud, vagy nem akar közreműködni. Ennek több oka lehet a kulcshordozó elvesztésétől kezdve egy véletlen balesetig. A továbbiakban a vállalati környezetben szükséges megoldásokat vizsgáljuk, bár sok részmegállapítás szinte változatlan formában igaz a másik két alkalmazási területre is.

A kulcsvisszaállító rendszer megtervezése előtt tisztázni kell, hogy mit várunk el a rendszertől, hogyan kapcsolódik az informatikai rendszer egyéb elemeihez, és azt is, hogy az informatikai rendszernek milyen segítséget kell adni a kulcsvisszaállító rendszer hatékony működéséhez.

Fontos eldöntendő kérdés, hogy milyen adatok visszaállítására van/lehet szükség. Nyilvánvaló, hogy még a kulcsvisszaállító rendszer segítségével is csak olyan adatok állíthatók vissza, amelyek később, amikor a visszaállítás szükségessé válik, legalább rejtjelzett formában hozzáférhetőek. Ez általában nincs így a rejtjelzéssel védett kommunikációs vonalak esetében.

Ha igényként merül föl a védett kommunikáció visszaállítása, akkor az különleges feladatokat ró a kulcsvisszaállító rendszerre. A legtöbb kommunikációs protokoll esetében az aktuálisan használt kapcsolati kulcs mindkét fél kulcsától függ, vagyis a visszaállításhoz mindkét fél kulcsára szükség van. Emellett a kulcsvisszaállító rendszernek képesnek kell lennie szimulálni azt a módszert, amivel a protokoll a két kommunikáló fél kulcsából az aktuális kommunikációs kulcsot előállítja. Ez sokszor, pl. a leggyakrabban használt TLS/SSL protokoll esetén meglehetősen bonyolult. A kulcskialakítás módja attól is függhet, hogy a két fél között az első kapcsolat jön létre, vagy már a sokadik.

Vállalati környezetben védett kommunikáció visszaállítása ritkán merül fel igényként, ez inkább a kormányzati szintű rendszerekre jellemző. A vállalati környezetre a tárolt adatok és az e-mail forgalom visszaállításának az igénye a jellemző.

A feladat konkrét technikai megoldása során általában két megoldási módot szokás követni. Az egyik esetben az aktuális rejtjelző kulcsot egy Megbízható Kívülálló (TTP = Trusted Third Party) nyilvános kulcsával is rejtjelezzük, és a kulcs rejtjeles képét együtt kezeljük (tároljuk, továbbítjuk) a rejtjelzett anyaggal. Így a rejtjelzett anyag mellett mindig rendelkezésre áll a rejtjelzéséhez használt kulcs, igaz, csak a TTP által elérhető, rejtjelzett formában. Ezt a megközelítést alkalmazza többek között a hazánkban általánosan alkalmazott PGP (Pretty Good Privacy, „tök jó biztonság") eljárás is. Ebben a TTP kulcsát Additional Decryption Key (= ADK, „kiegészítő megoldó kulcs)-nek nevezik. A megoldás hátránya, hogy a TTP privát kulcsának kompromittálódása minden, korábban végzett rejtjelzést veszélyeztet, ezért ezt kiemelten kell védeni. A megoldás tökéletes lehet egy magánszemély számára, azonban vállalati környezetben nem megfelelő.

A rendszer elterjedésénél a legkomolyabb problémát az okozza, hogy nincs ipari szabvány ezen a területen. Ez a technika a korábban készült alkalmazásokba utólagosan gyakorlatilag már nem építhető be. A korábban készült rendszerek fejlesztői sokszor nem elérhetők, a módosítást jogi problémák akadályozhatják, esetleg túl költséges ez a megoldás. Elmondhatjuk, hogy ezen a módon egységes, minden érintett applikációt lefedő kulcsvisszaállító rendszer gyakorlatilag nem építhető ki.

A nyilvános kulcsú kriptográfiai rendszerekben a másik gyakori megközelítési mód az, hogy a felhasználók titkos kulcsait tárolja le a TTP megfelelő védelem mellett. (Emlékeztetünk arra, hogy ezekben a rendszerekben a felhasználók két kulcsot használnak, egy nyilvános kulcsot, amelyet „telefonkönyv-szerűen" nyilvánosságra hoznak, és egy titkos kulcsot, amelyet a legszigorúbban őriznek. Egy megbízható harmadik fél, u.n. „hitelesítésszolgáltató" tanúsítja, igazolja, hogy a nyilvános kulcsot valóban az helyezte el, akinek vallja magát.) A védelem egyik eleme az, hogy a titkos kulcsot több TTP között osztják meg. Ehhez legtöbbször a Shamir–Blakley (függelék) sémát használják.

Ez a megközelítési mód akkor lehet hatásos, ha a cégen belül a kriptográfiai alkalmazások kulcsellátására egységes, centrális rendszert alkalmaznak. A gyakorlatban a kulcsvisszaállító rendszert célszerű a vállalati hitelesítés-szolgáltatási rendszerhez kapcsolni.

4. Kulcsvisszaállító rendszer egyéni használatra

Az alkalmazási spektrum másik végén az az igény áll, hogy egy egyedi felhasználó is legyen képes visszaállítani rejtjelzett fájljainak tartalmát, ha a kulcs hordozója elveszett, megsérült, vagy a felhasználó elfelejtette a kulcs védelméhez használt jelszót. A megjelenő igény valós, ugyanis a kriptográfiai védelem elterjedésének egyik legnagyobb gátja éppen az, hogy a megfelelően rejtjelzett adat gyakorlatilag senki által semmilyen módon nem helyreállítható a kulcs nélkül, s ettől a felhasználók idegenkednek. Megszokták, hogy a papíralapú kommunikációnál, de még a számítógépes adattárolás esetén is mindig van, marad esély az adatok visszaállítására. Ennél a megközelítésnél a problémát az okozza, hogy vagy bevonunk külső szereplőket a felhasználó mellé, aki, vagy aki segítségével a felhasználó vissza tudja állítani a védett adatokat, vagy csupán annyit teszünk, hogy a felhasználó környezetében a visszaállításhoz szükséges állományokat megtöbbszörözzük. Az első megközelítés új szereplőt hoz a színre, ezért hátrányos, míg a második megközelítésben ismét minden a felhasználóra van bízva, még ha a kulcselvesztés valószínűségét csökkentjük is.

5. Egy konkrét gyakorlati probléma

A cikk további részében egy konkrét feladat kapcsán esettanulmány jelleggel szeretnénk ismertetni egy nagyméretű, jelentős informatikai potenciált használó cég esetében felmerült meggondolásokat, illetve az ott alkalmazott egyedi megoldást. A cég már eddig is jelentős, részben központosított infrastruktúráját kulcsvisszaállítási képességekkel kívánta kiegészíteni. Az informatikai rendszerben több különböző korú, és különböző forrásból származó kriptográfiai képességekkel rendelkező szoftvert használnak, ezért az Additional Decryption Key-re épülő módszereket ki kellett zárni. Az alkalmazott informatikai rendszerben a rejtjelzéssel kapcsolatban felmerült igények meglehetősen tipikusak. Szükség van:

- Titkosított e-mail forgalomra,
- Rejtjelzésel védett lokális adattárolásra,
- Adatbázisrekordok védett tárolására, ezek távoli elérésére,
- Kommunikációs kapcsolatok rejtjelezéssel történő védelmére (TSL, SSH, különböző, főként IPSEC alapú VPN megoldásokra,
- A rendszerben egy belső hitelesítésszolgáltató biztosítsa a nyilvános kulcsú infrastruktúra alapjait, amelyhez saját fejlesztésű RSA kulcsgenerátor is csatlakozzon, így garantálva a kulcsgenerálás biztonságát, átláthatóságát, auditálhatóságát.

Fontos eleme a biztonsági környezetnek, hogy a felhasználók titkos kulcsai modern, a legfrissebb támadási módszerek ellen (Differential Power Analysis, Fault Analysis) is megbízható smart cardra kerülnek. A rendszer rendelkezik olyan védett adatközponttal, amely megbízható fizikai védelmet adhat egy olyan kulcsnak,

amelynek megismerése a teljes informatikai rendszer biztonságára kihat. Ilyen kulcs a PKI Hitelesítés Szolgáltató mester kulcsa, de egy bevezetendő kulcsvisszaállító rendszer is egy ilyen, kiemelt védelmet igénylő kulcsot visz a rendszerbe. Rendelkezésre áll olyan szervezeti egység is, amely alkalmas lehet a Megbízható Kívülálló (Trusted Third Party) szerepére is. Az adatközpont kulcsainak védelme a teljes rendszer szempontjából rendkívül fontos, ezért ennek védelmét kiemelt fizikai biztonsági és rezsim intézkedésekkel kell biztosítani.

A kulcsok generálása központilag történik, vagyis lehetőség van arra, hogy egy kulcsvisszaállító rendszer a központban működve, a központi kulcsgenerálásra alapozva tegye lehetővé az egyedi felhasználói kulcsok elérését.

Mivel a rendszer smart cardokat alkalmaz kulcshordozóként, a kulcsok generálásának kézenfekvő módja a smart card kulcsgeneráló funkciójának használata, amely azt is lehetővé teszi, hogy a generált kulcs ne legyen kiolvasható, kívülről elérhető, csak a kártya operációs rendszere számára legyen látható. Ez a megoldás, bár jelentősen növeli a biztonságot, csökkenti a szervezési feladatokat és növeli a felhasználók rendszerbe vetett bizalmát, megakadályozza, hogy a kulcsvisszaállító rendszer hozzáférjen a generált kulcsokhoz.

Ugyanakkor a kártyán generált kulcsok optimálisak a digitális aláíráshoz használt kulcsok esetében, sőt a gyakorlatban ez az egyetlen módszer, amely a digitális aláírás letagadhatatlanságát biztosítani tudja. (A kártyán kívül más entitás nem ismeri, nem kerül kapcsolatba a felhasználó digitális aláírásra használt titkos kulcsával).

Ma már elfogadott szakmai tény, hogy nem célszerű digitális aláírásra és rejtjelzésre ugyanazt a kulcsot használni. Ezt az álláspontot képviseli a jelenleg érvényben levő elektronikus aláírás-törvény is. A kétféle kulcsnak más az életciklusa, szerepe, mások az esetleges visszavonás körülményei, mások a kompromittálódás kockázatai. Ezért a megoldás lényege az, hogy míg a digitális aláíráshoz használt kulcsok a kártyán generálódnak, addig a rejtjelzésre használt kulcsokat egy külső, a Hitelesítés Szolgáltató rendszer részeként működő modul generálja, és így lehetővé válik az is, hogy egy kulcsvisszaállító rendszer hozzáférjen a generált kulcsokhoz mielőtt azok a kártyára felírásra kerülnek.

6. A javasolt megoldás

Az alkalmazandó kulcs-vissszaállító rendszerrel szembeni elvárások az alábbiakban foglalhatók össze:

- Legyen képes a központilag generált, rejtjelzésre használandó kulcsok visszaállítására.
- Legyen képes tárolt rejtjelzett fájlok, illetve adatbázisrekordok visszaállítására a rejtjelzést kezdeményező, illetve végrehajtó felhasználó együttműködése nélkül is.
- Legyen képes a rejtjelzett e-mail forgalom visszaállítására is.

- A rendszer kialakítása olyan legyen, hogy a potenciális felhasználókban bizalmat ébresszen, garanciákat adjon arra, hogy a kulcsvisszaállító rendszer csak indokoltan, megfelelő kontroll mellett kerül felhasználásra.
- A rendszer rendelkezzen olyan kriptográfiai mechanizmusokkal, amelyek a biztonság szubjektív sugalmazása mellett objektíven is védelmet adnak, biztosítják az előző pontban leírt célok megvalósulását.
- A digitális aláíráshoz használt kulcsok visszaállítása nem elvárás.
- Védett kommunikációs kapcsolatok (kivéve e-mail) visszaállítása nem elvárás.

A tervezett megoldás lényege, hogy az RSA kulcsgenerátor által elkészített kulcsot megkapja a kulcsvisszaállító rendszer. A kulcsvisszaállító rendszer a kulcsigénylés alapadatai mellett rejtjelzett formában tárolja az igényelt kulcs titkos részét. A rejtjezés visszaállításához szükséges kulcsot a Shamir séma (ld. függelék) segítségével szétosztjuk a titokbirtokosok között. Ennek az úgynevezett küszöbsémának a lényege az, hogy n kiosztott információdarab, úgynevezett árnyék közül bármely m elegendő a titok visszaállításához úgy, hogy bármely $m-1$ árnyék (a sémától függően gyakorlatilag, vagy elméletileg is bizonyítható módon) semmi információt nem ad a titokra. Az alkalmazott séma *véges test fölötti polinom interpoláción* alapul.

Az adatbázis titkos részének védelmére olyan módszert kell alkalmazni, amely annak ellenére biztosítja a titkos kulcsok védelmét, hogy a rejtjelzéshez szükséges kulcs megjelenik a rendszert futtató számítógép memóriájában, hiszen a folyamatosan keletkező titkos kulcselemeket védeni kell. Dolgozatunkban erre a problémára javasolunk egy szimmetrikus kriptográfián alapuló gyors, hatékony megoldást.

Nyilván nem kivitelezhető az a megoldás, hogy a titokbirtokosoknak minden kulcsrejtjelzés előtt vissza kell állítaniuk az adatbázis védelmét biztosító kulcsot.

Kézenfekvő megoldás, hogy az adatbázis védelmére nyilvános kulcsú megoldást alkalmazzunk, hiszen ekkor az adatbázis védelmét biztosító, az újabban keletkező rekordok rejtjelzését lehetővé tevő kulcsot az operációs rendszerben tarthatjuk, míg a megoldáshoz, a védett felhasználói kulcsok hozzáféréséhez az adatbázisvédő kulcs titkos párja szükséges, amelyet a titokbirtokosok megosztva őriznek.

Mivel azonban a nyilvános kulcsú algoritmusok nagyságrendekkel lassabbak a szimmetrikus algoritmusoknál, felmerül a kérdés, hogy az elvárások teljesíthetők-e szimmetrikus kriptográfiai algoritmus alkalmazásával is.

6.1. A múlt titkosságának megőrzése

Bizonyos kriptográfiai protokolloknak van olyan tulajdonsága, hogy az aktuális kommunikációt védő kulcs kompromittálódása nem segíti a támadót a későbbi védett kapcsolatfelvételek megtámadásában. Ezt a tulajdonságot angolul forward secrecy típusnak nevezik. Tipikus megoldási mód, hogy a rejtjelzéshez szükséges kulcsot a kapcsolat elején hozzák létre, például a Diffie–Hellmann kriptorendszer segítségével. A forward secrecy értékes tulajdonság, mert olcsóbbá, biztonságosabbá teszi a rendszer működését, és a rendszer biztonságának bevizsgálása is könnyebb.

Esetünkben a megoldandó alapprobléma az, hogy az operációs rendszerben jelen levő kulcs ne kompromittálja a korábban rejtjelzett rekordokat, vagyis az aktuális kulcs ismerete ne adjon lehetőséget a korábban rejtjelzett rekordokhoz való hozzájutásra. Szükséges még az is, hogy a rekordok rejtjelzésére használt kulcsok származtathatók legyenek abból a mesterkulcsból, amit a titokmegosztási séma segítségével szétosztottunk a visszaállítással foglalkozó (recovery) adminisztrátorok között.

A forward secrecy mintájára ezt az elvárást a továbbiakban nevezzük backward secrecy-nek.

6.2. Az implementáció részletei

Az implementált megoldás lényege, hogy a megosztott mester-kulcsból (Sharedmasterkey) az első rejtjelzés előtt létrehozzuk az AK0 alapkulcsot. Ehhez egy speciális egyirányú transzformációt használunk. Az alapkulcsból iterációval hozzuk létre az AKi Aktuális Kulcsokat. Az AKi kulcsból minden rekord rejtjelzése után egy egyirányú transzformációval létrehozzuk az aktuális AK($i+1$) kulcsot. Egyirányú transzformációként az USA szabványban szereplő SHA256 hash-függvényt használjuk. Ennek nagyobb outputja csökkenti a rövid sorozatok hash-eléséből következő elvi sebezhetőséget.

Az AKi.kulcsból és az i. rekord névmezőjéből származtatjuk a rejtjelzésére használt ENCRYPTKEYi kulcsot. Az előző kulcsra gyakorlatilag nem lehet visszakövetkeztetni, de mivel természetesen van kapcsolat az AKi kulcsok között, a választott megoldás gyakorlati és nem elméleti szinten ad biztonságot.

A kulcsvisszaállító rendszer egy külön modul, amely inputként a megosztott mesterkulcs visszaállításához szükséges árnyékokat, a visszaállítandó kulcs i. sorszámát és a felhasználó nevét (commonname) kéri be. Az alrendszer outputja az ENCRYPTKEYi, illetve a kapcsolódó kezdővektor. Az ENCRYPTKEYi szerepe az, hogy a kulcsvisszaállítás egyedivé váljék, csak a kérdéses rekord kerüljön visszaállításra. Ha közvetlenül az AKi kulcsot használnánk rejtjelzésre, akkor nemcsak az i. rekord, hanem minden utána következő is visszaállíthatóvá válna.

A kulcsadatbázis létrehozása során egy rekordjának szerkezete a következő típusú lesz:

A kulcsbirtokos azonosítói

- i, azaz a AKi sorszáma
- A Privi titkos kulcs
- Hitelesítő kód az előző mezőkre

azaz

- Commonname(i)
- i, ENC(AKi, PRIVi)
- MAC BKi.

A kulcsbirtokos azonosítója bármi lehet, ami megfelel az x509v3 tanúsítványformátumnak. Emellett szerepel a cégnél bevezetett univerzális (minden személy-

zeti és jogosultsági adatokat tároló) adatbázisban használt univerzális azonosító is.

A sorszám egy két bájt méretű nem negatív egész.

Az AKi kulcsot az f egyirányú függvénnyel hozzuk létre az AK($i-1$) kulcsból $i > 0$ esetén. Az AKi kulcs mérete 128 bit. Rejtjelző algoritmusként bármelyik modern algoritmus használható, amelyet megfelelő erővel vizsgált a nemzetközi kriptográfus közösség. Elvárásnak tekintjük, hogy blokkos legyen az algoritmus, és a blokkméret legyen legalább 128 bites. Ez a gyakorlatban azt jelenti, hogy valamelyik, az AES projectre készült algoritmust célszerű használni.

Az utolsó mezőben elhelyeztünk egy kontrollösszeget, amely a rekord tartalmát, annak integritását védi szándékos módosítás ellen is. Erre a célra a **HMAC-SHA1** USA-szabvány algoritmust használjuk. A HMAC algoritmushoz az AKi kulcsból származtatott BKi kulcsot használjuk MAC értékként az algoritmus outputjának első 14 bájtját tároljuk le. (Az SHA1 ellen a közelmúltban nyilvánosságra került támadási mód nem veszélyezteti a HMAC részekért történő biztonságos használatot.)

Javasolt megoldásunk főbb pontjai:

- Az installálás folyamán a rendszer generálja a SHARED-MASTERKEY kulcsot, amely 800 bit méretű véletlen sorozat,
- Elkészítjük a templ kulcsot a templ=3DSHA256 [SHARED-MASTERKEY],
- Legyen t=3Dtempl[0]&7,
- A templ értékére végezzük el az SHA256 operációt t-szer egymás után, s az eredmény legyen a 0. rekord rejtjelzésének alapkulcsa. RECORDKEY0 =3DSHA256 t [templ],
- Az i. rekord rejtjelzéséhez az alapkulcsot az előző alapkulcsból kapjuk,
- RECORDKEYi =3D SHA256[RECORDKEYi-1] $i > 0$ esetén,
- Az i. rekord rejtjelzéséhez szükséges kulcsot és kezdővektort a következőképpen kapjuk: (ENCRYPTKEYi IVi) =3D SHA256[RECORDKEYi-1,felhasználó-név]. Itt ENCRYPTKEYi IVi egyaránt 128 bites elemek. A rejtjelzést az AES algoritmussal CBC módban végezzük,
- A sikeres rejtjelzés után $i = 3Di + 1$, kiszámítjuk RECORDKEYi új értékét, és az (i, RECORDKEYi) párost védett formában fájlba is elmentjük.

7. Záró megjegyzések

Dolgozatunkban egy konkrét, hatékony, szimmetrikus kriptográfián alapuló módszert adtunk arra a problémára, hogy a kulcsvisszaállító rendszerekben a rekordok védelmére használt kulcs folyamatosan a memóriában van, ezzel számítástechnikai értelemben veszélyeztetve a biztonságot. A javasolt séma biztosítja, hogy a már letárolt adatok nem fejthetők vissza a gép memóriájában levő éppen aktuális kulcs ismeretében sem.

A séma alkalmazása során egymás után többször hash-elünk viszonylag rövid bitsorozatokat. Az ebből adódható problémák elkerülése miatt választottunk viszonylag hosszú, 256 bites outputtal rendelkező algoritmust.

A témakörhöz tartozó további gyakorlati probléma, hogy mi a teendő akkor, ha az egyik árnyékbirtokos elveszti az árnyékot tartalmazó adathordozót. Erre jelenleg nincs jobb megoldásunk, mint az alaptitok visszaállítása és újraosztása.

Szükséges lenne egy olyan protokoll kidolgozása, amely lehetővé teszi, hogy az árnyék birtokosoknak ne kelljen személyesen, egy időben megjelenni egy felhasználói kulcs visszaállításához.

Hivatkozások

[1] Abelson, H., Ross Anderson, R., Steven M., Bellovin, S. M., Benaloh, B., Blaze, M., Diffie, W., Gilmore, J., Neumann, P. G., Rivest, R. L., Schiller, J. I. and Schneier, B., The Risks of Key Recovery, Key Escrow, and Trusted Third Party Encryption, *The World wide Web Journal* **2/3** (1997).

[2] Dennings, D. and Branstad, D. K., A Taxonomy of Key Escrow Encryption Schemes, *Communications of ACM,* Vol. 39, n. 3.

[3] HMAC, Keyed Hashing and Message Authentication, *FIPS 198* (2002).

[4] Micali, S., Fair Public-Key Cryptosystems, *Laboratory for Computer Science, MIT, Crypto* **92** (Aug. 21, 1992).

[5] 2001. évi XXXV. törvény az elektronikus aláírásról.

[6] Diffie, W. and Hellman, M.E., New directions in cryptography, *IEEE Transactions on Information Theory,* **22** (1976), 644–654.

[7] Rivest, A., Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of ACM,* Vol. 21, n. 2

[8] Escrowed Encryption Standard (EES), *FIPS 185* (1994).

[9] Shamir, A., How to Share a Secret, *Communications of ACM,* Vol. 22, n. 11

1. Függelék

A CLIPPER project áttekintése

A CLIPPER chip az amerikai NSA által 1993-ban kifejlesztett, alapvetően beszédrejtjelzésre kifejlesztett chip. A chipet az AT&T 3600-as telefonba szánta az NSA. A chip a SKIPJACK nevű algoritmust tartalmazza, amelyet erre a célra fejlesztettek ki. Az algoritmust titokban kívánták tartani, ezért nem esett át nyilvános értékelésen, és a chipet is úgy tervezték hogy a tartalom visszafejtését minél jobban megnehezítsék. Azóta az algoritmust nyilvánosságra hozták. Az algoritmus 64 bites blokkmérettel dolgozó blokkos algoritmus (mint a DES), 80 bit kulcsmérettel, 32 iterációs ciklussal. Sebessége 15 Mbit/sec. A hardware az akkori legmodernebb

(„0.8 mikronos") technológiával a kaliforniai MYKOTRONIX cégnél a legszigorúbb biztonsági, titokvédelmi körülmények között készül. A chip kiolvasás, utólagos megismerés ellen védett. A 80 bites kulcs akkoriban a még regnáló DES algoritmus 56 bites kulcsméretéhez képest meglehetősen sok volt, de a mai gyakorlatban használt legmodernebb algoritmusok minimum 128 bites kulcsához képest meglepően kevés. Az algoritmust Output Feedback módban használták bitfolyam előállítására.

A CLIPPER chip érdekességét a beépített kulcsvisszanyerő módszer adta, amelynek segítségével a tervek szerint az amerikai hatóságok hozzájuthattak volna az aktuális beszélgetés rejtjelzéséhez használt kulcshoz. A koncepció lényeges eleme volt, hogy középtávon csak ilyen chippel ellátott eszközöket lehet majd használni. Az ötlet komoly felzúdulást váltott ki a civil társadalomban, melynek központi kérdése az volt hogy indokolt-e a magánszféra ilyen mértékű korlátozása. Az NSA a koncepció kongresszusi jóváhagyása előtt kiadatta az Escrowed Encryption Standard nevű szabványt (EES.)

Ugyanakkor szakmai vitát is kiváltott a koncepció A kulcsfelfedő rendszer lényege, hogy minden chip rendelkezik egy egyedi azonosítóval, az ennek megfelelő egyedi mesterkulccsal s egy „családi" kulccsal, amely azonos a chipek egy nagy csoportjában. A rejtjelzést kezdeményező chip rendelkezik egy előzetesen kialakított egyszeri kulccsal (session key) amellyel a rejtjelzés történni fog. A rejtjelzés elején a chip létrehoz egy mezőt, (Law Enforcement Field, LEF) amely a következőképpen jön létre; az egyszeri kulcs mesterkulccsal rejtjelzett képe, a chip sorszáma, s néhány bit redundáns információ. Mindezt lerejtjelezzük a családi kulccsal, s megkapjuk a LEF-et. A fogadó chip csak akkor kezdi el a megoldását a kapott rejtjelesnek, ha a megkapott LEF korrekt. Ez azt jelenti, hogy a kapott LEF-et megoldja a családi kulccsal, s az ekkor kapott mezőnek „egy LEF-nek megfelelően kell kinézni." (Pl. ha a LEF-be betett redundáns információ az „USA1993", akkor a fogadó oldal csak akkor fogadja a rejtjelzést, ha a családi kulccsal megoldott LEF-mező végén is ez található.) A hatóságok pedig a chip sorszámának ismeretében adatbázisukból hozzájutnak a chip mesterkulcsához, s ezután a mesterkulcs, s a LEF ismeretében az egyszeri kulcshoz is. Ezután a kommunikáció fejthető. Visszaélések megakadályozására a chip készítésekor a mesterkulcsot két, önmagában semmitmondó komponensre bontják, s két külön adatbázisba helyezik el. A két adatbázist két független, az Igazságügyminiszter által megbízott szervezet birtokolja. Amennyiben bíróság engedélyezi a lehallgatást, a chip sorszámának ismeretében mindkét szervezet átadja a mesterkulcs általa birtokolt komponensét. Arról, hogy a hatóságok az engedély lejárta után ne használják a megkapott mesterkulcsot, rezsim szabályokkal kívántak gondoskodni.

A szakmai és politikai problémák miatt a koncepció nem került át a gyakorlatba, fokozatosan elhalt. A szakirodalomban továbbra is fontos téma egy, minden jelentős követelménynek eleget tevő kulcsvisszaállító rendszer kifejlesztése, de a gyakorlatban a kérdés visszaszorult a vállalati szféra területére.

2. Függelék

Micali fair nyilvános kulcsú rendszere

Micali rendszerének célja egy olyan nyilvános kulcsú infrastruktúra létrehozása, amely az EES-hez hasonlóan lehetővé teszi bizonyos esetekben az államhatalom számára a rejtjelzett üzenetekhez való hozzáférést, ugyanakkor garanciákat is ad arra nézve, hogy ez csak a megfelelő bírósági eljárások után tehető meg. A gondolat lényege, hogy a felhasználók a szokásos PKI rendszerhez hasonlóan generálnak önmaguk számára egy magánkulcs-nyilvános kulcs párt. (Az Elektronikus aláírás törvényben magánkulcsnak nevezik a nyilvános kulcsú rendszerekben használt kulcspárok titokban tartandó részét.) A magánkulcsot több (legyen ez pl. öt) részre osztják, és ezt az öt részt eljuttatják öt különböző megbízotthoz. A szétosztásnak a következő követelményeket kell teljesíteni:

1. Az öt darab birtokában a magánkulcsnak rekonstruálhatónak kell lennie.
2. A magánkulcs ötnél kevesebb darabból nem rekonstruálható. (Ötnél kevesebb darab információelméleti értelemben nem ad információt a magánkulcsról)
3. Minden egyes darabról egyedileg megállapítható, hogy az egy valós, korrekt darabja a magánkulcsnak.

Az utolsó feltétel teljesítése adja a séma erejét. Nyilvánvaló, hogy a magánkulcs ismeretében a darabok korrektsége ellenőrizhető. Azonban ebben a sémában ezt úgy kell megtenni, hogy a megbízottak ellenőrizhessék a hozzájuk eljuttatott darab korrektségét (azaz azt, hogy az öt darabból szükség esetén tényleg előállhat a kulcs), de maguk, sőt ötük közül négyen együtt se juthassak információhoz a magánkulcsról.

Ezek alapján a séma a következő:

- A felhasználók a szokásos módon készítenek maguknak egy kulcspárt. A magánkulcsot öt részre vágják, és egy-egy darabot és a nyilvános kulcsot egy megadott protokoll szerint elküldik az öt megbízottnak.
- Minden megbízott ellenőrzi, hogy a neki eljuttatott darab korrekt-e, és erről értesíti a tanúsítványkibocsátót. (hitelesítésszolgáltatót – HSZ – a magyar terminológiája szerint).
- A HSZ kiadja a felhasználó tanúsítványát a nyilvános kulcsáról, ha mind az öt megbízottól pozitív válasz érkezett.
- Amennyiben bírósági engedély érkezik a felhasználó kommunikációjának lehallgatására, a megbízottak átadják titokdarabjaikat a kormányzatnak, s ezzel lehetővé válik a felhasználó lehallgatása.

A Micali-séma kritikus részének, a titokdarabok megbízottakhoz való eljuttatásának módjának gyakorlati megvalósítása minden rendszerben más. Elvi szinten a főbb lépések a következők:

1. A felhasználó generál egy kulcspárt magának, és annak nyilvános részét átadja a megbízottaknak.
2. Lerejtjelzi, leképezi a magánkulcsot egy adott módon.

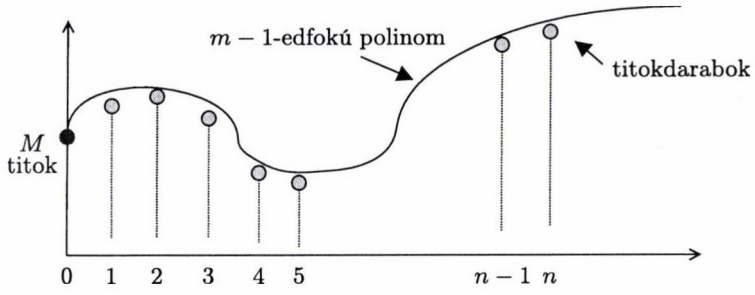
3. A megbízott megkapja a lerejtjelzett képet, és egy Zero-knowledge protokollon alapuló bizonyítékot arról, hogy a rejtjelzés alatt levő kép valóban a megadott nyilvános kulcshoz tartozik.
4. A rejtjelzés alatt levő képet szétosztja egy Verifiable Secret Sharing séma segítségével.

3. Függelék

Shamir-féle titokmegosztási séma

A. Shamir dolgozott ki egy olyan sémát, amellyel egy információ redundánsan többfelé osztható. A feltalált sémát (n, m) küszöbsémának nevezzük $n \geq m$. Lényege, hogy a szétosztandó titokból n darab képet állítunk elő. A darabokat árnyéknak is nevezzük. A szétosztott titok bármely m árnyék birtokában visszaállítható, de $m-1$-ből még nem.

A módszer a Langrange-féle polinominterpoláción alapul. Ennek lényege az, hogy egy $m-1$-edfokú polinom együtthatóit vissza tudjuk állítani, ha ismerjük az m különböző helyen felvett értékét. Legyen a titok a 0 helyen felvett érték, (vagyis a konstans együttható) és legyen a többi együttható véletlenül választott. Ha n részre akarjuk bontani a titkot, akkor legyenek a titokdarabok a polinom $1, 2, \ldots, n$ helyen felvett értékei. Ha az n érték közül tudunk m darabot, s azt, hogy az adott értéket hol vette fel a polinom, akkor a polinom együtthatói visszaállíthatóak. Így a 0. együttható is, ami a titok.



Ha a polinomot véges test fölött tekintjük, akkor az együtthatók értékei korlátosak lesznek, s pontosan látható, hogy mekkora alaptestet kell választanunk ahhoz, hogy egy adott bithosszúságú információt szét tudjunk osztani.

A fentieket konkretizálva, egy lehetséges matematikai keret:

Válasszunk egy p prímszámot, amely nagyobb mint a titok (ha a titkot mint bitekkel felírt számnak tekintjük). Legyen (n) a kiosztandó titokdarabok (árnyékok) száma, és legyen m árnyék szükséges az M titok rekonstruálásához. Véges test feletti algebrai egyenleteket használunk, ami azt jelenti, hogy az alapműveleteket modulo p végezzük el.

A titok megosztásához generáljunk egy tetszőleges $m-1$-ed fokú polinomot a fenti p elemű test felett. Tehát ha egy (n,m)-megosztási sémát akarunk definiálni, akkor generáljunk egy $m-1$-edfokú

$$\left(a_{m-1}x^{m-1}+a_{m-2}x^{m-2}+\ldots+a_1x+M\right) \bmod p$$

polinomot, ahol a p véletlen prím, nagyobb valamennyi generált együtthatónál. Az a_j $(j=1,\ldots,m-1)$ együtthatók véletlenszerűen választottak, titkosak és az árnyékok szétosztása után eldobhatóak. M azonos a titokkal, a p prím akár nyilvánossá is tehető.

Az árnyékokat a polinom n különböző pontban kiszámolt helyettesítési értékeként nyerhetjük:

$$k_i = F(x_i) \bmod p \qquad (i=1,\ldots,n).$$

Más szóval, az első árnyék lehet a polinom értéke az $x=1$ helyen, a második árnyék lehet a polinom értéke az $x=2$ helyen, és így tovább. Ezzel lezártuk a feladat első részét, a titok szétosztását.

Mivel egy $m-1$-edfokú polinomnak m ismeretlen együtthatója van (most: $a_{m-1},\ldots,a_1$ és M), bármely m árnyék segítségével felállítható m lineáris egyenlet, amelyben az együtthatók az ismeretlenek. Ugyanakkor $m-1$, illetve ennél kevesebb árnyék nem elegendő a lineáris egyenletrendszer megoldásához, míg m-nél több árnyék már redundanciához vezet. Ha tehát megoldjuk a lineáris egyenletrendszert (például Gauss-eliminációval), akkor a megoldások között szerepel a nulladfokú tag M együtthatója is. Így megoldottuk a feladat második részét is, visszaállítottuk a titkot. Vegyük észre, hogy nincs szükségünk az összes együtthatóra, csak az M-re. Ez az ötlet gyorsíthatja az egyenletmegoldást.

Egy rövid példán, amelyben n és m értéke nem nagy, szemléletesen bemutatható a fenti séma és megoldása.

Példa egy $(3,5)$-titokmegosztási sémára, amelyben bármely három résztvevő együttesen rekonstruálni tudja az M titkot.

Legyen a titok $M=11$. Legyenek a másodfokú polinom véletlenszerűen választott együtthatói 7 és 8, a véges test elemszáma pedig 13. Tehát a következő másodfokú polinomot generáltuk:

$$F(x_i) = (7x^2+8x+11) \bmod 13$$

A „szétosztott" öt árnyék a következő:

$$k_1 = F(1) = 7+8+11 \equiv 0 \pmod{13}$$

$$k_2 = F(2) = 28+16+11 \equiv 3 \pmod{13}$$

$$k_3 = F(3) = 63+24+11 \equiv 7 \pmod{13}$$

$$k_4 = F(4) = 112+32+11 \equiv 12 \pmod{13}$$

$$k_5 = F(5) = 175 + 40 + 11 \equiv 5 \pmod{13}$$

Rekonstruáljuk az M titkot tetszőleges három árnyékából. Legyenek ezek például k_2, k_3 és k_5. Ekkor a következő lineáris egyenletrendszert kapjuk:

$k_2 = 3$ esetében: $a \cdot 2^2 + b \cdot 2 + M = 3 \pmod{13}$

$k_3 = 7$ esetében: $a \cdot 3^2 + b \cdot 3 + M = 7 \pmod{13}$

$k_5 = 5$ esetében: $a \cdot 5^2 + b \cdot 5 + M = 5 \pmod{13}$

azaz:

$$4 \cdot a + 2 \cdot b + M = 3 \pmod{13}$$

$$9 \cdot a + 3 \cdot b + M = 7 \pmod{13}$$

$$25 \cdot a + 5 \cdot b + M = 5 \pmod{13}$$

A három egyenletből álló egyenletrendszerünk megoldása: $a = 7$, $b = 8$ és $M = 11$. Valóban visszanyertük az M titkot.

A Shamir-féle séma segítségével a résztvevők fontossága, súlya is különböző lehet, ha egyesek egynél több árnyékot kapnak.

A séma nagy hiányossága, hogy feltételezi a résztvevők korrektségét, azaz az árnyékról annak birtokosa nem tudja megállapítani, hogy az valóban a titoknak egy darabja-e, vagy csupán egy véletlen szám. Ezen segítenek az ellenőrizhető titokmegosztási sémák (Verifiable Secret Sharing, VSS)

KEY RECOVERY SYSTEMS IN CRYPTOGRAPHY

PÁL PAPP

One of the most important element of the security of a cryptosystem is the secrecy of the key(s). That is, the key is impossible to recover or to break by an unauthorized entity. However, users may require to recover the key when the key is lost somehow. Similarly, law enforcement agencies may also want to get the key of an ecrypted material. In this paper we will give an overview about the methods avaiable to resolve this „contradicton" flashing the mathematical background too.

In the second part of the paper we focus on a practical problem, and overview the correspondig technical and implementation tasks as well. We have developed a new, symmetric key based method to protect the master key of the key recovery system, which must appear unprotectedly in the RAM of the computer.

Alkalmazott Matematikai Lapok **23** (2006), 405–432.

ALAPVETŐ MATEMATIKAI TRANSZFORMÁCIÓK A KRIPTOGRÁFIÁBAN

TÓTH MIHÁLY

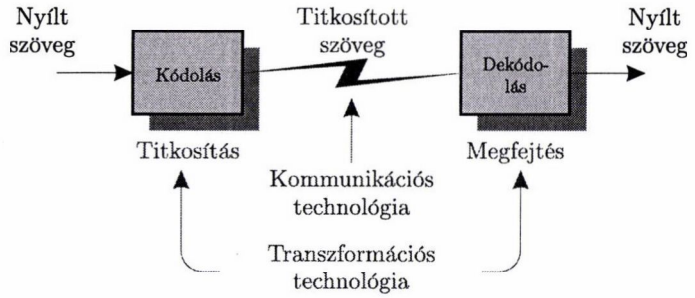
Budapest

Ez a cikk a kriptográfiai rendszerek ún. generációkba sorolásával, valamint a transzformációk általános jellemzőivel, tulajdonságaival foglalkozik. Heurisztikus módon megfogalmazza e transzformációk általános követelményeit és ezek alapján kísérletet tesz a kriptográfiai rendszerek formálnyelvi definíciójára. Tisztázza a kriptográfiában alkalmazott ún. blokk fogalmát és ennek segítségével a linearitásra is ad egy definiciót. Megfogalmaz egy sejtést, amely szükséges és elegendő feltétel a klasszikus helyettesítési és permutációs transzformációk felcserélhetőségére, valamint összevonására. Foglalkozik az egyszerű leképezőfüggvények inverz tulajdonságaival. Claude E. Shannon javasolta elsőként az ún. produkt transzformációkat, amelyek valójában vagy nem kommutatív produktumok, vagy összetett függvények. Minden esetre komoly szerepük van a modern, ún. negyedik generációs, szimmetrikus kriptorendszerekben. A cikk kitér az ún. Feistel transzformációkra és alkalmazásaikra, a többkomponensű transzformációk komponensei invertálhatóságának a kérdésére s végül az ún. aszimmetrikus (nyíltkulcsú) kriptorendszerekben alkalmazott transzformációk alapelveire.

Titkosítási rendszerek (kriptorendszerek) generációi

Jóllehet ma már nem szokás a számítástechnika (illetve a számítógépek) és alkalmazásaik újabb s újabb eredményeit egy-egy új „generációként" emlegetni, érdemes arra emlékezni, hogy korábban a számítógépek ún. generációit lényegében az alkalmazott technika határozta meg. A kriptorendszerek generációinál egészen biztosan nagy szerepe van az alkalmazott titkosítási/megfejtési módszereknek, technikának, közös néven az ún. *transzformációs módszereknek,* (matematikai terminológiával: leképezéseknek) de nagy szerepe van az üzenet továbbítási technikájának is, vagyis a *kommunikációs technológiának és a protokolloknak* is.

Eszerint a kriptogenerációkat két technológia együttesen határozza meg. Nevezetesen a transzformációs és a kommunikációs technológia. Valahogyan úgy, ahogyan ezt az 1. ábra szemlélteti.



1. ábra. Kriptorendszerek generációinak osztályozási szempontjai

GENERÁCIÓ	JELLEMZŐJE	TRANSZFORMÁCIÓS	KOMMUNIKÁCIÓS
		TECHNOLÓGIA	
1.	Monoalfabetikus	Helyettesítés Nagyon ritkán: keverés	– Nem jellemző
2.	Polialfabetikus és blokkos	Helyettesítés (kézi módszerrel) KULCSSZÓ	Nem jellemző
3.	Mint az előző, de *nagyon sok* ábécével	Betűnkénti helyettesítés (elektro) mechanikus (pl. rotoros) géppel	Rádió kommunikáció
4.	Produkt-transzformációk sok rundban. Igen nagy kulcstér. **Polialfabetikus** rendszerek.	Számítógéppel végzett iteratív transzformációk, amelyeknek nem létezik a nyers erő módszerénél gyorsabb, kulcs nélküli, algoritmusos módszere.	Kommunikáció a világhálón, magánhálózatokon, vagy legalább virtuális magánhálózatokon
5.	Aszimmetrikus kriptorendszerek. A kódolás és a dekódolás azonos leképzésekkel, de inverz kulcsokkal történik. **Monoalfabetikus** rendszerek	Számítógéppel, vagy célhardverrel megvalósított, számításigényes leképezések igen nagy (1000 bitnél nagyobb) kulcsokkal és ennek megfelelő kulcstérrel	Kommunikáció a világhálón, magánhálózatokon, vagy legalább virtuális magánhálózatokon

A transzformációk[1]

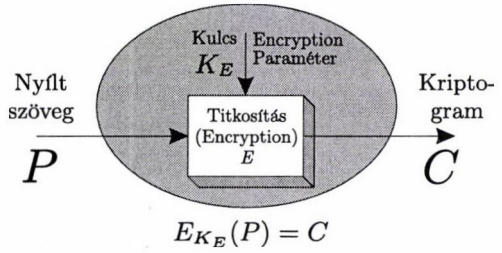
A titkosítás nélküli, ún. nyílt szöveget (P) valamilyen módon a be nem avatottak számára szándékoltan érthetetlen szimbólum-sorozattá, ún. kriptogrammá[2] (C) alakítja át a titkosítás.

[1] Szigorúbb matematikai értelemben leképezésekről van szó.

[2] A kriptogram aránylag új elnevezés. Korábban a francia eredetű *siffre* jelentette a titkosított szöveget és a sifrírozás a titkosítást.

A nyílt szövegnek is és a kriptogramnak is külön-külön ábécéje van, sőt a kriptogramnak több ábécéje is lehet, s a kriptogram ábécé meg is egyezhet a nyílt szöveg ábécéjével. Aszerint beszélünk egy– vagy többábécés kriptorendszerről, hogy hány ábécét használ a kriptogram.

A nyílt szöveget egy *transzformáció* képezi le a kriptogramba. Ezt teheti úgy is, hogy a nyílt szöveg minden egyes betűjét külön-külön transzformálja a kriptogram egyes betűivé, és teheti úgy is, hogy egy meghatározott hosszúságú betűcsoporton egyszerre hajt végre valamilyen transzformációt. Az előbbit *folyamatos titkosításnak* (Stream Ciphering) az utóbbit pedig *blokk titkosításnak* (Block Ciphering) nevezik.



2. ábra. A titkosító transzformáció

Ha a nyílt szöveg betűit egy-egy bájt helyettesíti, akkor nincs is lényeges különbség a betűnkénti és a blokkos titkosítás között, csak a blokkok mérete különböző.

Az ábécékből kiindulva két, alapvető transzformáció típust szokás megkülönböztetni, nevezetesen *a helyettesítést* és *a keverést*. Az előbbi mind a betűnkénti, mind blokkos titkosítás esetében alkalmazható, az utóbbi csak blokkokra. Számos érdekes példát ír le –többek között – *David Kahn* alapműve [15].

A transzformációk ezen archetípusai nem is olyan nagyon különbözőek, mint azt sokáig gondolták. Mindenesetre a bináris rendszerekben ezek a különbözőségek eléggé összemosódnak [24]. Feltétlenül meg kell említeni, hogy a kezdetektől fogva diszkrét ábécé vagy blokkok transzformációjáról volt szó. Jóval azelőtt, hogy a digitális, illetve diszkrét rendszerek olyan széles körben elterjedtek volna, mint manapság tapasztalhatjuk azt.

Kézenfekvő matematikai modellek voltak a diszkrét algebra, illetve matematika egyes fejezetei, ide értve a számelméletet is. A titkosítás végül is ilyen diszkrét elemek *leképezését* jelenti, s ez az, amit alább részletezünk is.

Itt kell megemlíteni azt is, hogy a titkosító rendszerek igen korai feltalálói között is van, aki már a XVI. század vége felé tudatosan alkalmazott ilyen matematikai modellt.

Blaise De Vigenere-ről [1523–1596], a „Látnokról" van szó, aki lényegében a Caesar-féle monoalfabetikus titkosítást fejlesztette tovább és az általa alkalmazott matematikai modell a modulo n összeadás, illetve kivonás volt, ahol n az ábécé betűinek a száma volt: [15], [21] és [22].

A leképező függvények is diszkrét véges elemű, vagy legalább is megszámlálhatóan végtelen sok elemű halmazok felett értelmezettek, és soha nem merült fel, hogy a leképező függvények esetleg folytonos függvények is lehetnének.

Nem merült fel az sem, hogy a transzformációkra formálnyelvi, illetve absztrakt algebrai modelleket állítsanak fel, bár elvileg ezek is célravezetők lehetnének (diszkrét esetekben is).

A formális nyelvek és az absztrakt automaták modelljeinek alkalmazhatósága más kérdés. Ezek ugyanis olyan hosszúságtartó (vagy pl. a GSM és a Turing–gép esetében nem hosszúságtartó) leképezéseket hajtanak végre, amelyek nagyon is alkalmazhatók lennének a kriptográfiai leképezések modellezésére. Egy indirekt kivételtől eltekintve azonban nem találkoztam ilyen jellegű publikációkkal.

A kivétel *Arto Salomaa*, akinek a fő kutatási területe éppen a formális nyelvekkel és automatákkal foglalkozik [26], de az utóbbi években a matematikai bonyolultság elmélettel (és a kiszámíthatósággal) is kapcsolatba hozta ezt a kutatási területet [27], sőt kifejezetten az aszimmetrikus kriptorendszerekkel kapcsolatban is publikált [28].

Alább még visszatérek erre a kérdésre.

A pontosabb formális leírásnak azonban több akadálya is van. Itt talán elegendő csak annyit megemlíteni, hogy ahányféle titkosító leképzés van, annyiféle leképzési szabály, ezért általános képzési szabályokat nagyon nehéz adni. Vannak azonban emellett más problémák is.

A kriptográfiai leképezések alapkövetelményei

A kriptográfiai transzformációk alkalmas matematikai modelljei tehát a *leképezések*.

Heurisztikus módon kikövetkeztethető, hogy a kriptográfiai transzformációknak milyen feltételeket kell kielégíteniük. Ezek a következők:

a. Mivel mind a nyíltszöveg ábécé, mind a kriptogram ábécé véges halmazok[3], ezért a leképező függvény véges halmazt képez le véges halmazra.
b. A leképező függvény egyértékű és
c. ha egyértelműen visszafejthetőnek kell lennie (márpedig annak kell lennie), akkor léteznie kell a leképezés inverzének is, továbbá
d. az inverz leképezésnek is egyértékűnek kell lennie,
e. a c. és a d. feltételekből pedig az következik, hogy a titkosítási leképezés kölcsönösen egyértelmű (bijektív) kell, hogy legyen.[4]

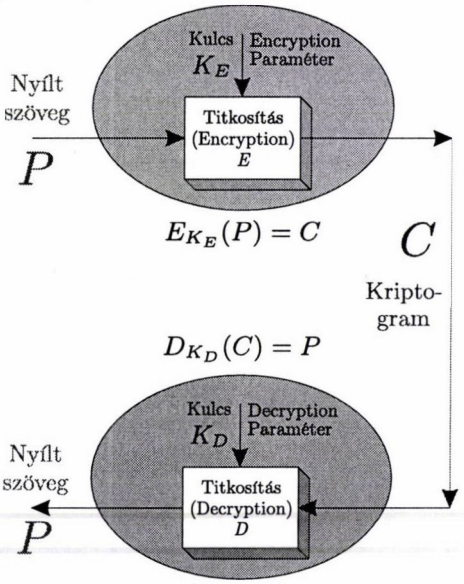
A felsorolt követelmények nagyon logikusnak tűnnek, de ezek alól a szabályok alól is van kivétel. A modern kriptográfiában ugyanis találhatók ún. valószínűségi

[3] ... és az ezekből az elemekből alkotott véges hosszúságú blokkok (*string*) száma is véges

[4] Létezik olyan kriptográfiai transzformáció is, amely a nyílt ábécé elemeit a képtartomány egy-egy valódi részhalmazára képezi le, de ezek a részhalmazok diszjunktak és a leképző függvényre ilyen kiterjesztéssel is érvényesek a mondott feltételek.

kriptorendszerek is, amelyeknél a leképezésekhez valószínűségek tartoznak, s véges (bár nagyon kicsi) valószínűséggel előfordulhat az is, hogy egy titkosított üzenet nem fejthető vissza. Némely alkalmazásnál (pl. titkosított beszéd-átvitel), amely redundáns információt továbbít, ez az információvesztés nem okoz gondot.

A felsorolt öt feltétel tehát szükséges, de még nem elégséges feltétel. Ezért a leképezések tulajdonságaira alább még visszatérünk.



3. ábra. A titkosítás teljes folyamata

A módszer és a kulcs

Vannak még a gyakorlati alkalmazhatóságból következő feltételek is. A leképezés bonyolultságát, a hozzá szükséges számítási kapacitást (és/vagy idő-igényt), valamint az algoritmus gyorsaságát már említettük.

Régi tapasztalati tény, hogy ha sokszor használják ugyanazt a titkosító leképezést, akkor azt majdnem biztosan feltörik.[5] Ezért aztán időről-időre változtatni kell azt. Nagyon nehéz azonban magát a módszert változtatni, mert nem csak arról van szó, hogy egyre újabb s újabb módszert kell kitalálni, hanem arról is, hogy ha a módszer alkalmazásához már gépet is szerkesztettek, akkor minden egyes módszerváltáskor a titkosító/megfejtő gép helyett is újat kell készíteni. Ezért már a XVII.

[5] Tulajdonképpen ilyen feltöréseknek tekinthetők a rég elfeledett ókori írások megfejtése, pedig több ilyen esetben maga az írás nyelve sem volt ismert. Minden ilyen sikeres megfejtés alapvető feltétele volt azonban, hogy *sok* írott szöveg állt az elemzők rendelkezésére.

században felmerült és a XIX. században *expressis verbis* meg is fogalmazták, hogy egy titkosítási transzformáció erőssége[6] végső soron az abban alkalmazható kulcsok számától függ: minél több a lehetséges kulcsok száma, azaz minél nagyobb a *kulcstér*, annál nehezebb vagy reménytelenebb a kulcs ismerete nélkül hozzájutni a titkosított információhoz.[7]

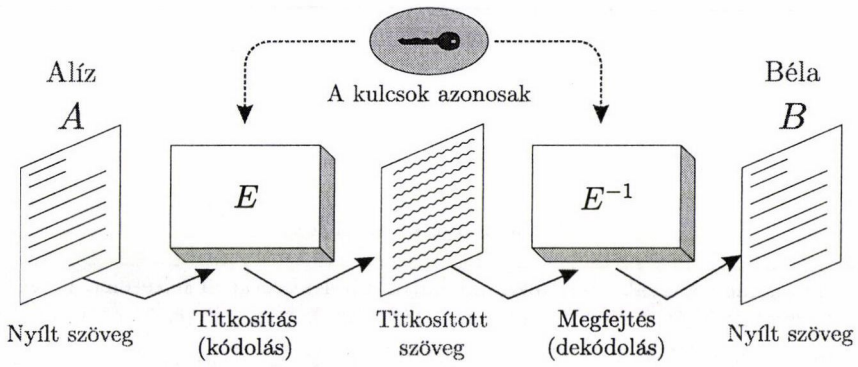
A titkosító transzformációk közös és általános tulajdonsága, hogy a leképző függvénynek van legalább egy *paramétere*, amely ismerete nélkül a leképzés nem valósítható meg. Ugyanez igaz a transzformáció inverzére, azaz a titkosított információ visszafejtésére is. Ez a paraméter a titkosítás *kulcsa*. Ha a kulcstér gyakorlatilag igen nagy, akkor maga a leképezés akár közzé is tehető, mert a kulcs önmagában garantálja a titkosítás biztonságát. Ez, nevezetesen az extrém nagy kulcstér, valamennyi mai kriptorendszer alapvető és közös elve.

A szimmetrikus és az aszimmetrikus kriptorendszerek összevetése

Valamennyi tradicionális kriptorendszer ugyanazt a kulcsot alkalmazta mind a titkosításhoz, mind a visszafejtéshez, pedig mint ma már tudjuk, ennek nem kell feltétlenül így lennie.

Vizsgáljuk meg ehhez a teljes titkosítás–visszafejtés folyamatot!

Az E_{K_E} és a D_{K_D} leképező függvényeknek *a paramétereikkel együtt* kell egymás inverzeinek lenniük.



4. ábra. Szimmetrikus (tradícionális) kriptorendszer blokkvázlata

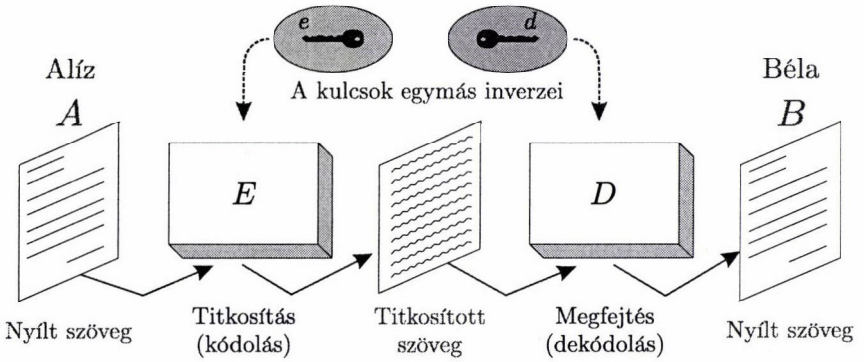
[6] Pontosabb elemzéssel alább megmutatjuk, hogy a kriptogram ábécé (vagy ábécék), mint halmazok rangja alapvetően meghatározó a kriptorendszer feltörése szempontjából és a lehetséges kulcsok számát is az ábécé(k) számossága korlátozza.

Egy kriptorendszer erősségét a feltörésének a nehézsége határozza meg. Ma már nem tekintik erősnek az olyan titkosításokat, amelyek algoritmusos módon feltörhetők. Az erős kriptorendszerek csakis úgy törhetők fel, hogy valamennyi lehetséges kulcsot végig kell próbálgatni. Ezt a *nyers erő* módszerének nevezik.

[7] Alább majd megmutatjuk, hogy elsődlegesen a kriptogram ábécék elemszáma a meghatározó. A kulcstér „mindössze" illeszkedik ehhez.

A tradicionális kriptorendszereknél magától értetődőnek tekintették, hogy a K_E és a K_D kulcsok *azonosak,* és az imígyen azonos paraméterekkel rendelkező E és D leképező függvények egymás inverzei.

Az ilyen rendszereket ma *szimmetrikus kriptorendszereknek* nevezzük és kizárólagos alkalmazásuk az ősidőktől kezdve az 1970-es évek közepéig tartott, de a mai legmodernebb kriptográfiában is megvan a jelentőségük (mármint a modern változataiknak).



5. ábra. Az aszimmetrikus kriptorendszer blokkvázlata

A transzformációk (azaz leképezések) egymás inverzei, *a kulcsok* pedig *azonosak.*

Tulajdonképpen kézenfekvő megoldás az is, hogy nem a leképző függvények egymás inverzei, hanem a titkosító és a visszafejtő kulcsok vannak valamilyen műveletre nézve inverz viszonyban egymással. Erre az ötletre azonban Whitfield Diffie előtt (1976) senki sem jött rá.

Ezekkel a rendszerekkel majd alább, az *aszimmetrikus kriptorendszerek* kapcsán foglalkozunk. Egy szimmetrikus kriptorendszerben tehát egyetlen kulcsot alkalmaznak, amelyet természetesen titokban kell tartani. Mondhatjuk úgy is, hogy ettől függ a rendszer biztonsága.

A titkosító (E) és a visszafejtő (D) algoritmusok, illetve függvények egymás inverzei, de általános esetben nem azonosak. Kérdés, hogy lehet-e olyan, elegendően biztonságos, titkosító rendszert szerkeszteni, amelyben nem csak a kulcsok, hanem ez a két függvény is azonos.

Nos, lehet, bár a modern, ún. erős kriptorendszerek esetében csak valami hasonló, de nem egészen azonos dolgot.

Mivel az inverz algebrai fogalma mindig csakis egy adott műveletre vonatkozik, az aszimmetrikus rendszerekben az E leképző függvény az, amelyre nézve az e és a d kulcsok egymás inverzei.

Ezt a két ábrát éppen azért tüntettük fel így, egymás alatt, hogy felhívjuk a figyelmet a szimmetrikus és az aszimmetrikus kriptorendszerek – jobb szó híján – szimmetriájára.

A tradicionális kriptorendszerek körében azonban megvalósítható, hogy egy titkosító-megfejtő függvénypár két tagja azonos legyen, vagyis egy olyan leképző függvényt találjunk, amely saját maga inverze. Elvileg az aszimmetrikus kriptorendszerek esetében is szerkeszthető olyan, amelyben a „két" kulcs azonos és saját maga inverze, de ez határeset, amikor csakis a leképezőfüggvény jellegéből állapítható meg, hogy a kriptorendszer szimmetrikus vagy aszimmetrikus-e. Az ilyen rendszernek azért nincs is gyakorlati jelentősége, mert az aszimmetrikus rendszerek transzformációinak a számítási kapacitás-igénye nagyságrenddel nagyobb, mint az azonos kriptográfiai erősségű szimmetrikus rendszereké, és értelmetlenül gazdaságtalan lenne sokkal nagyobb költséggel megvalósítani egy szimmetrikus funkciójú rendszert, mint azt a szimmetrikus algoritmusokkal lehet.

A transzformációk tulajdonságai

Leképezések vagy függvények?

Az eddigiekben utaltunk ugyan arra, hogy a *transzformáció* szigorúan vett matematikai értelemben nem azonos a kriptográfiában alkalmazott *leképezésekkel*, de ezt az állítást nem vizsgáltuk. Nos ennek itt a helye.

A kriptográfiai függvények (nyílt) ábécéből (titkos) ábécébe képező transzformációk. Matematikai értelemben azonban transzformációnak olyan *leképezést* nevezünk, amely egy halmazt saját magába képez le. Általános esetben a kriptográfiai leképezésektől nem követeljük meg, hogy értelmezési tartományuk és értékkészletük megegyezzen; de még azt sem, hogy akár csak tartalmazzák egymást.

Másrészt nem árt tisztázni, hogy tulajdonképpen mi is ezeknek a kriptográfiai leképezéseknek az értelmezési tartománya és az értékkészlete.

(Az itt következő diszkusszió során megkísérlem a kriptográfiai leképezések egy – nagyon egyszerű – formálnyelvi megközelítését.)

A kriptográfiai leképezések egy véges szimbólumhalmazból (nyílt ábécé) alkotott véges sorozatok halmazából képeznek egy vagy több véges szimbólumhalmaz (titkos ábécék) uniójából képzett sorozatok halmazára.

Formálisan: egy φ leképzés kriptográfiai transzformációnak tekinthető, ha $\varphi : P^* \to C^*$ alakú,

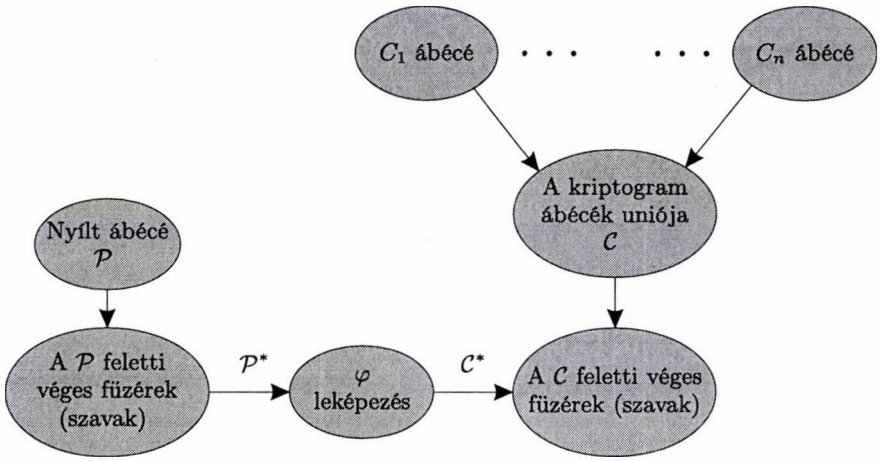
ahol P – input szimbólumok halmaza, nyílt ábécé,

P^* – P elemeiből alkotott füzérek halmaza

$C = \bigcup_{i=1}^{n} C_i$, output szimbólumok halmazainak (kriptogram ábécék) egyesítése,

ahol C^* – C elemeiből alkotott sorozatok halmaza.

A $C = \bigcup_{i=1}^{n} C_i$, definícióban szereplő n érték a kriptogram ábécék számát jelenti. A n=1 esetben *monoalfabetikus* titkosírásokról beszélünk Ismert $n > 1$ esetén és *első fajú polialfabetikus* titkosírásoknak nevezhetjük őket, míg abban az esetben, ha az ábécék száma – kívülállók számára – nem ismert, akkor nevezhetjük



6. ábra. Kriptográfiai leképzések
Polialfabetikus (kriptogram) ábécék

azokat *másodfajú polialfabetikus* titkosírásoknak.[8] Az első- és a másodfajú polialfabetikus rendszerek közti lényegi megkülönböztetést indokolja, hogy az első fajú többábécés rejtjelezések megfejthetők[9] betűgyakorisági analízis segítségével, míg a másodfajú polialfabetikus titkosírások nem. A de Vigenere kód esetében Babbage zsenialitása éppen abban nyilvánult meg, hogy módszert talált arra, hogy hogyan alakítson át egy másodfajú polialfabetikus rendszert első fajú rendszerré, azaz hogyan határozza meg az ábécék számát vagy – ezzel lényegében azonos jelentéssel – az alkalmazott blokk hosszát.

Ez a módszer – nevezetesen a dupletek ismétlődési távolságának a megkeresése és e távolságok legnagyobb közös osztójának a meghatározása – azonban távol áll a direkt megfejtésre való betűgyakoriság analízistől. Jó arra, hogy visszavezesse a megoldási problémát az első fajú polialfabetikus rendszerekére, de máskülönben nem része a megoldásnak. (Erre utaltunk a 6. lábjegyzetben is.)

Babbage előtt több mint 300 évig feltörhetetlennek tartották de Vigenere titkosítási módszerét, és az is volt a Babbage-féle visszavezetés ismerete híján. A sors és iróniája, valamint a brit katonai bürokrácia vaskalapossága, hogy Babbage-nak nem engedték meg a visszavezetési módszere publikálását. (Ez aztán még kétszer megismétlődött: Turing Enigma feltörése és a Colossus megépítése, valamint a GCHQ nyíltkulcsú kriptorendszerének Diffie előtti felfedezése esetében is.)

A formális megközelítésből látszik, hogy a leképezések értelmezési tartománya és értékkészlete is *véges* vagy megszámlálhatóan végtelen halmaz, hiszen véges hal-

[8] Ezek itt bevezetett, a szakirodalomban nem ismert elnevezések.

[9] Ti. az egyszerű helyettesítéses rendszerekhez hasonlóan, amelyek archetipusa pl. a Caesar-féle titkosítás.

mazok elemeiből alkotott füzérek (*string*) halmazai[10]. A leképző függvények tehát legfeljebb megszámlálhatóan végtelen halmazok felett vannak értelmezve, „folytonos" halmazok feletti függvények értelemszerűen szóba sem jöhetnek.

A formálnyelvi leírás egyik alapproblémája már e ponton jelentkezik és azzal kapcsolatos, hogy mit tekintünk ábécének.

Itt – egyelőre – azoknak a szimbólumoknak a P halmazát, amelyből a nyílt üzenet szavait képezzük és értelemszerűen kiterjesztjük ezt a C kriptogram ábécé(k)re is. Ezek nyílvánvalóan véges ábécék.

Blokkrendszerű titkosítások esetén az ábécékből alkotott szavak hossza nem csak korlátozott, hanem egy megadott véges érték (az AES-nél pl. 128 bit). Ezekből ugyancsak végesszámú van (az AES esetében éppen 2^{128} db), és e halmaz véges volta miatt ezek is tekinthetők egy véges ábécé elemeinek.[11] Mivel minden gyakorlati esetben megegyezik a kriptogram és a nyílt szöveg blokkjainak a hosszúsága, (sőt ugyanez szokott lenni a kulcshossz is) ezért a mondottak a kriptogram ábécé(k)re is érvényesek. A leképezések vizsgálatakor egyáltalán nem mindegy, hogy az itt említett kétféle ábécé-értelmezés közül melyiket fogadjuk el. A 6. ábrán szemléletesen bemutatott halmazok és jelölésrendszer remélhetően világossá teszi az értelmezést. Az ábra a polialfabetikus rendszerek kedvéért mind az ábécéket, mind azok elemeiből alkotott szavak halmazait feltünteti.

A leképezések linearitása

A kriptográfiai leképezések általános vizsgálatakor a $P*$ nyíltszöveg füzérek közötti műveleteket és ezeknek a $C*$ képtartományban való megjelenését vizsgáljuk. Formálnyelvi megközelítésben tehát ekkor éppen a korábban említett második ábécé-értelmezést alkalmazzuk, vagyis azt, amikor a leképezés értelmezési tartományának (és értékkészletének is) az elemei a $P*$, illetve a $C*$ füzérek. Emlékeztetünk arra, hogy e diszkusszió során nem tekintjük e füzérek hosszát sem előre definiáltnak, sem azonosnak, legfeljebb felülről korlátosnak, hogy megszámlálható halmazokkal tudjunk számolni.

Hagyományosan egy leképzést akkor szokás lineárisnak nevezni, ha az értelmezési tartomány elemein értelmezett valamely műveletekre nézve műveletтartó, azaz mindegy, hogy a műveletet előbb elvégezzük az értelmezési tartomány elemei közt, majd az eredményt képezzük le, vagy előbb a leképzést az operandusokra alkalmazzuk, és a képelemek közt végezzük el a megfelelő[12] műveletet.

[10] Vagy véges sok véges halmaz uniójából alkotott füzérek halmaza

[11] Igaz, hogy ennél az ábécénél igen nagy az elemek száma. Ha azonban nem így lenne, akkor – az alkalmazott transzformáció bonyolultságától függetlenül – az elemek gyakoriságának az elemzésével tulajdonképpen nagyon könnyen feltörhető lenne a kriptogram.

[12] Hangsúlyozni kell a „megfelelő" szó jelentőségét a megfogalmazásban. Általános esetben ugyanis nem várható el, hogy a leképzés értelmezési tartománya és értékkészlete megegyezzen. Így a leképzés előtti művelet egy az értelmezési tartományon értelmezett művelet, míg a képelemeken végzett művelet az értékkészlet felett értelmezett művelet.

Formálisan: egy $\varphi : A \to B$ leképezés lineáris valamely $+$ és $\oplus$ műveletek szempontjából, ha

$$\forall a, b \in A : \varphi(a + b) = \varphi(a) \oplus \varphi(b),$$

ahol $+ : A \times A \to A$ és $\oplus : B \times B \to B$ megfelelő A és B feletti műveletek[13].

Kriptográfiai leképezések esetében a leképezés értelmezési tartománya egy adott szimbólumhalmaz elemeiből alkotott füzérek[14] halmaza. Hasonlóképpen az értékkészlet is füzérek halmaza, csak a füzérek halmaza nem ugyanaz, mint az értelmezési tartomány esetében. Sőt a füzéreket alkotó ábécék sem feltétlenül azonosak a nyílt P ábécé és a C kriptogram ábécé esetében.

Ilyen halmazokon természetes módon definiálhatjuk a füzérek összefűzésének műveletét (catenation, katenáció, konkatenáció,). Szokásos jelölése a $\|$ jel, de használatos még a $\hat{}$ jelölés is. Ezek szerint $a \| b$, illetve $a\hat{}b$ olyan szimbólum sorozatot jelöl, amelynek első tagjai az a füzér elemeivel egyeznek meg, a továbbiak pedig a b sorozat megfelelő elemeivel.

Kriptográfiai leképzések linearitásáról tehát az összefűzés művelet tekintetében beszélhetünk. Ezek szerint lineárisnak nevezünk egy $\varphi : A^* \to B^*$ kriptográfiai transzformációt, ha

$$\forall a, b \in A^* : \varphi(a \| b) = \varphi(a) \| \varphi(b).$$

A legtöbb kriptográfiai leképzés nyilván nem lineáris a fenti definíció szerint. Ez ugyanis azt jelentené, hogy a teljes titkos szöveg a nyílt szöveg külön-külön részeinek titkosításával áll elő, amelyeket a nyílt szövegnek megfelelő sorrendben fűzünk össze. Ez viszont lehetőséget nyújtana a támadónak arra, hogy ha sikerül az üzenet küldőjét rávennie, hogy ugyanazt az üzenetet jól körülhatárolható helyen történő változtatással küldje el, akkor a titkos szövegben történt változásból következtetni lehessen a kódolás miként jére, sőt magára a kulcsra is (*choosen plain text attack*).

Blokkhosszúság[15]

A következőkben megmutatjuk, hogy a blokkhosszúság fogalma a linearitás segítségével sokkal pontosabban meghatározható, mint az a mai, modern kriptorendszerekben szokásos. Ezeknél ugyanis blokknak nevezzük azt a szimbólumfüzért, amelyen a kriptotranszformációt végrehajtjuk. A vonatkozó szakirodalom többnyire megemlíti ugyan, hogy ezek a transzformációk nemlineárisak, de eleddig

[13] Vektorterek (lineáris terek) esetében nem csak az alaphalmazon belül értelmezett művelet művelettartását követelik meg, de az ún. skalárral való szorzás műveletre is előírják ugyanezt. Kriptográfiai leképzések esetén ilyen külső műveletet nem definiálunk, így ennek vizsgálatától eltekinthetünk.

[14] Algebrai értelemben itt szimbólum-sorozatokról van szó. Mégis, a gyakorlatban elterjedt szóhasználat miatt ezeket inkább füzéreknek (string) nevezzük vagy – a formálnyelvi megközelítésnél – az adott nyelv szavainak.

[15] A blokkhosszúság e definíciója új, eleddig ezt a megközelítést – tudtommal – nem használta a szakirodalom. Azért vezettük be, mert a leképezések linearitása, illetve nemlinearitása szempontjából hasznos.

sehol sem bukkantam rá a linearitás, illetve a nemlinearitás pontosabb definíciójára. Más szóval arra, hogy az említett kriptotranszformációk *miért,* illetve *mitől* nemlineárisak.

Technológiai szempontból megoldhatatlan, hogy egy kriptográfiai leképzés tetszőlegesen hoszszú szövegekre is nemlineáris módon viselkedjék. Minden gyakorlatban használt algoritmus esetében van egy a sorozatok hosszára vonatkozó felső korlát, hogy ha az ilyen hosszúságú sorozatokat tekintjük az ábécé elemeinek, akkor az ezekből alkotott sorozatok halmazán az összefűzés műveletére vonatkozóan a leképzés már lineáris. *Azt a legkisebb ilyen értéket, amelyre az adott leképzés lineárisan viselkedik, a leképzés* blokkhosszúságának *nevezzük.*

Vegyük észre, hogy itt nem a hagyományos értelemben vett „blokkonkénti" leképezés linearitásáról vagy nemlinearitásáról van szó, hanem épp fordítva:

A kriptográfiai leképzés linearitásának előbbi definíciója segítségével lehet a blokk fogalmát és a blokkhosszúságot definiálni.[16]

Nagyon leegyszerűsítve: Blokkok azok a legrövidebb füzérek, amelyekre, mint argumentumokra a kriptográfiai transzformációk linearitása teljesül.[17]

Formálisan: Legyen $\varphi : A^* \to B^*$ kriptográfiai leképzés, n természetes szám, továbbá $\varphi' : (A^n)^* \to B^*$ leképzés úgy, hogy $\forall s \in (A^n)^*$ sorozat esetén $\varphi(s) = \varphi'(s)$.

Ekkor azt a legkisebb n természetes számot, amelyre φ' lineáris, a φ leképezés *blokkhosszának* nevezzük.

Ehhez érdemes némi magyarázatot fűzni: φ az A ábécé elemeiből alkotott tetszőleges hosszúságú sorozatokon értelmezett függvény. A^n az A ábécé elemeiből alkotott pontosan n-hosszúságú sorozatok (blokkok) halmaza, az ezekből mint elemekből alkotott sorozatok alkotják $(A^n)^*$ elemeit. Az ezen a halmazon értelmezett φ' leképezés csak annyiban tér el a φ leképezéstől, hogy nincs feltétlenül bármilyen hosszúságú sorozaton értelmezve, csak azokon, amelyek hosszúsága a talált n szám egész számú többszöröse. Akkor mondjuk, hogy ez az n a leképzésre jellemző ún. blokkhossz, ha a leképzés a blokkokra megszorítva lineáris, és nincs olyan n-nél kisebb korlát, amelyre a leképzés ugyanezt tudná.

A mai, gyakorlatban alkalmazott kriptorendszerek mind meghatározott hosszúságú blokkokra osztják a nyílt szöveget és e blokkokat külön-külön transzformálják

[16] Ez a fajta linearitás nem is ad választ arra, hogy *mennyire* nemlineáris egy ilyen blokk leképezése.

Ez a kérdés pedig a gyakorlatban létezik. A DES esetében – a NIST javaslatára – éppen a „nemlinearitás növelésére" bevezették a rund transzformációk sorába az expanzió-kompreszszió párt. Közvetve az ún. lavinahatás növelése és ez által a feltörés megnehezítése miatt. Ez a fajta művelet azonban a későbbi szimmetrikus rendszerekben nincs meg, tehát a mondott okból nem is volt rá szükség. Ennek az okát viszont – tudtommal – nem publikálták, hanem „csak tudomásul vették", hogy a Feistel függvények szerinti iteratív transzformációk az expanzó-kompresszió pár nélkül is rendelkeznek a lavinahatással.

[17] Ennek a definíciónak az ötlete és a formális megfogalmazása Tóth Gergelytől, a Veszprémi Egyetem székesfehérvári AIFSz Képző Központjának a tanárától származik.

kriptogram blokkokká. Mind a nyílt szöveg, mind a kriptogram blokkjai ugyanolyan hosszúságúak s ezzel azonos a kulcs hosszúsága is.[18]

Nyilvánvaló, hogy

a. A blokk nem osztható, azaz nem alkalmazható a titkosítási leképezés részblokkokra. Ha a teljes titkosítandó szöveg blokkokra bontása végén részblokk adódna, akkor azt valamilyen előzetes megegyezés szerint fel kell tölteni (padding).
b. Az egymást követő blokkok titkosítási leképezése egymás után és egymástól függetlenül történik. Ilyen értelemben a blokkokból alkotott sorozat részeire (ti. az egyes blokkjaira) érvényes az előbbiekben megfogalmazott linearitás-definició.

Azonban ez alól is van kivétel, nevezetesen a szimmetrikus rendszerekben alkalmazott láncolás elve, amelyet ugyan a DES kapcsán vezettek be, de bármelyik „blokkos" titkosításnál alkalmazható [5].

Az utóbb említett ellenpélda szemlélteti, hogy láncolt titkosításnál a „blokk" fogalmát pontatlanul alkalmazzák az egymás után végrehajtott transzformációkra, amelyek ekkor egyáltalán nem függetlenek egymástól. Ilyen esetekben tulajdonképpen az egész transzformálandó nyílt szöveget kellene egyetlen oszthatatlan „blokknak" tekinteni.

A tükörszimmetrikus, öninvertáló transzformációk

E leképzés-típusoknak a szimmetrikus kriptorendszerek körében van jelentősége. Gyakorlati szempontból nagyon is jelentős dolog, hogy ha ugyanazzal a géppel (vagy algoritmussal) tudunk titkosítani és visszafejteni, a gép mindennemű átállítása nélkül. Így működött pl. a második világháborúban elhíresült német titkosító gép, az Enigma is, amelyet a maga idején megfejthetetlen kódolónak tartottak. Nos, végül is nem volt megfejthetetlen, de a feltöréséhez zsenik kellettek és nagyon időigényes feladatnak bizonyult.

Az Enigma egy érdekes tanulsága az, hogy a megfejtést az ábécék extrém nagy száma tette szinte lehetetlenné. Ez, ti. az ábécék száma még a legegyszerűbb, kereskedelmi változatnál is 26^3 volt, de a katonai változatoknál ezt kb. ezerszeresére növelték. Ráadásul a kulcstér is nagyon nagy (de az ábécék számától elvileg független) volt.

A titkosító és a visszafejtő algoritmus hasonló „tükörszimmetriája" fellelhető a modern iterációs kriptorendszerekben is – ha eltekintünk a kulcsütemezés megfordításától. Az Enigma a mindaddig legelterjedtebb titkosítási transzformációt alkalmazta, ti. a *helyettesítést.* A 26 betűs ábécéből képzett nyílt szöveg minden egyes betűjét egy kriptogram betűbe képezte le. A bonyolultságát azzal érték el, hogy egy meglehetősen hosszú betűsorozaton belül ugyanazt a nyíltszöveg betűt mindig más és más kriptogram betűnek feleltette meg.

[18] Egyetlen egy, részbeni kivétel a DES ún. rund transzformációján belül alkalmazott expanzió, majd a szubkulcs-művelet utáni kompresszió.

Itt most nem foglakozunk azzal, hogy ezt hogyan valósította meg az Enigma. Az egyik lényeges következmény az, hogy az ilyen tükörszimmetrikus és polialfabetikus helyettesítés messze nem használja ki a lehetséges kriptogram ábécék mindegyikét. Ezt azonban nagyon nagyszámú ábécé esetén „megengedheti magának" a kriptorendszer. Ezt a tükörszimmetriát megörökölték az Enigmától a modern iterációs kriptorendszerek is (DES, IDEA, AES...), de a visszafejtésnél alkalmazott fordított szubkulcs-sorozattal kiküszöbölték a tükörszimmetriának az Enigmánál még meglévő hátrányát.

A tükörszimmetrikus kriptotranszformációkat leképezéseknek tekintve teljesülnie kell annak a feltételnek is, hogy *a leképezés értéktartománya és értékkészlete azonos.* Más szóval a nyílt ábécé azonos a kriptogram ábécével – vagy ábécékkel, polialfabetikus rendszerek esetén[19].

Ekkor, és csakis ekkor lehetséges, hogy *a leképezés önmaga inverze.* Mint említettük, ennek gyakorlati szempontból van jelentősége, jóllehet csökkenti az aktuális kriptorendszer erősségét. Az Enigma esetében pl. egyfajta „fogódzót" jelentett a kódfejtőnek az, hogy az Enigma sohasem képzett le egy betűt önmagába. Ez a gép leképzési tükörszimmetriájának a következménye volt.[20]

A tükörszimmetrikus leképezést még a legegyszerűbb helyettesítő titkosításoknál is meg lehet valósítani. Egy sztenderd, n betűs ábécé és Caesar-féle titkosítás esetén azonban ilyenkor a kulcstér a felére csökken. Az n betűs kevert ábécék esetén pedig $(n-1)!$ helyett $(n/2)!$-ra csökken a kulcstér.

Az alap-transzformációk

Két alapvető transzformáció-típus létezik ősidők óta. Ezek a következők:

- helyettesítés (substitution),
- transzpozíció (más néven keverés vagy permutáció).

A helyettesítés alapértelmezésben betűt helyettesít betűvel, a transzpozíció pedig egy meghatározott hosszúságú szövegblokkon belül áthelyezi, összekeveri a betűket.[21]

Érdemes megjegyezni, hogy e két alaptranszformációnak nem csak történeti érdekessége van, hanem fellelhetők a legmodernebb titkosító rendszerekben is. Igaz, nem egymagukban, hanem összetett, beágyazott alkalmazásaikban.

Végül is mindkét alaptranszformáció megfogalmazható függvényként is, amelyek bijektív leképezést hajtanak végre.[22]

---

[19] Több-ábécés rendszerek és bonyolultabb – pl. iterációs – transzformációk esetében bonyolultabb feltételek is megfogalmazhatók. Az iterációs, szimmetrikus kriptorendszereknél pl. csakis úgy teljesül ez a „tükörszimmetria", hogy az inverz leképezésnél fordított sorrendben kell alkalmazni az egyes menetek (rundok) szubkulcsait, mint a titkosításnál. Ez más szóval az alkalmazott ábécék sorrendjének a megfordítását (is) jelenti.

[20] Valamint annak, hogy a valamennyi kriptogram ábécé páros számú betűből (26 betűből) állt.

[21] A kettő közötti határ egyáltalán nem olyan éles, mint az korábban látszott.

A „betűnkénti helyettesítés" pl. ASCII kód esetén bináris *blokkot* helyettesít blokkal. Ez a folyamatos (stream) titkosítás és a blokkos titkosítás határait mossa össze.

[22] A véges elemű nyílt ábécé az elemek számának összes permutációja szerint keverhető össze, tehát ennyi féle keverési kulcs létezhet. Ha ezeket a permutációkat megszámozzuk, akkor az i-edik

Ha a nyílt szöveget és a kriptogramot egy-egy ábécé felett értelmezzük, akkor ezeknek a transzformációknak az értelmezése triviális. Az is nyilvánvaló, hogy egymástól függetlenek, tehát akár fel is cserélhetők. A transzformációk felcserélhetősége másképp fogalmazva sorrend-függetlenséget is jelent s a titkosításnál, illetve a visszafejtésnél nagyon is jelentős lehet.

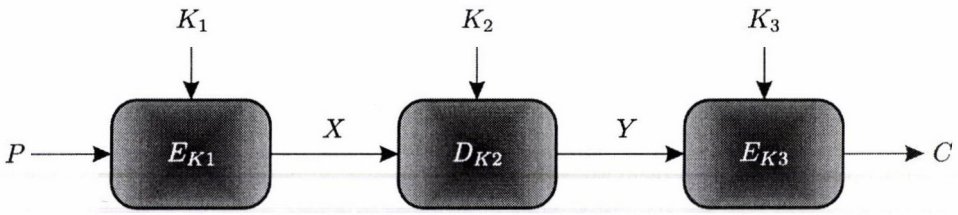
Nem nyilvánvaló a helyettesítés és a transzpozíció felcserélhetősége, ha a betűk helyett bináris blokkokban gondolkodunk.

Például minden egyes betűt egy-egy 8 bites blokk (byte) helyettesít. Ebben az esetben csak akkor sorrend-független a transzformációk, ha az egyik blokkhosszúsága egész számú többszöröse a másik transzformáció blokkhosszúságának [1].

Ez szigorúbb feltétel annál, amit korábban a linearitással kapcsolatban megadtunk, de nincs azzal ellentmondásban.

A transzformációk egyesítése

Vizsgáljuk meg a következő ábrát, amely a megfejtés megnehezítése céljából három transzformációt hajt végre egymás után rendre a K_1, K_2 és K_3 kulcsokkal.



$$C = E_{K3}\{D_{K2}[E_{K1}(P)]\}$$

7. ábra. Több egymás utáni transzformáció

Vegyük észre, hogy ha a K_1, K_2 és K_3 kulcsok azonosak, akkor ez a három transzformáció egyetlen egykulcsos transzformációval helyettesíthető. A komponens transzformációk számának növelése bonyolítja ugyan a titkosítási transzformációt, de a nyers erő módszerével szemben nem nyújt nagyobb védelmet, mint egyetlen transzformáció. Másképpen fogalmazva: ha kitalálják a kulcsot, akkor annak ismételt alkalmazása sem jelent komoly védelmet a feltöréssel szemben.

Az itt bemutatott összetett transzformáció szó szoros értelmében nem kommutatív produkt transzformáció, mert a komponensei nem felcserélhetőek. A leképezések felcserélhetősége viszont fontos kérdés, mint korábban már megmutattam.

Itt célszerű megjegyezni, hogy a „dupla DES„ titkosítást nem használják, mert találtak olyan feltörési módszert, amely a fent ábrázolt feltörési láncot mindkét végéről egyszerre támadja a nyers erő módszerével és kimutatható, hogy a krip-

permutáció egy i sorszámú leképezésként is felfogható. Nincs tehát éles elvi határ a helyettesítő és a permutációs leképezések között sem.

tográfiai erősség szempontjából lényeges, ún. ekvivalens kulcshossz nem több mint egyetlen kulcs duplája.

A tripla DES esetében viszont háromszoros ekvivalens kulcshosszal lehet számolni.

A Feistel transzformációk

A 7. ábra kapcsán láttuk, hogy a transzformációk ismételt alkalmazása csak akkor jelent nagyobb védelmet, ha mindegyik komponens-transzformációnak más-más kulcsa van. A többkulcsos rendszer használata viszont bonyolultabb. Kérdés, hogy hogyan lehet egykulcsos (szimmetrikus) titkosító rendszert úgy megszerkeszteni, hogy úgy működjön, mint egy többkulcsos, ismételt transzformációkat alkalmazó rendszer.

A megoldást *Horst Feistel*, az IBM munkatársa találta ki az 1970-es évek elején.[23]

A titkosítási rendszeréhez felhasználta a 70-es években már széles körben rendelkezésre álló számítógépeket.

Nagyon leegyszerűsítve azt találta ki, hogy a titkosítás K kulcsából egy ún. kulcs ütemező algoritmussal meghatározott számú „alkulcsot" (szubkulcsot) állít elő, és ugyanazt az összetett transzformáció sorozatot – „menet" (*round, kör*) – többször egymás után végrehajtja, de minden egyes alkalommal más-más szubkulccsal. Tehát minden egyes rundhoz más-más szubkulcsot alkalmaz.

Ezzel a megoldással a titkosítás ún. erőssége nem lett nagyobb, mint amit a K kulcs hossza (kulcstere) meghatározott, de maga a módszer annyira bonyolulttá vált, hogy csakis a nyers erő módszerével lehetett próbálkozni a feltörésénél. A DES mellesleg 64 bites nyíltszöveg blokkokhoz 64 bites kulcsot használt, de a 64 bites (8 bájtos) kulcsban „csak" 56 független bit volt, mert minden kulcsbájt egyik bitje paritásbit volt. Így a DES kulcstere 2^{56} különböző kulcsból állt.

Ennek a megoldásnak különös jelentőséget ad az a tény, hogy egyrészt 25 évig kiválóan működött, másrészt a mai szabványosított utódja[24] is lényegében hasonló

[23] Feistel 1932-ben emigrált Németországból az USA-ba. A háború alatt titkos üzenetek megfejtésével foglalkozott. Az IBM a 70-es évek elején a Lloyd biztosító társaság számára fejlesztett titkosító rendszert. Feistel a rendszerét „Dataseal"-nek akarta elnevezni de az IBM csak Demonstration Cipher-nek nevezte, amely elnevezés rövidített változataként a „Demon" elnevezést használta. Ebből lett később a „Lucifer" kriptorendszer, amely alapját képezte az 1975-ben Data Encryption Standard (DES) néven szabványosított ún. iterációs kriptorendszernek [15].

[24] Az ún. Advanced Encryption Standard (AES), amit egy hosszú pályázati és döntési folyamat után a belga Vincent Rijmen és Joan Damon Rijndael nevű iterációs kriptorendszere nyert el. Az AES 128 bites változatát 2000 októberében szabványosította a NIST.

Ha arra gondolunk, hogy de Vigenere polialfabetikus rendszere kb. 300 évig ellenállt a feltörésnek, akkor a 25 év nem tűnik soknak.

De Vigenere rendszerének a feltöréséhez azonban nem voltak meg sem azok a matematikai eszközök, amelyek a II. világháborúban már rendelkezésre álltak, sem a feltörő-gépek megszerkesztéséhez szükséges technológia nem állt még rendelkezésre.

Babbage zsenialitása kellett hozzá, mint ahogyan Rejewskyé az Enigma feltöréséhez.

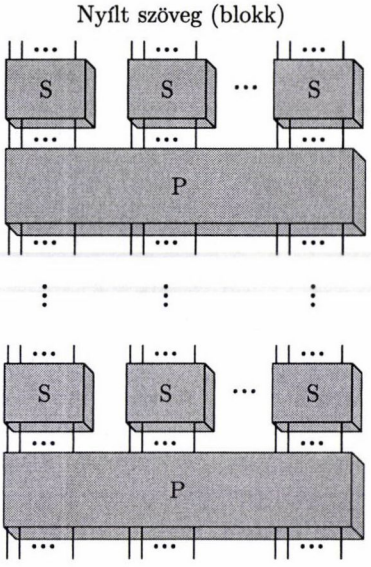
elven épül fel, csak egy-egy menetben másféle transzformációkat alkalmaz és a K kulcs hossza is 128 bit lett.

A Feistel transzformációhoz előbb még három segédfogalmat kell definiálni. Ezek a következők:

a. Egy szorzat transzformáció (*product cipher*) két vagy több komponens transzformációt kombinál olyan módon és céllal, hogy az eredő transzformáció biztonságosabb legyen, mint a komponensek bármelyike.[25] A továbbiakban ezt inkább *összetett transzformációnak* nevezzük.

b. Egy helyettesítési-permutációs hálózat (SP) olyan összetett transzformációt hajt végre, amely számos, egymást követő fokozatból áll, s e fokozatok mindegyike vagy helyettesítést, vagy permutációt hajt végre (lásd 8. ábra).

c. Egy *iterált blokk transzformáció* egy belső, ún. *rund függvény* meghatározott számú sorozatos megismétlését jelenti. Fontos paramétere a menetek (rundok) száma: r; a blokkban lévő bitek száma: n és az ún. bemeneti kulcs bitszáma: k.



8. ábra. Egy összetett transzformációt végrehajtó SP hálózat

A Feistel transzformáció az iterált blokk transzformáció egy tovább bonyolított változata. Eredeti (vagy első, a Luciferben alkalmazott) változatában a nyílt szöveg blokkjainak hosszúsága $n = 2t$, a rundok száma pedig legalább 4. Jellemzően páros számú rundot alkalmaz. A bemeneti kulcs bitjeinek száma megegyezik

[25] Figyeljük meg, hogy az elnevezés ugyan transzformációk szorzatára utal, de a definició megengedi a komponensek egymásba való beépítését, az összetett függvényeket. Ezért pontosabb is összetett transzformációról beszélni.

a blokkmérettel, tehát szintén $2t$ bit. Tartozik hozzá egy kulcs ütemező folyamat, amely minden rund számára más-más szubkulcsot állít elő a bemeneti kulcsból.

A páros számú rundnak jó oka van. A Feistel transzformáció ugyanis a nyíltszöveg blokkot megfelezi[26] és a (szub)kulcsfüggő komponens transzformációt csak egy t hosszúságú félblokkon hajtja végre egy-egy menetben. A két félblokkot aztán összekeveri a menet végén, és a következő menetben megcseréli a félblokkokat. Szigorúan véve csakis azokat a szimmetrikus transzformációkat nevezik Feistel függvényeknek, amelyekre jellemzők a fél-blokkok és ezek menetenkénti felcserélése.

Az egész eljárás jobban megérthető a 9. ábra alapján.

Az eljárásban nem tüntettük fel a szubkulcsokat ütemező algoritmust és még egy-két finom részletet sem. Egyébként igen figyelemre méltó az f Feistel függvény, amely itt a 32 bites félblokkokhoz 48 bites szubkulcsokat alkalmaz, mert – többek között – tartalmaz egy kiterjesztési és egy kompressziós transzformációt is.

Emiatt előfordulhat (és a DES-ben elő is fordul), hogy maga az f Feistel függvény nem is invertálható, de ettől még a teljes rund, illetve azok sorozata mégis csak invertálható.[27] A megfejtéskor ugyanezt a 16 menetes eljárást alkalmazzák, de a szubkulcsok sorrendje fordítottja a titkosításkor alkalmazott sorrendének. A félblokk kiterjesztése, majd a kulcsfüggő művelet végrehajtása utáni kompressziója (amelyek beleértendők az f Feistel transzformációba) ún. *lavina hatást* eredményez, ami azt jelenti, hogy a nyíltszöveg blokk egyetlen bitjének a megváltoztatása a hozzátartozó kriptogram blokkban legalább 32 bit változását okozza. A lavinahatás a rundok számának növelésével növekszik. Ez minden iterációs kriptorendszerben így van. Az AES-ben is és több más, ismert rendszerben is.

Egyébként a 128 bites AES nem bontja félblokkokra a bemeneti nyíltszöveg blokkot és 12 menetes iterációt alkalmaz.

A lavinahatás miatt nem véletlen, hogy az iterációs, szimmetrikus kriptorendszerek megfejtési kísérletei során először csak kevesebb számú runddal titkosított változatok megfejtésére szoktak törekedni. Ez még a legújabb AES feltörési kísérleteinél is így van.

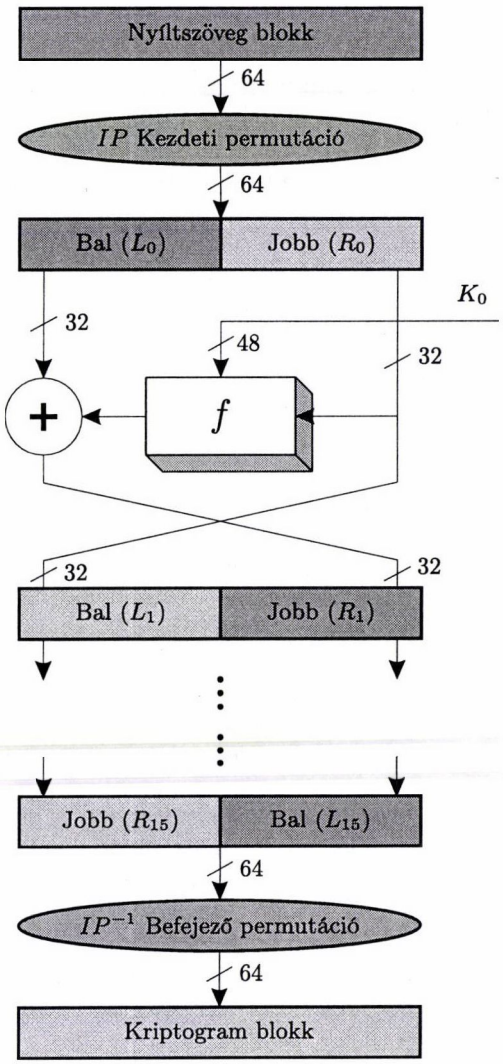
A lavinahatásról még annyit, hogy az közvetlen rokonságban van a matematikai értelemben vett kaotikus folyamatokkal.

---

[26] Van negyedelő eljárás is, pl. az IDEA esetében.

[27] Ez Menezes [24] állítása, de nem bizonyítja. A DES részletesebb elemzése azonban alátámasztja ezt az állítást. Ténykérdés ugyanis, hogy az expanzió-kompresszió transzformáció-pár alkalmazása miatt legalább is kétségek merülnek fel az f függvény invertálhatóságával kapcsolatban. Az is tény azonban, hogy a DES 25 évig jól működött. Az invertálhatóságot viszont e cikk elején a transzformációk általános követelményei során triviális alapfeltételnek tekintettünk. Mármint a teljes transzformációét, s nem annak komponenseiét. Feistel egyik óriási innovációja az, hogy rájött, hogy összetett transzformációk esetében nem feltétlenül kell minden komponensnek invertálhatónak lenni ahhoz, hogy az eredő transzformáció invertálható legyen. Az igazsághoz hozzátartozik azonban, hogy a Luciferben még nem volt ilyen expanzió-kompresszió pár, s azt az NSA javaslatára építette be a DES-be Feistel.

A nem invertálható komponensek beépítésének az a fő előnye és célja, hogy lehetetlenné tegye a transzformáció algoritmikus feltörését.



9. ábra. A Feistel transzformációt alkalmazó 16 menetes, 64 bites DES vázlata (csavart létra)

Az aszimmetrikus kriptorendszerek

Említettük, hogy kb. az 1970-es évek közepén következett be az az áttörés, amely a nagyon bonyolult és nagy számítás igényű szimmetrikus kriptorendszerek

bevezetését jelentette, méghozzá a civil alkalmazásokba.[28] Innen számítjuk az ún. negyedik és ötödik generációs kriptorendszerek korszakát.[29] A szimmetrikus kriptorendszerek elterjedésének azonban óriási gyakorlati akadálya volt, hogy a titkos kulcsot a titkosított kommunikációt megelőzően és igen nagy biztonsággal el kellett juttatni a kommunikációban részt vevő másik partnernek. A kulcsok eljuttatásának, nyilvántartásának és egyáltalában a kulcsok menedzselésének (a szimmetrikus rendszerekben) megoldatlan problémája sürgetően megkövetelte valamilyen új megoldás kialakítását. Állíthatjuk, hogy tulajdonképpen ez az igény váltotta ki az aszimmetrikus (nyíltkulcsú) kriptorendszerek feltalálását.

Kb. ugyanebben az időben támadt egy zseninek, nevezetesen Whitfield Diffienek az az ötlete, hogy valahogyan el kellene kerülni a kulcsok utaztatását.

Ennél, persze, konkrétabb ötlete is volt a megoldásra, amin aztán Martin Hellmannal és Ralph Merklevel együtt dolgoztak. 1976-ban jelentették be az ötletüket egy konferencián [35], de ők akkor még nem találtak rá működőképes implementációt.

Diffie olyan, ún. egyirányú függvényeket keresett, amelyeket az egyik irányban aránylag könnyen ki lehet számolni, de a másik irányban csak olyan sok számítási kapacitással, ami nem éri meg a dolgot, vagy nincs annyi ráfordítható idő.

Ma már széles körben ismert Ron Rivest, Adi Shamir és Leon Adleman feltalálók neveinek kezdőbetűiről elnevezett RSA titkosítás, amely alapja a természetes számok prím faktorizációja. (IFP = Integer Factorization Problem). Nagyon leegyszerűsítve ugyanis arról van szó, hogy két nagy prímszámot aránylag nem nehéz összeszorozni, de egy igen nagy szám esetében csak nagyon sok próbálkozással lehet az adott szám törzstényezőit meghatározni. Ez egy tipikus egyirányú függvényprobléma.

No de hogyan lehet összehozni egy szöveg titkosítását számokkal? Nos, blokk titkosítás esetén egy $n$bites blokk mindig tekinthető olyan egészszámnak, amely értéke 0 és $2^n - 1$ közé esik. Fontos, hogy egészszámokról van szó, s ennek a következőkben komoly jelentősége lesz. Fontos az is, hogy egy korlátos tartományba eső pozitív egészekről, vagyis természetes számokról van szó.

Térjünk vissza azonban a 3. ábra teljes titkosítási/visszafejtési folyamatához.

Láttuk, hogy triviális megoldásnak tekintették korábban, hogy az E és a D transzformációk kulcsai azonosak, maguk a transzformációk pedig egymás inverzei. Sőt: gyakorlati szempontok miatt esetleg saját maguk inverzei. Ez valamennyi szimmetrikus kriptorendszer jellemzője.

Ma már eléggé kézenfekvő az, hogy nem feltétlenül kell az E és a D transzformációk kulcsainak azonosaknak lenniük, hanem elképzelhető, hogy azok például

[28] A katonai alkalmazások körében természetesen korábban is léteztek ilyenek, de azért érdemes felfigyelni arra, hogy az amerikai hadsereg még a koreai háborúban is használt az Enigmához hasonló titkosító/megfejtő gépeket

[29] A kriptorendszer-generációkat ugyanúgy az alkalmazott technológia jellemzi, mint a számítógépek generációit, de ezzel a kérdéssel itt nem foglalkozunk. Megemlítjük azonban, hogy nem csak a transzformációs technológia, hanem a jellemző kommunikációs technológia is meghatározó [2].

egymásnak az aktuális transzformációra vonatkoztatott inverzei, s maguk a transzformációk azok, amelyek azonosak.

Az előbbiekben láttuk, hogy maguk a transzformálandó blokkok egy felülről korlátos tartományba eső természetes számok. Egyszerű esetben a kulcsok is természetes számok.

A kérdés tehát úgy tehető fel, hogy van-e a természetes számok halmaza felett értelmezett zárt művelet, amelynek az eredménye is természetes szám, s ráadásul egy felülről korlátos halmaz tagja.

Nos, a modulo n összeadás és a modulo n szorzás például ilyen műveletek. Nem véletlen, hogy az RSA rendszer is ezt alkalmazta. A modulo n direkt műveletek inverzei is léteznek.

A modulo n összeadás és kivonás inverz műveletpárt már a XVI. és XVII. század fordulóján de Vigenere alkalmazta az általa kitalált többábécés rendszerhez, amelynek alsó határesete az egyábécés Caesar-féle titkosítás is és felső határesete a nagyon sok ábécés Enigma titkosítás is.

A modulo n kongruenciák a 0-tól $(n-1)$-ig terjedő, n darab *maradékosztályba* képezik le a természetes számok megszámlálhatóan végtelen nagy halmazát. Ezek a maradékosztályok a 0 és $(n-1)$ közötti természetes számokkal jelölhetők.

A modulo n műveleteknek van egy elméleti és egy abból következő gyakorlati előnye is. Az elméleti előny az, hogy ezek a műveletek a természetes számok halmaza felett zártak. A gyakorlati előny pedig az, hogy a maradékosztályok véges száma miatt a számítások során elkerülhető a túlcsordulás.

Evariste Galois (1811–1832) megmutatta, hogy minden véges halmazon értelmezett művelet izomorf a modulo m maradékosztályokon értelmezett valamilyen modulo m művelettel. Ennek következménye az, hogy ha a kriptográfiai leképezések véges (vagy legalább megszámlálhatóan végtelen) halmazokon értelmezettek, akkor a modulo m műveletek gyakran feltűnnek a legkülönbözőbb elvű kriptográfiai leképezések körében is.

Az inverz fogalmáról itt annyit, hogy ha egy A halmaz felett értelmezett és zárt a $\otimes$ művelet és az A halmaznak eleme mind a, mind b, akkor ez a két elem egymásnak inverze a $\otimes$ műveletre nézve, ha

$$a \otimes b = e,$$

ahol e a $\otimes$ művelet egységeleme (amely, persze, szintén eleme az A halmaznak). Azt, hogy a b elem az a elem inverze, a

$$b = a^{-1},$$

jelöléssel jelöljük. Például a szorzás művelete esetén két szám (a és a^{-1}) akkor inverzei egymásnak, ha

$$a \cdot a^{-1} = 1.$$

Jelöljünk most egy n blokkhosszúságú nyílt szöveget P-vel és a mondottak értelmében értelmezzük P-t számként. Igen egyszerűen végrehajtható ekkor a következő titkosítási transzformáció:

$$C \leftarrow P^a \bmod n$$

A kapott eredmény mindenképpen 0 és $(n-1)$ közé esik, tehát alkalmasan választott hatványozási algoritmus (pl. a gyors hatványozás) esetében nem fordulhat elő túlcsordulás, akármekkora is az a hatványkitevő. Vegyük észre, hogy itt a titkosítás kulcsa maga az a kitevő.

A visszafejtés ugyanazzal a hatványozási művelettel történhet, de ekkor az a^{-1} (inverz) hatványra kell emelni a C kriptogramot:

$$C^{a^{-1}} \bmod n = (P^a)^{a^{-1}} \bmod n = P^{a \cdot a^{-1}} = P^1 = P.$$

A visszafejtés kulcsa tehát a^{-1}, vagyis a titkosító kulcs inverze, a titkosító transzformáció pedig pontosan megegyezik a visszafejtő transzformációval.

A két kulcs nem azonos, hanem egymás inverzei. Bizonyos szempontból éppen olyan triviális megoldás, mint a kulcsok azonossága volt a szimmetrikus kriptorendszerek esetében. Míg azoknál a transzformációk voltak egymás inverzei, és a kulcsok azonosak, az aszimmetrikus kriptorend-szereknél a kulcsok egymás inverzei és a transzformációk azonosak.[30] Maga az „aszimmetrikus" elnevezés arra utal, hogy a kulcsok nem azonosak. Egyébként azonban ezek a kétkulcsos rendszerek a szó szoros értelmében legalább annyira szimmetrikusak, mint a szimmetrikusnak nevezett (egykulcsos) kriptorendszerek.

A 4. ábra és az 5. ábra pontosan ezt a szimmetriát hivatott bemutatni.

Vegyük észre azt is, hogy a bemutatott rendszerben fontos szerepe van az n modulusnak is, azaz az egész titkosító/visszafejtő rendszer működtetéséhez az inverz kulcspáron kívül a modulust is ismerni kell, tehát legalább három paraméter van (Itt!).

Az aszimmetrikus kriptorendszerek azért nem annyira egyszerűek, mint azt az előbbiekben bemutattuk. Számos gyakorlati követelmény is létezik, amelyeket ki kell elégíteni.

Alapelv, hogy ha a két kulcs nem azonos, akkor az egyiket nyilvánosságra is lehet hozni. Ebből azonban következik, hogy a nyilvános kulcsból semmilyen módon ne lehessen kiszámítani a párját, vagyis a titkos kulcsot.

Vizsgáljuk meg most ezt a transzformációt egy kissé közelebbről!

Az RSA nyíltkulcsú rendszer és az egyirányú függvények[31]

Az előzőekben (amikor csak első közelítésben kívántuk bemutatni az alapelvet) ismételt hatványozással értük el, hogy visszakapjuk az eredeti szöveget.

A példaként bemutatott egyszerű hatványozás esetén, ha az egyik kitevő és a modulus ismert, akkor az inverz kitevő kiszámolható. Az RSA rendszerben ezért

[30] Mármint az itt bemutatott egyszerű esetben. Bonyolultabb rendszereknél azért nem lehet a transzformációkat teljesen azonosaknak tekinteni, mint ahogyan a negyedik generációs iteratív kriptorendszerekben sem voltak teljesen azonosak.

[31] Köszönetet kell mondanom Tóth Gergelynek, a Veszprémi Egyetem Székesfehérvárra kihelyezett AIFSz központja tanárának e fejezet megírásához nyújtott segítségéért.

nem is egyszerűen csak inverz kitevőkkel, hanem két nagy (10^{150} nagyságrendű) prímszám Euler függvényével dolgoznak.

Ha találunk olyan e és d számokat, amelyek szorzata valamilyen alkalmas közös n modulussal osztva 1-et ad, akkor titkosíthatunk az e és n számokat használva:

$$c \leftarrow p^e \bmod n.$$

A visszafejtésre a $p \leftarrow c^d \bmod n$ formulát használhatjuk, hiszen

$$c^d = (p^e)^d = p^{e \cdot d} \equiv p^1 \bmod n. \tag{0.1}$$

A probléma alapvetően az, hogy míg valós számok esetében egy 0-tól különböző számnak mindig van inverze a szorzásra nézve (történetesen a reciproka), addig ez a maradékosztályokban már korántsem mindig van így. Például egy páros számot bármilyen számmal megszorozhatunk, a szorzat soha sem fog 1 maradékot adni egy páros modulus esetében.

Pierre Fermat [1601–1665] 1660 körül rájött[32], hogy az eljárás „működik", ha az $e \cdot d$ szorzat prímszám. Fermat eredményének azonban nem sok gyakorlati alkalmazása van; tekintve, hogy csak az $e = 1$ vagy $d = 1$ esetben működik. Ez kriptográfiai szempontból azt jelentené, hogy magát a nyílt szöveget küldjük el a címzettnek.

Szerencsére *Leonard Euler* [1707–1783] általánosabb formulát talált a Fermat képletnél, amelyben a kitevőnek nem kell prímszámnak lennie, és a hatványozás után mégis visszakapjuk osztási maradékként az eredeti számot.

Az Euler féle képletben

$$p^{e \cdot d} \equiv p \bmod n,$$

$$\text{ha } e \cdot d = k \cdot \varphi(n) + 1 \text{ alakú}, \tag{0.2}$$

ahol k tetszőleges természetes szám. A $\varphi(n)$ jelölés az n számnál kisebb, n-hez relatív prímek darabszámát jelöli, azaz hogy hány olyan szám van 0 és n között, amelyeknek 1-en kívül nincs olyan osztójuk, amely n-et is osztaná.[33] A (0.2) feltétel ekvivalens azzal a feltétellel, hogy e és d egymás inverzei a $\varphi(n)$ modulusra. Ha tehát a kódolást és a dekódolást az üzenetblokk hatványozásával oldjuk meg, amint azt a (0.1) formulában tettük, akkor ez működni fog abban az értelemben, hogy az ismételt hatványozással visszakapjuk az eredeti üzenetet. A gond csak $\varphi(n)$ kiszámításánál van.

Prímszámokra elég kézenfekvő, hogy $\varphi(p) = p - 1$, és azt is viszonylag könnyű belátni, hogy szorzat φ-je a φ-k szorzata, azaz

$$\varphi(p \cdot q) = \varphi(p) \cdot \varphi(p). \tag{0.3}$$

[32] Kis (más néven: „karácsonyi") Fermat tétel: $ap - 1 \equiv 1 \bmod p$, ha p prímszám.

[33] Fermat-val ellentétben Euler bebizonyította a saját tételét, és ezzel visszamenőleg a kis-Fermat tétel is bizonyítást nyert. Egy prímszámot ugyanis 1-en kívül egyetlen nála kisebb természetes szám sem oszt, azaz hozzá képest az összes nála kisebb szám relatív prímszám. Ezek szerint prímszámokra $\varphi(p) = p - 1$, amiből következik a kis-Fermat tétel állítása.

Vagyis egy tetszőleges n modulus φ-je könnyen meghatározható, ha ismerjük n prímtényezős felbontását.

Rivest, Shamir és Adleman frappáns módon oldották meg, hogy a kulcsokat létrehozó „illetékes" ismerje a modulus prímfelbontását, az illetéktelen támadó azonban ne kerülhessen birtokába ennek az információnak: egyszerűen választottak két prímszámot, és ezek szorzatát használták a kódoláskor modulusként.

A modulus prímtényezőinek segítségével meghatározható az n modulus Euler-féle φ-je, ebből pedig az e és d kitevők, amelyek egymás inverzei a $\varphi(n)$ számra nézve.[34]

Ha egy illetéktelen akarja megfejteni a titkos d számot a közzétett e és n számok alapján, mindenképpen meg kell határoznia $\varphi(n)$-et, hogy megtudja, egyáltalán milyen modulusra kell kiszámítania az e szám inverzét.

Jelenlegi ismereteink szerint viszont az Euler féle $\varphi(n)$-et csakis n prímtényezőinek ismeretében lehet meghatározni. Ezek megkeresése a jelenleg használt kulcsméretek és számítástechnika esetén sokszorosa a Világegyetem várható életkorának.

Az RSA kódolás erősségét tehát egy szemtelenül egyszerű ún. egyirányú függvény adja. A titkosítás kulcsának létrehozója maga választja meg a törzstényezőket, majd azok *összeszorzása* révén számítja ki a kódoláskor/dekódoláskor használt n modulust. A törzstényezők ismeretében a modulus kiszámítása nevetségesen egyszerű: egyetlen szorzási művelet. A támadó azonban a fordított műveletet kénytelen elvégezni, azaz egyedül az n modulus alapján meghatározni azokat a prímszámokat, amelyek szorzataként a modulus létrejött. A művelet, amely az egyik irányban akár tollal és papíron is viszonylag gyorsan elvégezhető, visszafelé még a világ minden informatikai kapacitásának birtokában is elképzelhetetlenül hosszú időkig tartó feladatot ró a rendszer feltörőjére.

[34] A módszer elvéből következik, hogy az n modulusnak és e-nek relatív prímeknek kell lenniük. Igen nagy számokról lévén szó, ezt nem is olyan egyszerű megvizsgálni. Ehelyett a gyakorlatban azt teszik, hogy e-nek egy nagy prímszámot választanak, amely ha sem p-vel, sem q-val nem egyezik, akkor biztosan relatív prím n-hez képest is.

Erre a célra Mersenne vagy Fermat prímeket szokás választani, mert a bináris alakjaik nagyon könnyen előállíthatók. A gyors hatványozás alkalmazhatósága miatt a Fermat prímek tulajdonképpen alkalmasabbak lennének, de mindössze 6 ilyen prímszám van, amelyek közül a legnagyobbat valóban gyakorta használják is nyílt kulcsként.

A Mersenne prímekből sokkal több van, de ezekkel az eljárás számításigénye lényegesen nagyobb.

(Az oka a gyors hatványozásban rejlik.)

A Fermat számok $2^{2^k} - 1$ alakú számok, de csak $k < 6$ esetén prímek. A gyakorlatban éppen a $k = 5$-öt szokták alkalmazni. Az ilyen bináris számok Hamming súlya csak 2, és ezért a gyors hatványozás valóban gyorsan végrehajtható.

A Mersenne számok $2^k - 1$ alakúak és nem mindegyik ilyen szám prím. A k természetes számra viszont nincs semmilyen kikötés.

További, titkosításra alkalmazható csoportműveletek

A diszkrét logaritmus probléma (DLP)

Véges elemszámú G csoportban a g generáló elem és egy szabadon választott a csoport-elem ismeretében határozzuk meg azt az x kitevőt, amelyre $g^x = a$ ($x = \log_a g$, de x-nek egészszámnak kell lennie) [22].

A Diffie–Hellman (DH) probléma

Véges elemszámú G csoportban a g generáló elem, valamint g^a és g^b ismeretében határozzuk meg g^{ab}-t.

A következő csoportok használata terjedt el:

- A p elemből álló mod p maradékosztályok multiplikatív csoportjaként értelmezhető csoport, amely valódi (és véges) részhalmaza a természetes számok halmazának.
- A $GF(2^m)$ test $2^m - 1$ elemű multiplikatív csoportja.
- Az elliptikus görbék pontjain értelmezett csoport.

Ezek részletesebb diszkussziója nélkül csak azt szeretnénk kiemelni, hogy mindegyik ilyen aszimmetrikus kriptorendszer az inverz kulcsok elvét alkalmazza ugyanazon csoportművelet mellett.

Mindegyikre léteznek megoldási algoritmusok, amelyek elsősorban a számításigény és a futási idők tekintetében versenyeznek egymással. Mégsem jelenthető ki azonban, hogy egyik algoritmus jobb, mint egy másik, mert a hatékonyságuk a konkrét alkalmazástól is függ.

Az RSA algoritmus pl. máig is a legelterjedtebb, és nem is ok nélkül, de csak adott (s nem is túl nagy) blokkhosszúságú nyíltszövegek titkosítására alkalmas.

Nem véletlen az sem, hogy pl. a digitális aláírások esetében a Diffie–Hellman algoritmust szabványosították, és az sem véletlen, hogy nagyon hosszú üzenetek egy menetben való nyíltkulcsú titkosítására pedig ma az ún. NTRU (Number Theory Research Group at MIT) rendszert tartják a legalkalmasabbnak.

Mindezek bemutatása azonban messze meghaladja e cikk terjedelmi korlátait és célkitűzését is.

Összefoglalás

A kriptográfia történetének és fejlődésének nagy mérföldkövei jó rendszerezési alapot nyújtanak e cikkben bevezetett kriptogenerációk fogalmához. Áttekintve a kriptorendszerek fejlődését azt találtam, hogy néhány alapvető fogalom nem, vagy nem egyértelműen definiált. Ilyen volt pl. az ún. blokkos titkosítások blokk fogalma, a több helyütt említett lineáris transzformációk fogalma és maguk a kriptográfiai transzformációk is.

E cikkben leírt áttekintés alapvető motívuma éppen e fogalmak „rendbetétele" volt. Legalább is megkíséreltem néhány általam felfedezni vélt ilyen hiányt pótolni

és ebben az írásban összefoglalni mindazt, amire jutottam. Szép számmal maradtak nyitott kérdések is.

Írásommal buzdítani is szeretnék fiatal kutatókat arra, hogy *érdemes* egy tudományterület alapkérdéseivel foglakozni.

A cikk a kriptorendszerek generációinak fogalmi és definitív bevezetése után a kriptográfiai transzformációk (matematikai értelemben leképezések) általános jellemzőivel, tulajdonságaival foglalkozik.

Megmutatja, hogy matematikai értelemben miért helyesebb leképezésekről beszélni, mint transzformációkról és vizsgálja e leképzések értelmezési tartományait, valamint az értékkészletét.

Definiálja a leképzések őstartományán értelmezhető műveletet, nevezetesen a konkatenációt, és ennek az értelmezési tartománya segítségével definíciót ad a kriptográfiai „blokk" fogalmára.

Heurisztikus módon megfogalmazza e leképezések általános követelményeit. Megfogalmaz egy sejtést, amely szükséges és elegendő feltétel a klasszikus helyettesítési és permutációs transzformációk felcserélhetőségére, valamint összevonására. (A blokkhosszúságok egészszámú arányára vonatkozó kijelentésről van szó.)

Foglalkozik az egyszerű leképző függvények inverz tulajdonságaival. Claude E. Shannon javasolta elsőként az ún. produkt transzformációkat, amelyek valójában vagy nem kommutatív produktumok, vagy összetett függvények. Minden esetre komoly szerepük van a modern, ún. negyedik generációs, *szimmetrikus* kriptorendszerekben. A cikk kitér az ún. Feistel transzformációkra és alkalmazásaikra, a többkomponensű transzformációk komponensei invertálhatóságának a kérdésére, s végül az ún. aszimmetrikus (nyíltkulcsú) kriptorendszerekben alkalmazott transzformációk kapcsán azt igyekszik bemutatni, hogy azok a tradicionális kriptorendszereknek mintegy a tükörképei.

Megfogalmaz egy állítást is arra, hogy az aszimmetrikus kriptorendszerekben alkalmazott művelet (amelyre az inverz kulcsok vonatkoznak) véges halmazok esetén szükségképpen izomorf a modulo m maradékosztályok feletti valamilyen művelettel.

Ezért aztán egyáltalán nem véletlen, hogy a modulo m aritmetika a legkülönbözőbb alapelvű kriptorendszerek esetében minduntalan visszaköszön.

(A szerző számára talán legfrappánsabb módon a Knapsack-féle nyíltkulcsú rendszerben, amely egy távolság különböző távolságokkal való lefedéséből indul ki, vagyis egy geometriai problémából. A fedő távolságoknak szupernövekvő sorozatot kell alkotniuk, s akkor a lefedés kimutathatóan egyértelmű és unikális. A Knapsack rendszert ugyan nem szabványosították, de nagyszerűen bemutatható a segítségével a modulo n műveletek előfordulása. [4]-ben megtalálható a részletes leírása is kidolgozott példával együtt.)

Szakirodalmi Hivatkozások

[1] Tóth Mihály, *A kriptográfia alap-transzformációiról,* Természet-, Műszaki- és Gazdaságtudományok alkalmazásának 2. nemzetközi konferenciája. 2003. május 10. Szombathely.

[2] Tóth, Mihály, The Four Generation of Criptosystems, in: *The Second Word Meeting of Information Scientists held in Budapest,* 5-8 June, 2000.

[3] Tóth Mihály, *Bevezetés a kriptográfiába I. A kriptorendszerek első generációja,* Kandós jegyzet, 2002. szept., PDF formátum, 575 KB, http://www.szgti.bmf.hu/~mtoth/download/Kriptografia/.

[4] Tóth Mihály – Prof. Randall K. Nichols, *Aszimmetrikus kriptorendszerek,* Főiskolai jegyzet BMF 163/2000.

[5] Tóth Mihály, *A DES (Data Encryption Standard),* Főiskolai jegyzet; BMF 27/2000.

[6] Tóth Mihály – Tóth Gergely: *Az elektronikus információ titkosítása és hitelesítése,* Cikk a GTE Gépgyártástechnológia c. folyóirata XXXIX. évf. 10. számában, p. 241–251.

[7] Tóth Mihály, *Szimmetrikus és aszimmetrikus kriptorendszerek,* Alkalmazott matematika, statisztika és informatika konferencia A Kodolányi János Főiskolán 2000 szeptemberében.

[8] Tóth Mihály, *Nyiltkulcsú kriptorendszerek és alkalmazásuk digitális aláírásra,* Alkalmazott matematika, statisztika és informatika konferencia A Kodolányi János Főiskolán 2000 szeptemberében.

[9] Tóth Mihály, *Karakter statisztikák,* Prezentáció a „60 év a műszaki felsőoktatásért" c. Kandó konferencián, 2002 november, ISBN: 963 7158 03 0.

[10] Tóth Mihály, *Adatrejtés és szövegstatisztikák,* Cikk és előadás az Informatika a felsőoktatásban 2002 konferencián Debrecenben, ISBN 963 472 691 7.

[11] Tóth Mihály, *A kriptográfiai transzformációkról,* Cikk és előadás a Természet-, műszaki- és gazdaságtudományok alkalmazása c. 2. nemzetközi konferencián, Szombathelyen, 2003 május.

[12] *Iterative cryptosystems,* Modern Symmetric Cryptography by Mihaly Tóth (Departemental Scientific Seminar) H.T.I. 10th of Ramadan City, Egypt, Dec. 2003.

[13] *Modern Cryptosystems,* Symmetric and Public key Cryptography Presentation by Dr. Mihaly Toth, http://www.infosec-technologies.com/.

[14] Végül megemlítem itt a saját honlapomat, amelyen számos munkámat közzétettem, hogy a hallgatóim – és mások – számára hozzáférhető legyen: http://www.szgti.bmf.hu/~mtoth/.

[15] *The Codebrakers by David Kahn Scribner NY,* First published in 1967. The cited one is published in 1996. ISBN: 0-684-83130-9

[16] *Codes, Ciphers & Other Cryptic & Clandestine,* Communication by Fred B. Wrixon Black Dog & Leventhal Publishers, NY 1998, ISBN: 1-57912-040-7

[17] *ICSA (International Computer Security Association),* Guide to Cryptography by Randall K. Nichols. Published by McGraw-Hill NY... 1999. ISBN: 0-07-047166-5 *Megj: Szerintem alapműnek tekinthető, bár ma már ráférne némi frissítés, de ugyancsak a McGraw Hill kiadásában már készülőben van egy nagy, több kötetes kriptográfiai enciklopédia, amelynek egy fejezetét – megtisztelő felkérésre – lektoráltam is.*

[18] Feistel összefoglalja az előzményeket és bemutatja az iterációs rendszerét: http://williamstallings.com/Extras/Security-Notes/lectures/blockA.html.

[19] Enigma: http://www.codesandciphers.org.uk/enigma/.

[20] The Zimmerman telegram: http://www.cypher.com.au/crypto_history.htm.

[21] De Vigenere (1): http://www.bibmath.net/crypto/poly/vigenere.php3.

[22] De Vigenere (2): http://www.antilles.k12.vi.us/math/cryptotut/vigenere.htm.

[23] *A de Vigenere rendszer feltörése* (Babbage és Kasisky) http://www.infowrangler.com/phpwiki/wiki.phtml?title=Vigen%E8re_Cipher.

[24] Handbook of Applied Cryptography by A. J. Menezes at al., CRC Press, 1997. ISBN: 0-8493-8523-7.

[25] A PGP egy gyűjtő webkikötője: http://www.pgpi.org/.

[26] Grzegorz Rozenberg (editor), Arto Salomaa (editor), *Handbook of Formal Languages: Volume 1,* Springer, 1997, ISBN: 3540604200.

[27] G. Paun (editor), Gheorghe Paun (editor), Grzegorz Rozenberg (editor), Arto Salomaa (editor), *Current Trends in Theoretical Computer Science Algorithms and Complexity,* World Scientific Publishing Company (April 1, 2004), ISBN: 9812389652

[28] Salomaa, Arto, Public-Key Cryptography (Texts in Theoretical Computer Science. An EATCS Series, 1996).

[29] Egy rövid összefoglalás a Chomsky féle nyelvcsaládokról: http://www.absoluteastronomy.com/encyclopedia/C/Ch/Chomsky_hierarchy.htm.

[30] Ron Rivest egyik korai összefoglaló munkája, amelyben a szimmetrikus rendszerekről és a protokollokról is ír: http://theory.lcs.mit.edu/~rivest/Rivest-Cryptography.pdf.

[31] Pierre Blanc: *Vigenere cipher,* http://it.geocities.com/teutoburgo/java/javi/algorithm.html.

[32] Kupás Péter, *Elliptikus görbék az oktatásban és rejtjelző rendszerekben,* Előadás az Informatika a felsőoktatásban 2002. c. debreceni konferencián.

[33] Endrődi Csilla, *Az RSA és az ECC gyakorlati összehasonlítása,* BMGE MIT Ph.D. dolgozat.

[34] *Practical Cryptography by Niels Ferguson & Bruce Schneier,* Wiley Publishing Inc. 2003, ISBN: 0-471-22357-3

[35] Merkle, Ralph, *The New Directions in Cryptography,* National Computer Conference, Berkeley, 1976 június.

[36] Virasztó Tamás, *Titkosítás és adatrejtés,* NetAcademia Kft. 2004. ISBN: 963 214 253 5. *Ez a végül, de egyáltalán nem utolsó sorban megemlített könyv ma a legjobb (ha nem az egyetlen említésre méltó) mű az utóbbi 5 év e témakörbe tartozó magyar könyvei közül.*

TÓTH MIHÁLY
BUDAPESTI MŰSZAKI FŐISKOLA
KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
e-mail: toth.mihaly@szgti.bmf.hu

BASIC MATHEMATICAL TRANSFORMATIONS IN CRYPTOGRAPHY

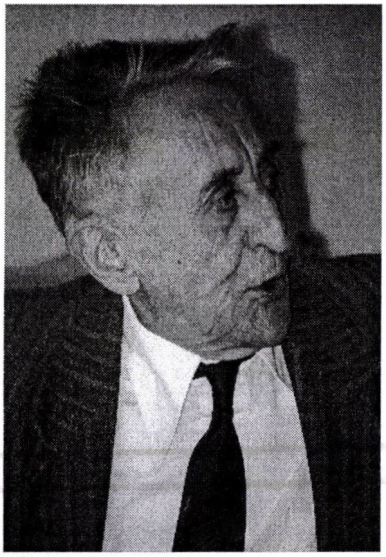
MIHÁLY TÓTH

This paper introduces the concepts of Crypto-generations then deals with the mapping and transformation functions of Cryptosystems. It pays particular interest on the concept of blocks and alternating block cipher transformations, the concept of linearity and the changeability of substitution and permutation type transformations. It defines the general conditions of those transformation. The paper discusses the invertibility of crypto-transformations particularly Feistel functions.

The paper demonstrates the symmetry of iterative and public key transformations. Its main goal is to make clear some basic concepts of four and fifth generation cryptosystems.

Alkalmazott Matematikai Lapok **23** (2006), 433–434.

TANDORI KÁROLY
(1925–2005)



2005. január 24-én elhunyt Tandori Károly akadémikus, a Szegedi Egyetem Kossuth- és Széchenyi-díjas professzor emeritusa, sokak szeretett mestere.

Amellett, hogy saját munkássága a matematikai analízis elméletében hozott neki világhírnevet, az alkalmazott matematika ügyét és a matematika magyar nyelven történő művelését annyira fontosnak tartotta, hogy lapunk indításakor a négy alapító szerkesztő egyikének a szerepét is felvállalta 1975-ben, és szerkesztőként haláláig, harminc éven át áldozatosan szolgálta azt. Emlékének itt a vele készült interjúk és a róla szóló írások jegyzékének közzétételével tisztelgünk.

[1] Petri Ferenc, Lebesgue és Fourier útján: Beszélgetés Tandori Károllyal, *Délmagyarország* **62/310** (1972. december 28., csütörtök), 4. oldal.

[2] Sulyok Erzsébet, A matematika – az más: Beszélgetés Tandori Károly és Leindler László Széchenyi-díjasokkal, *Szeged* **4/5** (1992. május), 4. oldal.

[3] Sulyok Erzsébet, „Diákkoromban azt hittem, a matematikával minden megoldható és megmagyarázható": Tandori Károly akadémikus, *Szeged* **6/3** (1994. március), 10–14. [Másodközlés in: Sulyok Erzsébet, *Aranymosás: Beszélgetések szegedi akadémikusokkal* (Szerkesztette Tandi Lajos), pp. 134–141. Délmagyarország Könyv-, Lapkiadó és Nyomdaipari Kft. Szeged, 1995.]

[4] Csörgő Sándor, Tandori Károly 70 éves, *Polygon* **5/1** (1995), 1–18.

[5] Totik Vilmos, A tribute to Károly Tandori and László Leindler, *Acta Scientiarum Mathematicarum (Szeged),* **60** (1995), 3–30.

[6] Móricz Ferenc, On the scientific work of Károly Tandori, in: *Approximation Theory and Function Series,* Dedicated to Károly Tandori on his 70th birthday, Bolyai Society Mathematical Studies **5** (Budapest, 1996), pp. 7–34.

[7] Árvai Mária, „Önként és egyedül Szegedet soha nem hagytam el": Beszélgetés Dr. Tandori Károly matematikussal, in: *Portrék a Szegedi Tudományegyetemről,* Messzelátó (Szeged, 2002), pp. 309–320.

[8] Sulyok Erzsébet, Fölmenni a homokfövenyen: Elhunyt Tandori Károly matematikus, *Délmagyarország* **95/21** (2005. január 26., szerda), 5. oldal.

[9] Szegedi Tudományegyetem, Természettudományi Kar, Bolyai Intézet: In memoriam Tandori Károly (1925–2005), *Délmagyarország* **95/38** (2005. február 15., kedd), 7. oldal.

[10] (Szerkesztőség), Utolsó útjára kísérték Tandori Károlyt Szegeden, *Délmagyarország* **95/40** (2005. február 17., csütörtök), 7. oldal.

[11] Csörgő Sándor, Győry Kálmán, Császár Ákos, Tandori Károly sírjánál, *Szeged* **17/3** (2005. március), 20–22. [Másodközlés in: *Emlékbeszédek 2002–2005,* Magyar Tudományos Akadémia, Budapest, sajtó alatt.]

[12] Csörgő Sándor, Tandori Károly (1925–2005), *Magyar Tudomány* **2005/7**, 907–909.

[13] Csörgő Sándor, Károly Tandori (1925–2005), publications of Károly Tandori, *Acta Scientiarum Mathematicarum (Szeged),* **71** (2005), 3–18.

[14] (Szerkesztőség), Károly Tandori (1925–2005), *Acta Mathematica Hungarica* **107** (2005), 175.

[15] (Szerkesztőség), Károly Tandori (1925–2005), *Analysis Mathematica* **31** (2005), i–iv.

[16] Sindely Pál, Matematikus emlékek vonzásában: Tandori Károly professzor születésének 80. évfordulójára, *Szeged* **17** (2005. szeptember–október), 31–33.

[17] `www.math.u-szeged.hu`.

A kiadásért felelős a BJMT főtitkára
Szedte és tördelte az Egyenes Bt.

Nyomta a Nagy és Társa Nyomda és Kiadó Kft., Budapest
Felelős vezető: Szűcs Ernő

Budapest, 2006
Megjelent 20 (A/5) ív terjedelemben
250 példányban
HU ISSN 0133-3399

ÚTMUTATÁS A SZERZŐKNEK

Az Alkalmazott Matematikai Lapok csak magyar nyelvű dolgozatokat közöl. A kéziratok gépelését olyan formában kérjük, hogy minden gépelt oldal 25, egyenként átlag 50 betűhelyes sort tartalmazzon. A közlésre szánt dolgozatokat e-mailen az `aml@math.elte.hu` címre kérjük elküldeni az ábrákat tartalmazó fájlokkal együtt. Előnyben részesülnek a TEX-ben elkészített dolgozatok.

A kéziratok szerkezeti felépítésének a következő követelményeket kell kielégíteni. A fejlécnek tartalmaznia kell a dolgozat címét, a szerző teljes nevét, valamint annak a városnak a nevét, ahol a szerző dolgozik. A fejléc után egy, képletet nem tartalmazó, legfeljebb 200 szóból álló kivonatot kell minden esetben megadni. A dolgozatot címmel ellátott szakaszokra kell bontani, és az egyes szakaszokat arab sorszámozással kell ellátni. Az esetleges bevezetésnek mindig az első szakaszt kell alkotnia. Az irodalomjegyzék után, a kézirat befejezéseképpen fel kell tüntetni a szerző teljes nevét és a munkahelye (illetve lakása) pontos címét. A dolgozatban előforduló képleteket szakaszonként újrakezdődően, a képlet előtt két zárójel közé írt kettős számozással kell azonosítani. Természetesen nem szükséges minden képletet számozással ellátni. Az esetleges definíciókat és tételeket (segédtételeket és lemmákat) ugyancsak szakaszonként újrakezdődő, kettős számozással kell ellátni. Kérjük a szerzőket, hogy ezeket, valamint a tételek bizonyítását a szövegben kellő módon emeljék ki. Minden dolgozathoz csatolni kell egy angol, német francia vagy orosz nyelvű, külön oldalra gépelt összefoglalót.

Mind az ábrákat, mind a lábjegyzeteket a dolgozat szakaszokra bontásától független, folytatólagos arab sorszámozással kel ellátni. Az ábrák elhelyezését a dolgozat megfelelő helyén, széljegyzetként feltüntetett, ábraazonosító sorszámokkal kell megadni. A lábjegyzetekre a dolgozaton belül az azonosító sorszám felső indexkénti használatával lehet hivatkozni.

Az irodalmi hivatkozások formája a következő. Minden hivatkozást fel kell sorolni a dolgozat végén található irodalomjegyzékben, a szerzők, illetve a társszerzők esetén az első szerző neve szerint alfabetikus sorrendben úgy, hogy a cirill betűs szerzők nevét a Mathematical Reviews átírási szabályai szerint latin betűsre kell átírni. A folyóiratban megjelent cikkekre [1], a könyvekre [5], a kötetben megjelent dolgozatokra [4], a disszertációkra [3] és a gépi program leírásokra [2] a következő minta szerint kell hivatkozni:

[1] Farkas, J., Über die Theorie der einfachen Ungleichungen, *Journal für die reine und angewandte Mathematik* **124** (1902) 1–27.

[2] Kéri, G., „DUALSIMP", rutin a CDC 3300-ás gépekre (Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete, CDC 3300 felhasználói ismertetők 2. 1973. május) 19–20.

[3] Prékopa, A. „Sztochasztikus rendszerek optimalizálási problémáiról", doktori értekezés. Magyar Tudományos Akadémia, Budapest, 1970.

[4] Prabhu, N. U. „Recent research on the ruin problem of collective risk theory", in: *Inventory Control and Water Storage* Ed. A. Prékopa (János Bolyai Mathematical Society and North-Holland Publishing Company, Amsterdam–London, (1973) 221–228.

[5] Zoutendijk, G. *Methods of Feasible Directions* (Elsevier Publishing Company, Amsterdam and New York, 1960).

A dolgozatok szövegében az irodalmi hivatkozás számait szögletes zárójelben kell megadni, mint például [5] vagy [4, 76–78]. A szerzők a dolgozatukról 50 darab ingyenes különlenyomatot kapnak. A dolgozatok után szerzői díjat az Alkalmazott Matematikai Lapok nem fizet.

TARTALOMJEGYZÉK

INDEX