Alkalmazott Matematikai Lapok **23** (2006), 405–432.

ALAPVETŐ MATEMATIKAI TRANSZFORMÁCIÓK A KRIPTOGRÁFIÁBAN

TÓTH MIHÁLY

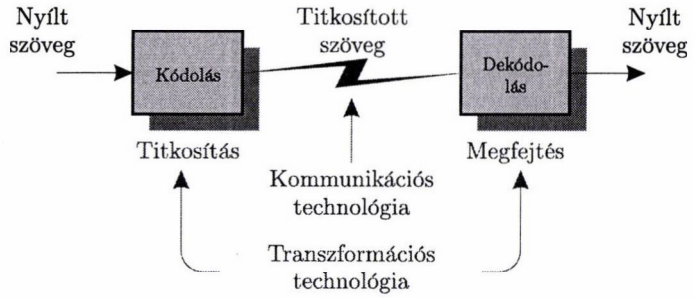
Budapest

Ez a cikk a kriptográfiai rendszerek ún. generációkba sorolásával, valamint a transzformációk általános jellemzőivel, tulajdonságaival foglalkozik. Heurisztikus módon megfogalmazza e transzformációk általános követelményeit és ezek alapján kísérletet tesz a kriptográfiai rendszerek formálnyelvi definíciójára. Tisztázza a kriptográfiában alkalmazott ún. blokk fogalmát és ennek segítségével a linearitásra is ad egy definiciót. Megfogalmaz egy sejtést, amely szükséges és elegendő feltétel a klasszikus helyettesítési és permutációs transzformációk felcserélhetőségére, valamint összevonására. Foglalkozik az egyszerű leképezőfüggvények inverz tulajdonságaival. Claude E. Shannon javasolta elsőként az ún. produkt transzformációkat, amelyek valójában vagy nem kommutatív produktumok, vagy összetett függvények. Minden esetre komoly szerepük van a modern, ún. negyedik generációs, szimmetrikus kriptorendszerekben. A cikk kitér az ún. Feistel transzformációkra és alkalmazásaikra, a többkomponensű transzformációk komponensei invertálhatóságának a kérdésére s végül az ún. aszimmetrikus (nyíltkulcsú) kriptorendszerekben alkalmazott transzformációk alapelveire.

Titkosítási rendszerek (kriptorendszerek) generációi

Jóllehet ma már nem szokás a számítástechnika (illetve a számítógépek) és alkalmazásaik újabb s újabb eredményeit egy-egy új „generációként" emlegetni, érdemes arra emlékezni, hogy korábban a számítógépek ún. generációit lényegében az alkalmazott technika határozta meg. A kriptorendszerek generációinál egészen biztosan nagy szerepe van az alkalmazott titkosítási/megfejtési módszereknek, technikának, közös néven az ún. *transzformációs módszereknek,* (matematikai terminológiával: leképezéseknek) de nagy szerepe van az üzenet továbbítási technikájának is, vagyis a *kommunikációs technológiának és a protokolloknak* is.

Eszerint a kriptogenerációkat két technológia együttesen határozza meg. Nevezetesen a transzformációs és a kommunikációs technológia. Valahogyan úgy, ahogyan ezt az 1. ábra szemlélteti.



1. ábra. Kriptorendszerek generációinak osztályozási szempontjai

GENERÁCIÓ	JELLEMZŐJE	TRANSZFORMÁCIÓS	KOMMUNIKÁCIÓS
		TECHNOLÓGIA	
1.	Monoalfabetikus	Helyettesítés Nagyon ritkán: keverés	– Nem jellemző
2.	Polialfabetikus és blokkos	Helyettesítés (kézi módszerrel) KULCSSZÓ	Nem jellemző
3.	Mint az előző, de *nagyon sok* ábécével	Betűnkénti helyettesítés (elektro) mechanikus (pl. rotoros) géppel	Rádió kommunikáció
4.	Produkt-transzformációk sok rundban. Igen nagy kulcstér. **Polialfabetikus** rendszerek.	Számítógéppel végzett iteratív transzformációk, amelyeknek nem létezik a nyers erő módszerénél gyorsabb, kulcs nélküli, algoritmusos módszere.	Kommunikáció a világhálón, magánhálózatokon, vagy legalább virtuális magánhálózatokon
5.	Aszimmetrikus kriptorendszerek. A kódolás és a dekódolás azonos leképzésekkel, de inverz kulcsokkal történik. **Monoalfabetikus** rendszerek	Számítógéppel, vagy célhardverrel megvalósított, számításigényes leképezések igen nagy (1000 bitnél nagyobb) kulcsokkal és ennek megfelelő kulcstérrel	Kommunikáció a világhálón, magánhálózatokon, vagy legalább virtuális magánhálózatokon

A transzformációk[1]

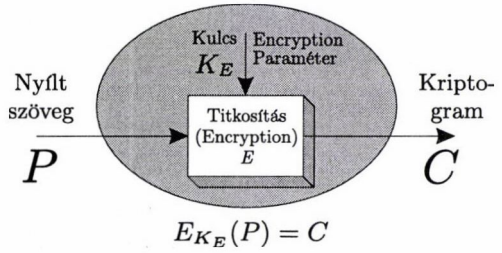
A titkosítás nélküli, ún. nyílt szöveget (P) valamilyen módon a be nem avatottak számára szándékoltan érthetetlen szimbólum-sorozattá, ún. kriptogrammá[2] (C) alakítja át a titkosítás.

[1] Szigorúbb matematikai értelemben leképezésekről van szó.

[2] A kriptogram aránylag új elnevezés. Korábban a francia eredetű *siffre* jelentette a titkosított szöveget és a sifrírozás a titkosítást.

A nyílt szövegnek is és a kriptogramnak is külön-külön ábécéje van, sőt a kriptogramnak több ábécéje is lehet, s a kriptogram ábécé meg is egyezhet a nyílt szöveg ábécéjével. Aszerint beszélünk egy– vagy többábécés kriptorendszerről, hogy hány ábécét használ a kriptogram.

A nyílt szöveget egy *transzformáció* képezi le a kriptogramba. Ezt teheti úgy is, hogy a nyílt szöveg minden egyes betűjét külön-külön transzformálja a kriptogram egyes betűivé, és teheti úgy is, hogy egy meghatározott hosszúságú betűcsoporton egyszerre hajt végre valamilyen transzformációt. Az előbbit *folyamatos titkosításnak* (Stream Ciphering) az utóbbit pedig *blokk titkosításnak* (Block Ciphering) nevezik.



2. ábra. A titkosító transzformáció

Ha a nyílt szöveg betűit egy-egy bájt helyettesíti, akkor nincs is lényeges különbség a betűnkénti és a blokkos titkosítás között, csak a blokkok mérete különböző.

Az ábécékből kiindulva két, alapvető transzformáció típust szokás megkülönböztetni, nevezetesen *a helyettesítést* és *a keverést*. Az előbbi mind a betűnkénti, mind blokkos titkosítás esetében alkalmazható, az utóbbi csak blokkokra. Számos érdekes példát ír le –többek között – *David Kahn* alapműve [15].

A transzformációk ezen archetípusai nem is olyan nagyon különbözőek, mint azt sokáig gondolták. Mindenesetre a bináris rendszerekben ezek a különbözőségek eléggé összemosódnak [24]. Feltétlenül meg kell említeni, hogy a kezdetektől fogva diszkrét ábécé vagy blokkok transzformációjáról volt szó. Jóval azelőtt, hogy a digitális, illetve diszkrét rendszerek olyan széles körben elterjedtek volna, mint manapság tapasztalhatjuk azt.

Kézenfekvő matematikai modellek voltak a diszkrét algebra, illetve matematika egyes fejezetei, ide értve a számelméletet is. A titkosítás végül is ilyen diszkrét elemek *leképezését* jelenti, s ez az, amit alább részletezünk is.

Itt kell megemlíteni azt is, hogy a titkosító rendszerek igen korai feltalálói között is van, aki már a XVI. század vége felé tudatosan alkalmazott ilyen matematikai modellt.

Blaise De Vigenere-ről [1523–1596], a „Látnokról" van szó, aki lényegében a Caesar-féle monoalfabetikus titkosítást fejlesztette tovább és az általa alkalmazott matematikai modell a modulo n összeadás, illetve kivonás volt, ahol n az ábécé betűinek a száma volt: [15], [21] és [22].

A leképező függvények is diszkrét véges elemű, vagy legalább is megszámlálhatóan végtelen sok elemű halmazok felett értelmezettek, és soha nem merült fel, hogy a leképező függvények esetleg folytonos függvények is lehetnének.

Nem merült fel az sem, hogy a transzformációkra formálnyelvi, illetve absztrakt algebrai modelleket állítsanak fel, bár elvileg ezek is célravezetők lehetnének (diszkrét esetekben is).

A formális nyelvek és az absztrakt automaták modelljeinek alkalmazhatósága más kérdés. Ezek ugyanis olyan hosszúságtartó (vagy pl. a GSM és a Turing–gép esetében nem hosszúságtartó) leképezéseket hajtanak végre, amelyek nagyon is alkalmazhatók lennének a kriptográfiai leképezések modellezésére. Egy indirekt kivételtől eltekintve azonban nem találkoztam ilyen jellegű publikációkkal.

A kivétel *Arto Salomaa*, akinek a fő kutatási területe éppen a formális nyelvekkel és automatákkal foglalkozik [26], de az utóbbi években a matematikai bonyolultság elmélettel (és a kiszámíthatósággal) is kapcsolatba hozta ezt a kutatási területet [27], sőt kifejezetten az aszimmetrikus kriptorendszerekkel kapcsolatban is publikált [28].

Alább még visszatérek erre a kérdésre.

A pontosabb formális leírásnak azonban több akadálya is van. Itt talán elegendő csak annyit megemlíteni, hogy ahányféle titkosító leképzés van, annyiféle leképzési szabály, ezért általános képzési szabályokat nagyon nehéz adni. Vannak azonban emellett más problémák is.

A kriptográfiai leképezések alapkövetelményei

A kriptográfiai transzformációk alkalmas matematikai modelljei tehát a *leképezések*.

Heurisztikus módon kikövetkeztethető, hogy a kriptográfiai transzformációknak milyen feltételeket kell kielégíteniük. Ezek a következők:

a. Mivel mind a nyíltszöveg ábécé, mind a kriptogram ábécé véges halmazok[3], ezért a leképező függvény véges halmazt képez le véges halmazra.
b. A leképező függvény egyértékű és
c. ha egyértelműen visszafejthetőnek kell lennie (márpedig annak kell lennie), akkor léteznie kell a leképezés inverzének is, továbbá
d. az inverz leképezésnek is egyértékűnek kell lennie,
e. a c. és a d. feltételekből pedig az következik, hogy a titkosítási leképezés kölcsönösen egyértelmű (bijektív) kell, hogy legyen.[4]

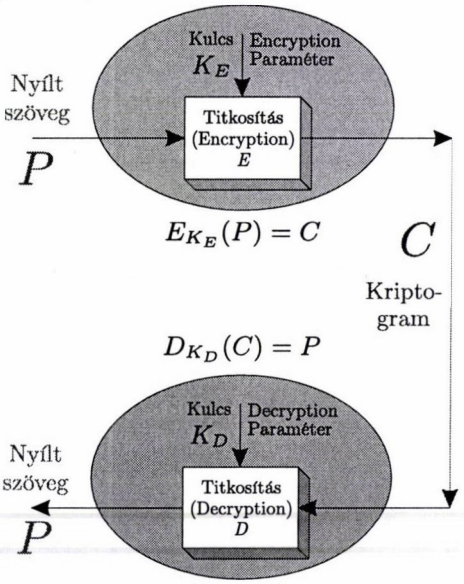
A felsorolt követelmények nagyon logikusnak tűnnek, de ezek alól a szabályok alól is van kivétel. A modern kriptográfiában ugyanis találhatók ún. valószínűségi

[3] ... és az ezekből az elemekből alkotott véges hosszúságú blokkok (*string*) száma is véges

[4] Létezik olyan kriptográfiai transzformáció is, amely a nyílt ábécé elemeit a képtartomány egy-egy valódi részhalmazára képezi le, de ezek a részhalmazok diszjunktak és a leképző függvényre ilyen kiterjesztéssel is érvényesek a mondott feltételek.

kriptorendszerek is, amelyeknél a leképezésekhez valószínűségek tartoznak, s véges (bár nagyon kicsi) valószínűséggel előfordulhat az is, hogy egy titkosított üzenet nem fejthető vissza. Némely alkalmazásnál (pl. titkosított beszéd-átvitel), amely redundáns információt továbbít, ez az információvesztés nem okoz gondot.

A felsorolt öt feltétel tehát szükséges, de még nem elégséges feltétel. Ezért a leképezések tulajdonságaira alább még visszatérünk.



3. ábra. A titkosítás teljes folyamata

A módszer és a kulcs

Vannak még a gyakorlati alkalmazhatóságból következő feltételek is. A leképezés bonyolultságát, a hozzá szükséges számítási kapacitást (és/vagy idő-igényt), valamint az algoritmus gyorsaságát már említettük.

Régi tapasztalati tény, hogy ha sokszor használják ugyanazt a titkosító leképezést, akkor azt majdnem biztosan feltörik.[5] Ezért aztán időről-időre változtatni kell azt. Nagyon nehéz azonban magát a módszert változtatni, mert nem csak arról van szó, hogy egyre újabb s újabb módszert kell kitalálni, hanem arról is, hogy ha a módszer alkalmazásához már gépet is szerkesztettek, akkor minden egyes módszerváltáskor a titkosító/megfejtő gép helyett is újat kell készíteni. Ezért már a XVII.

[5] Tulajdonképpen ilyen feltöréseknek tekinthetők a rég elfeledett ókori írások megfejtése, pedig több ilyen esetben maga az írás nyelve sem volt ismert. Minden ilyen sikeres megfejtés alapvető feltétele volt azonban, hogy *sok* írott szöveg állt az elemzők rendelkezésére.

században felmerült és a XIX. században *expressis verbis* meg is fogalmazták, hogy egy titkosítási transzformáció erőssége[6] végső soron az abban alkalmazható kulcsok számától függ: minél több a lehetséges kulcsok száma, azaz minél nagyobb a *kulcstér*, annál nehezebb vagy reménytelenebb a kulcs ismerete nélkül hozzájutni a titkosított információhoz.[7]

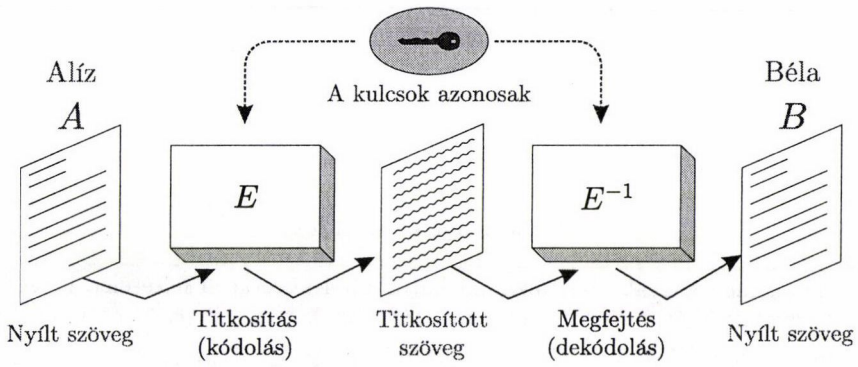
A titkosító transzformációk közös és általános tulajdonsága, hogy a leképző függvénynek van legalább egy *paramétere*, amely ismerete nélkül a leképzés nem valósítható meg. Ugyanez igaz a transzformáció inverzére, azaz a titkosított információ visszafejtésére is. Ez a paraméter a titkosítás *kulcsa*. Ha a kulcstér gyakorlatilag igen nagy, akkor maga a leképezés akár közzé is tehető, mert a kulcs önmagában garantálja a titkosítás biztonságát. Ez, nevezetesen az extrém nagy kulcstér, valamennyi mai kriptorendszer alapvető és közös elve.

A szimmetrikus és az aszimmetrikus kriptorendszerek összevetése

Valamennyi tradicionális kriptorendszer ugyanazt a kulcsot alkalmazta mind a titkosításhoz, mind a visszafejtéshez, pedig mint ma már tudjuk, ennek nem kell feltétlenül így lennie.

Vizsgáljuk meg ehhez a teljes titkosítás–visszafejtés folyamatot!

Az E_{K_E} és a D_{K_D} leképező függvényeknek *a paramétereikkel együtt* kell egymás inverzeinek lenniük.



4. ábra. Szimmetrikus (tradícionális) kriptorendszer blokkvázlata

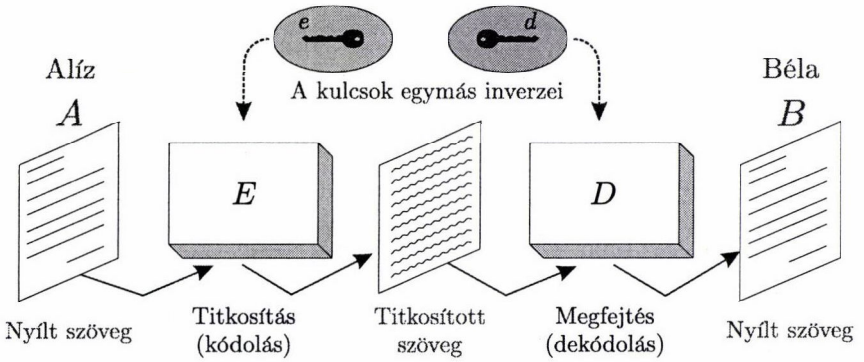
[6] Pontosabb elemzéssel alább megmutatjuk, hogy a kriptogram ábécé (vagy ábécék), mint halmazok rangja alapvetően meghatározó a kriptorendszer feltörése szempontjából és a lehetséges kulcsok számát is az ábécé(k) számossága korlátozza.

Egy kriptorendszer erősségét a feltörésének a nehézsége határozza meg. Ma már nem tekintik erősnek az olyan titkosításokat, amelyek algoritmusos módon feltörhetők. Az erős kriptorendszerek csakis úgy törhetők fel, hogy valamennyi lehetséges kulcsot végig kell próbálgatni. Ezt a *nyers erő* módszerének nevezik.

[7] Alább majd megmutatjuk, hogy elsődlegesen a kriptogram ábécék elemszáma a meghatározó. A kulcstér „mindössze" illeszkedik ehhez.

A tradicionális kriptorendszereknél magától értetődőnek tekintették, hogy a K_E és a K_D kulcsok *azonosak,* és az imígyen azonos paraméterekkel rendelkező E és D leképező függvények egymás inverzei.

Az ilyen rendszereket ma *szimmetrikus kriptorendszereknek* nevezzük és kizárólagos alkalmazásuk az ősidőktől kezdve az 1970-es évek közepéig tartott, de a mai legmodernebb kriptográfiában is megvan a jelentőségük (mármint a modern változataiknak).



5. ábra. Az aszimmetrikus kriptorendszer blokkvázlata

A transzformációk (azaz leképezések) egymás inverzei, *a kulcsok* pedig *azonosak.*

Tulajdonképpen kézenfekvő megoldás az is, hogy nem a leképző függvények egymás inverzei, hanem a titkosító és a visszafejtő kulcsok vannak valamilyen műveletre nézve inverz viszonyban egymással. Erre az ötletre azonban Whitfield Diffie előtt (1976) senki sem jött rá.

Ezekkel a rendszerekkel majd alább, az *aszimmetrikus kriptorendszerek* kapcsán foglalkozunk. Egy szimmetrikus kriptorendszerben tehát egyetlen kulcsot alkalmaznak, amelyet természetesen titokban kell tartani. Mondhatjuk úgy is, hogy ettől függ a rendszer biztonsága.

A titkosító (E) és a visszafejtő (D) algoritmusok, illetve függvények egymás inverzei, de általános esetben nem azonosak. Kérdés, hogy lehet-e olyan, elegendően biztonságos, titkosító rendszert szerkeszteni, amelyben nem csak a kulcsok, hanem ez a két függvény is azonos.

Nos, lehet, bár a modern, ún. erős kriptorendszerek esetében csak valami hasonló, de nem egészen azonos dolgot.

Mivel az inverz algebrai fogalma mindig csakis egy adott műveletre vonatkozik, az aszimmetrikus rendszerekben az E leképző függvény az, amelyre nézve az e és a d kulcsok egymás inverzei.

Ezt a két ábrát éppen azért tüntettük fel így, egymás alatt, hogy felhívjuk a figyelmet a szimmetrikus és az aszimmetrikus kriptorendszerek – jobb szó híján – szimmetriájára.

A tradicionális kriptorendszerek körében azonban megvalósítható, hogy egy titkosító-megfejtő függvénypár két tagja azonos legyen, vagyis egy olyan leképző függvényt találjunk, amely saját maga inverze. Elvileg az aszimmetrikus kriptorendszerek esetében is szerkeszthető olyan, amelyben a „két" kulcs azonos és saját maga inverze, de ez határeset, amikor csakis a leképezőfüggvény jellegéből állapítható meg, hogy a kriptorendszer szimmetrikus vagy aszimmetrikus-e. Az ilyen rendszernek azért nincs is gyakorlati jelentősége, mert az aszimmetrikus rendszerek transzformációinak a számítási kapacitás-igénye nagyságrenddel nagyobb, mint az azonos kriptográfiai erősségű szimmetrikus rendszereké, és értelmetlenül gazdaságtalan lenne sokkal nagyobb költséggel megvalósítani egy szimmetrikus funkciójú rendszert, mint azt a szimmetrikus algoritmusokkal lehet.

A transzformációk tulajdonságai

Leképezések vagy függvények?

Az eddigiekben utaltunk ugyan arra, hogy a *transzformáció* szigorúan vett matematikai értelemben nem azonos a kriptográfiában alkalmazott *leképezésekkel*, de ezt az állítást nem vizsgáltuk. Nos ennek itt a helye.

A kriptográfiai függvények (nyílt) ábécéből (titkos) ábécébe képező transzformációk. Matematikai értelemben azonban transzformációnak olyan *leképezést* nevezünk, amely egy halmazt saját magába képez le. Általános esetben a kriptográfiai leképezésektől nem követeljük meg, hogy értelmezési tartományuk és értékkészletük megegyezzen; de még azt sem, hogy akár csak tartalmazzák egymást.

Másrészt nem árt tisztázni, hogy tulajdonképpen mi is ezeknek a kriptográfiai leképezéseknek az értelmezési tartománya és az értékkészlete.

(Az itt következő diszkusszió során megkísérlem a kriptográfiai leképezések egy – nagyon egyszerű – formálnyelvi megközelítését.)

A kriptográfiai leképezések egy véges szimbólumhalmazból (nyílt ábécé) alkotott véges sorozatok halmazából képeznek egy vagy több véges szimbólumhalmaz (titkos ábécék) uniójából képzett sorozatok halmazára.

Formálisan: egy φ leképzés kriptográfiai transzformációnak tekinthető, ha $\varphi : P^* \to C^*$ alakú,

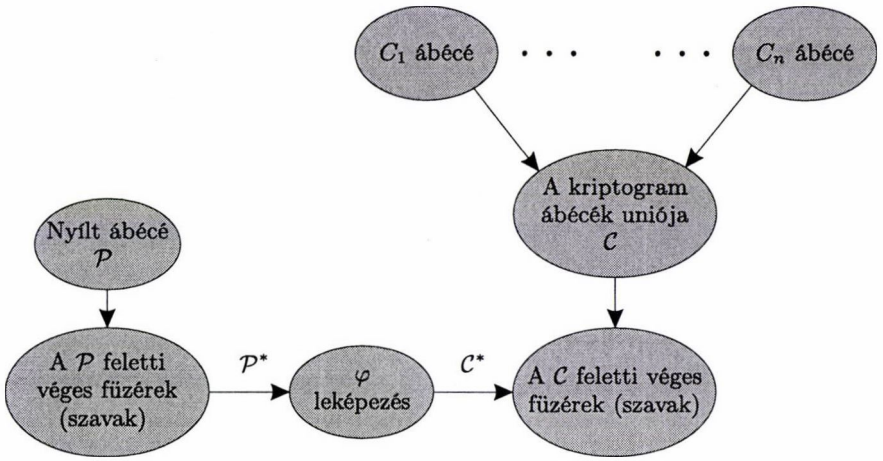
ahol P – input szimbólumok halmaza, nyílt ábécé,

P^* – P elemeiből alkotott füzérek halmaza

$C = \bigcup_{i=1}^{n} C_i$, output szimbólumok halmazainak (kriptogram ábécék) egyesítése,

ahol C^* – C elemeiből alkotott sorozatok halmaza.

A $C = \bigcup_{i=1}^{n} C_i$, definícióban szereplő n érték a kriptogram ábécék számát jelenti. A n=1 esetben *monoalfabetikus* titkosírásokról beszélünk Ismert $n > 1$ esetén és *első fajú polialfabetikus* titkosírásoknak nevezhetjük őket, míg abban az esetben, ha az ábécék száma – kívülállók számára – nem ismert, akkor nevezhetjük



6. ábra. Kriptográfiai leképzések
Polialfabetikus (kriptogram) ábécék

azokat *másodfajú polialfabetikus* titkosírásoknak.[8] Az első- és a másodfajú polialfabetikus rendszerek közti lényegi megkülönböztetést indokolja, hogy az első fajú többábécés rejtjelezések megfejthetők[9] betűgyakorisági analízis segítségével, míg a másodfajú polialfabetikus titkosírások nem. A de Vigenere kód esetében Babbage zsenialitása éppen abban nyilvánult meg, hogy módszert talált arra, hogy hogyan alakítson át egy másodfajú polialfabetikus rendszert első fajú rendszerré, azaz hogyan határozza meg az ábécék számát vagy – ezzel lényegében azonos jelentéssel – az alkalmazott blokk hosszát.

Ez a módszer – nevezetesen a dupletek ismétlődési távolságának a megkeresése és e távolságok legnagyobb közös osztójának a meghatározása – azonban távol áll a direkt megfejtésre való betűgyakoriság analízistől. Jó arra, hogy visszavezesse a megoldási problémát az első fajú polialfabetikus rendszerekére, de máskülönben nem része a megoldásnak. (Erre utaltunk a 6. lábjegyzetben is.)

Babbage előtt több mint 300 évig feltörhetetlennek tartották de Vigenere titkosítási módszerét, és az is volt a Babbage-féle visszavezetés ismerete híján. A sors és iróniája, valamint a brit katonai bürokrácia vaskalapossága, hogy Babbage-nak nem engedték meg a visszavezetési módszere publikálását. (Ez aztán még kétszer megismétlődött: Turing Enigma feltörése és a Colossus megépítése, valamint a GCHQ nyíltkulcsú kriptorendszerének Diffie előtti felfedezése esetében is.)

A formális megközelítésből látszik, hogy a leképezések értelmezési tartománya és értékkészlete is *véges* vagy megszámlálhatóan végtelen halmaz, hiszen véges hal-

[8] Ezek itt bevezetett, a szakirodalomban nem ismert elnevezések.

[9] Ti. az egyszerű helyettesítéses rendszerekhez hasonlóan, amelyek archetipusa pl. a Caesar-féle titkosítás.

mazok elemeiből alkotott füzérek (*string*) halmazai[10]. A leképző függvények tehát legfeljebb megszámlálhatóan végtelen halmazok felett vannak értelmezve, „folytonos" halmazok feletti függvények értelemszerűen szóba sem jöhetnek.

A formálnyelvi leírás egyik alapproblémája már e ponton jelentkezik és azzal kapcsolatos, hogy mit tekintünk ábécének.

Itt – egyelőre – azoknak a szimbólumoknak a P halmazát, amelyből a nyílt üzenet szavait képezzük és értelemszerűen kiterjesztjük ezt a C kriptogram ábécé(k)re is. Ezek nyílvánvalóan véges ábécék.

Blokkrendszerű titkosítások esetén az ábécékből alkotott szavak hossza nem csak korlátozott, hanem egy megadott véges érték (az AES-nél pl. 128 bit). Ezekből ugyancsak végesszámú van (az AES esetében éppen 2^{128} db), és e halmaz véges volta miatt ezek is tekinthetők egy véges ábécé elemeinek.[11] Mivel minden gyakorlati esetben megegyezik a kriptogram és a nyílt szöveg blokkjainak a hosszúsága, (sőt ugyanez szokott lenni a kulcshossz is) ezért a mondottak a kriptogram ábécé(k)re is érvényesek. A leképezések vizsgálatakor egyáltalán nem mindegy, hogy az itt említett kétféle ábécé-értelmezés közül melyiket fogadjuk el. A 6. ábrán szemléletesen bemutatott halmazok és jelölésrendszer remélhetően világossá teszi az értelmezést. Az ábra a polialfabetikus rendszerek kedvéért mind az ábécéket, mind azok elemeiből alkotott szavak halmazait feltünteti.

A leképezések linearitása

A kriptográfiai leképezések általános vizsgálatakor a $P*$ nyíltszöveg füzérek közötti műveleteket és ezeknek a $C*$ képtartományban való megjelenését vizsgáljuk. Formálnyelvi megközelítésben tehát ekkor éppen a korábban említett második ábécé-értelmezést alkalmazzuk, vagyis azt, amikor a leképezés értelmezési tartományának (és értékkészletének is) az elemei a $P*$, illetve a $C*$ füzérek. Emlékeztetünk arra, hogy e diszkusszió során nem tekintjük e füzérek hosszát sem előre definiáltnak, sem azonosnak, legfeljebb felülről korlátosnak, hogy megszámlálható halmazokkal tudjunk számolni.

Hagyományosan egy leképzést akkor szokás lineárisnak nevezni, ha az értelmezési tartomány elemein értelmezett valamely műveletekre nézve műveletтartó, azaz mindegy, hogy a műveletet előbb elvégezzük az értelmezési tartomány elemei közt, majd az eredményt képezzük le, vagy előbb a leképzést az operandusokra alkalmazzuk, és a képelemek közt végezzük el a megfelelő[12] műveletet.

[10] Vagy véges sok véges halmaz uniójából alkotott füzérek halmaza

[11] Igaz, hogy ennél az ábécénél igen nagy az elemek száma. Ha azonban nem így lenne, akkor – az alkalmazott transzformáció bonyolultságától függetlenül – az elemek gyakoriságának az elemzésével tulajdonképpen nagyon könnyen feltörhető lenne a kriptogram.

[12] Hangsúlyozni kell a „megfelelő" szó jelentőségét a megfogalmazásban. Általános esetben ugyanis nem várható el, hogy a leképzés értelmezési tartománya és értékkészlete megegyezzen. Így a leképzés előtti művelet egy az értelmezési tartományon értelmezett művelet, míg a képelemeken végzett művelet az értékkészlet felett értelmezett művelet.

Formálisan: egy $\varphi : A \to B$ leképezés lineáris valamely $+$ és $\oplus$ műveletek szempontjából, ha

$$\forall a, b \in A : \varphi(a + b) = \varphi(a) \oplus \varphi(b),$$

ahol $+ : A \times A \to A$ és $\oplus : B \times B \to B$ megfelelő A és B feletti műveletek[13].

Kriptográfiai leképezések esetében a leképezés értelmezési tartománya egy adott szimbólumhalmaz elemeiből alkotott füzérek[14] halmaza. Hasonlóképpen az értékkészlet is füzérek halmaza, csak a füzérek halmaza nem ugyanaz, mint az értelmezési tartomány esetében. Sőt a füzéreket alkotó ábécék sem feltétlenül azonosak a nyílt P ábécé és a C kriptogram ábécé esetében.

Ilyen halmazokon természetes módon definiálhatjuk a füzérek összefűzésének műveletét (catenation, katenáció, konkatenáció,). Szokásos jelölése a $\|$ jel, de használatos még a $\hat{}$ jelölés is. Ezek szerint $a \| b$, illetve $a\hat{}b$ olyan szimbólum sorozatot jelöl, amelynek első tagjai az a füzér elemeivel egyeznek meg, a továbbiak pedig a b sorozat megfelelő elemeivel.

Kriptográfiai leképzések linearitásáról tehát az összefűzés művelet tekintetében beszélhetünk. Ezek szerint lineárisnak nevezünk egy $\varphi : A^* \to B^*$ kriptográfiai transzformációt, ha

$$\forall a, b \in A^* : \varphi(a \| b) = \varphi(a) \| \varphi(b).$$

A legtöbb kriptográfiai leképzés nyilván nem lineáris a fenti definíció szerint. Ez ugyanis azt jelentené, hogy a teljes titkos szöveg a nyílt szöveg külön-külön részeinek titkosításával áll elő, amelyeket a nyílt szövegnek megfelelő sorrendben fűzünk össze. Ez viszont lehetőséget nyújtana a támadónak arra, hogy ha sikerül az üzenet küldőjét rávennie, hogy ugyanazt az üzenetet jól körülhatárolható helyen történő változtatással küldje el, akkor a titkos szövegben történt változásból következtetni lehessen a kódolás miként jére, sőt magára a kulcsra is (*choosen plain text attack*).

Blokkhosszúság[15]

A következőkben megmutatjuk, hogy a blokkhosszúság fogalma a linearitás segítségével sokkal pontosabban meghatározható, mint az a mai, modern kriptorendszerekben szokásos. Ezeknél ugyanis blokknak nevezzük azt a szimbólumfüzért, amelyen a kriptotranszformációt végrehajtjuk. A vonatkozó szakirodalom többnyire megemlíti ugyan, hogy ezek a transzformációk nemlineárisak, de eleddig

[13] Vektorterek (lineáris terek) esetében nem csak az alaphalmazon belül értelmezett művelet művelettartását követelik meg, de az ún. skalárral való szorzás műveletre is előírják ugyanezt. Kriptográfiai leképzések esetén ilyen külső műveletet nem definiálunk, így ennek vizsgálatától eltekinthetünk.

[14] Algebrai értelemben itt szimbólum-sorozatokról van szó. Mégis, a gyakorlatban elterjedt szóhasználat miatt ezeket inkább füzéreknek (string) nevezzük vagy – a formálnyelvi megközelítésnél – az adott nyelv szavainak.

[15] A blokkhosszúság e definíciója új, eleddig ezt a megközelítést – tudtommal – nem használta a szakirodalom. Azért vezettük be, mert a leképezések linearitása, illetve nemlinearitása szempontjából hasznos.

sehol sem bukkantam rá a linearitás, illetve a nemlinearitás pontosabb definíciójára. Más szóval arra, hogy az említett kriptotranszformációk *miért,* illetve *mitől* nemlineárisak.

Technológiai szempontból megoldhatatlan, hogy egy kriptográfiai leképzés tetszőlegesen hoszszú szövegekre is nemlineáris módon viselkedjék. Minden gyakorlatban használt algoritmus esetében van egy a sorozatok hosszára vonatkozó felső korlát, hogy ha az ilyen hosszúságú sorozatokat tekintjük az ábécé elemeinek, akkor az ezekből alkotott sorozatok halmazán az összefűzés műveletére vonatkozóan a leképzés már lineáris. *Azt a legkisebb ilyen értéket, amelyre az adott leképzés lineárisan viselkedik, a leképzés* blokkhosszúságának *nevezzük.*

Vegyük észre, hogy itt nem a hagyományos értelemben vett „blokkonkénti" leképezés linearitásáról vagy nemlinearitásáról van szó, hanem épp fordítva:

A kriptográfiai leképzés linearitásának előbbi definíciója segítségével lehet a blokk fogalmát és a blokkhosszúságot definiálni.[16]

Nagyon leegyszerűsítve: Blokkok azok a legrövidebb füzérek, amelyekre, mint argumentumokra a kriptográfiai transzformációk linearitása teljesül.[17]

Formálisan: Legyen $\varphi : A^* \to B^*$ kriptográfiai leképzés, n természetes szám, továbbá $\varphi' : (A^n)^* \to B^*$ leképzés úgy, hogy $\forall s \in (A^n)^*$ sorozat esetén $\varphi(s) = \varphi'(s)$.

Ekkor azt a legkisebb n természetes számot, amelyre φ' lineáris, a φ leképezés *blokkhosszának* nevezzük.

Ehhez érdemes némi magyarázatot fűzni: φ az A ábécé elemeiből alkotott tetszőleges hosszúságú sorozatokon értelmezett függvény. A^n az A ábécé elemeiből alkotott pontosan n-hosszúságú sorozatok (blokkok) halmaza, az ezekből mint elemekből alkotott sorozatok alkotják $(A^n)^*$ elemeit. Az ezen a halmazon értelmezett φ' leképezés csak annyiban tér el a φ leképezéstől, hogy nincs feltétlenül bármilyen hosszúságú sorozaton értelmezve, csak azokon, amelyek hosszúsága a talált n szám egész számú többszöröse. Akkor mondjuk, hogy ez az n a leképzésre jellemző ún. blokkhossz, ha a leképzés a blokkokra megszorítva lineáris, és nincs olyan n-nél kisebb korlát, amelyre a leképzés ugyanezt tudná.

A mai, gyakorlatban alkalmazott kriptorendszerek mind meghatározott hosszúságú blokkokra osztják a nyílt szöveget és e blokkokat külön-külön transzformálják

[16] Ez a fajta linearitás nem is ad választ arra, hogy *mennyire* nemlineáris egy ilyen blokk leképezése.

Ez a kérdés pedig a gyakorlatban létezik. A DES esetében – a NIST javaslatára – éppen a „nemlinearitás növelésére" bevezették a rund transzformációk sorába az expanzió-kompreszszió párt. Közvetve az ún. lavinahatás növelése és ez által a feltörés megnehezítése miatt. Ez a fajta művelet azonban a későbbi szimmetrikus rendszerekben nincs meg, tehát a mondott okból nem is volt rá szükség. Ennek az okát viszont – tudtommal – nem publikálták, hanem „csak tudomásul vették", hogy a Feistel függvények szerinti iteratív transzformációk az expanzó-kompresszió pár nélkül is rendelkeznek a lavinahatással.

[17] Ennek a definíciónak az ötlete és a formális megfogalmazása Tóth Gergelytől, a Veszprémi Egyetem székesfehérvári AIFSz Képző Központjának a tanárától származik.

kriptogram blokkokká. Mind a nyílt szöveg, mind a kriptogram blokkjai ugyanolyan hosszúságúak s ezzel azonos a kulcs hosszúsága is.[18]

Nyilvánvaló, hogy

a. A blokk nem osztható, azaz nem alkalmazható a titkosítási leképezés részblokkokra. Ha a teljes titkosítandó szöveg blokkokra bontása végén részblokk adódna, akkor azt valamilyen előzetes megegyezés szerint fel kell tölteni (padding).
b. Az egymást követő blokkok titkosítási leképezése egymás után és egymástól függetlenül történik. Ilyen értelemben a blokkokból alkotott sorozat részeire (ti. az egyes blokkjaira) érvényes az előbbiekben megfogalmazott linearitás-definició.

Azonban ez alól is van kivétel, nevezetesen a szimmetrikus rendszerekben alkalmazott láncolás elve, amelyet ugyan a DES kapcsán vezettek be, de bármelyik „blokkos" titkosításnál alkalmazható [5].

Az utóbb említett ellenpélda szemlélteti, hogy láncolt titkosításnál a „blokk" fogalmát pontatlanul alkalmazzák az egymás után végrehajtott transzformációkra, amelyek ekkor egyáltalán nem függetlenek egymástól. Ilyen esetekben tulajdonképpen az egész transzformálandó nyílt szöveget kellene egyetlen oszthatatlan „blokknak" tekinteni.

A tükörszimmetrikus, öninvertáló transzformációk

E leképzés-típusoknak a szimmetrikus kriptorendszerek körében van jelentősége. Gyakorlati szempontból nagyon is jelentős dolog, hogy ha ugyanazzal a géppel (vagy algoritmussal) tudunk titkosítani és visszafejteni, a gép mindennemű átállítása nélkül. Így működött pl. a második világháborúban elhíresült német titkosító gép, az Enigma is, amelyet a maga idején megfejthetetlen kódolónak tartottak. Nos, végül is nem volt megfejthetetlen, de a feltöréséhez zsenik kellettek és nagyon időigényes feladatnak bizonyult.

Az Enigma egy érdekes tanulsága az, hogy a megfejtést az ábécék extrém nagy száma tette szinte lehetetlenné. Ez, ti. az ábécék száma még a legegyszerűbb, kereskedelmi változatnál is 26^3 volt, de a katonai változatoknál ezt kb. ezerszeresére növelték. Ráadásul a kulcstér is nagyon nagy (de az ábécék számától elvileg független) volt.

A titkosító és a visszafejtő algoritmus hasonló „tükörszimmetriája" fellelhető a modern iterációs kriptorendszerekben is – ha eltekintünk a kulcsütemezés megfordításától. Az Enigma a mindaddig legelterjedtebb titkosítási transzformációt alkalmazta, ti. a *helyettesítést.* A 26 betűs ábécéből képzett nyílt szöveg minden egyes betűjét egy kriptogram betűbe képezte le. A bonyolultságát azzal érték el, hogy egy meglehetősen hosszú betűsorozaton belül ugyanazt a nyíltszöveg betűt mindig más és más kriptogram betűnek feleltette meg.

[18] Egyetlen egy, részbeni kivétel a DES ún. rund transzformációján belül alkalmazott expanzió, majd a szubkulcs-művelet utáni kompresszió.

Itt most nem foglakozunk azzal, hogy ezt hogyan valósította meg az Enigma. Az egyik lényeges következmény az, hogy az ilyen tükörszimmetrikus és polialfabetikus helyettesítés messze nem használja ki a lehetséges kriptogram ábécék mindegyikét. Ezt azonban nagyon nagyszámú ábécé esetén „megengedheti magának" a kriptorendszer. Ezt a tükörszimmetriát megörökölték az Enigmától a modern iterációs kriptorendszerek is (DES, IDEA, AES...), de a visszafejtésnél alkalmazott fordított szubkulcs-sorozattal kiküszöbölték a tükörszimmetriának az Enigmánál még meglévő hátrányát.

A tükörszimmetrikus kriptotranszformációkat leképezéseknek tekintve teljesülnie kell annak a feltételnek is, hogy *a leképezés értéktartománya és értékkészlete azonos.* Más szóval a nyílt ábécé azonos a kriptogram ábécével – vagy ábécékkel, polialfabetikus rendszerek esetén[19].

Ekkor, és csakis ekkor lehetséges, hogy *a leképezés önmaga inverze.* Mint említettük, ennek gyakorlati szempontból van jelentősége, jóllehet csökkenti az aktuális kriptorendszer erősségét. Az Enigma esetében pl. egyfajta „fogódzót" jelentett a kódfejtőnek az, hogy az Enigma sohasem képzett le egy betűt önmagába. Ez a gép leképzési tükörszimmetriájának a következménye volt.[20]

A tükörszimmetrikus leképezést még a legegyszerűbb helyettesítő titkosításoknál is meg lehet valósítani. Egy sztenderd, n betűs ábécé és Caesar-féle titkosítás esetén azonban ilyenkor a kulcstér a felére csökken. Az n betűs kevert ábécék esetén pedig $(n-1)!$ helyett $(n/2)!$-ra csökken a kulcstér.

Az alap-transzformációk

Két alapvető transzformáció-típus létezik ősidők óta. Ezek a következők:

- helyettesítés (substitution),
- transzpozíció (más néven keverés vagy permutáció).

A helyettesítés alapértelmezésben betűt helyettesít betűvel, a transzpozíció pedig egy meghatározott hosszúságú szövegblokkon belül áthelyezi, összekeveri a betűket.[21]

Érdemes megjegyezni, hogy e két alaptranszformációnak nem csak történeti érdekessége van, hanem fellelhetők a legmodernebb titkosító rendszerekben is. Igaz, nem egymagukban, hanem összetett, beágyazott alkalmazásaikban.

Végül is mindkét alaptranszformáció megfogalmazható függvényként is, amelyek bijektív leképezést hajtanak végre.[22]

---

[19] Több-ábécés rendszerek és bonyolultabb – pl. iterációs – transzformációk esetében bonyolultabb feltételek is megfogalmazhatók. Az iterációs, szimmetrikus kriptorendszereknél pl. csakis úgy teljesül ez a „tükörszimmetria", hogy az inverz leképezésnél fordított sorrendben kell alkalmazni az egyes menetek (rundok) szubkulcsait, mint a titkosításnál. Ez más szóval az alkalmazott ábécék sorrendjének a megfordítását (is) jelenti.

[20] Valamint annak, hogy a valamennyi kriptogram ábécé páros számú betűből (26 betűből) állt.

[21] A kettő közötti határ egyáltalán nem olyan éles, mint az korábban látszott.

A „betűnkénti helyettesítés" pl. ASCII kód esetén bináris *blokkot* helyettesít blokkal. Ez a folyamatos (stream) titkosítás és a blokkos titkosítás határait mossa össze.

[22] A véges elemű nyílt ábécé az elemek számának összes permutációja szerint keverhető össze, tehát ennyi féle keverési kulcs létezhet. Ha ezeket a permutációkat megszámozzuk, akkor az i-edik

Ha a nyílt szöveget és a kriptogramot egy-egy ábécé felett értelmezzük, akkor ezeknek a transzformációknak az értelmezése triviális. Az is nyilvánvaló, hogy egymástól függetlenek, tehát akár fel is cserélhetők. A transzformációk felcserélhetősége másképp fogalmazva sorrend-függetlenséget is jelent s a titkosításnál, illetve a visszafejtésnél nagyon is jelentős lehet.

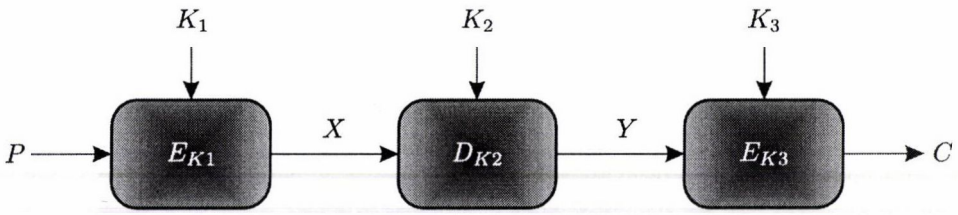
Nem nyilvánvaló a helyettesítés és a transzpozíció felcserélhetősége, ha a betűk helyett bináris blokkokban gondolkodunk.

Például minden egyes betűt egy-egy 8 bites blokk (byte) helyettesít. Ebben az esetben csak akkor sorrend-független a transzformációk, ha az egyik blokkhosszúsága egész számú többszöröse a másik transzformáció blokkhosszúságának [1].

Ez szigorúbb feltétel annál, amit korábban a linearitással kapcsolatban megadtunk, de nincs azzal ellentmondásban.

A transzformációk egyesítése

Vizsgáljuk meg a következő ábrát, amely a megfejtés megnehezítése céljából három transzformációt hajt végre egymás után rendre a K_1, K_2 és K_3 kulcsokkal.



$$C = E_{K3}\{D_{K2}[E_{K1}(P)]\}$$

7. ábra. Több egymás utáni transzformáció

Vegyük észre, hogy ha a K_1, K_2 és K_3 kulcsok azonosak, akkor ez a három transzformáció egyetlen egykulcsos transzformációval helyettesíthető. A komponens transzformációk számának növelése bonyolítja ugyan a titkosítási transzformációt, de a nyers erő módszerével szemben nem nyújt nagyobb védelmet, mint egyetlen transzformáció. Másképpen fogalmazva: ha kitalálják a kulcsot, akkor annak ismételt alkalmazása sem jelent komoly védelmet a feltöréssel szemben.

Az itt bemutatott összetett transzformáció szó szoros értelmében nem kommutatív produkt transzformáció, mert a komponensei nem felcserélhetőek. A leképezések felcserélhetősége viszont fontos kérdés, mint korábban már megmutattam.

Itt célszerű megjegyezni, hogy a „dupla DES„ titkosítást nem használják, mert találtak olyan feltörési módszert, amely a fent ábrázolt feltörési láncot mindkét végéről egyszerre támadja a nyers erő módszerével és kimutatható, hogy a krip-

permutáció egy i sorszámú leképezésként is felfogható. Nincs tehát éles elvi határ a helyettesítő és a permutációs leképezések között sem.

tográfiai erősség szempontjából lényeges, ún. ekvivalens kulcshossz nem több mint egyetlen kulcs duplája.

A tripla DES esetében viszont háromszoros ekvivalens kulcshosszal lehet számolni.

A Feistel transzformációk

A 7. ábra kapcsán láttuk, hogy a transzformációk ismételt alkalmazása csak akkor jelent nagyobb védelmet, ha mindegyik komponens-transzformációnak más-más kulcsa van. A többkulcsos rendszer használata viszont bonyolultabb. Kérdés, hogy hogyan lehet egykulcsos (szimmetrikus) titkosító rendszert úgy megszerkeszteni, hogy úgy működjön, mint egy többkulcsos, ismételt transzformációkat alkalmazó rendszer.

A megoldást *Horst Feistel*, az IBM munkatársa találta ki az 1970-es évek elején.[23]

A titkosítási rendszeréhez felhasználta a 70-es években már széles körben rendelkezésre álló számítógépeket.

Nagyon leegyszerűsítve azt találta ki, hogy a titkosítás K kulcsából egy ún. kulcs ütemező algoritmussal meghatározott számú „alkulcsot" (szubkulcsot) állít elő, és ugyanazt az összetett transzformáció sorozatot – „menet" (*round, kör*) – többször egymás után végrehajtja, de minden egyes alkalommal más-más szubkulccsal. Tehát minden egyes rundhoz más-más szubkulcsot alkalmaz.

Ezzel a megoldással a titkosítás ún. erőssége nem lett nagyobb, mint amit a K kulcs hossza (kulcstere) meghatározott, de maga a módszer annyira bonyolulttá vált, hogy csakis a nyers erő módszerével lehetett próbálkozni a feltörésénél. A DES mellesleg 64 bites nyíltszöveg blokkokhoz 64 bites kulcsot használt, de a 64 bites (8 bájtos) kulcsban „csak" 56 független bit volt, mert minden kulcsbájt egyik bitje paritásbit volt. Így a DES kulcstere 2^{56} különböző kulcsból állt.

Ennek a megoldásnak különös jelentőséget ad az a tény, hogy egyrészt 25 évig kiválóan működött, másrészt a mai szabványosított utódja[24] is lényegében hasonló

[23] Feistel 1932-ben emigrált Németországból az USA-ba. A háború alatt titkos üzenetek megfejtésével foglalkozott. Az IBM a 70-es évek elején a Lloyd biztosító társaság számára fejlesztett titkosító rendszert. Feistel a rendszerét „Dataseal"-nek akarta elnevezni de az IBM csak Demonstration Cipher-nek nevezte, amely elnevezés rövidített változataként a „Demon" elnevezést használta. Ebből lett később a „Lucifer" kriptorendszer, amely alapját képezte az 1975-ben Data Encryption Standard (DES) néven szabványosított ún. iterációs kriptorendszernek [15].

[24] Az ún. Advanced Encryption Standard (AES), amit egy hosszú pályázati és döntési folyamat után a belga Vincent Rijmen és Joan Damon Rijndael nevű iterációs kriptorendszere nyert el. Az AES 128 bites változatát 2000 októberében szabványosította a NIST.

Ha arra gondolunk, hogy de Vigenere polialfabetikus rendszere kb. 300 évig ellenállt a feltörésnek, akkor a 25 év nem tűnik soknak.

De Vigenere rendszerének a feltöréséhez azonban nem voltak meg sem azok a matematikai eszközök, amelyek a II. világháborúban már rendelkezésre álltak, sem a feltörő-gépek megszerkesztéséhez szükséges technológia nem állt még rendelkezésre.

Babbage zsenialitása kellett hozzá, mint ahogyan Rejewskyé az Enigma feltöréséhez.

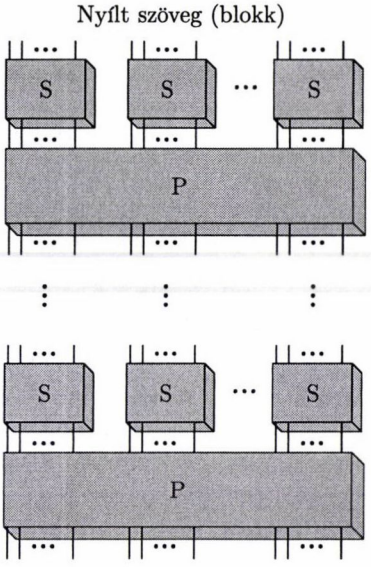
elven épül fel, csak egy-egy menetben másféle transzformációkat alkalmaz és a K kulcs hossza is 128 bit lett.

A Feistel transzformációhoz előbb még három segédfogalmat kell definiálni. Ezek a következők:

a. Egy szorzat transzformáció (*product cipher*) két vagy több komponens transzformációt kombinál olyan módon és céllal, hogy az eredő transzformáció biztonságosabb legyen, mint a komponensek bármelyike.[25] A továbbiakban ezt inkább *összetett transzformációnak* nevezzük.

b. Egy helyettesítési-permutációs hálózat (SP) olyan összetett transzformációt hajt végre, amely számos, egymást követő fokozatból áll, s e fokozatok mindegyike vagy helyettesítést, vagy permutációt hajt végre (lásd 8. ábra).

c. Egy *iterált blokk transzformáció* egy belső, ún. *rund függvény* meghatározott számú sorozatos megismétlését jelenti. Fontos paramétere a menetek (rundok) száma: r; a blokkban lévő bitek száma: n és az ún. bemeneti kulcs bitszáma: k.



8. ábra. Egy összetett transzformációt végrehajtó SP hálózat

A Feistel transzformáció az iterált blokk transzformáció egy tovább bonyolított változata. Eredeti (vagy első, a Luciferben alkalmazott) változatában a nyílt szöveg blokkjainak hosszúsága $n = 2t$, a rundok száma pedig legalább 4. Jellemzően páros számú rundot alkalmaz. A bemeneti kulcs bitjeinek száma megegyezik

[25] Figyeljük meg, hogy az elnevezés ugyan transzformációk szorzatára utal, de a definició megengedi a komponensek egymásba való beépítését, az összetett függvényeket. Ezért pontosabb is összetett transzformációról beszélni.

a blokkmérettel, tehát szintén $2t$ bit. Tartozik hozzá egy kulcs ütemező folyamat, amely minden rund számára más-más szubkulcsot állít elő a bemeneti kulcsból.

A páros számú rundnak jó oka van. A Feistel transzformáció ugyanis a nyíltszöveg blokkot megfelezi[26] és a (szub)kulcsfüggő komponens transzformációt csak egy t hosszúságú félblokkon hajtja végre egy-egy menetben. A két félblokkot aztán összekeveri a menet végén, és a következő menetben megcseréli a félblokkokat. Szigorúan véve csakis azokat a szimmetrikus transzformációkat nevezik Feistel függvényeknek, amelyekre jellemzők a fél-blokkok és ezek menetenkénti felcserélése.

Az egész eljárás jobban megérthető a 9. ábra alapján.

Az eljárásban nem tüntettük fel a szubkulcsokat ütemező algoritmust és még egy-két finom részletet sem. Egyébként igen figyelemre méltó az f Feistel függvény, amely itt a 32 bites félblokkokhoz 48 bites szubkulcsokat alkalmaz, mert – többek között – tartalmaz egy kiterjesztési és egy kompressziós transzformációt is.

Emiatt előfordulhat (és a DES-ben elő is fordul), hogy maga az f Feistel függvény nem is invertálható, de ettől még a teljes rund, illetve azok sorozata mégis csak invertálható.[27] A megfejtéskor ugyanezt a 16 menetes eljárást alkalmazzák, de a szubkulcsok sorrendje fordítottja a titkosításkor alkalmazott sorrendének. A félblokk kiterjesztése, majd a kulcsfüggő művelet végrehajtása utáni kompressziója (amelyek beleértendők az f Feistel transzformációba) ún. *lavina hatást* eredményez, ami azt jelenti, hogy a nyíltszöveg blokk egyetlen bitjének a megváltoztatása a hozzátartozó kriptogram blokkban legalább 32 bit változását okozza. A lavinahatás a rundok számának növelésével növekszik. Ez minden iterációs kriptorendszerben így van. Az AES-ben is és több más, ismert rendszerben is.

Egyébként a 128 bites AES nem bontja félblokkokra a bemeneti nyíltszöveg blokkot és 12 menetes iterációt alkalmaz.

A lavinahatás miatt nem véletlen, hogy az iterációs, szimmetrikus kriptorendszerek megfejtési kísérletei során először csak kevesebb számú runddal titkosított változatok megfejtésére szoktak törekedni. Ez még a legújabb AES feltörési kísérleteinél is így van.

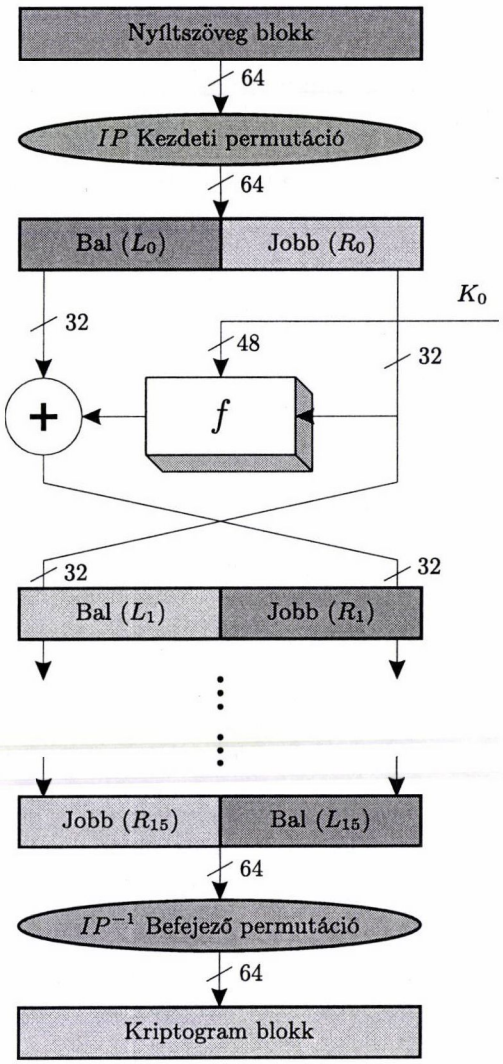
A lavinahatásról még annyit, hogy az közvetlen rokonságban van a matematikai értelemben vett kaotikus folyamatokkal.

---

[26] Van negyedelő eljárás is, pl. az IDEA esetében.

[27] Ez Menezes [24] állítása, de nem bizonyítja. A DES részletesebb elemzése azonban alátámasztja ezt az állítást. Ténykérdés ugyanis, hogy az expanzió-kompresszió transzformáció-pár alkalmazása miatt legalább is kétségek merülnek fel az f függvény invertálhatóságával kapcsolatban. Az is tény azonban, hogy a DES 25 évig jól működött. Az invertálhatóságot viszont e cikk elején a transzformációk általános követelményei során triviális alapfeltételnek tekintettünk. Mármint a teljes transzformációét, s nem annak komponenseiét. Feistel egyik óriási innovációja az, hogy rájött, hogy összetett transzformációk esetében nem feltétlenül kell minden komponensnek invertálhatónak lenni ahhoz, hogy az eredő transzformáció invertálható legyen. Az igazsághoz hozzátartozik azonban, hogy a Luciferben még nem volt ilyen expanzió-kompresszió pár, s azt az NSA javaslatára építette be a DES-be Feistel.

A nem invertálható komponensek beépítésének az a fő előnye és célja, hogy lehetetlenné tegye a transzformáció algoritmikus feltörését.



9. ábra. A Feistel transzformációt alkalmazó 16 menetes, 64 bites DES vázlata (csavart létra)

Az aszimmetrikus kriptorendszerek

Említettük, hogy kb. az 1970-es évek közepén következett be az az áttörés, amely a nagyon bonyolult és nagy számítás igényű szimmetrikus kriptorendszerek

bevezetését jelentette, méghozzá a civil alkalmazásokba.[28] Innen számítjuk az ún. negyedik és ötödik generációs kriptorendszerek korszakát.[29] A szimmetrikus kriptorendszerek elterjedésének azonban óriási gyakorlati akadálya volt, hogy a titkos kulcsot a titkosított kommunikációt megelőzően és igen nagy biztonsággal el kellett juttatni a kommunikációban részt vevő másik partnernek. A kulcsok eljuttatásának, nyilvántartásának és egyáltalában a kulcsok menedzselésének (a szimmetrikus rendszerekben) megoldatlan problémája sürgetően megkövetelte valamilyen új megoldás kialakítását. Állíthatjuk, hogy tulajdonképpen ez az igény váltotta ki az aszimmetrikus (nyíltkulcsú) kriptorendszerek feltalálását.

Kb. ugyanebben az időben támadt egy zseninek, nevezetesen Whitfield Diffienek az az ötlete, hogy valahogyan el kellene kerülni a kulcsok utaztatását.

Ennél, persze, konkrétabb ötlete is volt a megoldásra, amin aztán Martin Hellmannal és Ralph Merklevel együtt dolgoztak. 1976-ban jelentették be az ötletüket egy konferencián [35], de ők akkor még nem találtak rá működőképes implementációt.

Diffie olyan, ún. egyirányú függvényeket keresett, amelyeket az egyik irányban aránylag könnyen ki lehet számolni, de a másik irányban csak olyan sok számítási kapacitással, ami nem éri meg a dolgot, vagy nincs annyi ráfordítható idő.

Ma már széles körben ismert Ron Rivest, Adi Shamir és Leon Adleman feltalálók neveinek kezdőbetűiről elnevezett RSA titkosítás, amely alapja a természetes számok prím faktorizációja. (IFP = Integer Factorization Problem). Nagyon leegyszerűsítve ugyanis arról van szó, hogy két nagy prímszámot aránylag nem nehéz összeszorozni, de egy igen nagy szám esetében csak nagyon sok próbálkozással lehet az adott szám törzstényezőit meghatározni. Ez egy tipikus egyirányú függvényprobléma.

No de hogyan lehet összehozni egy szöveg titkosítását számokkal? Nos, blokk titkosítás esetén egy $n$bites blokk mindig tekinthető olyan egészszámnak, amely értéke 0 és $2^n - 1$ közé esik. Fontos, hogy egészszámokról van szó, s ennek a következőkben komoly jelentősége lesz. Fontos az is, hogy egy korlátos tartományba eső pozitív egészekről, vagyis természetes számokról van szó.

Térjünk vissza azonban a 3. ábra teljes titkosítási/visszafejtési folyamatához.

Láttuk, hogy triviális megoldásnak tekintették korábban, hogy az E és a D transzformációk kulcsai azonosak, maguk a transzformációk pedig egymás inverzei. Sőt: gyakorlati szempontok miatt esetleg saját maguk inverzei. Ez valamennyi szimmetrikus kriptorendszer jellemzője.

Ma már eléggé kézenfekvő az, hogy nem feltétlenül kell az E és a D transzformációk kulcsainak azonosaknak lenniük, hanem elképzelhető, hogy azok például

[28] A katonai alkalmazások körében természetesen korábban is léteztek ilyenek, de azért érdemes felfigyelni arra, hogy az amerikai hadsereg még a koreai háborúban is használt az Enigmához hasonló titkosító/megfejtő gépeket

[29] A kriptorendszer-generációkat ugyanúgy az alkalmazott technológia jellemzi, mint a számítógépek generációit, de ezzel a kérdéssel itt nem foglalkozunk. Megemlítjük azonban, hogy nem csak a transzformációs technológia, hanem a jellemző kommunikációs technológia is meghatározó [2].

egymásnak az aktuális transzformációra vonatkoztatott inverzei, s maguk a transzformációk azok, amelyek azonosak.

Az előbbiekben láttuk, hogy maguk a transzformálandó blokkok egy felülről korlátos tartományba eső természetes számok. Egyszerű esetben a kulcsok is természetes számok.

A kérdés tehát úgy tehető fel, hogy van-e a természetes számok halmaza felett értelmezett zárt művelet, amelynek az eredménye is természetes szám, s ráadásul egy felülről korlátos halmaz tagja.

Nos, a modulo n összeadás és a modulo n szorzás például ilyen műveletek. Nem véletlen, hogy az RSA rendszer is ezt alkalmazta. A modulo n direkt műveletek inverzei is léteznek.

A modulo n összeadás és kivonás inverz műveletpárt már a XVI. és XVII. század fordulóján de Vigenere alkalmazta az általa kitalált többábécés rendszerhez, amelynek alsó határesete az egyábécés Caesar-féle titkosítás is és felső határesete a nagyon sok ábécés Enigma titkosítás is.

A modulo n kongruenciák a 0-tól $(n-1)$-ig terjedő, n darab *maradékosztályba* képezik le a természetes számok megszámlálhatóan végtelen nagy halmazát. Ezek a maradékosztályok a 0 és $(n-1)$ közötti természetes számokkal jelölhetők.

A modulo n műveleteknek van egy elméleti és egy abból következő gyakorlati előnye is. Az elméleti előny az, hogy ezek a műveletek a természetes számok halmaza felett zártak. A gyakorlati előny pedig az, hogy a maradékosztályok véges száma miatt a számítások során elkerülhető a túlcsordulás.

Evariste Galois (1811–1832) megmutatta, hogy minden véges halmazon értelmezett művelet izomorf a modulo m maradékosztályokon értelmezett valamilyen modulo m művelettel. Ennek következménye az, hogy ha a kriptográfiai leképezések véges (vagy legalább megszámlálhatóan végtelen) halmazokon értelmezettek, akkor a modulo m műveletek gyakran feltűnnek a legkülönbözőbb elvű kriptográfiai leképezések körében is.

Az inverz fogalmáról itt annyit, hogy ha egy A halmaz felett értelmezett és zárt a $\otimes$ művelet és az A halmaznak eleme mind a, mind b, akkor ez a két elem egymásnak inverze a $\otimes$ műveletre nézve, ha

$$a \otimes b = e,$$

ahol e a $\otimes$ művelet egységeleme (amely, persze, szintén eleme az A halmaznak). Azt, hogy a b elem az a elem inverze, a

$$b = a^{-1},$$

jelöléssel jelöljük. Például a szorzás művelete esetén két szám (a és a^{-1}) akkor inverzei egymásnak, ha

$$a \cdot a^{-1} = 1.$$

Jelöljünk most egy n blokkhosszúságú nyílt szöveget P-vel és a mondottak értelmében értelmezzük P-t számként. Igen egyszerűen végrehajtható ekkor a következő titkosítási transzformáció:

$$C \leftarrow P^a \bmod n$$

A kapott eredmény mindenképpen 0 és $(n-1)$ közé esik, tehát alkalmasan választott hatványozási algoritmus (pl. a gyors hatványozás) esetében nem fordulhat elő túlcsordulás, akármekkora is az a hatványkitevő. Vegyük észre, hogy itt a titkosítás kulcsa maga az a kitevő.

A visszafejtés ugyanazzal a hatványozási művelettel történhet, de ekkor az a^{-1} (inverz) hatványra kell emelni a C kriptogramot:

$$C^{a^{-1}} \bmod n = (P^a)^{a^{-1}} \bmod n = P^{a \cdot a^{-1}} = P^1 = P.$$

A visszafejtés kulcsa tehát a^{-1}, vagyis a titkosító kulcs inverze, a titkosító transzformáció pedig pontosan megegyezik a visszafejtő transzformációval.

A két kulcs nem azonos, hanem egymás inverzei. Bizonyos szempontból éppen olyan triviális megoldás, mint a kulcsok azonossága volt a szimmetrikus kriptorendszerek esetében. Míg azoknál a transzformációk voltak egymás inverzei, és a kulcsok azonosak, az aszimmetrikus kriptorend-szereknél a kulcsok egymás inverzei és a transzformációk azonosak.[30] Maga az „aszimmetrikus" elnevezés arra utal, hogy a kulcsok nem azonosak. Egyébként azonban ezek a kétkulcsos rendszerek a szó szoros értelmében legalább annyira szimmetrikusak, mint a szimmetrikusnak nevezett (egykulcsos) kriptorendszerek.

A 4. ábra és az 5. ábra pontosan ezt a szimmetriát hivatott bemutatni.

Vegyük észre azt is, hogy a bemutatott rendszerben fontos szerepe van az n modulusnak is, azaz az egész titkosító/visszafejtő rendszer működtetéséhez az inverz kulcspáron kívül a modulust is ismerni kell, tehát legalább három paraméter van (Itt!).

Az aszimmetrikus kriptorendszerek azért nem annyira egyszerűek, mint azt az előbbiekben bemutattuk. Számos gyakorlati követelmény is létezik, amelyeket ki kell elégíteni.

Alapelv, hogy ha a két kulcs nem azonos, akkor az egyiket nyilvánosságra is lehet hozni. Ebből azonban következik, hogy a nyilvános kulcsból semmilyen módon ne lehessen kiszámítani a párját, vagyis a titkos kulcsot.

Vizsgáljuk meg most ezt a transzformációt egy kissé közelebbről!

Az RSA nyíltkulcsú rendszer és az egyirányú függvények[31]

Az előzőekben (amikor csak első közelítésben kívántuk bemutatni az alapelvet) ismételt hatványozással értük el, hogy visszakapjuk az eredeti szöveget.

A példaként bemutatott egyszerű hatványozás esetén, ha az egyik kitevő és a modulus ismert, akkor az inverz kitevő kiszámolható. Az RSA rendszerben ezért

[30] Mármint az itt bemutatott egyszerű esetben. Bonyolultabb rendszereknél azért nem lehet a transzformációkat teljesen azonosaknak tekinteni, mint ahogyan a negyedik generációs iteratív kriptorendszerekben sem voltak teljesen azonosak.

[31] Köszönetet kell mondanom Tóth Gergelynek, a Veszprémi Egyetem Székesfehérvárra kihelyezett AIFSz központja tanárának e fejezet megírásához nyújtott segítségéért.

nem is egyszerűen csak inverz kitevőkkel, hanem két nagy (10^{150} nagyságrendű) prímszám Euler függvényével dolgoznak.

Ha találunk olyan e és d számokat, amelyek szorzata valamilyen alkalmas közös n modulussal osztva 1-et ad, akkor titkosíthatunk az e és n számokat használva:

$$c \leftarrow p^e \bmod n.$$

A visszafejtésre a $p \leftarrow c^d \bmod n$ formulát használhatjuk, hiszen

$$c^d = (p^e)^d = p^{e \cdot d} \equiv p^1 \bmod n. \tag{0.1}$$

A probléma alapvetően az, hogy míg valós számok esetében egy 0-tól különböző számnak mindig van inverze a szorzásra nézve (történetesen a reciproka), addig ez a maradékosztályokban már korántsem mindig van így. Például egy páros számot bármilyen számmal megszorozhatunk, a szorzat soha sem fog 1 maradékot adni egy páros modulus esetében.

Pierre Fermat [1601–1665] 1660 körül rájött[32], hogy az eljárás „működik", ha az $e \cdot d$ szorzat prímszám. Fermat eredményének azonban nem sok gyakorlati alkalmazása van; tekintve, hogy csak az $e = 1$ vagy $d = 1$ esetben működik. Ez kriptográfiai szempontból azt jelentené, hogy magát a nyílt szöveget küldjük el a címzettnek.

Szerencsére *Leonard Euler* [1707–1783] általánosabb formulát talált a Fermat képletnél, amelyben a kitevőnek nem kell prímszámnak lennie, és a hatványozás után mégis visszakapjuk osztási maradékként az eredeti számot.

Az Euler féle képletben

$$p^{e \cdot d} \equiv p \bmod n,$$

$$\text{ha } e \cdot d = k \cdot \varphi(n) + 1 \text{ alakú}, \tag{0.2}$$

ahol k tetszőleges természetes szám. A $\varphi(n)$ jelölés az n számnál kisebb, n-hez relatív prímek darabszámát jelöli, azaz hogy hány olyan szám van 0 és n között, amelyeknek 1-en kívül nincs olyan osztójuk, amely n-et is osztaná.[33] A (0.2) feltétel ekvivalens azzal a feltétellel, hogy e és d egymás inverzei a $\varphi(n)$ modulusra. Ha tehát a kódolást és a dekódolást az üzenetblokk hatványozásával oldjuk meg, amint azt a (0.1) formulában tettük, akkor ez működni fog abban az értelemben, hogy az ismételt hatványozással visszakapjuk az eredeti üzenetet. A gond csak $\varphi(n)$ kiszámításánál van.

Prímszámokra elég kézenfekvő, hogy $\varphi(p) = p - 1$, és azt is viszonylag könnyű belátni, hogy szorzat φ-je a φ-k szorzata, azaz

$$\varphi(p \cdot q) = \varphi(p) \cdot \varphi(p). \tag{0.3}$$

[32] Kis (más néven: „karácsonyi") Fermat tétel: $ap - 1 \equiv 1 \bmod p$, ha p prímszám.

[33] Fermat-val ellentétben Euler bebizonyította a saját tételét, és ezzel visszamenőleg a kis-Fermat tétel is bizonyítást nyert. Egy prímszámot ugyanis 1-en kívül egyetlen nála kisebb természetes szám sem oszt, azaz hozzá képest az összes nála kisebb szám relatív prímszám. Ezek szerint prímszámokra $\varphi(p) = p - 1$, amiből következik a kis-Fermat tétel állítása.

Vagyis egy tetszőleges n modulus φ-je könnyen meghatározható, ha ismerjük n prímtényezős felbontását.

Rivest, Shamir és Adleman frappáns módon oldották meg, hogy a kulcsokat létrehozó „illetékes" ismerje a modulus prímfelbontását, az illetéktelen támadó azonban ne kerülhessen birtokába ennek az információnak: egyszerűen választottak két prímszámot, és ezek szorzatát használták a kódoláskor modulusként.

A modulus prímtényezőinek segítségével meghatározható az n modulus Euler-féle φ-je, ebből pedig az e és d kitevők, amelyek egymás inverzei a $\varphi(n)$ számra nézve.[34]

Ha egy illetéktelen akarja megfejteni a titkos d számot a közzétett e és n számok alapján, mindenképpen meg kell határoznia $\varphi(n)$-et, hogy megtudja, egyáltalán milyen modulusra kell kiszámítania az e szám inverzét.

Jelenlegi ismereteink szerint viszont az Euler féle $\varphi(n)$-et csakis n prímtényezőinek ismeretében lehet meghatározni. Ezek megkeresése a jelenleg használt kulcsméretek és számítástechnika esetén sokszorosa a Világegyetem várható életkorának.

Az RSA kódolás erősségét tehát egy szemtelenül egyszerű ún. egyirányú függvény adja. A titkosítás kulcsának létrehozója maga választja meg a törzstényezőket, majd azok *összeszorzása* révén számítja ki a kódoláskor/dekódoláskor használt n modulust. A törzstényezők ismeretében a modulus kiszámítása nevetségesen egyszerű: egyetlen szorzási művelet. A támadó azonban a fordított műveletet kénytelen elvégezni, azaz egyedül az n modulus alapján meghatározni azokat a prímszámokat, amelyek szorzataként a modulus létrejött. A művelet, amely az egyik irányban akár tollal és papíron is viszonylag gyorsan elvégezhető, visszafelé még a világ minden informatikai kapacitásának birtokában is elképzelhetetlenül hosszú időkig tartó feladatot ró a rendszer feltörőjére.

[34] A módszer elvéből következik, hogy az n modulusnak és e-nek relatív prímeknek kell lenniük. Igen nagy számokról lévén szó, ezt nem is olyan egyszerű megvizsgálni. Ehelyett a gyakorlatban azt teszik, hogy e-nek egy nagy prímszámot választanak, amely ha sem p-vel, sem q-val nem egyezik, akkor biztosan relatív prím n-hez képest is.

Erre a célra Mersenne vagy Fermat prímeket szokás választani, mert a bináris alakjaik nagyon könnyen előállíthatók. A gyors hatványozás alkalmazhatósága miatt a Fermat prímek tulajdonképpen alkalmasabbak lennének, de mindössze 6 ilyen prímszám van, amelyek közül a legnagyobbat valóban gyakorta használják is nyílt kulcsként.

A Mersenne prímekből sokkal több van, de ezekkel az eljárás számításigénye lényegesen nagyobb.

(Az oka a gyors hatványozásban rejlik.)

A Fermat számok $2^{2^k} - 1$ alakú számok, de csak $k < 6$ esetén prímek. A gyakorlatban éppen a $k = 5$-öt szokták alkalmazni. Az ilyen bináris számok Hamming súlya csak 2, és ezért a gyors hatványozás valóban gyorsan végrehajtható.

A Mersenne számok $2^k - 1$ alakúak és nem mindegyik ilyen szám prím. A k természetes számra viszont nincs semmilyen kikötés.

További, titkosításra alkalmazható csoportműveletek

A diszkrét logaritmus probléma (DLP)

Véges elemszámú G csoportban a g generáló elem és egy szabadon választott a csoport-elem ismeretében határozzuk meg azt az x kitevőt, amelyre $g^x = a$ ($x = \log_a g$, de x-nek egészszámnak kell lennie) [22].

A Diffie–Hellman (DH) probléma

Véges elemszámú G csoportban a g generáló elem, valamint g^a és g^b ismeretében határozzuk meg g^{ab}-t.

A következő csoportok használata terjedt el:

- A p elemből álló mod p maradékosztályok multiplikatív csoportjaként értelmezhető csoport, amely valódi (és véges) részhalmaza a természetes számok halmazának.
- A $GF(2^m)$ test $2^m - 1$ elemű multiplikatív csoportja.
- Az elliptikus görbék pontjain értelmezett csoport.

Ezek részletesebb diszkussziója nélkül csak azt szeretnénk kiemelni, hogy mindegyik ilyen aszimmetrikus kriptorendszer az inverz kulcsok elvét alkalmazza ugyanazon csoportművelet mellett.

Mindegyikre léteznek megoldási algoritmusok, amelyek elsősorban a számításigény és a futási idők tekintetében versenyeznek egymással. Mégsem jelenthető ki azonban, hogy egyik algoritmus jobb, mint egy másik, mert a hatékonyságuk a konkrét alkalmazástól is függ.

Az RSA algoritmus pl. máig is a legelterjedtebb, és nem is ok nélkül, de csak adott (s nem is túl nagy) blokkhosszúságú nyíltszövegek titkosítására alkalmas.

Nem véletlen az sem, hogy pl. a digitális aláírások esetében a Diffie–Hellman algoritmust szabványosították, és az sem véletlen, hogy nagyon hosszú üzenetek egy menetben való nyíltkulcsú titkosítására pedig ma az ún. NTRU (Number Theory Research Group at MIT) rendszert tartják a legalkalmasabbnak.

Mindezek bemutatása azonban messze meghaladja e cikk terjedelmi korlátait és célkitűzését is.

Összefoglalás

A kriptográfia történetének és fejlődésének nagy mérföldkövei jó rendszerezési alapot nyújtanak e cikkben bevezetett kriptogenerációk fogalmához. Áttekintve a kriptorendszerek fejlődését azt találtam, hogy néhány alapvető fogalom nem, vagy nem egyértelműen definiált. Ilyen volt pl. az ún. blokkos titkosítások blokk fogalma, a több helyütt említett lineáris transzformációk fogalma és maguk a kriptográfiai transzformációk is.

E cikkben leírt áttekintés alapvető motívuma éppen e fogalmak „rendbetétele" volt. Legalább is megkíséreltem néhány általam felfedezni vélt ilyen hiányt pótolni

és ebben az írásban összefoglalni mindazt, amire jutottam. Szép számmal maradtak nyitott kérdések is.

Írásommal buzdítani is szeretnék fiatal kutatókat arra, hogy *érdemes* egy tudományterület alapkérdéseivel foglakozni.

A cikk a kriptorendszerek generációinak fogalmi és definitív bevezetése után a kriptográfiai transzformációk (matematikai értelemben leképezések) általános jellemzőivel, tulajdonságaival foglalkozik.

Megmutatja, hogy matematikai értelemben miért helyesebb leképezésekről beszélni, mint transzformációkról és vizsgálja e leképzések értelmezési tartományait, valamint az értékkészletét.

Definiálja a leképzések őstartományán értelmezhető műveletet, nevezetesen a konkatenációt, és ennek az értelmezési tartománya segítségével definíciót ad a kriptográfiai „blokk" fogalmára.

Heurisztikus módon megfogalmazza e leképezések általános követelményeit. Megfogalmaz egy sejtést, amely szükséges és elegendő feltétel a klasszikus helyettesítési és permutációs transzformációk felcserélhetőségére, valamint összevonására. (A blokkhosszúságok egészszámú arányára vonatkozó kijelentésről van szó.)

Foglalkozik az egyszerű leképző függvények inverz tulajdonságaival. Claude E. Shannon javasolta elsőként az ún. produkt transzformációkat, amelyek valójában vagy nem kommutatív produktumok, vagy összetett függvények. Minden esetre komoly szerepük van a modern, ún. negyedik generációs, *szimmetrikus* kriptorendszerekben. A cikk kitér az ún. Feistel transzformációkra és alkalmazásaikra, a többkomponensű transzformációk komponensei invertálhatóságának a kérdésére, s végül az ún. aszimmetrikus (nyíltkulcsú) kriptorendszerekben alkalmazott transzformációk kapcsán azt igyekszik bemutatni, hogy azok a tradicionális kriptorendszereknek mintegy a tükörképei.

Megfogalmaz egy állítást is arra, hogy az aszimmetrikus kriptorendszerekben alkalmazott művelet (amelyre az inverz kulcsok vonatkoznak) véges halmazok esetén szükségképpen izomorf a modulo m maradékosztályok feletti valamilyen művelettel.

Ezért aztán egyáltalán nem véletlen, hogy a modulo m aritmetika a legkülönbözőbb alapelvű kriptorendszerek esetében minduntalan visszaköszön.

(A szerző számára talán legfrappánsabb módon a Knapsack-féle nyíltkulcsú rendszerben, amely egy távolság különböző távolságokkal való lefedéséből indul ki, vagyis egy geometriai problémából. A fedő távolságoknak szupernövekvő sorozatot kell alkotniuk, s akkor a lefedés kimutathatóan egyértelmű és unikális. A Knapsack rendszert ugyan nem szabványosították, de nagyszerűen bemutatható a segítségével a modulo n műveletek előfordulása. [4]-ben megtalálható a részletes leírása is kidolgozott példával együtt.)

Szakirodalmi Hivatkozások

[1] Tóth Mihály, *A kriptográfia alap-transzformációiról,* Természet-, Műszaki- és Gazdaságtudományok alkalmazásának 2. nemzetközi konferenciája. 2003. május 10. Szombathely.

[2] Tóth, Mihály, The Four Generation of Criptosystems, in: *The Second Word Meeting of Information Scientists held in Budapest,* 5-8 June, 2000.

[3] Tóth Mihály, *Bevezetés a kriptográfiába I. A kriptorendszerek első generációja,* Kandós jegyzet, 2002. szept., PDF formátum, 575 KB, http://www.szgti.bmf.hu/~mtoth/download/Kriptografia/.

[4] Tóth Mihály – Prof. Randall K. Nichols, *Aszimmetrikus kriptorendszerek,* Főiskolai jegyzet BMF 163/2000.

[5] Tóth Mihály, *A DES (Data Encryption Standard),* Főiskolai jegyzet; BMF 27/2000.

[6] Tóth Mihály – Tóth Gergely: *Az elektronikus információ titkosítása és hitelesítése,* Cikk a GTE Gépgyártástechnológia c. folyóirata XXXIX. évf. 10. számában, p. 241–251.

[7] Tóth Mihály, *Szimmetrikus és aszimmetrikus kriptorendszerek,* Alkalmazott matematika, statisztika és informatika konferencia A Kodolányi János Főiskolán 2000 szeptemberében.

[8] Tóth Mihály, *Nyiltkulcsú kriptorendszerek és alkalmazásuk digitális aláírásra,* Alkalmazott matematika, statisztika és informatika konferencia A Kodolányi János Főiskolán 2000 szeptemberében.

[9] Tóth Mihály, *Karakter statisztikák,* Prezentáció a „60 év a műszaki felsőoktatásért" c. Kandó konferencián, 2002 november, ISBN: 963 7158 03 0.

[10] Tóth Mihály, *Adatrejtés és szövegstatisztikák,* Cikk és előadás az Informatika a felsőoktatásban 2002 konferencián Debrecenben, ISBN 963 472 691 7.

[11] Tóth Mihály, *A kriptográfiai transzformációkról,* Cikk és előadás a Természet-, műszaki- és gazdaságtudományok alkalmazása c. 2. nemzetközi konferencián, Szombathelyen, 2003 május.

[12] *Iterative cryptosystems,* Modern Symmetric Cryptography by Mihaly Tóth (Departemental Scientific Seminar) H.T.I. 10th of Ramadan City, Egypt, Dec. 2003.

[13] *Modern Cryptosystems,* Symmetric and Public key Cryptography Presentation by Dr. Mihaly Toth, http://www.infosec-technologies.com/.

[14] Végül megemlítem itt a saját honlapomat, amelyen számos munkámat közzétettem, hogy a hallgatóim – és mások – számára hozzáférhető legyen: http://www.szgti.bmf.hu/~mtoth/.

[15] *The Codebrakers by David Kahn Scribner NY,* First published in 1967. The cited one is published in 1996. ISBN: 0-684-83130-9

[16] *Codes, Ciphers & Other Cryptic & Clandestine,* Communication by Fred B. Wrixon Black Dog & Leventhal Publishers, NY 1998, ISBN: 1-57912-040-7

[17] *ICSA (International Computer Security Association),* Guide to Cryptography by Randall K. Nichols. Published by McGraw-Hill NY... 1999. ISBN: 0-07-047166-5 *Megj: Szerintem alapműnek tekinthető, bár ma már ráférne némi frissítés, de ugyancsak a McGraw Hill kiadásában már készülőben van egy nagy, több kötetes kriptográfiai enciklopédia, amelynek egy fejezetét – megtisztelő felkérésre – lektoráltam is.*

[18] Feistel összefoglalja az előzményeket és bemutatja az iterációs rendszerét: http://williamstallings.com/Extras/Security-Notes/lectures/blockA.html.

[19] Enigma: http://www.codesandciphers.org.uk/enigma/.

[20] The Zimmerman telegram: http://www.cypher.com.au/crypto_history.htm.

[21] De Vigenere (1): http://www.bibmath.net/crypto/poly/vigenere.php3.

[22] De Vigenere (2): http://www.antilles.k12.vi.us/math/cryptotut/vigenere.htm.

[23] *A de Vigenere rendszer feltörése* (Babbage és Kasisky) http://www.infowrangler.com/phpwiki/wiki.phtml?title=Vigen%E8re_Cipher.

[24] Handbook of Applied Cryptography by A. J. Menezes at al., CRC Press, 1997. ISBN: 0-8493-8523-7.

[25] A PGP egy gyűjtő webkikötője: http://www.pgpi.org/.

[26] Grzegorz Rozenberg (editor), Arto Salomaa (editor), *Handbook of Formal Languages: Volume 1,* Springer, 1997, ISBN: 3540604200.

[27] G. Paun (editor), Gheorghe Paun (editor), Grzegorz Rozenberg (editor), Arto Salomaa (editor), *Current Trends in Theoretical Computer Science Algorithms and Complexity,* World Scientific Publishing Company (April 1, 2004), ISBN: 9812389652

[28] Salomaa, Arto, Public-Key Cryptography (Texts in Theoretical Computer Science. An EATCS Series, 1996).

[29] Egy rövid összefoglalás a Chomsky féle nyelvcsaládokról: http://www.absoluteastronomy.com/encyclopedia/C/Ch/Chomsky_hierarchy.htm.

[30] Ron Rivest egyik korai összefoglaló munkája, amelyben a szimmetrikus rendszerekről és a protokollokról is ír: http://theory.lcs.mit.edu/~rivest/Rivest-Cryptography.pdf.

[31] Pierre Blanc: *Vigenere cipher,* http://it.geocities.com/teutoburgo/java/javi/algorithm.html.

[32] Kupás Péter, *Elliptikus görbék az oktatásban és rejtjelző rendszerekben,* Előadás az Informatika a felsőoktatásban 2002. c. debreceni konferencián.

[33] Endrődi Csilla, *Az RSA és az ECC gyakorlati összehasonlítása,* BMGE MIT Ph.D. dolgozat.

[34] *Practical Cryptography by Niels Ferguson & Bruce Schneier,* Wiley Publishing Inc. 2003, ISBN: 0-471-22357-3

[35] Merkle, Ralph, *The New Directions in Cryptography,* National Computer Conference, Berkeley, 1976 június.

[36] Virasztó Tamás, *Titkosítás és adatrejtés,* NetAcademia Kft. 2004. ISBN: 963 214 253 5. *Ez a végül, de egyáltalán nem utolsó sorban megemlített könyv ma a legjobb (ha nem az egyetlen említésre méltó) mű az utóbbi 5 év e témakörbe tartozó magyar könyvei közül.*

TÓTH MIHÁLY
BUDAPESTI MŰSZAKI FŐISKOLA
KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
e-mail: toth.mihaly@szgti.bmf.hu

BASIC MATHEMATICAL TRANSFORMATIONS IN CRYPTOGRAPHY

MIHÁLY TÓTH

This paper introduces the concepts of Crypto-generations then deals with the mapping and transformation functions of Cryptosystems. It pays particular interest on the concept of blocks and alternating block cipher transformations, the concept of linearity and the changeability of substitution and permutation type transformations. It defines the general conditions of those transformation. The paper discusses the invertibility of crypto-transformations particularly Feistel functions.

The paper demonstrates the symmetry of iterative and public key transformations. Its main goal is to make clear some basic concepts of four and fifth generation cryptosystems.