tanulmányok

170/1985

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest

Magyar Tudományos Akadémia
Számitástechnikai és Automatizálási Kutató Intézete

ALGORITMUS
MÁTRIX ALAPÚ LOGARITMUS KISZÁMÍTÁSÁRA
KRIPTOGRÁFIAI ALKALMAZÁSOKKAL

Irta:

MÁRKUS GÁBOR

Tanulmányok 170/1985

A kiadásért felelős:

Dr. VÁMOS TIBOR

Főosztályvezető:

SOMLÓ JÁNOS

ISBN 963 311 191 9
ISSN 0324 - 2951

TARTALOM

KÖSZÖNETNYILVÁNITÁS

Szeretnék köszönetet mondani Dr. Seitz Károlynak, aki figyelmemet a témára irányitotta és munkámat észrevételeivel, megjegyzéseivel mindvégig nagy mértékben segitette.

BEVEZETÉS

"Nem titok az, akit sok ember tud"

(Gróf Zrinyi Miklós)

ALAPFOGALMAK

A *kriptográfia* olyan módszerekkel foglalkozik, melyek segitségével *kriptogrammák*at állitanak elő, azaz olyan üzeneteket, amelyeknek értelmét lehetőleg csak a jogosult cimzett képes megfejteni. Egy kriptográfiai rendszer vagy *kriptorendszer* tervezője a *kriptográfus* (sifrirozó, desifrirozó). Egy kriptorendszer "ellenfele" a *kriptoanalitikus* (rejtjelfejtő, intercettáns). A *kriptoanalizis* olyan eljárásokkal foglalkozik, amelyeket a kriptogrammák rejtjelkulcs nélküli megfejtésére, ill. az ismeretlen rejtjelkulcs meghatározására használnak.

A *kriptológia* a bizalmas kommunikáció tudománya; magában foglalja a kriptográfiát és a kriptoanalizist egyaránt.

A kriptográfia lényege olyan módszerek kidolgozása, amelyekkel az *üzenet*ből kriptogrammát, *rejtjelezett üzenet*et állitanak elő. Célja kétféle lehet ([1] , [2]): Annak megakadályozása, hogy egy arra illetéktelen személy a nyilvános csatornán át továbbitott üzenetet megértse; és az azonositás (ez utóbbi különösen fontos az üzleti életben, a számitógéprendszerekben). Az üzenet a következő két eljárás egyikével védhető: egy kódrendszer alkalmazásával *kódolhatjuk*, vagy egy rejtjelező rendszer alkalmazásával *rejtjelezhetjük*.

A *kódrendszer* egy olyan kódjegyzéken vagy szótáron alapul, amelyben a szókészlet szavai, kifejezései, mondatai kölcsönösen egyértelmüen meg vannak feleltetve egy kódcsoportnak. Igy a kódolható üzenetek száma a kódjegyzékben szereplő kifejezések kombinációinak számától függ. Bár ez a szám nagy lehet, nem lehet minden kombináció kódolt üzenet. Ezért a kódrendszerek kevéssé alkalmazhatóak, különösen számitógépek felhasználásakor. (1.Ábra)

A *rejtjelrendszer* egy algoritmus segitségével az üzenet egyes karaktereinek egy-egy keraktert feleltet meg a kriptogrammában, és viszont. E müveletnél felhasznál egy *rejtjelkulcsot* (röviden *kulcsot*), amelyet a lehetséges kulcsok halmazából választ ki. Ez a halmaz sok elemet tartalmaz és csak a felhasználók ismerik. (Bővebb információ található [1]-[4]-ben.)

TÖRTÉNETI ÁTTEKINTÉS

A titkos kommunikáció már a nyugati civilizáció kezdetétől ismeretes. (A Kelet kulturáinak titkosirására csak szórványos utalások ismertek [6].) Ismeretes, hogy amikor Dáriusz perzsa király Görögországot készült megtámadni, egy Perzsiában élő görög egy üzenetet faragott egy fa irótáblába és bevonta viasszal, ugyhogy olyan volt, mint egy uj irófelület. Elküldte Spártába, ahol Leonidasz spártai király felesége, Goro rájött, hogy az üres viasz irófelület valami fontosat rejteget, lekaparta a viaszt és felfedezte az üzenetet. Igy Görögország fel tudott ké-

-52666 *C*....Ship's papers
-00547 *C*....Ship ready for
-07197 *C*....Ship Rock

A 28810 *A*
B 07827 *A* } Ship shoned
C 11096 *A*

-53316 *B*....Ship should be
-30764 *A*....Ship's Steward-*s-from-of*
-07488 *A*....Ship still afloat
-00982 *C*....Ship to ship

A 35503 *B*
B 27649 *C* } Ship was
C 11028 *B*

-51867 *B*....Ship will

-07482 *C*...Pushed
-07483 *A*...Your 501
-07484 *B*...Tajura
-07486 *A*...Your 251
A 07487 *B*...Rear Admiral, Egypt
B 07487 *A*...Received-*at-on*
C 07487 *C*...Lamlash
-07488 *A*...Ship still afloat
-07489 *B*...When ready to
-07491 *B*...*Spanish* Mac-Mahon
A 07492 *A*...Submit I may be
B 07492 *C*...These
C 07492 *B*...She

1. Ábra. A Brit Haditengerészet Kódjegyzékének részlete az I. világháborubói. Számcsoportokat sorol fel azon kifejezések mellett, amelyeket helyettesitenek. Az egyes cimkarakterek alfabetikus sorrendben vannak felsorolva kódoláshoz és növekvő számsorrendben dekódoláshoz. A gyakori szavaknak és kifejezéseknek több helyettesitőjük van, hogy ezáltal növeljék a megbizhatóságot.

szülni Dáriusz támadására és le is győzték a perzsákat. Az ókori görögök már az i.e.VII.században alkalmaztak egy henger alaku botra tekert bőrszalagot, amelyre a henger tengelyének irányában irták rá az üzenetet. Ezt az ugynevezett szkütalét csak az tudta elolvasni, akinek ugyanolyan vastag és ugyanolyan hosszu botja volt. Sok hasonló történet és egyszerü módszer ismert ([4]-[6]).

Az ilyen módszereket, amelyek a titkos üzenet elrejtésére irányulnak *szteganográfiának* nevezik. A szteganográfia más formái például: a vegytinta; az olyan rádiórendszerek, amelyek hosszu üzeneteket adnak le rendkivül rövid idő alatt; a "virágnyelv", amelyben az "ártatlan" szavaknak titkos jelentésük van; az olyan rendszerek, amelyekben csak bizonyos betük vagy szavak alkotják a bizalmas üzenetet, a többi csupán azért szerepel, hogy együtt egy "semleges" szöveget alkotva elrejtse az üzenetet. Ilyen technikák alkalmazására különösen az ókorban és a középkorban és később az arisztokrácia körében találunk sok példát. A sztaganográfia abban különbözik a kriptográfiától, hogy ez utóbbi esetben nem próbáljuk meg elrejteni a kódolt vagy rejtjelezett üzenetet.

A tényleges kriptográfiai módszerek alkalmazása az ókori Rómába nyulik vissza. Legegyszerübb formáját, az *egyszerü helyettesitést* Iulius Caesar alkalmazta először. (Ő az ábécé minden betüjét a rákövetkező harmadik betüvel helyettesitette ciklikusan.)

Egy ezzel rokon eljárás a *transzpozició*, amikor nem az ábécé betüit, hanem az üzenet betüit permutáljuk.

Az igy készült kriptogrammák azonban viszonylag könnyen megfejthetők betü-, betükettős-, betühármas-, szó-, szókapcsolatgyakoriségi vizsgálatok alapján.

1466 körül egy olasz épitész, Leo Battista Alberti egy uj kriptográfiai eljárást dolgozott ki, amelyben több különböző helyettesitő ábécét használt periódikusan az üzenet rejtjelezésére. Ezt *több ábécés módszer*nek nevezik. Ezt az eljárást fejlesztette tovább egy Johannes Trithemius nevü benedekrendi abbé, megalkotva a *haladó módszer*nek nevezett eljárást. Ebben az ábécé minden betüjére egy másik helyettesitő ábécét alkalmazott és végighaladt rendre az ábécé összes betüjén. 1553-ban Giovanni Battista Belasco közzétett egy egyszerü és megbizható eljárást ilyen, használható ábécék előállitására.

Itt megjegyezzük, hogy kb. a XX. század közepéig a kriptográfiát elsősorban diplomáciai és katonai érintkezések során alkalmazták. A kriptográfia alkalmazásának nyomait vallásos tárgyu müvekben felfedezhetjük már a bibliai időkben is (pl. Isten – Jahve – nevét nem volt szabad kimondani és leirni, ezért más szóval helyettesitették: YHWH helyett Adonáj-t (Ur), Elohim-ot (Istenség) vagy három pontot irtak, illetve az ATBAS kulcsra épülő helyettesitő titkosirást alkalmazták [6]), de a vallásos müvekben való alkalmazás fénykora kétségkivül a középkor (rózsakeresztesek; szövegmagyarázatok tényleges és vélt helyettesitések és transzpoziciók alapján).

A több ábécés rendszer mintegy 300 évig lényegében véve biztonságot nyujtott. Azonban Friedrich Kasiski, egy német nyugalmazott gyalogsági őrnagy felfedezett és 1863-

ban közzétett egy olyan általános módszert, amely megfejtette a több ábécés rejtjelezéssel készült kriptogrammákat, ha a rejtjelezés során ismételt kulcsot használtak.

Ez arra ösztönözte a kriptográfusokat, hogy egy un. *futó kulcsos rendszert* alkalmazzanak, azaz egy aperiódikus több ábécés módszert, ahol egyetlen kulcs sem ismétlődik. Azonban 1883-ban egy francia nyelvtanár, Auguste Kerckhoffs általános eljárást adott a több ábécés rendszerek megoldására.

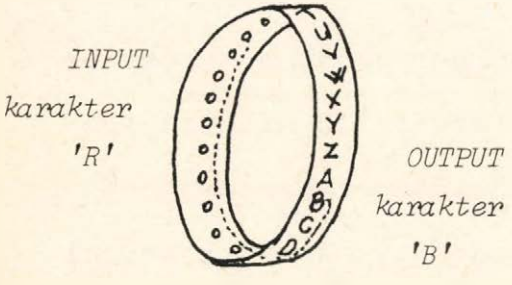
A futó kulcsos rendszereket vizsgálta 1914-ben Parker Hitt kapitány és 1917-ben William F. Friedman. Munkáikból az a következtetés vonható le, hogy csak az olyan rejtjelrendszer megtörhetetlen, amelynek kulcsa sohasem ismétlődik és amely kulcsának nincs értelme, sőt semmiféle szabályosság sincs benne. E következtetésre Joseph O. Manborgne hadnagy jutott 1918-ban. Ekkor találta fel Gilbert S. Vernam, aki az American Telephone and Telegraph Company-nél dolgozott, az első rejtjelező eszközt, amely már automatikusan rejtjelezte és továbbitotta az üzeneteket.

Ez a megtörhetetlen "egyszeri módszer" azonban mégsem használatos általánosan, mivel alkalmazásához nagyon sok kulcsra és elég sok időre van szükség. De bizonyos területeken, a diplomáciai és titkosügynökségi kommunikációban fontos szerepet játszik. Elsőként Németország vezetett be egy ilyen rendszert az 1920-as évek elején. A Szovjetunió az 1930-as években kezdte alkalmazni diplomáciai kommunikációk lebonyolitásásra. Ezenkivül használják többek között az USA külügyminisztériumában, az ENSZ-

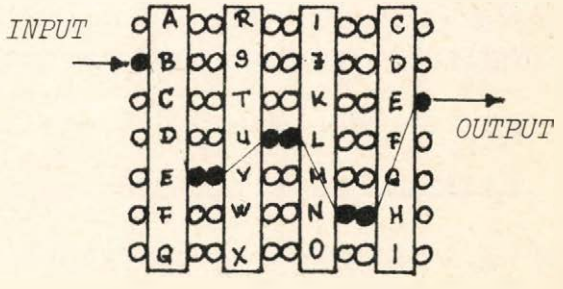
ben és a Nemzetközi Pénzalapnál. A Washingtont Moszkvával összekötő "forró drót"-on küldött üzenetváltásoknál is Vernam-féle mechanizmust használnak, egyszeri szalaggal.

A *rotor gép* volt a II. világháboru legfontosabb kriptográfiai eszköze, és jelentős maradt még az ötvenes évek végén is. Ezt elsőként az amerikai Edward H. Hebern találta fel 1917-ben, tőle és egymástól függetlenül néhány éven belül szintén feltalálta a holland Hugo Koch, a svéd Arvid G, Damn és a német Arthur Scherbius.

A rotor gép fő alkotóeleme a rotor, vagyis egy huzalozott tárcsa, amely a rejtjel-ábécé implementáslására szolgál (2.Ábra).



2. Ábra Rotor



3. Ábra

Ha az ábécé betüinek száma *M*, akkor a tárcsa két oldalán *M* pár érintkező van: egy betühöz egy érintkezőpár. Minden első érintkező össze van kötve egy második (hátsó) érintkezővel huzal segitségével. Igy a tárcsa egy karaktert reprezentáló elektromos jelet permutál, amig az a rotor első részéből a hátsó, második részére ér. A rotor gép több rotorból áll, amelyek ugy vannak összeillesztve, hogy egy mechanizmus képes legyen az egyes rotorokat az

inputkarakterek változásának megfelelően forgatni (3.Ábra). Tehát a rotor gép egy időfüggő helyettesitő rejtjelező eszköz [7].

Az egyszeri rendszerek és a rotor gépek mellett egy harmadik, egyszerübb és olcsóbb, de kevésbé megbizható több ábécés rendszer is nagyon elterjedt. Ezt 1934-ben találta fel Boris C. W. Hagelin svéd gépészmérnök. A *Hagelin-gép* egy olyan eszköz, amely változó számu foggal van ellátva és ezekkel mozgatja a rejtjelező ábécét változó számu helyre. Hagelin gépe a variációkat oly módon generálja, hogy 2.8 milliárd betüt rejtjelez ismétlődés nélkül.

Mint már emlitettük, kb. a XX. század közepéig a kriptográfiát elsősorban a diplomáciai , katonai kommunikációban használták. Az ötvenes években azonban ugrásszerü fejlődés kezdődött az analóg és digitális elektrotechnikában. Ez a fejlődés egyrészt lehetőséget biztositott uj kriptográfiai eljárások kidolgozására, másrészt sokkal nagyobb követelményeket támasztott a kriptográfiával szemben. Ez részben annak a következménye volt, hogy a számitógépek rohamos elterjedésével előtérbe került a kriptográfia üzleti életben való alkalmazása [8]-[9](time-sharing számitógéprendszerek, adatbankok, elektronikus postai szolgáltatások, stb.).

Az egyik fejlődési irány az *analóg rendszerek* kifejlesztése volt. Ezeket elsősorban a beszédhang rejtjelezésére használják. A legegyszerübb analóg keverők a *frekvenciafordítók*, amelyek a hangjel spektrumát csak átforditják egy transzponáló operációval.

Összetettebb módszer a *sávcsusztatás*. Ennek során a beszédet rendszerint öt frekvencia tartományra osztják és ezeket permutálják.

A *gördülő kód keverők* másodpercenként többször változtatják a permutációt és ugyanakkor az öt sáv egy változó részhalmazában frekvenciafordítást is végrehajtanak.

Az *időfelosztó keverők* a beszédet rövid darabokra vágják és ezeket permutálják egymás között, hasonlóan a transzpoziciós rejtjelező készülékhez. Az idő- és frekvencia-tartományok egyidejü permutálása is lehetséges. Ilyenkor *kétdimenziós keverő*ről beszélünk.

A másik fejlődési irány a *digitális rendszerek* kialakitása. A legleterjedtebb technika a *shift-register*ek alkalmazása ([10]-[13]). Itt a kulcs egy teljesen véletlen bitstring, amelyet modulo 2 összeadunk az üzenettel, amelyet szintén bináris formában reprezentálunk, és igy kapjuk a rejtjelezett üzenetet. A kulcsnak ugyanolyan hosszunak kell lennie, mint amilyen hosszu az üzenet, és nem használható fel több alkalommal. Azért, hogy rövidebb kulcsokat alkalmazhassanak, a felhasználók gyakran folyamodnak pszeudovéletlen-szám generátorokhoz, nevezetesen feedback shift-register-ekhez, hogy egy rövid kulcsból hosszu, nem ismétlődő, véletlenszerü bináris sorozatokat állitsanak elő.

A számitógépek elterjedése és a számitógépeken alkalmazott kriptorendszerek hétköznapivá válása sok uj problémát is felvetett. Egyik ilyen probléma annak az igénye, hogy a régebben rejtjelezett üzeneteket, adatokat később nyilvánosságra lehessen hozni. Először 1960 körül álli-

tottak müködésbe egy olyan kriptorendszert, amely feltételezhetően elég erős ahhoz, hogy a régi, rejtjelezett adatok nyilvánosságra hozása sem rontja jelentősen a rendszer megbizhatóságát. Egy másik uj követelményt szült az egy adott rendszert használó felhasználók számnak növekedése. Sok felhasználó esetén ugyanis nehezen valósitható meg a kulcsok titokban tartása és igy olyan kriptorendszerekre van szükség, amelyekben a kulcsot nyilvánosságra lehet hozni (*nyilvános jelkulcsu kriptorendszerek*) [14]-[16], vagy amelyekben a használt kulcsot nyilvánosan lehet a jelhasználók között elosztani (*nyilvános kulcselosztó rendszer*) [1]-[2]. Külön figyelmet érdemel az *adatbankok*kal kapcsolatban a *felhasználó- és adataz onositás* ([1]-[2], [10]-[11], [17]-[21]), valamint a többek között az *elektronikus postai szolgáltatások*kal [22] kapcsolatban felmerülő *digitális aláirás* problémája ([1]-[2], [14], [23]-[26]). E modern kriptográfiai eljárások felvetik a kriptográfia és a komplexitáselmélet kapcsolatának problémáját is [27].

Végezetül annak illusztrálására, hogy feltétlenül szükséges a számitógéphálózatok minél tökéletesebb védelme (fizikai és kriptográfiai uton egyaránt), álljon itt néhány adat ([28]-[29], [46]-[47]).

Bár a számitógépes bünözés mértékéről pontos adatok nem állnak rendelkezésre (Mike Comer, a Computer Fraud and Security Bulletin kiadójának véleménye szerint [47] a számitógéppel elkövetett büncselekményeknek még 20 %-ára sem derül fény), az FBI vizsgálata szerint az egy ilyen büneset okozta pénzügyi veszteség átlagosan 7 000 $

volt a hatvanas évek végén, ez a hetvenes évek végére 19 000 $-ra emelkedett. Más tanulmányok szerint 1962 és 1975 között a számitógéppel végrehajtott bankcsalás és sikkasztás egy esetre jutó átlagos vesztesége 430 000 $ volt. Ez összesen 18 millió dollárt jelentett, magában foglalva a 200 $-tól a 6.8 millió $-ig terjedő visszaéléseket. Amerikai felmérések [46] arról számolnak be, hogy az USA-ban az 1976-os veszteség 100 millió $ volt, évi 400 %-os emelkedés mellett.

Az 1. Táblázat a kivizsgált számitógépes bünözések megoszlását mutatja az USA-ban.

Év	Pénzügyi csalás	Információ vagy tulajdon ellopása	Nem-azonositott felhasználás	Vandalizmus	Összesen
1969	3	6	0	3	12
1970	7	5	9	8	29
1971	22	18	6	6	52
1972	12	15	16	12	55
1973	21	15	8	9	53

1. Táblázat

A fejlett számitástechnikával rendelkező országokban a számitógépes bünesetek száma rendkivül magas. Például az utóbbi esztendőben az USA-ban 472, Svédországban 35, az Egyesült Királyságban 23, az NSZK-ban 21, Olaszországban 19 számitógépes bünesetet vettek nyilvántartásba (Ötlet, 1982 julius; 15.old.).

A dolgozat 1.RÉSZében egy olyan algoritmust ismertetünk, amely lehetővé teszi bizonyos gyürü feletti mátrix alapu logaritmus meghatározását. A 2.RÉSZben ezen algoritmus alkalmazási lehetőségeit tárgyaljuk a kriptográfia területén: leirunk egy hagyományos kriptorendszert, amely elég erős ahhoz, hogy a régebben rejtjelezett adatokat nyilvánosságra hozzuk, leirunk továbbá egy nyilvános kulcselosztó rendszert és két felhasználó-azonositó eljárást.

1. RÉSZ:

ALGORITMUS-KONSTRUKCIÓ

1.1. ELÖKESZITÖ TÉTELEK, DEFINICIÓK

Jelöljön R egy egységelemes gyürüt (gyürün asszociativ, de nem feltétlenül kommutativ gyürüt értünk). Legyen n egy rögzitett pozitiv egész szám és jelölje $\mathbb{M}$ az R gyürü feletti $n\times n$-es olyan mátrixok halmazát, amelyekmek egy pozitiv kitevős hatványa az I R feletti $n\times n$-es egységmátrix. Nyilvánvalóan $I \in \mathbb{M}$ és $\mathbb{M}\setminus\{I\} \neq \emptyset$ (ha $n \geq 2$), ezenkivül $\mathbb{M}$ zárt a mátrixok inverzének képzésére.
A dolgozat 1.RÉSZében az

$$\begin{aligned} Y &= A^x \quad R \text{ felett} \\ x &= \log_A Y \quad R \text{ felett} \end{aligned} \tag{1}$$

inverz függvénypárt vizsgáljuk, amelyeket R feletti mátrixhatványozásnak, illetve R feletti mátrix alapu logaritmusnak nevezünk, ahol $A \in \mathbb{M}$ tetszőleges, az I egységmátrixtól különböző mátrix.

A továbbiakban "müvelet"-en egy R-beli összeadást/kivonást/szorzást (esetenként modulo h vett müveletet, ahol h egy pozitiv egész szám) értünk, "M-müvelet"-en R feletti $n\times n$-es mátrixok közötti összeadást/kivonást/szorzást értünk.

Ismeretes [30], hogy a mod p vett hatványozás elvégezhető legfeljebb $2\lceil \log_2 p \rceil$ mod p vett szorzással és három, egyenként $\lceil \log_2 p \rceil$ bites memória-szóval, ahol $\lceil \cdot \rceil$ azt a legkisebb egész számot jelöli, amely nagyobb vagy egyenlő a zárójelben levőnél. Az algoritmus a kitvő bi-

náris felbontására épül:

$$\alpha^{18} = (((\alpha^2)^2)^2)^2 \cdot \alpha^2 \qquad (2)$$

Hasonló igaz az R gyürü feletti $n \times n$-es mátrixokra is. Ha R egy elemének a tárolásához b bitre van szükség, akkor fennáll a következő tétel:

1.TÉTEL: Legyen A egy R feletti $n \times n$-es mátrix, ahol R egy egységelemes gyürü. Ekkor A hatványozása legfeljebb $2\lceil \log_2 h \rceil$ R feletti mátrixszorzással elvégezhető, ahol h az A mátrix különböző hatványainak a száma, illetve ha ez nem ismert, akkor a kitevő; és ehhez kettő, egyenként $n^2 b$ bites és egy $\lceil \log_2 h \rceil$ bites memóriaszóra van szükség.

Megjegyezzük, hogy ha $R = GF(q)$ valamilyen q primhatványra, akkor $b = \lceil \log_2 q \rceil$.

A mátrixszorzás müveletigényére vonatkozik a következő, V. Strassentől származó eredmény [31]:

Két $n \times n$-es tetszőleges gyürü feletti mátrix

$$O(n^{\log_2 7}) \approx O(n^{2.807}) \qquad (3)$$

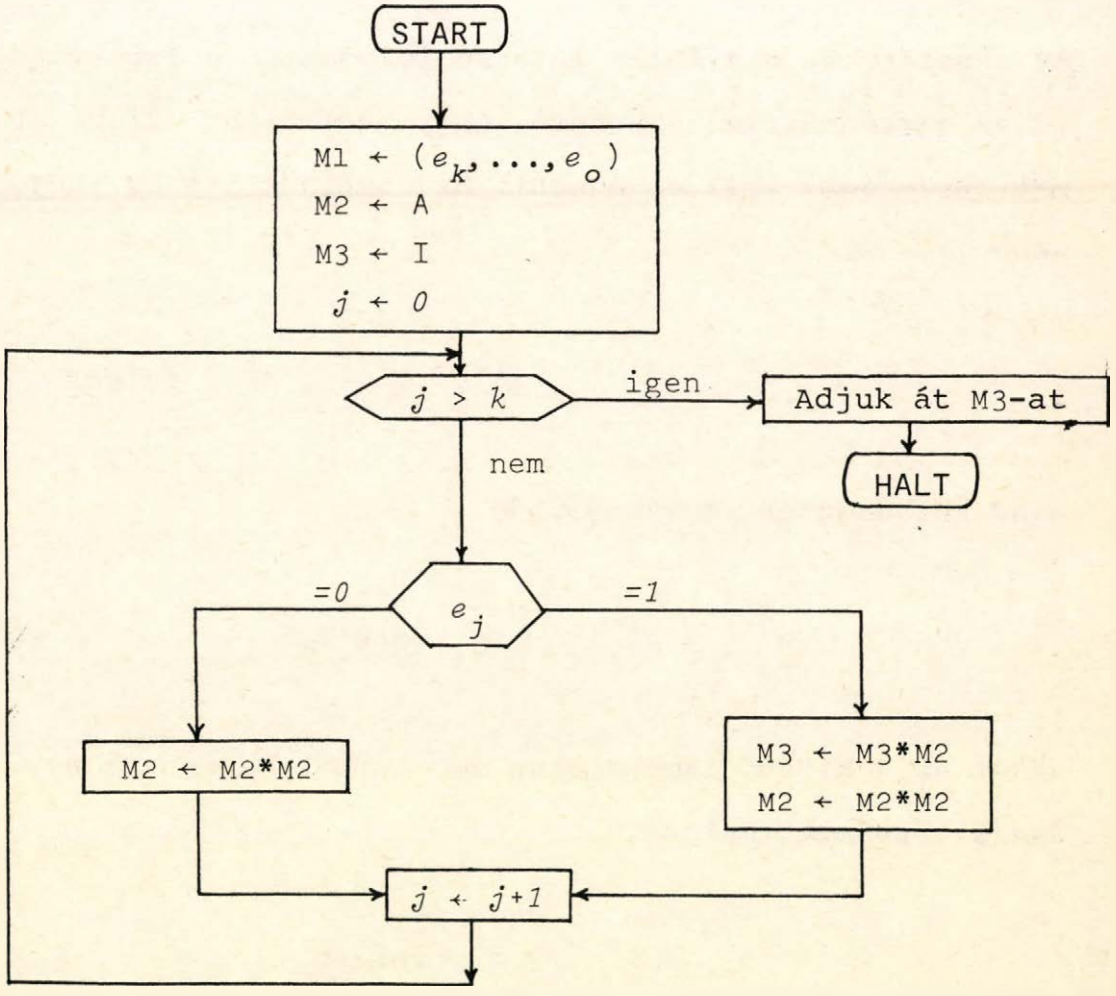
gyürübeli aritmetikai müvelettel összeszorozható.

Pan [32] ezt az eredményt

$$O(n^{\log_{48} 47210}) \approx O(n^{2.780}) \qquad (4)$$

müveletre javitotta.

Ezek a módszerek a müveletigény csökkentésével egyidejüleg a memóriaszükségletet jelentősen megnövelik (ez a növekmény függ a konkrét módszertől). A továbbiakban a "mátrixszorzás"-ba mindig beleértjük azt a módszertől függő plusz memóriaigényt is, ami a két mátrix tárolásán felül a müvelet elvégzéséhez szükséges.



4. Ábra

BIZONYITÁS: Legyen a kitevő bináris előállitása:

$$e = e_k \cdots e_o \,. \tag{5}$$

Ekkor az A R feletti $n \times n$-es mátrixra

$$A^e = (A^{e_k 2^k})(A^{e_{k-1} 2^{k-1}}) \cdots (A^{e_1 2})(A^{e_o}) \,. \tag{6}$$

Az algoritmust a 4.Ábrán látható folyamatábra szemlélteti (I az egységmátrix). Látható, hogy legfeljebb $2\lceil \log_2 e \rceil$ mátrixszorzást kell elvégezni. Ha ismerjük h-t és tudjuk, hogy

$$A \,,\; A^2 \,,\; \ldots \,,\; A^i \,,\; \ldots \,,\; A^h \quad \text{R felett} \tag{7}$$

mind különböznek egymástól, de

$$A^{h+1} = A^{i+1} \quad \text{R felett,} \tag{8}$$

akkor az e kitevő ismeretében meg tudunk határozni egy $0<e' \leq h$ számot, melyre

$$A^{e'} = A^e \quad \text{R felett.} \tag{9}$$

Ha $e \leq h$, akkor $e' \leftarrow e$.

Ha $e > h$, akkor $e' \leftarrow i+d+1$, ahol

$0 \leq d$ és $d \equiv e - h - 1 \pmod{(h-i)}$.

Q.E.D.

Az inverz problémára, a logaritmus meghatározására az eddig publikált két legjobb algoritmus mod p (p prim) vett logaritmus kiszámitásával foglalkozik: Knuth [33] algoritmusa $O(\sqrt{p})$ komplexitásu memóriában és időben egyaránt; Pohling és Hellman [34] algoritmusa az előbbinek javitott és $GF(q)$-ra (q primhatvány) általánositott változata, és ha $(p-1)$ csak kicsi primfaktorokkal rendelkezik ($GF(p)$ esetén), akkor komplexitása $O(\log_2 p)$. Az 1.2. Fejezetben leirt algoritmus ez utóbbinak az általánositása.

Az R feletti mátrixhatványozás lehet egyirányu függvény. Az f függvényt *egyirányu függvény*nek nevezzük, ha minden, az f értelmezési tartományába eső x argumentumra könnyen ki tudjuk számitani az $f(x)$ értéket, de majdnem minden, az f értékkészletébe tartozó y-ra számitástechnikailag megoldhatatlan az $y=f(x)$ egyenlet x ismeretlenének meghatározása. Fontos megjegyezni, hogy a most definiált egyirányu függvény *számitástechnikai* szempontból nem invertálható, de ez a nem-invertálhatóság alapvetően különbözik a matematikában általában használt fogalomtól, mivel itt nem azt követeljük meg, hogy egy y érték inverz képe ne legyen egyértelmü, hanem azt, hogy y és f ismeretében rendkivül nehéz legyen egy $f(x)=y$ tulajdonsággal rendelkező x argumentumot találni.

Az előző definicióban használtuk a "számitástechnikailag megoldhatatlan" és a "könnyü" kifejezéseket, amelyeket szándékosan nem definiáltunk precizebben, minthogy e fogalmak időtől és technikától függően változnak. E probléma kiküszöbölhető lenne, ha a "könnyü" jelenleg elfoga-

dott és a "megoldhatatlan" fizikailag korlátozott definicióját használnánk. Bármely ma könnyünek tartott számitás nem lesz nehezebb a jövõben sem, és egy 10^{60} bites memória sohasem lesz elérhető, mivel annak elõállitása több anyagot igényelne, mint amennyi a Naprendszerben van, még akkor is, ha egy molekulát használnánk fel egy bit memóriához. A termodinamika a végrehajtható müveletek számát közelitõleg 10^{70}-ben korlátozza, még akkor is, ha a Nap összes, létezése során kibocsátott energiáját fel tudnánk használni [35],[36]. Azonban nem célszerü az ilyen konzervativ definició, mivel ez gyakorlatilag használható egyirányu függvényeket zárhat ki. De majd láthatjuk, hogy az R feletti mátrixok hatványozása általában még az egyirányu függvény ilyen konzervativ definiciójának is eleget tesz.

Az egyirányu függvényeket elsősorban a time-sharing számitógéprendszerekben a password-file védelmére használják [17], [18], [37]; ezzel rokon más felhasználási területeik is vannak [1]-[2], [15]. Fontosak továbbá a megbizható kriptorendszerek létezéséhez is, mivel minden megbizható kriptorendszerből elõállitható egy egyirányu függvény [1]; ezen állitás megforditása általában nem igaz.

Az 1.2.Fejezet egy R feletti mátrix alapu logaritmust kiszámitó algoritmus részletes leirását tartalmazza, az 1.3.Fejezetben pedig ezen algoritmus hatékonyságát, alkalmazhatóságát vizsgáljuk.

1.2. ALGORITMUS MÁTRIX ALAPU LOGARITMUS KISZÁMITÁSÁRA

E fejezetben jelöljön R továbbra is egy egységelemes gyürüt, n egy pozitiv egész számot, I az R feletti $n\times n$-es egységmátrixot és A egy olyan R feletti $n\times n$-es (invertálható) mátrixot, amelyre teljesül:

$$I=: A^0 , A , A^2 , \ldots , A^{h-1} , A^h = I \quad R \text{ felett} \tag{10}$$

és

$$A^s \neq I \quad R \text{ felett, ha} \quad 1 \leq s < h . \tag{11}$$

Továbbá jelölje $\iota(n)$ az R gyürü feletti $n\times n$-es invetálható mátrixok kiszámitásához szükséges M-müveletek számát (erre vonatkozó eredmények és eljárások találhatók pl. [31]-[32]-ben), és b az R gyürü egy elemének tárolásához szükséges bitek számát.

Tételezzük fel, hogy ismerjük A-t, h-t és egy Y mátrixot, amelyről tudjuk, hogy

$$Y = A^x \quad R \text{ felett} \tag{12}$$

valamilyen x pozitiv egész számra. x nyilvánvalóan csak modulo h egyértelmü. Keressük a

$$0 \leq x \leq h-1 \tag{13}$$

számot, azaz határozzuk meg a következő mátrix alapu loga-

ritmust:

$$x = \log_A Y \qquad R \text{ felett, modulo } h. \quad (14)$$

Ennek természetesen csak $h>1$ esetén van értelme (különben $A = I$ R felett, és x akármilyen nemnegativ egész lehet, de az $x \leq h-1$ megszoritás $x=0$-t ad). Igy

$$A \neq I \qquad R \text{ felett.} \quad (15)$$

Ezt a továbbiakban feltesszük, továbbá feltételezzük, hogy az A-val és Y-nal jelölt mátrixok rendelkeznek a fenti tulajdonságokkal.

Hogy az algoritmus érthetőbb legyen, vizsgáljuk először a

$$h = 2^N \quad (16)$$

esetet, ahol h az előbb definiált minimális kitevő, N egy pozitiv egész szám. Tekintsük x-nek $(0 \leq x \leq h-1)$ a következő bináris felbontását:

$$x = \sum_{i=0}^{N-1} b_i 2^i. \quad (17)$$

Az A mátrixra tett feltevések miatt $h/2 = 2^{N-1}$-re

$$B := A^{h/2} \qquad R \text{ felett,} \quad (18)$$

ahol

$$B^2 = I \quad R \text{ felett}, \quad \text{de} \quad B \neq I \quad R \text{ felett}. \tag{19}$$

Igy

$$Y^{h/2} = (A^x)^{h/2} = (A^{h/2})^x = B^x =$$

$$= B^{2\sum_{i=1}^{N-1} b_i 2^{i-1} + b_o} = B^{b_o} \qquad R \text{ felett}. \tag{20}$$

Ezért ha

$$Y^{h/2} \quad R \text{ felett} \begin{cases} = I, \text{ akkor} & b_o = 0 . \\ \neq I, \text{ akkor} & b_o = 1 . \end{cases} \tag{21}$$

x előállitásában a következő jegyet a

$$Z = YA^{-b_o} = A^{x_1} \qquad R \text{ felett} \tag{22}$$

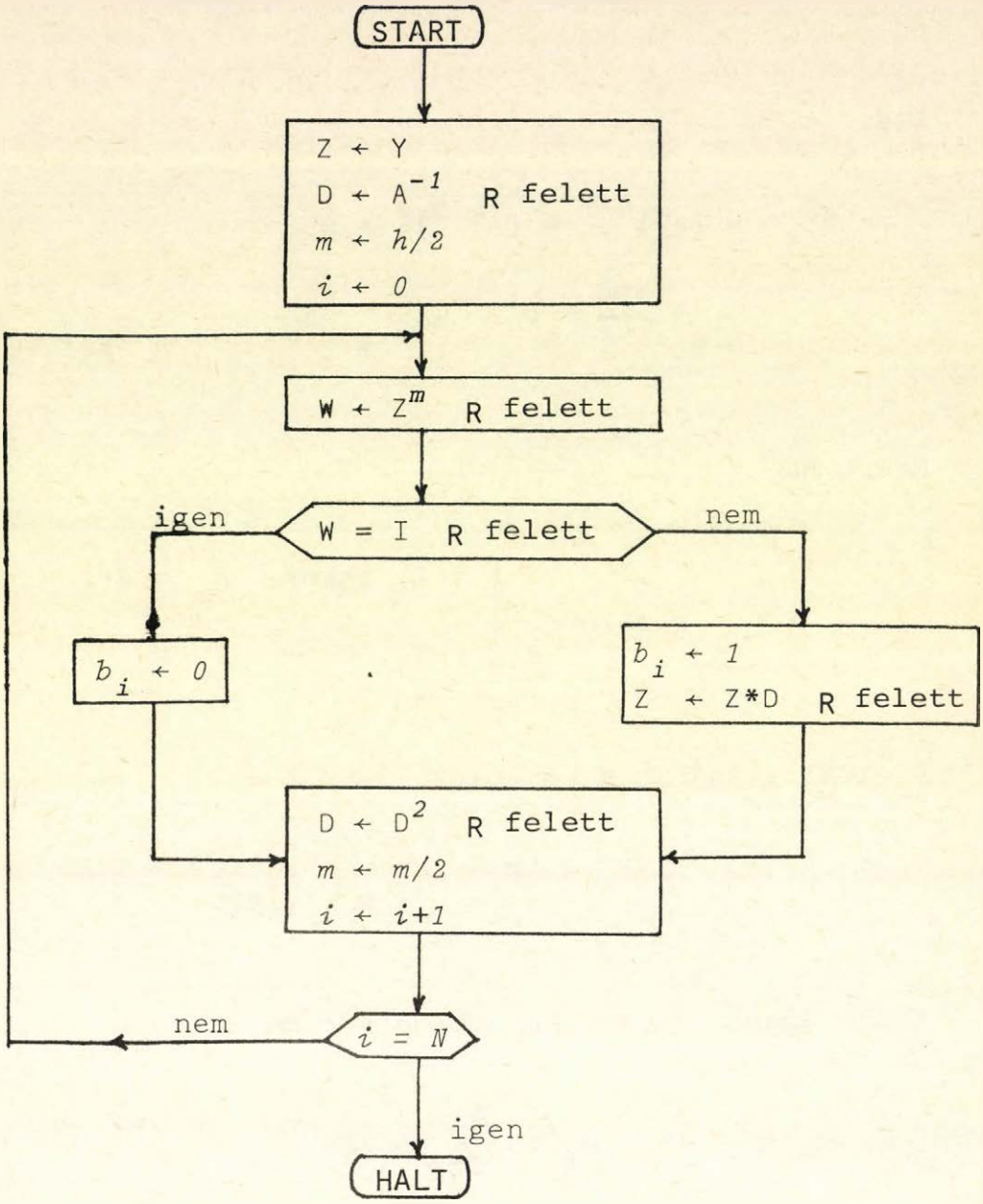
összefüggés alapján határozhatjuk meg, ahol

$$x_1 = \sum_{i=1}^{N-1} b_i 2^i \quad . \tag{23}$$

Nyilvánvaló, hogy x_1 pontosan akkor osztható néggyel, ha $b_1 = 0$. Ha $b_1 = 1$, akkor x_1 páros, de néggyel nem osztahó. Az előzőhöz hasonló megfontolással kaphatjuk, hogy ha

$$Z^{h/4} \quad R \text{ felett} \begin{cases} = I , \text{ akkor } b_1 = 0. \\ \neq I , \text{ akkor } b_1 = 1. \end{cases} \tag{24}$$

x többi jegye is hasonlóan határozható meg.



5. Ábra

Az algoritmust az 5.Ábra blokkdiagrammja illusztrálja. A folyamatábrán jól nyomon követhető, hogy az i-edik ciklus kezdetekor

$$m = h/2^{i+1} \quad , \quad Z = A^{x_i} \quad R \text{ felett}, \qquad (25)$$

ahol

$$x_i = \sum_{j=i}^{N-1} b_j 2^j \quad . \tag{26}$$

Igy Z-t m-edik hatványra emelve R felett kapjuk, hogy

$$Z^m = A^{x_i m} = A^{(h/2)(x_i/2^i)} = B^{x_i/2^i} = B^{b_i} \text{ R felett,} \tag{27}$$

és ezért

$$Z^m = I \text{ R felett} \tag{28}$$

pontosan akkor, ha $b_i = 0$ és

$$Z^m \neq I \text{ R felett} \tag{29}$$

(B^2=I miatt) pontosan akkor, ha $b_i = 1$.

Megjegyezzük, hogy $n = 1$, R = GF(p) (p prim) és A primitiv elem modulo p esetén I-nek az 1, B-nek a -1 GF(p)-beli elem felel meg.

Most általánositjuk a fenti algoritmust tetszőleges h értékre.

Legyen h promfaktorizációja

$$h = p_1^{N_1} p_2^{N_2} \cdots p_k^{N_k}, \quad p_i < p_{i+1}, \quad (i=1,\ldots,k-1), \tag{30}$$

ahol p_i-k különböző primek, N_i-k pozitiv egészek $(i=1,\ldots,k)$.

Először határozzuk meg x értékét modulo $p_i^{N_i}$, $i = 1,\ldots,k$; majd ezután ezekből kapjuk x-et modulo h.

Tekintsük x-nek a következő előállitását modulo $p_i^{N_i}$:

$$x = \sum_{j=0}^{N_i-1} b_i p_i^j, \tag{31}$$

ahol $0 \leq b_j \leq p_i - 1$. Az utolsó együttható, b_o ugy határozható meg, hogy Y-t h/p_i-edik hatványra emeljük R felett:

$$Y^{h/p_i} = A^{hx/p_i} = C_i^x = C_i^{p_i \sum_{j=1}^{N_i-1} b_j p_i^{j-1} + b_o} =$$

$$= C_i^{b_o} \quad R \text{ felett}, \tag{32}$$

ahol

$$C_i = A^{h/p_i} \quad R \text{ felett} \tag{33}$$

és az A-ra tett feltevések miatt

$$C_i \neq I \quad R \text{ felett}, \tag{34}$$

de pontosan egy olyan $2 \leq s \leq p_i$ egész s-re teljesül, hogy

$$C_i^s = I \quad R \text{ felett} \tag{35}$$

és ugyanakkor

$$C_i^{\ell} \neq I \quad R \text{ felett}, \quad \ell = 1, 2, \ldots s-1 \ . \tag{36}$$

Igy ha

$$Y^{h/p_i} \quad R \text{ felett} \begin{cases} = I \ , \text{ akkor } \ b_o = 0 \ . \\ = C_i \neq I \ \text{ és ha a } C_i\text{-hez az előző tulajdonságu } s \text{ tartozik, akkor } \ b_o = s-1 \ . \end{cases} \tag{37}$$

x-nek a vizsgált előállitásban szereplő következő jegye, a b_1 a $h=2^N$ esetben tárgyaltakhoz hasonlóan határozható meg. Legyen

$$Z = YA^{-b_o} = A^{x_1} \quad R \text{ felett}, \tag{38}$$

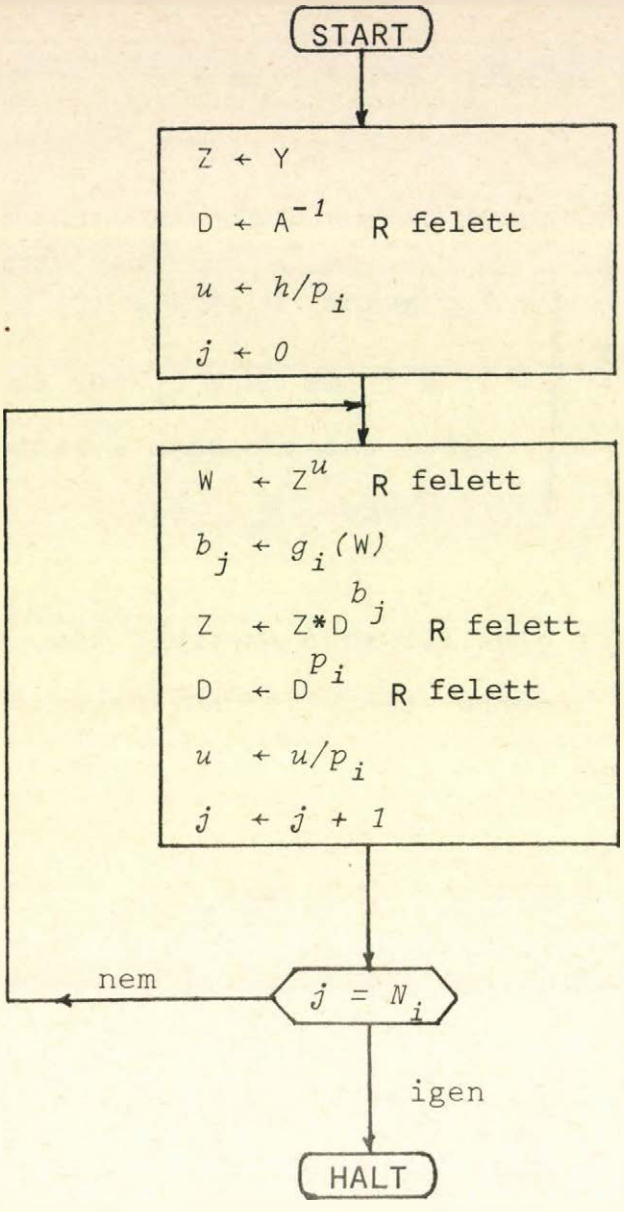
ahol

$$x_1 = \sum_{j=1}^{N_i-1} b_j p_i^{\,j} \ . \tag{39}$$

Most Z-t a $h/p_i^{\,2}$ kitevõre emelve R felett

$$Z^{h/p_i^{\,2}} = A^{hx/p_i^{\,2}} = C_i^{\,x_1/p_i} = C_i^{\,b_1} \quad R \text{ felett} \tag{40}$$

adódik. Ekkor $Z^{h/p_i^{\,2}}$ értékei ismét p_i csoportba oszthatók annak megfelelően, hogy hányadik hatványuk lesz I R felett, és ezek meghatározzák b_1 értékét. Ezt az eljárást folytatva minden b_j-t $(j=1, \ldots, N_i-1)$ előállithatunk.



6. Ábra

A 6.Ábrán látható blokkdiagramm a fenti előállitás együtthatóinak a meghatározását mutatja. (Az A^{-1} mátrixot ezen algoritmusrészlet sorra kerülése előtt kiszámitottuk, igy itt csupán $D \leftarrow A^{-1}$ értékátadásról van szó.) Az ott szereplő $g_i(W)$ függvény a következő:

$$C_i^{g_i(W)} = W \quad R \text{ felett}, \quad 0 \leq g_i(W) \leq p_i - 1 , \qquad (41)$$

ahol C_i a fent definiált mátrix.

Ezt az algoritmust az $i=1,\ldots,k$ esetekben végrehajtva megkapjuk

$$x \pmod{p_i^{N_i}} , \quad i=1,\ldots,k \qquad (42)$$

értékét és ezekből a kinai maradéktétel [31] alapján előállitható

$$x = x \pmod{h} = x \pmod{\prod_{i=1}^{k} p_i^{N_i}} , \qquad (43)$$

mert $0 \leq x \leq h-1$. A kinai maradéktétel implementálható ugy, hogy ez az utolsó lépés $O(k\log_2 h)$ (modulo h vett) aritmetikai müveletet és $O(k\log_2 h)$ memóriabitet igényel.

Ha h primfaktorai kicsik és n is kicsi, akkor a $g_i(W)$ függvényértékeket viszonylag könnyen táblázatba foglalhatjuk és a logaritmus kiszámitása

$$\{O(b) \text{ M-müvelet}\} \times \{O(\log_2 h) \text{ müvelet}\} \qquad (44)$$

müveletet és a $g_i(W)$ értékek táblázata viszonylag kevés memóriát igényel. Ez esetben a domináns müvelet a $W = Z^u$ R felett kiszámitása, amihez $O(b)$ M-müvelet szükséges. A ciklust $\sum_{i=1}^{k} N_i$ alkalommal kell végrehajtani, és mivel a p_i-k kicsik, $\sum_{i=1}^{k} N_i$ közelitőleg

$\log_2 h$, de ezt az értéket sohasem haladja meg. Igy ha h csak kicsi primfaktorokkal rendelkezik és n sem nagy, akkor az R feletti mátrixhatványozás nem egyirányu függvény, mivel egy egyirányu függvény esetében megköveteljük, hogy inverzének kiszámitásához legalább 10^9-szer annyi időre legyen szükség, mint amennyi idő alatt a függvény értékei kiszámithatók [34] , vagy az inverz meghatározásához legalább 2^{100} memóriabit legyen szükséges.

Azonban ha h-nak van nagy primfaktora (pl. $h = p$ vagy $h = 2p_k$) vagy/és n nagy, akkor a $g_i(W)$ kiszámitása fog dominálni, vagy a memóriaigény lesz igen nagy, vagy mindkettő egyszerre. Ezt támasztja alá a következő lemma (a g_i és C_i szimbólumok mellől az i indexeket az egyszerübb irásmód érdekében elhagyjuk):

2.LEMMA: Legyen

$$W = C^g \quad R \text{ felett,} \tag{45}$$

ahol

$$0 \leq g \leq p_i - 1 \qquad \text{és} \qquad C = A^{h/p_i} \tag{46}$$

egy mondott tulajdonságu A mátrixra. Ekkor adott A és W esetén g meghatározható

$$O(p_i^{(1-r)}(1+\log_2 p_i^{\,r})) \tag{47}$$

W-től függő M-művelet és

$$O((3+p_i^r)n^2b) \tag{48}$$

memóriabit felhasználásával, ahol $0 \leq r \leq 1$.

Ha $r < 1/2$, akkor az előkészitő müveletek, amelyek

$$O(p_i^r \log_2 p_i^r + \iota(n)) \tag{49}$$

M-müveletet igényelnek , nem számottevőek, ha a W-től függő M-müveletek számával vetjük össze őket.

BIZONYITÁS: Legyen

$$m = \lceil p_i^r \rceil . \tag{50}$$

Ekkor van olyan c és d egész, hogy

$$g=cm+d \text{ és } 0 \leq c < \lceil p_i/m \rceil \approx p_i^{1-r} \text{ és } 0 \leq d < m \approx p_i^r. \tag{51}$$

A

$$W = C^g \quad R \text{ felett} \tag{52}$$

egyenlet megoldása ekvivalens olyan c és d meghatározásával, amelyekre:

$$C^d = W \cdot C^{cm} \quad R \text{ felett.} \tag{53}$$

Ahhoz, hogy c-t és d-t megkapjuk, előzetesen ki kellett számitanunk a C^d R feletti mátrixokat a $d=0,1,\ldots,m-1$ esetekben (ez $O(p_i{}^r)$ M-művelet), majd a kapott eredményeket rendezni kell (ez $O(p_i{}^r \log_2 p_i{}^r)$ M-művelet — R elemei között a rendezés például bináris kódjuk alapján lehetséges, a mátrixok között pedig a lexikografikus rendezés értelmezhető), C^{-1} kiszámitásához $\iota(n)$ M-műveletre van szükség. Ezt követően kiszámitjuk a W, WC^{-m}, WC^{-2m},... R feletti mátrixokat és összehasonlitjuk a táblázatba gyüjtött $\{C^i\}$ mátrixokkal. c-nek minden kipróbált értéke esetén egy R feletti mátrixszorzásra és $\log_2 p_i{}^r$ mátrix-összehasonlitásra van szükség, azaz összesen $(1+\log_2 p_i{}^r)$ M-műveletre. És $O(p_i{}^{1-r})$ mennyiségü c-értéket kell kipróbálni.

Tárolni kell a C^{-1}, C^o, C, ... , C^{m-1}, W mátrixokat és szükség van egy munka-mátrixra, amely a $WC^{-\tau_m}$ mátrixokat tartalmazza. Az egyes M-műveletknél az egy, ill. két mátrix tárolásán felül szükséges plusz memóriaigény nem növeli a (48) nagyságrendet.

Q.E.D.

A 2.Lemma bizonyitása módszert is ad a $g(W)$ függvényértékek kiszámitására, továbbá tekinthető Knuth [33] algoritmusa (ott $r = 1/2$ szerepelt) általánositásának.

$r = 1$ esetén $g(W)$ meghatározása táblázatból való kikeresés; $r = 0$ esetén pedig $g(W)$ értékét ugy határozzuk meg, hogy kiszámitjuk a C^g R felett értékeket $g = 0,1, \ldots ,(p_i-1)$ -ig (legfeljebb), amig a $C^g = W$ R felett egyenlőség nem teljesül. A 2.Lemma azt mutat-

ja, hogy a logaritmust tartalmazó faktorokat és a b faktort figyelmen kivül hagyva, az idő-memória-szorzat fixen tartható ($p_i n^2$), amig a két extremális helyzet, a táblázatból való kikeresés és a teljes halmazt kimeritő próbálkozom-és-hibázom eljárás között választjuk a stratégiát.

Mivel az a tendencia, hogy a memória költségesebb, mint a számitás, az $r < 1/2$ értékek a legérdekesebbek, és ekkor az előzetes számitások müveletigénye nem számottevő.

3. TÉTEL: Ha

$$h = p_1^{N_1} \cdot \ldots \cdot p_k^{N_k}, \quad p_i < p_{i+1} \quad (i=1,\ldots,k-1) \tag{54}$$

a h primfaktorizációja (p_i-k különböző primek és $N_i \geq 1$, $i=1,\ldots,k$), akkor tetszőleges

$$\{r_i\}_{i=1}^{k}, \quad 0 \leq r_i \leq 1, \quad i=1,\ldots,k \tag{55}$$

esetén a fent emlitett mátrix alapu logaritmus kiszámitható

$$O\Big(\sum_{i=1}^{k} N_i \log_2 h + p_i^{1-r_i}\,(1 + \log_2 p_i^{r_i})\Big) \text{ M-művelet}$$

$$+ \; O(k) \quad \text{művelet} \tag{56}$$

müvelettel és

$$O(n^2 b(\sum_{i=1}^{k} p_i^{r_i} + 3k)) + O(k\log_2 h) \qquad (57)$$

memóriabit felhasználásával. Az előzetes számitás — amelynek müveletigénye

$$O(\sum_{i=1}^{k} p_i^{r_i} \log_2 p_i^{r_i} + \iota(n)) \quad \text{M-művelet}$$

$$+ O(k\log_2 h) \quad \text{művelet} \qquad (58)$$

— nem számottevő, ha $r_i < 1/2 \quad (i=1,\ldots,k)$.

BIZONYITÁS: A bizonyitás a 2.Lemmából következik, figyelembe véve, hogy a kinai maradéktétel $O(k)$ müveletet és $O(k\log_2 h)$ memóriabitet, valamint $O(k\log_2 h)$ előzetes számitást igényel. A $\sum_{i=1}^{k} N_i \log_2 h$ tag az M-müveletek számában a $W = Z^u$ R felett és a $D = D^{p_i}$ R felett, a ciklusban végrehajtott M-müveletkből adódik. A ciklust $\sum_{i=1}^{k} N_i$ alkalommal hajtjuk végre.

Q.E.D.

1.3. MEGJEGYZÉSEK

Az előző fejezetben leirt, egységelemes gyürü feletti mátrix alapu logaritmust kiszámitó algoritmus ilyen formában nem terjeszthető ki sem másfajta reguláris, sem pedig szinguláris R feletti alapmátrixokra.

Ugy tünhet, hogy e mátrix alapu logaritmust kiszámitó algoritmus annál hatékonyabb lesz, minél kisebb az n értéke. Általában azonban ez nem igy van, ui. minden R egységelemes gyürü és tetszőlegesen nagy n méret esetén elég könnyen konstruálható olyan, az előző fejezetben megkövetelt feltételeknek eleget tevő A mátrix, amelyre az algoritmus triviális. Természetesen az eljárás hatékonysága nagy mértékben függ az R gyürütől.

Rögzitett R és n esetén a hatékonyság csak a h értékétől függ. Az algoritmus akkor a leghatékonyabb, ha h-nak csak kicsi primfaktorai vannak. Ez esetben az algoritmus lényegesen jobb, mint az, amelyet Knuth[33] algoritmusának az általánositásával kapnánk. Az algoritmus a $h = p$ és $h = 2p$ (p prim) esetekben a legkevésbé hatékony (ekkor a domináns számitás a $g(W)$ meghatározása $p_i = p$ esetén), és lényegében megegyezik Knuth algoritmusának az általánositásával, de nem szükséges, hogy $r = 1/2$ legyen.

Ezért tehát lényeges, hogy a gyürü feletti mátrixhatványozásra, mint egyirányu függvényre épülő kriptorendszerekben (ld. 2.RÉSZ) elkerüljük az olyan h értékkel rendelkező alapmátrixok alkalmazását (itt h továbbra is az illető mátrix különböző hatványainak a számát jelöli), amelyeknek csak kicsi primfaktorai vannak, hacsak n-et nem választottuk akkorára, hogy az még igy is képes biztositani a mátrixhatványozás egyirányu függvény voltát. Olyan alapmátrixot célszerü választani, amelynél h-nak van nagy primfaktora.

Az, hogy egy A_1 R feletti mátrixra a h érték ($A_1^h = I$ R felett, és h a minimális ilyen pozitiv egész) nagy legyen, könnyen elérhető, ha R-ben van olyan α elem, melyre $\alpha^m = e$ (ahol $e \in R$ a gyürü egységeleme és m a minimális ilyen tulajdonságu pozitiv egész) és m nagy (pl. α primitiv elem vagy primitiv egységgyök,stb.), vagy/és ha van R-ben olyan β elem, amelynek m' rendje az R additiv csoportjában nagy véges szám. Ugyanis legyen A egy olyan $(n-1)\times(n-1)$-es R feletti mátrix, amelyre $A^{\hat{h}}$ az R feletti $(n-1)\times(n-1)$-es egységmátrix és $\hat{h} > 1$ a legkisebb ilyen pozitiv egész. Ekkor például az

$$A_1 = \begin{array}{|ccc|c|}\hline & & & \Theta \\ & A & & \Theta \\ & & & \cdot \\ & & & \cdot \\ & & & \cdot \\ & & & \Theta \\ \hline \Theta & \cdots & \Theta\ \Theta & \alpha \\ \hline \end{array} \quad \text{ill.} \quad A_1 = \begin{array}{|c|ccc|c|}\hline e & \Theta & \cdots & \Theta & \beta \\ \hline \Theta & & & & \Theta \\ \cdot & & & & \cdot \\ \cdot & & A & & \cdot \\ \cdot & & & & \cdot \\ \Theta & & & & \Theta \\ \hline \Theta & \Theta & \cdots & \Theta & e \\ \hline \end{array} \qquad (59)$$

mátrix $n\times n$-es, ill. $(n+1)\times(n+1)$-es R feletti mátrix lesz (Θ az R gyürü zéruseleme), hatványai között szerepelni fog az R feletti $n\times n$-es, ill. $(n+1)\times(n+1)$-es egységmátrix, és ugyanakkor

$$h = \text{l.k.k.t.}[\hat{h},\ m] \ , \text{ ill. } h = \text{l.k.k.t.}[\hat{h},\ m'] \qquad (60)$$

is teljesül.

Az 1.2. Fejezet algoritmusa elvileg lehetővé teszi tetszőleges R egységelemes gyürü feletti mátrix alapu logaritmus kiszámitását. Ez azt jelenti, hogy a gyürü feletti mátrixhatványozásra, mint egyiránya függvényre épülő kriptorendszerekben is tetszőleges, egységelemes gyürüvé szervezett ábécét használhatunk. A tényleges megvalósitásnak természetesen gyakorlati korlátai vannak. Ezek közül a leglényegesebb, hogy csak fixpontos számitógépes ábrázolás engedhető meg, mivel az összehasonlitásokban a tényleges egyenlőség vagy nem-egyenlőség eldöntése a lebegőpontos hibák miatt lehetetlen lenne; továbbá hogy a számitógépben csak korlátos sok gyürüelem tárolható. Ez utóbbi adottság gyakorlatilag csak olyan R gyürük használatát engedi meg, amelyek esetében a választott vagy választható A alapmátrix összes különböző hatványában R elemeinek csak egy (az A mátrix, vagy bizonyos tulajdonsággal meghatározott mátrix-család ismeretében) előre kijeleölhető, nem tul nagy számosságu véges részhalmaza fordul elő (azaz e részhalmaz a számitógépben tárolható vagy elemei könnyen előállithatóak, és elemei között viszonylag hatékonyan elvégezhetők a gyürübeli müveletek). Tehát ha az R gyürü számossága végtelen, vagy véges ugyan, de rendkivül nagy, akkor a gyakorlati alkalmazhatóságot a választott vagy választható A mátrix alapvetően befolyásolja. A fixpontos ábrázolással kapcsolatos korlátozás kizárja többek között az $R = \mathbb{R}$ (valós számok teste), $R = \mathbb{C}$ (komplex számok teste) és $R = \mathbb{Q}$ (kvaterniótest) speciális esetek gyakorlati alkalmazható-

ságát. Ugyanakkor az $R = \mathbb{Q}$ (racionális számok teste) esetben ez a probléma a memóriaszükséglet megduplázásával és jelentős többletszámitással megoldható ugy, hogy az egyes racionális számokat redukált tört alakban, egész számpárokként kezeljük. Ez utóbbi megjegyzés értelmében többek között az $R = \mathbb{Q}$, $R = \mathbb{Z}$ (egész számok gyürüje) speciális esetek gyakorlati alkalmazhatósága az alapmátrix választásán mulik. Ha R nem tul nagy elemszámu véges egységelemes gyürü (pl. maradékosztálygyürü, $GF(p^r)$ véges test, véges egységelemes gyürü feletti mátrixgyürü, egy ábécé betüiből szervezett egységelemes gyürü, stb.), akkor ilyen probléma nem merül fel.

Az $n=1$, $R=GF(p^r)$ speciális eset részletes vizsgálata megtalálható [34]-ben. Annak a speciális esetnek, amikor az R gyürü az ábécé betüiből (és esetleg számokból és irásjelekből) szervezett egységelemes gyürü különös jelentősége lehet akkor, ha az R feletti mátrixok hatványozását a time-sharing számitógéprendszerekbe való bekapcsolódás esetében a felhasználó-azonositásánál ([1], [2], [17]-[18]; [37]) használjuk fel (ld. még 2.3.Fejezet).

Amennyiben az itt leirt algoritmus közel optimális, és találunk olyan A mátrixot, melyre n és h olyanok, hogy az R gyürü feletti mátrixhatványozás az 1.2.Fejezet algoritmusa alapján egyirányu függvény lesz, még a legkonzervativabb definició értelmében is (ilyen mátrix sok konkrét esetben egyszerüen megadható), akkor minden, az R feletti mátrixhatványozásra épülő kriptorendszer a legkonzervativabb definició szerint is megbizható lesz.

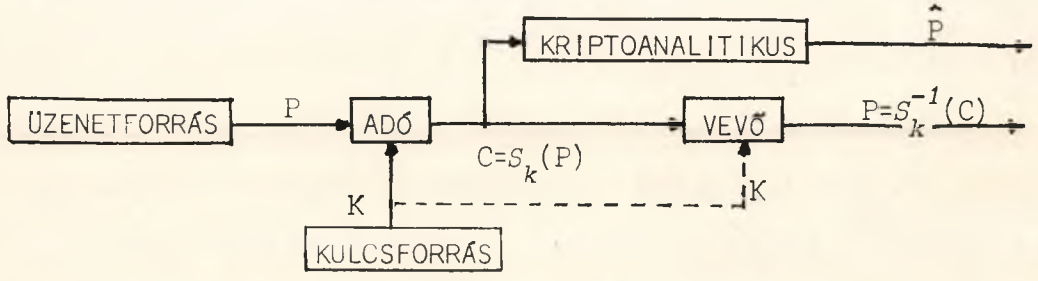
A kérdés tehát az, hogy létezik-e az 1.2. Fejezetben leirtnál hatékonyabb, még felfedezetlen algoritmus gyürü feletti mátrix alapu logaritmus kiszámitására. Egy hatékonyabb algoritmusnak azonban egyszerre kell javitania (nem rontania) a számitás- és memóriaszükséglet paramétereit (idő-memória átváltás [38] nem elegendő), mivel akár a memóriaigény, akár a futási idő egymagában is biztositathatja az R feletti mátrixhatványozás egyirányu függvény voltát az n mátrixméret és a mátrix kellő megválasztásával. Ha sikerül az R gyürü feletti mátrix alapu logaritmus kiszámitásának müveletigényére és/vagy memóriaszükségletére megfelelően nagy alsó korláto(ka)t adni, akkor matemetikailag bizonyitható lesz a gyürü feletti mátrixhatványozásra, mint egyirányu függvényre épülő kriptorendszerek megbizhatósága, ha valóban megbizhatók.

2. RÉSZ:

ALKALMAZÁSOK

2.1. KONVENCIONÁLIS KRIPTORENDSZER

Ebben a fejezetben egy példát mutatunk arra, hogy hogyan lehet a gyürü feletti mátrixhatványozást, mint egyirányu függvényt egy konvencionális kriptorendszer előállitására felhasználni.



7. Ábra Az információ áramlása a konvencionális kriptorendszerben

A 7.Ábra egy *konvencionális kriptorendszer*ben mutatja az információ áramlását, amikor a kriptorendszert bizalmas üzenetek közvetitésére használják. E kriptorendszer három részből áll: egy adóból, egy vevőből és egy lehallgatóból. Az adó egy nem-rejtjelezett P üzenetet generál és ezt akarja a nem-megbizható csatornán át a vevőhöz továbbitani. Azért, hogy a lehallgató ne értse meg P-t, az adó P-re egy invertálható S_k transzformációt alkalmaz, és ezzel alőállitja a rejtjelezett üzenetet: $C = S_k(P)$. Az adó a K kulcsot csak a jogosult vevőnek küldi el egy megbizható csatornán át (ezt az ábrán szaggatott vonal

jelzi). Minthogy a jogosult vevő ismeri a K kulcsot, C-ből meg tudja határozni az eredeti P üzenetet S_k^{-1} segitségével:

$$S_k^{-1}(C) = S_k^{-1}(S_k(P)) = P \quad . \tag{61}$$

A megbizható csatornán át P maga nem továbbitható a kapacitáskorlát vagy az információkésés miatt. Például a megbizható csatorna lehet egy hetente járó megbizott, a nem-megbizható csatorna pedig egy telefonvonal.

Alkalmazva az 1.1.Fejezetben bevezetett jelöléseket, legyen $M \neq I$, $M \in |M|$ egy olyan R feletti $n \times n$-es mátrix, melyre valamilyen $h > 1$ pozitiv egész esetén

$$M^h = I \quad R \text{ felett.} \tag{62}$$

(Nem fontos, hogy h a legkisebb ilyen tulajdonságu szám legyen.) Legyen továbbá k egy olyan pozitiv egész szám, melyre

$$\text{l.n.k.o.}(k\ ,\ h) = 1. \tag{63}$$

E feltétel biztositja, hogy létezik egyértelmü multiplikativ inverz modulo h, azaz a

$$d \equiv k^{-1} \pmod{h} \tag{64}$$

szám jóldefiniált.

Természetes megkötésként adódik, hogy k-nak ne válasszunk tul egyszerü módszert eredményező értéket (pl. $k \neq 1$).

Most legyen a sifrirozó eljárás

$$C = M^k \quad R \text{ felett,} \tag{65}$$

a desifrirozás ekkor

$$C^d = (M^k)^d = M^{kd} = M \quad R \text{ felett,} \tag{66}$$

ahol a kitevőben a müveletek modulo h értendők. Mindkét eljárás könnyen végrehajtható ($2\lceil \log_2 h \rceil$ M-müvelettel), d meghatározása k-ból az euklideszi algoritmussal $\lceil \log_2 h \rceil$ aritmetikai müveletet igényel.

A kriptológus esetleg egyszerüen meg tudja határozni a desifrirozó módszert, azaz a d értéket, ha ismeri a k számot, ugyanis ehhez csak arra van szüksége, hogy egy alkalmas h kitevőt találjon. Az általános feladat megoldatlan, de bizonyos speciális esetben ez nem okoz gondot. Nevezetesen fennáll a következő, egyszerüen igazolható tétel:

4. TÉTEL: Tételezzük fel, hogy az R egységelemes gyürü q rendje véges. Legyen A egy olyan R feletti $(n+1)\times(n+1)$-es háromszögmátrix, amelynek főátlójában végig az R egységeleme áll. Ekkor az A hatványai között előfordul az I $(n+1)\times(n+1)$-es R feletti egységmátrix; és egy olyan (egyszerüen meghatározható) h kitevő, melyre

$$A^h = I \quad R \text{ felett,} \tag{67}$$

lehet

$$h = q^t \quad , \tag{68}$$

ahol

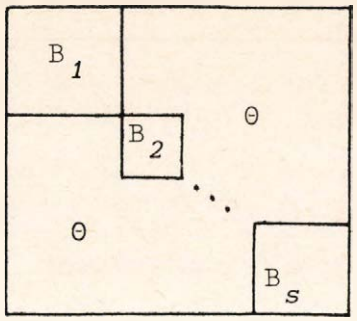
$$z_{t-1} < n \leq z_t \tag{69}$$

a következő $\{z_k\}$ sorozattal:

$$z_o := 0; \quad z_{k+1} := 2z_k + 1 \quad , \quad k \geq 1 \quad . \tag{70}$$

A bizonyitás azon az észrevételen alapul, hogy ha az $A_s := A^{q^s}$ felső/alsó háromszögmátrixokban a közvetlenül a fődiagonális felett/alatt lévő, csupa R-beli zéruselemből álló átlók száma l, akkor az $A_{s+1} := A_s^q$ mátrix legalább $l+(l+1)$ csupa R-beli zéruselemből álló diagonálissal rendelkezik közvetlenül a főátló felett/alatt.

Megjegyezzük, hogy a 4.Tétel birtokában egyszerüen konstruálhatók olyan más felépitésü, R feletti mátrixok, amelyeknek hatványai között előfordul a megfelelő méretü R feletti egységmátrix és egy ezt előállitó kitevő egyszerüen meghatározható. Nevezetesen a



(71)

felépitésü mátrixok eleget tesznek a fenti követelményeknek, ahol a B_i $(i=1,\ldots,s)$ blokkok a 4.Tétel feltételeit kielégitő $n_i \times n_i$ méretü $(n_i \geq 1,\ i=1,\ldots,s)$ R feletti mátrixok, Θ az R gyürü zéruseleme; a kérdéses h kitevő pedig a blokkokra adódó h_i kitevők legkisebb közös többszöröse:

$$h = \text{l.k.k.t.}[h_1, \ \ldots\ , h_s] \ . \tag{72}$$

Igy a kriptoanalitikus célja — több szempontból is — a k szám meghatározása. Erre azonban még C és M együttes ismeretében sincs más ismert módszer a próbálgatáson kivül, mint a logaritmusszámitás:

$$k = \log_M C \qquad R \ \text{felett.} \tag{73}$$

Ez pedig alkalmas R gyürüben megfelelően választott n és M esetén számitástechnikailag megoldhatatlan.

Ha C és M a k kulcs által kapcsolódnak egymáshoz, akkor tetszőleges pozitiv egész m szám esetén

$$M' = M^m \quad R \text{ felett} \tag{74}$$

és

$$C' = C^m \quad R \text{ felett} \tag{75}$$

szintén a k kulcs által kapcsolódik egymáshoz, azaz

$$C' = (M')^k \quad R \text{ felett.} \tag{76}$$

Tehát ha a kriptoanalitikus ismer egyetlen M—C párt, akkor tetszőlegesen sok, ugyanazzal a k kulccsal kapcsolatban álló M'—C' párt tud generálni. Ez a módszer azonban nem alkalmazható arra, hogy egy táblázat felállitásával a rendszert megtörje, amennyiben az M mátrix különböző hatványainak a száma elég nagy (1.3.Fejezet), például ha a táblázatban tárolandó mátrixelemek száma eléri vagy meghaladja a 2^{100} ($\approx 1.26\times10^{30}$) nagyságrendet.

Nem tünik előnyösnek, ha a kriptoanalitikus maga választhatja meg az M üzenetet az általa ismert M—C párban, bár azt sem tudjuk igazolni, hogy a rendszer ellent tud állni az ilyen tipusu támadásnak.

Probláma, hogy előfordulhat, hogy a választható k értékek száma kicsi. Szerencsére azonban általában a használható ($1 < k < h$) értékek ρ aránya nem tul kicsi. Az Euler-függvény definiciójából

$$\rho = \phi(h)/h \quad , \tag{77}$$

ahol $\phi(u)$ az u-nál nem nagyobb, az u-hoz relativ prim pozitiv egészek száma. Ismeretes [39],[40] , hogy

$$\rho = \prod_{p_i | h} (1-p_i^{-1}) \quad , \tag{78}$$

ahol a p_i-k a h primosztói. Ha például $h = 2p$, ahol p prim, akkor nagy h esetén

$$\rho = (1/2)[1 - (1/p)] \approx 1/2 \quad . \tag{79}$$

De ha h nem $2p$ alaku, akkor is elég sok érték közül válogathatunk, mivel $h < 1.6\times10^{1o3}$ esetén $\rho > 0.1$ ([34]). Igy átlagosan legfeljebb tiz próbálkozás szükséges egy megfelelő k érték megtalálásához, és bár minden egyes érték tesztjéséhez az euklideszi algoritmust használjuk, e számitásigény nem jelentős.

Lényegesen nehezebb probléma az, hogy egy adott desifrirozó kulcshoz megfelelő üzenetet és sifrirozó kulcsot állitsunk elő. Ehhez arra van szükség, hogy az R gyürü feletti $n\times n$-es mátrixok között jellemezzük azon mátrixok családját, amelyekhez (viszonylag egyszerüen) meghatározható egy olyan $1 < h$ kitevő, amelyre a mátrixcsalád tetszőleges elemét emelve az R feletti $n\times n$-es egységmátrixot kapjuk. Lényeges, hogy a jellemzés a mátrixok strukturájára vonatkozzon és az adódó mátrixcsalád(ok) számossága a gyakorlat szempontjából elegendően nagy legyen. Ilyen tipusu jellemzések lényegesen bővitenék a

kriptorendszer alkalmazási lehetőségeit. A 4.Tételen és ennek közvetlen következményein kivül ilyen jellegü eredmény nem ismeretes.

Tehát e kriptorendszer használhatósága valójában az R gyürü feletti mátrix alapu logaritmus meghatározásának a nehézségén mulik. E nehézség a számitástechnikai megvalósitás (végrehajtandó müveletek száma, memóriaigény, futási idő, stb.) problémáin felül magában foglalja az alapmátrix különböző hatványai számának meghatározását is (1.2.Fejezet), amire nem ismert hatékony eljárás.

Tételezzük fel, hogy a közvetitendő üzenet mindig N darab R-beli elemből álló jelsorozeat. Ekkor a 4.Tétel adta lehetőségekre támaszkodva az M $(n+1)\times(n+1)$-es üzenet-mátrix méretét válasszuk meg ugy, hogy

$$n(n-1)/2 \geq N \geq (n-1)(n-2)/2 \qquad (80)$$

legyen és az N elemmel valamilyen módszer alapján töltsük fel pl. a felső háromszöget (a főátlóban mindenütt R egységeleme áll). Ekkor a felső háromszögben legfeljebb $(n-1)$ darab neminformativ elem (olyan R-beli elem, amely nem lehet üzenet eleme) áll, ugy a redundancia $\leq 200/n$ százalék, de ha $2N = n(n-1)$, akkor nincs redundancia.

Látható, hogy e rendszerrel a sifrirozás a közvetitendő üzenetet gyakorlatilag csak jelentéktelen mértékben növeli meg, és ebből adódóan nagy információátviteli sebesség érhető el, ami sok alkalmazási lerületen élnyeges, esetleg döntő szempont.

2.2. NYILVÁNOS KULCSELOSZTÓ RENDSZER

Az R egységelemes gyürü feletti mátrixhatványozás, mint egyirányu függvény felhasználható nyilvános kulcselosztó rendszerek generálására is ([1]-[2],[34]). Az ilyen rendszer célja az, hogy egy nyilvános csatornán át titkosan "továbbitson" egy jelkulcsot anélkül, hogy ezt a müveletet bármiféle bizalmas információcsere megelőzte volna.

Tételezzük fel, hogy egy hálózat tagjai bizalmas információikat rejtjelezve kivánják közölni egymással, éspedig ugy hogy közülük bármely kettő üzenetváltása a többiek számára megfejthetetlen maradjon. Tegyük fel, hogy a hálózat tagjai e cél érdekében választanak egy R egységelemes gyürüt, egy n pozitiv egész számot és egy olyan A $n\times n$-es R feletti mátrixot, amelynek sok különböző hatványa van és ezek között előfordul az I $n\times n$-es R feletti egységmátrix is. Meghatároznak egy olyan $h > 1$ potitiv egész számot, melyre

$$A^h = I \quad R \text{ felett.} \tag{81}$$

(A 4.Tétel példát ad a kivánat tulajdonságu mátrixokra és a h szám meghatározására.)

A hálózat minden tagja véletlenszerüen választ egy h-hoz relativ prim x_i számot az $\{1,\ldots,h-1\}$ halmazból. Az

x_i számot mindenki titokban tartja, de az

$$Y_i = A^{x_i} \qquad R \text{ felett} \tag{82}$$

mátrixot elhelyezi egy nyilvános file-ba a nevével és a cimével együtt. Ekkor ha az i és j tag bizalmas kapcsolatot akar teremteni egymással, kulcsként a

$$K_{ij} = A^{x_i x_j} \qquad R \text{ felett} \tag{83}$$

mátrixot használják. Az i felhsználó ugy kapja meg ezt a kulcsot, hogy a nyilvános file-ba elhelyezett Y_j mátrixot x_i-edik hatványra emeli R felett:

$$\begin{aligned} K_{ij} &= Y_j^{x_i} \qquad R \text{ felett} \\ &= \left(A^{x_j}\right)^{x_i} \qquad R \text{ felett} \\ &= A^{x_j x_i} = A^{x_i x_j} \qquad R \text{ felett.} \end{aligned} \tag{84}$$

A j felhasználó hasonlóan kapja meg a kulcsot:

$$K_{ij} = Y_i^{x_j} \qquad R \text{ felett.} \tag{85}$$

A hálózat egy harmadik tagja azonban K_{ij}-t az Y_i és Y_j mátrixokból, ugy tünik, csak próbálgatás vagy logaritmusszámitás utján tudja meghatározni:

$$K_{ij} = Y_i^{(\log_A Y_j)} \qquad R \text{ felett.} \tag{86}$$

n-et és A-t a következők figyelembevételével kell megválasztani:

A h számnak olyannak kell lennie, hogy mind $\phi(h)$ (ϕ az Euler függvény), mind A különböző hatványainak a száma gyakorlatilag lehetetlenné tegye a rendszer megtörését az összes lehetőséget kimeritő próbálkozom-és-hibázom eljárással. Ekkor egyébként $\phi(h)$ lényegesen nagyobb a hálózat tagjainak a számánál és igy az azonos x_i érték választásának valószinüsége elenyészővé válik. Továbbá mivel az A mátrixnak sok különböző hatványa van, azért annak is nagyon kicsi lesz a valószonüsége, hogy különböző párok kulcsai megegyeznek.

Az, hogy az x_i szám és h relativ prim legyen, azért szükséges, hogy ne fordulhasson elő $K_{ij} = I_R$ felett degenerált kulcs. E feltétel elhagyása esetén a degenerált kulcs előfordulásának π valószinüségére:

$$\pi \geq (h-1-\phi(h))/(h-1)^2 \quad , \tag{87}$$

ez pedig pozitiv, ha h nem prim. Egyszerüen belátható, hogy ha h éppen az A mátrix különböző hatványainak a száma, akkor

$$(h-1-\phi(h))/(h-1)^2 \leq \pi \leq (h-h/c-c+1)/(h-1)^2, \tag{88}$$

ahol

$$c = \max\{q:\ q \text{ pozitiv egész},\ q \mid h,\ q < \sqrt{h}\ \}. \tag{89}$$

Bár nem tudjuk igazolni, hogy e rendszer megbizható, ha az R feletti mátrixhatványozás egyirányu függvény, arra sem látunk módot, hogy Y_i-ből és Y_j-ből K_{ij}-t anélkül határozzuk meg, hogy előzőleg x_i-t vagy x_j-t kiszámitanánk.

A Márkus [50] dolgozat e nyilvános kulcselosztó rendszer egy lehetséges alkalmazását tárgyalja. A vizsgált eljárás egy speciális tulajdonságu, un. szupernövekvő együtthatósorozatos, egydimenziós, egyenlőséggel megfogalmazott általánositott hátizsák feladatot használ az üzenet rejtjelezésére. A módszer biztonsága a kulcselosztó rendszer megbizhatóságán és azon mulik, hogy elég sok lehetőség van-e a hátizsákegyütthatók megválasztására.

2.3. FELHASZNÁLÓ-AZONOSITÓ ELJÁRÁSOK

A time-sharing számitógéprendszerek alkalmazásakor az egyik központi feladat annak biztositása, hogy a számitógéphez csak az arra jogosultak férhessenek hozzá. Különösen nagy jelentőségü ez a probléma akkor, ha a számitógéprendszer több független felhasználó által használt adatbázist vagy adatbankot kezel. Azért, hogy a time-sharing rendszert megóvjuk az illetéktelen felhasználóktól, szükséges a felhasználók valamiféle ellenőrzése, azonositása.

A felhasználók ellenőrzése többféleképpen megoldható. Például bizonyos felhasználóknak csak bizonyos terminálok meghatározott időszakokban való használatát engedjük meg, ezáltal korlátozva a rendszerbe való bekapcsolódást és a rendszer által tartalmazott információ elérését. Egy másik, nagyon elterjedt módszer a *password*-ök alkalmazása. Minden felhasználó amikor először létesit kapcsolatot a rendszerrel, választ egy jelsorozatot (password) és ezt elhelyezik egy password-ök számára fenntartott táblában, a *password directory*-ban. Ezután minden alkalommal, amikor az illető felhasználó be akar kapcsolódni a rendszerbe, az megkérdezi tőle a password-öt és akkor és csak akkor engedélyezi számára a bekapcsolódást, azaz akkor és csak akkor fogadja őt el a rendszer azonositottnak, ha ismeri a password-jét. Ha egy felhasználó a saját password-jét mások előtt eltitkolja, akkor minden tőle telhetőt megtett a rendszer biztonsága érdekében.

A hagyományos password-rendszer magában foglal egy elrejtett password directory-t, amelyet egy olyan fileban tárolnak, amelyhez csak az azonositást végző program tud hozzáférni, a közönséges felhasználó nem. Természetesen van egy személy, nevezzük "rendszer-adminisztrátor"-nak, aki e file-t karbantartja, tehát jogosult e file tartalmának olvasására és megváltoztatására egyaránt. Ez a file egy olyan táblázatot tartalmaz, amelynek elemei az egyes hozzáférésre jogosult felhasználók neve és password-je. Egy felhasználót a rendszer akkor tekint ténylegesen azonositottnak, ha az illető ismeri a nevéhez tartozó pass-

word-öt. Természetesen az a felhasználó, aki elfelejtette a password-jét, csak ugy tudja ismét igénybe venni a rendszert, ha uj password-öt választ és azt a nevéhez kapcsoltan elhelyezi a password directory-ban. A rendszer megbizhatósága — a felhasználó elővigyázatossága mellett — a password directory titokban tartásán mulik. Ezért az operációs rendszert ugy kell kiépiteni, hogy lehetőség nyiljon arra, hogy a rendszer-adminisztrátoron kivül az összes felhasználót megakadályozzuk e táblázat elérésében.

E rendszernek több hátránya is van. Először: a rendszer-adminisztrátor ismerheti a password-öket, jóllehet semmi sem indokolja, hogy ezeket ismernie kellene, vagy akár csak módja legyen ismerni. Másodszor: a password directory-t bármely listázázása esetén (még ha erre komoly ok is volt) véletlenül megláthatja egy arra illetéktelen személy. Harmadszor: bárki, aki a számitógép fizikai egységeihez hozzáférhet (pl. egy operátor), viszonylag könnyen kinyomtathatja a password-file tartalmát. Végül: ez a rendszer nem valósitható meg olyan eszközökkel, amelyek file-kezelő rendszere nincs védve az illetéktelen file-leolvasástól.

Ez vezetett ahhoz a majdnem képtelen követelményhez, hogy olyan bekapcsolódó, azonositó eljárást dolgozzanak ki, amely ugy állapitja meg egy jelsorozatról, hogy az password-e, hogy magukat a password-öket ténylegesen nem ismeri. Bár lehetetlennek tünik, e kivánalom elég könnyen kielégithető.

2.3.1. MEGBIZHATÓ FELHASZNÁLÓ-AZONOSITÁS

Elsőként Wilkes [37] irt le egy olyan R.M.Needham-től származó és az angliai Cambridge-ben megvalósitott eszközt (Wilkes ezt "one-way cipher"-nek nevezi), amelylyel megoldható az azonositás a password tényleges ismerete nélkül.

Needham módszerének lényege a következő. Amikor a felhasználó először küldi el *PW* password-jét a rendszernek vagy bármikor, amikor azt meg akarja változtatni, a számitógép automatikusan kiszámitja egy *f* függvénynek az *f(PW)* értékét, és *PW* helyett ezt a számitott értéket tárolja a password directory-ban. Minden egyes bekapcsolódás alkalmával a számitógép kiszámitja *f(X)*-et, ahol *X* a betáplált password, és összehasonlitja *f(X)*-et a tárolt *f(PW)* értékkel. A felhasználót a rendszer akkor és csak akkor fogadja el azonositottnak, ha a két érték megegyezik.

Mivel az *f* függvényt minden bekapcsolódáskor ki kell számitani, a kiszámitáshoz szükséges időnek megfelelően kicsinek kell lennie. Ha el tudjuk érni, hogy ugyanakkor az f^{-1} inverz függvény egy értékeének a kiszámitásához legalább 10^{30} utasitásra és/vagy legalább 2^{100} bit memóriára legyen szükség, akkor gyakorlatilag hiába ismeri valaki a password directory *f(PW)* tartalmát, ebből nem tudja meghatározni *PW*-t és igy nem tud azonositás nélkül a rendszerbe lépni.

Megjegyezzük, hogy a bekapcslolódást ellenőrző program $f(PW)$-t megfelelő f függvényválasztás esetén nem fogadja el password-nek, mivel azonnal az $f(f(PW))$ értéket számitja ki (ha lehet) és ez nem fog szerepelni a password directory-ban.

Tételezzük fel, hogy az f^{-1} inverz függvény kiszámitására ismert leghatékonyabb eljárás vagy számitástechnikailag kivitelezhetetlen (idő- vagy/és memóriaszükséglete miatt), vagy pedig kevésbé hatékony, mint a próbálkozom-és-hibázom kereső eljárás, amely tetszőleges f függvény esetén alkalmazható. Nyilván csak ez utóbbi esettel kell foglalkozni. Ekkor ha valaki még magát az f függvényt is ismeri, akkor sincs előnyös helyzetben. Ugyanis ekkor ugyan tetszés szerinti mennyiségben tud $(X , f(X))$ párokat generálni, azonban ha a felhasználók password-jeiket véletlenszerüen választják ki egy megfelelően nagy számosságu halmazból, akkor egy adott $f(X)$ érték egy inverz képének meghatározásához szükséges próbálkozások számának várható értéke nem tulságosan degenerált f függvény esetén igen nagy lesz.

Ennek megmutatása céljából definiáljuk az f függvény *degenerációját* [18] , mint az f függvény $D(f)$ értelmezési tartományában lévő azon X_i argumentumok maximális számát (feltételezzük, hogy van maximum), amelyeket f mind ugyanarra az Y_j elemre képez le, miközben Y_j végigfut az f függvény $R(f)$ értékkészletén, azaz

$$d := \sup_{Y_j \in R(f)} |\{X_i \in D(f) : f(X_i) = Y_j\}| = \max_{Y_j \in R(f)} |\{X_i \in D(f) : f(X_i) = Y_j\}| . \quad (90)$$

Tételezzük fel, hogy az X_i-ket a felhasználók egy H ($H < \infty$) halmazból választják véletlenszerüen, továbbá hogy az a C személy, aki egy $f(X)$ értékből egy hozzá tartozó X argumentumot akar meghatározni, ismeri az összes (a $H<\infty$ feltevésből adódóan véges sok) Y_j értéket. Tegyük fel, hogy C a halmazból véletlenszerüen vesz X elemeket, esetleg ugyanazt az X elemet több alkalommal is kipróbálja, egészen addig, amig egy olyan X-et nem talál, melyre

$$f(X) = Y_j \quad \text{valamely } Y_j\text{-re.} \quad (91)$$

(Az az eset, amikor C ügyel arra, hogy egyetlen X elemet se próbáljon ki többször, ha H igen nagy, nagyon közel van az itt tárgyalt esethez, de vizsgálata lényegesen bonyolultabb — lásd Purdy [18] —; másrészt ennek megvalósitása is komoly nehézséget okoz C-nek, különösen igen nagy H érték mellett.) Az teljesen nyilvánvaló, hogy C-nek ugyanolyan eloszlás szerint célszerü a kipróbálandó X értékeket kiválasztania, mint amilyen eloszlás szerint a felhasználók a password-jeiket kiválasztották.

Legyen m a password directory-ban szereplő Y_i értékek száma. Ha f bijektiv lenne, akkor minden egyes próbálkozásnál a siker valószinüsége m/H lenne. Ha azonban az f függvény d degenerációjára $d > 1$ teljesül, akkor

e valószinüség novekszik. Ekkor az f függvény az m darab Y_i password mindegyikét legfeljebb d dülönböző X esetén állitja elő, miáltal legfeljebb md darab X érték válik elfogadhatóvá. Igy minden egyes próbálkozásnál a siker valószinüsége legfeljebb dm/H (ez el is érhető). Jelölje

$$b = dm/H \quad . \tag{92}$$

Ekkor a legrosszabb esetet vizsgálva, azaz feltételezve, hogy tetszőleges X argumentum elfogadási valószinüsége egyenlő b-vel, annak valószinüsége, hogy a k-ik próbálkobálkozás sikeres, de a korábbiak nem sikerültek:

$$b\ (\ 1 - b\ )^{k-1} \quad . \tag{93}$$

Igy az első sikeres kisérletig végzett próbálkozások számának várható értéke:

$$k_e = \sum_{k=1}^{\infty} kb(1-b)^{k-1} = 1/b = H/(dm) \quad . \tag{94}$$

Tehát ha f degenerációja nagy, akkor a rendszer egyáltalán nem megbizható. Ha d-re nem ismerünk korlátot, akkor a rendszer megbizhatóságát sem ismerjük.

Az azonositást végző programban szereplő f függvényt azonban senkinek sem szükséges ismernie; az e függvény kiszámitására szolgáló programot fizikailag is lehet és kell is védeni. Az erre az f függvényre tett megkötések-

nek egy jól megválasztott egyirányu függvény nyilván eleget tesz. A megvalósitásokban a megbizhatóság mértékének és az $f(X)$ függvényértékek kiszámitási idejének megfelelő arányát kell megtalálni. Célszerünek tünik magukba a terminálokba azonositó egységeket (mikroszámitógép) beépiteni. Ekkor egy terminálon különböző felhasználók különböző eljárások szerint azonosithatják magukat.

Evans, Kantorowitz és Weiss [17] az f egyirányu függvény előállitására iterativ módszert javasol, minden iterációban egy másik "egyszerü függvény"-t használva (legalább egyik függvény nemlineáris). A szerzők az "egyszerü függvény"-ekre is adnak javaslatokat: bitpermutáció, táblázatból való kikeresés, bithozzávétel, stb. E módszereket alkalmazva az f függvény nyilvánosságra hozása sem rontja jelentősen a rendszer megbizhatóságát.

Purdy [18] az f egyirányu függvénynek primtest feletti magas fokszámu polinomokat ajánl. Legyen P egy nagy prim, $n, a_1, \ldots, a_n$ egész számok és tekintsük a

$$p(x) = x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n \qquad (95)$$

polinomot. Legyen $1 \leq x \leq P$, és definiáljuk f-et:

$$f(x) \equiv p(x) \pmod{P} . \qquad (96)$$

Ekkor f a $\{0, 1, 2, \ldots, P-1\}$ halmazt önmagába képezi. Mivel az

$$x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n \equiv 0 \pmod{P} \qquad (97)$$

kongruenciának (P prim) legfeljebb n gyöke lehet ([39], [40]), az igy adódó f függvény degenerációja legfeljebb n.

Purdy a $p(x)$ polinomra a következőt javasolja:

$$p(x) = x^n + b_1 x^{n_1} + b_2 x^{n_2} + \cdots + b_k x^{n_k} + a_{k+1} , \quad (98)$$

ahol $n > n_1 > \cdots > n_k \geq 1$ azért, hogy a Knuth [30, 4.6.3.] módszerekkel az $f(x)$ értékének kiszámitása

$$O(k \log_2 n \log_2 P) \quad (99)$$

idő alatt elvégezhető legyen, mig ugyanakkor az inverz meghatározásához ([18], [30], [41], [42]) szükséges

$$O(n^2 (\log_2 P)^2) \quad (100)$$

és

$$O(P/nm) \quad (101)$$

idők (m a password-ök száma) csillagászativá növekednek.

Ennek illusztrálására [18] legyen:

$$f(x) \equiv p(x) = \\ = x^n a_1 x^{n_1} + a_2 x^3 + a_3 x^2 + a_4 x + a_5 \pmod{P} , \quad (102)$$

ahol

$$P = 2^{64} - 59 , \quad n = 2^{24} + 17 , \quad n_1 = 2^{24} + 3 \quad (103)$$

és az a_i $(i=1, 2, 3, 4, 5)$ együtthatók tetszőleges 19-jegyü számok (nincs értelme az a_i-ket $P \approx 10^{19}$-nél nagyobbra választani).

Ez a választás elég megbizható a polinom-gyök-kereső módszerekkel szemben, mivel ezek most

$$n^2(\log_2 P)^2 \approx 10^{14}(19\log_2 10)^2 > 10^{16} \qquad (104)$$

müveletet igényelnek. Feltéve, hogy a számitógép 10^6 müveletet hajt végre másodpercenként, ilyen eljárással egy gyök megtalálása több, mint 10^{10} másodpercet, körülbelül 400 évet igényelne.

A próbálkozom-és-hibázom eljárásnál, feltételezve, hogy a password-ök száma

$$m = 100 , \qquad (105)$$

az első sikeres kisérletig végzett próbálkozások számának várható értéke

$$k_e = P/m = 10^{19}/(10^7 \times 10^3) = 10^9 . \qquad (106)$$

Feltételezve az előbbi müveleti sebességet és azt, hogy egy $f(\cdot)$ függvényérték kiszámitásához 10^3 müveletre van szükség, a kongruencia egy megoldásának kiszámitásához szükséges idő várható értéke 10^6 másodperc, azaz körülbelül 15 hét.

2.3.2. GYÜRÜ FELETTI MÁTRIXHATVÁNYOZÁSRA ÉPÜLÖ FELHASZNÁLÓ-AZONOSITÁS

Most, használva az 1.2.Fejezetben alkalmazott jelölésrendszert, két, az R gyürü feletti mátrixhatványozáson alapuló felhasználó-azonositó eljárást ismertetünk. (Feltételezzük, hogy az R egységelemes gyürü összes, vagy a leirandó eljáráshoz szükséges összes elemét tárolni tudjuk a számitógépben, vagy hatékony eljárással elő tudjuk állitani, és hogy az R-beli müveletek számitógépes implementálására is hatékony eljárások állnak rendelkezésre.)

I)

Legyen J egy pozitiv egészekből álló véges halmaz és legyen $M \subset \mathbb{M}$ olyan véges részhalmaz, hogy az ebbe tartozó mátrixok különböző hatványainak száma nagy legyen (olyan nagy, hogy figyelembe véve a mátrixok méretét is, a kitevő meghatározása gyakorlatilag megoldhatatlan legyen az 1.2.Fejezet algoritmusával és a próbálkozom-és-hibázom eljárással egyaránt).

Az egyes felhasználók password-nek egy

$$(A, e) \qquad (107)$$

párt választanak, ahol A-t az M halmazból, e-t a J halmazból választják véletlenszerüen. Ha bekapcsolódni ki-

vánnak a rendszerbe, akkor betáplálják az (A,e) párt. Az azonositást végző program kiszámitja az

$$A^e \quad R \quad \text{felett} \tag{108}$$

hatványt és ezt az értéket elhelyezi a password dorectory-ban az első alkalommal, a továbbiakban pedig e kiszámitott értéket hasonlitja össze a password directory tartalmával. Egyezés esetén, és csak ekkor, a felhasználót azonositottnak tekinti.

Nyilvánvaló, hogy M számosságának növelésével minimálisra csökkenthető annak valószinüsége, hogy két felhasználó ugyanazt a mátrixot választja. Azonban az e kitevő választásától függően még előfordulhat, hogy különböző felhasználók tárolt password-jei azonosak lesznek. Ennek valószinüsége a J halmaz számosságának növelésével várhatóan csökken. Azonban ha a password directory-ban a tárolt értékek a felhasználó nevével együtt szerepelnek, akkor nem feltétlenül szükséges, hogy a tárolt password-ök különbözzenek egymástól.

Megjegyezzük, hogy e rendszer használatához a felhasználóknak csak az M és a J halmazt kell ismerniük, az R gyürüt nem.

Tételezzük fel, hogy egy C illetéktelen személy megszerzi egy U felhasználó password-jének mátrix részét, tudomására jut az R gyürü és hozzáfér valamilyen módon a password directory tartalmához. Ekkor a tárolt mátrixok közül ki kell választania az U felhasználó mátrixának

tárolt képét (ez csak akkor probléma, ha a password directory-ban nem szerepel a felhasználók neve, ugyanis nem ismert hatékony módszer annak eldöntésére, hogy két mátrix közül egyik hatványa-e a másiknak, vagy sem), majd U password-jének kitevő részét (vagy egy azzal egyenértékü másik számot) kell kiszámitania. Ez utóbbi müvelet alkalmas választásokkal gyakorlatilag kivitelezhetetlen.

Egy ember számára nyilván lehetetlen elegendően hosszu, "értelmetlen" jelsorozat megjegyzése, és annak terminálon történő begépelése sem lehet kielégitő megoldás. Ezért célszerü a hosszu jelsorozatokat valamilyen adathordozón, pl. mágneskártyán a rendszerbe táplálni. A teljes password mágneskártyán való tárolása azonban rendkivül veszélyes, mivel egy ilyen kártya elvesztése, esetleg ellopása lehetővé teszi illetéktelen személyek hozzáférését a rendszerhez.

Az I) módszer megbizhatóságának kulcsa lényegében az e kitevő titokban tartása, igy ezt semmiképpen sem szabad szerepeltetni a mágneskártyán. Ugyanakkor egyetlen szám emlékezetben tartása nem okoz problémát (de természetesen ez a kitevő is szerepelhet egy külön mágneskártyán). Tehát rendszerünk biztonságosan müködhet, ha bekapcsolódáskor a felhasználó beteszi kártyáját (amely a mátrixot tartalmazza) egy olvasó berendezésbe és a kitevőt hozzá vagy terminálon gépeli be, vagy egy másik mágneskártyán táplálja a rendszerbe.

Megemlitjük, hogy a fentiekben szereplő "elegendően hosszu 'értelmetlen' jelsorozat" e módszerrel "értelmes" jelsorozattal váltható fel, amennyiben az R gyürüt az ábécé betüiből irásjelekből és esetleg számjegyekből szervezzük. Ezáltal nagy, mátrixokat kitöltő szövegek (jelsorozatok) jegyezhetők meg és ezek terminálon történő begépelése is magyságrendekkel hatékonyabb lehet, mint ugyanilyen hosszu "értelmetlen" jelsorozatoké ([49]). Ugyanakkor az R gyürü feletti mátrixhatványozás és az ehhez hasonló eljárások várhatóan lehetlenné teszik a betügyakoriságon és egyéb nyelvstatisztikai megfontolásokon alapuló megfejtést — ilyen irányu vizsgálatok folyamatban vannak.

II)

A második módszerben felhasználjuk a már emlitett iterativ gondolatot és a polinomkongruenciákat is, és egy olyan két számból álló password-öt adunk, amelynek bármelyik fele önmagában is biztositathatja — alkalmas paraméterválasztás esetén — a rendszer megbizhatóságát.

Legyenek k_1 és k_2 nagy pozitiv egészek, P és P' legyen két nagy prim (a nagy primszámokra vonatkozóan lásd pl. [30], [43]-[45]). Legyen $p(x)$ és $p'(x)$ két Purdy által javasolt alaku polinom és legyen

$$f_1(x) \equiv p(x) \pmod{P} \qquad (109)$$

és

$$f_2(x) \equiv p'(x) \pmod{P'} . \qquad (110)$$

Ekkor f_1 és f_2 egyirányu függvények. Tételezzük fel, hogy

$$k_2 \geq P' . \qquad (111)$$

Minden felhasználó választ magának egy

$$(w , v) \qquad (112)$$

password-öt az

$$[1 , k_1] \times [1 , k_2] \qquad (113)$$

téglalap egész koordinátáju pontjai közül.

Az azonositást végző program a következőképpen müködik. Először kiszámitja az $f_1(w)$ számot, majd egy M-beli $n \times n$-es mátrixokból álló, egy fizikailag jól védett file-ban tárolt mátrix-sorozatból kiválasztja az $f_1(w)$-ik elemet; vagy egy eljárás-sorozatból, melynek elemei egy-egy ilyen, egymástól különböző mátrixot állitanak elő, kiválasztja az $f_1(w)$-ik elemet. E sorozat P elemből áll, és a sorozatban szereplő mátrixok, vagy a sorozatban szereplő eljárások által előállitott mátrixok különböző hatványai (R felett) számának minimumát jelölje ℓ. Most a program kiszámitja az $f_2(v)$ értéket és ezután az előbb kiválasztott, vagy az előbb kiválasztott eljárással előállitott $M_{f_1(w)}$ mátrixot $f_2(v)$-ik hatványra emeli

R felett, és ezt az értéket tárolja a password directory-ban, hogy a felhasználót azonositani tudja.

Elérhető, hogy e [1] módszer degenerációja csak a polinomok degenerációjától függjön. Ugyanis válasszuk meg P-hez a tárolt mátrix-, ill. eljárás-sorozatban levő elemeket ugy, hogy az egyes, a sorozatban szereplő, ill. a megfelelő eljárással előállitott mátrixoknak legalább az első $P\ell$ hatványai egymástól is különbözzenek, azaz az

$$\{ M_1, M_1^2, \ldots, M_1^{\ell}, \ldots, M_P, \ldots, M_P^{\ell} \} \quad (114)$$

halmazban ne legyen két R felett egyenlő mátrix. Ha

$$\ell \geq P' , \quad (115)$$

akkor az eljárásnak a mátrixhatványozást tartlamazó része egy-egyértelmü.

Ha most feltételezzük, hogy az f_1 függvény degenerációja d_1, az f_2 függvényé pedig d_2, akkor annak valószinüsége, hogy két különböző felhasználó password-jének tárolt alakja azonos, kisebb, mint

$$\pi = \lceil k_1/P \rceil \cdot d_1 \cdot k_1^{-1} \cdot \lceil k_2/P' \rceil \cdot d_2 \cdot k_2^{-1} . \quad (116)$$

Ha ez a π érték megfelelően kicsi, akkor biztonságosan (a biztonság foka π) megengedhető, hogy a password directory-ban csak a transzformált password szerepeljen, a felhasználó neve ne. Ugyanis ekkor legalább $(1-\pi)$ valószi-

nüséggel a password tárolt alakja önmagában, név nélkül is azonositja a jelhasználót, és igy lehetőség van arra, hogy egy-egy tárolt password-höz kapcsoljuk hozzá a korábban a felhasználó nevéhez kötött funkciókat (pl. bizonyos adatterülethez, programokhoz való hozzáférés, gépidőkönyvelés, stb.).

Ha a $p'(x)$ polinom az identitás, akkor a mátrixhatványozás egy-egyértelmüségét $\ell \geq k_2$ biztositja; ha $p(x)$ is az identitás és még $k_1 \leq P$ is teljesül, akkor az egész azonositó eljárás kölcsönösen egyértelmü leképezés.

A [I]) módszer a felhasználótól csak egy számpár ismeretét kivánja meg (és nyilván ismernie kellett a k_1 és k_2 számokat is) és e két számból álló password begépelése terminálon nem tul nagy munka; bár ha van erre alkalmas olvasó, akkor célszerübb e két számot két különböző mágneskártyán táplálni a rendszerbe, annál is inkább, mert e számok elég nagyok is lehetnek.

Egy azonositás nélkül, illetéktelenül bekapcsolódni akaró C személynek ismernie kell *(i)* az R gyürüt, *(ii)* a tárolt mátrix- vagy eljárás-sorozatot, *(iii)* a password directory tartalmát, *(iv)* a $p(x)$ és $p'(x)$ polinomokat.

Ha C csak ezeket az adatokat ismeri, ill. fér hozzájuk, akkor legfeljebb a sorozatban levő vagy ott előállitott mátrixok és a password directory-ban szerplő mátrikszok között kereshet valamilyen módon alap-hatvány párokat és megállapithatja a kitevőt. Mint már emlitettük, nem ismeretes olyan hatékony algoritmus, amely két mátrix-

ról eldöntené, hogy egyik hatványa-e a másiknak, vagy sem. Ha C talált alap-hatvány párokat, akkor a kitevő meghatározása (mind logaritmus-számitással, mind próbálgatás utján) rendkivül lassu és fáradságos, esetleg gyakorlatilag megvalósithatatlan. Ezért várhatóan C itt igen nagy munka árán sem ér el számottevő eredményt. (Itt segitségére lehet k_2 ismerete.) Ha sikerült meghatároznia, hogy a sorozatban pl. az i-edik (szereplő vagy előállitott) mátrixnak hányadik hatványa szerepel a password directory-ban és melyik az a tárolt mátrix, akkor még két kongruencia egy-egy megoldását kell előállitania ahhoz, hogy egy, a tényleges password-del egyenértékü számpárt kapjon. Ez alkalmas polinomválasztással számitástechnikailag megoldhatatlan.

Ha a password directory-ban név nem szerepel, akkor külön probléma a password-höz tartozó felhasználó kilétének a megállapitása.

Ha C (i) - (iv) -n felül még egy password első, w részét is ismeri, akkor miután valahogyan meghatározta, hogy az $f_1(w)$-ik mátrixnak a password directory-ban melyik mátrix lehet valamilyen kitevős hatványa, például egy R feletti mátrix alapu logaritmus kiszámitásával kaphat egy y értéket, és ezután meghatározva a

$$p'(v') \equiv y \pmod{P'} \qquad (117)$$

kongruencia egy megoldását, az eredeti password-del egyenértékü (w, v') számpárt kap. Az utolsó két lépés

alkalmas választásokkal számitástechnikailag megoldhatatlanná tehető.

Ha C-nek az (i) - (iv) információkon felül még egy password második, v részét is sikerült megtudnia, akkor először ki kell számitania a

$$\hat{v} \equiv p'(v) \pmod{P'} \tag{118}$$

értéket, majd meg kell határoznia az $f_1(w)$ értéket (ezt legrosszabb esetben — próbálgatással — $P-1$ mátrix $\hat{v}$-edik hatványra emelésével megoldható, bár ez P-től és $\hat{v}$-től függően igen nehéz lehet). Végül pedig a

$$p(w') \equiv f_1(w) \pmod{P} \tag{119}$$

kongruencia egy megoldását kell megtalálnia, és igy egy, az eredeti password-del egyenértékü számpárt kap. Ez az utolsó lépés alkalmas $p(x)$ polinom esetén számitástechnikailag véghezvihetetlen.

Ha a password directory-ban szerepelnek a felhasználók nevei, akkor az utóbbi két esetben lényeges egyszerüsitésekkel élhetünk és ha C-nek sikerül egy számpárt előállitania, akkor az biztosan az eredeti password-del egyenértékü számpár. Ha azonban a password directory-ban nevek nem szerepelnek, akkor C végül ugyan olyan számpárt kap (ha sikerült előállitania), amely password, de előfordulhat, hogy nem annak a felhasználónak a password-jét számitotta ki, amelyiktől a password általa ismert része származott (ennek valószinüsége legfeljebb π).

Igy ekkor még vagy meg kell keresnie a személyhez a password "valódi másik részét", vagy meg kell keresnie a talált password-höz a hozzá tartozó felhasználót.

Látható, hogy sem az I) , sem a II) nem teszi még csak lehetővé sem az illetéktelen bekapcsolódás megkisérlését az $F(F(X))$ kiszámitása, azaz az $F(X)$ password-ként való alkalmazása alapján, minthogy az $F(X)$ érték nincs az F függvény értelmezési tartományában. (Az I) módszernél

$$F(A,e) = A^e \quad R \text{ felett;} \tag{120}$$

a II) módszernél

$$F(w,v) = (S(f_1(w))^{f_2(v)} \quad R \text{ felett,} \tag{121}$$

ahol $S(j)$ jelöli a mátrix- vagy eljárás-sorozatból a j-edik elem kiválasztását és eljárás esetén a megfelelő mátrix előállitását.)

2.3.3. MEGJEGYZÉSEK

A 2.3.Fejezetben leirt felhasználó-azonositó eljárások valóban megbizhatóak, ha feltételezzük (e feltétele-

zéssel a fejezet során — hallgatólagosan — mindvégig éltünk is), hogy a felhasználói terminál és a számitógép között az információ-áramlás védett csatornán át zajlik le. Ez a feltételezés helytálló, ha a felhasználó terminálja közel van a számitógéphez. Ekkor ugyanis elfogadható költséggel megvalósitható az információs csatorna megfelelő fizikai védelme. Ha azonban a felhasználói terminál és a számitógép távol van egymástól, akkor az őket összekötő információs vonal (telfonvonal) fizikai védelme rendkivül költséges lenne. Ezért ebben az esetben az előbbi feltételezés nem helytálló, hiszen egy illetéktelen személy most ugy is szert tehet egy felhasználó password-jére, hogy lehallgatja a felhasználói terminált a számitógéppel összekötő vonalat. Egyirányu függvények alkalmazásával ez a probléma is megoldható; részletesen lásd [48], [49].

A 2.3.Fejezetből és a felhasználó-azonositással foglalkozó dolgozatokból világosan kitünik egy, a felhasználók azonositásának megbizhatóságával kapcsolatos ellentmondás. Azért, hogy csökkentsük az illetéktelen beavatkozás lehetőségét, szükséges, hogy a felhasználók elővigyázatosak legyenek és pl. ne válasszanak könnyen kitalálható password-öket. Ezért a password-öknek hosszabbaknak és "értelmetlen"-eknek kell lenniük. Az illetéktelen bekapcsolódások ellen eddig kidolgozott módszerek legtöbbje szintén meglehetősen hosszu és rendszerint "értelmetlen" password-öket igényel a kielégitő megbizhatóság érdekében. (Sok rendszer megbizhatósága nő, ha

a password hossza növekszik.) Az ember biológiai korlátai miatt azonban a hosszu és "értelmetlen" password-öt a felhasználó kénytelen leirni vagy pedig egy/több adathordozón rögziteni. Ekkor pedig azonnal felmerül a password lemásolásának, elvesztésének, ellopásának veszélye. Következésképpen ismét elsődleges fontosságot kap a felhasználó elővigyázatossága. Tehát az ellentmondás abban áll, hogy a felhasználó akkor tud egy password-öt kényelmesen kezelni, ha a password-öt könnyen az emlékezetében tudja tartani, a rendszer pedig akkor képes nagy biztonságot nyujtani, ha a password hosszu és "értelmetlen". Jelenleg ugy tünik, hogy igen kevés remény van ezen ellentmondás teljes feloldására. (Részleges feloldását kiséreli meg a 2.3.2.Fejezetben az]) módszer — lásd még Márkus[49].) Ezért a müködő rendszerek megalkotásakor a konkrét helyzet felmérése után mérlegelni kell a felmerülő problémák lehetséges következményeit, és ezután kell létrehozni egy, az adott lehetőségek között optimálisnak igérkező rendszert.

ÖSSZEFOGLALÁS

A számitógépek széles körü elterjedésének egyenes következménye azok felhasználása a kriptológia területén. A mikroszámitógépek térhóditása nagy mértékben felgyorsitja ezt a folyamatot és lehetővé teszi a modern kriptográfiai módszerek alkalmazását a mindennapi életben. Ezért fontos uj, megbizható kriptográfiai eljárások kidolgozása.

1. A kriptográfia modern elméletének egyik kulcs fontosságu fogalma az *egyirányu függvény*. A dolgozat első részében leirunk egy uj egyirányu függvényt:az egységelemes gyürü feletti mátrixhatványozást.

1.1. A dolgozat fő eredménye egy *algoritmus-konstrukció* egységelemes gyürü feletti mátrix alapu logaritmus kiszámitására (1.2. Fejezet). Az algoritmus az eddig ismert leghatékonyabb, véges testekbeli logaritmus meghatározására szolgáló algoritmus általánositása.

1.2. Az általánositott algoritmus *idö- és memóriakomplexitására* vonatkozó eredményt a 3.Tétel tartalmazza. Rögzitett gyürü esetén algoritmusunk az alapmátrixtól (pontosabban annak különböző hatványai számától) függően lehet hatékony vagy akár rendkivül hatékonytalan is. Ez utóbbi esetben a mátrixhatványozás egyirányu függvény.

1.3. *Az alapmátrix* speciális tulajdonságu kell legyen: *hatványai között elö kell fordulnia a* megfelelő méretü *egységmátrixnak*. Véges rendü gyürük esetén ilyen tulajdonságu mátrixok előállitására vonatkozó eredményt tartalmaz a 4. Tétel és ennek közvetlen következményei.

1.4. Az alkalmazások szempontjából fontos, hogy az alapmátrixnak sok különböző hatványa legyen. Ilyen mátrixok konstrukciójára vonatkozó eredmény található az 1.3. Fejezetben.

1.5. Vizsgáljuk továbbá a kidolgozott algoritmus *általánositásának* lehetőségeit és *számitógépes implementálásának* problémáit különböző konkrét gyürük esetén.

2. A dolgozat második része a mátrix alapu logaritmus kiszámitására szolgáló algoritmus *kriptográfiai alkalmazásait tárgyalja.*

2.1. Az *1.2.* pontbeli egyirányu függvényt felhasználva kialakitottunk egy *konvencionális kriptorendszert,* amely akkor sem veszit jelentősen megbizhatóságából, ha korábban rejtjelezett üzenetek forrásszövegeit később nyilvánosságra hozzuk.

2.2. Kifejlesztettünk egy *nyilvános jelkulcselosztó rendszert,* amely a modern kriptográfiai eszközök közé sorolható.

2.3. Bár a *2.1.* és *2.2.* pontokban emlitett rendszerek egy-egy véges testekben kialakitott rendszernek az általánositásai, *a mátrixokra való általánositás* sok *uj lehetőséget rejt magában* (ezt jól mutatja a kulcselosztó rendszer egy alkalmazása, ld. Márkus [50]).

2.4. Az alkalmazások között fontos jelyet foglalnak el azok a rendszerek, amelyek a time-sharing számitógéprendszereknél *a felhasználók azonositását, autentikusságuk megállapitását* hivatottak elvégezni. A dolgozatban két ilyen uj rendszert irunk le.

2.4.1. Az első a gyürü feletti mátrixhatványozást, mint egyirányu függvényt használja és lehetőséget nyujthat pl. "értelmes" password-ök alkalmazására megbizható rendszerekben.

2.4.2. A második rendszer nemcsak a gyürü feletti mátrixhatványozást használja egyirányu függvényként, hanem az egész számokon értelmezett polinom-kongruenciákat is. Igy egy olyan rendszer áll elő, amelyben egy password egy olyan számpár, amelynek bármely tagja önmagában is képes garantálni a rendszer megbizhatóságát. Ez utóbbi rendszer arra is lehetőséget ad, hogy a password fileból elhagyjuk a felhasználók neveit — az ebből adódó esetleges problémák előfordulási valószinüségére *felső becslést adunk* (2.3.2. Fejezet).

2.5. Vizsgáljuk a felhasználó-azonositó eljárások *megvalósitásának lehetőségeit és korlátait*, különös tekintettel a *2.4.1.* és *2.4.2.* pontokban leirt rendszerekre.

IRODALOM

1. DIFFIE,W. & M.E.HELLMAN, "New Directions in Cryptography." *IEEE Trans.Inform.Theory IT-22 (1976), 644-654.*

2. DIFFIE,W. & M.E.HELLMAN, "Privacy and Authentication: An Introduction to Cryptography." *Proc.IEEE 67 (1979), 397-427.*

3. EHRSAM,W.F. - S.M.MATYAS - C.H.MEYER - W.L.TUCHAM, "On cryptographic key management scheme for implementing the Data Encription Standard." *IBM Syst. J. 17 (1978), 106-125.*

4. KAHN,D., "Modern Cryptology." *Scientific Amer. 215 (July 1966), 38-46.*

5. KAHN,D., The Codebreakers, The Story of Secret Writing. *New York, Macmillan, 1967.*

6. RÉVAY Z., **Titkosirások.** *Zrinyi Katonai Kiadó, Budapest, 1978.*

7. ANDELMAN,D. & J.REEDS., "On the Cryptanalysis of Rotor Machines and Substitution-Permutation Networks." *IEEE Trans. Infor. Theory IT-28 (1982), 578-584.*

8. KAHN,D., "Cryptography Goes Public." *IEEE Comm. Soc. Mag. 18 (March 1980), 19-28.*

9. KOLATA,G.B., "Cryptography: On the Brink of a Revolution?" *Science 197 (1977), 747-748.*

10. FEISTEL,H., "Cryptography and Computer Privacy." *Scientific Amer. 228 (May 1973), 15-23.*

11. FEISTEL,H. - W.A.NOTZ - J.L.SMITH, "Some Cryptographic Techniques for Machine-to-Machine Data Communications." *Proc. IEEE 63 (1975), 1545-1554.*

12. GEFFE,P.R., "How to protect data with ciphers that are really hard to break." *Electronics (Jan.1973), 99-101.*

13. DIFFIE,W. & M.E.HELLMAN, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard." *Computer (June 1977), 74-84.*

14. RIVEST,R.L. - A.SHAMIR - L.ADLEMAN, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." *Comm. ACM 21 (1978), 120-126.*

15. MERKLE,R.C., "Secure Communication Over Insecure Channels." *Comm. ACM 21 (1978), 294-299.*

16. BABAI L., "Primszámok és titkosirás." *Természet Világa 112 (1981), 250-253.*

17. EVANS,A.Jr. - W.KANTOROWITZ - E.WEISS, "A User Authentication Scheme Not Requiring Secrecy in the Computer." *Comm. ACM 17 (1974), 437-442.*

18. PURDY,G.B., "A High Security Log-in Procedure." *Comm. ACM 17 (1974), 442-445.*

19. BERKOVITS,S. - J.KOWALCHUK - B.SCHANNING, "Implementing Public Key Scheme." *IEEE Comm. Soc. Mag. 17 (May 1979), 2-3.*

20. GAIT,J., "Can the MITRE Public Key System Be Short-Cycled?" *IEEE Comm. Soc Mag. 17 (Nov. 1979), 2.*

21. BERKOVITS,S., "Cycling is Just a Random Search." *IEEE Comm. Soc. Mag. 17 (Nov, 1979), 2-3.*

22. POTTER,R.J., "Electronic Mail." *Science 195 (1977), 1160-1164.*

23. WILLIAMS,H.C., "A Modification of the RSA Public-key Encryption Procedure." *IEEE Trans. Inform. Theory IT-26 (1980), 726-729.*

24. MERKLE,R.C. & M.E.HELLMAN, "Hiding Information and Signatures in Trapdoor Knapsacks." *IEEE Trans. Inform. Theory IT-24 (1978), 525-530.*

25. SHAMIR,A. & R.E.ZIPPEL, "On the Secirity of the Merkle-Hellman Cryptographic Scheme." *IEEE Trans. Inform. Theory IT-26 (1980), 339-340.*

26. ARAZI,B., "A Trapdoor Multiple Mapping." *IEEE Trans. Inform. Theory IT-26 (1980), 100-102.*

27. BRASSARD,G., "A Note on the Complexity of Cryptography." *IEEE Trans. Inform. Theory IT-25 (1979), 232-233.*

28. "Digitális bünözés." *Ötlet 1 (1982 julius), 15.*

29. TURN,R. & W.H.WARE, "Privacy and Security in Computer Systems." *Amer. Scientist 63 (1975), 195-203.*

30. KNUTH,D.E., The Art of Computer Programming. Vol.2 Seminumerical Algorithms. *Addison-Wesley, Reading, Mass. 1969.*

31. AHO,A.V. - H.E.HOPCROFT - J.D.ULLMAN, The Design and Analysis of Computer Algorithms. *Addison-Wesley, Reading, Mass. 1975.*

32. PAN,V.YA., "New Fast Algorithms for Matrix Operations." *SIAM J. Comp. 9 (1980), 321-342.*

33. KNUTH,D.E., The Art of Computer Programming. Vol.3 Sorting and Searching. *Addison-Wesley, Reading, Mass. 1973.*

34. POHLING,S.C. & M.E.HELLMAN, "An Improved Algorithm for Computing Logarithms over GF(q) and Its Cryptographic Significance." *IEEE Trans. Inform. Theory IT-24 (1978), 106-110.*

35. KEYES,R.W., "Physical limits in digital electronics." *Proc. IEEE 63 (1975), 740-767.*

36. LANDAUER,R.W., "Irreversibility and heat generation in the computing process." *IBM J. Res. Develop. 5 (1961), 183-191.*

37. WILKES,M.V., Time-Sharing Computer Systems. *New York: American Elsevier, 1968.*

38. HELLMAN,M.E., "A Cryptanalytic Time-Memory Trade-Off." *IEEE Trans. Inform. Theory IT-26 (1980), 401-406.*

39. HARDY,G.H. & E.M.WRIGHT, An Introduction to Theory of Numbers. *Oxford Press, New York, 1960.*

40. GYARMATI E. & TURÁN P., Számelmélet. (Kézirat) *Tankönyvkiadó, Budapest, 1975.*

41. ZIMMER,H.G., Computational Problems, Methods and Results in Algebraic Number Theory. *Lecture Notes in Math. Vol. 262 , Springer-Verlag, New York, 1972.*

42. BERLEKAMP,E.R. "Factoring polynomials over large finite fields." *Mathematics of Comput. 24 (July 1970).*

43. WILLIAMS,H.C., "Primality Testing on a Computer." *Ars Combinatoria 6 (1978), 127-185.*

44. POLLARD,J.M., "Theorems on factorisation and primality testing." *Proc. Camb. Phil. Soc. 76 (1974), 521-528.*

45. SOLOVAY,R. & V.A.STRASSEN, "Fast Monte-Carlo test for primality." *SIAM J. Comp. 6 (1977), 84-85.*

46. PANTAGES,A., "The Price of Protection." *Datamation (March 1976), 143-144.*

47. JÄGERMANN,T., "Az adatvédelem bennünket is kötelez I-II." *Számítástechnika 13 (1982 február), 12; Számítástechnika 13 (1982 március), 12.*

48. LAMPORT,L., "Password Authentication with Insecure Communication." *Comm. ACM 24 (1981), 770-772.*

49. MÁRKUS G., Megbizható felhasználó-azonositó eljárások. *Working Paper AP/21 , MTA SzTAKI, Budapest, 1983.*

50. MÁRKUS G., A Public Key Distribution System and Its Application. *Kézirat 1983.*

1984-BEN JELENTEK MEG:

155/1984 Deák, Hoffer, Mayer, Németh, Potecz, Prékopa, Straziczky: Termikus erőműveken alapuló villamos-energiarendszerek rövidtávu, optimális, erőművi menetrendjének meghatározása hálózati feltételek figyelembevételével.

156/1984 Radó Péter: Relációs adatbáziskezelő rendszerek összehasonlitó vizsgálata

157/1984 Ho Ngoc Luat: A geometriai programozás fejlődései és megoldási módszerei

158/1984 PROCEEDINGS of the 3rd International Meeting of Young Computer Scientists.
Edited by: J. Demetrovics and J. Kelemen

159/1984 Bertók Péter: A system for monitoring the machining operation in automatic manufacturing systems

160/1984 Ratkó István: Válogatott számitástechnikai és matematikai módszerek orvosi alkalmazása

161/1984 Hannák László: Többértékü logikák szerkezetéről.

162/1984 Kocsis J. - Fetyiszov V.: Rugalas autamatizált rendszerek: megbizhatóság és irányitási problémák

163/1984 Kalavszky Dezső: Meleghengermüvi villamos hurokemelő hajtás vizsgálata

164/1984 Knuth Előd: Specifikációs adatbázis modellek

165/1984 Petróczy Judit: Publikációk 1983

1985-BEN EDDIG MEGJELENTEK:

166/1985 Radó Péter: Információs rendszerek számitógépes tervezése

167/1985 Studies in Applied Stochastic Programming I. Szerkesztette: Prékopa András

168/1985 Böszörményi László - Kovács László - Martos Balázs Szabó Miklós: LILIPUTH

169/1985 Horváth Mátyás: Alkatrészgyártási folyamatok automatizált tervezése

COOPRINT 85-243