Gonda János

A REJTJELEZÉS NÉHÁNY KÉRDÉSE

Budapest, 2010

Lektorálta: Kovács Attila

Utolsó módosítás: 2011. május 18.

TARTALOMJEGYZÉK

Vissza a tartalomhoz

Bevezetés

A titkosság a társadalmak, egymástól elkülönült közösségek kialakulásához kapcsolódik. Egyrészt a rendelkezésre álló erőforrások különbözősége, másrészt az emberi léthez kapcsolódó bizonyos negatív tulajdonságok arra vezettek, hogy az egyes csoportok egymás rovására jutottak meghatározott javakhoz. A javak megszerzésében azok számíthattak nagyobb sikerre, akik képesek voltak meglepni a konkurens társaságot. A meglepetés alapja viszont az, hogy az egyik társaság tud valami olyat, amelyet a másik csoport nem ismer, és amit az egymással való vetélkedés során eredményesen fel lehet használni.

A titkosítás története több könyvben is megtalálható. Közülük minden bizonnyal a leghíresebb *David Kahn* könyve, ugyanis szinte nincs olyan, a kriptológiával foglalkozó könyv, amely ne hivatkozna erre a több mint ezer oldalas könyvre. A történeti szemléletű magyar nyelvű könyvek közül említésre méltó *Simon Singh* műve, a *Kódkönyv*, amely már a legújabb rejtjelező eljárásokról is számot tud adni, hiszen ebben az évezredben jelent meg. Igen tanulságos elolvasni *Révay Zoltán Titkosírások* című könyvét. Ez a könyv egyrészt azért érdemel figyelmet, mert igen sok jeles magyar személyiségről derül ki, hogy intenzíven alkalmazta a titkosítás tudományát, és számos érdekes megoldást találtak ki a rejtjelezéshez, másrészt viszont a megjelenésének dátuma szempontjából is érdekes ez a könyv (bár ez elmondható *Kahn* könyvéről is). *Révay Zoltán* idézi *Aineiasz Taktikosz Taktika* című művének egyik könyvét, a *Poliorkétika*-t, közelebbről ennek XXXI. fejezetét, amelyben *Aineiasz* a titkos levelekről ír. Ez a fejezet azzal kezdődik, hogy „A titkos leveleknek mindenféle küldési módjuk van, de a küldőnek és a címzettnek egymás között előzőleg meg kell állapodnia.”. Ez az idézet azért érdekes, mert a könyv első megjelenése előtt két évvel jelent meg *Diffie* és *Hellman* cikke, amely alapjaiban rázkódtatta meg a rejtjelezés világát, és amely alaposan rácáfolt *Aineiasz*-ra, és közvetve *Révay*ra is, aki a fenti gondolatot lényegében véve a titkosítás alapjának tekintette. (Ez nem csökkenti a *Révay*-könyv értékét, csupán arra mutat rá, hogy a világ forgandó, és igen rövid idő alatt fenekestül tud egy-egy tudományág megváltozni. Hasonló változás történt például 1900-ban vagy 1905-ben a fizikában.)

Minden titkosító eljárás esetén lényeges, hogy az alkalmazott algoritmusról feltételezzük, azt mindenki ismeri, és a titkosságot az úgynevezett **kulcs** biztosítja. A kulcs az algoritmus egy olyan paramétere, amelytől függően ugyanaz az eljárás ugyanazon titkosítandó üzenetből a kulcs függvényében más és más rejtjelezett szöveget állít elő. A klasszikus rejtjelező eljárásoknál a visszafejtéshez szükséges kulcs vagy megegyezett a titkosításhoz használt kulccsal, vagy abból könnyen ki lehetett számolni, így szükséges volt a rejtjelezéshez használt kulcsot is titokban tartani, továbbá az üzenetváltásban résztvevő két fél között biztonságosan kicserélni. Más a helyzet akkor, ha az oda-, illetve visszatranszformáláshoz használt kulcsok olyanok, hogy az egyik ismeretében a másik csak olyan nagy költséggel határozható meg, hogy az meghaladja a megszerzett információ értékét. Ebben az esetben a titkosító kulcs akár nyilvános is lehet, mégsem képes senki illetéktelen elolvasni a rejtjelezett szöveget, mivel nem rendelkezik a visszafejtéshez szükséges kulccsal. Ez az a gondolat, amely *Diffie* és *Hellman* cikkében jelent meg, és amely alapján kialakult a nyilvános kulcsú rejtjelezés. E nélkül a mai világ egészen más lenne. A régi időkben lényegében véve csak az államnak voltak féltve őrzött titkai (persze a szépasszonyok sem akartak mindent az uruk orrára kötni, de ezek kevésbé lényeges titkok...), így elegendő volt csupán néhány tucat kulcsot előállítani és kicserélni. (Ez utóbbi egy kényes pontja a rejtjelezésnek, hiszen a kulcsot biztonságosan és titkosan kell eljuttatni a másik félnek, amikor persze felmerül a kérdés, hogy miért nem magát az üzenetet cserélik ez alkalommal ki. Erre azonban könnyű a válasz: a kulcsot bármely időben cserélhetjük, és a cserére ritkán van szükség, továbbá a kulcs általában rövid az üzenethez képest.) A mai világban viszont a titkosítandó információk túlnyomó többsége gazdasági jellegű, és magánszemélyekhez, vállalatokhoz kapcsolódik. Potenciálisan minden ember és minden vállalkozás rendelkezik titkolandó adattal, amelyet a legkülönbözőbb intézményekkel kell kicserélnie. A titkos kulcspár alkalmazása esetén különböző partnerhez más és más kulcsra lenne szükség, ami azt jelentené, hogy hihetetlenül sok kulcsot kellene igen gyakran rendkívül sok pár között kicserélni, és a titkos kulcsokat megfelelően adminisztrálva biztonságosan tárolni, ami megoldhatatlan feladat elé állítaná az egyszerű honpolgárokat. Még azt is figyelembe kell venni, hogy a kulcsot viszonylag gyakran kell cserélni, még az előtt, hogy illetéktelen személy megfejtené, és így a továbbiakban a titkos információnkat olvasni tudná.

Az előbbi gondolatok alapján joggal merül fel a kérdés, hogy kell-e egyáltalán foglalkozni a klasszikus, szimmetrikus rejtjelező rendszerekkel. A válasz meglepő módon igen. A helyzet ugyanis az, hogy a szimmetrikus rendszerek lényegesen gyorsabbak, mint a nyilvános kulcsú eljárások, ezért a legtöbb esetben egy-egy konkrét üzenetváltás előtt a nyilvános kulcsú rejtjel segítségével a két partner kicserél egy kulcsot, és a továbbiakban az aktuális információt az így megismert kulcs segítségével, egy klasszikus módszerrel küldik egy nyilvános csatornán keresztül.

Vissza a tartalomhoz

1. A rejtjelezés alapjai

Mint a Bevezetésben említettük, az információ érték, amelyet ezért óvni kell. Az információbiztonság megteremtése három, egymást kiegészítő tevékenységgel érhető el:

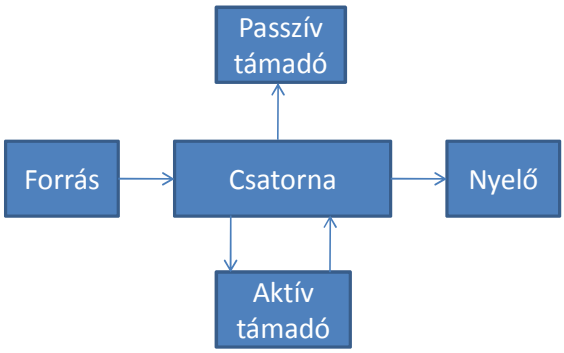
1. fizikai módszerek;
2. ügyvitel-technikai előírások;
3. algoritmikus eljárások.

A fizikai módszerek arra szolgálnak, hogy illetéktelenek fizikailag ne férhessenek hozzá olyan adatokhoz, amelyek számunkra védendő információt hordoznak. Egy egyszerű példa, amikor páncélszekrénybe zárunk bizonyos dokumentumokat (és a páncélszekrény kulcsát nem tesszük a lábtörlő alá!). Az ügyvitel-technikai előírások megszabják például az iratkezelést, ki mikor, milyen feltételekkel juthat hozzá egy adott irathoz, milyen szinten és mennyi időre szükséges az adott dokumentum tartalmát titkosítani, stb. Komoly nemzetközi és hazai szabványok írják le az informatikai biztonság fogalmát, valamint az ezt biztosító fizikai és ügyvitel-technikai teendőket. Az informatikai biztonság igen nagy mértékben, döntően függ az előbbi két terület, tehát a fizikai és az ügyvitel-technikai tevékenységek megfelelő kiépítésétől, azonban ezekkel a kérdésekkel ebben a jegyzetben a továbbiakban nem foglalkozunk, a tárgyunk csupán a harmadikként említett kérdéskör, az algoritmikus biztonság.

Az információt hordozó adatokat egy **csatorná**n keresztül juttatjuk el a **forrás**tól a **címzett**hez. Maga a csatorna lehet olyan, amelynek még a létéről sem tud az illetékeseken kívül más, vagy legalábbis az alkalmazóik azt szeretnék és remélik, hogy csak ők tudnak erről a csatornáról. Ez a felállás nyilván nagyon ritka, és a megvalósítása igen körülményes. Egy ennél könnyebben megvalósítható, és gyakrabban előforduló megoldás, amikor a csatorna használata, az ahhoz való hozzáférés csak meghatározott körben lehetséges. Ilyen például a kormányzati kommunikációs rendszer, a hadsereg, a rendőrség, a MÁV hírközlő rendszere, és más, hasonló szervezetek saját adatátviteli rendszere. Nyilván ez a rendszer, ez a felállás sem alkalmas arra, hogy nagy tömegek tudjanak meghatározott körben adatokat továbbítani úgy, hogy az adatok által hordozott információ csak meghatározott személyek számára legyen hozzáférhető. Ez csak oly módon érhető el, ha az adatokat fizikailag lényegében véve bárki által elérhető, nyilvános csatornán forgalmazzuk, ám valamilyen módon az illetéktelenek számára elérhetetlenné tesszük az információt. Ezt a feladatot oldják meg, több-kevesebb sikerrel, az algoritmikus eljárások.

A számunkra fontos, védendő információ más számára is értékes lehet, így azt próbálja valamilyen módon megszerezni. Az ilyen résztvevő a **támadó**. A támadókat két csoportba soroljuk:

- **passzív támadó**;
- **aktív támadó**.



1. ábra

Az első típusba sorolt támadó csupán megszerezni akarja az információnkat, míg az aktív támadó az információt hordozó adatokat manipulálja is azáltal, hogy üzenetet, adatokat

- **kivon**;
- **beszúr**;
- **módosít**.

A passzív támadással szemben a **titkosítás**sal, a **rejtjelezés**sel védekezhetünk, és majd látjuk, hogy ez adott módszerrel tökéletesen elérhető, vagyis tudunk úgy titkosítani, az információt el tudjuk úgy rejteni, hogy a lehallgatott adatokból semmivel nem tudunk annál többet meg, mint amennyi információval a lehallgatás nélkül is rendelkeztünk. Más a helyzet az aktív támadást illetően: ezt megakadályozni a nyilvános csatornán történő adatátvitel esetén teljes mértékig nem tudjuk, így csupán arra törekedhetünk, hogy az illetéktelen manipulációt felismerjük. Ezzel kapcsolatban három eljárással foglalkozunk:

- az üzenetek **integritás**a;
- **személyazonosítás**;
- **hitelesítés**.

Az üzenet integritása annak változatlanságát jelenti, vagyis azt, hogy a létrehozása óta nem változott. A személyazonosítás segítségével azt igyekszünk elérni, hogy a kommunikációs eszközhöz illetve az adatokhoz csak arra illetékes személy férjen hozzá. Végül a hitelesítéssel azt biztosítjuk, hogy valóban az az információ kibocsátója, aki azt feladóként jegyzi, és ez az az adat, amelyet ő létrehozott. Ebből következik, hogy a hitelesítés az integritást is biztosítja, hiszen ha igaz, hogy a vizsgált adaton az aláírás hiteles, akkor feltehetően az aláírt dokumentum tartalma is változatlan, mert majd látjuk, hogy a digitális aláírás tartalomfüggő.

A titkosítás (**sifrírozás**) során az eredeti szöveget egy **algoritmus** segítségével átalakítjuk (**sifrírozzuk**), majd a címzett egy másik algoritmus segítségével a kapott szöveget visszaalakítja (**desifríroz**). Az előbbi algoritmust általában E-vel, míg az utóbbit D-vel jelöljük (E az ***E**nciphering* illetve az ***E**ncrypting – titkosítás* szóból, míg a párja, D, a ***D**eciphering* illetve a ***D**ecrypting – visszafejtés, desifrírozás* – szóból származik). Az eredeti szövegeket **nyílt szöveg**nek mondjuk és általában m-mel (az angol ***m**essage – üzenet* szóból) vagy p-vel (***p**laintext – nyílt szöveg*) jelöljük, és ezek halmaza, az **üzenettér**, $\mathcal{M}$, míg a titkosítás eredményeként kapott szöveg a **rejtjelezett szöveg**, röviden a **rejtjel**, vagy **kriptogramm,** amit általában c-vel jelölünk az angol ***c**ipher* szó alapján, és a rejtjelek halmaza, a **rejtjeltér**, $\mathcal{C}$. Ennek megfelelően a titkosítás és a fejtés egy

$$m \in \mathcal{M}: m \mapsto c = E(m),$$

$$c \in \mathcal{C}: c \mapsto m = D(c)$$

pár. Nyilván feltétel, hogy E injektív legyen, mert ellenkező esetben a rejtjelből nem lehetne egyértelműen visszanyerni az eredeti szöveget.

Ebben a formában a rendszer nem alkalmas hosszú távú, gyakori titkosított üzenetváltásra. Elegendően sok különböző titkosított üzenet megfigyelése esetén szinte minden titkosító algoritmus kiismerhető, és így a támadó által is olvashatóak az üzenetek. Ezt elkerülendő, az algoritmust időnként változtatni kell, még olyan időben, amikor a támadónak nem sikerült a rendszer **feltörés**e. Akkor mondjuk, hogy valaki feltörte a rendszert, ha bármely, az adott algoritmussal titkosított szöveget lényegében véve azonnal képes fejteni. Egy ilyen rendszer használata rosszabb, mint ha nyílt szövegként továbbítanánk az üzeneteinket, hiszen az utóbbi esetben tudjuk, hogy illetéktelenek is olvashatják, míg a másik esetben biztonságban érezzük magunkat, holott nem vagyunk biztonságban. Sajnos az algoritmus gyakori változtatása gyakorlatilag lehetetlen, ugyanis egy ilyen szabály csak akkor alkalmas a titkosításra, ha garantáltan biztonságos, vagyis biztosak lehetünk benne, hogy gyakorlatilag egy idegen

nem képes az alatt az idő alatt, amíg használjuk (és amíg a vele titkosított üzenetek információval bírnak), feltörni. A lehetséges algoritmusok, a valóban biztonságos algoritmusok száma nem túlságosan nagy, másrészt csak akkor lehetünk biztosak benne, hogy az algoritmus szinte biztosan olyan jó, amilyennek véljük, ha azt hozzá értő emberek alaposan megvizsgálták. Ez viszont akkor lehetséges, ha az algoritmus nyilvános, és így valóban a legjobb szakemberek tanulmányozzák és keresik a gyenge pontjait. Ha viszont az algoritmus nyilvános, akkor önmagában nem alkalmas a titkosításra. A megoldás kulcsa a **kulcs**. Ez az algoritmus egy paramétere, amelynek a megváltoztatásával egy adott nyílt szövegnek más lesz a titkosított párja. A kulcs egy könnyen változtatható, könnyen megjegyezhető, könnyen kicserélhető adat. Ezzel kapcsolatban fogalmazta meg Kerckhoffs (*Jean-Guillaume-Hubert-Victor-François-Alexandre-Auguste Kerckhoffs von Nieuwenhof*, vagy röviden *Dr. Auguste Kerckhoffs* holland nyelvész és kriptográfus) a titkosítással kapcsolatos elvárásokat (DESIDERATA DE LA CRYPTOGRAPHIE MILITAIRE) a *La Cryptographie Militaire* című munkájában, amely 1883-ban jelent meg a *Le Journal des Sciences Militaires* című folyóiratban:

1. ha egy rendszer elméletileg nem feltörhetetlen, akkor gyakorlatilag legyen az;
2. ne igényeljen titkosságot, és ne okozzon gondot, ha az ellenséghez kerül;
3. a kulcs legyen könnyen megjegyezhető, ne kelljen lejegyezni, és szükség szerint legyen könnyen kicserélhető, megváltoztatható;
4. táviratként lehessen továbbítani;
5. legyen hordozható, és egyetlen ember is tudja működtetni;
6. legyen könnyen kezelhető, ne kelljen hozzá rengeteg szabályt figyelembe venni.

A fenti hat pontból a legfontosabb a második, amelynek a lényege, hogy a rendszer biztonsága kizárólag a kulcs biztonságától függ. Ez a **Kerckhoffs-elv**. Az elv szerint az algoritmus nyilvános lehet, csupán a kulcsot kell titokban tartani. De ha az algoritmus nyilvános, akkor akár szabványosítani is lehet, aminek az a nagy előnye, hogy előtte valóban a legjobb szakemberek vizsgálják meg, hogy ténylegesen megbízható-e az adott leképezés.

Az elvárások egy része ma már túlhaladott, de, a mai körülményeket figyelembe véve, hasonló kívánságok fogalmazhatóak meg (például nyilván nem táviratilag továbbítjuk a rejtjeleket, de a lényege ma is az, hogy könnyen hozzáférhető, nyilvános csatornán küldjük az információkat).

Szó volt a rendszer biztonságáról. Ennek – ha biztonságos – két osztálya van:

1. **elméletileg biztonságos** illetve **feltétlenül biztonságos** a rendszer, ha a támadó bármekkora erőforrás birtokában sem képes feltörni;
2. **gyakorlatilag biztonságos**, ha elméletileg törhető, de a feltöréshez szükséges idő illetve tárkapacitás (vagy valamilyen más, szükséges erőforrás) nem áll rendelkezésre.

A 2. ponttal kapcsolatban érdemes megjegyezni, hogy a titkosításnak költsége van, és csak annyit érdemes áldozni rá, amennyit maga a védendő információ ér. Ez azt is magában foglalja, hogy olyan rendszert kell alkalmazni, amelynél az alatt az idő alatt, amíg információs értéke van a titkosított adatnak, az a támadó által feltehetően nem fejthető, de ez ismét az alkalmazott rendszer költségével függ össze.

A kulcs használatával a rendszer leírása módosul. Mind a titkosításhoz, mind a fejtéshez szükség van egy kulcsra. Az előbbiek halmaza $\mathcal{K}$, a **kulcstér**, amely a lehetséges titkosító kulcsok halmaza, míg az utóbbiaké $\mathcal{K}'$ (a két halmaz akár azonos is lehet). A két kulcs nem feltétlenül azonos, sőt, bizonyos rendszereknél nem is lehet azonos. A fejtő kulcsot nyilván titokban kell tartani. Ezzel szemben a titkosító kulcs akár nyilvános is lehet, feltéve, hogy a fejtéshez használt kulcs ebből gyakorlatilag meghatározhatatlan. Ez persze eleve feltételezi, hogy a kulcstér mérete igen nagy, mert ellenkező esetben az összes kulcs kipróbálásával a titkosított szöveg nagy valószínűséggel fejthető. Most tehát azt tesszük fel, hogy van egy $k \in \mathcal{K}$ kulcs a titkosításhoz és egy $k' \in \mathcal{K}'$ kulcs a fejtéshez, ahol az sem kizárt, de nem is szükségszerű, hogy $k = k'$. Ekkor van az aktuális kulcstól függő $\mathcal{M}_k$ halmaz, amely a nyílt szövegeket tartalmazza, valamint a k'-höz tartozó $\mathcal{C}_{k'}$, a rejtjelezett szövegek halmaza. A titkosítás egy $E_k: \mathcal{M}_k \to U_k$, míg a fejtés egy $D_{k'}: \mathcal{C}_{k'} \to V_{k'}$ leképezés azzal a megkötéssel, hogy E_k min-

den kulcsra injektív, és minden $k \in \mathcal{K}$ kulcshoz létezik olyan $k' \in \mathcal{K}'$ kulcs, amellyel $E_k(\mathcal{M}_k) \subseteq \mathcal{C}_{k'}$ és az $\mathcal{M}_k$ bármely m elemére $D_{k'}\big(E_k(m)\big) = m$.

Az előbb leírtaknak megfelelően mára a titkosításnak két nagy osztálya van:

- **szimmetrikus** (**klasszikus**, **egykulcsos**) **titkosító rendszerek**;
- **nyilvános kulcsú** (**kétkulcsos**) **titkosító rendszerek**.

A két osztály megnevezésében az egykulcsos illetve kétkulcsos név kissé félrevezető. Ez csupán azt fejezi ki, hogy a klasszikus rendszereknél a titkosító és a fejtő kulcs igen gyakran azonos, így ekkor a rendszer valóban csak egy kulcsot használ, míg a nyilvános kulcsú rendszereknél a két kulcs szükségszerűen különböző.

Az első csoport olyan algoritmusokat használ, amelynél a titkosító és a fejtő kulcs vagy azonos, vagy a titkosító kulcsból a másik kulcs könnyedén meghatározható, így a rendszer biztonságos működéséhez azt is titokban kell tartani. Ez azzal is jár, hogy mindkét félnek rendelkeznie kell a közöttük folyó titkos kommunikációt biztosító kulcsokkal, vagyis valamilyen biztonságos csatornán keresztül (például futár által) azt mindkét félhez el kell juttatni. Ez olyan esetben, amikor csupán néhány személyről, néhány párról vagy kisebb csoportról van szó, ha nem is könnyen, de megvalósítható, ám akkor, amikor szinte mindenki részt vesz valamilyen titkosított üzenetváltásban (például rendelkezik bankkártyával), akkor ez nem használható módszer.

A nyilvános kulcsú rejtjelező rendszerek lehetősége Diffie és Hellman 1976-ban megjelent *New directions in cryptography* című cikkében került megfogalmazásra. Valójában ők elsősorban a **kulcscsere** problémáját akarták megoldani, vagyis azt mutatták meg, hogy lehetséges nyilvános csatornán, mindenki orra előtt kicserélni a titkos kulcsot úgy, hogy csupán az illetékesek ismerik meg a kulcsot. E mellett azonban a cikkükben felvetették a nyilvános kulcsú titkosítás gondolatát is. Amennyiben csupán a fejtéshez használt kulcsot kell titokban tartani, akkor nincs szükség a kulcsok biztonságos kicserélésére, hiszen a titkosító kulcs mindenki számára nyilvános lehet, a titokban tartandó, a fejtéshez használt kulcsot pedig csupán annak a személynek kell tudnia, akinek az üzeneteket az ő nyilvános kulcsával titkosítva küldik. Ezen túl a nyilvános kulcsú rendszer még azzal az előnnyel is jár, hogy kevesebb kulcsra van szükség. Ha ugyanis n résztvevő van, és bármely kettőnek kell egymással titkosítva kommunikálni, de minden pár külön titokkal rendelkezik, vagyis páronként más és más kulcsra van szükség, akkor összesen $\binom{n}{2} \sim n^2$ kulcsra, míg a nyilvános kulcsú rendszerben minden résztvevőnek csupán egy kulcsra (pontosabban **kulcspár**ra), azaz az n résztvevőnek együttesen csupán n kulcs-(pár)ra van szüksége.

A rejtjelezés tudománya a **kriptológia**. Ennek két ága van, a **kriptográfia** és a **kriptoanalízis**. Az előbbi a biztonságos rendszerek előállításával, megtervezésével, illetve magával a titkosítással foglalkozik, míg az utóbbinak az ilyen rendszerek gyenge pontjainak felkutatása, a rendszerek feltörhetőségének vizsgálata illetve feltörése a célja. Ilyen tevékenységet végeznek például a támadók. A passzív támadásnak több fokozatát különböztetjük meg:

- **csak rejtjelszövegű támadás**;
- **ismert nyílt szövegű támadás**;
- **választott (nyílt/rejtett) szövegű támadás**.

A csak rejtjelszövegű támadás esetén a támadónak csupán rejtjelezett szövegek állnak rendelkezésére, ezek vizsgálatával próbál olyan információkhoz jutni, amelyek segítségével meg tudja határozni a kulcsot, vagy ha a kulcsot nem is sikerül megtalálnia, de valamilyen más módon képes bármely, az adott kulccsal titkosított szöveget olvasni.

Az ismert nyílt szövegű támadás esetén a támadó rendelkezik néhány összetartozó nyílt szöveg - rejtjelezett szövegpárral, míg a választott szövegű támadó maga választhatja meg a pár valamelyik tagját, amelynek a párját szeretné megkapni. Ez nem annyira elképzelhetetlen, mint amennyire elsőként gondolnánk. Ha átutalok a banknak egy összeget, akkor tudom, hogy mi megy át titkosítva a csa-

tornán, és már csak meg kell figyelnem a forgalmat. De ezt alkalmazták az ellenség rejtjelrendszerének kifürkészésére régebben is: megszellőztettek egy-egy hírt, annak biztos tudatában, hogy ezt majd az érintett követség rejtjelezve továbbítja saját országának.

A választott szövegű támadás esetén a klasszikus rendszereknél általában a választott nyílt szövegű támadás fordul elő, aminek viszont nyilvános kulcsú rendszer esetén semmi értelme sincs, hiszen egy nyílt szöveg rejtjelezett párját a nyilvános kulccsal magunk is elő tudjuk állítani.

A választott szövegű támadás egy változata az **adaptív támadás**, amikor egy kapott válasz alapján kérjük a következő választ és így tovább. Az persze kizárt, hogy az éppen aktuális fejtendő szöveg nyílt párját kérjük (ami igen kényelmes lenne a támadónak).

Egy titkosító rendszert csak akkor tekintünk elfogadhatónak, ha ellenáll a választott szövegű támadásnak is (amely nyilván a legerősebb támadási forma).

Vannak egyéb támadási módok is, amelyek azonban nem algoritmikusak. Ilyen lehet a megvesztegetés, a zsarolás, a fenyegetés, a fizikai erőszak, kínzás alkalmazása, stb.

A fejezet végén nézzünk egy egyszerű példát.
Legyen a rejtjelezett szöveg az alábbi:

```
T HXZJZXBXRE JXHMXREZXDXA JSIRJTUTD AXBB BXDDSX TR TBTFMXJE
AHSFJTDTBSRSI CEWIRXHXAAXB SI, OSIRXD T MTHOTJE JTCTWTIEA
DTKQCXHJXAUXD UXYEBQTIEBOTJZTA T JXHMXRXIJ. XR EBQTDDQSHT
SKTR, OEKQ T JXHMXRXIS AHSJXHSLCEA DTKQ HXIRX MTBTCSBQXD
JTCTWTI XBBXDS MXWXAXRXI XBBXDIRXHXAXDJ IRLBXJXJJ. OT CXK
JLWZLA CEDWTDS, OEKQ XKQ HXZJZXBXRE CXDDQS XHEYXIRSJXI THTD
JEHOXJE YXB XKQ TWEJJ JTCTWTIS CEWIRXHHXB, TAAEH XRRXB
BXDQXKXUXD T HXZJZXBXREJ CSDEISJXJJLA, SKQ T AHSFJTDTBSRSIJ
CSDEISJXIHX SI OTIRDTBOTJZLA. SJJ CEIJ HEMSWXD EIIRXYEKBTBZLA
T CT SICXHJ AXJ BXKXHEIXUU JTCTWTI, T WSYYXHXDVSTBSI XI T
BSDXTHSI AHSFJTDTBSRSI TBTFZTSJ.
```

Az algoritmus minden egyes betűt ugyanazon szabály szerint az ábécé egy másik (az eredetitől nem feltétlenül különböző) betűjével helyettesít. Meg kell határozni az eredeti szöveget.

A fejtés alapja a **betűstatisztika**. Mivel minden betűt mindig ugyanazon szabály szerint helyettesítünk egy másik betűvel, ezért ahányszor a nyílt szövegben előfordult mondjuk a b, ugyanannyiszor található meg a b-nek megfeleltetett betű a fenti szövegben. A nyelvben bizonyos betűk nagy gyakorisággal fordulnak elő, mások viszont igen ritkák. Amennyiben a titkosított szöveg elég általános, nem valamilyen szakma speciális szövege, akkor az ábécé egyes betűi nagyjából a nyelvre jellemző statisztikát követve fordulnak elő a nyílt szövegben. Itt máris egy fontos dolgot látunk. Ahhoz, hogy egy titkosított szöveget fejteni tudjunk, tudnunk kell, hogy milyen nyelven íródott, ugyanis az egyes nyelvekben más a betűk eloszlása. A fenti szövegnél is tudnunk kell az eredeti szöveg nyelvét. Ennek a szövegnek az eredetije magyar nyelven íródott, így a magyar nyelv statisztikájára van szükségünk (az ékezetes betűket a megfelelő, ékezet nélküli betűvel, a kisbetűket nagybetűvel helyettesítettük, és csak a betűket helyettesítettük). Ez látható az 1. és a 2. táblázaton. Most készítsük el a rejtjelszövegünk statisztikáját. Ekkor a 3. táblázatot kapjuk.

A	1247	*H*	165	*O*	683	*U*	226
B	190	*I*	444	*P*	96	*V*	199
C	65	*J*	105	*Q*	0	*W*	9
D	197	*K*	541	*R*	367	*X*	1
E	1407	*L*	605	*S*	600	*Y*	261
F	78	*M*	352	*T*	761	*Z*	437
G	326	*N*	638				

1. táblázat

Az ékezet nélküli magyar nyelv statisztikája az ábécé sorrendjében (10 000 betűs szövegben)

E	1407	*K*	541	*U*	226	*P*	96
A	1247	*I*	444	*V*	199	*F*	78
T	761	*Z*	437	*D*	197	*C*	65
O	683	*R*	367	*B*	190	*W*	9
N	638	*M*	352	*H*	165	*X*	1
L	605	*G*	326	*J*	105	*Q*	0
S	600	*Y*	261				

2. táblázat

Az ékezet nélküli magyar nyelv statisztikája a gyakoriságok sorrendjében (10 000 betűs szövegben)

X	80	*H*	29	*W*	12	*Y*	6
T	66	*E*	28	*Z*	12	*F*	5
J	44	*R*	25	*O*	9	*V*	1
S	39	*A*	21	*M*	8	*G*	0
I	34	*C*	17	*L*	7	*N*	0
B	32	*Q*	13	*U*	6	*P*	0
D	30	*K*	12				

3. táblázat

A rejtjelezett szöveg statisztikája a gyakoriságok sorrendjében

A táblázatokból látjuk, hogy a magyar nyelvű nyílt szövegekben az *E* (itt, és a többi magánhangzó esetén is beleértve a megfelelő ékezetes betűt), majd az *A* gyakorisága jóval nagyobb a többi betű gyakoriságánál, és ugyanezt látjuk a rejtjelünk két leggyakoribb betűjénél, az *X*-nél és a *T*-nél. A harmadik leggyakoribb betűnél, *J*-nél is vélelmezhetjük, hogy az a magyarban harmadik leggyakoribb betűnek, *T*-nek felel meg. Cseréljük ki a rejtjelezett szövegben az előbb említett betűket (a kicserélt betűket aláhúzott, félkövér, dőlt karakterek jelölik).

A H**E**Z**T**Z**E**B**E**RE **TE**HM**E**REZ**E**D**E**A **T**SIR**TA**U**A**D A**E**BB B**E**DDS**E** **A**R **A**B**A**FM**ET**E AHSF**TA**D**A**BSRSI CEWIR**E**H**E**AA**E**B SI, OSIR**E**D **A** M**A**HO**AT**E **TA**C**A**W**A**IEA D**A**KQC**E**H**TE**AU**E**D U**E**YEBQ**A**IEBO**AT**Z**A**A **A** **TE**HM**E**R**E**I**T**. **E**R EBQ**A**DDQSH**A** SK**A**R, OEKQ **A** **TE**HM**E**R**E**IS AHS**TE**HSLCEA D**A**KQ H**E**IR**E** M**A**B**A**CSBQ**E**D **TA**C**A**W**A**I **E**BB**E**DS M**E**W**E**A**E**R**E**I **E**BB**E**DIR**E**H**E**A**E**D**T** IRLB**ETETT**. O**A** C**E**K **T**LWZLA CEDW**A**DS, OEKQ **E**KQ H**E**Z**T**Z**E**B**E**RE C**E**DDQS **E**HEY**E**IRS**TE**I **A**H**A**D **T**EHO**ET**E Y**E**B **E**KQ **A**WE**TT** **TA**C**A**W**A**IS CEWIR**E**HH**E**B, **A**AAEH **E**RR**E**B B**E**DQ**E**K**E**U**E**D **A** H**E**Z**T**Z**E**B**E**RE**T** CSDEIS**TETT**LA, SKQ **A** AHSF**TA**D**A**BSRSI**T** CSDEIS**TE**IH**E** SI O**A**IRD**A**BO**AT**ZLA. S**TT** CEI**T** HEMSW**E**D EIIR**E**YEKB**A**BZLA **A** C**A** SIC**E**H**T** A**ET** B**E**K**E**HEI**E**UU **TA**C**A**W**A**I, **A** WSYY**E**H**E**DVS**A**BSI **E**I **A** BSD**EA**HSI AHSF**TA**D**A**BSRSI **A**B**A**FZ**A**S**T**.

A fenti szövegben szerepel **A**R és **E**R. Bár más választás is értelmes eredményt adna, próbálkozzunk az $R \mapsto Z$ cserével:

A H**E**Z**T**Z**E**B**EZ**E **TE**HM**EZ**EZ**E**D**E**A **T**SI**ZTA**U**A**D A**E**BB B**E**DDS**E** **AZ** **A**B**A**FM**ET**E AHSF**TA**D**A**BS**Z**SI CEWI**ZE**H**E**AA**E**B SI, OSI**ZE**D **A** M**A**HO**AT**E **TA**C**A**W**A**IEA D**A**KQC**E**H**TE**AU**E**D U**E**YEBQ**A**IEBO**AT**Z**A**A **A** **TE**HM**EZE**I**T**. **EZ** EBQ**A**DDQSH**A** SK**AZ**, OEKQ **A** **TE**HM**EZE**IS AHS**TE**HSLCEA D**A**KQ H**E**I**ZE** M**A**B**A**CSBQ**E**D **TA**C**A**W**A**I **E**BB**E**DS M**E**W**E**A**EZE**I **E**BB**E**DI**ZE**H**E**A**E**D**T** I**Z**LB**ETETT**. O**A** C**E**K **T**LWZLA CEDW**A**DS, OEKQ **E**KQ H**E**Z**T**Z**E**B**EZ**E C**E**DDQS **E**HEY**E**I**Z**S**TE**I **A**H**A**D **T**EHO**ET**E Y**E**B **E**KQ **A**WE**TT** **TA**C**A**W**A**IS CEWI**ZE**HH**E**B, **A**AAEH **EZZE**B B**E**DQ**E**K**E**U**E**D **A** H**E**Z**T**Z**E**B**EZE**T CSDEIS**TETT**LA, SKQ **A** AHSF**TA**D**A**BS**Z**SI**T** CSDEIS**TE**IH**E** SI O**A**I**Z**D**A**BO**AT**ZLA. S**TT** CEI**T** HEMSW**E**D EII**ZE**YEKB**A**BZLA **A** C**A** SIC**E**H**T** A**ET** B**E**K**E**HEI**E**UU **TA**C**A**W**A**I, **A** WSYY**E**H**E**DVS**A**BSI **E**I **A** BSD**EA**HSI AHSF**TA**D**A**BS**Z**SI **A**B**A**FZ**A**S**T**.

Most látunk egy EZZEB szót, így szinte biztos, hogy a B az L helyett áll. Csere után lesz egy AELL, ami lehetne például MELL is, de azért inkább a KELL-lel próbálkozunk:

A HEZTZELEZE TEHMEZEZEDEK TSIZTAUAD KELL LEDDSE AZ ALAFMETE KHSFTADALSZSI CEWIZEHEKKEL SI, OSIZED A MAHOATE TACAWAIEK DAKQCEHTEKUED UEYELQAIELOATZAK A TEHMEZEIT. EZ ELQADDQSHA SKAZ, OEKQ A TEHMEZEIS KHSTEHSLCEK DAKQ HEIZE MALACSLQED TACAWAI ELLEDS MEWEKEZEI ELLEDIZEHEKEDT IZLLETETT. OA CEK TLWZLK CEDWADS, OEKQ EKQ HEZTZELEZE CEDDQS EHEYEIZSTEI AHAD TEHOETE YEL EKQ AWETT TACAWAIS CEWIZEHHEL, AKKEH EZZEL LEDQEKEUED A HEZTZELEZET CSDEISTETTLK, SKQ A KHSFTADALSZSIT CSDEISTEIHE SI OAIZDALOATZLK. STT CEIT HEMSWED EIIZEYEKLALZLK A CA SICEHT KET LEKEHEIEUU TACAWAI, A WSYYEHEDVSALSI EI A LSDEAHSI KHSFTADALSZSI ALAFZAST.

Ebben a szövegben nézzük a KHSFTADALSZSI szót. Azt látjuk, hogy a rejtett szövegbeli S a nyílt szövegben két mássalhangzó között áll, és ha a nyílt szövegben ennek az S-nek szintén S felelne meg, még akkor is három mássalhangzó következne egymás után (L, SZ és S vagy L, S és ZS), ami ismert magyar szóban nem található, így S valószínűleg egy magánhangzót helyettesít. Mivel E és A már „elkelt”, ezért S az I, O és U valamelyike lehet. Ha az S más előfordulásait is figyelembe vesszük, akkor valószínűsíthetjük, hogy S helyére I-t kell írnunk. Ekkor ADALIZII alapján D-t N helyett, I-t pedig S helyett gondolhatjuk. Az előbbi három csere után kapott szöveg a következő:

A HEZTZELEZE TEHMEZEZENEK TISZTAUAN KELL LENNIE AZ ALAFMETE KHIFTANALIZIS CEWSZEHEKKEL IS, OISZEN A MAHOATE TACAWASEK NAKQCEHTEKUEN UEYELQASELOATZAK A TEHMEZEST. EZ ELQANNQIHA IKAZ, OEKQ A TEHMEZESI KHITEHILCEK NAKQ HESZE MALACILQEN TACAWAS ELLENI MEWEKEZES ELLENSZEHEKENT SZLLETETT. OA CEK TLWZLK CENWANI, OEKQ EKQ HEZTZELEZE CENNQI EHEYESZITES AHAN TEHOETE YEL EKQ AWETT TACAWASI CEWSZEHHEL, AKKEH EZZEL LENQEKEUEN A HEZTZELEZET CINESITETTLK, IKQ A KHIFTANALIZIST CINESITESHE IS OASZNALOATZLK. ITT CEST HEMIWEN ESSZEYEKLALZLK A CA ISCEHT KET LEKEHESEUU TACAWAS, A WIYYEHENVIALIS ES A LINEAHIS KHIFTANALIZIS ALAFZAIT.

TISZTAUAN alapján U a B helyett, OISZEN alapján O a H helyett, míg IKAZ alapján K a G helyett áll. Csere után látjuk, hogy Q Y-t helyettesíti, majd a csere után adódik az $E \mapsto O$ váltás. Ez után természetes a $H \mapsto R$, $L \mapsto U$, $C \mapsto M$ csere, és a kapott szöveg

A REZTZELEZO TERMEZOZENEK TISZTABAN KELL LENNIE AZ ALAFMETO KRIFTANALIZIS MOWSZEREKKEL IS, HISZEN A MARHATO TAMAWASOK NAGYMERTEKBEN BEYOLYASOLHATZAK A TERMEZEST. EZ OLYANNYIRA IGAZ, HOGY A TERMEZESI KRITERIUMOK NAGY RESZE MALAMILYEN TAMAWAS ELLENI MEWEKEZES ELLENSZEREKENT SZULETETT. HA MEG TUWZUK MONWANI, HOGY EGY REZTZELEZO MENNYI EROYESZITES ARAN TORHETO YEL EGY AWOTT TAMAWASI MOWSZERREL, AKKOR EZZEL LENYEGEBEN A REZTZELEZOT MINOSITETTUK, IGY A KRIFTANALIZIST MINOSITESRE IS HASZNALHATZUK. ITT MOST ROMIWEN OSSZEYOGLALZUK A MA ISMERT KET LEGEROSEBB TAMAWAS, A WIYYERENVIALIS ES A LINEARIS KRIFTANALIZIS ALAFZAIT.

Végül a fenti szövegből kapjuk a $Z \mapsto J$, $M \mapsto V$, $F \mapsto P$, $W \mapsto D$, $Y \mapsto F$ és $V \mapsto C$ cseréket, és így az eredeti, nyílt szöveg

A REJTJELEZO TERVEZOJENEK TISZTABAN KELL LENNIE AZ ALAPVETO KRIPTANALIZIS MODSZEREKKEL IS, HISZEN A VARHATO TAMADASOK NAGYMERTEKBEN BEFOLYASOLHATJAK A TERVEZEST. EZ OLYANNYIRA

IGAZ, HOGY A TERVEZESI KRITERIUMOK NAGY RESZE VALAMILYEN TAMADAS ELLENI VEDEKEZES ELLENSZEREKENT SZULETETT. HA MEG TUDJUK MONDANI, HOGY EGY REJTJELEZO MENNYI EROFESZITES ARAN TORHETO FEL EGY ADOTT TAMADASI MODSZERREL, AKKOR EZZEL LENYEGEBEN A REJTJELEZOT MINOSITETTUK, IGY A KRIPTANALIZIST MINOSITESRE IS HASZNALHATJUK. ITT MOST ROVIDEN OSSZEFOGLALJUK A MA ISMERT KET LEGEROSEBB TAMADAS, A DIFFERENCIALIS ES A LINEARIS KRIPTANALIZIS ALAPJAIT.

illetve „olvasható” formában

A rejtjelező tervezőjének tisztában kell lennie az alapvető kriptanalízis módszerekkel is, hiszen a várható támadások nagymértékben befolyásolhatják a tervezést. Ez olyannyira igaz, hogy a tervezési kritériumok nagy része valamilyen támadás elleni védekezés ellenszereként született. Ha meg tudjuk mondani, hogy egy rejtjelező mennyi erőfeszítés árán törhető fel egy adott támadási módszerrel, akkor ezzel lényegében a rejtjelezőt minősítettük, így a kriptanalízist minősítésre is használhatjuk. Itt most röviden összefoglaljuk a ma ismert két legerősebb támadás, a differenciális és a lineáris kriptanalízis alapjait.

(Az idézet Buttyán Levente és Vajda István *Kriptográfia és Alkalmazásai* című könyve 2004-es kiadásának 52. oldalán található.)

Az előbbiek alapján meg tudjuk adni a kódtáblát:

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*
T	*U*	*V*	*W*	*X*	*Y*	*K*	*O*	*S*	*Z*	*A*	*B*	*C*	*D*	*E*	*F*		*H*	*I*	*J*	*L*	*M*			*Q*	*R*

A hiányzó helyeken olyan betűk vannak, amelyek nem fordultak elő a szövegben. Azonban figyelmesen megtekintve a kitöltött helyeket, azt látjuk, hogy a nyílt szöveg K betűjétől kezdve, ciklikusan tekintve F-ig, a második sorban lényegében véve az ábécé betűi a természetes sorrendjükben állnak, kivéve a felső sorban G-től H-ig terjedő rész alatti betűket. Ez a kódtábla az úgynevezett **kulcsszó Cæsar** helyettesítésnek felel meg: választunk egy könnyen megjegyezhető, viszonylag rövid **kulcsszó**t, ebből elhagyjuk azokat a betűket, amelyek korábban már előfordultak benne, és ezt az ábécé egy adott betűjétől kezdve – ha kell, ciklikusan – leírjuk, majd az utolsó betű után elkezdjük az ábécé kimaradt betűit sorban egymás után írni. Most a kulcsszó *koszos* volt, ebből elhagyva az ismétléseket kapjuk a *KOSZ*-t, és ezt G-től kezdve írva, majd utána az ábécé kimaradt betűivel folytatva kapjuk a **helyettesítő táblá**t. Ennek megfelelően a kimaradt helyekre sorban G-t, N-et és P-t kell írni.

Látjuk, hogy a betűk eloszlása alapján viszonylag könnyen meg tudtuk fejteni a rejtjelezett szöveget. Ehhez tudnunk kellett, hogy milyen nyelven íródott, és milyen algoritmussal titkosítottunk. Természetesen azt is látni kell, a fejtést egyszerűsítette, hogy a rejtett szövegben benne voltak a szóközök és az írásjelek, ám ezek nélkül is fejthető lenne a szöveg (érdemes ezeket kihagyni, és az így nyert szöveget megpróbálni fejteni). Majd látjuk, hogy egy ilyen egyszerű titkosítás esetén már egészen rövid szöveg lényegében véve egyértelműen desifrírozható. Ilyen kérdéseket vizsgálunk a következő két fejezetben.

Vissza a tartalomhoz

2. Az entrópiáról

Az **entrópia** információelméleti fogalmát **Shannon** határozta meg. Korábban **Heartley** vizsgálta matematikai szempontból az **információt**, és úgy találta, hogy ha n különböző üzenet lehetséges, akkor egy-egy **üzenet információtartalma**, az **egyedi információmennyiség** $I = \log n$. E szerint a kifejezés szerint azonban a különböző üzenetek azonos mennyiségű információt, új ismeretet közölnek a fogadóval. Ezzel szemben Shannon úgy gondolta, hogy egy üzenet annál több információt szolgáltat, minél váratlanabb, minél kevésbé lehet rá számítani, azaz minél kisebb a valószínűsége. Ha X egy véges eseménytér, az üzenetek halmaza, és az $x_i \in X$ üzenet p_i valószínűséggel fordul elő, akkor tehát Shannon szerint az x_i üzenet $I(p_i)$ információt szolgáltat, ahol I egyelőre ismeretlen függvény. A teljes **üzenethalmaz átlagos információtartalma** az egyes üzenetek egyedi információtartalmának várható értéke, $H(p_{n-1}, \cdots, p_0) = \sum_{i=0}^{n-1} p_i I(p_i)$, ahol n a különböző üzenetek száma és H az **entrópiafüggvény**. Mivel egyelőre I ismeretlen, ezért H-t sem ismerjük. H meghatározásához bizonyos feltételeket kell megfogalmazni. Egy lehetséges axiomatikus bevezetés az alábbi kikötéseket tartalmazza:

1. $(p_{n-1}, \cdots, p_0) \in]0,1]^n \subseteq \mathbb{R}^n$ véges diszkrét valószínűségi eloszlás;
2. $H(p_{n-1}, \cdots, p_0)$ a változóinak szimmetrikus függvénye;
3. $H(tp_{n-1}, (1-t)p_{n-1}, \cdots, p_0) = H(p_{n-1}, \cdots, p_0) + p_{n-1}H(t, 1-t)$, ha $t \in]0,1[\subseteq \mathbb{R}$;
4. $H(t, 1-t)$ t-nek folytonos függvénye, ha $t \in]0,1[\subseteq \mathbb{R}$;
5. $H\left(\frac{1}{2}, \frac{1}{2}\right) > 0$.

A fenti feltételeknek pontosan egy folytonos függvény, $H(p_{n-1}, \cdots, p_0) = -\sum_{i=0}^{n-1} p_i \log p_i$ felel meg, és ebből leolvasva $I(p_i) = -\log p_i$. Ha minden üzenet valószínűsége azonos, tehát bármelyik $\frac{1}{n}$ valószínűséggel fordul elő, akkor valóban igaz, hogy az egyedi üzenetek által közvetített információ mértéke $I = \log n$. Általános esetben viszont az egyes üzenetek bekövetkezése különböző valószínűséggel történik, tehát általában $I(p_i) \neq \log n$. A valós értékű logaritmusfüggvény csak a pozitív valós számokra értelmezett, és ha x a pozitív valós számokon keresztül tart a 0-hoz, akkor a logaritmusfüggvény értéke abszolút értékben a ∞-hez tart, így $|x \log x| \to 0 \cdot \infty$. De $\lim_{x \to 0+0}(x \log x)$ létezik, és 0-val egyenlő, ezért az entrópiafüggvényt kiterjeszthetjük arra az esetre is, amikor egy vagy több valószínűség értéke 0, azzal, hogy ekkor $p_i \log p_i = 0$.

Az előbbi felírásban nem adtuk meg konkrétan a logaritmus alapját, ám erre nincs is szükség. Ha ugyanis egy alapról áttérünk egy másikra, az csupán a mértékegység megváltozását jelenti (hasonlóan mondjuk a méterhez és lábhoz), hiszen $\log_a u = \log_a b \cdot \log_b u$. Magát a logaritmus r alapját $H\left(\frac{1}{2}, \frac{1}{2}\right) = c$ határozza meg, ugyanis $c = H\left(\frac{1}{2}, \frac{1}{2}\right) = -\left(\frac{1}{2}\log_r \frac{1}{2} + \frac{1}{2}\log_r \frac{1}{2}\right) = \log_r 2$-ből $r = 2^{\frac{1}{c}} > 1$. Az alap szokásos értéke az információelméletben 2, és ekkor az entrópia egysége a **bit**. Ezt az elnevezést **John W. Tukey** vezette be a **binary digit** rövidítéseként. Tekintettel arra, hogy ugyanez a neve egy kettes számrendszerben felírt szám egy-egy számjegyének, ezért megkülönböztetésül az információelméleti egységet szokás **binary unit**-ként, a **binary unit** rövidítéseként említeni.

Ha a p_i valószínűségek az X eseményhalmaz elemei előfordulásainak a valószínűségei, akkor $H(p_{n-1}, \cdots, p_0)$ tulajdonképpen az X tér entrópiája, ezért ezt az értéket $H(X)$-szel is jelölhetjük.

A fenti H-függvény a **Shannon-féle entrópiafüggvény**. Léteznek általánosabb kifejezések is az entrópiára. Egyik a $H_\alpha(p_{n-1}, \cdots, p_0) = \frac{1}{1-\alpha}\log \sum_{i=0}^{n-1} p_i^\alpha$ **Rényi-féle entrópia**, ahol $1 > \alpha \in \mathbb{R}_0^+$. Ez a kifejezés határértékként tartalmazza a Shannon-féle entrópiát, ha α balról tart 1-hez.

Csupán az érdekesség és bizonyos patriotikus büszkeség miatt jegyezzük meg, hogy Shannonnak **Neumann János** javasolta az entrópia elnevezést, lévén, hogy a kifejezés matematikailag hasonló alakú, mint a korábbi fizikai entrópia. Az elnevezést a formális hasonlóságon kívül bizonyos tartalmi azonosságok is alátámasztják, bár igen komoly eltérések is kimutathatóak a két entrópiafogalom között, amiért többen károsnak tartják az azonos megnevezést. Shannonnak más „magyar kapcsolata” is volt: foglalkozott sakkautomatával, és ezzel kap-

csolatban megemlítette **Kempelen Farkas** nevét, valamint a kommunikációról szólva **Gábor Dénes**t nevezi meg egyik úttörőként.

A továbbiakban részletesebben megvizsgáljuk az entrópiát. Mindenekelőtt bizonyítás nélkül ismertetjük a konvex függvények néhány jellemzőjét.

2.1. Definíció

Legyen $f: X \to \mathbb{R}$, ahol $X \subseteq \mathbb{R}$, és $I \subseteq X$ egy intervallum, továbbá a és $b \neq a$ az I két eleme. Ekkor $h_{a,b}(x) = f(a) + \frac{f(b)-f(a)}{b-a}(x-a)$ az $f\left(a, f(a)\right)$, $\left(b, f(b)\right)$ **pontjain átmenő szelő**je, és ennek a két pont közé eső része az $f\left(a, f(a)\right)$, $\left(b, f(b)\right)$ **pontjait összekötő húr**.

Δ

2.2. Definíció

Legyen $X \subseteq \mathbb{R}$, $f: X \to \mathbb{R}$, és $I \subseteq X$ egy intervallum. f **konvex az** I **intervallumon**, ha az I bármely $a < c < b$ elemeire $f(c) \leq h_{a,b}(c)$. f **szigorúan konvex az** I **intervallumon**, ha konvex I-n és az előbbi egyenlőtlenségben mindig a szigorú egyenlőtlenség is teljesül. f **konkáv** illetve **szigorúan konkáv az** I **intervallumon**, ha $-f$ konvex illetve szigorúan konvex I-n.

Δ

2.3. Tétel

Legyen $f: X \to \mathbb{R}$, ahol $X \subseteq \mathbb{R}$, és $I \subseteq X$ egy intervallum. f akkor és csak akkor konvex az I intervallumon, ha $f(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda)f(b)$ az I bármely a, b elemére és tetszőleges $0 < \lambda < 1$ valós számra, és akkor és csak akkor szigorúan konvex, ha $a \neq b$ esetén az előbbi egyenlőtlenség bal oldala mindig kisebb a jobb oldali értéknél.

Δ

2.4. Tétel (Jensen-egyenlőtlenség)

Legyen $f: X \to \mathbb{R}$, ahol $X \subseteq \mathbb{R}$, és $I \subseteq X$ egy intervallum. f akkor és csak akkor konvex az I intervallumon, ha minden $n \in \mathbb{N}^+$, $\{a_i \in I | n \geq i \in \mathbb{N}^+\}$ és $\{0 < \lambda_i \in \mathbb{R} | n \geq i \in \mathbb{N}^+ \wedge \sum_{i=1}^{n} \lambda_i = 1\}$ esetén $f(\sum_{i=1}^{n} \lambda_i a_i) \leq \sum_{i=1}^{n} \lambda_i f(a_i)$, és akkor és csak akkor szigorúan konvex, ha az előbbi egyenlőtlenség bal oldala mindig kisebb a jobb oldali értéknél, ha az a_i-k nem mindegyike azonos.

Δ

2.5. Megjegyzés

Ha $n \in \mathbb{N}^+$, minden $n \geq i \in \mathbb{N}^+$-ra $\mu_i \in \mathbb{R}^+$, és $\sum_{i=1}^{n} \mu_i = \mu$, akkor minden előbbi i indexre $0 < \lambda_i = \frac{\mu_i}{\mu} \in \mathbb{R}$, és $\sum_{i=1}^{n} \lambda_i = 1$.

Δ

2.6. Tétel

Legyen $n \in \mathbb{N}^+$, minden $n \geq i \in \mathbb{N}^+$-ra $0 \leq a_i \in \mathbb{R}$ és $b_i \in \mathbb{R}^+$, $a = \sum_{i=1}^{n} a_i$ és $b = \sum_{i=1}^{n} b_i$. Ekkor 1-nél nagyobb alap esetén $\sum_{i=1}^{n} a_i \log \frac{b_i}{a_i} \leq a \log \frac{b}{a}$, és egyenlőség akkor és csak akkor teljesül, ha minden i-re $\frac{b_i}{a_i} = \frac{b}{a}$.

Δ

Bizonyítás:
1-nél nagyobb alap esetén a logaritmusfüggvény a teljes értelmezési tartományában szigorúan konkáv. Ekkor a Jensen-egyenlőtlenséggel

$$\sum_{i=1}^{n} a_i \log \frac{b_i}{a_i} = a \sum_{i=1}^{n} \frac{a_i}{a} \log \frac{b_i}{a_i} \leq a \log \sum_{i=1}^{n} \frac{a_i}{a} \frac{b_i}{a_i} = a \log \frac{b}{a},$$

és egyenlőség akkor és csak akkor lesz, ha minden i-re $\frac{b_i}{a_i}$ azonos. Legyen $\frac{b_i}{a_i} = c$. Ekkor $b_i = ca_i$, tehát $b = ca$, és innen $c = \frac{b}{a}$.

□

2.7. Következmény

Ha az előbbi tételben

a) $a = b$ (speciális esetként $a = 1 = b$), akkor $\sum_{i=1}^{n} a_i \log \frac{b_i}{a_i} \leq 0$, vagyis $\sum_{i=1}^{n} a_i \log b_i \leq \sum_{i=1}^{n} a_i \log a_i$, és a két oldal akkor és csak akkor egyenlő, ha valamennyi i-re $a_i = b_i$;
b) $a = 1$ és minden i-re $b_i = 1$, akkor $-\sum_{i=1}^{n} a_i \log a_i = \sum_{i=1}^{n} a_i \log \frac{1}{a_i} = \sum_{i=1}^{n} a_i \log \frac{b_i}{a_i} \leq \log n$, és pontosan akkor lesz a két oldal egyenlő, ha valamennyi i-re $a_i = \frac{1}{n}$.

Δ

Bizonyítás:
Mindkét állítás közvetlenül kapható az előző tételből.

□

2.8. Definíció

Legyen $m \in \mathbb{N}^+$, $n \in \mathbb{N}^+$, $\underline{\xi} = (\xi_1, \cdots, \xi_m)$ valószínűségi változó, $\underline{\xi} \in \left\{ \underline{x}_k \in \mathbb{R}^m \middle| n \geq k \in \mathbb{N}^+ \right\}$, $n \geq k \in \mathbb{N}^+$-ra $p_k = P\left(\underline{\xi} = \underline{x}_k\right)$ és $1 < r \in \mathbb{R}$. Ekkor $H_r = H_r\left(\underline{\xi}\right) = -\sum_{k=1}^{n} p_k \log_r p_k$ a $\underline{\xi}$ (r-**alapú**) **entrópiája**, és $i_{\underline{\xi}} = -\log_r p\left(\underline{\xi}\right)$ az (r-**alapú**) **entrópia-sűrűség** vagy **egyedi entrópia**.

Δ

$r = 2$-nél az entrópia egysége a bit (13. oldal). Ha nem szükséges, külön nem jelöljük r-et.

2.9. Tétel

Legyen $m \in \mathbb{N}^+$, $n \in \mathbb{N}^+$, $\underline{\xi} = (\xi_1, \cdots, \xi_m)$ valószínűségi változó, $\underline{\xi} \in \left\{ \underline{x}_k \in \mathbb{R}^m \middle| n \geq k \in \mathbb{N}^+ \right\}$, és legyen $n \geq k \in \mathbb{N}^+$-ra $p_k = P\left(\underline{\xi} = \underline{x}_k\right)$ és $1 < r \in \mathbb{R}$. Ekkor $0 \leq H_r \leq \log_r n$, továbbá $H_r = 0$ akkor és csak akkor, ha van olyan l, amellyel $p_l = P\left(\underline{\xi} = \underline{x}_l\right) = 1$ és $p_k = P\left(\underline{\xi} = \underline{x}_k\right) = 0$ minden más indexre, míg $H_r = \log_r n$ pontosan akkor igaz, ha minden k-ra $p_k = P\left(\underline{\xi} = \underline{x}_k\right) = \frac{1}{n}$.

Δ

Bizonyítás:
Mivel $0 \leq p_k \leq 1$ és $1 < r$, ezért $p_k \log_r p_k \leq 0$, tehát $-\sum_{k=1}^{n} p_k \log_r p_k \geq 0$, és az összeg csak úgy lehet 0, ha minden tagja 0. Ha $0 < p_k < 1$, akkor $\log_r p_k < 0$ és $p_k \log_r p_k \neq 0$, így $H_r = 0$

esetén minden k-ra p_k csak 0 vagy 1 lehet. Ugyanakkor $\sum_{k=1}^{n} p_k = 1$, ezért nem lehet minden k-ra $p_k = 0$ és nem lehet egynél több k-ra $p_k = 1$, vagyis $H_r = 0$ akkor és csak akkor, ha egy és csak egy k indexre $p_k = 1$, és minden más indexre $p_k = 0$.

Ami az entrópia maximumát illeti, az a 2.7. Következmény b) pontjából adódik.

□

A $H(p_{n-1}, \cdots, p_0)$ függvénynek a fentiek szerint az értelmezési tartományában pontosan egy maximuma van, a $(p_{n-1}, \cdots, p_0) = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$ helyen, és ekkor az értéke $\log n$. Ez az entrópia intuitív értelmezése alapján világos, hiszen átlagosan akkor jutunk a legtöbb információhoz, akkor lehet a legkevésbé megjósolni a soron következő üzenetet, ha lényegében véve semmit sem tudunk az egyes üzenetekről, bármelyik esemény azonos valószínűséggel következhet be.

Most legyen $\underline{\eta} = (\eta_1, \cdots, \eta_u)$ is egy valószínűségi változó az $Y = \left\{\underline{y}_k \in \mathbb{R}^u \middle| v \geq k \in \mathbb{N}^+\right\}$ értékekkel és $q_k = P\left(\underline{\eta} = \underline{y}_k\right)$ valószínűségekkel. Ekkor valamennyi rögzített $v \geq l \in \mathbb{N}^+$-ra létezik a $t_{k|l} = P\left(\underline{\xi} = \underline{x}_k \middle| \underline{\eta} = \underline{y}_l\right)$ feltételes eloszlás és a $H\left(\underline{\xi} \middle| \underline{\eta} = \underline{y}_l\right) = -\sum_{k=1}^{n} t_{k|l} \log t_{k|l}$ entrópia.

2.10. Definíció

Legyen $\underline{\xi} = (\xi_1, \cdots, \xi_m)$ és $\underline{\eta} = (\eta_1, \cdots, \eta_u)$ valószínűségi változó, és $\underline{\eta}$ lehetséges értékeinek halmaza $Y = \left\{\underline{y}_k \in \mathbb{R}^u \middle| v \geq k \in \mathbb{N}^+\right\}$. Ekkor $H\left(\underline{\xi} \middle| \underline{\eta}\right) = E\left(H\left(\underline{\xi} \middle| \underline{\eta} = \underline{y}_l\right)\right)$ a **$\underline{\xi}$-nek $\underline{\eta}$-ra vonatkozó feltételes entrópiája** (E a várható érték).

Δ

2.11. Tétel

Legyen $m \in \mathbb{N}^+$, $n \in \mathbb{N}^+$, $u \in \mathbb{N}^+$, $v \in \mathbb{N}^+$, $\underline{\xi} = (\xi_1, \cdots, \xi_m)$ és $\underline{\eta} = (\eta_1, \cdots, \eta_u)$ valószínűségi változó, $X = \left\{\underline{x}_k \in \mathbb{R}^m \middle| n \geq k \in \mathbb{N}^+\right\}$ a $\underline{\xi}$, $Y = \left\{\underline{y}_k \in \mathbb{R}^u \middle| v \geq k \in \mathbb{N}^+\right\}$ az $\underline{\eta}$ lehetséges értékeinek halmaza, $n \geq k \in \mathbb{N}^+$-ra és $v \geq l \in \mathbb{N}^+$-ra $r_{k,l} = P\left(\underline{\xi} = \underline{x}_k, \underline{\eta} = \underline{y}_l\right)$, és $t_{k|l} = P\left(\underline{\xi} = \underline{x}_k \middle| \underline{\eta} = \underline{y}_l\right)$. Ekkor $H\left(\underline{\xi} \middle| \underline{\eta}\right) = -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log t_{k|l}$, $0 \leq H\left(\underline{\xi} \middle| \underline{\eta}\right) \leq H\left(\underline{\xi}\right)$, és $H\left(\underline{\xi} \middle| \underline{\eta}\right) = 0$ akkor és csak akkor, ha létezik olyan f függvény, hogy 1 valószínűséggel $\underline{\xi} = f\left(\underline{\eta}\right)$, míg $H\left(\underline{\xi} \middle| \underline{\eta}\right) = H\left(\underline{\xi}\right)$ pontosan akkor teljesül, ha $\underline{\xi}$ és $\underline{\eta}$ függetlenek.

Δ

Bizonyítás:

$H\left(\underline{\xi} \middle| \underline{\eta} = \underline{y}_l\right)$ entrópia, így $0 \leq H\left(\underline{\xi} \middle| \underline{\eta} = \underline{y}_l\right)$, de akkor $H\left(\underline{\xi} \middle| \underline{\eta}\right) = E\left(H\left(\underline{\xi} \middle| \underline{\eta} = \underline{y}_l\right)\right) \geq 0$ is teljesül, és a várható érték akkor és csak akkor lesz 0, ha minden l-re $H\left(\underline{\xi} \middle| \underline{\eta} = \underline{y}_l\right) = 0$. Ez pontosan akkor következik be, ha minden l indexhez pontosan egy i_l indexre $t_{i_l|l} = 1$, és minden más indexre $t_{k|l} = 0$. Ez azt jelenti, hogy minden $\underline{y}_l$-hez van egy és csak egy $\underline{x}_{i_l}$, hogy $P\left(\underline{\xi} = \underline{x}_{i_l} \middle| \underline{\eta} = \underline{y}_l\right) = 1$, és minden más k indexre $P\left(\underline{\xi} = \underline{x}_k \middle| \underline{\eta} = \underline{y}_l\right) = 0$, vagyis létezik egy f függvény, amellyel 1 valószínűséggel $\underline{\xi} = f\left(\underline{\eta}\right)$, $\underline{\xi}$ az $\underline{\eta}$ függvénye.

Nézzük a másik határt. Legyen $q_k = P\left(\underline{\eta} = \underline{y}_k\right)$. Ekkor a Jensen-egyenlőtlenséggel

$$\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log \frac{p_k q_l}{r_{k,l}} \leq \log \sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \frac{p_k q_l}{r_{k,l}} = \log \sum_{l=1}^{v}\sum_{k=1}^{n} p_k q_l = \log 1 = 0,$$

és ezt alkalmazva

$$H\left(\underline{\xi}\right) - H\left(\underline{\xi}\middle|\underline{\eta}\right) = -\sum_{k=1}^{n} p_k \log p_k + \sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log t_{k|l}$$
$$= -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log p_k + \sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log t_{k|l}$$
$$= -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log \frac{p_k}{t_{k|l}} = -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log \frac{p_k q_l}{r_{k,l}} \geq 0.$$

$H\left(\underline{\xi}\right) - H\left(\underline{\xi}\middle|\underline{\eta}\right) \geq 0$-ból $H\left(\underline{\xi}\right) \geq H\left(\underline{\xi}\middle|\underline{\eta}\right)$, továbbá egyenlőség akkor és csak akkor lesz, ha minden k, l indexre $p_k q_l = r_{k,l}$, vagyis pontosan akkor, ha $\underline{\xi}$ és $\underline{\eta}$ függetlenek.

□

Az előbbihez hasonlóan definiáljuk a $H\left(\underline{\eta}\middle|\underline{\xi}\right)$ feltételes entrópiát, és erre hasonló tulajdonság igaz, mint az előző entrópiára.

2.12. Definíció

Legyen $m \in \mathbb{N}^+$, $n \in \mathbb{N}^+$, $u \in \mathbb{N}^+$, $v \in \mathbb{N}^+$, $\underline{\xi} = (\xi_1, \cdots, \xi_m)$ és $\underline{\eta} = (\eta_1, \cdots, \eta_u)$ valószínűségi változó, $X = \left\{\underline{x}_k \in \mathbb{R}^m \middle| n \geq k \in \mathbb{N}^+\right\}$ a $\underline{\xi}$, $Y = \left\{\underline{y}_k \in \mathbb{R}^u \middle| v \geq k \in \mathbb{N}^+\right\}$ az $\underline{\eta}$ lehetséges értékeinek halmaza, és $n \geq k \in \mathbb{N}^+$-ra és $v \geq l \in \mathbb{N}^+$-ra $r_{k,l} = P\left(\underline{\xi} = \underline{x}_k, \underline{\eta} = \underline{y}_l\right)$. Ekkor $\underline{\xi}$ **és** $\underline{\eta}$ **együttes entrópiája** $H\left(\underline{\xi}, \underline{\eta}\right) = -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log r_{k,l}$.

Δ

2.13. Tétel

A 2.12. Definíció jelöléseivel $0 \leq H\left(\underline{\xi}, \underline{\eta}\right) = H\left(\underline{\xi}\right) + H\left(\underline{\eta}\middle|\underline{\xi}\right) \leq H\left(\underline{\xi}\right) + H\left(\underline{\eta}\right)$, és akkor és csak akkor lesz $H\left(\underline{\xi}, \underline{\eta}\right) = H\left(\underline{\xi}\right) + H\left(\underline{\eta}\right)$, ha $\underline{\xi}$és $\underline{\eta}$ függetlenek.

Δ

Természetesen a fenti állítások $\underline{\xi}$ és $\underline{\eta}$ felcserélésével is teljesülnek.

Bizonyítás:

$H\left(\underline{\xi}, \underline{\eta}\right) \geq 0$ a definíció közvetlen következménye, továbbá ha $t_{k|l} = P\left(\underline{\xi} = \underline{x}_k \middle| \underline{\eta} = \underline{y}_l\right)$ valamint $q_k = P\left(\underline{\eta} = \underline{y}_l\right)$, akkor

$$H\left(\underline{\xi}, \underline{\eta}\right) = -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log r_{k,l} = -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log\left(t_{k|l} q_l\right)$$
$$= -\sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log q_l - \sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log t_{k|l}$$
$$= -\sum_{l=1}^{v} q_l \log q_l - \sum_{l=1}^{v}\sum_{k=1}^{n} r_{k,l} \log t_{k|l} = H\left(\underline{\eta}\right) + H\left(\underline{\xi}\middle|\underline{\eta}\right).$$

$H\left(\underline{\xi}\middle|\underline{\eta}\right) \leq H\left(\underline{\xi}\right)$, ezért $H\left(\underline{\xi},\underline{\eta}\right) \leq H\left(\underline{\xi}\right) + H\left(\underline{\eta}\right)$, és $H\left(\underline{\xi}\middle|\underline{\eta}\right) = H\left(\underline{\xi}\right)$ akkor és csak akkor, ha $\underline{\xi}$ és $\underline{\eta}$ függetlenek, így igaz a tétel utolsó állítása is.

□

A $\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(w)}$ valószínűségi változók együttes entrópiája és az $\underline{\eta}^{(1)}, \cdots, \underline{\eta}^{(z)}$ valószínűségi változókra vonatkozó feltételes entrópiája hasonlóan definiálható, és indukcióval könnyen belátható, hogy bármely $1 \leq l \leq w$-re

$$H\left(\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(w)}\right) = H\left(\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(l)}\right) + \sum_{i=l+1}^{w} H\left(\underline{\xi}^{(i)}\middle|\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(i-1)}\right)$$

és

$$H\left(\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(w)}\middle|\underline{\eta}^{(1)}, \cdots, \underline{\eta}^{(z)}\right) = H\left(\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(l)}\middle|\underline{\eta}^{(1)}, \cdots, \underline{\eta}^{(z)}\right) + \sum_{i=l+1}^{w} H\left(\underline{\xi}^{(i)}\middle|\underline{\eta}^{(1)}, \cdots, \underline{\eta}^{(z)}, \underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(i-1)}\right),$$

vagyis például $l = 1$ esetén

$$H\left(\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(w)}\right) = H\left(\underline{\xi}^{(1)}\right) + \sum_{i=2}^{w} H\left(\underline{\xi}^{(i)}\middle|\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(i-1)}\right)$$

valamint

$$H\left(\underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(w)}\middle|\underline{\eta}^{(1)}, \cdots, \underline{\eta}^{(z)}\right) = H\left(\underline{\xi}^{(1)}\middle|\underline{\eta}^{(1)}, \cdots, \underline{\eta}^{(z)}\right) + \sum_{i=2}^{w} H\left(\underline{\xi}^{(i)}\middle|\underline{\eta}^{(1)}, \cdots, \underline{\eta}^{(z)}, \underline{\xi}^{(1)}, \cdots, \underline{\xi}^{(i-1)}\right).$$

A $H\left(\underline{\xi}\right) \geq H\left(\underline{\xi}\middle|\underline{\eta}\right)$ egyenlőtlenség azt fejezi ki, hogy ha $\underline{\xi}$-ről már van valamilyen előzetes ismeretünk, akkor legfeljebb annyi új információhoz jutunk, mint az előbbi ismeretek nélkül. $H\left(\underline{\xi},\underline{\eta}\right) = H\left(\underline{\xi}\right) + H\left(\underline{\eta}\middle|\underline{\xi}\right)$ viszont azt jelenti, hogy az együttes eloszlás átlagos információtartalmát például úgy kapjuk meg, hogy meghatározzuk önmagában a $\underline{\xi}$ információtartalmát, és ehhez még hozzávesszük azt az információmennyiséget, amelyet már a $\underline{\xi}$ ismeretében $\underline{\eta}$-ról nyerhetünk.

Most egy fontos fogalmat definiálunk.

2.14. Definíció

Legyen $\underline{\xi}$ és $\underline{\eta}$ valószínűségi változó. Ekkor $I\left(\underline{\xi},\underline{\eta}\right) = H\left(\underline{\xi}\right) - H\left(\underline{\xi}\middle|\underline{\eta}\right)$ a $\underline{\xi}$ **és** $\underline{\eta}$ **kölcsönös információja**.

Δ

2.15. Tétel

$0 \leq I\left(\underline{\xi},\underline{\eta}\right) \leq \min\left\{H\left(\underline{\xi}\right), H\left(\underline{\eta}\right)\right\}$. $I\left(\underline{\xi},\underline{\eta}\right) = 0$ akkor és csak akkor, ha $\underline{\xi}$ és $\underline{\eta}$ függetlenek, és $I\left(\underline{\xi},\underline{\eta}\right) = H\left(\underline{\xi}\right)$, ha $\underline{\xi}$ 1 valószínűséggel az $\underline{\eta}$ függvénye.

Δ

Bizonyítás:
$H\left(\underline{\xi}\right)+H\left(\underline{\eta}\middle|\underline{\xi}\right)=H\left(\underline{\xi},\underline{\eta}\right)=H\left(\underline{\eta}\right)+H\left(\underline{\xi}\middle|\underline{\eta}\right)$-ból $H\left(\underline{\xi}\right)-H\left(\underline{\xi}\middle|\underline{\eta}\right)=H\left(\underline{\eta}\right)-H\left(\underline{\eta}\middle|\underline{\xi}\right)$, így az is igaz, hogy $I\left(\underline{\xi},\underline{\eta}\right)=H\left(\underline{\eta}\right)-H\left(\underline{\eta}\middle|\underline{\xi}\right)$. A tételben megfogalmazott állítások ezek után következnek a $H\left(\underline{\xi}\right)$ és $H\left(\underline{\xi}\middle|\underline{\eta}\right)$ közötti (illetve a $H\left(\underline{\eta}\right)$ és $H\left(\underline{\eta}\middle|\underline{\xi}\right)$ közötti hasonló) összefüggésekből.

□

Szintén $H\left(\underline{\xi}\right)+H\left(\underline{\eta}\middle|\underline{\xi}\right)=H\left(\underline{\xi},\underline{\eta}\right)=H\left(\underline{\eta}\right)+H\left(\underline{\xi}\middle|\underline{\eta}\right)$-ból látható, hogy a kölcsönös információ kifejezhető az $I\left(\underline{\xi},\underline{\eta}\right)=H\left(\underline{\xi}\right)+H\left(\underline{\eta}\right)-H\left(\underline{\xi},\underline{\eta}\right)$ alakban is, amiből látszik, hogy a kölcsönös információ szimmetrikus a két változójában.

A kölcsönös információ azt fejezi ki, hogy $\underline{\eta}$-t megfigyelve még mennyi bizonytalanság marad $\underline{\xi}$ vonatkozásában (illetve fordítva).

Az entrópia fontos kérdések tisztázására használható a kriptológiában. Ezzel foglalkozunk a következő fejezetben.

3. A rejtjelezés információelméleti alapjai

Most nézzük meg, hogyan alkalmazhatóak az előbbi eredmények a kriptográfiában.

A továbbiakban $S = (\mathcal{M}, \mathcal{C}, \mathcal{K}, E, D)$ egy kriptorendszer, ahol $\mathcal{M}$ a nyílt, $\mathcal{C}$ a rejtjelezett szövegek és $\mathcal{K}$ a kulcsok halmaza, E a titkosító és D a fejtő algoritmus, továbbá A_M a nyílt és A_C a titkosított szövegekhez alkalmazott ábécé, és $\mathcal{M}^{(n)}$ illetve $\mathcal{C}^{(n)}$ az n-betűs nyílt és titkosított szövegek halmaza. Pr_M egy eloszlás az $\mathcal{M}$ halmazon egy adott nyelv mellett, Pr_K egy eloszlás a $\mathcal{K}$ halmazon, míg $Pr_{M\times K}$ az indukált eloszlás $\mathcal{M} \times \mathcal{K}$-n: $Pr(m,k) = Pr_{M\times K}(m,k) = Pr_M(m)Pr_K(k)$ (mert m és k függetlenek). Minden $m \in \mathcal{M}$-re m ismerete előtt új $k \in \mathcal{K}$ kulcsot választunk, így k választása független m-től.

Adott $m \in \mathcal{M}$ és $k \in \mathcal{K}$ egyértelműen meghatároz egy és csak egy $E_k(m) = c \in \mathcal{C}$-t, és (az injektivitás miatt) hasonlóan, adott $c \in \mathcal{C}$-hez és $k \in \mathcal{K}$-hoz van egy és csak egy olyan $m \in \mathcal{M}$, amellyel $D_k(c) = m \in \mathcal{M}$. Ebből következik, hogy $H(\mathcal{M}|\mathcal{K},\mathcal{C}) = 0 = H(\mathcal{C}|\mathcal{K},\mathcal{M})$.

3.1. Definíció

$H(\mathcal{M}|\mathcal{C})$ és $H(\mathcal{K}|\mathcal{C})$ az S **üzenet-ekvivokáció**ja illetve **kulcs-ekvivokáció**ja.

Δ

A kulcs-ekvivokáció azt méri, mennyi információ nyerhető a kulcsról a rejtjel ismeretében. Értéke – mint diszkrét rendszerekben minden entrópiáé – nem negatív, és pontosan akkor 0, ha a rejtjelezett szöveg 1 valószínűséggel egyértelműen meghatározza a kulcsot, vagyis ha 1 valószínűséggel pontosan egy olyan kulcs van, amellyel egy értelmes nyílt szövegből az adott rejtjelezett szöveget kapjuk. Az üzenet-ekvivokáció értelmezése hasonló.

3.2. Tétel

$H(\mathcal{K}|\mathcal{C}) = H(\mathcal{M}|\mathcal{C}) + H(\mathcal{K}|\mathcal{M},\mathcal{C})$.

Δ

Bizonyítás:
$H(\mathcal{K}|\mathcal{C}) = H(\mathcal{M},\mathcal{K}|\mathcal{C}) - H(\mathcal{M}|\mathcal{K},\mathcal{C}) = H(\mathcal{M},\mathcal{K}|\mathcal{C}) = H(\mathcal{M}|\mathcal{C}) + H(\mathcal{K}|\mathcal{M},\mathcal{C})$.

□

Mivel diszkrét rendszerekben minden entrópia nem negatív, ezért igaz az alábbi.

3.3. Következmény

$H(\mathcal{K}|\mathcal{C}) \geq H(\mathcal{M}|\mathcal{C})$.

Δ

A fenti következmény szerint a kulcs-ekvivokáció legalább akkora, mint az üzenet-ekvivokáció.

3.4. Tétel

$H(\mathcal{K}|\mathcal{C}) = H(\mathcal{K}) + H(\mathcal{M}) - H(\mathcal{C})$.

Δ

Bizonyítás:
$H(\mathcal{C},\mathcal{K},\mathcal{M}) = H(\mathcal{K},\mathcal{M}) + H(\mathcal{C}|\mathcal{K},\mathcal{M}) = H(\mathcal{K},\mathcal{M}) + 0 = H(\mathcal{K},\mathcal{M})$. $\mathcal{K}$ és $\mathcal{M}$ független, így $H(\mathcal{K},\mathcal{M}) = H(\mathcal{K}) + H(\mathcal{M})$, és ekkor egyúttal nyerjük a $H(\mathcal{C},\mathcal{K},\mathcal{M}) = H(\mathcal{K}) + H(\mathcal{M})$ egyenlőséget. Az előbbihez hasonlóan, most felhasználva, hogy $H(\mathcal{M}|\mathcal{K},\mathcal{C}) = 0$, $H(\mathcal{C},\mathcal{K},\mathcal{M}) = H(\mathcal{K},\mathcal{C})$. Mindent egybevetve kapjuk, hogy

$$\begin{aligned} H(\mathcal{K}|\mathcal{C}) &= H(\mathcal{K},\mathcal{C}) - H(\mathcal{C}) = H(\mathcal{C},\mathcal{K},\mathcal{M}) - H(\mathcal{C}) \\ &= H(\mathcal{K}) + H(\mathcal{M}) - H(\mathcal{C}). \end{aligned}$$

□

Amennyiben $H(\mathcal{M}) = H(\mathcal{C})$, akkor a $H(\mathcal{K}|\mathcal{C})$ kulcs-ekvivokáció megegyezik a kulcs *a priori* bizonytalanságával, $H(\mathcal{K})$-val. Mint később majd látjuk, ez a helyzet a tökéletes rendszereknél.

Az üzenet-ekvivokációra vonatkozó kifejezés hasonló:

$$H(\mathcal{C}) + H(\mathcal{M}|\mathcal{C}) = H(\mathcal{M},\mathcal{C}) = H(\mathcal{M}) + H(\mathcal{C}|\mathcal{M}),$$

és ebből

$$H(\mathcal{M}|\mathcal{C}) = H(\mathcal{M}) + H(\mathcal{C}|\mathcal{M}) - H(\mathcal{C}).$$

Most definiáljuk egy nyelv entrópiáját és redundanciáját.

3.5. Definíció

Legyen $\mathcal{L}$ egy természetes nyelv az A ábécével. Ekkor $H^{(\mathcal{L})} = \lim_{n\to\infty} \frac{H(A^n)}{n}$ az $\mathcal{L}$ **betűnkénti entrópiája**, és $R^{(\mathcal{L})} = 1 - \frac{H^{(\mathcal{L})}}{\log|A|}$ az $\mathcal{L}$ **redundanciája**, feltéve, hogy a határérték létezik.

Δ

$\frac{1}{n}H(X_1,\cdots,X_n) = \frac{1}{n}H(X_1) + \frac{1}{n}\sum_{i=2}^{n} H(X_i|X_{i-1},\cdots,X_1)$, és amennyiben a forrás stacionárius, akkor $H(X_i|X_{i-1},\cdots,X_1) = H(X_{i+1}|X_i,\cdots,X_2) \geq H(X_{i+1}|X_i,\cdots,X_2,X_1)$, vagyis a $H(X_i|X_{i-1},\cdots,X_1)$ sorozat monoton csökkenő. Az entrópia nem negatív, így a sorozat alulról korlátos is, tehát konvergens. Innen az $\frac{1}{n}H(X_1) + \frac{1}{n}\sum_{i=2}^{n} H(X_i|X_{i-1},\cdots,X_1)$ sorozat is konvergens, tehát stacionárius forrás esetén létezik $H^{(\mathcal{L})}$. A monoton csökkenésből az is adódik, hogy $nH^{(\mathcal{L})} \leq H(A^n) \leq nH(A)$.

Most tegyük fel, hogy a nyílt szövegeket ugyanazzal a kulccsal titkosítjuk, és tegyük még fel, hogy $|A_M| = |A_C|$. Elegendően nagy n-re $H(\mathcal{M}^n) \approx nH^{(\mathcal{L})} = n\left(1 - R^{(\mathcal{L})}\right)\log|A_M|$, és jó rejtjelező rendszernél a rejtjelszövegek eloszlása megközelítőleg egyenletes, vagyis $H(\mathcal{C}^n) \approx n\log|A_C|$, így

$$\begin{aligned} H(\mathcal{K}|\mathcal{C}^n) &= H(\mathcal{K}) + H(\mathcal{M}^n) - H(\mathcal{C}^n) \\ &\approx H(\mathcal{K}) + n\left(1 - R^{(\mathcal{L})}\right)\log|A_M| - n\log|A_C| \\ &= H(\mathcal{K}) - nR^{(\mathcal{L})}\log|A_M|. \end{aligned}$$

A fenti kifejezésből kapjuk, hogy ha

$$n \geq n_0 = \frac{H(\mathcal{K})}{R^{(\mathcal{L})}\log|A_M|},$$

akkor $H(\mathcal{K}|\mathcal{C}^n) \approx 0$, vagyis a legalább n_0 hosszúságú rejtjelezett szöveg 1-hez közeli valószínűséggel meghatározza az alkalmazott kulcsot. Ez azt jelenti, hogy ilyen hosszúságú rejtjelezett szöveg esetén lényegében véve már semmi bizonytalanságunk sem marad a kulcsra vonatkozóan, vagyis a kulcs

már egyértelmű. A fent meghatározott n_0 tehát a rejtjelezett szöveg azon várható hossza, amelynél csak egyetlen kulccsal visszafejtve kapunk értelmes szöveget. Ezt az értéket **egyértelműségi távolság**nak nevezzük. Láthatóan annál nagyobb ez az érték, minél kisebb a titkosítandó szövegek redundanciája, ami például tömörítéssel csökkenthető. Ha a redundancia 0, akkor az egyértelműségi távolság végtelen, vagyis véges hosszúságú szövegből nem lehet megállapítani a kulcsot.

Ha a kulcsok azonos valószínűséggel fordulnak elő, akkor az előbbi hossz

$$n_0 = \frac{\log|\mathcal{K}|}{R^{(\mathcal{L})}\log|A_M|}.$$

Legyen $|A_M| = r$. Ekkor $n_0 = \frac{\log_r|\mathcal{K}|}{R^{(\mathcal{L})}}$, és $\log_r|\mathcal{K}|$ azt adja meg, hogy a kulcsot milyen hosszon lehet megadni az r-betűs ábécében. Az egyértelműségi távolság innen a kulcs hosszának $\frac{1}{R^{(\mathcal{L})}}$-szerese.

A legjobb titkosító algoritmus az, amely elvileg fejthetetlen, vagyis amely bármekkora számítási kapacitás igénybevételével sem törhető fel. A kérdés az, hogy létezik-e ilyen **feltétel nélkül biztonságos** rendszer. Az alábbiakban ezt vizsgáljuk.

3.6. Tétel

$I(\mathcal{M},\mathcal{C}) \geq H(\mathcal{M}) - H(\mathcal{K})$.

Δ

Bizonyítás:
$H(\mathcal{M}|\mathcal{C}) \leq H(\mathcal{K}|\mathcal{C}) \leq H(\mathcal{K})$, és így $I(\mathcal{M},\mathcal{C}) = H(\mathcal{M}) - H(\mathcal{M}|\mathcal{C}) \geq H(\mathcal{M}) - H(\mathcal{K})$.

□

3.7. Definíció (Tökéletes titkosság)

S pontosan akkor garantál **tökéletes titkosság**ot, ha $\forall(m \in \mathcal{M})\forall(c \in \mathcal{C})$: $Pr(m|c) = Pr(m)$.

Δ

A definíció ekvivalens azzal, hogy $I(\mathcal{M},\mathcal{C}) = 0$, vagyis $H(\mathcal{M}|\mathcal{C}) = H(\mathcal{M})$. Ez pontosan azt jelenti, hogy $\mathcal{M}$ és $\mathcal{C}$ egymástól teljesen függetlenek. Valóban:

$$\begin{aligned} I(\mathcal{M},\mathcal{C}) &= H(\mathcal{M}) - H(\mathcal{M}|\mathcal{C}) \\ &= -\sum_m Pr(m)\log Pr(m) + \sum_m\sum_c Pr(m,c)\log Pr(m|c) \\ &= -\sum_m\sum_c Pr(m,c)\log\frac{Pr(m)}{Pr(m|c)} = -\sum_m\sum_c Pr(m,c)\log\frac{Pr(m)Pr(c)}{Pr(m,c)} \\ &\geq -\sum_m\sum_c \log Pr(m)Pr(c) \geq 0\,, \end{aligned}$$

és egyenlőség pontosan akkor lesz, ha $\frac{Pr(m)}{Pr(m|c)}$ állandó, vagyis valamilyen t-vel $Pr(m) = tPr(m|c)$. De $1 = \sum_m Pr(m) = t\sum_m Pr(m|c) = t$, így $t = 1$, tehát $Pr(m) = Pr(m|c)$, vagyis $I(\mathcal{M},\mathcal{C}) = 0$ akkor és csak akkor, ha minden $m \in \mathcal{M}$-re és $c \in \mathcal{C}$-re $Pr(m) = Pr(m|c)$, azaz pontosan akkor, ha $\mathcal{M}$ és $\mathcal{C}$ egymástól teljesen függetlenek.

A fenti definíció szerint a rendszer akkor és csak akkor tökéletes titkosságú, ha egy támadó semmit nem tud meg m-ről c ismeretében, vagyis c elfogása után pontosan annyi ismerete van m-ről, mint korábban volt.

3.8. Tétel

Legyen $|\mathcal{C}| = |\mathcal{K}|$ és $\forall(m \in \mathcal{M}): Pr(m) > 0$. Ekkor S pontosan akkor garantál tökéletes titkosságot, ha

1. $\forall(m \in \mathcal{M})\forall(c \in \mathcal{C})\exists!\,(k \in \mathcal{K}): E_k(m) = c$;
2. Pr_K egyenletes.

Δ

Bizonyítás:

a) Először tegyük fel, hogy S garantálja a tökéletes titkosságot. Legyen $m \in \mathcal{M}$ rögzített, és tegyük fel, hogy egy $c \in \mathcal{C}$-hez nincs olyan $k \in \mathcal{K}$, hogy $E_k(m) = c$. Ekkor $Pr(m) \neq 0 = Pr(m|c)$, vagyis S nem garantálja a tökéletes titkosságot, ami a feltevésünkkel ellentétes, így, mivel $|\mathcal{C}| = |\mathcal{K}|$, pontosan egy olyan $k \in \mathcal{K}$ van, amellyel $E_k(m) = c$, vagyis teljesül 1.

Most rögzítsünk egy $c \in \mathcal{C}$-t, és legyen $m \in \mathcal{M}$-re $k(m)$ az az egyetlen $k \in \mathcal{K}$, amellyel $E_k(m) = c$. Bayes tétele szerint $Pr(m|c) = \frac{Pr(c|m)Pr(m)}{Pr(c)} = \frac{Pr(k(m))Pr(m)}{Pr(c)}$ minden $m \in \mathcal{M}$-re. Mivel S garantálja a tökéletes titkosságot, ezért $Pr(m|c) = Pr(m)$, és így, az előbbi egyenlőség alapján, $Pr\big(k(m)\big) = Pr(c)$, és a jobb oldal független m-től, tehát $Pr(k)$ minden $k \in \mathcal{K}$-ra azonos, ezért $Pr(k) = \frac{1}{|\mathcal{K}|}$, Pr_K egyenletes.

b) Visszafelé, legyen $k = k(m, c)$ az egyetlen kulcs, amellyel $E_k(m) = c$. Ekkor

$$Pr(m|c) = \frac{Pr(m)Pr(c|m)}{Pr(c)} = \frac{Pr(m)Pr\big(k(m,c)\big)}{\sum_{q \in M} Pr(q)Pr\big(k(q,c)\big)}.$$

Mivel Pr_K egyenletes, ezért $Pr\big(k(m,c)\big) = \frac{1}{|\mathcal{K}|}$, továbbá

$$\sum_{q \in M} Pr(q)Pr\big(k(q,c)\big) = \frac{\sum_{q \in M} Pr(q)}{|\mathcal{K}|}.$$

Ezeket figyelembe véve $Pr(m|c) = Pr(m)$, és így S garantálja a tökéletes titkosságot.

□

$Pr(m|c) = Pr(m)$-ből következik, hogy *m* és *c* független, és ekkor $Pr(c|m) = Pr(c)$ is igaz.

3.9. Definíció

Legyen $r \in \mathbb{N}^+$, $A = \{u \in \mathbb{N} | u < r\}$ ábécé, $n \in \mathbb{N}^+$, $\mathcal{M} = \mathcal{C} = \mathcal{K} = A^n$, és $m \in \mathcal{M}$, $k \in \mathcal{K}$-ra $E_k(m)_i = c_i = (m_i + k_i) \bmod r$, ahol $n > i \in \mathbb{N}$, és k egyenletes eloszlású, teljesen véletlen, az m-től független sorozat egy eleme. Ekkor E a **véletlen átkulcsolás**, angolul a **one-time pad**, az **OTP**.

Δ

3.10. Tétel

A véletlen átkulcsolás tökéletes titkosító algoritmus.

Δ

Bizonyítás:

$E_k(m)_i = c_i = (m_i + k_i) \bmod r$-ből $k_i = (c_i - m_i) \bmod r$, vagyis minden $m \in \mathcal{M}$ és $c \in \mathcal{C}$ meghatároz egy és csak egy kulcsot, amellyel $E_k(m) = c$. A másik feltétel a definícióból adódik.

□

3.11. Tétel

Tökéletes titkosító algoritmus esetén $H(\mathcal{K}) \geq H(\mathcal{M})$.

Δ

Bizonyítás:

Tökéletes titkosító algoritmus esetén $I(\mathcal{M}, \mathcal{C}) = 0$. De $I(\mathcal{M}, \mathcal{C}) \geq H(\mathcal{M}) - H(\mathcal{K})$, és így azonnal kapjuk az állítást.

□

Mivel tökéletes titkosítás esetén $\mathcal{K}$ egyenletes eloszlású, ezért ekkor $H(\mathcal{M}) \leq \log|\mathcal{K}| = l(\mathcal{K})$, ahol $l(\mathcal{K})$ a **kulcs effektív hossza**, vagyis l azt adja meg, hogy milyen hosszúak, hány jegyből állnak a kulcsok.

A tökéletes titkosítás más szavakkal az alábbi. Azon kulcsok teljes valószínűsége, amelyek m_i-t egy adott c-be transzformálnak azonos azon kulcsok teljes valószínűségével, amelyek m_j-t ugyanebbe a c-be transzformálják. Most a c-k száma azonos kell, hogy legyen az m-ek számával, mivel egy rögzített k-ra E_k egy-egyértelmű megfeleltetést hoz létre $\mathcal{M}$ elemei és $\mathcal{C}$ bizonyos elemei között. Tökéletes titkosítás esetén $Pr(c|m) = Pr(c) \neq 0$ minden ilyen c-re és minden m-re. Ennél fogva van legalább egy olyan kulcs, amely egy tetszőleges m-et bármely ilyen c-be transzformál. De az egy adott m-et különböző c-be transzformáló kulcsoknak különbözőeknek kell lenniük, és ezért a különböző kulcsok száma legalább akkora, mint az m-ek száma. Tökéletes titkosság csak a kulcsok ilyen számával nyerhető.

Tökéletes rendszerek, amelyekben a kriptogrammok, az üzenetek és a kulcsok száma azonos, jellemezhetőek azzal, hogy

- az összes kulcsot tekintve, minden m-nek minden c pontosan egy kulcsnál a képe;
- a kulcsok azonos valószínűségűek.

Legyen egy tökéletes titkosító rendszer, és legyen a lehetséges üzenetek száma n. Tekintsük bármely rögzített k kulcsot. Ekkor k az n különböző $m_0, \cdots, m_{n-1}$ nyílt szöveget a páronként különböző $c_0, \cdots, c_{n-1}$ kriptogrammba transzformálja, így $Pr(c_j) = Pr(c_j|m_j) > 0$ minden $0 \leq j < n$-re. De a rendszer tökéletes, és így bármely $u \neq j$-re $Pr(c_j|m_u) = Pr(c_j) > 0$. Ekkor kell lennie egy másik k' kulcsnak, amelyikre $E_{k'}(m_u) = c_j$. Ennek minden $0 \leq u < n$, $u \neq j$-re teljesülnie kell, és minden ilyen kulcsnak különbözőnek kell lennie, következésképpen legalább n kulcsnak kell lennie.

4. Klasszikus rejtjelezés

A titkosítás legegyszerűbb módja, ha a nyílt szöveg minden egyes betűjét ugyanazon szabály szerint helyettesítjük, különböző betűhöz különböző jelet rendelve. Ez az **egyszerű helyettesítés**, **egyábécés helyettesítés**, **monoalfabetikus helyettesítés**. Legyen a nyílt szöveg ábécéje A, az ábécé betűinek száma $|A| = r(\in \mathbb{N}^+\backslash\{1\})$, és legyen $\varphi' : A \to C$ a helyettesítési szabály, továbbá $\mathrm{Im}(\varphi') = B$. Ha $\varphi : A \to B$ olyan, hogy az A minden a elemére $\varphi(a) = \varphi'(a)$, akkor φ A-nak B-re való bijekciója. Legyen m egy nyílt szöveg, és a hossza – betűkben mérve – n. Ha $a \in A$ ebben a szövegben n_a-szor fordul elő, akkor a **relatív gyakorisága**, vagy másként a **frekvenciája** ebben a szövegben $f_a = \frac{n_a}{n}$. Nyilván az ábécé minden betűjére $0 \leq f_a \leq 1$, és $\sum_{a\in A} f_a = 1$. Mivel a szöveg minden betűjét azonos szabállyal, továbbá egy betűt egy betűvel helyettesítünk, ezért az m-nek megfeleltetett c rejtjelezett szöveg hossza azonos a nyílt szöveg hosszával, és a rejtjelezett szövegben $\varphi(a)$ pontosan ugyanannyiszor fordul elő, mint a a nyílt szövegben, így $\varphi(a)$ relatív gyakorisága c-ben azonos a m-beli relatív gyakoriságával, azaz $f_{\varphi(a)} = f_a$ a nyílt szöveg ábécéjének minden betűjére. Egy adott nyelvben az ábécé egy-egy betűjének relatív gyakorisága a tipikus szövegekben, elegendően hosszú szövegeket tekintve, meglehetős stabilitást mutat, és ez lehetővé teszi a tipikus és elegendően hosszú rejtjelezett szövegek fejtését. Mivel a fejtés nem függ attól, hogy a rejtjelezett szövegeket milyen jelekkel írjuk le, vagyis nem függ magától B-től, csupán az egyes jelek előfordulásának gyakoriságától, ezért nyugodtan használhatjuk a titkosított szövegek megadására is az eredeti A ábécét, és ebben az esetben φ az A egy permutációja. Nyilván az sem érdekes, hogy konkrétan milyen jeleket tartalmaz A, így feltehetjük, hogy $A = \mathbb{N}_r = \{u \in \mathbb{N} | u < r\}$. Ez azért előnyös, mert így „számolni” is tudunk az ábécé elemeivel.

Egy egyszerű helyettesítési szabály, ha $a \in A$-t $\varphi(a) = (a + k) \bmod r$-rel helyettesítjük, ahol k egy r-nél kisebb nem negatív egész szám, vagyis $k \in \mathbb{N}_r$ ($k = 0$ is lehetséges, bár ennek sok értelme nincs, hiszen ekkor valójában nem titkosítunk). Ez a **Cæsar-rejtjel** (Cæsarnál konkrétan $k = 3$-mal). Ekkor a kulcstér mérete $|K| = |\mathbb{N}_r| = r$. A vizsgálatok azt mutatják, hogy a magyar nyelvben vannak olyan egy-, két- és hárombetűs szavak, amelyeket a fenti módon egy k kulccsal titkosítva, egy k-tól különböző k' kulccsal fejtve – az eredetitől nyilván eltérő, de – értelmes magyar szót kapunk, de ilyen legalább négybetűs szó már nem létezik (más nyelvekben is hasonló a helyzet, esetleg nem négy, hanem mondjuk ötbetűs szavakkal). Ez azt jelenti, hogy ha a titkosított szövegünk legalább négybetűs, akkor egymás után kipróbálva az egyes eltolásokat (elegendő a titkosított szöveg első néhány betűjére alkalmazva), legfeljebb 25 kísérlet után (nem tekintve azt az esetet, amikor $k = 0$, hiszen ekkor minden kísérlet nélkül készen vagyunk) értelmes magyar szöveghez jutunk. Ekkor tudhatjuk, hogy megtaláltuk az egyértelműen meghatározott kulcsot, amellyel ezek után a teljes szöveget desifrírozhatjuk, és ez egyben azt is jelenti, hogy feltörtük a rendszert. Az előbbi támadási módot, tehát amikor egymás után minden lehetséges kulcsot kipróbálunk, mígnem megfejtettük a szöveget, **kimerítő keresés**nek (**exhaustive search**) illetve **nyers erőn alapuló támadás**nak (**brute force attack**) mondjuk.

Láthatjuk, hogy ilyen kicsi kulcstér esetén a kimerítő kereséssel gyorsan törhető a rendszer. Most legyen a kulcstér az ábécé összes lehetséges permutációjának halmaza, ekkor a kulcstér mérete $|\mathcal{K}| = r! \approx \sqrt{2\pi r}\left(\frac{r}{e}\right)^r$, és ez az angol ábécé esetén $26! = 403\,291\,461\,126\,605\,635\,584 \cdot 10^6$, vagy a Stirling-formulával számolva a közelítő értékét, $26! \approx \sqrt{2\pi \cdot 26}\left(\frac{26}{e}\right)^{26} = 4{,}020\,009\,949 \cdot 10^{26}$. Ez már akkora érték, hogy a kimerítő keresés még a mai számítógépekkel is csak nehezen jöhet szóba, így arra gondolhatunk, hogy így már biztonságos a rendszer. Sajnos a nagy kulcstér önmagában nem jelent feltétlenül biztonságot. Most a kulcstér entrópiája – feltéve, hogy a kulcsok azonos valószínűséggel kerülnek felhasználásra – $H(\mathcal{K}) = \log r! \approx r(\log r - 1)$, és az angol ábécére bitben mérve ennek az értéke 88,38, illetve a közelítő képlettel – szintén bitben megadva – körülbelül 84,70. Ha most figyelembe vesszük, hogy a Cæsar-rejtjelnél a kulcstér entrópiája $\log_2 26 = 4{,}70$ bit, és ezt négybetűs szöveggel egyértelműen fejtettük, akkor nagyon durván számolva azt mondhatjuk, hogy az összes permutációt lehetséges kulcsnak tekintve, az egyértelmű fejtéshez szükséges hossz a Cæsarnál szükséges 4 betűnek a $\frac{84{,}70}{4{,}70} = 20{,}47$-szorosa, azaz körülbelül 82 betű. A helyzet ennél rosszabb, ugyanis a fejtést nyilván egyszerűsíti, ha figyelembe vesszük az egymás mellett álló betűpárok – **digramm**ok – , a há-

rom betűből álló betűcsoportok – **trigramm**ok – , általában az n**-gramm**ok statisztikáját, a kettős betűk gyakoriságát, a szókezdő és szózáró betűk statisztikáját – ez utóbbiaknál feltéve, hogy a rejtjelezésnél nem hagytuk el a szóközöket – , stb. Ha mindezeket figyelembe vesszük, akkor a felmérések szerint egy nagyjából 26 betűs, szokványos angol szöveg elegendő az egyszerű helyettesítés kulcsának egyértelmű megtalálásához, azaz a rendszer töréséhez. Valóban, a vizsgálatok szerint az angol nyelv betűnkénti entrópiája $1{,}2 \sim 1{,}5$ bit, és innen az angol nyelvre az egyszerű helyettesítés egyértelműségi távolsága a korábban megadott képlet (lásd a 23. oldalon) alapján $\frac{\log_2 26!}{\log_2 26 - 1{,}25} \approx 25{,}61$, és feltehetően más nyelveknél is nagyjából ez az érték adódik.

Nézzük, hogyan lehetne javítani a helyzeten. Hogy mit tekintünk egy szöveg elemi egységeinek, betűknek, az meglehetősen szabad. Tekinthetjük például egy betűnek az eredeti szöveg két-két, vagy három-három, akár n-n egymás melletti betűjéből álló csoportot, vagyis a digrammokat, trigrammokat, n-grammokat (ahol n egy pozitív egész szám), és ezt az ábécét helyettesítjük. Ha az n-grammokat tekintjük, akkor $A' = A^n = \mathbb{N}_{r^n}$, és most a kulcstér mérete $r^n!$ (feltéve, hogy a nyílt és a rejtett szövegek ábécéje azonos méretű; általánosabban lehetséges, hogy a titkosított szövegben az eredeti n-grammokat egy s-betűs B ábécé betűiből álló q-grammokkal helyettesítjük, ahol $|A^n| \leq |B^q|$ és ekkor a lehetséges kulcsok száma $\binom{s^q}{r^n} r^n!$). Az n-grammokhoz többféleképpen rendelhetünk egy-egy betűt az új ábécében. Az egyik lehetőség, hogy ha az n-gramm $u_0 \cdots u_{n-1}$, akkor az ennek megfelelő betű $u = \sum_{i=0}^{n-1} u_i r^i$, míg egy másik hozzárendelésnél $\underline{u}$ egy n-komponensű vektor, amelynek i-edik koordinátája ($n > i \in \mathbb{N}$) u_i. Most a fejtés nem csak azért nehezebb, mert nagyobb a kulcstér, hanem azért is, mert az n-grammok statisztikája egyenletesebb, mint az eredeti ábécé statisztikája.

Egy konkrét kulcs megadása az összes lehetséges n-gramm képének, tehát egy n-grammnak a megadását jelenti. Egy ilyen kulcs alkalmazása csak úgy lehetséges, ha ezt a megfeleltetést tároljuk, és egy konkrét alkalmazásakor az adott szöveg egyes n-grammjaihoz megkeressük az őt helyettesítő n-grammot. E helyett arra gondolhatunk, hogy ha n elegendően nagy, akkor már speciális permutációkkal is boldogulunk, olyanokkal, amelyeknél könnyű a kulcs megjegyzése, és a transzformáció is könnyű, mert nem kell hozzá táblázat, számítással is meghatározható. Ilyen szabály például az **affin rejtjel**. Ekkor az ábécé u betűjéhez $(au + b) \bmod r^n$-et rendeljük, ahol $a \in \mathbb{N}_{r^n}$, $b \in \mathbb{N}_{r^n}$ és $(a, r^n) = 1$. Az ilyen transzformációk száma $\varphi(r^n) \cdot r^n$ az Euler-féle φ-függvénnyel.

A várakozással ellentétben ez a rendszer könnyen fejthető. Legyen $v = (au + b) \bmod r^n$ a rejtjelszöveg egy betűje, és legyen $n > t \in \mathbb{N}$. Ekkor

$$\begin{aligned} v \bmod r^t &= \big((au + b) \bmod r^n\big) \bmod r^t = (au + b) \bmod r^t \\ &= (a \bmod r^t)(u \bmod r^t) + (b \bmod r^t) = a'\left(\sum_{i=0}^{n-1} u_i r^i \bmod r^t\right) + b' \\ &= a'\left(\left(\sum_{i=0}^{t-1} u_i r^i + r^t \sum_{i=0}^{(n-1)-t} u_{i+t} r^i\right) \bmod r^t\right) + b' \\ &= a' \sum_{i=0}^{t-1} u_i r^i + b' = a'u' + b', \end{aligned}$$

ahol $a' \in \mathbb{N}_{r^t}$, $b \in \mathbb{N}_{r^t}$ és $u' = \sum_{i=0}^{t-1} u_i r^i \in \mathbb{N}_{r^t}$. Ha $t = 1$, akkor v-nek ez a maradéka a nyílt szöveg n-grammja első betűjének egy affin transzformációja, és ha a rejtjelezett szöveg minden betűjének vesszük ezt a maradékát, akkor ez ugyanaz, mintha a nyílt szöveg minden n-edik betűjéből álló szöveget az a', b' paramétereknek megfelelő affin leképezéssel transzformáltuk volna. Ez a kivonat az adott nyelv egybetűs statisztikáját követi, és így, ha a nyílt szöveg elegendően hosszú, akkor fejthető. Most ennek ismeretében $t = 2$-vel fejtjük a digrammokat, majd ennek ismeretében a trigrammokat, és így tovább.

A másik esetben, amikor az n-grammokat vektoroknak tekintjük, akkor az affin transzformáció $\underline{v} = \mathbf{A}\underline{u} + \underline{b}$, ahol $\mathbf{A}$ egy $n \times n$-es, $\mathbb{N}_r$ fölött invertálható mátrix, és $\underline{b}$ egy n-méretű vektor, vagyis $\mathbf{A} \in \mathbb{N}_r^{n \times n}$ és $\underline{b} \in \mathbb{N}_r^n$. Amennyiben r prímszám, vagy ha r egy prímhatvány, és a műveleteket a meg-

felelő testben végezzük, akkor a reguláris mátrixok száma $\prod_{i=0}^{n-1}\left(r^n - r^i\right)$, és $\underline{b}$ az $\mathbb{N}_r^n$ bármely eleme lehet, tehát a számuk r^n, így ebben az esetben összesen $r^n \prod_{i=0}^{n-1}\left(r^n - r^i\right)$ affin transzformáció van (amennyiben az ábécé a megadott műveletekkel nem test, akkor ennél kisebb értéket kapunk, mert a mátrix két sora nem csak akkor lehet lineárisan összefüggő, ha egyik a másik többszöröse). De ez a rendszer is, mint minden affin rendszer, ismert nyílt szövegű támadással törhető. Ha ismerünk $n+1$ összetartozó $\left(\underline{m}^{(k)}, \underline{c}^{(k)}\right)$ párt, ahol $n \geq k \in \mathbb{N}$, akkor az $\mathbf{A}$ n^2 és a $\underline{b}$ n darab komponense meghatározható: $v_i^{(n)} - v_i^{(k)} = \sum_{j=0}^{n-1} a_{i,j}\left(u_j^{(n)} - u_j^{(k)}\right) \bmod r$-ből meghatározzuk az $a_{i,j}$ elemeket, és ezek ismeretében például $v_i^{(n)} = \left(\sum_{j=0}^{n-1} a_{i,j} u_j^{(n)} + b_i\right) \bmod r$-ből b_i-t. Ezeknek az egyenleteknek biztosan van megoldásuk, például az eredeti értékek, de nem biztos, hogy a megoldás egyértelmű, mert előfordulhat, hogy egy $ux \equiv v\ (r)$ kongruenciában u és r nem relatív prím. Ekkor az egyértelműséghez esetleg több összetartozó párt kell ismerni. Ha azonban választhatjuk a nyílt szövegeket, akkor a relatív prímség már biztosítható, és ekkor a megoldás egyértelmű lesz.

Más lehetőség, hogy a kulcstér méretét csökkentsük, és egyszerűbb számítással elő tudjuk állítani a rejtjelezett szöveget, a **transzpozíció**. Ennél az n-grammot úgy helyettesítjük, hogy a betűk sorrendjét változtatjuk meg. Most a kulcs az n-elemű halmaz egy permutációja, amely a betűk új sorrendjét adja. A kulcstér mérete ekkor $n!$. Régebben erről is azt gondolták, hogy ha n egy viszonylag nagy szám, akkor olyan sok kulcs van, hogy gyakorlatilag törhetetlen a rendszer. Például $n = 50$ esetén a transzpozíciók száma a Stirling-formulával körülbelül $3{,}036\,344\,619 \cdot 10^{64}$ (a kulcsok pontos száma $50! = 30\,414\,093\,201\,713\,378\,043\,612\,608\,166\,064\,768\,844\,377\,641\,568\,960\,512 \cdot 10^{12}$). Ez ismét téves elképzelés. Az adott nyelvben az ábécé minden a, b párjára ismert az ab digramm $P(b|a)$ valószínűsége, vagyis annak a valószínűsége, hogy egy értelmes szövegben az a betűt a b betű követi. Tekintsünk egy transzpozíciós rejtjelrendszert az n blokkmérettel, és tegyük fel, hogy rendelkezésünkre áll l darab rejtjelezett blokk. Készítsünk egy $n \times n$-es T mátrixot, amelyben minden $n > i \in \mathbb{N}$, $n > j \in \mathbb{N}$ indexre $T_{i,j} = -\frac{1}{l}\sum_{k=0}^{l-1} \log P\left(a_j^{(k)} \middle| a_i^{(k)}\right)$, ahol $a_h^{(k)}$ a k-adik blokkban a h-adik betű. Ha $a_i^{(k)}$ és $a_j^{(k)}$ a nyílt szövegben egymás mellett álltak (ebben a sorrendben), vagyis ha a kulcs az eredetileg a mondjuk t-edik pozíción álló betűt az i-edik, míg a $t+1$-edik helyen álló betűt a j-edik pozícióba mozgatta, akkor az adott i,j-párra a $P\left(a_j^{(k)} \middle| a_i^{(k)}\right)$ valószínűségek eloszlása hasonló lesz (elegendően sokk blokkot figyelve) a nyelvben megfelelő betűpárok statisztikájához, míg ellenkező esetben olyan betűpárok lesznek, amelyek a valódi nyelvben többnyire nem, vagy csak nagyon ritkán állnak egymás mellett, és ekkor ezek a valószínűségek kisebbek. De egynél kisebb számok esetén a kisebb érték logaritmusa abszolút értékben nagyobb, így azt kapjuk, hogy az olyan i,j párokra, amelyekben eredetileg egymás mellett álló betűk vannak, vagyis amely indexek egymást követő két szám permutálásából származnak, a mátrix értéke várhatóan kisebb, mint a többi indexpárhoz tatozó érték. Ebből következően elegendő egy olyan, csupa különböző, n-nél kisebb nem negatív egész számokból álló $i_o, \cdots, i_{n-1}$ sorozatot keresni, amelyre $\sum_{k=0}^{n-2} T_{i_k, i_{k+1}}$ a lehető legkisebb, illetve a legkisebbhez közeli. Ekkor a π permutációra $\pi(i) = k_i$, és ellenőrizhető, hogy ezt a permutációt feltételezve, értelmes szöveget kapunk-e. Néhány próbálkozás után szinte biztosan megkapjuk a valódi permutációt.

Az eddigiek szerint az a próbálkozás, hogy nagy blokkokat veszünk, de az összes lehetséges permutációnak csak egy részét alkalmazzuk, hogy a kulcs könnyen megjegyezhető és a transzformáció könnyen elvégezhető legyen, nem sokat javít a helyzeten. Más irányú próbálkozás lehet, ha változtatgatjuk a helyettesítés szabályát. Tekintsük az ábécé egyszerű helyettesítéseit, legyen minden nem negatív egész i-re k_i egy-egy ilyen egyszerű helyettesítés kulcsa, és legyen a nyílt szöveg i-edik betűjére $c_i = \pi_{k_i}(m_i)$, ahol m_i a nyílt szöveg, c_i a titkosított szöveg i-edik betűje, és π_{k_i} az ábécé k_i kulcshoz tartozó permutációja. Az ilyen titkosítást **többábécés helyettesítés**nek, **polialfabetikus helyettesítés**nek nevezik. Ezt a rendszert több évszázadon keresztül fejthetetlennek tartották, és többnyire a legegyszerűbb formájában alkalmazták, amikor minden alkalmazott helyettesítés egyszerű **eltolás**, azaz Cæsar-rejtjel volt. Ez a **Vigenère-rendszer** (*Blaise de Vigenère*), és igen könnyen végrehajtható az úgynevezett **Vigenère-tábla** segítségével. Ez egy $r \times r$-es tábla, ahol r az ábécé mérete, a tábla sorai a kulcsoknak (amelyek most az ábécé betűi), oszlopai a helyettesítendő betűnek felelnek meg. A táblában az adott sor és oszlop metszéspontjában az oszlop által megadott betűnek a sorhoz tartozó kulcs

által meghatározott képe áll, vagyis $a_{i,j} = \left(a^{(j)} + k^{(i)}\right) \bmod r$, ahol $a^{(j)}$ az ábécének a j-edik oszlopot meghatározó betűje és $k^{(i)}$ az i-edik permutációhoz tartozó kulcs betűjele. Az angol ábécé esetén például a tábla a 4. táblázaton látható (ez egy **Latin-négyzet**, vagyis egy olyan táblázat, ahol minden sorban és minden oszlopban minden betű pontosan egyszer található; valójában elegendő lenne, hogy csak a sorokra – ha a sorok felelnek meg egy-egy kulcshoz tartozó leképezésnek – teljesüljön ez a tulajdonság).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

4. táblázat: Vigenère-tábla

A kulcs általában egy rövid szó, amely periodikusan ismétlődik. A Vigenère-rendszer apró módosításával kapjuk a **Beaufort-rejtjel**t (*Francis Beaufort*). A Vigenère-rendszernél a titkosítási szabály $c = (m + k) \bmod r$, és ennek megfelelően a fejtés algoritmusa $m = (c - k) \bmod r$, míg Beaufort-nál a két eljárás egyforma, nevezetesen $c = (k - m) \bmod r$, és ebből $m = (k - c) \bmod r$. Ehhez hasonló táblát lehet megadni, és most a titkosításnál és a fejtésnél azonos módon keressük ki a megadott két betűhöz a harmadikat.

A fejtés nehézségét az okozza, hogy most a titkosított szöveg statisztikája egészen más, mint a nyílt szövegé, nem igaz, hogy ugyanazok a frekvenciák fordulnak elő a két szövegben, hiszen például két, különböző helyen előforduló *A* betűt többnyire különböző betűbe transzformálunk. A lényeg azonban, hogy többnyire, de nem mindig. Ha ugyanis a kulcs hossza l, és a két *A* betű távolsága ennek a hossznak egész számú többszöröse, akkor ezt a két betűt ugyanazzal a szabállyal transzformáljuk, és

így azonos betűbe mennek át. Ha tehát meg tudjuk határozni a kulcshosszt, akkor minden rögzített $l > p \in \mathbb{N}$-re a nyílt szöveg $p + tl$ indexű pozícióin álló betűket ugyanazon k_p kulccsal transzformáljuk, és így az ezekhez tartozó rejtjelszövegbeli betűkből álló szöveg statisztikája már hasonló a nyelv statisztikájához, amennyiben ez a szövegrész elegendően hosszú, és így fejthető. A feladat tehát a kulcshossz meghatározása. Erre egy porosz tüzértiszt, *Friedrich Wilhelm Kasiski* az 1863-ban megjelent *Die Geheimschriften und die Dechiffrierkunst* című könyvében adott egy módszert (bár valószínű, hogy *Charles Babbage* is ismerte ezt a módszert, csak nem hozta nyilvánosságra). A módszer azt használja ki, hogy ha a rejtjelezett szövegben ugyanaz az elegendően hosszú betűsorozat két különböző helyen előfordul, akkor valószínűleg a nekik megfelelő szövegek az eredeti szövegben is azonosak voltak és azonos kulccsal transzformáltuk őket, így a távolságuk a kulcshossz egész számú többszöröse. Ha több ilyen ismétlődést figyelünk meg, akkor a távolságok legnagyobb közös osztója lesz a kulcshossz egész számú többszöröse, és így feltehetően a kulcshossz valamely kis többszörösét kapjuk, vagyis ha a legnagyobb közös osztó t, akkor ennek osztói lesznek a lehetséges kulcshosszak. Ezek a lehetséges hosszak kipróbálhatóak úgy, hogy ha egy osztó s, akkor a rejtjelszöveget felszabdaljuk s részre, egy-egy részbe a $p + ts$ pozíción álló betűket véve (most $s > p \in \mathbb{N}$), és vizsgálva ennek statisztikáját. Ha a statisztika hasonló az adott nyelv statisztikájához, akkor a már ismert módon fejthetjük, és valószínűleg értelmes szöveget kapunk. Néhány próbálkozás után megtaláljuk a tényleges kulcshosszúságot (feltéve, hogy az eredeti szöveg egy értelmes, szokványos szöveg, és a hossza elegendően nagy). Arra azonban ügyelni kell, hogy ha egy olyan pár távolságát is belevesszük a legnagyobb közös osztó számításába, amelynek a tagjai bár egyformák, de nem azonos nyílt szövegből álltak elő azonos kulcsok felhasználásával, akkor egy ilyen hamis távolság olyan legnagyobb közös osztót eredményezhet, amely nem többszöröse a kulcshossznak (sőt, akár kisebb is lehet annál).

A kulcshossz meghatározására vannak más módszerek is, például egy ilyen a **Friedman-féle koincidencia-index** (*William Frederick Friedman*).

Legyen $N \in \mathbb{N}^+$, $A = \{a_0, \cdots, a_{N-1}\}$ egy N-betűs ábécé, $M \in \mathbb{N}^+$, $M > i \in \mathbb{N}$-re $t_i \in A$ és $t'_i \in A$, és legyen $T = (t_0,\ t_1,\ \cdots, t_{M-1})$ illetve $T' = (t'_0,\ t'_1,\ \cdots, t'_{M-1})$. Ekkor

$$\kappa(T, T') = \frac{\sum_{\mu=0}^{M-1} \delta(t_\mu, t'_\mu)}{M},$$

ahol $\delta(x, y) = \begin{cases} 1, \text{ha } x = y \\ 0, \text{ha } x \neq y \end{cases}$ ($\delta(x, y) = 1$ a **koincidencia** – *események vagy jelenségek látszólagos ok nélküli együtt előfordulása, illetve egybeesése* – , x és y **koincidenciája**, κ a **koincidencia-index**, $I.C.$ – *index of coincidence* – , a két szöveg **kappá**ja; *William F. Friedman*, 1925). Ha a két szöveget azonos kulcshoz tartozó polialfabetikus helyettesítéssel vagy transzpozícióval titkosítjuk, akkor az így kapott szövegek koincidencia-indexe nyilván nem változik. Az is könnyen látható, hogy $\kappa(T, T') \leq 1$, és $\kappa(T, T') = 1$ akkor és csak akkor, ha $T = T'$.

Ha T a Q és T' a Q' sztochasztikus forrásból származik, és az ábécé i-edik betűje az első forrás esetén p_i, míg a másiknál p'_i valószínűséggel fordul elő, akkor a koincidencia-index várható értéke

$$E_{Q,Q'}\big(\kappa(T, T')\big) = \sum_{i=0}^{N-1} (p_i \cdot p'_i).$$

Azonos forrás esetén a várható érték

$$E_Q\big(\kappa(T, T')\big) = \sum_{i=0}^{N-1} p_i^2,$$

és ekkor

$$\frac{1}{N} \leq E_Q\big(\kappa(T, T')\big) \leq 1.$$

A bal oldalt akkor és csak akkor kapjuk, ha a betűk eloszlása egyenletes, vagyis $p_i = \frac{1}{N}$ (ezt a forrást Q_r jelöli, és a hozzá tartozó várható értéket κ_r, ahol r a *random – rendszertelen*, *találomra történő*, *vaktában tett*, sokszor, és ebből gyakran hibásan, *véletlen* – rövidítése), míg a jobb oldalt pontosan akkor, ha az eloszlás determinisztikus, azaz egy $N > j \in \mathbb{N}$-re $p_j = 1$ (és ekkor minden más i-re $p_i = 0$). Természetes nyelvek esetén $E_Q\big(\kappa(T,T')\big)$ a nyelvtől függő állandó, **a**z adott **nyelv empirikus** κ-**ja**, a **nyelvállandó**, és ezt κ_p-vel jelöljük (p a plaintext rövidítése). 26-betűs ábécékre vonatkozó nyelvállandókat mutat az 5. táblázat (különböző statisztikák ettől eltérő értéket adhatnak). Ha a két szöveg azonos nyelven, azonos ábécével írt tipikus szöveg, és a hosszuk elegendően nagy, akkor $\kappa(T,T') \approx \kappa_p$, így például a rejtjel felfedi a mögötte lévő nyelvet.

nyelv	nyá
angol	0,0665
francia	0,0777
magyar	0,0705
német	0,0788
olasz	0,0746
orosz*	0,0677
portugál	0,0746
spanyol	0,0746

* a 32-betűs nyelvre 0,0529

5. táblázat

Nyelvállandók

Most legyen m_i és m_i' az a_i karakter gyakorisága T-ben és T'-ben ($\sum_{i=0}^{N-1} m_i = M = \sum_{i=0}^{N-1} m_i'$), $f_i = \frac{m_i}{M}$ és $f_i' = \frac{m_i'}{M}$ a megfelelő **relatív gyakoriságok** ($\sum_{i=0}^{N-1} f_i = 1 = \sum_{i=0}^{N-1} f_i'$), ekkor

$$\chi(T,T') = \frac{\sum_{i=0}^{N-1}(m_i \cdot m_i')}{M^2} = \frac{\sum_{i=0}^{N-1}(m_i \cdot m_i')}{(\sum_{i=0}^{N-1} m_i)\cdot(\sum_{i=0}^{N-1} m_i')} = \sum_{i=0}^{N-1}(f_i \cdot f_i')$$

a **keresztszorzat-összeg** (*Solomon Kullback*, 1935). $\chi(T,T') \le 1$, és $\chi(T,T') = 1$ akkor és csak akkor, ha a két szövegben együttesen az ábécé egyetlen betűje fordul elő. Ha az egyik szöveg az ábécé minden betűjét azonos gyakorisággal tartalmazza, akkor (a másik szövegtől függetlenül) $\chi(T,T') = \frac{1}{N} = \kappa_r$. Abban a fontos speciális esetben, amikor $T = T'$ (és ekkor $m_i = m_i'$), egy új függvényt kapunk:

$$\Psi(T) = \chi(T,T) = \frac{\sum_{i=0}^{N-1} m_i^2}{M^2} = \sum_{i=0}^{N-1} f_i^2.$$

Mivel $0 \le \frac{\sum_{i=1}^{N}\left(m_i - \frac{M}{N}\right)^2}{M^2} = \frac{\sum_{i=1}^{N} m_i^2}{M^2} - \frac{1}{N} \le 1 - \frac{1}{N}$, ezért

$$\frac{1}{N} \le \Psi(T) \le 1,$$

és $\Psi(T) = \frac{1}{N} = \kappa_r$ akkor és csak akkor, ha minden betű azonos gyakoriságú a szövegben, míg $\Psi(T) = 1$ pontosan akkor, ha a szöveg minden betűje azonos. Ha $M \le N$, vagyis a szöveg egészen rövid, akkor $\frac{1}{M} \le \Psi(T)$, és $\Psi(T) = \frac{1}{M}$ akkor és csak akkor, ha $m_i \in \{0,1\}$. Most az igaz, hogy monoalfabetikus helyettesítés valamint transzpozíció esetén χ valamint Ψ invariáns.

A várható értékek könnyen meghatározhatóak:

$$E_{Q,Q'}\big(\chi(T,T')\big) = \sum_{i=0}^{N-1}(p_i \cdot p_i'),$$

és $Q = Q'$ esetén

$$E_Q\big(\chi(T,T')\big) = \sum_{i=0}^{N-1} p_i^2,$$

ahonnan

$$E_Q\big(\Psi(T)\big) = \sum_{i=0}^{N-1} p_i^2.$$

Azonnal látható, hogy

$$E_{Q,Q'}\big(\chi(T,T')\big) = E_{Q,Q'}\big(\kappa(T,T')\big),$$

tehát

$$E_Q\big(\chi(T,T')\big) = E_Q\big(\kappa(T,T')\big),$$

és ekkor

$$\frac{1}{N} \le E_Q\big(\chi(T,T')\big) \le 1$$

és

$$\frac{1}{N} \le E_Q\big(\Psi(T)\big) \le 1.$$

A bal oldalt egyenletes eloszlásnál és csak ekkor, tehát Q_r esetén kapjuk, míg a másik szélsőértéket a determinisztikus eloszlásoknál, vagyis amikor a szöveg minden betűje azonos, és más esetben nem. κ és χ várható értékének azonosságából adódik, hogy tipikus szövegek esetén a nyelv meghatározható a kriptoszövegből.

κ és χ között további összefüggés állapítható meg. Legyen $g_{i,\mu} = \begin{cases} 1, \text{ha } t_\mu = a_i \\ 0 \text{ egyébként} \end{cases}$, és $g'_{i,\mu}$-t hasonlóan definiáljuk t'_μ-vel (t_μ a T és t'_μ a T' μ-edik karaktere, és a_i az ábécé i-edik eleme). Ekkor

$$\delta\big(t_\mu, t'_\nu\big) = \sum_{i=0}^{N-1} \big(g_{i,\mu} \cdot g'_{i,\nu}\big)$$

és

$$m_i = \sum_{\mu=0}^{M-1} g_{i,\mu},$$
$$m'_i = \sum_{\nu=0}^{M-1} g'_{i,\nu}.$$

Legyen $T^{(s)}$ a T s hellyel való ciklikus jobbraléptetésével kapott szöveg. Ekkor

$$\kappa\big(T^{(s)}, T'\big) = \frac{\sum_{\mu=0}^{M-1} \delta\big(t_{(\mu-s) \bmod M}, t'_\mu\big)}{M},$$

és speciálisan

$$\kappa\big(T^{(0)}, T'\big) = \kappa(T,T').$$

Nézzük az eltoltakkal számolt κ-k átlagát.

$$\frac{1}{M}\sum_{\rho=0}^{M-1}\kappa\left(T^{(\rho)},T'\right)=\frac{1}{M}\cdot\frac{1}{M}\cdot\sum_{\rho=0}^{M-1}\sum_{\mu=0}^{M-1}\delta\left(t_{(\mu-\rho)\bmod M},t'_{\mu}\right)=\frac{1}{M}\cdot\frac{1}{M}\cdot\sum_{\nu=0}^{M-1}\sum_{\mu=0}^{M-1}\delta\left(t_{\mu},t'_{\nu}\right)$$
$$=\frac{1}{M}\cdot\frac{1}{M}\cdot\sum_{\nu=0}^{M-1}\sum_{\mu=0}^{M-1}\sum_{i=0}^{N-1}g_{i,\mu}\cdot g'_{i,\nu}=\frac{1}{M}\cdot\frac{1}{M}\cdot\sum_{i=0}^{N-1}\sum_{\nu=0}^{M-1}\sum_{\mu=0}^{M-1}g_{i,\mu}\cdot g'_{i,\nu}$$
$$=\frac{1}{M}\cdot\frac{1}{M}\cdot\sum_{i=0}^{N-1}\left(\sum_{\nu=0}^{M-1}g'_{i,\nu}\cdot\sum_{\mu=0}^{M-1}g_{i,\mu}\right)=\frac{1}{M}\cdot\frac{1}{M}\cdot\sum_{i=0}^{N-1}(m'_i\cdot m_i)=\chi(T,T'),$$

vagyis

$$\frac{1}{M}\sum_{\rho=0}^{M-1}\kappa\left(T^{(\rho)},T'\right)=\chi(T,T')$$

(ez a κ-χ-**tétel**). Következményként kapjuk, hogy

$$\frac{1}{M}\sum_{\rho=0}^{M-1}\kappa\left(T^{(\rho)},T\right)=\Psi(T).$$

$\kappa\left(T^{(0)},T\right)=1$, míg $s\neq 0$ esetén $\kappa\left(T^{(s)},T\right)$ ennél lényegesen kisebb, így az átlagolásnál $s=0$ nem tipikus, ezért logikusabb az átlagolást csupán a többi értékre végezni. Legyen

$$\Phi(T)=\frac{\sum_{i=0}^{N-1}\binom{m_i}{2}}{\binom{M}{2}}=\frac{\sum_{i=0}^{N-1}\left(m_i\cdot(m_i-1)\right)}{M\cdot(M-1)}$$
$$=\frac{1}{1-\frac{1}{M}}\sum_{i=0}^{N-1}\left(f_i\cdot\left(f_i-\frac{1}{M}\right)\right)=\frac{1}{1-\frac{1}{M}}\left(\sum_{i=0}^{N-1}f_i^2-\frac{1}{M}\right).$$

Ekkor definiálunk egy $\Phi(T)$ függvényt:

$$\Phi(T)=\frac{1}{1-\frac{1}{M}}\left(\sum_{i=0}^{N-1}f_i^2-\frac{1}{M}\right)=\frac{1}{1-\frac{1}{M}}\left(\Psi(T)-\frac{1}{M}\right),$$

és ezzel

$$\frac{1}{M-1}\sum_{\rho=1}^{M-1}\kappa\left(T^{(\rho)},T\right)=\frac{1}{M-1}\left(\sum_{\rho=0}^{M-1}\kappa\left(T^{(\rho)},T\right)-1\right)$$
$$=\frac{1}{M-1}\cdot(M\cdot\Psi(T)-1)=\Phi(T)$$

(κ-Φ-**tétel**). $\Phi(T)=0$ akkor és csak akkor, ha minden i-re $m_i\in\{0,1\}$. Mivel $M\cdot\Psi(T)=(M-1)\cdot\Phi(T)+1$, és ebből $\Psi(T)-\Phi(T)=\frac{1-\Phi(T)}{M}=\frac{1-\Psi(T)}{M-1}$, továbbá $\frac{1}{N}\leq\Psi(T)\leq 1$, ezért

$$\frac{M-N}{M-1}\cdot\frac{1}{N}\leq\Phi(T)\leq\Psi(T),$$

ahol a bal oldalon az egyenlőség ekvivalens azzal, hogy valamennyi m_i azonos.

Mivel Φ és Ψ csaknem azonos, így nem meglepő, hogy invariancia szempontjából Φ-re hasonló állítás érvényes, mint Ψ-re.

Φ várható értéke függ M-től:

$$E_Q^{(M)}\big(\Phi(T)\big) = \frac{M}{M-1} \cdot \left(\sum_{i=0}^{N-1} p_i \cdot \left(p_i - \frac{1}{M} \right) \right),$$

és

$$E_Q^{(M)}\big(\Phi(T)\big) \geq \begin{cases} \frac{M}{M-1} \cdot \left(\frac{1}{N} - \frac{1}{M} \right) = \frac{1}{N} \cdot \frac{M-N}{M-1}, & \text{ha } M \geq N \\ 0, & \text{ha } M \leq N \end{cases}.$$

A fenti egyenlőtlenség akkor és csak akkor egyenlőség, ha valamennyi m_i értéke azonos, és aszimptotikusan

$$E_Q^{(\infty)}\big(\Phi(T)\big) = \lim_{M \to \infty} E_Q^{(M)} \Phi(T) = \sum_{i=0}^{N-1} p_i^2 = E_Q\big(\Psi(T)\big).$$

Legyen P a Q forrás egy M-hosszúságú nyílt szövege, $P^{(s)}$ ennek s pozícióval való ciklikus jobbratoltja, p_i a Q forrásnál az i-edik betű előfordulásának valószínűsége, d egy periodikus többábécés helyettesítésnél a kulcs hossza (feltesszük, hogy $d|M$), C a P-hez tartozó rejtjel, végül $C^{(k \cdot d)}$ a C $k \cdot d$ pozícióval való ciklikus eltoltja. Ekkor a $P^{(k \cdot d)}$-hez tartozó rejtjel egybeesik $C^{(k \cdot d)}$-vel, így minden k-ra

$$E_Q\left(\kappa\big(C^{(k \cdot d)}, C\big)\right) = E_Q\left(\kappa\big(P^{(k \cdot d)}, P\big)\right) = \sum_{i=0}^{N-1} p_i^2 = \kappa_p.$$

Másrészről, ha d nem osztója u-nak, akkor $E_Q\left(\kappa\big(C^{(u)}, C\big)\right)$-re ilyen kijelentést nem lehet tenni. Ebben az esetben $C^{(u)}$ egy Q' forrásból származó szöveg, és Q, Q' függetlenek, és ekkor

$$E_{Q',Q}\left(\kappa\big(C^{(u)}, C\big)\right) = \sum_{i=0}^{N-1} p_i' p_i,$$

amely érték $\frac{1}{N} = \kappa_r$ körül ingadozik.

Ha d nem osztója M-nek, akkor a $k \cdot d$-vel való ciklikus eltolásnál a szöveg utolsó $k \cdot d$ hosszúságú szakasza nem egy periódus első karakterénél kezdődik, ugyanakkor a szöveg első $k \cdot d$ karakteréből álló szövegrésszel kerül összehasonlításra, így az összehasonlítandó karakterek nem azonos kulcshoz tartoznak, míg más mértékű eltolásnál esetleg éppen különböző kulccsal titkosított szövegrészek kerülnek olyan viszonyba, mintha azonos kulccsal keletkeztek volna. Ezt a problémát úgy lehet kiküszöbölni, ha az eltolásnál túlcsorduló részt elhagyjuk, nem ciklikusan tolunk el, vagyis az u-val való eltoláskor, ahol $M > u \in \mathbb{N}$, a $T^{\langle u \rangle} = (t_0, \cdots, t_{M-1-u})$ és a ${}^{\langle u \rangle}T = (t_u, \cdots, t_{M-1})$ szövegeket hasonlítjuk össze, ennek a két $M-u$-hosszúságú szövegnek határozzuk meg a koincidencia-indexét.

Legyen $d > \rho \in \mathbb{N}$-re $C_\rho = \left(t_\rho, t_{\rho+d}, \cdots, t_{\rho+\left(\frac{M}{d}-1\right) \cdot d} \right)$. Ha valóban d a kulcshossz, akkor minden ρ-ra C_ρ monoalfabetikus helyettesítéssel keletkezett, és így valamennyi $\Phi\big(C_\rho\big)$ várhatóan közel van a $\kappa_r = \frac{1}{N}$-nél lényegesen nagyobb nyelvállandóhoz, míg ellenkező esetben ingadozást mutatnak. Ez a **Kullback-féle Φ-teszt**. Ha

$$\Phi^{(d)}(C) = \frac{1}{d} \cdot \sum_{\rho=0}^{d-1} \Phi\big(C_\rho\big),$$

akkor

$$\Phi^{(d)}(C) = \frac{1}{d} \cdot \sum_{\rho=0}^{d-1} \Phi\big(C_\rho\big) = \frac{\frac{1}{d} \cdot \sum_{\rho=0}^{d-1} \sum_{i=0}^{N-1} m_i^{(\rho)} \cdot \left(m_i^{(\rho)} - 1 \right)}{\frac{M}{d} \cdot \left(\frac{M}{d} - 1 \right)},$$

ahol $m_i^{(\rho)}$ az i-edik karakter frekvenciája C_ρ-ban, és

$$\Phi^{(d)}(C) = \frac{d \cdot \sum_{\rho=0}^{d-1} \Phi_\rho}{M \cdot (M-d)}.$$

Itt

$$\Phi_\rho = \sum_{i=0}^{N-1} m_i^{(\rho)} \cdot \left(m_i^{(\rho)} - 1\right).$$

A korábbihoz hasonlóan kapjuk, hogy

$$\frac{1}{M-1} \sum_{\rho=1}^{M-1} \kappa\left(T^{(u\cdot\rho)}, T\right) = \Phi^{(u)}(T)$$

(κ-$\Phi^{(u)}$ -**tétel**).

A κ-Φ tételből

$$E_Q^{(M)}\left(\Phi(T)\right) = \frac{1}{M-1} \sum_{\rho=1}^{M-1} E_Q\left(\kappa\left(T^{(\rho)}, T\right)\right),$$

és ebből $E_Q\left(\kappa\left(T^{(k\cdot d)}, T\right)\right) = \kappa_p$, míg ha u nem többszöröse d-nek, akkor $E_Q\left(\kappa\left(T^{(u)}, T\right)\right) \approx \kappa_r = \frac{1}{N}$. Feltéve még, hogy M többszöröse d-nek, akkor $\sum_{\rho=1}^{M-1}\langle\kappa\left(T^{(\rho)}, T\right)\rangle_Q$-ban κ_p $\frac{M}{d} - 1$-szer, κ_r pedig $M - \frac{M}{d}$-szer fordul elő, így $E_Q^{(M)}\left(\Phi(T)\right)$ a κ_p és κ_r egy közepe:

$$(M-1) \cdot E_Q^{(M)}\left(\Phi(T)\right) \approx \left(\frac{M}{d} - 1\right) \cdot \kappa_p + \left((M-1) - \left(\frac{M}{d} - 1\right)\right) \cdot \kappa_r.$$

Feltéve, hogy a megfigyelt $\Phi(T)$ approximálja a várható értéket, kapjuk, hogy

$$(M-1) \cdot \Phi(T) \approx \left(\frac{M}{d} - 1\right) \cdot \kappa_p + \left((M-1) - \left(\frac{M}{d} - 1\right)\right) \cdot \kappa_r$$

(*Abraham Sinkov*, 1935 körül). Ha M nagy és $d \ll M$, akkor

$$\Phi(T) \approx \frac{1}{d} \cdot \kappa_p + \left(1 - \frac{1}{d}\right) \cdot \kappa_r.$$

Sinkov relációja megoldható d-re:

$$\left(\frac{M}{d} - 1\right) \approx \frac{(M-1) \cdot (\Phi(T) - \kappa_r)}{\kappa_p - \kappa_r},$$

azaz

$$d \approx \frac{\kappa_p - \kappa_r}{\frac{1}{M} \cdot \left(\kappa_p - \Phi(T)\right) + (\Phi(T) - \kappa_r)}.$$

Ismét, ha M nagy és $d \ll M$, akkor

$$d \approx \frac{\kappa_p - \kappa_r}{\Phi(T) - \kappa_r}.$$

A helyzetet bonyolítja, hogy a kulcsszóban azonos betű többször is előfordulhat, és ekkor nem csupán a kulcshossz egész számú többszörösének megfelelő távolságban lesznek azonos kulccsal titkosított betűk, továbbá, ha a kulcsszó egy élő nyelv valamely szava, akkor a kulcsokat nem egyenletes valószínűséggel választjuk a teljes kulcstérből. Ezeket is figyelembe véve, a következő módon számolhatunk. Legyen $\mathcal{K}$ (a kulcstér) A (az ábécé) permutációinak egy halmaza, a kulcsok száma λ, K valószínűségi változó $\mathcal{K}$-n, $P_k(K=k)=q(k)$, továbbá k' a k titkosító kulcshoz tartozó fejtőkulcs. Legyen $M^{(1)}=\left(M_1^{(1)},\cdots,M_n^{(1)}\right)$ és $M^{(2)}=\left(M_1^{(2)},\cdots,M_n^{(2)}\right)$ A fölötti független valószínűségi változók olyan sorozata, hogy minden $n>j\in\mathbb{N}$-re, $i\in\{1,2\}$-re és $m\in A$-ra $P_p\left(M_j^{(i)}=m\right)=p(m)$, míg $C^{(i)}=\left(C_1^{(i)},\cdots,C_n^{(i)}\right)$ olyan sorozat, amelyet $M^{(i)}$-ből egyszerű helyettesítéssel kapunk a $k^{(i)}$ kulcs alkalmazásával. Annak a valószínűsége, hogy a rejtjelben egy adott pozíción c áll, megegyezik ősének a valószínűségével, ami függ az alkalmazott kucstól, így

$$P_c\left(C_j^{(i)}=c\right)=\sum_{k\in\mathcal{K}}q(k)\,p\left(k'(c)\right).$$

A későbbiek kedvéért ezt másként is felírjuk:

$$P_c\left(C_j^{(i)}=c\right)=\sum_{k\in\mathcal{K}}q(k)\,p\left(k'(c)\right)=\sum_{m\in A}\sum_{\substack{k\in\mathcal{K}\\k'(c)=m}}q(k)p\left(k'(c)\right)$$

$$=\sum_{m\in A}\sum_{\substack{k\in\mathcal{K}\\k'(c)=m}}q(k)p(m)=\sum_{m\in A}\left(p(m)\sum_{\substack{k\in\mathcal{K}\\k'(c)=m}}q(k)\right).$$

Ha K egyenletes eloszlású, vagyis $q(k)$ minden kulcsra azonos (tehát $q(k)=\frac{1}{\lambda}$), akkor

$$P_c\left(C_j^{(i)}=c\right)=\sum_{m\in A}\left(p(m)\sum_{\substack{k\in\mathcal{K}\\k'(c)=m}}q(k)\right)=\sum_{m\in A}\left(p(m)\sum_{\substack{k\in\mathcal{K}\\k'(c)=m}}\frac{1}{\lambda}\right)$$

$$=\frac{1}{\lambda}\sum_{m\in A}\left(p(m)\sum_{\substack{k\in\mathcal{K}\\k'(c)=m}}1\right)=\frac{1}{\lambda}\sum_{m\in A}\left(p(m)s(c,m)\right),$$

ahol $s(c,m)=|\{k\in\mathcal{K}|k'(c)=m\}|$, vagyis $s(c,m)$ azon kulcsok száma, amelyek a nyílt szövegbeli m betűt c-be transzformálják. Ha még az is teljesül, hogy $s(c,m)$ minden c-re mint m függvénye konstans (és ekkor mint c és m függvénye is konstans), vagyis $s(c,m)=\frac{\lambda}{N}$, akkor

$$P_c\left(C_j^{(i)}=c\right)=\frac{1}{\lambda}\sum_{m\in A}p(m)s(c,m)=\frac{1}{\lambda}\sum_{m\in A}\frac{\lambda}{N}p(m)=\frac{1}{N}\sum_{m\in A}p(m)=\frac{1}{N}.$$

$s(c,m)$ konstans például akkor, amikor $\mathcal{K}$ az ábécé valamennyi permutációját tartalmazza, ekkor $\lambda=N!$ és $s(c,m)=(N-1)!$ (azon permutációk száma, amelyek a rögzített c-be viszik a rögzített m-et), és $s(c,m)$ konstans a Vigenère-féle titkosításnál is, amikor $\lambda=N$ és $s(c,m)=1$ minden (c,m)-párra.

Szeretnénk most is meghatározni, hogy hány pozíción egyezik meg $M^{(1)}$ és $M^{(2)}$, vagyis hány helyen van koincidencia a két nyílt szövegben, azaz $\kappa\left(M^{(1)},\,M^{(2)}\right)$-t.

Egy adott pozíción a koincidencia valószínűsége

$$P_m\left(M_j^{(1)}=M_j^{(2)}\right)=\sum_{m\in A}P_m\left(M_j^{(1)}=m=M_j^{(2)}\right)=\sum_{m\in A}\left(p(m)\right)^2.$$

Most nézzük a megfelelő rejtjelszövegeket, és nézzük ebben az esetben is a koincidenciákat. Két esetet tekintünk: a két szöveget azonos kulccsal illetve különböző kulccsal titkosították. Az első esetben

$$P_c\left(C_j^{(1)} = C_j^{(2)}\right) = \sum_{m\in A} P_c\left(C_j^{(1)} = c = C_j^{(2)}\right) = \sum_{k\in\mathcal{K}} q(k) \sum_{c\in A} \left(p\left(k'(c)\right)\right)^2$$
$$= \sum_{k\in\mathcal{K}} q(k) \sum_{m\in A} \left(p(m)\right)^2 = \sum_{m\in A} \left(p(m)\right)^2,$$

míg a másik esetben

$$P_c\left(C_j^{(1)} = C_j^{(2)}\right) = \sum_{m\in A} P_c\left(C_j^{(1)} = c = C_j^{(2)}\right)$$
$$= \sum_{k^{(1)}\in\mathcal{K}} \sum_{k^{(2)}\in\mathcal{K}} q\left(k^{(1)}\right) q\left(k^{(2)}\right) \sum_{c\in A} \left(p\left(k^{(1)\prime}(c)\right) p\left(k^{(2)\prime}(c)\right)\right)$$
$$= \sum_{c\in A} \left(\sum_{k^{(1)}\in\mathcal{K}} q\left(k^{(1)}\right) p\left(k^{(1)\prime}(c)\right) \sum_{k^{(2)}\in\mathcal{K}} q\left(k^{(2)}\right) p\left(k^{(2)\prime}(c)\right)\right)$$
$$= \sum_{c\in A} \left(\sum_{k\in\mathcal{K}} q(k) p\left(k'(c)\right)\right)^2 = \sum_{c\in A} \left(P_c(C = c)\right)^2.$$

Látható, hogy azonos kulcs esetén

$$P_c\left(C_j^{(1)} = C_j^{(2)}\right) = P_m\left(M_j^{(1)} = M_j^{(2)}\right) = \sum_{m\in A} \left(p(m)\right)^2 = \kappa_p,$$

ahol κ_p a korábban már bevezetett nyelvállandó (lásd a 32. oldalon), míg különböző kulcs használata esetén

$$P_c\left(C_j^{(1)} = C_j^{(2)}\right) = \sum_{c\in A} \left(P_c\left(C_j^{(i)} = c\right)\right)^2 = \kappa_{r\prime}.$$

Ha K egyenletes eloszlású, és az ábécé minden egyes betűjét ugyanannyi kulcs transzformálja az ábécé egy adott betűjébe, vagyis ha $s(c, m)$ konstans, akkor

$$\kappa_{r\prime} = \sum_{c\in A} \left(P_c\left(C_j^{(i)} = c\right)\right)^2 = \sum_{c\in A} \left(\frac{1}{N}\right)^2 = \frac{1}{N^2} \sum_{c\in A} 1 = \frac{1}{N^2} N = \frac{1}{N} = \kappa_r,$$

vagyis ekkor visszakaptuk a 32. oldalon kapott korábbi értéket a különböző kulcsok esetére.

A koincidencia-indexet gyakran az

$$I.C. = \frac{\kappa(T, T')}{\frac{1}{N}} = \frac{\kappa(T, T')}{\kappa_r}$$

kifejezéssel definiálják (N az ábécé elemeinek száma), továbbá definiálják a

$$\sum_{i=0}^{N-1} \left(\frac{m_i}{M} - \frac{1}{N}\right)^2 = \frac{\sum_{i=0}^{N-1} m_i^2}{M^2} - \frac{1}{N} = \Psi(T) - \frac{1}{N}$$

értéket, ahol M a vizsgált szöveg hossza, N az ábécé mérete, és m_i az ábécé i-edik betűjének gyakorisága az M hosszúságú szövegben. Ennek várható értéke

$$MR = E_Q\left(\Psi(T) - \frac{1}{N}\right) = E_Q\left(\Psi(T)\right) - \frac{1}{N} = \sum_{i=0}^{N-1} p_i^2 - \frac{1}{N} = \kappa_p - \kappa_r,$$

az **egyenetlenség mértéke** (*measure of roughness*, *Sinkov*). Értéke egyenletes eloszlás esetén 0, és annál nagyobb, minél inkább eltér a betűk eloszlása az egyenletestől, minél inkább jellemző néhány betű gyakori előfordulása, míg néhány más betű lényegében véve szinte soha nem bukkan fel tipikus szövegekben.

A többábécés helyettesítés szélső esete, amikor a nyílt szövegeket egy egyenletes eloszlású, egyszer használt valódi véletlen sorozat egyes betűinek megfelelő kulccsal transzformáljuk. Ez a **véletlen átkulcsolás**, a **one-time pad**, röviden **OTP**. Itt az egyes betűk helyettesítéséhez a Cæsar-rejtjelt használjuk. Korábban már foglalkoztunk ezzel a titkosító algoritmussal, és láttuk (3.10 Tétel), hogy ez (és csak ez) tökéletes titkosságot biztosít. Az algoritmus gyenge pontja a szükséges kulcs generálása és kicserélése. A tökéletes biztonsághoz potenciálisan végtelen hosszúságú valódi véletlen sorozatra van szükség, amelyet a résztvevő felek nem tudnak külön-külön előállítani, vagyis mindenképpen szükséges egy megbízható csatornán eljuttatni a résztvevő felekhez, és soha nem lehet elölről kezdeni egy megkezdett sorozatot. Ha ugyanis a kulcs ismétlődik, és ezt a támadó felismeri, akkor kezében van ugyanazon kulccsal titkosított két különböző, összetartozó (m_1, c_1) és (m_2, c_2) nyílt szöveg - rejtett szöveg pár, és ekkor $c_2 - c_1 = (m_2 - m_1) \bmod r$ nem függ a kulcstól, viszont magában hordozza az adott nyelv statisztikáját, ha a szövegek elegendően hosszúak.

Egy próbálkozás a kulcsprobléma megoldására a **futókulcs**, ahol kulcsként mondjuk egy közösen használt könyv meghatározott részét használjuk. Ekkor azonban $c = (m + k) \bmod r$-ben a kulcs is egy élő nyelv statisztikáját követi, így statisztikai alapon a fejtés ismét könnyen végrehajtható.

Vernam 1919-ben szabadalmazott eljárásában a bináris nyílt szöveg egyes bitjeihez XOR-operációval kapcsolta a szintén binárisan megadott kulcs egyes bitjeit. Ekkor a visszafejtés pontosan ugyanúgy történik, mint a titkosítás, és a két eljárás során azonos a kulcs is. Kevéssel később *Mauborgne* javasolta, hogy a kulcs bitjei egy valódi véletlen sorozat legyenek.

Viszonylag rövid kulccsal hosszú, ismétlődés nélküli kulcsot lehet előállítani az **önkulcsolás**sal. Legyen $k_{-l} \cdots k_{-1}$ egy l-hosszúságú szó. Ekkor például az alábbi két módon tudunk ennek segítségével polialfabetikus helyettesítést készíteni:

1. $c_i = \pi_{m_{i-l}}(m_i)$, ahol $l \geq i \in \mathbb{N}$-re $m_{-i} = k_{-i}$;
2. $c_i = \pi_{c_{i-l}}(m_i)$, ahol $l \geq i \in \mathbb{N}$-re $c_{-i} = k_{-i}$.

A fenti mindkét esetben π az ábécé egy permutációja. A fejtés alapja most is a kulcshossz meghatározása és a nyelvi statisztika. Speciális esetben a transzformáció lehet a Cæsar-féle eltolásos titkosítás:

1. $c_i = (m_i + m_{i-l}) \bmod r$;
2. $c_i = (m_i + c_{i-l}) \bmod r$.

Most jól látszik, de az általános esetre is igaz, hogy a második módszernél az i-edik betű képe függ az őt az l többszöröseinek megfelelő távolsággal megelőző betűktől is, hiszen $i \geq l$ esetén

$$\begin{aligned} c_i &= (m_i + c_{i-l}) \bmod r = \Big(m_i + \big((m_{i-l} + c_{i-2l}) \bmod r\big)\Big) \bmod r \\ &= \big((m_i + m_{i-l}) + c_{i-2l}\big) \bmod r, \end{aligned}$$

és indukcióval bármely $i \in \mathbb{N}$-re

$$c_i = \left(\sum_{j=0}^{\left\lfloor \frac{i}{l} \right\rfloor} m_{i-jl} + c_{i-\left(\left\lfloor \frac{i}{l} \right\rfloor + 1\right)l} \right) \bmod r.$$

Ez azt eredményezi, hogy a betűstatisztika egyre inkább egyenletessé válik, és így bonyolódik a fejtés. Ugyanakkor a Cæsar-módszer most használhatatlan, mert ha valaki meghatározza a kulcshosszt, akkor már a fejtés csupán egy kivonás, ugyanis minden $i \in \mathbb{N}$-re most $m_i = (c_i - c_{i-l}) \bmod r$.

Az előbbiektől teljesen eltérő általánosítása az egyábécés helyettesítésnek a **homofonikus rejtjelezés**. Minden $a \in A$-hoz legyen A_a olyan halmaz, hogy ha $a \neq b \in A$, akkor $A_a \cap A_b = \emptyset$, valamint $\frac{|A_a|}{|A_b|} = \frac{f_a}{f_b}$, ahol f_t az ábécé t betűjének relatív gyakorisága. Ekkor $E_k(a) \in A_a$ úgy, hogy minden a-ra az a-nak megfeleltetett betűk eloszlása A_a-n egyenletes. Most a rejtjelezett szövegben – elegendően hosszú szöveg esetén – a betűk eloszlása egyenletes, vagyis az egybetűs statisztika semmit nem árul el a nyílt szövegről. Ez nyilván nehezíti a fejtést, de nem teszi lehetetlenné, mert bár a többi statisztikát is gyengítheti ez az eljárás, de nem tünteti el.

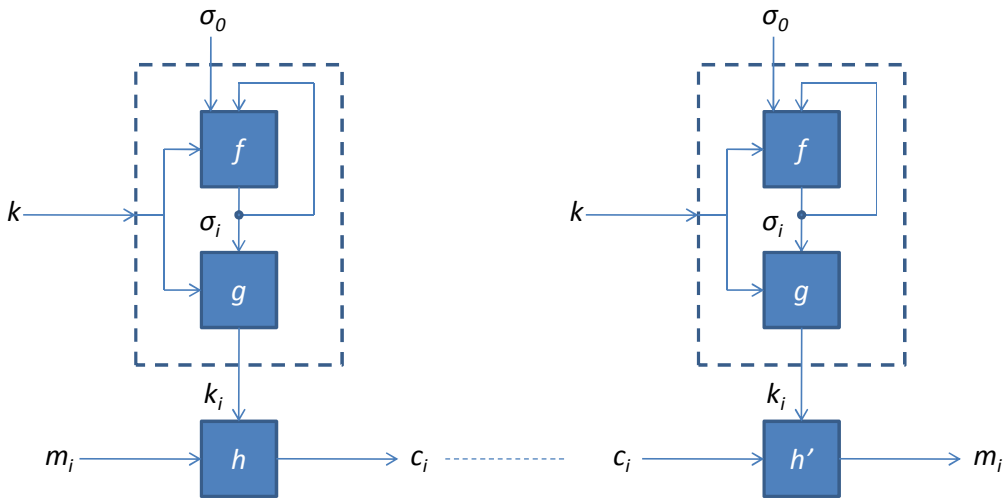
Az eddigiek alapján a titkosító algoritmusoknak két, egymástól eltérő csoportja alakult ki:

1. **folyamrejtjelek**;
2. **blokkrejtjelek**.

A folyamrejtjel – amelyet Vernam javasolt 1917-ben – a többábécés helyettesítésnek felel meg. Ennél a típusnál egy-egy menetben a rejtendő szöveg egy-egy karakterét titkosítjuk, ahol a karakterek az eredeti ábécé betűivel felírt, rögzített hosszúságú szavak (például bináris az ábécé, és a karakterek a bájtok). Az egyes karakterek rejtése a kulcs soron következő karakterével történik, vagyis $c_i = E_{k_i}(m_i)$. A **kulcsfolyam**ot egy **kulcsgenerátor** állítja elő. Ügyelni kell arra, hogy az egy adott pontig előállított kulcsfolyamból ne lehessen következtetni a kulcsfolyam következő karakterére.

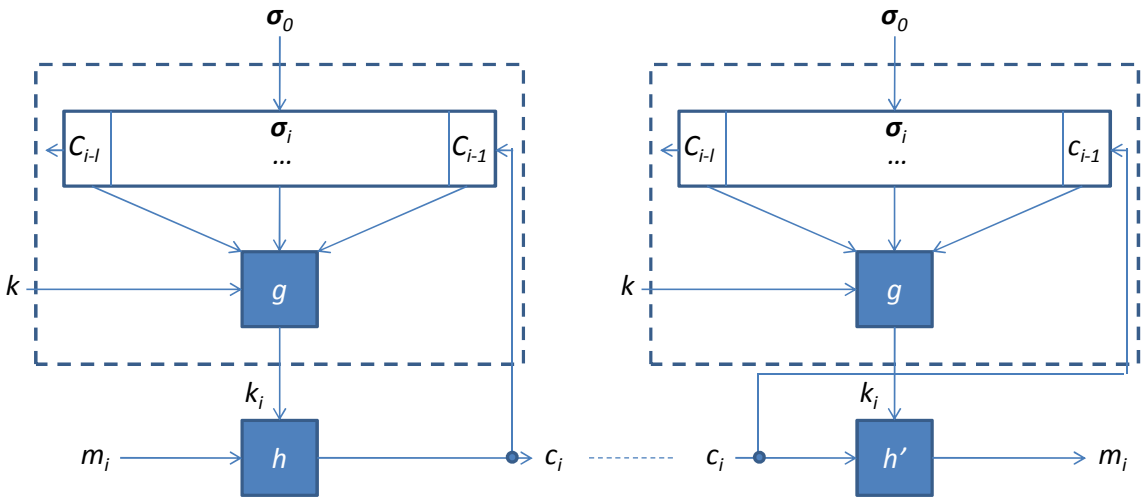
A folyamrejtjeleknek két nagy csoportját különböztetjük meg:

- **szinkron rendszerek**;
- **önszinkronizáló rendszerek**.



2. ábra Szinkron folyamrejtjel

Mindkét esetben a kulcsfolyam aktuális elemét a generátor aktuális **állapot**a határozza meg, és a két rendszert éppen az állapotok formája és a következő állapot előállítása különbözteti meg. Az mindkét rendszerben azonos, hogy az aktuális kulcs az aktuális állapottól és a felhasználó kulcsától függ, vagyis $k_i = g(\sigma_i, k)$, és az aktuális nyílt karakter rejtjelezett képe $c_i = h(m_i, k_i)$, illetve visszafejtésnél $m_i = h'(c_i, k_i)$. A szinkron rendszerben adott egy σ_0 **kezdő állapot**, és minden nem negatív egész i indexre $\sigma_{i+1} = f(\sigma_i, k)$ (2. ábra). Látható, hogy az egyes állapotok teljesen függetlenek a nyílt és a rejtjelezett betűtől. Ezzel szemben az önszinkronizáló rendszernél az állapot karakterek egy sorozata, ahol minden egyes nyílt betű titkosítása után az állapot karakterei eggyel balra lépnek, a bal szélső karakter elvész, és helyére a jobb szélen belép az éppen most kapott rejtjel-karakter (3. ábra).



3. ábra Önszinkronizáló folyamrejtjel

Vizsgáljuk meg a két módszer tulajdonságait. Azt nézzük, hogy mi történik, ha egy karakter

- meghibásodik;
- kimarad;
- beszúródik.

- A szinkron rendszer tulajdonságai:

  - ha átvitel során egy karakter meghibásodik, akkor ezt a karaktert hibásan desifrírozzuk, de a többi karakter fejtését ez a hiba nem befolyásolja;
  - egy karakter törlése vagy beszúrása esetén ettől a ponttól kezdve mindig hibásan fejtünk, hiszen ha az i-edik karakter marad ki, akkor $m'_{i+j} = h'(c_{i+1+j}, k_{i+j})$-t fejtünk ($j \in \mathbb{N}$), és ez általában nem azonos m_{i+j}-vel, mert más kulccsal fejtünk, mint amivel kellene. Beszúrásnál hasonló a helyzet, mert ismét, ha az i-edik beérkező karakter egy „kakukktojás", és ez u, akkor $m'_i = h'(u, k_i)$, és utána $m'_{i+j} = h'(c_{i+j}, k_{i+1+j})$ lesz. Ezt csak úgy tudjuk helyreállítani, ha a hiba helyétől kezdve megismételjük az adást, amit nem tehetünk ugyanazzal a kulccsal, mert ezt felismerve, a támadó nagy eséllyel fejteni tud.

- Az önszinkronizáló rendszer tulajdonságai:

  - ha átvitel során egy karakter meghibásodik, akkor ezt a karaktert hibásan desifrírozzuk, és még az ez után következő l számú karakter desifrírozása is hibás lesz, mert a hibásan beérkezett karakter bekerül a kulcsgenerátorba, és onnan az l-edik lépés után kerül csak ki, de ez alatt az idő alatt végig hibás kulccsal fejtünk;
  - egy karakter törlése esetén a következő karaktert nyilván rosszul fejtjük, és az ezután következő $l-1$-számú karakter fejtése is hibás lesz, de az ezek után következő karaktereket ismét jól fejtjük, vagyis lesz egy kimaradt karakterünk és utána l-számú hibásan fejtett karakter. Ezt annak alapján láthatjuk könnyen be, ha meggondoljuk, hogy helyes fejtésnél a kulcsgenerátorban álló karakterek indexei balról jobbra egyesével nőnek, és jobb szélen az éppen fejtendő karakternél eggyel kisebb indexű karakter van. Amikor kimarad egy karakter, akkor a következő karakter fejtésénél a kulcsgenerátorban a jobb szélső és a mellette álló karakter indexének eltérése 2, és ez mindaddig megmarad, amíg ez a jobb szélső karakter el nem ér a kulcsgenerátor bal széléig, ami éppen $l-1$ lépés után következik be. Beszúrásnál hasonló a helyzet, vagyis lesz egy beszúrt karakterünk, és ezt követően l rosszul fejtett karakterünk, ám ezek után, ha közben egyéb hiba nem lépett fel, is-

mét minden karaktert a „saját” kulcsával fejtünk (ugyanez igaz törlésnél is). Ez azt jelenti, hogy ennél a rendszernél nincs szükség kiesésnél vagy betoldásnál adásismétlésre, a rendszer magától is helyreáll. Ennek az ára azonban, amint láttuk, l egymás utáni hiba.

A folyamrejtjelek speciális esete, amikor a kulcs periodikus, ezeket néha **periodikus rejtjel**eknek nevezik. Ez tekinthető blokkos rejtjelnek, ahol a blokk mérete a periódushossz.

A blokkos rendszerekre az jellemző, hogy betűk egy csoportját egyetlen blokknak tekintve, ezt az adott, fix hosszúságú blokkot mindig ugyanazon szabállyal egy szintén fix – többnyire, de nem mindig az eredetivel azonos – méretű blokkba transzformálja, vagyis $E_k: A^n \to B^t$, ahol A a nyílt szöveg, míg B a rejtjel ábécéje, $n \in \mathbb{N}^+$ a nyílt és $t \in \mathbb{N}^+$ a titkosított szöveg egy blokkjának hossza. Az eljárás tekinthető egy egyábécés helyettesítésnek, ahol a helyettesítendő ábécé most A^n, és a B^t ábécé betűivel helyettesítünk. A továbbiakban az egyszerűség kedvéért legyen $A = B$ és $n = t$ (a valóságban ez a leggyakoribb elrendezés, és tudjuk, hogy a konkrétan alkalmazott jelek formája különösebben nem befolyásolja az alkalmazott rendszer bonyolultságát). Korábban láttuk, hogy ha $|A| = r$, akkor $|A^n| = r^n$, és így az összes lehetséges kulcsot tekintve $|\mathcal{K}| = r^n!$. Ha rögzítünk egy $k \in \mathcal{K}$ kulcsot, akkor ez lényegében véve azt jelenti, hogy megadjuk az r^n lehetséges n-hosszúságú nyílt szöveg egy adott permutációját. Amennyiben a kulcs bármi lehet, vagy legalábbis az összes lehetséges kulcs egy meglehetősen nagy halmazának valamely eleme, és nincs egy egyszerű matematikai eljárás, amellyel bármely nyílt szövegből meg tudjuk határozni az adott kulcshoz tartozó rejtett szöveget (és nagy n esetén szinte biztos, hogy ez a helyzet), akkor nincs más lehetőségünk, mint tárolni minden nyílt szöveg rejtjelezett párját. Ehhez egy adott kulcs esetén nr^n-méretű tárra van szükség, ha elemi adatnak egy-egy betűt tekintünk, és ha $\lceil \log_r |\mathcal{K}| \rceil = t$, vagyis ha a kulcsokat az ábécénk betűiből álló t-hosszúságú szavakkal tudjuk megadni – ez az érték a kulcs **effektív hossz**a – , akkor a szükséges tárméret $r^t n r^n = n r^{t+n}$. Például a majd tárgyalandó DES esetén $r = 2$, $n = 64$ és $t = 56$, és így az összes helyettesítés tárolásához egy $2^6 \cdot 2^{64+56} = 2^{126}$ bites, azaz 2^{123} bájtos tárolóra van szükség. Csak hogy érzékeltessük, ez 10 633 823 966 279 326 983 230 456 482 242 756 608 a tizes számrendszerben felírva, vagyis körülbelül 10^{37}. Ilyen tárolót biztosan nem találunk, így ebben a formában a blokkos rejtjel nem alkalmazható. Amennyiben viszont a blokkméret kicsi, vagy a szabály olyan, amellyel könnyű a rejtés és a fejtés, akkor a rendszer nem megbízható, könnyen törhető. A probléma megoldására Shannon a **keverő transzformáció**t javasolta. Bevezetett két fogalmat:

- **diffúzió**;
- **konfúzió**.

A diffúzió azt jelenti, hogy olyan átalakítást végzünk, amelynél a nyílt szöveg statisztikája szétterül a teljes blokkon, nagyjából egyenletessé válik. Ilyen például a transzpozíció, ahol az egybetűs statisztika megmarad, a több-betűsök teljesen megváltoznak, hiszen eredetileg egymás mellett álló betűk várhatóan nem szomszédos pozícióba kerülnek, és így a módosított szövegben szinte bármely betűpár azonos eséllyel előfordulhat. A konfúziónál viszont az a cél, hogy minél bonyolultabb legyen a rejtjelezett szöveg és a kulcs kapcsolata (vagyis legyen minél nagyobb a kulcs-ekvivokáció értéke). Ezt elsősorban nemlineáris helyettesítéssel érhetjük el.

Shannon szerint az előbbi két módszer egyike sem eredményez önmagában kriptográfiailag erős rendszert (kivéve a véletlen átkulcsolást, amely mint konfúzió, tökéletes biztonságot ad). Shannon többfordulós eljárást javasolt, ahol minden fordulóban először végrehajtunk egy, az eredeti kulcsból fordulónként származtatott kulccsal egy helyettesítést, majd utána egy transzpozíciót, vagyis egy-egy forduló $F(k_i) = S(k_i)P$ alakú, ahol k_i az adott fordulóban alkalmazott kulcs, S a kulcstól függő helyettesítés, míg P egy permutáció, a transzpozíció. Ez így fölösleges szétválasztásnak tűnik, hiszen maga $F(k_i)$ önmagában egy helyettesítés. Láttuk azonban, hogy kis blokk nem biztonságos, nagy blokknál viszont nagyon nagy memória kell a helyettesítések tábláinak megadásához. Shannon ezért azt javasolta, hogy az n-méretű blokkot osszuk fel viszonylag rövid, q-méretű részekre, ahol q az n osztója, ezeket egymástól függetlenül helyettesítsük, majd az így kapott teljes blokkon hajtsuk végre a transzpozíciót úgy, hogy az eddig egy blokkban lévő jelek a következő fordulóban más-más részblokkba kerüljenek. Ha a jelek ilyen szétválasztását az egymás utáni fordulókban ügyesen választjuk

meg, akkor elérhető, hogy az első forduló bemenetén egyetlen jelet megváltoztatva, az utolsó forduló kimenetén nagyjából az összes jel fele változzon, és minden kimeneti jel $1/2$-hez közeli valószínűséggel változzon egy-egy bemeneti jel megváltozásakor. Ez az úgynevezett **lavinahatás**, amely már erős titkosítást eredményezhet, annak ellenére, hogy az egyes helyettesítéseknél kis blokkokat transzformálunk. Az ilyen feldarabolásnál a szükséges tárméret már csak $r^t \frac{n}{q} q r^q = n r^{t+q}$, és ez a korábbi igénynek csupán $\frac{r^n}{r^q} = r^{n-q}$-szorosa, vagyis a konkrét példánk esetében, amennyiben a részblokkok hossza $q = 4$, akkor a társzükséglet $2^6 \cdot 2^{60} = 2^{66}$, ami az eredetinek 2^{60}-adrésze.

Amire ügyelni kell a részblokk méretének megválasztásakor, hogy legyen az adott hosszal nem affin helyettesítés. Határozzuk meg ehhez q minimális értékét. Amennyiben r prímszám (vagy prímhatvány, és a műveletet nem a maradékosztály-gyűrűben, hanem egy testben végezzük), akkor az affin transzformációk száma q hosszúságú blokkokon (lásd a 29. oldalon) $r^q \prod_{i=0}^{q-1}(r^q - r^i)$, míg összesen $r^q!$ helyettesítés van. Ha $r^q! \leq r^q \prod_{i=0}^{q-1}(r^q - r^i)$, akkor csak affin helyettesítés létezik. Ez ekvivalens az $(r^q - 1)! \leq \prod_{i=0}^{q-1}(r^q - r^i)$ feltétellel. Mindkét oldalon legalább egy tényezőt tartalmazó szorzat található. A faktoriálisban az r^q-nál kisebb minden pozitív egész pontosan egyszer fordul elő, a jobb oldali szorzat tényezői is páronként különbözőek, és ha $q > i \in \mathbb{N}^+$, akkor $1 = r^0 \leq r^i < r^q$, így ennek a szorzatnak minden tényezője tényezője a faktoriálisnak is. Ekkor a bal oldali szorzat biztosan nem kisebb a jobb oldalinál, így a megadott egyenlőtlenség csak úgy teljesülhet, ha a két szorzat azonos. Ha $q = 1$, akkor $(r-1)! = r - 1$-nek kell teljesülnie. Mivel $r - 1 > 0$, ezért ez pontosan akkor igaz, ha $(r-2)! = 1$, és ez $r - 2 = 0$ és $r - 2 = 1$, azaz $r = 2$ és $r = 3$ esetén, és csak ebben a két esetben teljesül. $q > 1$ esetén mindkét oldalon legalább kéttényezős szorzat áll. A korábban megállapított tulajdonságokkal $(r^q - 1)! = \prod_{i=0}^{q-1}(r^q - r^i)$ csak úgy lehet, ha a jobb oldalon az egymás utáni tényezők különbsége 1. Ekkor például $1 = (r^q - r^{q-2}) - (r^q - r^{q-1}) = r^{q-1} - r^{q-2} = r^{q-2}(r-1)$, ami akkor és csak akkor igaz, ha mindkét tényező értéke 1, vagyis ha $r = 2$ és $q = 2$. Ezekkel az adatokkal viszont teljesül az $(r^q - 1)! = \prod_{i=0}^{q-1}(r^q - r^i)$ feltétel. Összefoglalva, ha a blokkhossz 1 és az ábécé bináris vagy ternáris, illetve a blokkhossz 2 és kételemű az ábécé, akkor minden helyettesítés affin, minden más esetben van nem affin helyettesítés. A leggyakoribb eset a bináris ábécé használata, és ekkor, az előbbi eredmények szerint, a blokkméret legalább 3. De általában – és főleg bináris esetben – szeretjük a 2-hatvány értékeket, és a 2-nél nagyobbak közül a legkisebb ilyen a 4, ezért általában a blokkhossz minimum 4.

A blokkos rendszereknél a lavinahatás mellett még az is kívánatos, hogy a kimenet bármely betűje a bemenet valamennyi betűjétől függjön, és egy kimeneti betű megváltozásából ne lehessen következtetni a bemeneti változásra.

A blokkos rendszereket alapvetően négyféle üzemmódban használjuk (amelyek közül, mint majd látjuk, kettő lényegében véve folyamrejtjel).

1. **Elektronikus kódkönyv** (***E**lectronic **C**ode **B**ook – ECB*). Ez az eljárás az egyes blokkokat egymástól teljesen függetlenül titkosítja, illetve vételnél fejti a megadott kulcs szerint. Ha egy blokk megsérül (de a hossza nem változik!), mondjuk egy betű megváltozik, akkor ezt hibásan fejtjük, de ez a hiba nem befolyásolja egyetlen más blokk desifrírozását sem. A többi blokk fejtése szempontjából az sem okoz problémát, ha egy teljes blokkot kivonunk, módosítunk vagy beszúrunk, illetve, ha megváltoztatjuk a blokkok sorrendjét. Ugyanakkor nem teljes blokk kivonása vagy beszúrása teljesen tönkreteszi a további részek fejtését (illetve az addig terjedő részekét, ameddig esetleges több kivonás és beszúrás együttesen teljes blokkhossznyi változást nem okoz), hiszen a biztonságos rejtjelek nem affinak, így általában a rejtjelszöveg eltolása nem a nyílt szöveg eltolásának felel meg, és az összes rejtjelblokk tartalmazza az értelmetlen szövegek megfelelőjét is. Az így visszatranszformált szöveg tehát általában értelmetlen nyílt szöveget, úgynevezett **zaj**t eredményez. Ezt szinte azonnal észrevesszük, és jelezhetjük az adónak, aki ekkor megismételheti a hibás rész(eke)t. Ezt azonban nem teheti meg a korábbi kulccsal, mert a kulcsismétlés felismerése eredményes támadást eredményezhet.

A teljes blokkok módosítása, kivágása és beszúrása sokszor nehezen – vagy egyáltalán nem – fedezhető fel, ezért ezt az üzemmódot nem nagyon szabad használni.

2. **Blokkláncolás** (***C**ipher **B**lock **C**haining – CBC*). Választunk egy c_0 kezdő blokkot – szokás **kezdeti érték**nek (***I**nitial **V**alue – IV*) is mondani – , és most $c_i = E_k(m_i + c_{i-1})$ a pozitív egész i in-

dexekre. Az eljárást tekinthetjük egy olyan folyamrejtjelnek, ahol egy-egy blokk felel meg egy-egy karakternek, és ez egy önkulcsolt rendszer, ahol a kulcsot a rejtjelszöveg „betűi" adják egy kezdeti kulcsbetű alapján (de most ez nem lineáris, így a 39. oldalon említett probléma nem jelentkezik). A fejtés is könnyű, hiszen az előbbi felírásból $m_i = D_k(c_i) - c_{i-1}$. Innen rögtön látható, hogy egy blokk meghibásodása (ha a hossza nem változik) ezt a blokkot zajba dekódolja, és a következő blokkban az előbbi blokk hibás helyeit is hibásan kapjuk. Ez a hiba a többi blokk desifrírozását nem érinti. Ha kimarad egy teljes blokk, mondjuk az l indexű, akkor kimarad a nyílt szövegből m_l, és a következő blokkot $m'_{l+1} = D_k(c_{l+1}) - c_{l-1}$-ként fejtjük, amely általában ismét csak zaj, de a további blokkokat már rendben fejtjük (ha további változás nem történik), ám mindegyiket eggyel előrébb csúszva. Hasonlóan, ha az $l-1$-edik blokk után a várt c_l helyett egy b blokk érkezik, akkor a fejtéssel egy $m' = D_k(b) - c_{l-1}$ beszúrt blokkot kapunk, és utána a helyes m_{l+1} helyett $m'_{l+1} = D_k(c_{l+1}) - b$-t, majd az ezutáni blokkokat rendben fejtjük (ismét azzal a feltétellel, hogy további probléma nem lép fel az átvitelnél), de ezek a blokkok eggyel hátrébb állnak a fejtett szövegben, mint az eredeti nyílt szövegben. Azt látjuk, hogy egy-egy blokk módosítására, beszúrására vagy törlésére a rendszer, mint folyamrejtjel, önszinkronizáló. Más a helyzet, ha csak blokk egy része marad ki, vagy ilyennel bővül a rejtjel-folyam, hiszen ekkor ugyanaz a probléma lép fel, mint az ECB-módnál, vagyis ettől a ponttól kezdve már csak zajt kapunk a fejtés eredményeként.

Fontos észrevenni, hogy – mint azt az önkulcsolásnál már említettük – egy-egy rejtjelblokk az összes őt megelőző nyílt szövegtől is függ, így diffúziót eredményez, továbbá a blokkok felcserélése is lehetetlenné teszi egyes blokkok fejtését. A blokkláncolás üzenet-kivonatok készítésére is alkalmas, amikor az eredeti – általában egy blokknál lényegesen hosszabb – m üzenethez kivonatként a legutolsó blokk képét csatoljuk, illetve esetleg s= $E_k(r + c_t)$-t, ahol t az utolsó blokk indexe, és r egy, a küldő és a fogadó által ismert és esetleg bizonyos időközönként változó érték.

3. **Rejtjel-visszacsatolás** (***C**ipher **F**eed**b**ack – CFB*). Legyen a blokkhossz t karakter, ahol most egy-egy karakter az ábécé adott hosszúságú blokkja (például egy karakter egy bájt, azaz a bináris ábécé egy 8-betűs blokkja). A sifrírozás karakterenként történik. Induláskor adott egy kezdeti érték, $b_{-t+1} \cdots b_0$, és ebből a blokkos rejtjelezőnk a $k_{-t+1} \cdots k_0$ blokkot állítja elő. Ekkor a titkosítandó szöveg első karakterének képe $c_1 = m_1 + k_{-t+1}$ lesz, és a blokkos rendszer bemenetére $b_{-t+2} \cdots b_1 c_1$ kerül. Valahányszor egy karaktert titkosítottunk, a blokkos titkosító bemenetén addig volt értéket karakterenként egy hellyel balra léptetjük, vagyis minden jegy eggyel magasabb helyiértékre kerül, az addigi legmagasabb helyiértékű jegy kikerül a rendszerből, és helyette a legalacsonyabb helyiértékű jegyként bekerül az éppen most kapott rejtjel-karakter. Az ezen új bemenetből kapott karaktersorozat legfelső karaktere lesz ezek után a következő nyílt karakter kulcsa egyszerű összeadással, és folytatódik az előbb leírtakkal a folyamat. Ha az alapábécé r-betűs, és a betűk az r-nél kisebb nem negatív egészek, továbbá egy-egy karakter m-betűs, valamint a blokkos rejtjelező aktuális bemenete u, a kimenete pedig v (ahol u és v egy tm hosszúságú blokk az alapábécé betűivel), akkor az előbb leírtaknak megfelelően $c = (m + s) \bmod r^m$, ahol m az éppen titkosítandó karakter és $s = \left\lfloor \frac{v}{(r^m)^{t-1}} \right\rfloor$, és u új értéke $u' = (r^m u + c) \bmod (r^m)^t$ lesz.

$c = (m + s) \bmod r^m$-ből $m = (c - s) \bmod r^m$, és ez az egyetlen eltérés a titkosításhoz képest.

Ez a rendszer egy folyamrejtjel, ahol a blokkos rejtjelező a kulcsgenerátor szerepét tölti be, és az is azonnal látható az ott leírtak alapján, hogy ez egy önszinkronizáló rendszer, amelynek a tulajdonságait már megvizsgáltuk.

4. **Kimenet-visszacsatolás** (***O**utput **F**eed**b**ack – OFB*). Ez egyetlen pontban tér el az előző üzemmódtól, nevezetesen, hogy most nem az éppen kapott rejtjelkaraktert visszük be a blokkos rejtjelező bemenetére, hanem magát a kulcsként használt karaktert, vagyis a blokkos rejtjelező aktuális kimenetének legfelső karakterét, tehát most $u' = (r^m u + s) \bmod (r^m)^t$ lesz. Ebből közvetlenül láthatjuk, hogy ez egy szinkron folyamrejtjel.

Fontos megjegyezni, hogy a CFB és az OFB módszer nem használható nyilvános kulcsú rendszerekben kulcsgenerátorként, hiszen ennél a két eljárásnál a titkosítás és a fejtés kulcsa azonos.

Összehasonlítva a két módszert, a folyamrejtjelt és a blokkos rejtjelt, látjuk, hogy a blokkos rejtjel esetén be kell várni egy teljes blokk beérkezését, és csak ekkor lehet titkosítani illetve fejteni, ami azt jelenti, hogy szükség van tárolóra, és a várakozás késleltetést okoz. Ezzel szemben a folyamrejtjelnél a beérkezett karakter azonnal feldolgozható, és a feldolgozás kisebb adaton történik, így ez a módszer gyorsabb, és nincs külön memóriára szükség. Ennek ellenére a nyilvános rendszerekben inkább a blokkos titkosítók terjedtek el.

A következő részben két konkrét blokkos rejtjelt ismertetünk.

5. A DES és az AES

Az előző részben láttuk, hogy a klasszikus rendszerek alapvetően két módszert alkalmaztak. Az egyik a helyettesítés, amikor egy-egy betűt, vagy a betűk egy csoportját helyettesítik valamilyen jellel, vagy jelcsoporttal, míg a másiknál az üzenet egy-egy meghatározott hosszúságú szakaszán megváltoztatják a betűk sorrendjét, vagyis transzpozíciót alkalmaznak. Ha nagy redundanciájú üzeneteket sifrírozunk, akkor elegendően hosszú üzenet esetén a kulcs könnyen megfejthető. Ha a monoalfabetikus helyettesítést alkalmazzuk, és a rejtjelezett szöveg egy tipikus, hétköznapi szöveg, akkor egy körülbelül 20 betűs szöveg egyértelműen visszafejthető a kulcs előzetes ismerete nélkül. A fejtés alapja a betűfrekvencia. Nehezíti a fejtést, ha tömörítünk. Ha például számsorozatokat rejtjelezünk, amikor bármely sorozat értelmes üzenet lehet, vagyis ha a redundancia 0, akkor ilyen kapaszkodónk nincs a fejtéshez.

A módszert úgy lehet bonyolítani, ha vagy más és más szabállyal végezzük az egyes pozíciókon a helyettesítést, vagy sok betűből álló blokkokat helyettesítünk. Az előbbi szélső esete a véletlen átkulcsolás. Ennél a módszernél természetesen fokozottan igaz, hogy igen nehéz a kulcs biztonságos kicserélése a két fél között, ám mégis alkalmazták a gyakorlatban, nevezetesen a Moszkva és Washington közötti *forró drót*on.

A gyakorlatban inkább a blokkos rejtjelek terjedtek el. Egyik leghíresebb képviselője a **DES** (***D**ata **E**ncryption **S**tandard*), amelyet 1977-ben fogadtak el szabványként, és egészen 2002-ig volt szabvány – ekkor váltotta fel az **AES** (***A**dvanced **E**ncryption **S**tandard*) – , ám amelyet főleg a háromszor egymás után alkalmazott formájában még ma is igen széles körben alkalmaznak. Magának az algoritmusnak a neve **DEA** (***D**ata **E**ncryption **A**lgorithm*). A DES egy igen fontos jellemzője, hogy jóllehet elvileg bármely rejtjelrendszer esetén felteszik, hogy maga az algoritmus ismert, ez volt a világ első olyan rejtjelező algoritmusa, amelyet hivatalosan nyilvánosságra hoztak (bár vannak, akik ezt nem akarják elhinni, és feltételezik, hogy a rendszert kifejlesztő *IBM* bizonyos információt megtartott magának, amelynek a segítségével képes fejteni a titkosított üzeneteket). A DES jelentőségét még az is alátámasztja, hogy az azóta kifejlesztett blokkos rejtjelező rendszerek majd mindegyike többé-kevésbé a DES-nél alkalmazott elvekre, de legalábbis az elvek egy részére épül.

A *DES* három pilléren nyugszik.

1. A DES az előző fejezetben ismertetett, Shannon által javasolt keverő transzformáció alapján működik.
2. Tegyük fel, hogy egy r-betűs ábécével írt n-betűs blokkot p-hosszúságú kulccsal rejtjelezünk (maga a kulcs a szöveggel azonos ábécéből épül fel). Ekkor egy blokk kiszámítása lényegében véve egy $n+p$ változótól függő, n egyenletből álló egyenletrendszer:

$$n \geq i \in \mathbb{N}^+ : c_i = h_i(m_1, \cdots, m_n; k_1, \cdots, k_p),$$

ahol m_j a nyílt szöveg egy blokkjának j-edik, k_l a kulcs l-edik, és c_i a rejtjelezett szöveg blokkjának i-edik betűje. Ha r egy prímhatvány, akkor a szimbólumhalmaz egy véges testnek tekinthető. Ilyen esetben bármely leképezés, amely a véges testet önmagába képezi, megadható egy polinommal, így feltehetjük, hogy a h_i leképezés az f_i polinomhoz tartozó $\hat{f}_i$ polinomfüggvény. Hasonló a helyzet a desifrírozás esetén:

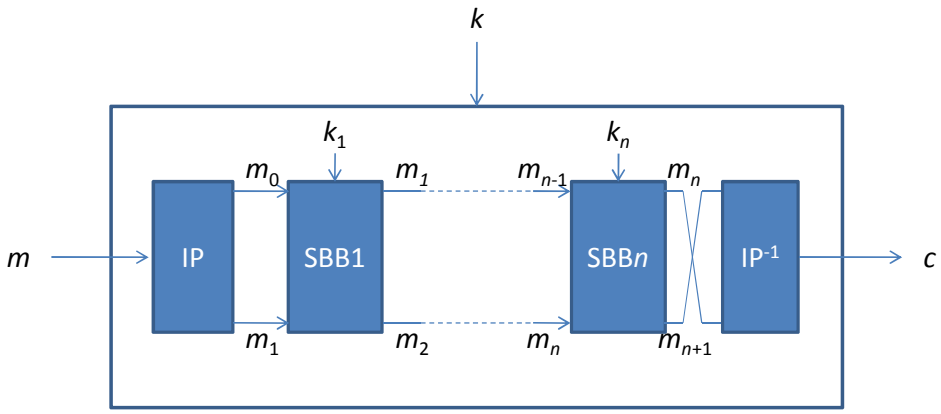
$$n \geq i \in \mathbb{N}^+ : m_i = \hat{g}_i(c_1, \cdots, c_n; k_1, \cdots, k_p).$$

Ha az algoritmus nyilvános, akkor ismertek a polinomok, és ekkor a fejtés egyszerű behelyettesítés, ám a kulcs ismerete nélkül a feladat az eredeti polinomok által meghatározott egyenletrendszer megoldását jelenti. Ha a polinomok nem lineárisak, akkor viszont az ilyen egyenletrendszer megoldása általában **NP**-nehéz, és így elegendően nagy blokkok esetén a fejtés – bár elméletileg lehetséges – gyakorlatilag a használható időn belül reménytelen feladat.

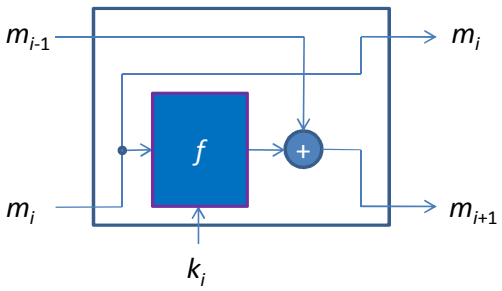
3. A DES az úgynevezett **Feistel-struktúra** alapján működik. Legyen a rejtjelezendő szöveg m, amelynek hossza $2n$, és válasszuk két részre úgy, hogy az egyik rész a szöveg első n betűjéből, míg a másik a hátsó n betűből áll, azaz az előbbit $m^{(0)}$-lal, a másodikat $m^{(1)}$-gyel jelölve $m = m^{(0)} \parallel m^{(1)}$ ($\parallel$ most és a későbbiekben a konkatenációt jelöli). Tegyük továbbá fel, hogy az algoritmus t **fordulóból**, t **menet**ből (*round*) áll, és minden egyes fordulóban az eredeti kulcsból származtatott alkulcsot, (**menetkulcs** vagy **fordulókulcs,** *round-key*) alkalmazunk. A Feistel-struktúrában alkalmazott transzformáció ezek után a következő:

$$t \geq i \in \mathbb{N}^+ : m^{(i+1)} = m^{(i-1)} + f\left(m^{(i)}, k^{(i)}\right).$$

Az m-hez tartozó rejtjelezett szöveg $c = m^{(t+1)} \parallel m^{(t)}$. A kulcs ismeretében a visszafejtés rendkívül egyszerű, hiszen c-ből ismert $m^{(t+1)}$ és $m^{(t)}$, és ha ismerjük valamilyen $t \geq j \in \mathbb{N}^+$-ra $m^{(j+1)}$-et és $m^{(j)}$-t, akkor $m^{(j+1)} - f\left(m^{(j)}, k^{(j)}\right) = m^{(j-1)}$, vagyis c-ből meg tudjuk határozni $m^{(1)}$-et és $m^{(0)}$-t, tehát m-et. Látható, hogy a rejtjelező és a visszafejtő algoritmus csak annyiban tér el, hogy az egyikben összeadás, a másikban kivonás áll (ami bináris esetben egyébként megegyezik), és a kulcsokat fordított sorrendben kell alkalmazni. Igen lényeges tulajdonsága a Feistel-struktúrának, hogy f nem feltétlenül invertálható, amely tulajdonság nagyon megkönnyíti a rejtjelezés szempontjából jó tulajdonságú függvény keresését. A struktúrát a 4.- 7. ábra mutatja (SBB a ***S**tandard **B**uilding **B**lock* rövidítése, míg IP a **kezdeti permutáció**t – ***I**nitial **P**ermutation* – jelöli).

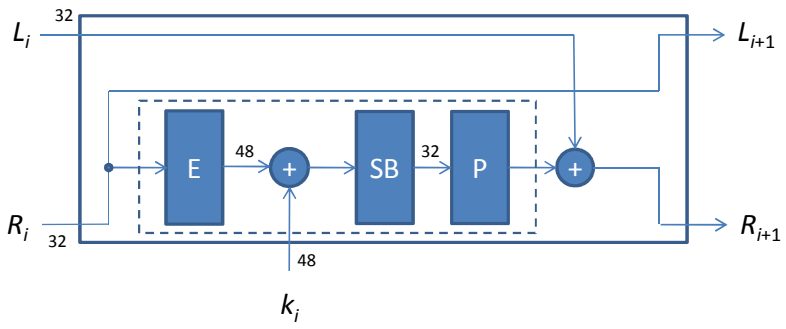


4. ábra Feistel-struktúra

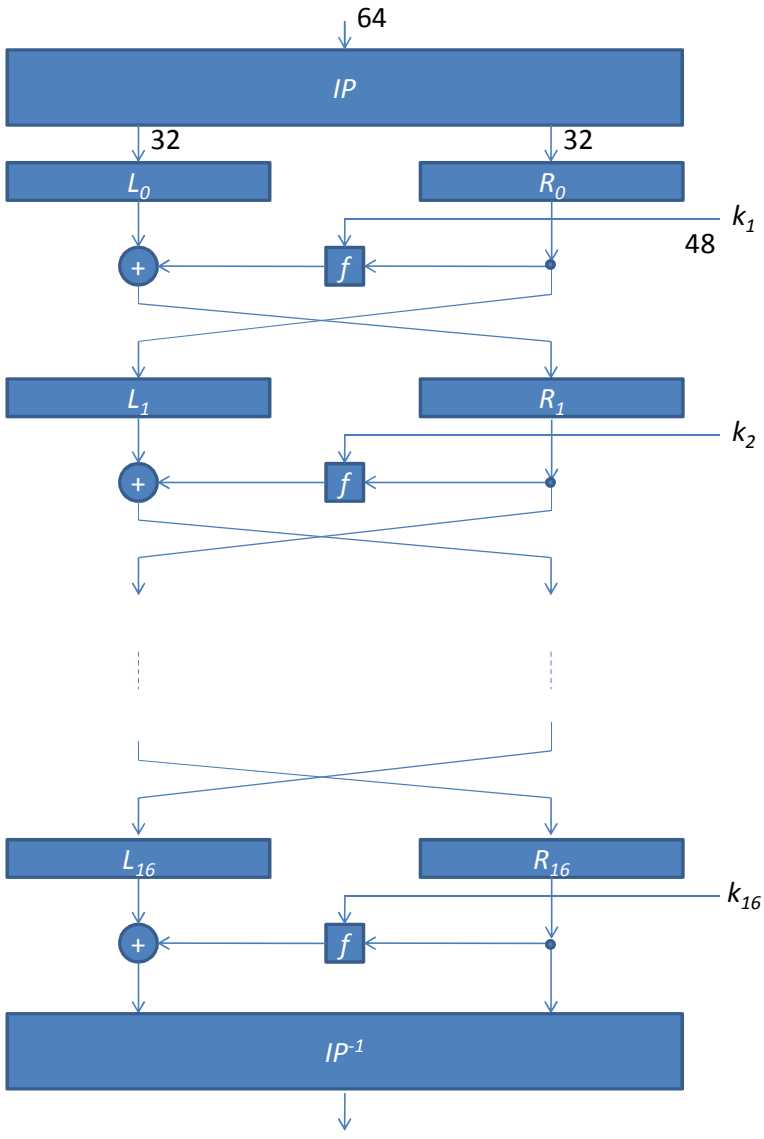


5. ábra Egy SBB

Konkrétan a DES esetén az ábécé két betűből, a 0-ból és az 1-ből áll, és a blokkméret 64 bit. A kulcs is 64 bites, de ebből 8 bit ellenőrző funkciót lát el, így valójában a kulcs 56 bites (ezt vetették leginkább a DES szemére, és sokan úgy vélték, hogy azért választották ilyen méretűre a kulcsot, mert a titkosszolgálatok a maguk számítástechnikai apparátusaikkal abban az időben ekkora méretekkel boldogultak). Az eljárás 16-fordulós. A bináris ábécé esetén a kivonás megegyezik az összeadással, így a desifrírozás teljes egészében megegyezik a sifrírozással, csupán a kulcsokat kell fordított sorrendben alkalmazni.



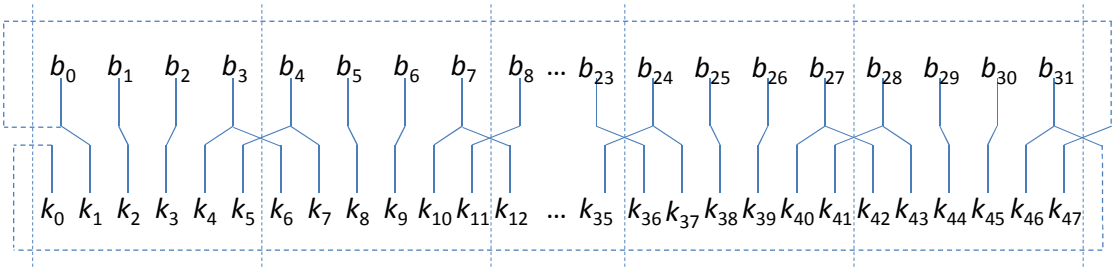
6. ábra SBB a DES-nél



7. ábra A DES létrája

Az 56-bites kulcsot két 28-bites részre osztják, és ezeket az egyes fordulók után – egymástól függetlenül – 1- illetve 2 hellyel ciklikusan elléptetik úgy, hogy a 16 fordulóban összesen mindkét fél 28 hellyel tolódik el, vagyis visszaáll az eredeti helyzet. Ennek csupán annyi jelentősége van, hogy az egyes blokkok titkosítása illetve fejtése után a kulcs azonnal felhasználható, nincs szükség egy kezdő-érték-adásra. A 48-bites forduló-kulcs az így elléptetett két félrészből a szabványban megadott módon kerül kiválasztásra.

Az *E*-blokk egy kiterjesztést végez 32 bitről 48 bitre, amint a 8. ábra mutatja. Láthatóan minden négybites blokkot kiegészítünk a szomszédja felé eső szélső bitjével (a két szélső blokkot is szomszédosnak tekintve).



8. ábra Kiterjesztés

A konfúziót a **helyettesítő doboz**ok (***S-Box***, ***S****election* ***Box***) végzik. A dobozokat 6-6 bit vezérli, és a kimenet egy 4-bites adat. Az egyes blokkok leképezéseit táblázatok tartalmazzák. Minden táblázat 4 sorból és 16 oszlopból áll. A négy sort a bemenet 6 bitjéből a két szélső mint egy kétjegyű bináris szám vezérli, míg az oszlopot a belső 4 bitből álló négyjegyű bináris szám határozza meg. A táblában minden adat egy 16-nál kisebb nem negatív egész szám, azaz éppen egy négyjegyű bináris szám. Végül a diffúziót a *P*-doboz végzi, amely egy permutációja az *S*-doboz kimenetén megjelenő 32-bites adatnak, biztosítva a lavinahatást.

Egy k kulcs **gyenge kulcs**, ha minden m üzenetre $E_k(m) = m$, és a (k_1, k_2) kulcspár **félig gyenge kulcspár**, ha ismét minden lehetséges m üzenet esetén $E_{k_2}\left(E_{k_1}(m)\right) = m$ (vagyis abban az esetben, ha mindig teljesül az $E_{k_1}(m) = D_{k_2}(m)$ egyenlőség, azaz ha k_1-gyel titkosítva mindig ugyanazt kapjuk, mint ha ugyanazt a szöveget k_2-vel fejtjük). Ezeket a kulcsokat illetve kulcspárokat nyilván nem célszerű alkalmazni. A DES esetén négy gyönge kulcs és hat félig gyönge kulcspár van.

Nézzük meg a 6. ábra segítségével, hogy mi történik, ha m helyett $\overline{m}$-et titkosítjuk, és a k kulcsot is kicseréljük a $\bar{k}$ kulcsra, ahol $\bar{u}$ az u bináris szóból úgy keletkezik, hogy minden egyes u_i bitet az ellentettjével, azaz $1 - u_i$-vel helyettesítünk. Az összeadást mindenütt modulo 2 végezzük, és ha ezt az összeadást $\oplus$ jelöli, akkor $\bar{u} \oplus v = (1 \oplus u) \oplus v = 1 \oplus (u \oplus v) = \overline{u \oplus v}$. Ebből az összefüggésből azt is kapjuk, hogy $\bar{u} \oplus \bar{v} = u \oplus v$. Vegyük még tekintetbe, hogy a 32 bitről 48 bitre való kiterjesztésnél csupán egyes biteket kétszer szerepeltetünk, így az E-blokk bemenetén minden bitet negálva a kimenet bitjei is az eredeti kimeneti bitek ellentettjei lesznek. Hasonlóan, a kulcs bitjeit negálva a fordulókulcsokkal is ez történik, hiszen a fordulókulcsot az eredeti kulcs bizonyos részének léptetésével, és egyes bitek kiválasztásával kapjuk. Mindebből az következik, hogy a helyettesítő blokkok bemenetén nem történik változás, így a kimenetük sem változik, ezt a változatlan adatot permutálva ismét ugyanazt kapjuk, mint az eredeti kulccsal az eredeti adatból, és végül ehhez a változatlan félszóhoz hozzáadva a másik félszó ellentettjét, a kimeneten ez a félszó negálódik, mint ahogy a kimenet másik félszava is, hiszen az a bemenetre adott negált félszó változtatás nélkül. Mindez azt jelenti, hogy minden fordulóban a bemenetre adott negált szó eredményeként a kimeneten is negálva jelenik meg az adat, tehát ez igaz a végeredményre is, vagyis azt kapjuk, hogy $E_{\bar{k}}(\overline{m}) = \overline{E_k(m)}$. Ez az eredmény csupán azért érdekes, mert választott nyílt szövegű támadásnál a szükséges vizsgálatok számát nagyjából a felére csökkenti. Valóban, tegyük fel, hogy rendelkezésünkre áll az (m, c_1) és az $(\overline{m}, c_2)$ nyílt szöveg - rejtjelszöveg pár. Ha egy adott kulccsal $E_k(m) = c_1$, akkor megtaláltuk a keresett kulcsot. Ellenkező esetben nézhetjük, hogy vajon teljesül-e az $E_k(m) = \bar{c}_2$ egyenlőség. Ha igen, akkor ez azt jelenti, hogy $E_{\bar{k}}(\overline{m}) = c_2$, vagyis $\bar{k}$ az aktuális kulcs, azaz a kimerítő keresésnél most elegendő a kulcsok felét, egy adott k kulcs esetén csupán k és $\bar{k}$ egyikét kipróbálni, csak az egyikkel transzformálni m-et, mert ez a fentiek tükrében egyúttal a másik kulcsot is leellenőrzi.

A mai számítástechnikai eszközökkel a DES fejtése könnyű feladat, ezért különböző kulccsal többször egymás után alkalmazzák. Egy rejtjelező rendszer több kulccsal való iterációja csak akkor

eredményezhet az egy kulccsal való titkosításnál erősebb védelmet, ha a rejtjelező rendszer nem rendelkezik a **csoporttulajdonság**gal, azaz a leképezések kompozíciója nem alkot csoportot, vagyis ha két egymás utáni titkosítás nem állítható elő valamely kulccsal történő egylépéses leképezésként (ha a nyílt és a rejtett szövegek halmaza azonos, akkor az invertálhatóság eleve teljesül, hiszen a titkosítás injektív, és véges halmaz önmagába való injektív leképezése szürjektív is, tehát egyben bijektív). Ez a DES esetén teljesül (és az egyes fordulókra is, hiszen ezekre is teljesülnie kell az előbbi megkötésnek). 64-bites blokk összesen 2^{64} van, és ezek összes permutációjának száma $2^{64}! \approx \sqrt{2\pi \cdot 2^{64}} \left(\frac{2^{64}}{e}\right)^{2^{64}} \approx 10^{10^{20}}$, vagyis összesen ennyi kulcs lehetséges. Ebből a DES mindössze $2^{64} \approx 1{,}84 \cdot 10^{19}$-t használ. Az összes kulcs csoportot, a 2^{64}-edfokú szimmetrikus csoportot alkotja, és a DES által használt kulcsok ennek a csoportnak egy részhalmazát – de nem részcsoportját! – alkotják. Ez a részhalmaz egy nagyságrendileg 10^{2499}-rendű részcsoportot generál.

Kevés számolással kimutatható, hogy a kétszeres DES (a **dupla** DES) sem nyújt ma már kellő védelmet. Ismét csupán az elv miatt megmutatjuk, hogy hogyan lehet kimerítő kereséssel feltörni ezt a rendszert. Az elv lényege a **középen találkozás**. Tegyük fel, hogy ismert egy összetartozó (m, c) pár. Most két kulccsal, egymás után alkalmazva történik a titkosítás, vagyis $c = E_{k_2}\left(E_{k_1}(m)\right)$, és innen átrendezéssel $D_{k_2}(c) = E_{k_1}(m)$. Rejtjelezzük m-et az összes lehetséges kulccsal, és az eredményt tároljuk el. Ez összesen $2^{56} \approx 7{,}2 \cdot 10^{16}$ titkosítást és egy (8 bájtban mért) ilyen méretű tárat igényel. Most kezdjük el egymás után a kulcsokkal kiszámítani $D_k(c)$-t, és minden kulcs felhasználása után ellenőrizzük le, hogy a kapott érték szerepel-e a táblázatban. Ha igen (és ilyen találkozás biztosan lesz, például amikor éppen k_2-t alkalmazzuk), akkor tehát az éppen alkalmazott k_2' kulccsal és a táblázatban megtalált adathoz tartozó k_1' kulccsal $D_{k_2'}(c) = E_{k_1'}(m)$. Sajnos ebből nem következik, hogy (k_1', k_2') a keresett kulcspár, vagyis hogy $(k_1', k_2') = (k_1, k_2)$. Mivel a kulcspár mindkét eleme 56 bites, a teljes kulcspár mérete 112 bit, az alkalmazott transzformációk száma tehát 2^{112}. Ugyanakkor a 64-bites szövegeket 64-bites szövegekbe transzformáljuk, így a 2^{112} különböző leképezés eredménye egy legfeljebb 2^{64}-méretű halmaz. Ez azt jelenti, hogy átlagosan $\frac{2^{112}}{2^{64}} = 2^{48}$ kulcs m-et ugyanabba a $\tilde{c}$-be titkosítja. Másként mondva, az előbb fellelt (k_1', k_2') kulcspár a lehetséges 2^{48} kulcspár egyike, de hogy melyik, azt az eddigi adatokból nem tudjuk eldönteni. Más a helyzet, ha rendelkezésünkre áll még egy ismert nyílt szöveg - rejtjelszöveg pár, mondjuk egy (m', c') pár. Ekkor ugyanis ellenőrizhetjük, hogy teljesül-e erre a párra is az adott kulcspárral a $D_{k_2'}(c') = E_{k_1'}(m')$ egyenlőség, vagyis, másként írva, fennáll-e az $E_{(k_1', k_2')}(m') = E_{k_2'}\left(E_{k_1'}(m')\right) = c'$ összefüggés. Jó rejtjelrendszer esetén feltehetjük, hogy a különböző kulcsok egy adott üzenetet egyenletes eloszlással transzformálnak, vagyis az előbbi egyenlőség valószínűsége egy hamis – tehát a ténylegestől különböző – kulcspárral 2^{-64}. Mivel összesen – 1 eltéréssel – 2^{48} hamis kulcspárunk van, ezért annak a valószínűsége, hogy egy hamis kulcspárral egyszerre teljesül az $E_{(k_1', k_2')}(m) = c$ és $E_{(k_1', k_2')}(m') = c'$ egyenlőség, csupán $\frac{2^{48}}{2^{64}} = 2^{-16}$, és ez már elhanyagolható, vagyis feltehetjük, hogy $(k_1', k_2') = (k_1, k_2)$. Láthatóan a második menet legfeljebb 2^{56} lépés után befejeződik. A kulcs felleléséhez szükséges lépések száma tehát a közvetlen keresés 2^{112} lépése helyett maximum $2 \cdot 2^{56} = 2^{57}$.

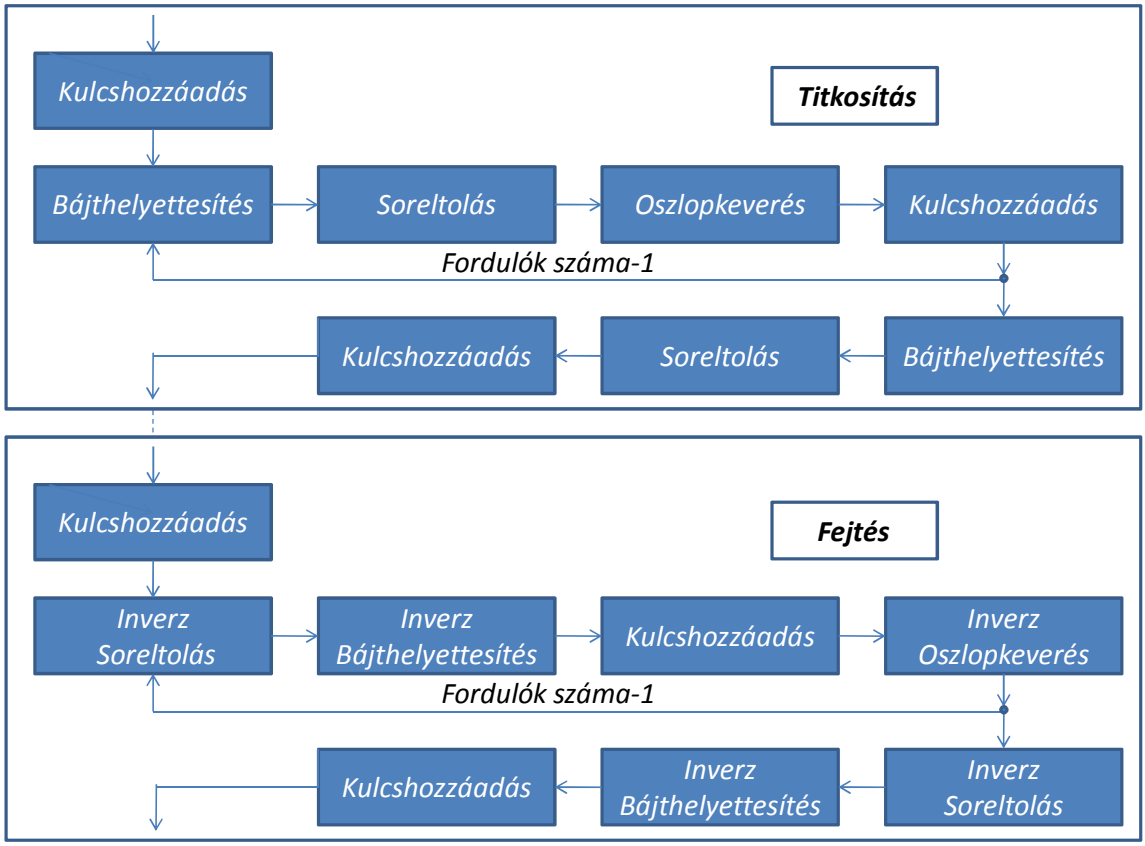
A DES hosszú élete során számtalan új, fontos és szép módszer született a rendszer analízisére és ennek eredményeként a feltörésére, olyan módszer, amely más blokkos rendszerek esetén is sikerrel alkalmazható. Ezek közül csupán kettőt említünk (és szinte tényleg csak említjük).

1. **Differenciál-kriptoanalízis**. Ha az $m \mapsto c$ leképezés affin, azaz $c = am + b$, akkor m-től függetlenül $a(m + \Delta) + b = (am + b) + a\Delta = \mathrm{c} + a\Delta = \mathrm{c} + \Delta'$, vagyis valahányszor m_1 és m_2 olyan üzenetek, hogy $m_1 - m_2 = \Delta$, mindannyiszor $c_1 - c_2 = \Delta'$. Más a helyzet, ha az E transzformáció nem affin. Legyen az üzenetek halmaza az n-nél kisebb nem negatív egészek halmaza, és képezzen E ugyanebbe a halmazba (az injektivitás következtében bijektíven a teljes halmazra). Ha most az n-hez relatív prím, $n > \Delta \in \mathbb{N}$-nel minden m-re $E(m + \Delta) - E(m) = \Delta'$ (az összeadást most mindig modulo n végezzük), akkor a leképezés affin lesz. $(n, \Delta) = 1$ következtében ugyanis létezik olyan

$n > k \in \mathbb{N}$, hogy $m = k\Delta$, vagyis $E(m) = E(k\Delta)$, és van olyan $n > \tilde{\Delta} \in \mathbb{N}$, amellyel $\Delta\tilde{\Delta} = 1$. De indukcióval könnyen kiadódik, hogy $E(k\Delta) = E(0) + k\Delta'$, és innen az $a = \tilde{\Delta}\Delta'$ és $b = E(0)$ jelöléssel $E(m) = E(k\Delta) = E(0) + k\Delta' = E(0) + k\Delta\tilde{\Delta}\Delta' = E(0) + (\tilde{\Delta}\Delta')m = am + b$. Ez tehát azt jelenti, hogy amennyiben E nem affin, akkor (az n-hez relatív prím) bármely bemeneti differencia nem állandó kimeneti differenciát eredményez. A differenciál-kriptoanalízis ezt használja ki. Sok különböző bemeneti differenciához választunk több azonos differenciával rendelkező nyílt szöveg-párt, és tekintjük a párok rejtjelszövegeinek eltérését. Ezek eloszlásából lehet következtetéseket levonni a kulcsra (egy hasonló mérőszám $D(a,b) = |\{n > m \in \mathbb{N} | E_k(m + a) - E_k(m) = b\}|$, ahol $n > a \in \mathbb{N}$, $n > b \in \mathbb{N}$, és láthatóan a kriptoanalízis az előbbi halmazok egy-egy részhalmazának nagyságát vizsgálja).

2. **Lineáris kriptoanalízis**. Most az $L(a,b) = |\{n > m \in \mathbb{N} | am - bE_k(m)\}| - 2^{t-1}$ értéket vizsgáljuk, ahol a és m t-bites, b és $E_k(m)$ r-bites vektorok, és a szorzás a vektorok skalárszorzása. a a bemeneti, b a kimeneti vektort maszkolja, vagyis azt nézzük, hogy az m azon bitjei által alkotott részvektora, ahol a-ban 1 áll, lineáris kapcsolatban áll-e a kimenet b 1-eseihez tartozó komponenseiből álló részvektorral. Ismét nem az összes lehetséges üzenetet tekintjük (hiszen ekkora munkával már kimerítő keresést is végezhetnénk), de jelentős számú összetartozó nyílt szöveg - rejtjelszöveg párral és több különböző (a,b) párral végezzük a vizsgálatot. Látható, hogy ez egy ismert nyílt szövegű támadás (míg a differenciál-kriptoanalízis választott nyílt szövegű támadás).

Láttuk, hogy a kétkulcsos megoldás nem biztonságos, ám a háromszoros DES (**tripla** DES) biztonságosnak mondható, és igen sok helyen alkalmazzák ma is. Egy változata valójában két kulcsot alkalmaz a három fordulóban: $E_k(m) = E_{k_1}\left(D_{k_2}\left(E_{k_1}(m)\right)\right)$.



9. ábra **Az AES titkosítása és fejtése**

Most áttérünk jelenünk szabványára, az **AES**-re (***A**dvanced **E**ncryption **S**tandard – Fejlett Titkosítási Szabvány*; az algoritmus neve ***A**dvanced **E**ncryption **A**lgorithm*). Ez szintén blokkos titkosító, a blokk mérete 128 bit, míg a kulcs választhatóan 128, 192 illetve 256 bites (az eredeti változatban, a

Rijndael-ben, amelyet **Joan Daemen** és **Vincent Rijmen** alkotott, mind a blokkméret, mind a kulcs hossza 128 bittől 32-bites növekménnyel tetszőlegesen választható azzal a megkötéssel, hogy a blokk hossza nem haladhatja meg a 256 bitet). Ez az algoritmus is többfordulós, de nem használja a Feistel-struktúrát. A fordulók száma a kulcs hosszának függvényében 10, 12 illetve 14.

Az AES a 256-elemű test, $\mathbb{F}_{2^8}$ felett működik, vagyis tulajdonképpen bájtokat manipulál. A test $\mathbb{Z}_2$-nek az $f = x^8 + x^4 + x^3 + x + 1$ polinom egy gyökével való bővítése (f irreducibilis $\mathbb{Z}_2$ fölött, mert például $d_k = \left(f, x^{2^k} - x\right) = 1$, ha $4 \geq k \in \mathbb{N}^+$). Mind az adatokat, mind a kulcsot egy négysoros mátrixba rendezi, amelyet az adat 16 és a kulcs 16, 24 illetve 32 bájtjával oszlopfolytonosan töltünk fel (így az adat mátrixa 4, a kulcs mátrixa a kulcs hosszától függően 4, 6 illetve 8 oszlopból áll). Egy-egy oszlop egy formálisan harmadfokú (vagyis legfeljebb harmadfokú) polinom $\mathbb{F}_{2^8}$ felett, míg egy $B = b_0b_1b_2b_3b_4b_5b_6b_7$ bájt (ahol b_0 a legkisebb helyiértékű bit) a $\mathbb{Z}_2$ fölötti legfeljebb hetedfokú $\tilde{B} = b_0 + b_1x + b_2x^2 + b_3x + b_4x^4 + b_5x^5 + b_6x^6 + b_7x^7$ polinomot reprezentálja.

A fordulók kulcsait a szabványban megadott módon generáljuk a felhasználó kulcsából. Az egyes fordulók négy különböző műveletet végeznek (9. ábra).

1. **Bájthelyettesítés** (*SubBytes*). Az adatblokk minden egyes nullától különböző B bájtját az $\mathbb{F}_{2^8}$-beli inverzével helyettesítünk, míg a nullát változatlanul hagyjuk, és utána a kapott C bájtot (tehát $B \neq 0$ esetén $C = B^{-1}$-et, míg $B = 0$ esetén $C = B = 0$-t) egy $\mathbb{Z}_2$ fölötti $B' = \mathbf{A}C + \mathbf{b}$ affin transzformációval átalakítjuk, ahol

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

A teljes helyettesítés megadható algebrailag is, mert az invertálás algebrailag elvégezhető a testben (illetve szorzótábla alkalmazásával közvetlenül leolvasható), és a fenti affin helyettesítés eredménye azonos a $\mathrm{B}' = \left(\widetilde{\mathrm{B}} \cdot (\mathrm{x}^4 + \mathrm{x}^3 + \mathrm{x}^2 + \mathrm{x} + 1) + (\mathrm{x}^6 + \mathrm{x}^5 + \mathrm{x} + 1)\right) \bmod (\mathrm{x}^8 + 1)$ kifejezésnek megfelelő bájttal. Ez a teljes helyettesítés megadható (és meg is adják) egyetlen helyettesítő táblázattal. Az algoritmusban ez a művelet biztosítja a nemlinearitást.

A művelet invertálható, ugyanis $h = x^4 + x^3 + x^2 + x + 1$ és $m = x^8 + 1$ relatív prímek. Ez például onnan látható, hogy $\mathbb{Z}_2$ fölött $x^8 + 1 = x^{2^3} + 1 = (x + 1)^{2^3} = (x + 1)^8$, és ennek a polinomnak egyetlen (nyolcszoros) gyöke van, az 1, ami viszont nem gyöke a $\mathbb{Z}_2$ fölött irreducibilis h-nak, így h nem osztója m-nek. Ekkor viszont létezik h-nak $\bmod m$ inverze (egyébként **A**-ról is belátható, hogy reguláris $\mathbb{Z}_2$ fölött, tehát van inverze a kételemű test fölött).

2. **Soreltolás** (*ShiftRows*). Az adatblokk 4×4-es mátrixában az i-edik sort (0-tól kezdve az indexelést) i hellyel ciklikusan balra léptetjük. Ez nyilván invertálható művelet.

3. **Oszlopkeverés** (*MixColumns*). Ennél a műveletnél az adatmátrix egyes oszlopait egy $\mathbb{F}_{2^8}$ fölötti legfeljebb harmadfokú polinomnak tekintjük (a 0-indexű sorban a polinom konstans tagjával). A polinomot az $\mathbb{F}_{2^8}$-beli szabályokkal megszorozzuk a $g = 03_{16}x^3 + 01_{16}x^2 + 01_{16}x + 02_{16} \in \mathbb{F}_{2^8}[x]$ polinommal, és vesszük a szorzat maradékát $\bmod (x^4 + 1)$. Könnyű ellenőrizni, hogy a nem negatív egész i-re $x^i \bmod \left(x^l - 1\right) = x^{i \bmod l}$, így

$$\begin{pmatrix} 02_{16} & 03_{16} & 01_{16} & 01_{16} \\ 01_{16} & 02_{16} & 03_{16} & 01_{16} \\ 01_{16} & 01_{16} & 02_{16} & 03_{16} \\ 03_{16} & 01_{16} & 01_{16} & 02_{16} \end{pmatrix} \cdot \begin{pmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{pmatrix}$$

a kívánt eredményt adja. A szorzat második tényezője az aktuális adatblokk, és a szorzást és összeadást a 256-elemű test műveletei szerint kell végezni.

Jóllehet $x^4 + 1$ nem irreducibilis, hiszen 2-karakterisztikájú test fölött $x^4 + 1 = (x + 1)^4$, de a g polinomnak van modulo $(x^4 + 1)$ inverze $\mathbb{F}_{2^8}$ fölött, a $\tilde{g} = 0B_{16}x^3 + 0D_{16}x^2 + 09_{16}x + 0E_{16}$ polinom, így ez a lépés is invertálható.

4. **Kulcshozzáadás** (*AddRoundKey*). A felhasználó kulcsából a **kulcsütemező** állítja elő a szabványnak megfelelően az egyes fordulók kulcsát, amelyet ebben a lépésben hozzáadnak az adatblokkhoz. Ez a lépés természetes módon invertálható.

A 9. ábra mutatja, hogy az egyes lépések hogyan követik egymást. Látható, hogy az utolsó körben kimarad az oszlopkeverés.

Jelenleg sokan vizsgálják az AES-t, keresik a gyenge pontjait, próbálják feltörni. Hallani néha híreket, hogy ezek a kísérletek sikerrel jártak, de a nyilvános adatok szerint ez a valóságban nem történt meg, az AES biztonságosnak mondható, és jelen ismereteink szerint sokáig az is marad. Persze a kriptográfiában a jövő sokszor kiszámíthatatlan, így ki tudja, mit hoz a jövő (és főleg, hogy mikor derül ki egy esetleges sikeres törés).

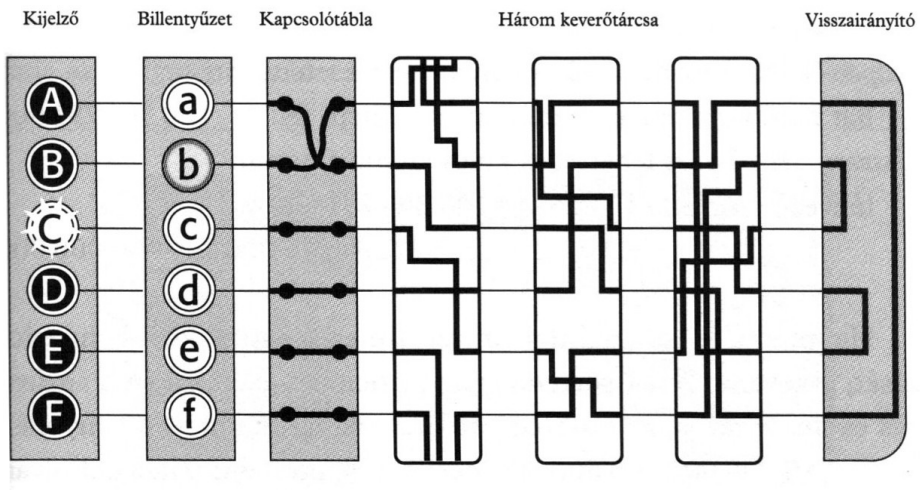
Vissza a tartalomhoz

6. Az ENIGMA

A rejtjelező - visszafejtő tevékenység gépesíthető is, így bonyolultabb eljárások alkalmazhatóak. Ennek egyik példája az **Enigma** (*αἴνιγμα ainigma*, kis rejtvény).

Az *Enigma*[1] egy olyan elektromos írógép, amelynek három főbb egysége van: egy billentyűzet a nyílt szöveg betűinek bevitelére, egy átalakító egység, amely a nyílt szöveg betűit a rejtjeles szöveg megfelelő betűivé alakítja, és egy kijelző panel, amelyen kis lámpácskák felvillanása jelzi a rejtjeles szöveg betűit. A felvillanó betűket leírva kapjuk meg a rejtjeles szöveget, amelyet azután rádión továbbítottak a címzettnek. A vevő oldalon egy azonos beállítású Enigmával írták le a szöveget, és a lámpák felvillanása adta vissza a nyílt üzenetet.

A gép legfontosabb része az átalakító egység, amely három kivehető keverőtárcsából áll, így ezek cserélhetőek. A keverőtárcsa egy vezetékekkel sűrűn teleszőtt, vastag gumitárcsa. A nyílt szöveg betűinek sifrírozását a keverőtárcsák belső huzalozása határozza meg. Ha a tárcsák fix helyzetűek lennének, akkor a tárcsák huzalozása egy egyszerű egyábécés helyettesítéses eljárást valósítana meg. *Scherbius* gépének viszont a legfontosabb jellemzője, hogy a keverőtárcsák forognak. Minden egyes betű sifrírozása után az első tárcsa $1/26$-nyival elfordul (26 betűs ábécé esetén). A második tárcsa csak akkor fordul $1/26$-nyit, ha az első tárcsa megtett egy teljes fordulatot, a harmadik tárcsa akkor fordul $1/26$-nyit, ha a második tárcsa megtett egy teljes fordulatot, miközben az első tárcsa már 26×26-szor fordult $1/26$-nyit. Ez a mechanizmus hasonlít az autók kilométerórájához. A rotáció révén a gép többábécés helyettesítéses eljárás megvalósítására használható. A három keverőtárcsa kezdeti beállítása $26 \times 26 \times 26 = 17\,576$ különböző kulcsnak felel meg. Az ábrán az *Enigma* kétdimenziós ábrázolása látható, az áttekinthetőség kedvéért hatbetűs ábécé esetén. A keverőtárcsa egybetűnyi elfordulása során a tárcsákat összekötő vezetékek egy hellyel lentebb kerülnek.



10**. ábra Enigma kapcsolási rajza**

A kapcsolási rajzon még két szerkezeti elem is látható. A visszairányító szintén egy belső huzalozású gumitárcsa, de nem forog. Mikor a kezelő begépel egy betűt, azzal egy elektromos jelet küld át a három keverőtárcsán. A visszairányító ezt a beérkező jelet küldi vissza, de más útvonalon. Az ábrán látható tárcsaállások esetén a leütött b betű a C-t villantja fel, ha azonban a c-t ütöttük volna le, akkor a kijelzőn a B villant volna fel. Ebből látható a visszairányító szerepe: a gép a nyílt szöveg egyik betűjét a rejtjeles szöveg egyik betűjévé alakítja, és ha egy másik gép ugyanígy van beállítva,

[1] Az *Enigma*-ra vonatkozó részt **Tóthné Mészáros Ágnes Rejtjelezés a középiskolában** című szakdolgozatából vettem át.

akkor az előbb megkapott rejtjeles szöveg betűjét leütve megkapjuk az eredeti nyílt szöveg betűjét, vagyis a sifrírozáshoz és a desifrírozáshoz ugyanaz a gép szükséges megegyező kezdő beállítással!

A másik új elem az ábrán a kapcsolótábla, amely a billentyűzet és az első keverőtárcsa közé van iktatva. E kapcsolótábla lehetővé teszi, hogy beiktassunk néhány vezetéket, amelyek még az első keverőtárcsába való belépés előtt felcserélnek bizonyos betűket. Az *Enigma* kezelőjének hat ilyen vezetéke van, miáltal hat betűpárt tud felcserélni a huszonhatból. Egy 26-betűs ábécé esetenkénti hat betűpárjának felcserélési lehetőségeinek száma

$$\frac{\prod_{k=0}^{5}\binom{26-2k}{2}}{6!} = 100\,391\,791\,500.$$

A gép alapbeállításához tartozik még a keverőtárcsák sorrendje is. Scherbius úgy szerkesztette meg gépét, hogy a keverőtárcsák sorrendjét meg lehessen változtatni, a keverőtárcsák kivehetőségével. A három tárcsa hatféleképpen helyezhető a gépbe, így a lehetséges kezdőbeállítások, vagyis kulcsok száma:

keverőtárcsák beállítása (minden tárcsa 26-féle pozícióba állítható):	17 576
keverőtárcsák sorrendje:	6
kapcsolótábla beállításai:	100 391 791 500
összesen (előző három tényező szorzata):	10 586 916 764 424 000

A rejtjelezés kulcsát (a gép kezdőbeállítását) naponta változtatták, amit a négyhetente szétosztott 28 kulcsot tartalmazó kódkönyv határozott meg. A kapcsolótábla eredményezi a kulcsok számának legnagyobb növekedését, de a sifrírozás megkezdése után már nem változik beállítása, így egyedüli alkalmazása olyan rejtjeles szöveget generálna, amely gyakorisági elemzéssel megfejthető. A keverőtárcsák kevesebb számú kulccsal járulnak hozzá a végeredményhez, de beállításuk folyamatosan változik, aminek eredményeképpen a rejtjeles szöveg gyakorisági elemzéssel nem fejthető meg. Mivel a rejtjelezés során a gép 17 576 különböző sifre-ábécét használ, Babbage módszerével sem fejthető meg a rejtjeles szöveg.

7. Nyilvános kulcsú rejtjelezés

Amint a 8. oldalon írtuk, megfelelő körülmények biztosítása esetén, nevezetesen, ha a titkosításra használt kulcsból gyakorlatilag nem lehet meghatározni a fejtéshez szükséges kulcsot, a sifrírozó kulcs akár nyilvános is lehet, aminek az a fontos következménye van, hogy nincs szükség kulcscserére. További előny, hogy csökken a szükséges kulcsok száma, hiszen egy résztvevőnek csupán egyetlen kulcspárra van szüksége, szemben a klasszikus módszerekkel, ahol különböző titok-csoportoknak különböző kulcspárral kell rendelkezniük, így egy résztvevőhöz annyi eltérő kulcspár tartozik, ahány különböző csoportnak a tagja.

Nyilvános kulcsú rendszerben mind az üzenettér, mind a kulcstér mérete nagy kell, hogy legyen. Amennyiben ugyanis adott egy c rejtjelszöveg, és kicsi az üzenettér, akkor minden egyes $\tilde{m} \in \mathcal{M}$-re kiszámíthatjuk a mindenki számára elérhető k nyilvános kulccsal $\tilde{c} = E_k(\tilde{m})$-t, és ha egy $\tilde{m}$-ra $\tilde{c} = c$, akkor az injektivitásnak köszönhetően $m = \tilde{m}$. Másrészt, ha a kulcstér kicsi, akkor egymás után kipróbálva c-re az összes lehetséges $\widetilde{k'}$ fejtőkulcsot, lesz közöttük olyan (például a valódi k' titkos kulcs), amellyel $D_{\widetilde{k'}}(c) = \tilde{m}$ értelmes szöveg, és elegendően hosszú szöveg esetén ez csak egyetlen kulccsal, k'-vel lehetséges.

A nyilvános kulcsú titkosítás alapja tehát, hogy m-ből $c = E_k(m)$ meghatározása egyszerű feladat, és hasonlóan egyszerű a titkos kulcs ismeretében $m = D_{k'}(c)$ kiszámítása is, ám k' ismeretének hiányában gyakorlatilag legyen lehetetlen c-ből m visszanyerése. Itt persze felmerül, hogy mit tekintünk „egyszerű”-nek, és mire mondjuk azt, hogy „gyakorlatilag lehetetlen”. Tekintettel arra, hogy a rejtjelezésben algoritmusokat használunk, kézenfekvőnek tűnik az algoritmusok bonyolultságát szokásosan mérő fogalmakat használni, így egyszerűnek tekintjük azokat a feladatokat, amelyekre létezik polinomiális algoritmus, és nehéznek, gyakorlatilag megoldhatatlannak azokat, amelyek nem polinomiálisak. Sajnos, ezek a mértékek nem teljesen alkalmasak a kriptográfiai algoritmusok minősítésére. Ennek több oka is van:

1. a bonyolultságelmélet elsősorban egy problémacsalád legrosszabb, esetenként az átlagos esetének összetettségét méri, míg a kriptográfiában éppen a könnyű eseteknek kell a kivételeknek lenniük;
2. a bonyolultságelmélet mérőszámai aszimptotikus értékek, vagyis azt mérik, hogy mennyire bonyolult egy feladat, ha a bemeneti adat nagysága tart a végtelenhez, míg a kriptográfiában az adatok, bár esetleg viszonylag nagyok, de mindig végesek és többnyire fix hosszúságúak;
3. Brassard megmutatta, hogy bizonyos ésszerű feltételek mellett egy kriptográfiai probléma nem lehet NP-nehéz, hacsak nem igaz, hogy $NP = co\text{-}NP$.

A fent említett problémák ellenére fontos megnézni egy adott kriptográfiai algoritmusról, hogy milyen bonyolultságú a szokásos hierarchia szerint, hiszen ha egy ilyen teszten könnyűnek találtatik, akkor szinte biztosan nem alkalmas nyilvános kulcsú rejtjelezésre. A fordítottja ennek nem mindig igaz. A hátizsák-probléma NP-teljes, és erre alapozva már 1978-ban megjelent egy titkosító algoritmus (Ralph Merkle and Martin Hellman: Hiding Information and Signatures in Trapdoor Knapsacks, *IEEE Trans. Information Theory*, 24(5), September 1978, pp525–530), de igen hamar kiderült, hogy ez a rendszer törhető (Adi Shamir: A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. CRYPTO 1982, pp279–288).

A nyilvános kulcsú titkosítás biztonságával kapcsolatban több fogalom is születetett. Csupán az említés szintjén ismertetünk kettőt:

- **polinomiálisan biztos** egy rendszer, ha várható értékben polinomiális időben egy passzív támadó nem képes két olyan különböző nyílt szöveget találni, amelyek titkosított párját $\frac{1}{2}$-nél lényegesen nagyobb valószínűséggel képes helyesen megkülönböztetni;
- **szemantikailag biztos**nak mondunk egy rendszert, ha a nyílt szövegek tetszőleges eloszlása mellett egy passzív támadó mindazt meg tudja határozni várható polinomiális időben egy

adott nyílt szövegről a rejtjelszöveg nélkül, amit a rejtjel ismeretében ugyanilyen időkeretben ki tud nyerni.

Megmutatható, hogy egy kriptorendszer pontosan akkor polinomiálisan biztos, ha szemantikailag biztos. Tulajdonképpen a szemantikai biztonság a tökéletes biztonság egy polinomiálisan korlátozott változata.

A nyilvános kulcsú rendszerek fontos fogalma az **egyirányú függvény** (*one-way function*). $f: X \to Y$ egyirányú, ha bármely $x \in X$-re $f(x)$ könnyen kiszámítható, de néhány kivételtől eltekintve egy adott $y \in Y$-ból gyakorlatilag lehetetlen egy olyan $x \in X$ meghatározása, amelyre $y = f(x)$. A néhány kivétel például azt jelentheti, hogy egyes $x \in X$-re kiszámolva $y = f(x)$-et, az ilyen y-okból könnyen meg tudjuk határozni a megfelelő $x \in X$-et. A nyilvános kulcsú rendszerekben érthető módon ilyen függvényeket kell használni a titkosításhoz. Ez azonban nem minden egyirányú függvény esetén jelent megoldást, hiszen így a legális címzett sem jut hozzá a rejtjelből a nyílt szöveghez. Vannak azonban olyan egyirányú függvények, amelyek egy paraméter segítségével könnyen invertálhatóak. Az ilyen függvényeket **csapda típusú egyirányú függvény**nek vagy **csapóajtó-függvény**nek (*trapdoor function*) nevezik. Ilyenre példa, mint majd hamarosan belátjuk, a moduláris hatványozás (lásd az RSA-nál).

Úgy tűnik, hogy a csapóajtó-függvények segítségével megoldottuk a nyilvános kulcsú rejtjelezés problémáját: a függvénnyel könnyű a nyílt szöveg rejtjelpárjának meghatározása, a titkos kulcs mint paraméter segítségével a legális fejtő könnyen tud fejteni, ám a fejtő kulcs ismerete nélkül gyakorlatilag reménytelen egy titkosított üzenetből visszanyerni az eredeti üzenetet. A dolog gyakorlatilag ilyen szép, ám elméletileg nem ennyire rózsás a helyzet, ugyanis a jelenleg használt csapóajtó-függvények egyikéről sincs bizonyítva, hogy valóban nehéz az invertálásuk az adott paraméter ismerete nélkül, és általában, az alkalmazott egyirányú függvények egyikéről sem tudjuk bizonyítottan, hogy valóban nehéz a fordított irányú számításuk (sem a csapda típusúakról, sem a többiről). Mindazonáltal, a jelenleg alkalmazott egyirányú függvények (legalábbis a publikus adatok ismeretében) a jelenben teljesítik az elvárásokat.

8. RSA

Az RSA a leggyakrabban alkalmazott és a legjobban bevált nyilvános kulcsú rejtjelezési algoritmus, amelyet sokan és igen alaposan vizsgáltak, és amely a publikus információk alapján gyakorlatilag fejthetetlen, ha a paramétereket a megfelelő gondossággal választják. Az algoritmus neve az őt kifejlesztő három matematikus: **R**ivest, **S**hamir és **A**dleman nevének kezdőbetűje.

Az a és a $b \neq 0$ valós szám esetén legyen $a \bmod b = a - b\left\lfloor\frac{a}{b}\right\rfloor$. Könnyen látható, hogy ha a egész és b pozitív egész szám, akkor $a \bmod b \equiv a\ (b)$ és $0 \leq a \bmod b < b$, vagyis ekkor $a \bmod b$ az a-nak b-vel való osztásakor keletkező nem negatív maradéka. Ha a 1-nél nagyobb, b pozitív és c tetszőleges valós szám, akkor $a^{-\infty}$-t nullának, $\frac{b}{0}$-t és $\left\lfloor\frac{b}{0}\right\rfloor$-t ∞-nek, végül $c - \infty$-t $-\infty$-nek tekintjük.

Használni fogjuk $n \in \mathbb{N}^+$-ra az $M^{(n)} = \{m \in \mathbb{N} | m < n\}$ jelölést. Nyilvánvalóan látszik, hogy minden $n \in \mathbb{N}^+$-ra $\left|M^{(n)}\right| = n$.

8.1. *Definíció*

Legyen $2 < p_1^{(A)} < p_2^{(A)}$ prímszám, $n^{(A)} = p_1^{(A)} p_2^{(A)}$, $e^{(A)}$ a $\varphi\left(n^{(A)}\right)$-hoz relatív prím pozitív egész, és $d^{(A)}$ az $e^{(A)}x \equiv 1\ \left(\varphi\left(n^{(A)}\right)\right)$ kongruencia tetszőleges pozitív megoldása. Ekkor $\left(n^{(A)}, e^{(A)}\right)$ az A nyilvános kulcsa, $d^{(A)}$ a titkos kulcsa, $M^{(A)} = M^{\left(n^{(A)}\right)} = C^{(A)}$ az A nyílt illetve rejtjeles szövegeinek halmaza, és az $m \in M^{(A)}$ nyílt szöveg rejtjeles párja $c = E_{e^{(A)}}(m) = m^{e^{(A)}} \bmod n^{(A)}$.

Δ

A továbbiakban az általános esetek vizsgálatánál a paraméterek birtokosát jelölő A-t elhagyjuk.

Maga a rejtjelezés könnyű feladat, polinomiális időben végrehajtható. Valóban: mivel a hatványozást csupán modulo n végezzük, ezért minden lépésben két, legfeljebb $b = \lfloor \log_r n \rfloor + 1$ hosszúságú számot szorzunk (r a számrendszer alapszáma), aminek az időigénye b^2 nagyságrendű, és ilyen szorzásból legfeljebb $2t$-re van szükség, ahol $t = \lfloor \log_2 e \rfloor$, amint az alábbi tétel mutatja.

8.2. *Tétel* (*gyorshatványozás*)

Legyen $1 < n \in \mathbb{N}$, $m \in \mathbb{Z}$ és $\sum_{i=0}^{t-1} u_i 2^i = u \in \mathbb{N}^+$, ahol $t = \lfloor \log_2 u \rfloor + 1$ és $t > i \in \mathbb{N}$-re $u_i \in \{0,1\}$. Ekkor az $m^{(t-1)} = m$, $t - 1 > i \in \mathbb{N}$: $m^{(i)} = m^{u_i}\left(m^{(i+1)}\right)^2 \bmod n$ algoritmus végén $m^{(0)} = m^u \bmod n$, és a szorzások s számára $t - 1 \leq s \leq 2(t-1)$.

Δ

Bizonyítás:

Könnyű igazolni, hogy $a(b \bmod n) = ab \bmod n$, ha a, b és n egész számok, így elég belátni, hogy ha $P^{(t-1)} = m$ és a $t - 1 > i \in \mathbb{N}$ indexekre $P^{(i)} = m^{u_i}\left(P^{(i+1)}\right)^2$, akkor $P^{(0)} = m^u$.

Ha $t = \lfloor \log_2 u \rfloor + 1$, akkor $2^{t-1} \leq u < 2^t$, vagyis u felíráshoz pontosan t bit kell, és $u_{t-1} = 1$. Most $P^{(t-1)} = m = m^1 = m^{u_{t-1}} = m^{\sum_{j=t-1}^{t-1} u_j 2^{j-(t-1)}}$, és ha $P^{(i+1)} = m^{\sum_{j=i+1}^{t-1} u_j 2^{j-(i+1)}}$ valamilyen $t - 1 > i \in \mathbb{N}$ indexre, akkor

$$P^{(i)} = m^{u_i}\left(P^{(i+1)}\right)^2 = m^{u_i}\left(m^{\sum_{j=i+1}^{t-1} u_j 2^{j-(i+1)}}\right)^2 = m^{u_i} m^{\sum_{j=i+1}^{t-1} u_j 2^{j-i}} = m^{\sum_{j=i}^{t-1} u_j 2^{j-i}},$$

innen pedig $i = 0$ esetén kapjuk, hogy $P^{(0)} = m^{\sum_{j=0}^{t-1} u_j 2^j} = m^u$.

Az algoritmus – t bit esetén – $t-1$ lépésből áll (leszámítva az inicializálást, amely egyszerű értékadás). Minden lépés tartalmaz egy négyzetre emelést, amely egy szorzás, ez összesen $t-1$ szorzás. u_i értéke 0 vagy 1; az első esetben $m^{u_i}=1$, tehát a négyzetre emeléssel megkaptuk $m^{(i)}$ értékét, míg $u_i=1$ esetén $m^{u_i}=m$, vagyis $\left(m^{(i+1)}\right)^2$-et még meg kell szorozni m-mel, így az ilyen szorzások száma legfeljebb $t-1$.

□

8.3. Megjegyzés

A bizonyításból látszik, hogy az algoritmus nem csak a moduláris, de a közönséges hatványozásra is hasonlóan működik, vagyis a tétel és a bizonyítás jelöléseivel $m^u=P^{(0)}$, és az eredmény most is legalább $t-1$ és legfeljebb $2(t-1)$ szorzással megkapható. Van azonban egy lényeges különbség a két hatványozás között. Legyen n valamilyen számrendszerben r-jegyű. Míg a moduláris hatványozás esetén minden lépésben n-nél kisebb, vagyis legfeljebb r-jegyű számokat kell szorozni, addig a közönséges hatványozásnál (nem nulla alap esetén) minden lépésben a négyzetre emelésnél megduplázódik a jegyek száma (vagy legfeljebb ennél eggyel kisebb lesz). Mivel a szorzáshoz nagyjából a tényezők jegyei számának szorzatával megegyező számú lépésre (egy-egy számjegy szorzására) van szükség, ezért most minden fordulóban hozzávetőleg négyszer több elemi számításra van szükség, mint az előző fordulóban. Ha m jegyeinek száma b, akkor tehát a moduláris hatványozásnál nagyságrendileg tb^2 elemi szorzás (tehát jegyenkénti szorzás) szükséges, míg a közönséges hatványozás esetén az ilyen lépések száma hozzávetőleg $\sum_{l=0}^{t-2}\left(2^l b\right)^2 = b^2\frac{4^{t-1}-1}{4-1}\sim 4^t b^2$, vagyis az előbbi esetben az elvégzendő műveletek száma t-nek polinomiális, a második esetben viszont exponenciális függvénye. Másként szólva, míg a moduláris hatványozás polinomiális időben elvégezhető, addig a közönséges hatványozás nem polinomiális bonyolultságú algoritmus.

Δ

Ahhoz, hogy a 8.1. definícióban valóban rejtjelezést adtunk meg, meg kell mutatni, hogy az $m \mapsto m^e \bmod n$ leképezés injektív $M^{(n)}$-en, a fejtés a titkos információ hiányában gyakorlatilag lehetetlen, de d birtokában könnyű végrehajtani. Ami a támadót illeti, neki egy $x^e \equiv c\ (n)$ kongruenciát kell megoldania. Jelenleg az egyetlen járható út (általában) az, ha c-nek vesszük a d-edik hatványát, mert amint a következő tételből kiderül, $m=c^d \bmod n$. Ám ehhez ismerni kellene d-t, és ehhez általában $\varphi(n)$-et, amit viszont csak akkor tudunk könnyen számítani, ha adott az n faktorizációja. Az utóbbi problémára – mármint adott szám felbontása prímtényezőinek szorzatára – nem ismeretes polinomiális idejű algoritmus, sőt, az tűnik valószínűnek, hogy ilyet nem is lehet megadni. Felhívjuk a figyelmet rá, hogy nem állítottuk, hogy a visszafejtés csupán így történhet, ezért nem mondtuk, hogy a fejtés nehézsége azonos a faktorizálás nehézségével; ezt sem nem bizonyították, sem nem cáfolták eddig (nyilvánosan!), továbbá azt sem mondtuk, hogy minden esetben a hatványozás a legkézenfekvőbb megoldás, bizonyos szerencsétlen (m,c) pár esetén egyszerűbben is megoldható a feladat (a szerencsétlen jelző a legális partnerek szempontjából értendő, a hívatlan támadó számára ez inkább szerencsés véletlen). Amit állíthatunk, az csupán annyi, hogy az RSA fejtése legfeljebb annyira nehéz, mint az összetett egész számok tényezőkre bontása, hiszen ha fel tudjuk n-et bontani, akkor már fejteni is tudunk, de elvileg elképzelhető, hogy van ennél egyszerűbb módja is a fejtésnek. Egyébként n felbontásának vagy $\varphi(n)$-nek az ismerete algoritmikus szempontból egyenértékű, mert egyikből a másik polinomiálisan számítható. Ez a felbontás ismeretében nyilvánvaló, hiszen $\varphi(n)=(p_1-1)(p_2-1)$, ami lényegében véve egyetlen szorzás. $p_1p_2-p_1-p_2+1=(p_1-1)(p_2-1)=\varphi(n)$ és $p_1p_2=n$, az előbbiből $p_1+p_2=n-\varphi(n)+1$, így $\varphi(n)$ ismeretében p_1 és p_2 az $x^2-(n-\varphi(n)+1)+n$ polinom két gyöke, és a két gyök polinomiális időben meghatározható. d ismeretében viszont m könnyen nyerhető, mert a moduláris hatványozásról már megmutattuk, hogy könnyű feladat, így a legális címzett könnyen hozzájut a nyílt szöveghez. Most megmutatjuk, hogy tetszőleges $m \in M^{(n)}$ nyílt üzenetre $(m^e)^d \bmod n = m$, ebből majd az is következik, hogy a rejtjelszabályunk injektív. Az alábbiakban általánosabban vizsgáljuk a kérdést.

8.4. Jelölés

Legyen $n \in \mathbb{N}^+$, $u \in \mathbb{N}$ és $u < v \in \mathbb{N}$. Ekkor

- $M_u^{(n)} = \{m \in M^{(n)} | m^u \bmod n = 0\}$ és $N_u^{(n)} = |M_u^{(n)}|$;
- $M_{v,u}^{(n)} = \{m \in M^{(n)} | (m^v - m^u) \bmod n = 0\}$ és $N_{v,u}^{(n)} = |M_{v,u}^{(n)}|$.

Δ

Láthatóan $M_u^{(n)}$ az x^u és $M_{v,u}^{(n)}$ az $x^v - x^u$ polinom n-nél kisebb nem negatív modulo n gyökeinek halmaza, és $N_u^{(n)}$ valamint $N_{v,u}^{(n)}$ az ilyen gyökök száma.

8.5. Tétel

Legyen $n = \prod_{i=1}^{s} p_i^{r_i}$ páratlan egész, ahol $s \in \mathbb{N}^+$, minden $s \geq i \in \mathbb{N}^+$ indexre r_i pozitív egész és a p_i prímek páronként különbözőek, és legyen $2 \nmid p$ prím és $r \in \mathbb{N}^+$. Ekkor $N_u^{(n)} = \prod_{i=1}^{s} N_u^{\left(p_i^{r_i}\right)}$ és $N_{v,u}^{(n)} = \prod_{i=1}^{s} N_{v,u}^{\left(p_i^{r_i}\right)}$, ahol $N_{v,u}^{(p^r)} = N_u^{(p^r)} + N_{v-u,0}^{(p^r)}$, $N_u^{(p^r)} = p^{r-\left\lceil \frac{r}{u} \right\rceil}$ és $N_{w,0}^{(p^r)} = \left(w, \varphi(p^r)\right)$.

Δ

Bizonyítás:

Ha $f \in \mathbb{Z}[x]$, akkor a fenti n-re f modulo n gyökeinek száma a modulo $p_i^{r_i}$ gyökei számának szorzata, ezért elegendő megmutatni, hogy $n = p^r$-rel $N_{v,u}^{(p^r)}$ éppen a tételben álló kifejezés.

$m^v - m^u \equiv 0\ (p^r)$ az m egésszel a kongruencia definíciója szerint akkor és csak akkor, ha a modulus osztója a két oldal különbségének, azaz $p^r | m^v - m^u = m^u(m^{v-u} - 1)$. De tetszőleges m egészre m^u és $m^{v-u} - 1$ relatív prímek. Ha ugyanis a p prím osztója m^u-nak, akkor $u > 0$ és p osztója m-nek, de akkor osztója m^{v-u}-nak, és így biztosan nem osztója $m^{v-u} - 1$-nek. Ekkor viszont p^r csak úgy osztja az $m^v - m^u$ különbséget, ha osztója m^u és $m^{v-u} - 1$ egyikének és csak egyikének. Ez azt mutatja, hogy $N_{v,u}^{(p^r)} = N_u^{(p^r)} + N_{v-u,0}^{(p^r)}$, így elegendő a jobb oldalon álló két értéket meghatározni, továbbá csupán a páronként inkongruens megoldások érdekesek, ezért a keresett számot úgy nyerjük, ha külön-külön meghatározzuk azon $p^r > m \in \mathbb{N}^+$ egészek számát, amelyekre m^u illetve $m^{v-u} - 1$ osztható p^r-rel, és a két számot összeadjuk.

Nézzük az előbbi oszthatóságot. $u = 0$ esetén $m^u = 1$, és mivel $r > 0$, tehát $p^r > 1$, így nincs olyan egész, amelyre m^u osztható p^r-rel, $N_0^{(p^r)} = 0$, és ugyanez az értéke $p^{r-\left\lceil \frac{r}{0} \right\rceil}$-nak is. Vizsgáljuk az $u > 0$ esetet. Ha p^r osztója m^u-nak, akkor ennek a prímfelbontásában p legalább az r-edik hatványon szerepel, és így m-ben p kitevője nem lehet kisebb $\frac{r}{u}$-nál, vagyis $\left\lceil \frac{r}{u} \right\rceil$-nél, mert ez a kitevő bizonyos tényezők száma, tehát egész szám. Ugyanez visszafelé is érvényes: ha m osztható $p^{\left\lceil \frac{r}{u} \right\rceil}$-val, akkor m^u is osztható p^r -rel, vagyis m^u akkor és csak akkor osztható p^r-rel, ha $m = kp^{\left\lceil \frac{r}{u} \right\rceil}$ alakú alkalmas k egésszel. Minket a páronként inkongruens megoldások érdekelnek, tehát például azok, amelyek egyben kielégítik a $p^r > m \in \mathbb{N}$ feltételt, ami ekvivalens a $p^{r-\left\lceil \frac{r}{u} \right\rceil} > k \in \mathbb{N}$ egyenlőtlenséggel, és ezen k egészek száma éppen a bal oldalon álló szám; ezzel megkaptuk $N_u^{(p^r)}$-t $u > 0$-ra is.

Most áttérünk $N_{w,0}^{(p^r)}$, vagyis az $x^w - 1$ polinom modulo p^r gyökei számának meghatározására, ahol a $w = v - u$ jelölést alkalmaztuk. Ennek nyilván csak olyan m lehet a megoldása, amellyel m^w és p^r legnagyobb közös osztója egyben osztója 1-nek, vagyis ez a legnagyobb közös osztó 1. $(ab, c) = 1$ pontosan akkor igaz, ha $(a, c) = 1$ és $(b, c) = 1$, így $(m^w, p^r) = 1 \Leftrightarrow (m, p^r) = 1$, tehát az $x^w - 1$ polinom bármely modulo p^r gyöke relatív prím a modulushoz.

Ha p páratlan, akkor a p^r modulusra létezik primitív gyök, vagyis olyan egész, amelynek a rendje modulo p^r pontosan $\varphi(p^r)$. Legyen g primitív gyök a p^r modulusra. Ez azt jelenti, hogy g $\varphi(p^r) > k \in \mathbb{N}$-kitevős hatványainak halmaza egyrétegűen lefed egy modulo p^r redukált maradék-

rendszert, így az előbb megadott intervallumba eső olyan k egész számokat kell megkeresni, amelyekkel $g^{kw} = \left(g^k\right)^w \equiv 1 = g^0\ (p^r)$. g primitív gyök a p^r modulusra, tehát a rendje ezzel a modulussal $\varphi(p^r)$, így a kongruencia megoldásai azok a k egészek, amelyekkel $kw \equiv 0\ \left(\varphi(p^r)\right)$ teljesül, azaz $k \equiv 0\ \left(\frac{\varphi(p^r)}{(w,\varphi(p^r))}\right)$, és ennek a kongruenciának $\left(w, \varphi(p^r)\right)$ páronként inkongruens megoldása van.

□

8.6. Kiegészítés

Tetszőleges p prímszám esetén $N_{w,0}^{(p^r)} = \left(\varepsilon w, \varphi(p^r)\right)$, ahol $\varepsilon = 1$, kivéve, ha w páros és $p = 2$ és $r \geq 3$, amikor $\varepsilon = 2$.

Δ

Bizonyítás:

Ha p páratlan, akkor $\varepsilon = 1$, és visszakapjuk a tételben megadott eredményt. Ha $p = 2$ és r legfelejebb 2, akkor létezik primitív gyök, és ekkor a bizonyítás megegyezik a páratlan prímek esetével. Maradt a $p = 2, r \geq 3$ eset. Ekkor van olyan g, hogy ennek $2^{r-2} > k \in \mathbb{N}$-kitevős hatványai és ezek ellentettjei kiadnak egy $\bmod 2^r$ redukált maradékrendszert, és a két csoport idegen. Ha w páratlan, akkor a két csoport elemeinek hatványai az eredeti csoporthoz tartoznak, míg páros w esetén g^{-k} és g^k w-edik hatványa azonos lesz, és mivel $g^0 = 1$, így elegendő g^{kw}-k között keresni azokat, amelyek $\bmod 2^r$ kongruensek 1-gyel, és páros w esetén ezek számát megduplázni. $g^{kw} \equiv 1 = g^0\ (2^r)$ akkor és csak akkor, ha $kw \equiv 0\ (2^{r-2})$, hiszen g rendje $\bmod 2^r$ pontosan 2^{r-2}. Ez utóbbi kongruencia ekvivalens a $k \equiv 0\ \left(\frac{2^{r-2}}{(w,2^{r-2})}\right)$ kongruenciával, és ennek $(w, 2^{r-2})$ páronként inkongruens megoldása van. Ha w páratlan, akkor $r \geq 3$ miatt ez a szám 1, ami megegyezik $(w, 2^{r-1})$-val, míg ha w páros, akkor $2(w, 2^{r-2}) = (w, 2^{r-1})$. De pozitív egész r esetén $\varphi(2^r) = 2^{r-1}$, így a bizonyítás kész.

□

A tételnek egy sereg következménye van.

8.7. Következmény

Legyen $t = \text{lkkt}\left(\varphi\left(p_i^{r_i}\right) \middle| s \geq i \in \mathbb{N}^+\right)$. Ekkor az előző tétel jelöléseivel és feltételeivel

1. $0 \leq N_u^{(n)} = \prod_{i=1}^{s} p_i^{r_i - \left\lceil \frac{r_i}{u} \right\rceil} \leq \prod_{i=1}^{s} p_i^{r_i - 1} = \frac{n}{P}$, ahol $P = \prod_{i=1}^{s} p_i$;
   a) $N_u^{(n)} = 0$ ekvivalens $u = 0$-val;
   b) 1.-ben a jobb oldalon akkor és csak akkor áll egyenlőség, ha $u \geq \max\{r_i | s \geq i \in \mathbb{N}^+\}$;
   c) 1.-ben a bal oldal értéke pontosan akkor 1, ha $u = 1$ vagy n négyzetmentes;
2. $1 \leq N_{v,0}^{(n)} = \prod_{i=1}^{s} \left(v, \varphi\left(p_i^{r_i}\right)\right) \leq \varphi(n)$;
   a) $N_{v,0}^{(n)} = 1$ akkor és csak akkor, ha $\left(v, \varphi(n)\right) = 1$;
   b) $N_{v,0}^{(n)} = \varphi(n)$ akkor és csak akkor, ha $t | v$;
   c) ha $v = \varphi(n)$, akkor $N_{v,0}^{(n)} = \varphi(n)$ (ez az **Euler-Fermat tétel** más megfogalmazásban);
3. $u \geq 1$ esetén $2^s \leq N_{v,u}^{(n)} = \prod_{i=1}^{s} \left(p_i^{r_i - \left\lceil \frac{r_i}{u} \right\rceil} + \left(v - u, p_i^{r_i} - p_i^{r_i - 1}\right) \right) \leq n$;
   a) $N_{v,u}^{(n)} = 2^s$ akkor és csak akkor teljesül, ha egyrészt $u = 1$ vagy n négyzetmentes, másrészt $\left(v - u, \varphi(n)\right) = 1$;
   b) $N_{v,u}^{(n)} = n$ akkor és csak akkor, ha $u \geq \max\{r_i | s \geq i \in \mathbb{N}^+\}$ és $t | v - u$;
   c) ha $v - u$ páros, akkor $N_{v,u}^{(n)} \geq 3^s$;
4. $N_{n,n-\varphi(n)}^{(n)} = n$;

5. $N_{v,1}^{(n)} = \prod_{i=1}^{s}\left(1 + \left(v-1, p_i^{r_i} - p_i^{r_i-1}\right)\right)$;
   a) $N_{v,1}^{(n)} = 2^s$ akkor és csak akkor, ha $v-1$ és $\varphi(n)$ relatív prím;
   b) ha v páratlan, akkor $N_{v,u}^{(n)} \geq 3^s$;
   c) $N_{v,1}^{(n)} = n$-hez szükséges és elégséges, hogy n négyzetmentes, és t osztója $v-1$-nek;
   d) ha n négyzetmentes, akkor minden $m \in \mathbb{Z}$-re $m^{1+k\varphi(n)} \equiv m\ (n)$, ahol $k \in \mathbb{N}$;
   e) ha minden $m \in \mathbb{Z}$-re $m^n \equiv m\ (n)$, akkor n négyzetmentes, és vagy $s = 1$, vagy $s \geq 3$;
6. ha n négyzetmentes, $k \in \mathbb{N}$, és $e \in \mathbb{N}$ a kt-hez relatív prím, akkor tetszőleges, az e-hez relatív prím $j \in \mathbb{N}$-re bármely, az $ex \equiv 1\ (jt)$ kongruenciát kielégítő $d \in \mathbb{N}$-nel minden $m \in \mathbb{Z}$-re $(m^e)^d \equiv m\ (n)$. Speciálisan, ha $\left(e, \varphi(n)\right) = 1$ és $ed \equiv 1\ \left(\varphi(n)\right)$, akkor $(m^e)^d \equiv m\ (n)$;
7. legyen $z \in \mathbb{N}^+$, és $z \geq i \in \mathbb{N}^+$-ra $1 < u_i \in \mathbb{N}$ olyan, hogy $n = \prod_{i=1}^{s} u_i \in \mathbb{N}$ négyzetmentes, legyen továbbá $t' = \prod_{i=1}^{s}(u_i - 1)$, e a t'-höz relatív prím pozitív egész, és d olyan természetes szám, amellyel $ed \equiv 1\ (t')$. Ekkor
   a) minden $m \in \mathbb{Z}$-re $(m^e)^d \equiv m\ (n)$ akkor és csak akkor teljesül, ha $t | ed - 1$;
   b) és minden, a t'-höz relatív prím e-re ez pontosan akkor igaz, ha $t | t'$.

Δ

Bizonyítás:

1. Nézzük $N_u^{(p^r)}$ értékét. p prímszám, így $p > 1$, tehát p^z z-nek szigorúan monoton növekvő függvénye. $N_u^{(p^r)} = p^{r-\left\lceil\frac{r}{u}\right\rceil}$, azaz $z = r - \left\lceil\frac{r}{u}\right\rceil$, ahol $u \geq 0$. Pozitív u esetén ez monoton nő, és mivel r is pozitív, ezért $u \geq r$ esetén $0 < \frac{r}{u} \leq 1$, tehát $\max\left\{r - \left\lceil\frac{r}{u}\right\rceil\right\} = r - 1$. Innen rögtön adódik, hogy $N_u^{(n)}$ maximális értéke $\prod_{i=1}^{s} p_i^{r_i-1}$, amit akkor érünk el, amikor minden i-re $u \geq r_i$. A másik oldalon, tehát amikor u csökken, akkor $\frac{r}{u}$ tart a pozitív végtelenhez, így a kitevő a negatív végtelenhez, és a hatvány a nullához. Végül legyen u pozitív, tehát $u \geq 1$. Ekkor $\frac{r}{u} \leq r$, vagyis $\left\lceil\frac{r}{u}\right\rceil \leq r$, így $r - \left\lceil\frac{r}{u}\right\rceil \geq 0$, és a hatvány értéke legalább 1. A hatvány értéke akkor és csak akkor 1, ha a kitevő 0, és a kitevőben nullát pontosan akkor kapunk, ha r és $\left\lceil\frac{r}{u}\right\rceil$ azonos. De bármely x valós számra $\lceil x\rceil - 1 < x \leq \lceil x\rceil$, így a kitevő $r - 1 < \frac{r}{u} \leq r$ esetén, és csak ekkor lesz 0. Mivel $u \geq 1$, ezért a jobb oldali feltétel biztosan teljesül, elég a másikat nézni. $r - 1 < \frac{r}{u}$ akkor és csak akkor, ha $(u-1)(r-1) < 1$. Mivel a bal oldalon mindkét tényező nem negatív egész szám, ezért az egyenlőtlenség pontosan akkor igaz, ha valamelyikük nulla, azaz akkor és csak akkor, ha u és r legalább egyike 1. Azt kaptuk tehát, hogy $N_u^{(n)}$ értéke akkor és csak akkor 1, ha $u = 1$, vagy ha minden i-re $r_i = 1$, ami azt jelenti, hogy n négyzetmentes.

2. $N_{v,0}^{(p^r)} = \left(v, \varphi(p^r)\right)$. Két szám legnagyobb közös osztója akkor és csak akkor 0, ha mindkét szám nulla, minden más esetben pozitív egész szám. Mivel $\varphi(p^r)$ biztosan nem nulla, ezért $N_{v,0}^{(p^r)}$ értéke is legalább 1. Ha $\left(v, \varphi(p^r)\right) = 1$, akkor $N_{v,0}^{(p^r)} = 1$, és ilyen v, p és r létezik. Ennek minden i-re teljesülnie kell, tehát a legkisebb értéket akkor és csak akkor kapjuk, amikor v relatív prím a $\varphi\left(p_i^{r_i}\right)$-k mindegyikéhez, vagyis a szorzatukhoz, ami éppen $\varphi(n)$.

A másik oldalon egyenlőség pontosan akkor lesz, ha $\varphi(p^r) | v$. $N_{v,0}^{(n)}$ maximumát akkor kapjuk, ha ez minden i-re teljesül, tehát ha a $\varphi\left(p_i^{r_i}\right)$-k legkisebb közös többszöröse osztja v-t.

A speciális eset is igaz, hiszen adott számok legkisebb közös többszöröse osztója a szorzatuk bármely többszörösének.

3. Ha $u > 0$, akkor $N_{v,u}^{(p^r)} = N_u^{(p^r)} + N_{v-u,0}^{(p^r)}$ és $N_{v,u}^{(n)} = \prod_{i=1}^{s} N_{v,u}^{\left(p_i^{r_i}\right)}$. Minden tényező mindkét tagjának minimuma 1, így az összeg minimuma 2^s, amit pontosan akkor kapunk, ha n négyzetmentes vagy $u = 1$, és ugyanakkor $v - u$ relatív prím $\varphi(n)$-hez. Azonban n páratlan, így $\varphi\left(p_i^{r_i}\right)$ minden i-re páros, ezért ha $v - u$ is páros, akkor a legnagyobb közös osztó minden tényezőben legalább kettő, és maga a tényező minimálisan 3.

A maximumot pontosan akkor kapjuk, ha $t|v-u$ és $u \geq \max\{r_i | s \geq i \in \mathbb{N}^+\}$. Ekkor minden i-re $\left(v-u, \varphi\left(p_i^{r_i}\right)\right) = \varphi\left(p_i^{r_i}\right) = p_i^{r_i} - p_i^{r_i-1}$ és $p_i^{r_i-\left\lceil \frac{r_i}{u} \right\rceil} = p_i^{r_i-1}$, ezek összege $p_i^{r_i}$, amelyek szorzata n.

4. Most $v-u = n-\left(n-\varphi(n)\right) = \varphi(n)$, és $t|\varphi(n)$, így t osztója $v-u$-nak. $s \geq i \in \mathbb{N}^+$-ra n felírható $p_i^{r_i} n_i$ alakban, ahol $p_i^{r_i}$ és n_i relatív prímek. Ekkor

$$u = n-\varphi(n) = p_i^{r_i} n_i - \varphi\left(p_i^{r_i}\right)\varphi(n_i) = p_i^{r_i} n_i - p_i^{r_i-1}(p_i-1)\varphi(n_i)$$
$$= p_i^{r_i}\left(n_i - \varphi(n_i)\right) + p_i^{r_i-1}\varphi(n_i) \geq p_i^{r_i-1} \geq r_i \geq r_i,$$

és ebből kapjuk, hogy $u \geq \max\{r_i | s \geq i \in \mathbb{N}^+\}$.

5. Ez a pont az utolsó alpont kivételével lényegében véve 3. speciális és $u=1$-re aktualizált esete, ahol az aktualizálás azt jelenti, hogy az $u \geq r_i$ feltételek következtében most n négyzetmentes. Nézzük e)-t. Amennyiben minden egész m-re $m^n \equiv m\ (n)$, akkor $N_{n,1}^{(n)} = n$, ami akkor és csak akkor teljesül, ha n négyzetmentes és $t|n-1$. Legyen n négyzetmentes. Ha $s=1$, akkor ez azt jelenti, hogy n prím, és ekkor $m^n \equiv m\ (n)$ éppen a kis Fermat-tétel, vagyis ekkor ez minden egész m-re igaz. Legyen most $s=2$, vagyis $n=pq$, ahol p és q két különböző páratlan prím, és mondjuk $p<q$. Ekkor $n-1 = pq-1 = p(q-1)+(p-1)$, és ez biztosan nem osztható $q-1$-gyel, de akkor még kevésbé $p-1$ és $q-1$ legkisebb közös többszörösével, t-vel.

6. Ha $(e,kt)=1$, akkor $(e,t)=1$ is igaz, és mivel j is relatív prím e-hez, ezért $(e,jt)=1$, így $ex \equiv 1\ (jt)$ megoldható. Ha d megoldás, akkor $t|jt|ed-1$, tehát 5. alapján igaz az állítás.

Ami a speciális esetet illeti, t nyilván osztója $\varphi(n)$-nek, tehát $\varphi(n)=kt$, és a felírás alapján $j=k$, így $1=\left(e,\varphi(n)\right)=(e,kt)$-ből $(e,j)=(e,k)=1$.

7. Az első állítás $ed=v$-vel 5. alapján igaz.

$t|t'$ esetén $ed \equiv 1\ (t')$-ből $t|d-1$. Fordítva, tegyük fel, hogy tetszőleges e, d pár jó, és legyen $ed=1+kt'$, ahol $(k,t)=1$. Ekkor viszont $t|ed-1=kt' \Leftrightarrow t|t'$, tehát igaz a második állítás is.

□

A páros n-re vonatkozó megállapításokat ismét külön fogalmaztuk meg.

8.8. Kiegészítés

Ha n páros is lehet, akkor az előbbi következmény egyes pontjai az alábbi módon változnak.

3.

c) ha $v-u$ páros, akkor ha n páratlan vagy néggyel osztható, akkor az alsó határ legalább 3^s, míg ha n egy páratlan szám kétszerese, akkor $N_{v,u}^{(n)} \geq 2 \cdot 3^{s-1}$;

5.

b) ha v páratlan, akkor ha n páratlan vagy néggyel osztható, akkor $N_{v,1}^{(n)} \geq 3^s$, míg ha n páros, de nem osztható néggyel, akkor $N_{v,1}^{(n)} \geq 2 \cdot 3^{s-1}$;

e) ha $\forall(m \in \mathbb{Z}): m^n \equiv m\ (n)$, akkor n négyzetmentes, és vagy s=1, vagy $2 \nmid n$ és $s \geq 3$.

Δ

Bizonyítás:

3.c) Ha n páros, akkor $n = 2^l n_1$ alakú egy páratlan n_1-gyel és pozitív egész l-lel, és ekkor $\varphi(2^l) = 2^{l-1}$. Ha n néggyel osztható, akkor $l \geq 2$ és $\varphi(2^l)$ páros, tehát a helyzet hasonló a páratlan prímhatványokéhoz. Amennyiben viszont $l=1$, akkor $\varphi(2^l)=1$, és $\left(v-u, \varphi(2)\right)=1$.

5.b) Ez az előző pont $u=1$ esetén.

5.e) Itt csak annyit kell belátni, hogy ha n nem prímszám, akkor szükségszerűen páratlan. Ha $s \geq 2$, akkor n prímosztói között van páratlan, és így a $\varphi(p_i)$-k között páros szám, ezért t páros. Ugyanakkor, ha n páros, akkor $n-1$ páratlan, és így nem lehet osztható t-vel, de akkor $N_{n,1}^{(n)} < n$.

□

8.9. Megjegyzés

Ha n páratlan, négyzetmentes egész szám legalább három különböző prímosztóval, akkor van olyan n, amelyre $N_{n,1}^{(n)} = n$. $n = 561 = 3 \cdot 11 \cdot 17$ a legkisebb, ekkor $t = [3-1,11-1,17-1] = 80$, és 80 osztója $560 = 561 - 1$-nek. Egy összetett n egész szám az a poztív egészre nézve **álprím**, ha $a^n \equiv a\ (n)$. Amennyiben egy adott n egész bármely egész számra vonatkozóan álprím, akkor n **Carmichael-szám**. A fenti 5.e) pont alapján n csak úgy lehet Carmichael-szám, ha páratlan, négyzetmentes és legalább három különböző prímosztója van, és az előbbi példa alapján létezik Carmichael-szám. Egy nem túl régi eredmény alapján végtelen sok Carmichael-szám létezik.

Δ

A 6. következmény mutatja, hogy d ismeretében a legális fejtő valóban könnyen hozzájut a nyílt szöveghez. Azt is látjuk, hogy amennyiben $n = pq$-ban p vagy q nem prím (és mindkettő nagyobb 1-nél), és legalább egyikük nem négyzetmentes, akkor biztosan van olyan nyílt üzenet, amelyet rejtjelezve nem a helyes eredményt kapjuk visszafejtéskor. Ha n mindkét tényezője négyzetmentes, de legalább egyikük összetett, továbbá e és d olyan egészek, hogy $ed \equiv 1\ \big((p-1)(q-1)\big)$ (ahol $(p-1)(q-1)$ a vélt $\varphi(n)$), tehát valamilyen k nem negatív egésszel $ed - 1 = k(p-1)(q-1)$, úgy akkor és csak akkor nem keletkezik hiba visszafejtéskor, ha $ed-1$ osztható a valódi p_i, páronként különböző prímosztóból számított $[p_i - 1 | s' \geq i \in \mathbb{N}]$ legkisebb közös többszörössel, ahol s' a faktorok száma (n páratlan, tehát a p_i-k is azok).

Végül az is kiolvasható a következményekből, hogy egy $n = pq$, e paraméterekkel megadott RSA rejtjelnek $\big(1 + (e-1, p-1)\big)\big(1 + (e-1, q-1)\big)$ **fixpont**ja van. Mivel a fixpont nem kívánatos (hiszen ekkor nem rejtjeleztünk), ezért az a jó, ha ez a szám minél kisebb. Ennek minimuma 9 (mert n páratlan, tehát e is az kell, hogy legyen), és ezt akkor érjük el, ha $(e-1, t) = 2$.

Most megmutatjuk, hogy RSA esetén tetszőleges m nyílt üzenetre $(m^e)^d \equiv m\ (n)$, ebből majd az is következik, hogy a rejtjelszabályunk injektív.

8.10. Tétel

Legyen $n \in \mathbb{N}$ páratlan egész, f az $M^{(n)}$ halmaz önmagába való olyan leképezése, hogy minden $M^{(n)}$-beli m-re $f: m \mapsto m^e \bmod n$, ahol e 1-nél nagyobb egész. f akkor és csak akkor injektív (és így bijektív), ha n négyzetmentes, és e relatív prím $\varphi(n)$-hez.

Δ

Bizonyítás:

Ha $\big(e, \varphi(n)\big) = 1$, akkor van olyan d pozitív egész, hogy $ed \equiv 1\ \big(\varphi(n)\big)$, és ha még n négyzetmentes, akkor ezzel a d-vel az $x^{ed} \equiv x\ (n)$ kongruencia megoldásainak száma az 5.d) következmény alapján n, vagyis ekkor a $h: m \mapsto m^{ed} \bmod n = (m^e \bmod n)^d \bmod n$ leképezés az $M^{(n)}$ önmagába való identikus leképezése, tehát h bijektív. Mivel h az egyaránt az $M^{(n)}$-t $M^{(n)}$-be képező $f: m \mapsto m^e \bmod n$ és $g: m \mapsto m^d \bmod n$ leképezések kompozíciója, ahol előbb f-et hajtjuk végre, ezért h csak úgy lehet bijektív, ha f injektív, így a tétel feltételei elégségesek.

Ha n nem négyzetmentes, akkor $e > 1$ miatt az 1.c) következmény szerint az $x^e \equiv 0\ (n)$ kongruenciának, ha viszont e nem relatív prím $\varphi(n)$-hez, akkor az $x^e \equiv 1\ (n)$ kongruenciának van a 2.a) következmény alapján egynél több megoldása, így az f leképezés egyik esetben sem injektív.

□

Bár az RSA szempontjából nem játszik közvetlen szerepet, ám a prímtesztelésnél fontos kérdés az $x^v + 1$ polinom modulo n gyökeinek száma. Mivel erős a hasonlóság a már megoldott $x^v - 1$ polinom modulo n gyökeinek problémájával, ezért ezt a kérdést is megvizsgáljuk, majd csupán a teljesség kedvéért az $x^v + x^u$ polinom modulo n gyökeinek számát is megnézzük.

8.11. Tétel

Legyen $s \in \mathbb{N}^+$, $n = \prod_{i=1}^{s} p_i^{r_i}$, ahol r_i pozitív egész, a p_i-k páronként különböző páratlan prímek, $\varphi\left(p_i^{r_i}\right) = 2^{k_i} q_i$ a pozitív egész k_i-vel és páratlan egész q_i-vel, továbbá $v = 2^k q$ pozitív egész a $k \in \mathbb{N}$, $2 \nmid q$ egészekkel. Ekkor az $x^v + 1$ polinom modulo n gyökeinek száma $2^{ks} \prod_{i=1}^{s} (q, q_i)$, ha $k < \min\{k_i | s \geq i \in \mathbb{N}^+\}$, egyébként 0.

Δ

Bizonyítás:

Már láttuk korábban, hogy elegendő prímhatványokra meghatározni a megoldásszámot, és ezeket összeszorozni. Azt is láttuk, hogy páratlan prím esetén a prímhatványra van primitív gyök, mondjuk g. -1 relatív prím p^r-hez, ezért egy és csak egyféleképpen írható g egy $d = \varphi(p^r) > i \in \mathbb{N}$-kitevős hatványaként. $g^d \equiv 1\ (p^r)$, és $(-1)^2 = 1 \equiv 1\ (p^r)$, továbbá d páros, így $-1 \equiv g^{\frac{d}{2}}\ (p^r)$. a^v csak úgy lehet kongruens -1-gyel, ha a relatív prím p^r-hez, és ekkor a is felírható g hatványaként, vagyis $a \equiv g^y\ (p^r)$. A megoldandó kongruencia ezek után $vy \equiv \frac{d}{2}\ (d)$. Ha $(v, d)\left|\frac{d}{2}\right.$, és csak ekkor, a kongruencia megoldható. Ekkor nyilván igaz, hogy (q, q') osztója $\frac{d}{2}$-nek, így még annak kell teljesülnie, hogy a legnagyobb közös osztóban fellépő 2-hatvánnyal is lehessen osztani $\frac{d}{2}$-t. Ez pontosan akkor teljesül, ha v-ben a 2 kitevője kisebb, mint d-ben. Ekkor a megoldások száma $(v, d) = \left(2^k q, 2^{k'} q'\right) = 2^k\left(q, 2^{k'-k} q'\right) = 2^k (q, q')$, ahol $p^r = 2^{k'} q'$ a páratlan q'-vel, hiszen q páratlan.

Visszatérve az eredeti modulusra, pontosan akkor van megoldása a megadott kongruenciának, ha $k < \min\{k_i | s \geq i \in \mathbb{N}^+\}$, és ebben az esetben a megoldások száma $2^{ks} \prod_{i=1}^{s} (q, q_i)$, hiszen 2^k minden tényezőben szerepel.

□

Az általános esetről szól a következő tétel.

8.12. Tétel

Legyen p prímszám, $r \in \mathbb{N}^+$, $u \in \mathbb{N}$ és $u < v \in \mathbb{N}$. Ekkor $x^v + x^u$ modulo p^r gyökeinek $T_{v,u}^{(p^r)}$ száma $u = v$ esetén

1. 2, ha $p = 2$ és $r = 1$;
2. $2N_u^{(2^r)}$, ha $p = 2$ és $r > 1$;
3. $N_u^{(p^r)}$, ha $p > 2$;

míg ha $v > u$, $\varphi(p^r) = 2^{k'} q'$ és $v - u = 2^k q$, ahol q és q' páratlan egészek, akkor

4. $N_u^{(2^r)} + 1$, ha $p = 2$ és $r = 1$;
5. $N_u^{(2^r)}$, ha $p = 2$, $r > 1$ és $v - u$ páros;
6. $N_u^{(2^r)} + 1$, ha $p = 2$, $r > 1$ és $v - u$ páratlan;
7. $N_u^{(p^r)}$, ha $p > 2$ és $k \geq k'$;
8. $N_u^{(p^r)} + 2^k (q, q')$, ha $p > 2$ és $k < k'$.

Δ

Bizonyítás:

Legyen először $u = v$, ekkor $x^v + x^u = 2x^u$. Ha p páratlan, akkor p^r akkor és csak akkor osztója $2a^u$-nak, ha osztója a^u-nak, így kapjuk 3.-at. Ha $p = 2$ és $r = 1$, akkor az osztó 2, és mivel $2a^u$ mindig páros, ezért az oszthatóság minden a egészre teljesül, és ezek között két inkongruens van modulo 2, ami igazolja 1.-et. Amennyiben viszont $p = 2$ és $r > 1$, 2^r akkor és csak akkor lesz osztója $2a^u$-nak, ha 2^{r-1} osztja a^u-t, ilyen a a

$2^{r-1} > a \in \mathbb{N}$ tartományban $N_u^{(2^{r-1})}$ van, és akkor ezek kétszerese is osztható 2^r-rel, és még ezek is kisebbek 2^r-nél, ezért igaz 2. is.

Most nézzük a $v > u$ eseteket. Ekkor $x^v + x^u = x^u(x^{v-u} + 1)$, és $v - u > 0$ következtében minden a egészre a^u és $a^{v-u} + 1$ relatív prím, ezért most is elegendő külön meghatározni az x^u és az $x^{v-u} + 1$ polinom modulo p^r gyökeinek számát, az eredeti probléma megoldását ezen két szám összege adja. Az első kongruencia megoldásainak számát már ismerjük, ez $N_u^{(p^r)}$, ezért csak az $x^w + 1$ alakú kifejezéssel kell foglalkoznunk, ahol $w = v - u > 0$. Páratlan prím esetére a kérdést az előző tételben megoldottuk, és éppen a 7.-ben és 8.-ban megfogalmazott eredményt kaptuk.

Most legyen $p = 2$. Ha $r = 1$, akkor olyan a-t keresünk, amelyre $a^w + 1$ osztható 2-vel, és $2 > a \in \mathbb{N}$. Ilyen a pontosan egy van, nevezetesen $a = 1$, és ezzel kész a 4. pont. Hátra van az $r > 1$ eset. Ha $r = 2$, akkor az előzőhöz hasonlóan az $a^w \equiv -1\ (4)$ feltételnek megfelelő, 4-nél kisebb, nem negatív egész a-kat keressük. Ehhez megint az kell, hogy a legyen relatív prím 4-hez. Ilyen a kettő van, $a = 1$ és $a = 3 \equiv -1\ (4)$. Innen látszik, hogy ha w páros, akkor nincs megoldás, míg ha w páratlan, akkor pontosan egy megoldás lesz, tehát most teljesül 5. és 6. Végül legyen $r \geq 3$. Ekkor $a^w \equiv -1\ (2^r)$-hez szükséges, hogy $a^w \equiv -1\ (4)$ is teljesüljön, így rögtön kapjuk, hogy páros w esetén most sincs megoldás. Amennyiben w páratlan, akkor $(-a)^w = -a^w$, így vizsgálhatjuk, hogy mikor osztható $b^w - 1$ 2^r-rel. Ez csak páratlan b-vel lehet, így nem fordulhat elő, hogy $b = -b$, ezért a b-k száma azonos lesz az eredeti kongruencia megoldásainak számával. Ez $N_{w,0}^{(2^r)}$, és ennek az értéke $\left(w, \varphi(2^r)\right) = (w, 2^{r-1}) = 1$, mert w páratlan és $r > 2$, amivel $r > 2$-re is igazoltuk 5.-öt és 6.-ot.

□

Az előbbi tételek ismeretében megmutatjuk, hogy ha a nyilvános (n, e) kulcs ismeretében polinomiálisan meghatározható a titkos d kulcs, akkor a faktorizálás is végrehajtható polinomiális időben.

Legyen s 1-nél nagyobb egész szám, $s \geq i \in \mathbb{N}^+$-ra r_i pozitív egész, a p_i-k páronként különböző páratlan prímszámok és $\varphi\left(p_i^{r_i}\right) = 2^{u_i} v_i$ a $2 \nmid v_i$ és u_i pozitív egésszel, $\hat{u} = \min\{u_i | s \geq i \in \mathbb{N}^+\}$, továbbá $n = \prod_{i=1}^s p_i^{r_i}$ és $\varphi(n) = 2^u v$, ahol u és $2 \nmid v$ pozitív egész. Ekkor $2^u v = \varphi(n) = \prod_{i=1}^s p_i^{r_i} = \prod_{i=1}^s 2^{u_i} v_i = 2^{\sum_{i=1}^s u_i} \prod_{i=1}^s v_i$, vagyis $u = \sum_{i=1}^s u_i$ és $v = \prod_{i=1}^s v_i$, ebből következően $\hat{u} \leq u_i < u$ és $v_i | v$, tehát $\min\{u, u_i\} = u_i$ és $(v, v_i) = v_i$. Legyen S azon, n-nél kisebb, az n-hez relatív prím nem negatív a egészek száma, amelyekre vagy $a^v \equiv 1\ (n)$, vagy van olyan, u-nál kisebb nem negatív egész i, amellyel $a^{2^i v} \equiv -1\ (n)$. Az első feltételnek megfelelő egészek száma $N_{v,0}^{(n)}$, míg adott nem negatív egész i-re $x^{2^i v} \equiv -1\ (n)$ megoldásainak száma $T_{2^i v,0}^{(n)}$. $S = N_{v,0}^{(n)} + \sum_{i=0}^{u-1} T_{2^i v,0}^{(n)}$, hiszen bármely, az n-hez relatív prím a egészre $1 \equiv a^{\varphi(n)} \equiv a^{2^u v}\ (n)$, másrészt, ha $a^v \equiv 1\ (n)$, akkor minden j-re, míg ha $a^{2^i v} \equiv -1\ (n)$, akkor minden, az i-nél nagyobb j-re teljesül, hogy $a^{2^j v} \equiv 1\ (n)$. A korábbi eredményeket figyelembe véve így azt kapjuk, hogy

$$S = N_{v,0}^{(n)} + \sum_{i=0}^{u-1} T_{2^j v,0}^{(n)} = \prod_{i=1}^{s} \left(v, \varphi\left(p_i^{r_i}\right)\right) + \sum_{j=0}^{\hat{u}-1} 2^{sj} \prod_{i=1}^{s} (v, v_i)$$
$$= \prod_{i=1}^{s} (v, 2^{u_i} v_i) + \sum_{j=0}^{\hat{u}-1} 2^{sj} \prod_{i=1}^{s} (v, v_i) = \prod_{i=1}^{s} v_i + \sum_{j=0}^{\hat{u}-1} 2^{sj} \prod_{i=1}^{s} v_i$$
$$= \left(1 + \sum_{j=0}^{\hat{u}-1} 2^{sj}\right) v = \left(1 + \frac{2^{s\hat{u}} - 1}{2^s - 1}\right) v < \left(1 + \frac{2^{s\hat{u}}}{2}\right) v = \left(1 + 2^{s\hat{u}-1}\right) v,$$

vagyis $S \leq 2^{s\hat{u}-1} v \leq 2^{\sum_{i=1}^s u_i - 1} v = 2^{u-1} v = \frac{\varphi(n)}{2}$, tehát a lehetséges a-knak legalább a fele olyan, hogy valamilyen i-re $a^{2^i v} \not\equiv -1\ (n)$, de $a^{2^{i+1} v} \equiv 1\ (n)$.

Az előbbi eredményből következik, hogy ha valaki meg tudja határozni d-t, akkor képes n-et faktorizálni. Tudjuk ugyanis, hogy $ed - 1 \equiv 0\ \left(\varphi(n)\right)$, így $a^{ed-1} \equiv 1\ (n)$ bármely, az n-hez relatív prím a-val. Legyen most $ed - 1 = 2^u v$, ekkor tehát annak a valószínűsége, hogy valamilyen nem negatív egész i-re $a^{2^i v} \not\equiv \pm 1\ (n)$, de $a^{2^{i+1} v} \equiv 1\ (n)$, legalább $\frac{1}{2}$, vagyis várhatóan legfeljebb két kísérlettel találunk ilyen a-t. Ezzel az a-val $n \left| a^{2^{i+1} v} - 1 = \left(a^{2^i v} - 1\right)\left(a^{2^i v} + 1\right)\right.$, és a jobb oldali szorzat egyik tényezője sem osztható n-nel, ami csak úgy lehet, ha $1 < \left(a^{2^i v} - 1, n\right) < n$, és így a legnagyobb közös osztó az n valamelyik faktora. Ha ismerjük d-t,

akkor a fenti eljárás minden része polinomiális, így az egész eljárás is az, vagyis ekkor polinomiális időben tudjuk felbontani n-et.

A fentiekből már következik, hogy ha a nyilvános adatokból polinomiális időben meghatározható $\varphi(n)$, akkor n is polinomiális időben faktorizálható, hiszen $\varphi(n)$ és e ismeretében polinomiális időben tudjuk d-t kiszámolni. Ezt az eredményt korábban, a 60. oldalon közvetlenül is beláttuk.

Most olyan fejtési módszert vizsgálunk, amelyhez nem kell ismerni a d titkos paramétert, és megnézzük, hogyan lehet ez ellen a támadás ellen védekezni. Az eljárás csak nyilvános adatokat alkalmaz, és ismételt hatványozással állítja elő a nyílt üzenetet. Szükségünk lesz az alábbi tételre.

8.13. Tétel

Ha u és v pozitív egész, és $u|v$, akkor $\varphi(u)|\varphi(v)$.

Δ

Bizonyítás:

Legyen $s \in \mathbb{N}^+$, $s \geq i \in \mathbb{N}^+$-ra és $i > j \in \mathbb{N}^+$-ra $r_i \in \mathbb{N}^+$ és $p_i \neq p_j$ prímek, és $u = \prod_{i=1}^{s} p_i^{r_i}$. Mivel $u|v$, ezért $v = v_1 v_2 = v_2 \prod_{i=1}^{s} p_i^{t_i}$ úgy, hogy $(u, v_2) = (v_1, v_2) = 1$, és valamennyi t_i az r_i-nél nem kisebb egész. Most

$$\varphi(v) = \varphi(v_1)\varphi(v_2) = \varphi(v_2)\prod_{i=1}^{s} p_i^{t_i-1}(p_i - 1)$$
$$= \varphi(v_2)\prod_{i=1}^{s} p_i^{r_i-1}(p_i - 1)\prod_{i=1}^{s} p_i^{t_i-r_i} = \varphi(u)\varphi(v_2)\prod_{i=1}^{s} p_i^{t_i-r_i},$$

így valóban igaz, hogy $\varphi(u)|\varphi(v)$.

□

Nézzük meg, hogy adott $1 < n \in \mathbb{N}$, $1 < e \in \mathbb{N}$, $c \in M^{(n)}$ esetén mikor lesz olyan $k \in \mathbb{N}^+$, amellyel $c^{e^{k-1}} \bmod n = m$, ha $m \in M^{(n)}$-re $c = m^e \bmod n$. Rögtön látjuk, hogy ha az előbb megadott feltételek teljesülnek, akkor $c = m^e \bmod n = \left(c^{e^{k-1}} \bmod n\right)^e \bmod n = c^{e^k} \bmod n$, vagyis ekkor $n \left| c^{e^k} - c = c\left(c^{e^k-1} - 1\right)\right.$, ami viszont pontosan akkor igaz, ha $\frac{n}{(c,n)} \left| c^{e^k-1} - 1\right.$. $\frac{n}{(c,n)} = o_n^+(c)$, és az előbbi oszthatóság $c^l \equiv 1 \left(o_n^+(c)\right)$, ahol $l = e^k - 1$. A kongruenciának akkor és csak akkor van megoldása, ha $1 = \left(c, o_n^+(c)\right) = \left(c, \frac{n}{(c,n)}\right)$, ami viszont akkor és csak akkor teljesül, ha a c bármely p prímosztója c-ben legalább akkora hatványon fordul elő, mint n-ben. Ez biztosan így van, ha n négyzetmentes. Ekkor $c^l \equiv 1 \left(o_n^+(c)\right)$-hez szükséges és elégséges, hogy $o_{o_n^+(c)}(c) \left| l = e^k - 1\right.$, vagy ismét átírva kongruenciába, ha $e^k \equiv 1 \left(o_{o_n^+(c)}(c)\right)$. Ilyen k pontosan akkor van, ha e relatív prím $o_{o_n^+(c)}(c)$-hez. De $o_{o_n^+(c)}(c) \left| \varphi\left(o_n^+(c)\right) = \varphi\left(\frac{n}{(c,n)}\right)\right| \varphi(n)$, így, ha $\left(e, \varphi(n)\right) = 1$, akkor van ilyen k, és a legkisebb ilyen k pozitív egész éppen $o_{o_{o_n^+(c)}(c)}(e)$. Ha tehát n négyzetmentes és e relatív prím $\varphi(n)$-hez, akkor $M^{(n)}$ egy c elemére a $k_c = o_{o_{o_n^+(c)}(c)}(e)$ pozitív egész számmal $c^{e^{k_c-1}} \bmod n = m$, és ha k a k_c-k legkisebb közös többszöröse, akkor valamennyi $c \in M^{(n)}$-re $c^{e^{k-1}} \bmod n = m$, és k a legkisebb ilyen tulajdonságú pozitív egész szám.

$o_u(v)$ osztója $\varphi(u)$- nak, $o_u^+(v)$ pedig u-nak, így felhasználva az előző eredményeket

$$o_{o_{o_n^+(c)}(c)}(e) \left| \varphi\left(o_{o_n^+(c)}(c)\right)\right| \varphi\left(\varphi\left(o_n^+(c)\right)\right)\left| \varphi\left(\varphi(n)\right)\right.$$

minden c-re, így $k_c | k \left| \varphi\left(\varphi(n)\right) \right.$, tehát, ha azt akarjuk, hogy k_c a lehető legtöbb c-re nagy legyen, akkor n-et úgy kell választani, hogy $\varphi\left(\varphi(n)\right)$-nek kevés kis osztója legyen, és a kis osztókkal csak kevés c-t lehessen fejteni. Természetesen mindig lesz olyan c, amely kis kitevővel fejthető, hiszen az RSA-nak vannak fixpontjai, és ezek már $k = 1$-gyel fejthetőek. Az lenne a jó, ha a fixpontok száma minél kisebb lenne, és minden más rejtjelezett szövegből csak nagy k' kitevővel lehetne visszanyerni az eredeti üzenetet.

Az előbb megfogalmazott gondolatokat pontosítjuk a következőkben.

8.14. Tétel

Legyen $1 < e \in \mathbb{N}$, $1 < n = \prod_{i=1}^{s} p_i^{r_i}$ és $m \in M^{(n)}$-re $c = m^e \bmod n$. Akkor és csak akkor létezik olyan $k \in \mathbb{N}^+$, hogy minden $m \in M^{(n)}$-re $c^{e^{k-1}} \bmod n = m$, ha $f: m \mapsto m^e \bmod n$ injektív az $M^{(n)}$ halmazon, és ekkor $o = o_t(e)$ a legkisebb ilyen k kitevő, ahol $t = \text{lkkt}\left(\varphi\left(p_i^{r_i}\right) \middle| s \geq i \in \mathbb{N}^+\right)$.

Δ

Bizonyítás:

Legyen most is $n = \prod_{i=1}^{s} p_i^{r_i}$, ahol $s \in \mathbb{N}^+$, a p_i-k páronként különböző prímszámok és az r_i-k pozitív egészek. Egy $c \in M^{(n)}$-re $c^{e^k} \bmod n = c$ pontosan akkor teljesül, ha $c \in M_{e^k,1}^{(n)}$, és az ilyen c-k száma $N_{e^k,1}^{(n)} = \prod_{i=1}^{s} \left(1 + \left(e^k - 1, \varphi\left(p_i^{r_i}\right)\right)\right)$. Minden lehetséges c-re akkor és csak akkor teljesül az adott e-vel és k-val, hogy $c^{e^k} \bmod n = c$, ha $N_{e^k,1}^{(n)} = n$, ami viszont ekvivalens azzal, hogy n négyzetmentes és minden i-re $p_i - 1 \left| e^k - 1 \right.$. Az utóbbi feltétel akkor és csak akkor teljesíthető, ha e relatív prím valamennyi $\varphi(p_i) = p_i - 1$-hez, azaz $\varphi(n)$-hez. Ha viszont ez a két feltétel teljesül, akkor az $a \mapsto a^e \bmod n$ leképezés injektív $M^{(n)}$-en, és

$$\left(c^{e^{k-1}} \bmod n\right)^e \bmod n = c^{e^k} \bmod n = c = m^e \bmod n,$$

vagyis $m = c^{e^{k-1}} \bmod n$.

Minden i-re $p_i - 1 \left| e^k - 1 \right.$ akkor és csak akkor igaz, ha a $\varphi(p_i) = p_i - 1$-ek legkisebb közös többszöröse, t is osztója $e^k - 1$-nek, vagyis $e^k \equiv 1\ (t)$, és a legkisebb ilyen k kitevő $o = o_t(e)$.

□

Nézzük, hogyan kell n-et választani, hogy a lehető legtöbb c-re az iterációs fejtés nehéz legyen.

Legyen $u \in \mathbb{N}$, $u < v \in \mathbb{N}$, $u_1 \in \mathbb{N}$, $u_2 \in \mathbb{N}$, $u_1 < v_1 \in \mathbb{N}$ és $u_2 < v_2 \in \mathbb{N}$. Ha $u_1 \leq u_2$ és $m \in M_{u_1}^{(n)}$, akkor $n | m^{u_1} | m^{u_2}$, azaz $m \in M_{u_2}^{(n)}$, tehát $M_{u_1}^{(n)} \subseteq M_{u_2}^{(n)}$, és ekkor $N_{u_1}^{(n)} \leq N_{u_2}^{(n)}$. Ha $u_1 \leq u_2$ mellett $v_1 - u_1 | v_2 - u_2$, és $m \in M_{v_1,u_1}^{(n)}$ akkor $n | m^{v_1} - m^{u_1} | m^{u_1}(m^{v_1 - u_1} - 1) | m^{u_2}(m^{v_2 - u_2} - 1) = m^{v_2} - m^{u_2}$, vagyis $m \in M_{v_2,u_2}^{(n)}$, és így $M_{v_1,u_1}^{(n)} \subseteq M_{v_2,u_2}^{(n)}$, valamint $N_{v_1,u_1}^{(n)} \leq N_{v_2,u_2}^{(n)}$. Ha most $M_{v_1,u_1}^{(n)} \subseteq M_{v_2,u_2}^{(n)}$ mellett még $N_{v_1,u_1}^{(n)} = N_{v_2,u_2}^{(n)}$, akkor azt kapjuk, hogy $M_{v_1,u_1}^{(n)} = M_{v_2,u_2}^{(n)}$, vagyis ilyen esetben a kitevők növelésével nem kapunk újabb modulo n gyököket az $x^v - x^u$ alakú polinomokhoz.

A fentebbiekben említett iterációs fejtési lehetőség akkor alkalmazható a gyakorlatban, ha vagy maga $o = o_t(e)$ értéke kicsi, vagy az üzenetek nagy része kis kitevővel fejthető. Egy biztonságos rendszerben tehát o értéke olyan nagy, hogy gyakorlatilag lehetetlen az ilyen iterációs fejtés, és az o-nál kisebb kitevőkkel fejthető üzenetek aránya kicsi. A legalább 3^s fixpont már $k = 1$-gyel fejthető. Ha $o_{p_i - 1}(e) < o$, ami normális esetben minden i-re igaz, akkor természetesen legalább $3^{s-1} p_i$ számú üzenet már $o_{p_i - 1}(e)$ kitevővel fejthető, így az elvárásunk csak az lehet, hogy a fixpontokon kívül ennél kisebb kitevővel ne lehessen fejteni.

Nagy o akkor érhető el, ha minden i-re $o_{p_i - 1}(e)$ a lehető legnagyobb, és $\left(o_{p_i - 1}(e), o_{p_j - 1}(e)\right)$ minden $i \neq j$-re a lehető legkisebb. $o_{p_i - 1}(e) \left| \varphi(p_i - 1) \right.$, így $o_{p_i - 1}(e)$ lehetséges legnagyobb értéke $\varphi(p_i - 1)$, és ilyen e akkor és csak akkor létezik, ha $p_i - 1$ értéke 2, 4 vagy $2 p_i^{(1)^{r_i}}$ (mert $p_i - 1$ páros), ahol $p_i^{(1)}$ páratlan prímszám

és $r_i \in \mathbb{N}^+$, vagyis RSA esetén pontosan akkor, ha $p_i - 1 = 2{p_i^{(1)}}^{r_i}$, hiszen a kis prím faktorok könnyen meghatározhatóak. Legyen tehát $p_i - 1 = 2{p_i^{(1)}}^{r_i}$, ekkor $\varphi(p_i - 1) = 2{p_i^{(1)}}^{r_i-1}\left(p_i^{(1)} - 1\right)$, és legyen e egy modulo $(p_i - 1)$ primitív gyök, vagyis $(e, p_i - 1) = 1$ és $o_{p_i-1}(e) = \varphi(p_i - 1)$. Ha $(e, p_i - 1) = 1$, akkor egyúttal $\left(e, {p_i^{(1)}}^{l}\right) = 1$ is teljesül valamennyi $r_i > l \in \mathbb{N}$ kitevő esetén. Ekkor $e^{\varphi\left(2{p_i^{(1)}}^{r_i-1}\right)} \equiv 1 \ \left(2{p_i^{(1)}}^{r_i-1}\right)$, tehát

$$2{p_i^{(1)}}^{r_i-1} \left| \left(e^{\varphi\left(2{p_i^{(1)}}^{r_i-1}\right)} - 1, 2{p_i^{(1)}}^{r_i}\right)\right.,$$

továbbá

$$\left(e^{\varphi\left(2{p_i^{(1)}}^{r_i-1}\right)} - 1, 2{p_i^{(1)}}^{r_i}\right) < 2{p_i^{(1)}}^{r_i},$$

mert $o_{p_i-1}(e) = \varphi\left(2{p_i^{(1)}}^{r_i}\right)$, és végül

$$\left(e^{\varphi\left(2{p_i^{(1)}}^{r_i-1}\right)} - 1, 2{p_i^{(1)}}^{r_i}\right) \left| 2{p_i^{(1)}}^{r_i}.\right.$$

A három összefüggés alapján $\left(e^{\varphi\left(2{p_i^{(1)}}^{r_i-1}\right)} - 1, 2{p_i^{(1)}}^{r_i}\right) < 2{p_i^{(1)}}^{r_i-1}$, és így $r_i > 1$ esetén nem teljesül, hogy csak a fixpontok fejthetőek $o_{p_i-1}(e)$-nél kisebb kitevővel. Legyen ezért minden $s \geq i \in \mathbb{N}^+$-ra $r_i = 1$, vagyis az n minden prímfaktorára legyen $p_i = 2p_i^{(1)} + 1$, ahol $p_i^{(1)}$ páratlan prímszám. Ekkor bármely pozitív egész k-val

$$(e^k - 1, p_i - 1) = \begin{cases} 2 \\ p_i - 1. \end{cases}$$

$e - 1 | e^k - 1$, ezért $M_{e,1}^{(p_i)} \subseteq M_{e^k,1}^{(p_i)}$, és ha $N_{e^k,1}^{(p_i)} = 2$, akkor $M_{e,1}^{(p_i)} = M_{e^k,1}^{(p_i)}$, vagyis a p_i-k ilyen választásával csupán a fixpontoknak megfelelő üzenetek fejthetőek kis kitevővel.

e rendje modulo $(p_i - 1)$ akkor és csak akkor o_i, ha $e^{o_i} \equiv 1 \ (p_i - 1)$ és $e^{\frac{o_i}{p}} \not\equiv 1 \ (p_i - 1)$ az o_i valamennyi p prímosztójára, és $o_i \left| \varphi(p_i - 1) = \varphi\left(p_i^{(1)}\right) = p_i^{(1)} - 1\right.$. A modulo $(p_i - 1)$ primitív gyököknek a száma $\varphi\left(p_i^{(1)} - 1\right)$. $\varphi\left(p_i^{(1)} - 1\right) \leq \frac{p_i^{(1)}-1}{2}$, hiszen $p_i^{(1)} - 1$ páros szám, és $\varphi\left(p_i^{(1)} - 1\right) < \frac{p_i^{(1)}-1}{2}$, ha $p_i^{(1)} - 1$ nem 2-hatvány. Ha $p_i^{(1)} - 1 = 2^l$, akkor $p_i^{(1)}$ Fermat-prím, és ez az eset nagyon valószínűtlen (talán lehetetlen). Ekkor $\varphi\left(p_i^{(1)} - 1\right) \leq \frac{p_i^{(1)}-1}{2} - 1$, és $\varphi\left(p_i^{(1)} - 1\right) = \frac{p_i^{(1)}-1}{2} - 1$ akkor és csak akkor, ha $\frac{p_i^{(1)}-1}{2} = p_i^{(2)}$ prímszám, más szavakkal, ha $p_i^{(1)} = 2p_i^{(2)} + 1$ egy $p_i^{(2)}$ prímszámmal. Ekkor egyrészt a primitív gyökök aránya

$$\frac{\varphi\left(p_i^{(1)} - 1\right)}{p_i} = \frac{p_i^{(2)} - 1}{p_i} = \frac{\frac{p_i^{(1)} - 1}{2} - 1}{p_i} = \frac{p_i^{(1)} - 3}{2p_i}$$
$$= \frac{\frac{p_i - 1}{2} - 3}{2p_i} = \frac{p_i - 7}{4p_i} \approx \frac{1}{4},$$

tehát könnyű a primitív gyök keresése, másrészt egy tetszőleges k pozitív egész szám esetén könnyű az ellenőrzés, ugyanis $o_{p_i-1}(k) = \varphi(p_i - 1)$, ha $p_i - 1 \nmid k^2 - 1$ és $p_i - 1 \nmid k^{p_i^{(2)}} - 1$.

Végül az előbbi választással $o = [o_i | s \geq i \in \mathbb{N}^+] = \left[2p_i^{(2)} \middle| s \geq i \in \mathbb{N}^+\right] = 2\prod_{i=1}^{s} p_i^{(2)}$, és felhasználva az előbbi eredményt, $\frac{o}{n} = \frac{2\prod_{i=1}^{s} p_i^{(2)}}{\prod_{i=1}^{s} p_i} \approx 2\prod_{i=1}^{s} \frac{1}{4} = 2^{-(2s-1)}$, vagyis $\frac{o}{n}$ annál nagyobb, minél kisebb s. RSA-nál n biztosan összetett szám, ezért a legjobb választás $s = 2$, ami a fixpontok szempontjából is a legjobb. Most $i = 1, 2$-re $N_{e^{2p_i^{(2)}},1}^{(n)} = 3p_i$, és ez akkor viszonylag nagy mindkét i-re, ha $p_1 \approx p_2$, azaz n mindkét faktora körülbelül $\sqrt{n}$.

Az **iteratív fejtés**nek (**ciklikus támadás** – *cyclic attack*) létezik a következő módosított változata. Amennyiben $\left(\left(c^{e^k} \bmod n\right) - c, n\right) > 1$ egy pozitív k egész kitevővel, de $c^{e^k} \bmod n \neq c$, akkor $\left(\left(c^{e^k} \bmod n\right) - c, n\right) = p_1$ vagy $\left(\left(c^{e^k} \bmod n\right) - c, n\right) = p_2$, és ekkor ismert n felbontása, tehát d meghatározható, a rendszert sikerült feltörni. Ám a faktorok fentiekben ismertetett választásával a legkisebb ilyen k kitevő $2p_1^{(2)} \approx \frac{\sqrt{n}}{2}$, feltéve, hogy $p_1 < p_2$.

A p prímszám **Sophie Germain-prím**, ha $p = 2p' + 1$ alakú a p' prímmel. Láttuk, hogy RSA-hoz a kétszeresen Sophie Germain prímek a jók (vagyis ahol p' is Sophie Germain-prím). Kérdés, hogy létezik-e ilyen prím. A válasz igenlő: például $2 \cdot 2 + 1 = 5$ és $2 \cdot 5 + 1 = 11$, $2 \cdot 5 + 1 = 11$ és $2 \cdot 11 + 1 = 23$, $2 \cdot 11 + 1 = 23$ és $2 \cdot 23 + 1 = 47$, $2 \cdot 41 + 1 = 83$ és $2 \cdot 83 + 1 = 167$ stb.

Az $n = pq$ választásánál az eddigieken túl egy további szempont, hogy $|q - p|$ sem lehet kicsi, ugyanis $(q + p)^2 - (q - p)^2 = 4pq = 4n$, innen $(q + p)^2 = 4n + (q - p)^2$, vagyis egy kis pozitív egész négyzetét $4n$-hez adva ismét négyzetszámot kapunk, amit könnyen lehet ellenőrizni. Ha tehát u és v olyan egészek, hogy $4n + u^2 = v^2$, akkor $a = \frac{v-u}{2}$, $b = \frac{v+u}{2}$ és $n = ab$, de n egyetlen felbontása pq, tehát $a = p$ és $b = q$, n-et könnyű faktorizálni, és így már könnyű $\varphi(n)$-et és az $ex \equiv 1\ (\varphi(n))$ megoldását megtalálni. Ez mutatja, hogy p és q választásánál a $|q - p| \approx p(\approx q)$ feltételnek is teljesülnie kell, ha azt akarjuk, hogy ne lehessen könnyen megfejteni a rejtjelünket.

Ha $p = 2p' + 1$ és $q = 2q' + 1$, ahol p' és q' páratlan prímszám, akkor $(p - 1, q - 1) = 2$, ami azért is fontos, mert $t = [p - 1, q - 1] = \frac{(p-1)(q-1)}{\delta} < \frac{pq}{\delta} = \frac{n}{\delta}$, ahol $\delta = (p - 1, q - 1)$, és ha δ nagy, akkor t kicsi, és kevés próbálkozással található olyan d, amellyel $c^d \bmod n = m$.

Most megnézzük, mi a kapcsolat a rejtjel biztonsága és a rejtjelből nyerhető részleges információ között. Megmutatjuk, hogy ha meg tudjuk állapítani a rejtjeles szövegből a nyílt szöveg utolsó bitjét, akkor már az egész szöveget könnyen fejthetjük (egy egyirányú függvény egy bitje **kemény bit**, ha a fejtése azonos nehézségű, mint a függvény invertálása). Először egy segéderedményt látunk be.

8.15. Tétel

Legyen $2 \nmid n \in \mathbb{N}$ négyzetmentes, e a $\varphi(n)$-hez relatív prím pozitív egész, $f: x \mapsto x^e \bmod n$ az $M^{(n)}$ halmaz önmagába való leképezése, $K \in \mathbb{Z}$ olyan, hogy $2^e K \equiv 1\ (n)$, $u \in M^{(n)}$, $v = f(u)$ és $z = u \bmod 2$. Ekkor $u' = 2^{-1}((-1)^z v \bmod n) \in M^{(n)}$ és $v' = K(-1)^z v \bmod n = f(u')$.

Δ

Bizonyítás:

Elsőként megjegyezzük, hogy létezik a tételben igényelt K, hiszen n páratlan.

$u \in M^{(n)}$ esetén $-n < (-1)^z u < n$-ből $(-1)^z u \bmod n = zn + (-1)^z u$. Ez egy n-nél kisebb, páros, nem negatív egész, tehát osztható kettővel, és u' ismét n-nél kisebb nem negatív egész. e páratlan, hiszen n páratlan, egynél nagyobb páratlan számra $\varphi(n)$ páros, és páros számhoz relatív prím páratlan, ezért $(-1)^{ez} = (-1)^z$. Ezt felhasználva $2u' = (-1)^z u \bmod n \equiv (-1)^z u\ (n)$-ből

$$\begin{aligned} u'^e &\equiv (2^e K)u'^e = K(2u')^e \equiv K(-1)^{ez} u^e \\ &\equiv K(-1)^z v \equiv K(-1)^z v \bmod n = v'(n), \end{aligned}$$

és így v' az u' képe az f leképezésnél, $v' = f(u')$, ahogy a tételben állítottuk.

□

Ezt az eredményt felhasználjuk a következő tétel bizonyításában.

8.16. Tétel

Legyen $r \in \mathbb{N}^+$, $2 \nmid n \in [2^{r-1}, 2^r[$ négyzetmentes és e a $\varphi(n)$-hez relatív prím egész, $K \in \mathbb{Z}$-re $2^e K \equiv 1\ (n)$, $M^{(n)}$-en $f: x \mapsto x^e \bmod n$, $m \in M^{(n)}$ és $c = f(m)$. Ha $g(v) = f^{-1}(v) \bmod 2$, úgy az alábbi algoritmus c-ből előállítja m-et:

$$\begin{array}{rrcl} & y_0 & = & c \\ r-1 > i \in \mathbb{N}: & y_{i+1} & = & K(-1)^{z_i} y_i \bmod n \end{array} \qquad \begin{array}{rcl} z_0 & = & g(y_0) \\ z_{i+1} & = & g(y_{i+1}) \end{array}$$

majd

$$\begin{array}{rrcl} & t_{r-1} & = & z_{r-1} \\ r-1 > i \in \mathbb{N}: & t_i & = & ((-1)^{z_i}(2t_{i+1}) \bmod n) \bmod 2^{r-i}. \end{array}$$

Δ

Bizonyítás:

Azt már tudjuk, hogy ha $x_0 = m$ és $x_{i+1} = 2^{-1}((-1)^{z_i} x_i \bmod n)$, akkor $y_i = f(x_i)$, így az algoritmusban meghatározott z_i éppen $x_i \bmod 2$. Azt fogjuk belátni, hogy minden $r - 1 > i \in \mathbb{N}$ egészre $t_i = x_i \bmod 2^{r-i}$. Ebből már következik az állítás, hiszen $n < 2^r$ következtében mind x_0, mind t_0 n-nél kisebb nem negatív egész a definíciók alapján, és így $m = x_0 = t_0$.

z_i az x_i paritását mutatja, így z_i éppen x_i jobb szélső bitje az x_i kettes számrendszerbeli felírásánál, és ha $i = r - 1$, akkor tehát $x_{r-1} \bmod 2 = z_{r-1} = t_{r-1}$. Tegyük fel, hogy egy $r - 1 > i \in \mathbb{N}$, esetén $t_{i+1} = x_{i+1} \bmod 2^{r-(i+1)}$, azaz $t_{i+1} \equiv x_{i+1}\ (2^{r-i-1})$. Ekkor $2t_{i+1} \equiv 2x_{i+1}\ (2^{r-i})$, továbbá az előző tétel alapján y_{i+1} az $x_{i+1} = 2^{-1}((-1)^{z_i} x_i \bmod n)$ üzenethez tartozó rejtjel. Innen

$$\begin{aligned} t_i &\equiv (-1)^{z_i}(2t_{i+1}) \bmod n = z_i n + (-1)^{z_i}(2t_{i+1}) \equiv z_i n + (-1)^{z_i}(2x_{i+1}) \\ &= z_i n + (-1)^{z_i}(z_i n + (-1)^{z_i} x_i) = z_i n + (-1)^{z_i} z_i n + x_i = x_i\ (2^{r-i}), \end{aligned}$$

mert $z_i n + (-1)^{z_i} z_i n = 0$, tekintettel arra, hogy z_i csupán nulla vagy egy lehet.

□

8.17. Megjegyzés

Az előző tétel alapján belátható, hogy amennyiben azt tudjuk y-ból megállapítani, hogy x kisebb-e, mint n fele, vagy nagyobb, akkor hasonlóan egyszerű már a fejtés (harmadik lehetőség, azaz egyenlőség most kizárt, mert n páratlan egész és x egész szám). Legyen ugyanis adott $n > y \in \mathbb{N}$-re $h(y) = \zeta = \left\lfloor \frac{2x}{n} \right\rfloor$, ahol $n > x \in \mathbb{N}$ és $y = f(x) = x^e \bmod n$. Mivel x korlátai alapján $0 \le 2x < 2n$, ezért $0 \le \frac{2x}{n} < 2$, így $\zeta = 0$, ha $x < \frac{n}{2}$, és 1, amennyiben $x > \frac{n}{2}$. $a = \frac{2x}{n}$-re alkalmazva az $a - 1 < \lfloor a \rfloor \le a$ relációt kapjuk, hogy $0 \le 2x - \zeta n < n$, vagyis $2x \bmod n = 2x - \zeta n$. De n a tétel feltétele alapján páratlan, így $n \equiv 1\ (2)$, és evvel $2x \bmod n = 2x - \zeta n \equiv \zeta n \equiv \zeta\ (2)$, vagy másként írva $(2x \bmod n) \bmod 2 = \zeta$.

Most megmutatjuk, hogy z és ζ bármelyikét ismerve, a másikat könnyű meghatározni. Ha $y' = 2^e y \bmod n$, akkor $(2x)^e = 2^e x^e \equiv 2^e y\ (n)$, így $x' = 2x \bmod n$ olyan, hogy $y' = f(x')$, tehát ha g-re van algoritmus, akkor $g(y') = x' \bmod 2 = (2x \bmod n) \bmod 2 = \zeta$. Fordítva, legyen $y^* = Ky \bmod n$. Mivel f bijektív, ezért van olyan $n > x^* \in \mathbb{N}$, amellyel $y^* = f(x^*) = (x^*)^e \bmod n$. Legyen $\zeta = h(y^*)$ meghatározható. Ekkor

$$(2x^* - \zeta n)^e \equiv (2x^*)^e = 2^e (x^*)^e \equiv 2^e y^* \equiv 2^e K y \equiv y \equiv x^e\ (n),$$

ami f injektivitása és $0 \le x < n$, $0 \le 2x^* - \zeta n < n$ miatt csak úgy lehet, ha $x = 2x^* - \zeta n$. Ez egyúttal azt is jelenti, hogy $x \equiv \zeta\ (2)$ és $g(y) = \zeta$. y-ból $2^e y \bmod n$ illetve $Ky \bmod n$ számítása könnyű feladat, és így g és h bonyolultsága megegyezik.

Még megemlítjük, hogy ha n bináris felírásában az utolsó s bit mindegyike 1, akkor amennyiben y-ból x alsó s bitjének bármelyikét meg tudjuk határozni, akkor ebből már könnyű a fejtés. Ennek bizonyítása valamivel hosszabb, mint az előbbi két eset, ezért eltekintünk tőle.

Δ

Most három kérdésre térünk ki. Mi történik, ha

1. $(m,n) > 1$ (m a nyílt szöveg és n a modulus);
2. n több kulcsra azonos;
3. illetve több résztvevőre megegyezik az e kitevő (de a modulusok különbözőek).

Az első kérdés könnyen elintézhető. Ha $n = pq$ és $m \neq 0$, akkor $1 < (m,n)$ csak p vagy q lehet. Ha a közös osztó p, akkor $(c,n) = p$ is igaz, és ebből a támadó meg tudja határozni q-t, $\varphi(n)$-et és d-t, vagyis feltöri a rendszert. Ennek valószínűsége azonban csekély, hiszen az n-hez nem relatív prím, n-nél kisebb nem negatív egészek száma $n - \varphi(n) = pq - (p-1)(q-1) = p + q - 1$, és ezek aránya az n-nél kisebb nem negatív egészekhez $\frac{p+q-1}{pq} \approx \frac{1}{p} + \frac{1}{q} \approx \frac{2}{\sqrt{n}}$, viszont n nagy.

Nézzük a második esetet. Legyen k résztvevő esetén azonos az n modulus, és közülük az i-edik nyilvános kulcsa e_i. Ha közülük akár csak kettő, mondjuk az 1 és 2 indexű, azonos nyílt szöveg rejtjeles változatát kapja (például egy körlevelet), és e_1, e_2 relatív prím, akkor egy támadó is vissza tudja fejteni c_1-ből és c_2-ből az m üzenetet. Most ugyanis $c_1 \equiv m^{e_1}\ (n)$ és $c_2 \equiv m^{e_2}\ (n)$. Mivel e_1 és e_2 relatív prím, ezért van olyan u_1 és u_2 egész, hogy $1 = u_1 e_1 + u_2 e_2$. e_1 és e_2 1-nél nagyobb pozitív egész, ezért az előbbi egyenlőség csak úgy állhat fenn, ha u_1 és u_2 egyike pozitív, a másik negatív. Szimmetriaokokból bármelyiküket tekinthetjük negatívnak, legyen például $u_1 < 0$ és $u_2 > 0$. Ekkor

$$m = m^1 = m^{u_1 e_1 + u_2 e_2} = (m^{e_1})^{u_1} (m^{e_1})^{u_1} \equiv c_1^{u_1} c_2^{u_2}\ (n),$$

innen $(c_1)^{(-u_1)} m \equiv c_2^{u_2}\ (n)$ ($-u_1$ már pozitív), vagyis m a $(c_1)^{(-u_1)} x \equiv c_2^{u_2}\ (n)$ kongruencia megoldása, és megoldás biztosan létezik, például az eredeti m üzenet, vagyis a rejtjeles szövegekből ismert kitevős hatványokkal egy egyismeretlenes lineáris kongruencia megoldásaként, tehát polinomiális időben megkapjuk a nyílt szöveget (egy ilyen kongruencia például euklideszi algoritmussal megoldható, és ez polinomiális algoritmus).

Most tegyük fel, hogy k-számú résztvevőnek azonos e rejtjelkitevője van, és az i-edikhez az n_i modulus tartozik, továbbá a modulusok páronként relatív prímek (ennél enyhébb feltétel is elegendő lenne). Ha egy körlevél következtében mindegyikük azonos rejtjeles szöveget kap, akkor a közös m könnyen meghatározható egy kívülálló részéről is. Legyen c_i az i-edik rejtjeles szöveg, akkor tehát $c_i \equiv m^e\ (n_i)$ valamennyi i-re, azaz m^e a megoldása a $c_i \equiv x\ (n_i)$ szimultán kongruenciarendszernek. De ennek egy és csak egy megoldása van modulo n, ahol n az n_i-k szorzata, így egy és csak egy olyan megoldás van, ahol $n > x \in \mathbb{N}$. Nyilván érvényesnek kell lennie az $n_i > m \in \mathbb{N}$ feltételnek minden i-re, így ha $k \geq e$, akkor $n > m^e \in \mathbb{N}$, és ezért most $m^e = x$, ahonnan m gyökvonással megkapható.

Az, hogy több felhasználó nyilvános rejtjelkitevője azonos, nem rendkívüli. e nagysága nem befolyásolja különösebben a rejtjel biztonságát, ezért a számítás egyszerűsége érdekében célszerű kicsire választani. Ha a rendszerben sok szereplő vesz részt, akkor előfordul, hogy bár egymástól függetlenül választják a paramétereket, de a kevés számú kis érték közül többen is azonosat választanak.

Még nézzük meg a paraméterek választását. Véletlen prímet például véletlenszám-generátorral nyerhetünk: generálunk egy számot, prímteszttel megvizsgáljuk, és ha nem prím (illetve nem minősítjük prímnek), akkor vehetjük a természetes számsorban következő páratlan egészt. Tegyük fel, hogy m-nagyságrendű prímet keresünk. Csebisev tétele szerint bármely szám és a kétszerese között van prím, és a nagy prímszámtétel szerint az x számnál nem kisebb prímek száma, $\pi(x)$, nagy x-ekre körülbelül $\frac{x}{\ln x}$, $\pi(x) \sim \frac{x}{\ln x}$. Ekkor az m és $2m$ közötti prímek várható aránya

$$\frac{\pi(2m) - \pi(m)}{m} \sim \frac{\frac{2m}{\ln(2m)} - \frac{m}{\ln m}}{m} = \frac{2}{\ln 2 + \ln m} - \frac{1}{\ln m} \approx \frac{1}{\ln m},$$

vagyis várhatóan $\ln m$ kísérlet után prímet kapunk, sőt, ha figyelembe vesszük, hogy a prímek páratlanok (kivéve a 2-t), és csak minden második szám páratlan (és csak ezekkel kísérletezünk), akkor átlagosan $\frac{\ln m}{2}$ kísérlettel prímhez jutunk. Konkrétan $m{\sim}10^{100}$ esetén ez körülbelül 115 próbálkozást jelent (megjegyezzük, hogy az előbbi kétszeres tartománynál lényegesen kisebb intervallumra is igaz, hogy van benne prím, ha x elegendően nagy, másrészről láttuk, hogy nem akármilyen prím alkalmas).

e-nek relatív prímnek kell lennie $\varphi(n)$-hez. Ez például úgy biztosítható, ha $q < e < n$ és e prím, ahol q az n-ben lévő nagyobbik prím. Ez az e valóban relatív prím $\varphi(n)$-hez, hiszen ez utóbbi minden prímosztója kisebb e-nél.

Amennyiben e kicsi, akkor viszonylag nagy $\sqrt[e]{n}$. De ha $0 \leq m < \sqrt[e]{n}$, akkor $m^e < n$, ekkor $c = m^e \bmod n = m^e$, és ebből egyszerű gyökvonással megkapjuk m-et (ha tudjuk, hogy a gyök egész szám, és most tudjuk, akkor a gyökvonás könnyen elvégezhető), vagyis ekkor könnyű a támadó dolga. De ha $(p-1, q-1)$ kicsi (ami egyébként, mint láttuk, kívánatos), akkor a kicsi d is problémás, ugyanis ha $d \approx \sqrt[4]{n}$, akkor létezik hatékony módszer, amely n és e ismeretében kiszámítja d-t. Ez az algoritmus nem terjeszthető ki arra az esetre, ha $d \approx n$.

$\left(m^{(1)}m^{(2)}\right)^e \bmod n = \left(\left(m^{(1)}\right)^e \bmod n\right)\left(\left(m^{(2)}\right)^e \bmod n\right)$, vagyis az RSA **multiplikatív tulajdonságú**, és így választott rejtjelszövegű támadással egy adott üzenet fejthető. Legyen ugyanis c az A résztvevő (e, n) nyilvános kulcsával rejtjelezett szöveg, és m a megfelelő nyílt szöveg. Válasszunk egy tetszőleges, modulo n invertálható r egészt, és kérjük $c' = r^e c \bmod n$ nyilvános párját. Ha ez m', akkor $m' = (r^e c \bmod n)^d \bmod n = rm \bmod n$, és ezt megszorozva r modulo n inverzével, maradékolás után megkapjuk m-et. Ez ellen úgy lehet védekezni, hogy az üzenetek formájára valamilyen előírást adunk. Annak a valószínűsége, hogy egy adott r-rel $rm \bmod n$ megfelel a formai előírásnak (m-et nem ismerjük!), igen csekély.

Vissza a tartalomhoz

9. A Rabin-variáns

Az alább ismertetendő rejtjelező algoritmus lényegében véve az RSA módosítása, ám ennél az eljárásnál bizonyítani tudjuk, hogy a bonyolultsága azonos az egész számok faktorizációjának bonyolultságával.

Legyen n egy páratlan, nem negatív egész szám és b tetszőleges egész szám, legyen továbbá minden $m \in M^{(n)}$-re $c = m(m+b) \bmod n$. Mivel n páratlan, ezért $(2,n) = 1$, és így van olyan u egész szám, amellyel $2u \equiv b\ (n)$. Ekkor $m(m+b) \bmod n = ((m+u)^2 - u^2) \bmod n$, tehát c-ből meghatározni m-et ekvivalens azzal, hogy meghatározunk egy olyan m'-t, amellyel $m'^2 \equiv c + u^2\ (n)$. Tekintettel arra, hogy egy rögzített b esetén u, és vele együtt u^2 konstans, ezért az egész eljárás lényegében véve azonos az $m \mapsto m^2 \bmod n$ leképezéssel. Ennek megfelelően a továbbiakban ezt az utóbbi eljárást vizsgáljuk.

Látható, hogy az $m \mapsto m^2 \bmod n$ leképezés $M^{(n)}$-en hasonló egy olyan RSA-hoz, ahol $e = 2$. Ugyanakkor tudjuk, hogy RSA-nál az egyértelmű fejtés csak akkor valósítható meg, ha $\big(e, \varphi(n)\big) = 1$, ami csak akkor lehet igaz, ha e páratlan, hiszen $n > 2$ esetén $\varphi(n)$ páros. Egyébként ennél a leképezésnél nyilvánvaló, hogy nem injektív, hiszen ha $m \in M^{(n)}$, akkor m és $n - m = -m \bmod n$ négyzete, tehát képe azonos. Majd látjuk, hogy a helyzet ennél még bonyolultabb, vagyis ha egy c-nek van modulo n négyzetgyöke, akkor összetett és páratlan modulus esetén kettőnél több ilyen gyöke van c-nek. Azt mindenesetre tegyük fel, hogy n négyzetmentes, vagyis $n = \prod_{i=1}^{s} p_i$, ahol s 1-nél nagyobb pozitív egész szám, és a p_i-k páronként különböző páratlan prímszámok.

Legyen $c \in M^{(n)}$. c-nek akkor és csak akkor van modulo n négyzetgyöke, ha minden i-re van modulo p_i négyzetgyöke. Ekkor minden i-re két négyzetgyöke van, kivéve, ha valamelyik p_i-vel osztható, és ha mindegyik i-re két négyzetgyök van, akkor ezekből a kínai maradéktétellel tudunk meghatározni összesen 2^s olyan u-t, amelynek a modulo n négyzete éppen c. Prímmodulus esetén könnyű feladat a moduláris gyökvonás, és különösen könnyű, ha $p \bmod 4 = 3$, azaz $p = 4k + 3$ alakú prímre. Ekkor $\left(c^{\frac{p+1}{4}}\right)^2 = c^{\frac{p+1}{2}} = c^{\frac{p-1}{2}} \cdot c$, és $c^{\frac{p-1}{2}} \bmod p = \begin{cases} +1 \\ -1 \end{cases}$, tehát $\left(c^{\frac{p+1}{4}}\right)^2 = \begin{cases} +c \\ -c \end{cases}$, vagyis ha van c-nek modulo p négyzetgyöke, akkor $\left(c^{\frac{p+1}{4}}\right)^2 = c$, különben $\left(c^{\frac{p+1}{4}}\right)^2 = -c$. Ha tehát $c = m^2 \bmod n$, akkor c-nek van modulo p_i négyzetgyöke, és $c^{\frac{p_i+1}{4}} \bmod p_i = m_i$, ha $p_i \bmod 4 = 3$ (ha p $4k+1$-alakú prímszám, akkor is van polinomiális algoritmus, amellyel meghatározható egy szám modulo p négyzetgyöke, feltéve, hogy van neki, de ekkor az előbbi hatványozásnál bonyolultabb az eljárás). Ha most u_i olyan, hogy $\frac{n}{p_i} u_i \equiv 1\ (p_i)$ (mivel n négyzetmentes, ezért ilyen u_i mindig van), akkor a kínai maradéktétel szerint $m' = \sum_{i=1}^{s} \left(\pm \frac{n}{p_i} u_i\right) m_i \bmod n$ mindegyike, és csak ezek, olyan, n-nél kisebb nem negatív egész számok, amelyeknek a modulo n négyzete éppen c (az összegben az egyes tagok előjele egymástól független, és az összes lehetséges kombinációban előfordul, így kapjuk a legfeljebb 2^s különböző gyököt).

Legyen $s = 2$, $n = pq$, $m_p^2 \equiv c\ (p)$, $m_q^2 \equiv c\ (q)$, $v_p = q u_p \equiv 1\ (p)$ és $v_q = p u_q \equiv 1\ (q)$. Ekkor $m' = \left(\pm v_p m_p \pm v_q m_q\right) \bmod n$ adja a legfeljebb négy megoldást (két megoldás van, ha c osztható n egyik és csak egyik prímtényezőjével, és egy megoldás van, ha $c = 0$), és a négy megoldás közül kettő (a két megoldás közül az egyik) kisebb, mint $\frac{n}{2}$.

Hogyan lehet kiválasztani a több megoldás közül az eredeti üzenetet? Ha kikötjük, hogy az üzenet legyen kisebb, mint n fele, akkor már csak két változat közül kell választani. Akár négy, akár két megoldás valamelyike az eredeti üzenet, egyszerű a választás, ha valamilyen szöveges üzenet megfejtéséről van szó, mert ekkor elég nagy valószínűséggel a több lehetséges változat közül csupán az egyik felel meg értelmes üzenetnek. Amennyiben viszont más jellegű az eredeti üzenet, akkor már nehezebb a több, látszólag értelmetlen szövegből kiválasztani a tényleges megfejtést. Ilyenkor (is) segít, ha az

üzenet egy előre meghatározott részén (például fejlécben) valamilyen előre rögzített formának vagy tartalomnak megfelelő információ van.

Az előbbiek szerint könnyű fejteni a Rabin-variánssal rejtjelezett szöveget, ha valaki ismeri n prímtényezőit. Ugyanakkor a moduláris négyzetgyökvonás és az egész számok faktorizációja algoritmikusan azonos nehézségű probléma, vagyis ha az egyik nem oldható meg polinomiális időben, akkor a másik sem, tehát az n felbontásának ismerete nélkül a Rabin-variáns gyakorlatilag fejthetetlen. Tegyük ugyanis fel, hogy képesek vagyunk polinomiális időben négyzetgyököt vonni egy összetett modulusra vonatkozóan, vagyis tetszőleges v esetén, feltéve, hogy v kvadratikus maradék modulo n, azaz van v-nek modulo n négyzetgyöke, az n prímfaktorainak ismerete nélkül polinomiális időben meg tudjuk határozni v valamely négyzetgyökét. Válasszunk ekkor egy n-nél kisebb nem negatív u véletlen számot, és legyen $v = u^2 \bmod n$, majd határozzuk meg az algoritmusunkkal v egy négyzetgyökét. Ha ez u_1, akkor $u_1^2 \bmod n = v = u^2 \bmod n$, vagyis $n \mid u^2 - u_1^2 = (u - u_1)(u + u_1)$. Amennyiben $n|u - u_1$ vagy $n|u + u_1$, akkor $u = u_1$ vagy $u = n - u_1$, és semmi új információnk nincs. Ám ha az előbbi két oszthatóság egyike sem igaz, de a szorzat osztható n-nel, akkor $(n, u - u_1) = n_1$, ahol n_1 az n egy nem triviális osztója, és ekkor faktorizáltuk n-et. Ha $n = pq$ a páratlan és különböző p és q prímszámokkal, és u relatív prím n-hez, aminek a valószínűsége, mint láttuk, csaknem 1 (ha pedig nem relatív prímek, akkor azonnal megkapjuk n felbontását), akkor v-nek négy négyzetgyöke van, és $\frac{1}{2}$ annak a valószínűsége, hogy az algoritmus éppen az általunk választott véletlen számot vagy annak ellentettjét számolja ki n négyzetgyökeként. Várhatóan tehát egy-két kísérlet után u_1 a másik két négyzetgyök egyike lesz, és végeredményben polinomiális időben faktorizáltuk n-et (nyilván hasonló a helyzet, ha n kettőnél több különböző prímszám szorzata). Mivel jelen ismereteink szerint a faktorizálásra nincs polinomiális futási idejű algoritmus, összetett modulus esetén a tényezők ismerete nélkül a $c \equiv x^2\ (n)$ kongruencia gyökeinek meghatározása algoritmikusan nehéz feladat, így a Rabin-variáns alkalmas rejtjelezésre.

A konkrét megvalósítás során általában n mindkét prímfaktora $4k + 3$ alakú. Az ilyen számokat **Blum-egész**nek hívják. Ekkor $\left(\frac{-1}{p}\right) = -1 = \left(\frac{-1}{q}\right)$, és ha a c négy lehetséges négyzetgyöke közül mondjuk $m' = v_p m_p + v_p m_q \bmod n$ olyan, hogy $\left(\frac{m'}{n}\right) = 1$, akkor $\left(\frac{-m'}{n}\right) = 1$, és a másik két gyökre $\left(\frac{-v_p m_p + v_p m_q}{n}\right) = -1 = \left(\frac{v_p m_p - v_p m_q}{n}\right)$, ahol az u és a páratlan v egész számra $\left(\frac{u}{v}\right)$ a Jacobi-szimbólum. (A modulo m nem 0 n egész modulo m **kvadratikus maradék**, ha van olyan u egész szám, hogy $u^2 \equiv n\ (m)$, egyébként n modulo m **kvadratikus nemmaradék**. Amennyiben $m = p$ prímszám, akkor

$$\left(\frac{n}{p}\right) = \begin{cases} 0, \text{ha } p|n \\ 1, \text{ha } n \text{ kvadratikus maradék modulo } p \\ -1, \text{ha } n \text{ kvadratikus nemmaradék modulo } p \end{cases}$$

az *n* **per** *p* **Legendre-szimbólum** (kiejtésben *Lözsandr*). Tetszőleges páratlan m pozitív egész modulusra definiáljuk az *n* **per** *m* **Jacobi-szimbólum**ot. Legyen $m = \prod_{i=1}^{s} p_i$, ahol a p_i prímek nem feltétlenül különbözőek. Ekkor $\left(\frac{n}{m}\right) = \prod_{i=1}^{s} \left(\frac{n}{p_i}\right)$ az $\left(\frac{n}{p_i}\right)$ Legendre-szimbólumokkal. A Jacobi-szimbólum, mint az könnyen belátható, nem mutatja, hogy n kvadratikus maradék-e modulo m, csak annyi biztos, hogy értéke 0 vagy ± 1, és ha -1, akkor n biztosan nem kvadratikus maradék az m modulussal.

10. Diszkrét logaritmus

Legyen G egy n-edrendű ciklikus csoport a g generátorelemmel. Ekkor a G bármely u eleméhez van egy, a g és u által egyértelműen meghatározott, $n > k \in \mathbb{N}$ egész szám, amellyel $g^k = u$, és ennek megfelelően az az $u \mapsto k$ szabály, amely u-hoz az előbbi k-t rendeli, G-nek $M^{(n)}$-be való bijektív leképezése ($M^{(n)}$ a korábban már más összefüggésben definiált halmaz, amely az n-nél kisebb nem negatív egész számokat tartalmazza.) Az előbbi leképezést $\operatorname{ind}_g u$-val vagy $\log_g u$-val jelöljük, és g-**alapú diszkrét logaritmus**nak vagy g**-alapú index**nek nevezzük. Könnyű ellenőrizni, hogy

- $\operatorname{ind}_g(uv) = (\operatorname{ind}_g u + \operatorname{ind}_g v) \bmod n$;
- $\operatorname{ind}_g u = 0 \Leftrightarrow u = e$;
- $u \neq e \Rightarrow \operatorname{ind}_g u^{-1} = n - \operatorname{ind}_g u$;
- $\operatorname{ind}_g u^r = (r \cdot \operatorname{ind}_g u) \bmod n$,

ahol e a csoport egységeleme, és r tetszőleges egész szám.

k ismeretében, adott g esetén, u meghatározása könnyű feladat, ám az inverz művelet a mai ismereteink szerint algoritmikusan nehéz feladat, ezért alkalmazhatjuk a rejtjelezésben.

A diszkrét logaritmus meghatározására több algoritmus is létezik, itt most kettőt ismertetünk.

1. Legyen g az n-elemű G ciklikus csoport generátoreleme, $s|n$ és $c = g^{\frac{n}{s}}$, ekkor c egy s-edrendű elem. Ha $\alpha^s = e$, akkor alkalmas $s > m \in \mathbb{N}$ kitevővel $\alpha = c^m$. Most α ismeretében meg akarjuk határozni az m kitevőt, feltéve, hogy $m \neq 0$, hiszen ellenkező esetben $\alpha = e$, és innen azonnal látjuk a megoldást. A feladatot megoldhatjuk például úgy, hogy $d_0 = e$-ből kiindulva addig képezzük $s > i \in \mathbb{N}^+$-ra a $d_i = d_{i-1}c$ elemet, amíg valamilyen $s > k \in \mathbb{N}^+$-ra d_k meg nem egyezik α-val. Ekkor $d_k = c^k$, és mivel c rendje s, ezért ez csak $k = m$-mel lehetséges. Egy másik megoldás, hogy előre kiszámítjuk $s > j \in \mathbb{N}$-re a $d_j = c^j$ hatványokat, és eltároljuk őket. Amikor α diszkrét logaritmusát kell meghatározni, akkor már nincs más dolgunk, mint egymás után összehasonlítani α-t a d_j elemekkel, és ha k-ra találunk egyezést, akkor az előző megfontolás alapján ismét azt kapjuk, hogy k és m azonos. Mindkét esetben feltesszük természetesen, hogy nem egyetlen logaritmus meghatározása a cél, így az első megoldásnál gyakorlatilag nincs memóriaigény, de sokat kell számolni, míg a második esetben csak egyszer kell számolni, utána már csupán összehasonlítások vannak, viszont nagy s esetén nagy tárolókapacitásra van szükség.

A két igény egymás rovására módosítható. Legyen t 1-nél nem nagyobb nem negatív valós szám, és $u = \lceil s^t \rceil$. Az előbbi m egyértelműen írható $m = au + b$ alakban, ahol b u-nál kisebb nem negatív egész; ekkor $a = \frac{m-b}{u} \leq \frac{m}{u} < \frac{s}{u} = \frac{s}{\lceil s^t \rceil} \leq \frac{s}{s^t} = s^{1-t} \leq \lceil s^{1-t} \rceil$. Számítsuk ki és tároljuk el c $u > k \in \mathbb{N}$ kitevős hatványait. m meghatározása ekvivalens a és b megadásával. Legyen $r = c^{-u}$, ekkor az $\alpha = c^m = c^{au+b} = (c^{-u})^{-a}c^b$ egyenlőségből $\alpha \cdot r^a = c^b$. Az eljárás tehát a következő. Induljunk ki $\alpha_0 = \alpha$-ból, és nézzük meg, hogy teljesül-e valamilyen $u > l \in \mathbb{N}$-nel az $\alpha_0 = c^l$ egyenlőség. Ha igen, akkor $a = 0$ és $b = l$-lel $c^{au+b} = \alpha$, és készen vagyunk. Ha nem, akkor legyen $\alpha_1 = \alpha_0 \cdot r$, és ismételjük meg a keresést. Ha sikerrel jártunk, akkor $a = 1$ és $b = l$-lel megtaláltuk a megoldást, ha nem, akkor legyen $\alpha_2 = \alpha_1 \cdot r$ stb. Ez az eljárás véges sok lépésben pozitív eredménnyel befejeződik, hiszen $i = a$-val a keresés α_i-re sikeres lesz.

2. Tegyük fel, hogy $n = \prod_{i=1}^{s} p_i^{r_i}$, ahol a p_i-k páronként különböző prímek, és s valamint az r_i-k pozitív egészek. Ha meg tudjuk határozni az $m^{(i)} = \alpha \bmod p_i^{r_i}$ egészeket, akkor ebből a kínai maradéktétellel egyértelműen megkapjuk α-t is. $m^{(i)}$ szintén egyértelműen írható $m^{(i)} = \sum_{j=0}^{r-1} m_j^{(i)} p_i^j$ alakban, ahol az $m_j^{(i)}$ együtthatók mindegyike p_i-nél kisebb nem negatív egész, így a feladat az, hogy minden i-re és j-re meghatározzuk $m_j^{(i)}$ értékét. Mivel a feladat minden i-re azonos, ezért a továbbiakban az i indexet elhagyjuk. Most ismét ki kell számolni és elraktározni a $d_j = \left(g^{\frac{n}{p}}\right)^j$ hatványokat, ahol $p > j \in \mathbb{N}$. Mivel $g^{\frac{n}{p}}$ primitív p-edik gyök, ezért a $p > t \in \mathbb{N}$-kitevős hatványok egyértelműen határozzák meg az előbbi határok közé eső kitevőket. $\alpha = m + t \cdot p^r$ egy alkalmas nem negatív t egésszel (amely persze általában nem kisebb a megfelelő prímnél), és így $r > k \in \mathbb{N}$-re

$$\alpha = \sum_{j=0}^{k-1} m_j p^j + m_k p^k + p^{k+1}\left(\sum_{j=0}^{(r-1)-(k+1)} m_{(k+1)+j} + tp^{r-(k+1)}\right)$$
$$= u_k + m_k p^k + v_k p^{k+1}.$$

Tegyük fel, hogy már ismerjük $m_0, \cdots, m_{k-1}$, vagyis u_k értékét, így a $c_k = cg^{-u_k}$ jelöléssel

$$c_k^{\frac{n}{p^{k+1}}} = (g^\alpha g^{-u_k})^{\frac{n}{p^{k+1}}} = \left(g^{\frac{n}{p}}\right)^{m_k} (g^n)^{v_k} = \left(g^{\frac{n}{p}}\right)^{m_k} = d_{m_k},$$

vagyis $c_0 = c$-ből indulva egymás után meg tudjuk határozni a d_{m_k} és ezáltal az m_k értékeket, és ebből a soron következő c_{k+1}-et.

A második eljárásban természetesen alkalmazhatjuk az első eljárást a keresési műveleteknél.

Most a diszkrét logaritmus három kriptográfiai alkalmazását mutatjuk meg.

1. **Kulcscsere**. Diffie és Hellman cikkében alapvetően nem a nyilvános kulcsú rejtjelezésről volt szó, ez csupán mint egy lehetőség merült fel, ám megoldást erre a kérdésre a cikk nem tartalmazott (viszont, meglepő módon, igen hamar megjelentek megoldások, például az azóta már eredeti formájában nem biztonságos, a hátizsák-algoritmuson alapuló módszer, és az azóta is talán leggyakrabban alkalmazott nyilvános kulcsú rejtjelező algoritmus, az RSA). A két szerző alapvetően a kulcscsere problémájával foglalkozott, azt a kérdést vizsgálták, hogy lehetséges-e, és ha igen, akkor például hogyan lehet nyilvános csatornán kicserélni két kommunikáló fél között a titkos kulcsukat.

A feladat tehát az, hogy két fél szeretne nyilvános csatornán keresztül titkos kulcsot cserélni. Legyen G egy nyilvánosan ismert n-edrendű ciklikus csoport a (szintén nyilvánosan ismert) g generátorelemmel. A választ egy $n > k_A \in \mathbb{N}^+$ és B egy $n > k_B \in \mathbb{N}^+$ értéket. A elküldi B-nek g^{k_A}-t, míg B A-nak g^{k_B}-t. Most A kiszámolja g^{k_B}-ből $g^{k_A k_B}$-t, és hasonlóan, B kiszámítja g^{k_A}-ból $g^{k_A k_B}$-t, és így, láthatóan, lesz egy közös kulcsuk. Ha valaki megfigyeli a csatornán g^{k_A}-t és g^{k_B}-t, és valamelyikből meg tudja határozni a kitevőt, vagyis meg tudja oldani a diszkrét logaritmus problémáját, akkor rendelkezik a közös kulccsal. Ebből látszik, hogy a közös kulcs támadó általi meghatározása legfeljebb olyan bonyolultságú, mint a diszkrét logaritmus problémája. Ám az nem biztos, hogy csak így juthat hozzá a közös kulcshoz. A feladat, amelyet meg kell oldania, az, hogy ha valaki ismeri g két hatványát, ebből meg tudja-e állapítani a két kitevő szorzatához tartozó hatványt, azaz g^{k_A} és g^{k_B} ismeretében ki tudja-e számítani $g^{k_A k_B}$-t. Ez a feladat a **Diffie-Hellman probléma**. Az előbbiek szerint ez tehát legfeljebb olyan bonyolultságú, mint a diszkrét logaritmus probléma, és a rejtjelezés szempontjából reméljük, amint ma hisszük, hogy azzal azonos nehézségű (**Diffie-Hellman hipotézis**).

Egy aktív támadó eredményesen tud beavatkozni a rendszerbe egyéb intézkedések hiányában. A módszer az úgynevezett **középre állás** (*man in the middle attack*). Legyen C a támadó, aki képes a csatornából üzeneteket kivonni illetve beszúrni. Amikor A elküldi B-nek g^{k_A}-t, akkor ezt C kivonja a csatornából, és helyette, választva egy k_{CA} kitevőt, visszaküldi A-nak $g^{k_{CA}}$-t. Hasonlóan, amikor B küldi A-nak g^{k_B}-t, ezt kiemeli a csatornából, és helyette egy általa választott k_{CB} kitevővel visszaküldi B-nek $g^{k_{CB}}$-t. Ha most A egy üzenetet akar váltani B-vel, akkor ezt titkosítja a B-vel közösnek gondolt $g^{k_A k_{CA}}$-val, és elküldi. Ezt elfogja C, megfejti az A-val közösen ismert $g^{k_A k_{CA}}$ kulccsal, majd akár ezt az üzenetet, akár helyette egy másikat a $g^{k_B k_{CB}}$ kulccsal titkosítva, elküldi B-nek. C ugyanígy tud eljárni, ha B küld rejtjelezett üzenetet A-nak.

2. **Közös kulcs nélküli üzenetváltás** Felmerül a kérdés, hogy lehet-e közös kulcs nélkül titkosított üzenetet cserélni. Ha igen, akkor elkerülhető a kulcs kicserélésének problémája. Nézzük a következő protokollt. A beteszi az üzenetet egy ládikába, és lelakatolja a ládikát (amelyhez csak neki van kulcsa), majd elküldi B-nek. B nem tudja kinyitni a ládikát, hiszen nincs kulcsa hozzá, ezért mérgében rátesz még egy lakatot, amelyhez viszont csak neki van kulcsa, és visszaküldi a feladónak. A mosolyogva leveszi a saját lakatját, és ismét elküldi a ládikát B-nek, aki most már hozzáfér az üzenethez.

Az előbbi eljárás ismét támadható a középre állással. Például a postás megteheti, hogy ahelyett, hogy kikézbesítené a ládikát B-nek, inkább ő lakatolja le és küldi vissza, majd amikor ismét visszaér-

kezik a ládika, akkor kiveszi a levelet. Ezek után vagy ugyanezt a levelet, vagy egy másikat, beteszi a ládikába, és végrehajtja a protokollt B-vel, mintha ő lenne A.

Az előbb leírt módszer a kriptográfia nyelvén is megadható. Legyen A titkosító algoritmusa $E^{(A)}$ és fejtő algoritmusa $D^{(A)}$, valamint egy összetartozó kulcspárja $k_A^{(E)}$ és $k_A^{(D)}$, és hasonlóan, legyen $E^{(B)}$ és $D^{(B)}$ a B titkosító és desifrírozó algoritmusa a $k_B^{(E)}$ és $k_B^{(D)}$ kulcspárral. Legyen m az üzenet, amelyet A titkosítva akar B-nek elküldeni. Ekkor kiszámítja $u = E^{(A)}_{k_A^{(E)}}(m)$-et, és ezt elküldi B-nek. B, mivel nem rendelkezik a visszafejtéshez szükséges kulccsal, nem tudja visszafejteni u-t, ezért visszaküldi A-nak $v = E^{(B)}_{k_B^{(E)}}(u)$-t. Most A erre az üzenetre alkalmazza a saját fejtő algoritmusát, vagyis meghatározza $w = D^{(A)}_{k_A^{(D)}}(v)$. Amennyiben $D^{(A)}_{k_A^{(D)}}\left(E^{(B)}_{k_B^{(E)}}\left(E^{(A)}_{k_A^{(E)}}(m)\right)\right) = E^{(B)}_{k_B^{(E)}}(m)$, akkor

$$w = D^{(A)}_{k_A^{(D)}}(v) = D^{(A)}_{k_A^{(D)}}\left(E^{(B)}_{k_B^{(E)}}\left(E^{(A)}_{k_A^{(E)}}(m)\right)\right) = E^{(B)}_{k_B^{(E)}}(m),$$

és ha ezt A visszaküldi B-nek, akkor ebből B a saját visszafejtő algoritmusával visszanyeri m-et, hiszen $D^{(B)}_{k_B^{(D)}}(w) = D^{(B)}_{k_B^{(D)}}\left(E^{(B)}_{k_B^{(E)}}(m)\right) = m$. Ha $E^{(A)}$ és $E^{(B)}$ felcserélhető (például, ha $E^{(A)} = E = E^{(B)}$, és E kommutatív), akkor az eljárás helyesen működik, hiszen ekkor $D^{(A)}_{k_A^{(D)}}\left(E^{(B)}_{k_B^{(E)}}\left(E^{(A)}_{k_A^{(E)}}(m)\right)\right) = D^{(A)}_{k_A^{(D)}}\left(E^{(A)}_{k_A^{(E)}}\left(E^{(B)}_{k_B^{(E)}}(m)\right)\right) = E^{(B)}_{k_B^{(E)}}(m)$ (ez lényegében véve azonos azzal, hogy $D^{(A)}$ és $E^{(B)}$ felcserélhető, mert ekkor is $D^{(A)}_{k_A^{(D)}}\left(E^{(B)}_{k_B^{(E)}}\left(E^{(A)}_{k_A^{(E)}}(m)\right)\right) = E^{(B)}_{k_B^{(E)}}\left(D^{(A)}_{k_A^{(D)}}\left(E^{(A)}_{k_A^{(E)}}(m)\right)\right) = E^{(B)}_{k_B^{(E)}}(m)$).

Az előbb ismertetett eljárás a **hárommenetes protokoll** (*three-pass protocol*) vagy **kulcs nélküli protokoll**. Legyen például G egy nyilvánosan ismert n-edrendű csoport. A választ egy, csak általa ismert $n > e_A \in \mathbb{N}^+$ és B egy $n > e_B \in \mathbb{N}^+$ értéket úgy, hogy mind e_A, mind e_B relatív prím n-hez. Ekkor mindketten meg tudnak határozni egy d_A és d_B egészt úgy, hogy teljesüljön az $e_A d_A \equiv 1\ (n)$ illetve $e_B d_B \equiv 1\ (n)$ kongruencia. e_A és e_B a két fél titkos rejtjelező kulcsa, míg d_A és d_B a szintén titkos fejtő kulcsuk. Tekintettel arra, hogy n-edrendű csoport minden g elemére $g^n = e$, ahol e a csoport egységeleme, $(g^e)^d = g$. Ha m az üzenet, amelyet A küld B-nek (és m eleme G-nek, ami kódolással megoldható), akkor, az előbbi jelölésekkel,

$$\begin{array}{lllllll} u & = & E^{(A)}_{k_A^{(E)}}(m) & = & m^{e_A} & & \\ v & = & E^{(B)}_{k_B^{(E)}}(u) & = & (m^{e_A})^{e_B} & & \\ w & = & D^{(A)}_{k_A^{(E)}}(v) & = & ((m^{e_A})^{e_B})^{d_A} & = & (m^{e_B})^{e_A d_A} = m^{e_B} \\ z & = & D^{(B)}_{k_B^{(E)}}(w) & = & (m^{e_B})^{d_B} & = & m. \end{array}$$

Ha egy támadó fel akarja törni a rendszert, akkor egy diszkrét logaritmus problémát kell megoldania, hiszen a nyilvánosan látható mondjuk $u = m^{e_A}$-ból a kitevőt kell meghatároznia.

Speciális esetként legyen p egy közösen ismert prímszám, e_A és e_B relatív prím $p-1$-hez. Ekkor $E_e = m^e \bmod p$ megfelel az előbbieknek. Ez az eredeti, **Shamir-féle hárommenetes protokoll** vagy **Shamir-féle kulcs nélküli protokoll**. Ennek egy javított változata a **Massey-Omura kriptorendszer**, ahol a csoport egy véges test, konkrétan egy 2^n-elemű test multiplikatív csoportja.

3. **AlGamal titkosítás** Megint legyen G egy nyilvánosan ismert n-edrendű ciklikus csoport a (szintén nyilvánosan ismert) g generátorelemmel. A választ egy, csak általa ismert $n > k_A \in \mathbb{N}^+$ érté-

ket, és meghatározza $u_A = g^{k_A}$-t. A továbbiakban k_A az A titkos, és u_A a nyilvános kulcsa. Ha B titkosított üzenetet akar küldeni A-nak, és m az üzenet (amely most G valamely eleme), akkor választ egy véletlen t egész számot, kiszámítja $r = g^t$-t valamint $s = u_A^t m$-et, és elküldi A-nak a $c = (r, s)$ párt. A ebből meg tudja határozni m-et, ugyanis $r^{n-k_A} s = (g^t)^{n-k_A} \left(g^{k_A}\right)^t m = g^{tn} m = m$. Ha viszont valaki nem ismeri k_A-t, de c-ből meg tudja határozni m-et, akkor m, $u_A = g^{k_A}$ és $r = g^t$ ismeretében meg tudja határozni $sm^{-1} = u_A^t = g^{k_A t}$-t, vagyis meg tudja oldani a Diffie-Hellman problémát.

Az AlGamal módszer sávszélesség-növekedéssel jár, hiszen m helyett egy hasonló méretű elemekből álló párt kell átvinni a csatornán. Ugyanakkor ez az eljárás randomizált: ugyanazon üzenetet különböző alkalmakkor ugyanannak a személynek elküldve minden egyes alkalommal más és más lesz a rejtjelezett üzenet, feltéve, hogy minden alkalommal egymástól függetlenül választott véletlen számot alkalmaz a küldő fél.

Ügyelni kell rá, hogy bármilyen is legyen az A-nak küldött üzenet, minden alkalommal más és más véletlenül választott számmal történjen a rejtés. Legyen ugyanis m_1 és $m_2 \neq m_1$ két üzenet, és mindkettőt titkosítsuk ugyanazon t kitevővel. Ekkor $\frac{m_1}{m_2} = \frac{s_1}{s_2} = s$ nem függ A titkos kulcsától, és ha az üzenet elegendően redundáns, akkor fejthető.

Az AlGamal rendszert széles körben alkalmazzák, az RSA mellett egy gyakran alkalmazott nyilvános kulcsú titkosító módszer. Ennek az elvnek fontos alkalmazása van a digitális aláírásnál is.

Vissza a tartalomhoz

11. Integritás, személyazonosítás, hitelesítés

Az aktív támadással szembeni védekezés során a következőkről van szó:

- a küldött üzenet integritásának ellenőrzése;
- a rendszerhez való hozzáférés jogosultságának ellenőrzése;
- a küldött üzenet hitelességének ellenőrzése.

1. Az üzenet **integritás**a annak sértetlenségét jelenti. Azt jelenti, amit úgy szoktak mondani, hogy „semmit el nem vettem belőle, és semmit hozzá nem tettem". Erre a célra úgynevezett **ujjlenyomat**ot, **kivonat**ot használnak, amelyet egy **hasítófüggvény**, másként egy **hash-függvény** állít elő. Az ilyen függvények tetszőleges hosszúságú karaktersorozatból egy fix hosszúságú karaktersorozatot állítanak elő, vagyis ha A egy véges ábécé (tehát $A \neq \emptyset$), akkor $h: A^+ \to A^n$, ahol $A^+ = \bigcup_{i=1}^{\infty} A^i$ az A ábécé betűivel felírható véges hosszúságú, nem üres szavak halmaza, és n egy rögzített pozitív egész szám. Ekkor $x \in A^+$-ra $h(x)$ az x **lenyomata** vagy **kivonata**. Egy hash-függvénytől még azt is elvárjuk, hogy könnyű legyen a kiszámítása.

A definícióból rögtön látható, hogy h biztosan nem injektív, hiszen a függvény értelmezési tartománya végtelen számosságú, míg a függvényértékek összessége egy véges halmaz részhalmaza, tehát véges. Ha x_1 és x_2 az A^+ két különböző eleme, de $h(x_1) = h(x_2)$, akkor ezt **ütközés**nek (másként **kollízió**nak – $collision$) mondjuk. Véges halmaz esetén is nézhetjük, hogy ha visszatevéssel húzunk a halmazból, mi a valószínűsége, hogy ugyanazt az elemet legalább kétszer kiemeltük. Amennyiben a halmaz elemeinek száma m, és r elemet választunk, akkor $r > m$ esetén biztosan van ismétlődés, így tegyük fel, hogy $r \leq m$. Ismétlés nélkül $\binom{m}{r} r!$ a lehetséges húzások száma, ennyiszer nincs ismétlés, míg az r elemet összesen m^r-féleképpen választhatjuk, tehát annak a valószínűsége, hogy a kihúzott elemek között van két egyforma, $p = 1 - \frac{\binom{m}{r} r!}{m^r} \approx 1 - e^{-\frac{r^2}{2m}}$. Ez $r = \sqrt{m}$ esetén körülbelül $p = 0{,}4$, vagyis például ha egy osztályban van $\sqrt{365} \approx 20$ tanuló, akkor 0,4-es (illetve ennél valamivel nagyobb) valószínűséggel lesz közöttük két olyan, akiknek az év ugyanazon napjára esik a születésnapja, és ha legalább 23-an vannak, akkor ez a valószínűség már meghaladja az $\frac{1}{2}$-et (ez a **születésnapi paradoxon**).

Az előbbiek szerint a hash-függvényekkel kapcsolatban feltételnek kell tekinteni, hogy gyakorlatilag egy $h(x)$-et egyértelműen azonosítson egy x (vagyis csak egy x-hez lehessen $h(x)$-et kötni), és ütközést számítástechnikailag nehéz legyen találni (vagyis a gyakorlatban lényegében véve soha ne forduljon elő).

A hash-függvény lehet

- **nem kulcsolt** (egyetlen bemenet az üzenet);
- **kulcsolt** (két, egymástól független bemenet az üzenet és a titkos kulcs).

A nem kulcsolt hash-függvények alosztálya az

1. **MDC** (***M**odification **D**etection **C**ode*, ***M**anipulation **D**etection **C**ode*), vagy **MIC** (***M**essage **I**ntegrity **C**ode*),

míg a kulcsolt hash-függvényeké a

2. **MAC** (***M**essage **A**uthentication **C**ode*).

Az MDC csupán az eredeti üzeneten végrehajtott módosítást jelzi, míg a MAC titkos kulcsú rendszerekben működik, és egyben hitelesítést is végez, amelyet úgy biztosít, hogy ehhez az eljáráshoz a feladó kulcsára van szükség. Az MDC egy m üzenethez egy $h(m)$ értéket, míg a *MAC* egy $h_k(m)$

értéket számít ki, ahol h a hasítófüggvény és k a kulcs. Látszólag könnyű egy nem kulcsolt hash-függvényből kulcsolt hash-függvényt készíteni úgy, hogy h-t $m \parallel k$-ra vagy $k \parallel m$-re alkalmazzuk (a kettős vonal a *konkatenáció*t, az egymás mellé írást jelöli), de ez a megoldás nem eléggé biztonságos. E helyett az egyik jó megoldás a **borítékmódszer kitöltéssel**, ahol $h(k \parallel p \parallel m \parallel k)$ lesz a kulcsolt függvény, míg a másik esetben $h\big(k \parallel p_1 \parallel h(k \parallel p_2 \parallel m)\big)$ állítja elő a kulcsolt függvényt (**hash-alapú** MAC). Mindkét esetben p kitöltő karaktersorozat. MAC-nél lényeges, hogy a gyakorlatban a k kulcs ismerete nélkül ne lehessen olyan (x, y) párt előállítani, ahol $y = h_k(x)$.

A nem kulcsolt hasítófüggvény egy tipikus alkalmazása, amikor egy adott m üzenetre kiszámítjuk $u = h(m)$-et (amely általában jóval rövidebb, mint m), ezt biztonságosan tároljuk, majd egy későbbi időpontban, amikor m-et fel akarjuk használni, az akkor rendelkezésre álló m'-ből meghatározzuk a kivonatát, és ha $h(m') = u\big(= h(m)\big)$, akkor feltesszük, hogy $m' = m$, vagyis az üzenet nem változott meg, az üzenetet nem manipulálták.

A hasítófüggvény **egyirányú** (***O**ne-**w**ay **h**ash **f**unction – **OWHF***), ha adott $y \in A^n$-hez lényegében nehéz olyan $x \in A^+$-t találni, amellyel $y = h(x)$ (a lényegében azt jelenti, hogy néhány x-re kiszámíthatjuk $y = h(x)$-et, és akkor ezekre az y-okra nyilván könnyen találunk megfelelő x-et). Ezt a tulajdonságot **ősrezisztencia**ként is említjük. E mellett az egyirányú hash-függvénytől a **második ősrezisztencia** is elvárt, ami azt jelenti, hogy számítástechnikailag egy adott x-hez nehéz (lényegében véve lehetetlen) legyen olyan $x \neq x'$-t találni, amellyel $h(x) = h(x')$. A hash-függvények másik csoportját alkotják az **ütközésrezisztens hash-függvény**ek (***C**ollision-**r**esistant **h**ash **f**unction – **CRHF***). Egy ilyen függvény esetén számítástechnikailag nehéz, gyakorlatilag lehetetlen olyan két különböző bemenetet találni, amelyek ugyanazt a kimenetet adják. Ettől a függvénytől általában elvárjuk, hogy egyúttal ősrezisztens is legyen. Szokásos más elnevezés is: a második ős-rezisztenciát **gyenge ütközésrezisztencia**ként, míg az ütközésrezisztenciát **erős ütközésrezisztencia**ként is említik.

Ha egy $h(x)$-re egy D digitális aláírást alkalmazunk, akkor h-nak rendelkeznie kell a második ős-rezisztenciával. Ellenkező esetben egy támadó megfigyelheti A aláírását $h(x)$-re, és ha $x \neq x'$-re $h(x) = h(x')$, akkor állíthatja, hogy A x'-t írta alá. Ha a támadónak még arra is van lehetősége, hogy megválassza azt az üzenetet, amelyet A aláír, akkor szükséges az ütközésrezisztencia, mert különben elegendő találnia egy x_1 és $x_2 \neq x_1$ párt, amelyekre $h(x_1) = h(x_2)$, és aláíratva az egyiket, később a két szöveg közül bármelyikre (amelyik számára az adott helyzetben kedvezőbb) mondhatja, hogy arra kapott A-tól aláírást. Ez utóbbi nyilván egyszerűbb, könnyebb, mint az előbbi, vagyis az ütközés-rezisztenciából következik a második ős-rezisztencia, hiszen ha egy rendszer nem második ős-rezisztens, akkor választva egy x-et, ehhez tudunk találni olyan x'-t, hogy $h(x) = h(x')$, és ekkor x és x' két különböző üzenet, amelyek kivonata azonos.

Az ütközésrezisztenciából azonban nem következik az ősrezisztencia. Ha például g egy ütközésrezisztens hash-függvény, amely n-betűs kimenetet generál, akkor legyen $h(x) = 1 \parallel x$, ha x hossza n, egyébként legyen $h(x) = 0 \parallel g(x)$. Ekkor h bármely bemenetből $n + 1$-hosszúságú kimenetet hoz létre, tehát hash-függvény, amely ütközésrezisztens, de nyilván nem ősrezisztens.

Kevésbé nyilvánvaló, hogy miért szükséges az ősrezisztencia bizonyos nyilvános kulcsú aláírásoknál. Egy példa az RSA mint aláírás. A támadó választ egy y-t, és kiszámítja A nyilvános kulcsával $z = y^e \bmod n$-et. Ha képes olyan x-et találni, amellyel $z = h(x)$, akkor mondhatja, hogy y az A aláírása az x üzenet z kivonatára (mert $y = z^d \bmod n$).

MDC esetén $h(m)$-et valamilyen módon védeni kell a támadótól, hiszen h normális körülmények között nyilvános. Ha az eredeti adat tárolása során felmerülő változások ellenőrzésére használjuk a kivonatot, akkor elegendő, ha ezt a kivonatot az eredeti adattól elkülönítve, biztonságos helyen tároljuk. Adatátvitel esetén az egyik lehetőség, hogy miközben m-et egy nyilvános csatornán küldjük, $h(m)$ egy biztonságos csatornán kerül átvitelre. Ha viszont $h(m)$-et m-mel együtt egy nyilvános csatornán küldjük, akkor gondoskodni kell arról, hogy $h(m)$-et ne lehessen az üzenet manipulálásával együtt, annak megfelelően változtatni. Az egyik lehetőség, hogy $h(m)$-et titkosítjuk a szimmetrikus kulcsunkkal, vagy aláírjuk a nyilvános kulcsú rendszerben, és ezt a titkosított vagy aláírt kivonatot mellékeljük m-hez (ez egyben már hitelesítés is), azaz $m \parallel E_k\big(h(m)\big)$-et vagy $m \parallel D_k\big(h(m)\big)$-et küldjük el (titkos kulcs esetén az előbbi, nyilvános kulcs esetén az utóbbi változatot). Más lehetőség, hogy

$m \parallel h(m)$-et titkosítjuk, vagyis a kivonatot egyszerűen az üzenet végéhez illesztjük, és az így toldalékolt szöveget sifrírozzuk. Amennyiben magát az üzenetet nem kell titkosítani, akkor ez utóbbi eljárás nem igazán hatékony, hiszen maga az üzenet általában lényegesen hosszabb, mint a kivonata. Az a megoldás pedig, ahol az üzenetet titkosítjuk, és a kivonatot közvetlenül a végéhez illesztjük, miközben az üzenet tartalma nem szorul titkosításra, hasonlóan gazdaságtalan, és az előbbihez képest általában nem jelent lényeges megtakarítást.

Egy f függvény **tömörítő függvény**, ha $f: A^m \to A^n$, ahol $m > n$. Egy ilyen függvény általában egyirányú. A tömörítőfüggvény felhasználható hasítófüggvény előállítására. Ekkor adott egy kezdeti érték, $H_0 = IV$ (***I**nitial **V**alue – **IV***), majd $1 \leq i \leq t$-re $H_i = f(H_{i-1}, x_i)$, és végül $h(x) = g(H_t)$, ahol $x = x_1 \parallel \cdots \parallel x_t$ (ha x_t hossza kisebb az elvártnál, akkor kitöltőkarakterekkel kell kiegészíteni). Bármely ütközésmentes *f* tömörítő függvény kiterjeszthető egy ütközésmentes hasítófüggvénnyé, vagyis elegendő ütközésmentes tömörítő függvényt találni. Ezt az állítást nem bizonyítjuk. Szintén nem bizonyítjuk, hogy ha h_1 és h_2 ütközésmentes hash-függvények, akkor $h(x) = h_1(x) \parallel h_2(x)$ is ütközésmentes.

Mivel a MAC esetén egy nem nyilvános kulccsal történik a kivonatolás, elegendő a kivonatot összefűzni az eredeti üzenettel. Egy MAC a k titkos kulccsal parametrizált h_k függvények családja, ha

- egyszerűen számítható;
- tömörít, azaz $h_k: A^+ \to A^n$;
- **számításrezisztens**.

Az utolsó feltétel azt jelenti, hogy adott $\big(x_i, h_k(x_i)\big)$-k ismeretében bármely $x \neq x_i$-re számítástechnikailag lehetetlen kiszámítani egy $\big(x, h_k(x)\big)$ párt, még abban az esetben is, ha esetleg valamely i-re $h_k(x) = h_k(x_i)$. Ha ez a feltétel nem teljesül, akkor az adott MAC esetén lehetséges a MAC-**hamisítás**. A hamisításnak két típusa van:

- **szelektív hamisítás**;
- **egzisztenciális hamisítás**.

A szelektív hamisító bármely x bemenethez képes a megfelelő $h_k(x)$ kivonatot előállítani, míg a másik esetben képes produkálni egy olyan (x, y) párt, hogy $y = h_k(x)$, de nem képes megválasztani ehhez az x-et.

A fenti harmadik feltételből következik a **kulcs-visszanemnyerés**, vagyis ha teljesül a harmadik feltétel, akkor hiába ismerünk egy adott, de nem ismert k kulcshoz $\big(x_i, h_k(x_i)\big)$ párokat, ezekből számítástechnikailag lehetetlen a k meghatározása, de ez visszafelé nem igaz.

Ismert MDC-algoritmusok az *MD*4, *MD*5 (***M**essage **D**igest algorithm*; az *MD*4 egyértelműen nem biztonságos, és a másikban is találtak ütközést, ezért nem javasolják a használatát), továbbá az *SHA*-1 (***S**ecure **H**ash **A**lgorithm*) és a *RIPEMD*-160 (***R**ACE* [***R**esearch and Development in **A**dvanced **C**ommunications Technology in **E**urope*] ***I**ntegrity **P**rimitives **E**valuation **M**essage **D**igest algorithm*). MAC-et például bármely blokkos rejtjellel elő lehet állítani *CBC*-üzemmódban, mint az utolsó blokk.

2. A következő kérdés a **személyazonosítás**, az **identifikáció**. Az információs biztonság megköveteli, hogy adott tevékenységet csak arra feljogosított személy végezhessen, vagyis a tevékenység megkezdése előtt igazolja a személyazonosságát.

Az identifikációs protokollok kapcsán az alábbi szempontok lehetnek érdekesek:

- kölcsönösség
  - egyirányú;
  - kétirányú;

- számítástechnikai hatékonyság;
- kommunikációs hatékonyság;
- harmadik fél valós idejű részvétele;
- a harmadik féltől elvárt megbízhatóság természete;
- a biztonsági garanciák természete;
- a titkok tárolása.

Különböző megoldási módszerek vannak, amelyek három fő csoportba sorolhatóak:

- az illető tud valamit (jelszó, PIN-kód (***P**ersonal **I**dentification **N**umber*), titkos kulcs);
- az illető birtokol valamit (mágneskártya, chipkártya, jelszógenerátor);
- az illető inherensen rendelkezik valamivel (kézírásos aláírás, ujjlenyomat, hang, szaruhártya-mintázat, a kéz geometriája).

Most az elsővel foglalkozunk.

Itt a leggyakoribb a **jelszavas identifikáció**. Minden résztvevő rendelkezik egy nyilvános **azonosító**val (***id**entifier* – ***id***) és titkos, csak általa és a rendszer által ismert **jelszó**val (***p**assw**o**rd* – ***pw***), amelyek egy **jelszófájl**ban vannak. Amikor valaki be akar lépni a rendszerbe, megadja az azonosítóját és a jelszavát, amelyet a rendszer ellenőriz, vagyis ellenőrzi, hogy a beadott azonosító él-e rendszerben, és ha igen, akkor az adott azonosítóhoz valóban az éppen bemutatott jelszó tartozik-e. Amennyiben igen, úgy a rendszer elfogadja a jelentkezőt, ellenkező esetben elutasítja. Mivel a titkosnak feltételezett jelszó nyílt szövegként szerepel a jelszófájlban, ezért ennek a fájlnak olvasásvédettnek kell lennie, és nyilván írásvédettnek is, hiszen ellenkező esetben bárki módosíthatná akárkinek a jelszavát.

Az előbbi módszerrel szemben támasztott elsődleges kifogás, hogy a jelszófájlhoz többen is hozzáférnek legálisan (például a rendszergazdák), akik esetleg illetéktelenül felhasználják a fájlban szereplő személyek jelszavait. Ez ellen egy egyszerű védekezés, ha a jelszófájlban nem maga a jelszó, hanem annak egy transzformált változata található. Legyen f egy egyirányú függvény. Ha most id mellett a pw helyett $f(pw)$ van a jelszófájlban, akkor hiába olvassa ki valaki ezt az értéket, a bejelentkezésnél nem $f(pw)$-t, hanem pw-t kell megadni, és ezt $f(pw)$-ből – bár elvileg igen, de gyakorlatilag – lehetetlen meghatározni. Ennél a változatnál elegendő, ha a jelszófájl írásvédett, az olvasás nem ad értékelhető új információt illetéktelen személyeknek.

Az előbbi utolsó megállapítás feltételezi, hogy mindenki kellő óvatossággal választja meg jelszavát. Önmagában a hossz növelése nem elegendő, a cél az, hogy az entrópia legyen nagy. Ha nem így van, akkor valakinek a jelszava viszonylag könnyen, kevés számú próbálkozással meghatározható, és a nevében más be tud lépni a rendszerbe. Az ilyen könnyű jelszavakat gyakran **jelszó-szótár**ba gyűjtik, és a rendszerre rácsatlakozva automatikusan keresnek olyan felhasználót, akinek a jelszava szerepel a szótárban (**szótár alapú támadás** – *dictionary attack*). Ez ellen egy védekezési mód, ha a jelentkezésre adott választ lelassítjuk, a választ **késleltetjük** olyan mértékben, hogy a legális felhasználó ezt ne vegye észre, de ésszerű idő alatt csak nagyon kevés számú jelszót lehessen kipróbálni.

A szótár alapú támadás elleni másfajta védekezési lehetőség a **sózás** (*salting*). Ennél a megoldásnál a jelszófájlban egy adott felhasználó sorában három adat található: az azonosítója, egy elegendően hosszú s véletlen szám, a „**só**”, valamint $f(s \parallel pw)$. Ha az id azonosítóhoz tartozó felhasználó bejelentkezik, akkor megadja a jelszavát. A rendszer a jelszófájlból kiolvassa az adott azonosítóhoz tartozó sót, és kiszámítja a beküldött pw'-vel $f(s \parallel pw')$-t. Ha $f(s \parallel pw') = f(s \parallel pw)$, akkor elfogadja, hogy $pw' = pw$, ellenkező esetben elutasítja a bejelentkezőt

Ha valaki egy adott azonosítóhoz tartozó jelszót szeretné a szótáralapú támadással meghatározni, akkor ez ellen a sózás nem nyújt védelmet, hiszen a só nyílt szövegként van a jelszó-fájlban, így az onnan kiolvasott sót konkatenálva a jelszó-fájl bejegyzéseivel, végig lehet próbálni a fájl bejegyzéseit. Más a helyzet, ha csupán azt akarjuk elérni, hogy a szótár elemeit egymás után kipróbálva, találjunk valakit, akinek a jelszava azonos a vizsgált jelszóval. Ebben az esetben a sózás ezt a tevékenységet nagyban megnehezíti, sőt, elegendően hosszú só esetén lényegében véve lehetetlenné teszi. Ekkor ugyanis nem tudjuk, hogy a keresett jelszóhoz milyen véletlen szám tartozik, így a szótárban szereplő minden egyes jelszót az összes lehetséges, a sókkal azonos hosszúságú véletlen számmal ki kell próbálni. Ha a só *t* bit hosszúságú, akkor ez egy-egy kipróbálandó jelszó esetén 2^t vizsgálatot jelent.

A jelszó-alapú identifikációnál egy jelentős problémát okoz, hogy ha valaki lehallgatja a beadott, tehát még nem transzformált jelszavunkat, akkor onnan kezdve bármikor be tud jelentkezni a nevünkben. Az ilyen támadás ellen véd az **egyszer felhasználható jelszó**. Erre egy lehetőség, amikor az aktuális jelszóból számolt kulccsal titkosítva, az adott jelszóval folytatott kommunikáció során küldjük be a rendszernek a következő alkalomra érvényes jelszavunkat. Egy másik módszernél választunk egy tetszőleges w_0 értéket, és ebből kiindulva $0 \leq i < t$-re kiszámítjuk a $w_{i+1} = h(w_i)$ értéket, ahol h egy egyirányú függvény, majd elküldjük a rendszernek a (t, w_t) párt. Az első bejelentkezésünkkor jelszóként (t', w')-t küldjük, ahol, ha valóban mi vagyunk az adott azonosítóhoz tartozó felhasználók, vagyis ismerjük a soron következő jelszót, $t' = t$ és $w' = w_{t-1}$. A rendszer leellenőrzi, hogy valóban teljesül-e az előbbi két egyenlőség. Ha igen, akkor feltételezi, hogy legális a bejelentkezés, és beengedi a bejelentkezőt a rendszerbe, és (t, w_t)-t kicseréli $(t-1, w_{t-1})$-re, ellenkező esetben elutasítja a bejelentkezést. Hasonlóan, ha az adott azonosítóhoz a jelszófájlban az (i, w_i) bejegyzés található, akkor az aktuális jelszó az (i, w_{i-1}) pár lesz, feltéve, hogy $i > 0$, és sikeres bejelentkezés után az új bejegyzés $(i-1, w_{i-1})$ lesz. Amennyiben $i - 1 = 0$, akkor elfogyott a jelszó-készletünk, és új jelszósorozatot kell generálnunk.

A fentiek szerint a jelszavas személyazonosítással szembeni leggyakoribb támadások

- a jelszó rendszeren kívüli felfedezése;
- a vonal lehallgatása (a rendszeren belül);
- a jelszó kitalálása.

Az első kettőre a **visszajátszás**, míg a harmadikra a szótár alapú támadás egy lehetőség. Általánosabban az identifikációval szembeni tipikus támadási módok az alábbiak:

- megszemélyesítés;
- visszajátszás;
- összefésülés;
- visszatükrözés (egy összefésült üzenet visszaküldése a feladónak);
- kikényszerített késleltetés;
- választott szövegű támadás.

A jelszavas identifikációnál erősebb módszer a **kihívás – válasz** (*challenge and response*). Fontos eleme ennek a fajta eljárásnak, hogy egy **időváriáns paramétert** használ.

Az alkalmazott módszer szerint az eljárás alapja

- szimmetrikus kulcsú rejtjelező rendszer;
- kulcsolt egyirányú hash-függvény;
- nyilvános kulcsú rejtjelező rendszer, ezen belül
  - titkosítás;
  - digitális aláírás.

Más szempontból az eljárás lehet

- egyirányú;
- kétirányú (kölcsönös).

A kihívás (az időváriáns érték) lehet

- időkeret (**időpecsét**tel);
- véletlen szám;
- sorszám.

A 6. táblázatban röviden megadunk az előbbi osztályokhoz néhány lehetséges megoldást.

A $SKID2$ és $SKID3$ szabvány.

Az időpecsételt üzenet akkor érvényes, ha az eltelt idő egy meghatározott időablakon belül van. Opcionálisan még az is szükséges lehet, hogy ugyanezzel az időpecséttel ugyanezen felhasználótól nem érkezett másik bejelentkezési igény.

Szimmetrikus kulcsú	Kulcsolt egyirányú hash-függvény	Nyilvános kulcsú
egyirányú		
időpecséttel (t)		
$A \to B: E_k(t_A \| B^*)$	$A \to B: t_A \| h_k(t_A \| B^*)$	$A \to B: cert_A \| t_A \| B \| S_A(t_A \| B)$
véletlen számmal (r)		
(1) $A \leftarrow B: r_B$ (2) $A \to B: E_k(r_B \| B^*)$ vagy (2) $A \to B: E_k(r_A \| r_B \| B^*)$	(1) $A \leftarrow B: r_B$ (2) $A \to B: h_k(r_B \| B^*)$ vagy (2) $A \to B: r_A \| h_k(r_A \| r_B \| B^*)$ ($SKID2$;ekkor B kötelező)	(1) $A \leftarrow B: r_B$ (2) $A \to B: cert_A \| r_A \| B \| S_A(r_A \| r_B \| B)$ vagy (1) $A \leftarrow B: h(r) \| B \| P_A(r \| B)$ (2) $A \to B: r$
kétirányú		
véletlen számmal (r)		
(1) $A \leftarrow B: r_B$ (2) $A \to B: E_k(r_A \| r_B \| B^*)$ (3) $A \leftarrow B: E_k(r_B \| r_A)$	(1) $A \leftarrow B: r_B$ (2) $A \to B: r_A \| h_k(r_A \| r_B \| B^*)$ (3) $A \leftarrow B: h_k(r_B \| r_A)$ vagy (2) $A \to B: r_A \| h_k(r_A \| r_B \| B)$ (3) $A \leftarrow B: h_k(r_B \| r_A \| A)$ ($SKID3$)	(1) $A \leftarrow B: r_B$ (2) $A \to B: cert_A \| r_A \| B \| S_A(r_A \| r_B \| B)$ (3) $A \leftarrow B: cert_B \| A \| S_B(r_B \| r_A \| A)$ vagy (1) $A \to B: P_B(r_A \| A)$ (2) $A \leftarrow B: P_B(r_A \| r_B)$ (3) $A \to B: r_B$
Megjegyzés: a *-gal jelölt adat opcionális		

6. táblázat

Az opcionális B paraméter megadása elsősorban az időpecsét mellett lényeges. Ezzel az úgynevezett **visszajátszás**t lehet megakadályozni. Ha ugyanis csak az időpecsétet adjuk meg, úgy egy támadó a lehetséges időkereten belül az A által B-nek küldött $E_k(t_A)$-t visszaküldve A-nak, B-ként sikeresen azonosítja magát A-nál. A fogadó fél azonosítójának jelenléte az elküldött időpecsét mellett ezt megakadályozza. Hasonlóan kivédhető a visszajátszás, ha más a kulcs a két irány esetén.

$E_k(r_B \| B^*)$ helyett $E_k(r_A \| r_B \| B^*)$-t, illetve $h_k(r_B \| B^*)$ helyett $r_A \| h_k(r_A \| r_B \| B^*)$-t az indokolja, hogy meggátoljuk a választott nyílt szövegű támadást. Hasonlóan, a nyilvános kulcsú eljárásnál $h(r)$-re a $h(r) \| B \| P_A(r \| B)$ üzenetnél a választott rejtjelszövegű támadás kivédésére van szükség, hiszen ha valaki r ismerete nélkül választja y-t $P_A(r \| B)$-ként, hogy aztán visszakapja ebből r-et, akkor nem tudja megadni $h(r)$ értékét, ahol h egy egyirányú hash-függvény. $h(r)$ a **tanú**, míg $P_A(r \| B)$ a **kihívás** (P a nyilvános kulcsú titkosító algoritmus).

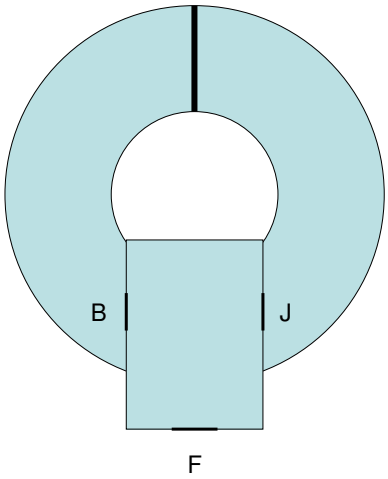
A nyilvános kulcsú aláírásoknál $cert$ az ellenőrző algoritmus, amely nyilván elmaradhat, ha ez nyilvánosan hozzáférhető, vagy ha az ellenőrző fél rendelkezik vele, míg S az aláíró algoritmus.

Kihívás-válasz típusú identifikációt használnak például katonai repülőgépeken a barát - ellenség felismerésére (*IFF - **I**dentification **F**riend or **F**oe*).

Most a legerősebb módszerrel, a *ZKP*-vel, a **nulla-ismeretű protokoll**al (***Z**ero **K**nowledge **P**rotocol*) foglalkozunk röviden.

A *ZKP* lényege, hogy az ellenőrző személy csupán egyetlen bitnyi információ birtokába jut az azonosítás végén, nevezetesen, hogy A az-e, akinek mondja magát. A megoldást a 11. ábra segítségé-

vel lehet megérteni. Az ábrán **B**, **J** és **F** ajtók, míg fölül a vastag vonal egy falat reprezentál. A azt állítja, hogy keresztül tud menni ezen a falon, és erről meg akarja győzni B-t, de úgy, hogy nem akarja megmutatni neki a trükköt. Az eljárás a következő. A az **F** ajtón keresztül belép az épületbe, majd becsukja az ajtót, és vagy **B**-n, vagy **J**-n megy tovább, becsukva maga mögött ezt az ajtót is. Ezek után B belép **F**-en keresztül az előtérbe, és szól A-nak, hogy jöjjön ki mondjuk a **J** ajtón keresztül. Ha A valóban keresztül tud menni a falon, akkor bármelyik oldalon is ment be az épület belsejébe, ki tud jönni **J**-n keresztül. Persze akkor is ki tud itt jönni, ha nem igaz, amit állított, de éppen ezen az ajtón ment be, vagyis ebben az esetben is van 50%-nyi sansza a sikerre. Ha azonban pechére a másik oldalon ment be, akkor lebukik. Ez azt jelenti, hogy elég nagy esélye van arra, hogy nem bukik le (pontosan akkora, mint annak, hogy lebukik). Meggyőző ez az eredmény? Ha elbukott, akkor igen, de ha sikerrel vette az akadályt, akkor nem túlságosan. Igen ám, de ha mondjuk tíz egymás utáni kísérlet mindegyikében a jó oldalon jelenik meg, akkor már csak 1 az 1000-hez (pontosabban az 1024-hez) az esélye, hogy mindegyik alkalommal jól teljesít, és ha még ez sem elég, akkor ennél is több próbát kérhet B. Ha összesen n fordulót játszanak le, akkor 2^{-n} annak a valószínűsége, hogy egy csalónak mindig szerencséje van, vagyis, hogy mindig előre megérzi, honnan kell majd kijönnie. Egy szemernyi kétség mindig maradhat B-ben, ha nagyon nem akar hinni A-nak, de azért a józan ész mégiscsak hajlik arra, hogy elegendően sok kísérlet után elhiggye, A valóban keresztül tud menni a falon.



11. ábra

Egy „matematikusabb” bemutatás például a következő. Ismeretes, hogy a Hamilton-kör problémája NP-teljes feladat. A azt állítja, hogy könnyen talál Hamilton-kört a G gráfban, de ezt úgy igazolja, hogy magát a Hamilton-kört nem mutatja meg. Ehhez csinál egy, a G-vel izomorf G' gráfot, és ezt elküldi az ellenőrzőnek, B-nek. Ez utóbbi ekkor két dolgot kérhet (a kettő közül az egyiket): vagy azt, hogy A adja meg az izomorfizmust adó leképezést, vagy mutasson egy Hamilton-kört G'-ben. Amennyiben A igazat mondott, akkor mindkét esetben helyesen tud válaszolni, hiszen az izomorf leképezést maga választotta, míg a másik esetben, ha megtalált egy Hamilton-kört G-ben, akkor az izomorf leképezéssel ismeri a G'-beli Hamilton-kört. Ha viszont A hamisan állította, hogy képes könnyen megtalálni egy Hamilton-kört, akkor arra a kérésre, hogy mutasson G'-ben egy Hamilton-kört, nem tud helyesen válaszolni (mert ha tudna, akkor G-ben is ismerné az ennek megfelelő kört). De persze A megpróbálhat csalni úgy, hogy előre megsaccolja, mit kívánhat B. Ha azt gondolja, hogy az izomorfizmust kérik tőle, akkor ugyanúgy járhat el, mint a fentiekben, vagyis elkészíti a G-vel izomorf G'-t, és elküldi B-nek. Ha most B, az A várakozásának megfelelően, az izomorfizmust kéri, akkor erre A helyesen tud válaszolni, de nem tud jó választ adni, ha a másik kérést kapja. Hasonlóan, ha arra számít, hogy a kört kérik tőle G'-ben, akkor nem egy, a G-vel izomorf gráfot küld el, hanem vesz egy G-vel azonos pontszámú kört, és ezt kiegészíti egy olyan gráffá, amelyben a fokszámsorozat azonos G fokszám-sorozatával, és más hasonló, könnyen ellenőrizhető jellemzőkben azonos G-vel, és ezt küldi el. Ha a kérés a kör felmutatása, az könnyedén sikerül, de bajban lesz A, ha az izomorfizmust kell felmu-

tatnia, hiszen az elküldött gráf minden bizonnyal nem izomorf G-vel, viszont a gráfizomorfizmus problémája, bár nem NP-teljes, de szintén nehéz kérdés.

A módszer általános sémája:

1. A választ egy **tanú**t, és elküldi B-nek (választ egy véletlen számot, és ebből kiszámítja a tanút egy egyirányú függvénnyel);
2. B visszaküld egy **kihívás**t;
3. A elküldi a választ.

Ez egy **vág és választ** (*cut and choose*) protokoll: az egyik gyerek vágja a tortaszeleteket, míg a másik választ, így biztosítva az igazságos elosztást.

Az identifikációs protokollal szembeni elvárások a következőek:

- ha A és B becsületes, A sikeresen tudja magát igazolni B-vel szemben;
- B ne legyen képes A egy korábbi azonosítási eljárását felhasználva A-ként azonosítani magát C-vel szemben;
- elhanyagolható legyen annak a valószínűsége, hogy egy A-tól különböző C magát A-ként igazolja B-vel szemben;
- az előbbiek akkor is teljesüljenek, ha C (polinomiálisan) sok korábbi, A és B közötti identifikációt figyelt meg, vagy korábban akár A-val, akár B-vel részt vett a protokollban, illetve, ha szimultán több folyamat résztvevője lehet C.

Az utolsó pontban a szimultán részvételre a példa, amikor egy sakk-csaknem-analfabéta két nagymesterrel játszik egyszerre, egyikük ellen a világos, míg másikuk ellen a sötét bábukkal (**sakk nagymester probléma** – *Chess Grandmaster Problem*). Ekkor valahányszor a világos bábut vezető ellenfele lép egyet, azt ő rögtön meglépi a másik táblán, és az erre kapott választ adja az előbbi táblán. Ilyen módon a két játékból 50%-os eredményt ér el a két nagymesterrel szemben, mert vagy mindkét partija döntetlen, vagy az egyik táblán győz, a másikon veszít (tulajdonképpen a két nagymester egymás ellen játszik, a mi emberünk csupán a „bábutologató”, és ez is középre állásos támadás).

Egy ilyen identifikációs algoritmus a *Fiat-Shamir protokoll*. Itt van mondjuk egy közös $n = pq$ modulus, ahol p és q különböző páratlan prímszám, minden résztvevőnek van egy titkos i, és nyilvános $s = i^2 \bmod n$ azonosítója. A úgy akarja igazolni magát B felé, hogy nem árulja el i-t. Ezt, mint a fentebbi példában, több fordulóban hajtja végre (annyiban, amennyit B óhajt – de azért az észszerűség határain belül). A minden fordulóban elküld B-nek egy u számot, amely, ha A becsületes, akkor egy általa ebben a fordulóban választott és titokban tartott r véletlen szám négyzetének a maradéka, vagyis $u = r^2 \bmod n$. Ekkor B visszaküld A-nak egy általa tetszés szerint választott b bitet, mire A-nak az a feladata, hogy elküldje B-nek $ri_A^b \bmod n$-et. Tegyük fel, hogy A egy v-t küldött most. B-kiszámítja $v^2 \bmod n$-et, és ezt az értéket egybeveti $us_A^b \bmod n$-nel. Ha A tényleg az, akinek mondja magát, akkor ismeri i_A-t, és becsületesen játszik, vagyis ekkor

$$\begin{aligned} v^2 \bmod n &= \left(ri_A^b \bmod n\right)^2 \bmod n = r^2 (i_A^2)^b \bmod n \\ &= (r^2 \bmod n)(i_A^2 \bmod n)^b \bmod n = us_A^b \bmod n. \end{aligned}$$

Amennyiben viszont A' nem A, csak annak mondja magát, akkor csak 50%-nyi esélye van minden fordulóban, hogy átmegy a teszten. Ha arra tippel, hogy B $b = 0$-t mond, akkor az első körben szabályosan elküldi a választott véletlen szám négyzetének maradékát, és ha B valóban a 0-ás bitet küldi, akkor A' vissza tudja küldeni r-et. Ha viszont B 1-est küld, akkor bajban lesz a hamis A', hiszen nem ismeri i_A-t, és jelenlegi tudásunk szerint összetett modulus esetén, a faktorok ismerete nélkül gyakorlatilag lehetetlen a négyzet maradékából az eredeti számot meghatározni, vagyis bukik. Ha viszont 1-re számít, akkor ravaszul u-ként nem $r^2 \bmod n$-et, hanem $v = \frac{r^2}{s_A} \bmod n$-et küldi B-nek (s_A relatív prím

n-hez, mert ha nem az, akkor n-nel való legnagyobb közös osztója vagy p vagy q, és ezzel bárki ki tudja számolni bárkinek a titkos azonosítóját a megfelelő nyilvános adatból, ugyanis prímszám modulus esetén a moduláris gyökvonás könnyű feladat). Ha b valóban 1, akkor A' a második körben r-et küldi vissza, és B az ellenőrzésnél egyezőséget talál. Ám, ha a b most A' számítása ellenére 0, akkor bajban lesz az ál-A, mert most olyan t számot kellene küldenie, amellyel $t^2 \bmod n = v$, vagyis egy moduláris gyökvonást kellene végrehajtania egy összetett modulusra nézve, amelynek nem ismeri a felbontását.

A *Feige-Fiat-Shamir protokoll* az előbbi módszerhez hasonló. Most mind p, mind q 3-mal kongruens modulo 4, azaz $n = pq$ egy Blum-egész. Ekkor $\left(\frac{-1}{n}\right) = 1$, de -1 kvadratikus nemmaradék, és az n-hez relatív prím bármely u egész esetén u és $-u$ közül pontosan az egyik kvadratikus maradék. Most A választ $s_1, \cdots, s_l$, az n-hez relatív prím egészeket valamint $b_1, \cdots, b_l$ biteket, ezek lesznek a titkos azonosítói, és kiszámolja $1 \leq i \leq l$-re a $v_i = (-1)^{b_i}\left(s_i^2\right)^{-1} \bmod n$ nyilvános azonosítóit. Amikor igazolni akarja magát B-nél, akkor választ egy r véletlen számor és egy b bitet, és elküldi B-nek $t = (-1)^b r^2 \bmod n$-et. B visszaküld egy $e_1, \cdots, e_l$ bitsorozatot, amire A $u = r\prod_{i=1}^{l} s_i^{e_i} \bmod n$-nel válaszol. B ellenőrzi, hogy teljesül-e a $\pm u^2 \prod_{i=1}^{l} v_i^{e_i} \bmod n = t$ egyenlőség. Ha nem, akkor nem fogadja el a bejelentkezőt A-ként, ellenkező esetben újabb kört kezdeményezhet, vagy elfogad.

A protokollban a b_i bitek szerepe, hogy az összes olyan szám előfordulhasson, amelynek a Jacobi-szimbóluma 1, ne csak a kvadratikus maradékok. Ily módon az n-hez relatív prím, nála kisebb nem negatív egészek fele előfordulhat, míg a kvadratikus maradékok száma ennek csupán a fele.

3. Az identifikáció csak egy adott pillanatban, egy rövid ideig azonosít egy személyt, míg az integritás biztosítása önmagában egyáltalán nem biztosítja az adott dokumentum hitelességét. Ezt a feladatot az **aláírás** oldja meg. Az aláírással szembeni elvárásaink az alábbiak:

- legyen hiteles;
- legyen hamisíthatatlan;
- ne lehessen újra felhasználni;
- ne lehessen az aláírt dokumentumot megváltoztatni;
- ne lehessen az aláírást letagadni.

A **digitális aláírás** lényegesen különbözik a hagyományos aláírástól. Az utóbbi független a dokumentum tartalmától, és éppen azt várják el az aláírótól, hogy különböző időpontban más és más dokumentumon elhelyezett kézjegye nagyjából legyen azonos. Ezzel szemben az elektronikus aláírás tartalomfüggő, vagyis az aláírás különböző dokumentumokon szinte biztosan más lesz, és ez jelentősen megnehezíti a hamisító dolgát. A másik oldalon viszont a kézírásos aláírás a hordozóhoz rögzített, míg a kriptográfiai aláírás bármikor áthelyezhető egy adathordozóról egy másikra, ezért nagyon lényeges, hogy tényleg erősen függjön az aláírás az aláírt dokumentum tartalmától.

A klasszikus rejtjelezés esetén a titkosítás egyben aláírás is, hiszen csak a feladó ismerhette a rejtjelező kulcsot (feltéve, hogy minden kulcsot csak egy küldő ismer). A nyilvános kulcsú rendszer esetén viszont a titkosítás semmilyen kapcsolatot nem biztosít a kulcs gazdájával, hiszen az nyilvános, bárki által hozzáférhető, ezért itt a titkosítás nem jelent egyben hitelesítést is.

A digitális aláírásnak két nagy csoportja van:

- **toldalékos**;
- **üzenet-visszanyeréses**.

Az előbbinél nem a teljes üzenetet írjuk alá, hanem annak csak a kivonatát, ami gyorsítja az eljárást. Ekkor az üzenettel együtt elküldjük az aláírt kivonatot is, és a címzett a megkapott üzenet kivonatát egybevetheti a kapott, aláírt kivonattal. A második módszer esetén a teljes üzenetet írjuk alá. Ekkor azonban megfelelő óvintézkedést kell tennünk. Tegyük fel, hogy az aláírásra a jól ismert *RSA*-t

használjuk, „fordított” üzemmódban. Ekkor az m üzenet A által aláírt példánya $m' = m^{d_A} \bmod n_A$, amit valóban csak a legális küldő tud kiszámítani, és amiből a címzett könnyen ellenőrizni tudja, hogy tényleg A küldte-e, és időközben nem módosult-e az üzenet. Ehhez A nyilvános kulcsát kell használnia, hiszen $m'^{e_A} \bmod n_A = m$, de ha a számítást nem A titkos kulcsával végezték, vagy módosították az aláírt üzenetet, akkor már (szinte biztosan) nem fog teljesülni az egyenlőség. Igen ám, de az a baj, hogy most B nem tudja, mi volt m, így nem tudja ellenőrizni, hogy nem történt-e változás. A megoldás, hogy az aláírás előtt redundanciát viszünk az üzenetbe, olyan redundanciát, amelyet az aláírt, vagy hamisan aláírt üzenettel nem lehet (vagy csak nagyon vak tyúk alapon lehet) elérni. Tipikusan ilyen redundancia, hogy az üzenetet „dadogósan”, kétszer egymás mellé másolva írjuk le, és ezt írjuk alá, erre alkalmazzuk a titkos kitevőnket. A redundanciát egy R injektív függvénnyel hozzuk létre, amely az aláírandó üzenetek halmazát egy annál (általában lényegesen) nagyobb elemszámú halmazba képezi le. Ha az aláíró algoritmus multiplikatív, azaz $S(m_1 m_2) = S(m_1)S(m_2)$, ahol m_1 és m_2 két üzenet, akkor lényeges, hogy szinte minden m_1, m_2 párra $R(m_1 m_2) \neq R(m_1)R(m_2)$ teljesüljön. Ha ez nem teljesül, akkor $s(m) = (SR)(m_1 m_2) = (SR)(m_1)(SR)(m_2) = s(m_1)s(m_2)$, és ekkor egy támadó különböző üzeneteket „össze tud gyúrni”, össze tud ragasztani.

Az **elektronikus aláírás**ról szóló törvény a digitális aláírás biztonsága szempontjából három fokozatot különböztet meg:

- elektronikus aláírás;
- fokozott biztonságú elektronikus aláírás;
- minősített elektronikus aláírás.

A törvény megfogalmazása szerint

- *Elektronikus aláírás:* elektronikus dokumentumhoz azonosítás céljából logikailag hozzárendelt és azzal elválaszthatatlanul összekapcsolt elektronikus adat, illetőleg dokumentum.
- *Fokozott biztonságú elektronikus aláírás:* elektronikus aláírás, amely megfelel a következő követelményeknek:

  a) alkalmas az aláíró azonosítására, és egyedülállóan hozzá köthető,
  b) olyan eszközzel hozták létre, amely kizárólag az aláíró befolyása alatt áll,
  c) a dokumentum tartalmához olyan módon kapcsolódik, hogy minden – az aláírás elhelyezését követően az iraton, illetve dokumentumon tett – módosítás érzékelhető.

- *Minősített elektronikus aláírás:* olyan – fokozott biztonságú – elektronikus aláírás, amely biztonságos aláírás-létrehozó eszközzel készült, és amelynek hitelesítése céljából minősített tanúsítványt bocsátottak ki.

A törvényhez kiadott irányelvek szerint a minősített aláíráshoz az *RSA*-t és a *DSS*-t (***D**igital **S**ignature **S**tandard*, a *DSA* mint szabvány neve) ajánlott alkalmazni, ez utóbbinak az *elliptikus görbés* változatát is. Az *RSA* üzenet-visszanyeréses, bár az ilyen típus mindig használható a másik üzemmódban is, míg a második algoritmus toldalékos, hiszen fix hosszúságon dolgozik. Az üzenet-visszanyeréses technikánál, ha szükséges, az aláírt üzenetet titkosíthatjuk a címzett nyilvános kulcsával, míg a másik módszer esetén ilyenkor az üzenethez láncolt aláírással együtt titkosítjuk az üzenetet. Az *RSA* esetén tehát ekkor $E_{e_B}\left(D_{d_A}(m)\right) = \left(m^{d_A} \bmod n_A\right)^{e_B} \bmod n_B$-t küldi A a másik félnek, B-nek. Ha a két modulus különböző, márpedig majdnem mindig ez a helyzet, akkor $n_B < n_A$ esetén problémás a titkosítás, amint azt könnyű végiggondolni. Erre egy megoldás, hogy aláírás után áttördeljük a szöveget. Más megoldás lehetne a következő. Első ránézésre úgy tűnhet, hogy az előbb megadott transzformáció helyett $D_{d_A}\left(E_{e_B}(m)\right) = (m^{e_B} \bmod n_B)^{d_A} \bmod n_A$-t is küldhetné A, B ugyanúgy helyre tudná állítani az eredeti üzenetet. Ekkor azonban egy aktív támadó A nyilvános kulcsával kiszámolhatná $E_{e_B}(m)$-et, és utána a saját kulcsával aláírva a levelet továbbítaná azt B-nek. Ha a titkosított szöveg

nem utal A-ra, akkor B azt hiszi, hogy C volt a feladó, és például válaszolva neki, C fontos információk birtokába juthat, vagyis lényeges a két transzformáció sorrendje.

Egy jó megoldás az előbbi problémára, ha van egy közös K korlát, és minden résztvevő úgy választja meg a paramétereit, hogy az aláíráshoz használt n modulusa legyen kisebb, mint K, míg a titkosításnál alkalmazott modulus legyen K-nál nagyobb.

A toldalékos módszerhez két algoritmust ismertetünk.

Elsőként a *DSA*-t nézzük. Az elnevezés a ***D*igital *S*ignature *A*lgorithm** (*Digitális Aláírási Algoritmus*) kezdőbetűiből származik. Maga az eljárás szabványosított, a szabvány neve ***D*igital *S*ignature *S*tandard** (*DSS*; *Digitális Aláírási Szabvány*). Az eljárás hash-függvényként a 160-bites SHA-1 algoritmust alkalmazza.

Ha A egy résztvevő a rendszerben, akkor választ egy 160-bites q_A prímszámot, azaz egy olyan prímet, amelyre $2^{159} < q_A < 2^{160}$. Csebisev tétele szerint minden szám és a kétszerese között van prímszám, így ilyen prímet lehet találni. Választ egy $8 \geq t_A \in \mathbb{N}$ egészt, amely a rendszer bonyolultságát határozza meg, A választásától függően, majd választ egy $2^{511+64t_A} < p_A < 2^{512+64t_A}$ prímet, vagyis egy, a t_A-tól függő nagyságú, legalább 512, legfeljebb 1024 bites prímet úgy, hogy $q_A | p_A - 1$. Most a $\mathbb{Z}_{p_A}$ egy tetszőleges, $g \neq 0$ elemével kiszámolja $g^{\frac{p_A-1}{q_A}} \bmod p_A$-t. Ha ez 1, akkor új g-vel számolunk, amíg 1-től különböző értéket nem kapunk. Legyen ez α_A. Most A választ egy $q_A > a \in \mathbb{N}^+$ egészt, és kiszámolja $y_A = \alpha_A^a \bmod p_A$-t. A titkos kulcsa a, míg az aláírása ellenőrzéséhez szükséges nyilvános adatok $(p_A, q_A, \alpha_A, y_A)$. Ezzel a rendszer paraméterei rendelkezésre állnak.

Legyen az A által aláírandó üzenet m. A választ egy véletlen $q_A > k \in \mathbb{N}^+$ egészt. k választásánál ügyelni kell arra, hogy ne ismétlődjön, mert ismétlődés esetén, ha valaki észreveszi, fel tudja törni a rendszert, hozzá tud jutni A titkos adatához, meg tudja határozni a-t, amint majd később, az általánosított AlGamal rendszernél ezt belátjuk. k választása után kiszámítja $r = \left(\alpha_A^k \bmod p_A\right) \bmod q_A$-t valamint $\kappa = k^{-1} \bmod q_A$-t, és végül $s = \kappa(h(m) + ar) \bmod q_A$-t (ügyelve arra, hogy ez utóbbi ne legyen 0). Az aláírás ekkor m-re az (r, s) pár.

Az előbbiekből következik, hogy ha $w = s^{-1} \bmod q_A$, akkor $k = w(h(m) + ar) \bmod q_A$, tehát $\alpha_A^k \equiv \alpha_A^{wh(m)} y_A^{wr}\ (p_A)$, és így $r = \left(\alpha_A^{wh(m)} y_A^{wr} \bmod p_A\right) \bmod q_A$

Ha egy $\left(m', (r', s')\right)$ pár állítólag egy üzenet A aláírásával, akkor az ellenőrzésnél először megnézzük, hogy teljesül-e $q_A > r' \in \mathbb{N}^+$ és $q_A > s' \in \mathbb{N}^+$. Ha nem, akkor nem fogadjuk el az aláírást. Ellenkező esetben kiszámítjuk $w = {s'}^{-1} \bmod q_A$-t, $u_1 = wh(m') \bmod q_A$-t és $u_2 = r'w \bmod q_A$-t, majd ellenőrizzük, hogy teljesül-e az $\left(\alpha_A^{u_1} y_A^{u_2} \bmod p_A\right) \bmod q_A = r'$ egyenlőség. Az előbbi bekezdés alapján akkor és csak akkor fogadjuk el (r', s')-t A aláírásának m'-re, ha az előbbi egyenlőség fennáll.

A második toldalékos aláírás, amellyel foglalkozunk, az **általánosított AlGamal rendszer**. Ismét A rendszerét vizsgáljuk. Legyen G az α által generált n-edrendű ciklikus csoport, $h: C^* \to \mathbf{Z}_n$, ahol C az üzenetekhez használt ábécé és $f: G \to C^*$ egy injektív függvény. A továbbiakban, a rövidség kedvéért, legyen $\varphi = hf$. A választ egy $n > a \in \mathbb{N}^+$ egészt, és kiszámítja $y = \alpha^a$-t. A nyilvános kulcsa $(\mathsf{G}, f, \alpha, y)$, míg az aláíráshoz használt titkos kulcsa a.

Amikor A aláírja az m üzenetet, akkor választ egy véletlen, n-hez relatív prím, $n > k \in \mathbb{N}^+$ egészt, és kiszámítja $r = \alpha^k$-t valamint $\kappa = k^{-1} \bmod n$-et, továbbá $s = \kappa(h(m) - a\varphi(r) \bmod n)$-et. Az A m-hez tartozó aláírása az (r, s) pár, az elküldött, aláírt üzenet $\left(m, (r, s)\right)$ lesz.

Az előbbi kifejezésből kapjuk, hogy $ks + a\varphi(r) \bmod n = h(m)$, és ebből $r^s y^{\varphi(r)} = \alpha^{h(m)}$.

Ellenőrzésnél az $\left(m', (r', s')\right)$ pár első eleméből meghatározzuk $h(m')$-t, (r', s')-ből $\varphi(r')$-t, majd ezekkel $v_1 = {r'}^{s'} y^{\varphi(r')}$-t és $v_2 = \alpha^{h(m')}$-t. Akkor és csak akkor fogadjuk el (r', s')-t mint A aláírását m'-re, ha $v_1 = v_2$.

Két lényeges kérdést említünk a rendszer használatával kapcsolatban.

Először tegyük fel, hogy A két különböző üzenetet ír alá úgy, hogy ugyanazt a k egészt használja. Ekkor a k-ból számított r-ek is megegyeznek. Legyen a két üzenet m_1 és m_2, és legyen a két aláírás (r, s_1) és (r, s_2). Most $k(s_1 - s_2) \equiv h(m_1) - h(m_1)\ (n)$, és ebben a kongruenciában egyedül k

nem ismert, amely így meghatározható (a kongruenciának biztosan van megoldása, például az eredeti k). Ha viszont ismerjük k-t, akkor például $s_1 \equiv \kappa\big(h(m_1) - a\varphi(r)\big)\ (n)$-ből megkapjuk a-t, A titkos kulcsát.

A másik probléma, ha nem használunk hash-függvényt, vagyis ha $h(m) = m$. Legyen ekkor u és v két pozitív egész, az utóbbi relatív prím n-hez. Legyen most $r = \alpha^u y^v$, $s = -\varphi(r)v^{-1} \bmod n$ és $m = su \bmod n$. Ezekkel az értékekkel $v_1 = (\alpha^u y^v)^{-\varphi(r)v^{-1}} y^{\varphi(r)} = \alpha^{-u\varphi(r)v^{-1}} = \alpha^{us} = \alpha^m = v_2$, és az így meghatározott (r, s) pár A érvényes aláírása m-re, jóllehet A feltehetően nem is látta m-et.

Az eredeti AlGamal rendszert úgy kapjuk ebből az általánosított rendszerből, hogy a ciklikus csoport $\mathbb{Z}_p^*$ valamilyen p prímszámmal, h most $\mathbb{Z}_p$-be képez, és φ az identikus leképezés, továbbá $y = \alpha^a \bmod p$ és aláírásnál $r = \alpha^k \bmod p$. Az ellenőrzés annyiban módosul, hogy a $v_1 \equiv v_1\ (p)$ feltételt kell ellenőrizni.

A rendszernél ellenőrzéskor lényeges megnézni, hogy teljesül-e a $p - 1 > r \in \mathbb{N}^+$ feltétel, mert ellenkező esetben lehetőség van csalásra. Tegyük ugyanis fel, hogy (r, s) A egy legális aláírása m-re. Választunk egy tetszőleges m' üzenetet, majd kiszámítjuk $h(m')$-t, és ha $(h(m), p - 1) = 1$, akkor $u = h(m')\big(h(m)\big)^{-1} \bmod (p-1)$-et. Ha $s' = su \bmod (p-1)$, és r' olyan, hogy $r' \equiv ru\ (p-1)$ és $r' \equiv r\ (p)$, akkor $y^{r'} r'^{s'} \equiv \left(\alpha^{(ar+ks)(h(m))^{-1} \bmod (p-1)}\right)^{h(m')} \equiv \alpha^{h(m')}\ (p)$, vagyis (r', s') A érvényes aláírása m'-re, amennyiben nem ellenőrizzük, hogy $p - 1 > r' \in \mathbb{N}^+$ (r'-t hatvány alapjaként r-rel, míg kitevőként ru-val helyettesítjük). Egyébként hasonló támadás a DSA-nál is lehetséges, így ott is fontos a megfelelő ellenőrzés.

Az ismertetett aláírási rendszereket használják elliptikus görbékkel is.

Vissza a tartalomhoz

12. Titokmegosztás

Egy önállóan megoldandó kérdéssel kezdjük: ha adott egy kincseskamra, amelyhez n embernek van kulcsa, de akkor és csak akkor lehet kinyitni az ajtaját, ha legalább k ember jelen van, akkor hány zárat kell elhelyezni a bejáraton, és egy-egy embernél hány zár kulcsa kell, hogy legyen?

A mi szempontunkból a probléma úgy merül fel, hogy ha adott egy T titkos információ, hogyan oldható meg, hogy adott n emberből bármely legalább k meg tudja határozni T-t, de tetszőlegesen kiválasztott k-nál kevesebb erre ne legyen képes. Az ilyen rendszereket szokás (n, k)-**küszöbrendszer**nek is nevezni.

A felvetett kérdésnek számos megoldása létezik, itt most közülük kettőt ismertetünk.

1. Legyen p, valamint $n \geq i \in \mathbb{N}^+$-ra m_i pozitív egész szám úgy, hogy az m_i-k páronként relatív prímek, továbbá a k legkisebb m_i szorzata, $M_{min}^{(k)}$, legyen nagyobb, mint a $k-1$ legnagyobb m_i szorzatának, $M_{max}^{(k-1)}$-nek a p-szerese, vagyis legyen $pM_{max}^{(k-1)} < M_{min}^{(k)}$, végül legyen $T < M_{min}^{(k)}$. Ha a résztvevőket az $n \geq i \in \mathbb{N}^+$ egészekkel azonosítjuk, akkor az i-edik résztvevő kulcsa az (m_i, T_i) pár, ahol $T_i = T \bmod m_i$. $a \bmod b$ definíciója alapján látjuk, hogy $T_i \equiv T\ (m_i)$, így T egy szimultán lineáris kongruencia-rendszer megoldása. Jelölje J az n-nél nem nagyobb pozitív egészek halmazának tetszőleges részhalmazát, azaz legyen $J \subseteq \{i \in \mathbb{N}^+ | i \leq n\}$. Ha a J által meghatározott személyek akarják meghatározni T-t, akkor megoldják az $i \in J: x \equiv T_i\ (m_i)$ kongruenciarendszert. Ennek egy és csak egy megoldása van az $M^{(J)} = \prod_{i \in J} m_i$-nél kisebb nem negatív egészek halmazán, és ha ez $T_0^{(J)}$, akkor a $T_r^{(J)} = T_0^{(J)} + rM^{(J)}$ egészek és csak ezek a megoldásai az előbbi rendszernek, ahol r végigfut az egész számok halmazán, az pedig nyilván igaz, hogy T is megoldás, így valamely r egészre $T = T_r^{(J)}$. Két eset lehetséges.

a) $|J| \geq k$. Ekkor $M^{(J)} \geq M_{min}^{(k)}$, és mind T, mind $T_0^{(J)}$ az $M^{(J)}$-nél kisebb nem negatív megoldás, így szükségszerűen megegyeznek, vagyis $T = T_0^{(J)}$, tehát a kongruencia megoldásával rendelkezésre áll a keresett titkos információ.

b) $|J| < k$. Ennél az esetnél $T_0^{(J)} < M^{(J)} \leq M_{max}^{(k-1)} < M_{min}^{(k)}$, és így $T_r^{(J)}$ minden olyan r nem negatív egésszel, amellyel $T_0^{(J)} + rM^{(J)} < M_{min}^{(k)}$, lehetséges megoldás. A lehetséges T értékek száma legalább p, hiszen ha $0 \leq r < p$, akkor $T_0^{(J)} + rM^{(J)} < M_{max}^{(k)} + (p-1)M_{max}^{(k-1)} = pM_{max}^{(k-1)} < M_{min}^{(k)}$. Ha p nagy, akkor T potenciális értékeinek száma nagy, és ezek közül semmilyen módszerrel nem tudjuk meghatározni a tényleges értéket, sőt, bármelyikük azonos eséllyel lehet a valódi megoldás.

Az a feltétel, hogy az m_i-k páronként relatív prímek, nem lényeges kikötés, amint alább majd látjuk. Az általános esetben $M^{(J)} = [m_i | i \in J]$, ahol a szögletes zárójel a legkisebb közös többszöröst jelöli, ekkor $M_{min}^{(k)}$ az összes lehetséges módon kiválasztott k különböző indexhez tartozó m_i legkisebb közös többszörösének minimuma, míg $M_{max}^{(k-1)}$ a $k-1$-elemű indexhalmazhoz tartozó m_i-k legkisebb közös többszöröseinek maximuma.

A rendszer a nem relatív prím modulusokkal bizonyos hibavédelmet is jelent (akár véletlen, akár szándékos hiba esetén), ugyanis a kongruencia-rendszernek pontosan akkor van megoldása, ha a J indexhalmazból választott bármely két különböző u és v indexre a megadott m'_u és m'_v legnagyobb közös osztója osztja a $T'_u - T'_v$ különbséget. Annak a valószínűsége, hogy ez valamennyi párra teljesül, amennyiben bizonyos indexekre a vesszős értékek egy része nem azonos az eredetivel, elég kicsi lehet (ez így persze nem egy pontos matematikai állítás, de nyilván pontossá tehető), hiszen a titokból mindenki csak a saját árnyékát ismeri, így nem tudhatja, hogyan kellene ezt az adatot úgy megváltoztatni, hogy minden szükséges oszthatóság teljesüljön.

2. A második módszer a véges testeket alkalmazza. Legyen q az n pozitív egésznél nagyobb prímhatvány, és f egy tetszőleges, legfeljebb $k-1$-edfokú polinom $\mathbb{F}_q$ fölött, ahol $n \geq k \in \mathbb{N}^+$. Ha az n-nél nem nagyobb i pozitív egészekre az u_i-k a test páronként különböző, nem nulla elemei, és $v_i = \hat{f}(u_i)$, akkor legalább k különböző (u_i, v_i) pár egyértelműen meghatározza a polinomot, és akkor $\hat{f}(0)$-t is. Legyen tehát $T = \hat{f}(0)$, és az i-vel azonosított résztvevő kulcsa (u_i, v_i). Azt már láttuk, hogy ha legalább k pár ismert, akkor T egyértelműen meghatározható. Most legyen $t < k$, $k - t = r$ $(r > 0)$, a rendelkezésre álló t pár indexeinek halmaza I_1, $I_2 = \{n + i | r > i \in \mathbb{N}^+\}$ és $I = \{0\} \cup I_1 \cup I_2$. Láthatóan I-nek pontosan k eleme van. Legyen $u_0 = 0$, és válasszunk minden $i \in I_2$-re a testből egy tetszőleges, de rögzített u_i elemet úgy, hogy az $\{u_i | i \in I\}$ halmaz elemei legyenek páronként különbözőek (ez a $\{0\} \cup I_1$-beli indexek esetén eleve így van). Az I_1-beli i indexekre legyen $w_i = v_i$, és a többi indexre az összes lehetséges módon válasszuk meg a w_i értékeket. Egy-egy ilyen választással k különböző pontban definiáltuk egy polinom értékeit, és egy ilyen rendszerhez van egy és csak egy, legfeljebb $k-1$-edfokú polinom, amely az adott helyeken a választott értékeket veszi fel. Különböző választáshoz különböző polinom tartozik, mert a polinom egyértelműen meghatározza a polinomfüggvényt, és minden legfeljebb $k-1$-edfokú polinomot, tehát f-et is megkapunk valamely választással, hiszen az adott pontokban minden lehetséges érték-k-ast hozzárendeltük a kiválasztott pontokhoz. Bármely rögzített $w_0 = c \in \mathbb{F}_q$ esetén a lehetséges választások száma q^{r-1} (mert w_0 választása után az I_2-beli, és csak az I_2-beli indexekhez, azaz $r-1$ indexre választhatjuk meg w_i-t), és ez az érték független c-től. Az előbbiek szerint tehát pontosan q^{r-1} olyan, legfeljebb $k-1$-edfokú $g \in \mathbb{F}_q[x]$ polinom van, amelynél minden $i \in I_1$ indexre $v_i = w_i = \hat{g}(u_i)$ és $c = w_0 = \hat{g}(u_0) = \hat{g}(0)$. Ez azt jelenti, hogy a rendelkezésre álló t pár által meghatározható q^r, páronként különböző, legfeljebb $k-1$-edfokú polinom konstans tagjai a test minden elemét azonos, q^{r-1} gyakorisággal veszik fel, vagyis T ugyanakkora valószínűséggel lehet a test bármely eleme, így a polinomok meghatározásával T-ről semmilyen információnk nem lesz.

Legyen az u_i-k rendjeinek legkisebb közös többszöröse t, és u primitív t-edik egységgyök $\mathbb{F}_q$ fölött. u eleme $\mathbb{F}_q$-nak, hiszen az u_i-k mindegyike benne van a testben, és így mindegyikük rendje osztója $q-1$-nek, de akkor t is osztója ennek az egésznek. Az is igaz, hogy $t \geq n$, hiszen az u_i-k páronként különbözőek, és mindegyikük az u egy nem negatív, t-nél kisebb egész kitevős hatványa. Feltehetjük, hogy n és t megegyezik, ugyanis ellenkező esetben fiktív résztvevőkkel bővíthetjük a rendszert. Ekkor u primitív n-edik egységgyök, megfelelő indexeléssel $u_i = u^i$, és

$$v_i = \hat{f}(u_i) = \hat{f}(u^i) = \sum_{j=0}^{k-1} f_j (u^i)^j = (ne)^{-1} \sum_{j=0}^{n-1} (nf_j)(u^i)^j,$$

ahol $k \leq j < n$-re $f_j = 0$, ami nem más, mint az $\mathbb{F}_q$ feletti n-dimenziós tér egy olyan elemének, nf-nek u-val vett inverz Fourier-transzformáltja, amelyben $n-k$ egymás utáni komponens értéke (nevezetesen a $k \leq j < n$ idexhez tartozó komponenseké) 0. Ez viszont azt jelenti, hogy az n darab v_i egy $\mathbb{F}_q$ fölötti, $[n,k]_q$-paraméterű Reed-Solomon kód egy kódszavának tekinthető, és az i-edik résztvevő kulcsa az (i, v_i) pár (ez persze lényegében véve azonos a korábbi kulccsal, hiszen i egyértelműen meghatározza u^i-t, tehát u_i-t). Tegyük fel, hogy $s = k + 2r$ kulcs áll rendelkezésünkre, amelyek közül r különbözik a valódi kulcstól (akár véletlenül, akár szándékosan), vagyis ismert egy olyan vektor s komponense, amely ezen pozíciók közül r helyen különbözik az eredeti kódszótól. Ez egyben azt is jelenti, hogy adott $n-s$ olyan pozíció, amelynek nem ismerjük az értékét. A helyzet úgy is felfogható, hogy van egy vektorunk, amely egy kódszóból $n-s$ törlődéses hibával és r további hibával keletkezett. Egy d távolságú kód helyesen javít, ha a törlődéses hibák számának és a további hibák r száma kétszeresének összege kisebb, mint a kód távolsága, ami most teljesül, hiszen $n - s + 2r = n - k$, és egy $[n,k]$-paraméterű Reed-Solomon kód távolsága $n-k+1$. A javítás után kapott vektor az eredeti kódszó, amelyből például Fourier-transzformációval meghatározható f, és akkor $T = \hat{f}(0)$ is. $\hat{f}(0)$-t azonban lényegesen könnyebben, sőt nagyon könnyen kiszámíthatjuk $\boldsymbol{v}$-ből:

$$\sum_{i=0}^{n-1} v_i = \sum_{i=0}^{n-1} \hat{f}\left(u^i\right) = \sum_{i=0}^{n-1}\sum_{j=0}^{k-1} f_j \left(u^i\right)^j = \sum_{j=0}^{k-1} f_j \sum_{i=0}^{n-1} \left(u^j\right)^i$$
$$= nf_0 + \sum_{j=1}^{k-1} f_j \frac{(u^n)^j - e}{u^j - e} = nf_0 = n\hat{f}(0),$$

ahonnan $\hat{f}(0) = (ne)^{-1} \sum_{i=0}^{n-1} v_i$.

Nézzünk egy példát az első módszerre[2]. Legyen $n = 5$, $k = 3$, $T = 100$, és az öt modulus legyen sorban $m_1 = 7$, $m_2 = 11$, $m_3 = 13$, $m_4 = 17$ és $m_5 = 19$. Ekkor $M_{min}^{(3)} = 7 \cdot 11 \cdot 13 = 1001$ és $M_{max}^{(2)} = 17 \cdot 19 = 323$, és $\left\lfloor \frac{M_{min}^{(3)}}{M_{max}^{(2)}} \right\rfloor = \left\lfloor \frac{1001}{323} \right\rfloor = 3$, így legyen $p = 3$. Ezekkel a paraméterekkel a titok árnyékai az indexek szerinti sorrendben megadva $t_1 = 100 \bmod 7 = 2$, $t_2 = 100 \bmod 11 = 1$, $t_3 = 100 \bmod 13 = 9$, $t_4 = 100 \bmod 17 = 15$ és $t_5 = 100 \bmod 19 = 5$. Ha az 1-es, 2-es és 5-ös indexű titokgazda adatai állnak rendelkezésre, akkor legyen $M = m_1 m_2 m_5 = 1463$, $M_1 = \frac{M}{m_1} = 209$, $M_2 = \frac{M}{m_2} = 133$ és $M_5 = \frac{M}{m_5} = 77$. $209x \equiv 1\ (7)$ megoldása $u_1 = -1$, $133x \equiv 1\ (11)$ megoldása $u_2 = 1$ és a harmadik kongruencia, $77x \equiv 1\ (19)$ megoldása ismét $u_5 = 1$. Ebből

$$T' = (-1 \cdot 209 \cdot 2 + 1 \cdot 133 \cdot 1 + 1 \cdot 77 \cdot 5) \bmod 1463 = 100 = T,$$

és a kongruencia-rendszernek nincs más olyan nem negatív megoldása, amely kisebb, mint 1001, vagyis megkaptuk a teljes titkot. Ha viszont csupán két indexhez tartozó adat áll rendelkezésünkre, mondjuk a 2-es és a 4-es, akkor $M = m_2 m_4 = 187$, $M_2 = \frac{M}{m_2} = 17$, $M_4 = \frac{M}{m_4} = 11$, $17x \equiv 1\ (11)$-ből $u_2 = 2$ és $11x \equiv 1\ (17)$-ből $u_4 = -3$, így most $T' = (2 \cdot 17 \cdot 1 - 3 \cdot 11 \cdot 15) \bmod 187 = 100$, illetve az ezzel modulo 187 kongruens bármely egész szám. Ezek között 5 olyan nem negatív megoldás van, amely kisebb, mint 1001, a 100, a 287, a 474, a 661 és a 848, és semmilyen lehetőségünk nincs annak eldöntésére, hogy melyik a valódi megoldás (az, hogy a valódi titok, a 100, az első a lehetséges megoldások között, csupán véletlen, az viszont nem, hogy a lehetséges megoldások között szerepel).

[2] A példát **Madarász János** javasolta.

13. Elektronikus pénz, kriptográfiailag hitelesített pénz

Már korábban szóba került egy alapvető konfliktus, amely az egyén és a társadalom érdekeinek ütközéséből fakadt. Hasonló problémával találkozhatunk a pénzforgalom területén is. A különböző készpénzkímélő fizetési módszerek többnyire igen hasznosak és kényelmesek, és ugyanakkor nagyon előnyösek az állam biztonsága szempontjából, hiszen könnyű egyénhez kapcsolódóan nyomon követni egy-egy tranzakció útját. A dolog másik oldala viszont, hogy vannak olyan esetek, amikor valaki anonim módon szeretne például valamit vásárolni. Ennek a lehetőségét a készpénzzel való fizetés teremti meg. Ez mutatja, hogy napjaink egyre elektronizáltabb világában is szükségünk lehet a régi, jól bevált, kézzel fogható bankókra és érmékre. Illetve nem is feltétlenül a materiális valóságában létező készpénzre van szükség, hanem csupán egy tetszőleges olyan eszközre, amely úgy funkcionál, mint a készpénz, vagyis amely képes részt venni a cserefolyamatban, de amely utólag nem köthető egy konkrét személyhez, egy konkrét tranzakcióhoz. Ilyen eszközt meg lehet valósítani kriptográfiai eszközök segítségével is.

A *CEPS* Csoport (*Europay International*, *Visa España/SERMEPA*, *Visa International* és *ZKA*) 1998. december végén közzétette azokat a specifikációkat, amelyek segítségével a világ különböző elektronikus pénztárca-programjai képesek az együttműködésre. A közös elektronikus pénztárca-specifikációk (*CEPS – **C**ommon **E**lectronic **P**urse **S**pecifications*) létrehozása jelentős lépés a nyílt elektronikus pénztárcaszabvány megteremtése felé, és világszerte elősegíti az intelligens kártyák számának növekedését.

A *CEPS* Csoport a specifikációkat először átadja ellenőrzésre a biztonsági laboratóriumoknak, majd a kiértékelés befejeztével közzéteszi a végleges specifikációkat.

Az elektronikus pénztárcaprogramok többsége Európában zajlik, így különböző európai intézmények – többek közt az *ECBS* (***E**uropean **C**ommittee for **B**anking **S**tandards*) – is nagyban hozzájárultak a specifikációk megalkotásához. Következésképpen várhatóan elsőként az európai programok fogják alkalmazni a specifikációkat, mivel itt az Euro bevezetése tovább fokozza a közös szabványok iránti igényt. A *CEPS* bevezetésével lehetővé válik, hogy a kártyabirtokosok belföldön és külföldön egyaránt használhassák elektronikus pénztárcájukat.

Huszonkét ország szervezetei – amelyek a világ elektronikus pénztárcáinak több mint kilencven százalékát képviselik – döntöttek már eddig a *CEPS* alkalmazása mellett. Ezek között a szervezetek között szerepel a *Visa International*, a *Visa España/SERMEPA*, a *ZKA*, a *Europay International*, a *Proton World International*, az olaszországi *SSB*, a szingapúri *NETS*, a „*Cash*" pénztárcaprogramot támogató svéd bankok és a *Europay Austria*. A *Groupement Cartes Bancaires* szintén elkötelezte magát a kezdeményezés mellett, és szívesen válna a *CEPS* Csoport aktív tagjává.

A *CEPS* felsorolja a feltételeit, hogy valamely szervezet bevezessen egy világszerte kompatibilis elektronikus pénztárcaprogramot. Megköveteli a chipkártyákra kidolgozott *EMV*-szabványokkal (***E**uropay*, ***M**astercard*, ***V**isa*) való kompatibilitást, továbbá definiálja a következő fogalmakat: kártyaalkalmazás, kártyaterminál-interfész, *POS* (*P*oint *o*f *S*ale)- és pénztárca-feltöltési tranzakciók terminálalkalmazásai, adatelemek, valamint a tranzakció-feldolgozási folyamat ajánlott üzenetformátumai. A *CEPS* megfogalmazza a különböző elektronikus pénztárcarendszerek résztvevői számára felállított funkcionális követelményeket, valamint a fokozott biztonságosság érdekében külön titkosítási kulcsot alkalmaz.

A közös elektronikus pénztárca-szabvány kialakítására tett erőfeszítések újabb lendületet kaptak 1998. júniusában, amikor a világ legnagyobb pénztárcarendszer-működtetői bejelentették, hogy munkacsoportot alakítottak a nyílt ipari specifikációk kidolgozására. A *CEPS* ennek a törekvésnek a megvalósulása.

Az **elektronikus** (**kész**)**pénz** (**digitális pénz** – *electronic/e-/digital money/cash/currency*), egy lehetséges megvalósítása a következő. A bank minden létező címlethez előállít egy-egy *RSA*-paraméter-rendszert, amelyek közül a nyilvános paramétereket, mondjuk a t címletre vonatkozóan (t, n_t, e_t)-t, nyilvánosságra hozza, a titkos paramétereket viszont (t esetén d_t-t) titokban tartja. Tegyük fel, hogy Pénzes úr, akinek ennél a banknál van számlája, szeretne egy t-egységnyi címletet készpénzben kivenni a bankból. Előállít két nagy, véletlen számot, r-et és M-et, amelyek közül r-nek létezik az inverze a nyilvános n_t paraméterre vonatkoztatva, vagyis amelyhez van olyan r', hogy

$rr' \bmod n_t = 1$. M természetesen valamilyen számrendszerben van megadva, mondjuk binárisan. Pénzes úr leírja M-et, és közvetlenül mellé másolja ismét M-et. A kapott M' eredményt $M \parallel M$-mel jelöljük (mondjuk, legyen $M = 5$, akkor ez bináris felírásban 101, és ebből $M' = M \parallel M = 101101 = 45$), majd kiszámítja $s = r^{e_t} M' \bmod n_t$-t, és elküldi a bankjába azzal, hogy a számlájáról emeljenek le t forintot, és adjanak ki készpénzben ugyanekkora összeget. A bank megterheli Pénzes úr számláját a t összeggel, és visszaküldi a megfelelő összeg igazolását, $s' = s^{d_t} \bmod n_t$-t. Ebből Pénzes úr kiszámítja $T = r's' \bmod n_t$-t, amiből a következőt kapja:

$$\begin{aligned} T &= r's' \bmod n_t = r'\left(s^{d_t} \bmod n_t\right) \bmod n_t = r's^{d_t} \bmod n_t \\ &= r'(r^{e_t} M' \bmod n_t)^{d_t} \bmod n_t = r'(r^{e_t} M')^{d_t} \bmod n_t \\ &= r'(r^{e_t})^{d_t} M'^{d_t} \bmod n_t = r'rM'^{d_t} \bmod n_t = M'^{d_t} \bmod n_t. \end{aligned}$$

A legutolsó eredmény, $T = M'^{d_t} \bmod n_t$ lesz a t összegnek megfelelő készpénz. Ha Pénzes úr fizetni akar egy T'-s, állítólag t-nek megfelelő pénzeszközzel egy boltban, akkor a boltos a nyilvános (t, n_t, e_t) paraméterek segítségével kiszámítja $M'' = T'^{e_t} \bmod n_t$-t. Ha az átadott T' valóban egy t-értékű fizetőeszköz, akkor $M'' = M''' \parallel M'''$-szerkezetű. Az említett számok igen nagyok, százas nagyságrendű decimális számjegyből állnak. Pénzes úr úgy tudna csalni, ha az előbb leírt folyamat kikerülésével, közvetlenül, a bank nélkül elő tudna állítani egy olyan $\tilde{T}$ számot, hogy $\tilde{T}^{e_t} \bmod n_t$ számjegyes felírásában a számjegyek bal oldali felének sorozata jegyről jegyre megegyezik a jobb oldali fél jegysorozatával. Ennek a valószínűsége d_t ismerete nélkül, a szám nagyságát figyelembe véve, amely kizárja a próbálkozáson alapuló keresést, gyakorlatilag 0, vagyis szinte kizárt egy ilyen szám „vak tyúk" alapon történő előállítása (ha valakinek ez véletlenül sikerül, az „káló"-nak tekinthető, hiszen itt egy-egy bankjegyről, vagy pénzérméről van szó). Ha tehát a boltban átadott „pénz" a sifrírozás után kielégíti a formai követelményeket, akkor a boltban azt elfogadják t-egységnyi fizetésnek. Még egy „apróság"-ot kell figyelembe venni. Pénzes úr, vagy bárki, akinél ez a pénzdarab megfordul, ismételten és többször is fel tudná használni, ezért a bank nyilvántartást vezet a hozzá benyújtott, a számlán jóváírandó fizetőeszközökről (vagyis a mi esetükben M'''-ről), és egy adott M'''-t csupán egyszer fogad el. Ez viszont azt jelenti, hogy az óvatos boltban rögtön a fizetéskor benyújtják a bankhoz a kapott „bankjegy"-et illetve „érmé"-t, és csak akkor fogadják el fizetés gyanánt, ha a bank azt érvényesnek találta, és jóváírta a bolt számláján. Ez persze azért nem jó, mert akkor a bolt ezt a pénzeszközt nem tudja felhasználni a pénzforgalmában. Ha viszont nem így jár el, akkor azt kockáztatja, hogy átverik, becsapják.

Ez a módszer valóban lehetővé teszi az anonimitást. A bank, amikor Pénzes úr tőle pénzt kér, nem tudja, hogy mi volt M, ő ugyanis csak $s = r^{e_t} M' \bmod n_t$-t látja, és ebből nem tudja kiszámítani M'-t, és így nem képes meghatározni M értékét, hiszen nem ismeri r-et. Ebből következik, hogy amikor a banknak benyújtják M'-t, a bank azt nem képes Pénzes úrhoz kapcsolni.

A fentieket egy egyszerű példán, kis számokkal mutatjuk be. Legyen a két prím 113 és 127, ekkor $n = 14\,351$, és $\varphi(n) = (113 - 1)(127 - 1) = 14\,112$. Ha nyilvános kitevőnek $e = 131$-et választjuk, akkor némi számolással $131 \cdot 9\,803 - 1 = 1\,284\,192 = 91 \cdot 14\,112$, így $d = 9\,803$. A nyilvános adatok $(100, n = 14\,351, e = 131)$. Most legyen például $r = 37$ és $M = 54$, ekkor $r' = 5\,818$ és $M' = 5\,454$. r^e 14 351-gyel való osztásakor keletkező maradék 814, majd ezt 5 454-gyel szorozva, és a szorzatot elosztva 14 351-gyel, a kapott maradék 5 097, vagyis a korábbi jelölésekkel $s = 5\,097$. s-t beküldi Pénzes úr a bankba, ahol kiszámítják s d-edik hatványának maradékát a 14 351-gyel való osztáskor, ami 10 137. Ezt a számot Pénzes úr megszorozza r'-vel, és veszi az n-nel való osztási maradékot, T-t, ami most 8 807, ez a szám tehát egy 100 forintost képvisel. Ha most valaki ezzel a pénzzel fizet, akkor a kereskedő veszi T e-edik hatványát, és elosztja a hatványt 14 351-gyel. Ez a maradék $T = 8\,807$ esetén 5 454, vagyis egy olyan szám, amelynek a bal oldali és jobb oldali fele megegyezik, és mivel a 100 forintoshoz tartozó adatokkal számoltuk, ezért elfogadjuk egy 100 forintos pénzeszköznek. Benyújtva T-t a banknak, az ellenőrzi, hogy nem fizetett-e már egyszer valaki ezzel a pénzzel, és ha nem, akkor jóváír a kereskedőnek 100 forintot.

A fenti számítás kézzel, papíron és ceruzával nem túl kényelmes, hiszen a számolás közben keletkező részeredmények igen sok jegyből álló számok, ám egy számítógép számára ez gyerekjáték. A valóságban persze az alkalmazott számok körülbelül 100-jegyűek, és az ilyen nagy számokkal már egy gyors számítógépnek is el kell bíbelődnie egy-két percig, de azért elboldogul a feladattal.

Röviden nézzük meg, hogy milyen módon képes a kriptográfia hozzájárulni a pénzhamisítás leküzdéséhez. A másolás ellen természetesen nem alkalmazható védőeszközként, de hamis pénz előállítása ellen igen. A módszer szerint minden egyes bankóra felvisznek egy véletlenszerű pontokból álló mintázatot. Ezt a mintázatot végigpásztázzák, és átalakítják egy bitsorozattá (PAT). Ezt a számot a kibocsátó bank aláírja a nyilvános kulcsú aláírásával, és az így kapott aláírást (SIG) egy hibakorlátozó kódolás után egy számjegysorozattá alakítják (SIG^*), amelyet rányomnak a bankjegyre. A hibakorlátozásra azért van szükség, mert ha hibásan olvassák le fizetéskor a számot, akkor az ellenőrzés biztosan sikertelen lesz. Felhasználáskor SIG^*-ot visszaalakítják SIG-gé, majd a nyilvános kulcs segítségével PAT-té, és ezt összehasonlítják a bankjegyről leolvasott és digitalizált bitsorozattal, PAT'-vel, ami általában nem lesz pontosan azonos PAT-tel a használat során „elszenvedett” apró változások miatt. A gyakorlatban akkor fogadható el eredetinek a bankó, ha az eltérés egy bizonyos értéket, mondjuk a 80%-ot nem haladja meg.

Vissza a tartalomhoz

14. A kincseskamra problémájának megoldása

Kérdés

Adott $n \in \mathbb{N}^+$ ember és egy trezor. Hány lakatra és egy embernél hány kulcsra van szükség, ha azt akarjuk, hogy legfeljebb $k-1$ ember ne tudja kinyitni a trezort, de bármely legalább k ember már igen, ahol $n \geq k \in \mathbb{N}^+$.

Megoldás

Ha egyetlen $k-1$ emberből álló csoport sem képes kinyitni a trezort, akkor nyilván nem képes $k-1$-nél kevesebb sem. Hasonlóan, ha bármely k ember már képes kinyitni, akkor k-nál több is képes erre, így elegendő azt vizsgálni, hogy milyen feltételek mellett nem elegendő $k-1$ ember a nyitáshoz úgy, hogy a kimaradók közül bárki eggyel kiegészülve már ki tudják nyitni a trezort.

n emberből $\binom{n}{k-1}$ olyan csoport képezhető, amelyben $k-1$ ember van. Minden ilyen csoporthoz kell lennie legalább egy olyan lakatnak, amelyhez a kiválasztott $k-1$ ember egyikénél sincs kulcs, tehát egy lakat biztosan kell. Egy $k-1$ emberből álló csoporthoz nyilván elegendő egyetlen ilyen lakat, ezért legfeljebb $\binom{n}{k-1}$ lakatra van szükség. Legyen C_1 és C_2 két *különböző* $k-1$-es csoport. Ekkor $\binom{n}{k-1} \geq 2$, tehát $k-1 > 0$, azaz $k \geq 2$. Ha a két csoport ugyanazt a lakatot nem tudja nyitni, akkor a két csoport tagjai együttesen sem képesek felnyitni a trezort, lenne $2(k-1) = 2k-2 = k+(k-2) \geq k$ ember, akik egy csoportot alkotva nem tudnák kinyitni a trezort, ami ellentmond a feladat feltételének. Ez tehát azt jelenti, hogy minden pontosan $k-1$ emberből álló csoporthoz kell egy és csak egy lakat, amelyet együttesen sem tudnak kinyitni, de különböző csoporthoz különböző lakat kell, és így a szükséges lakatok száma megegyezik a kiválasztható $k-1$-es csoportok számával, vagyis $\binom{n}{k-1}$-gyel.

Most nézzük, hogy egy-egy embernél hány lakatkulcsnak kell lennie. Ha u az n ember egyike, és C egy őt nem tartalmazó, $k-1$ emberből álló csoport, akkor van egy és csak egy lakat, amelyet C egyetlen tagja sem képes kinyitni, de amelyhez u-nak van kulcsa. Azt is láttuk, hogy különböző csoport esetén különböző lakat van, így u-nál minden olyan, $k-1$ emberből álló csoporthoz kell lenni egy és csak egy kulcsnak, amely egy olyan lakatot nyit, amelyet a csoport tagjai nem tudnak kinyitni. Ebből következik, hogy az u-nál lévő kulcsok száma megegyezik az $n-1$ emberből kiválasztható $k-1$-es csoportok számával, ami $\binom{n-1}{k-1}$.

Vissza a tartalomhoz

15. Etimológia

kripto- *gör* előtagként vminek a titkos v. rejtett voltát jelöli; titkos-, rejtett
κρύπτω (krüpto) a ***κρυπτός*** (krüptosz) *rejtett*, *titkos* görög szóból
κρύπτω (főnévi igenév: ***κρύπτειν*** krüptein) *elrejt*

-lógia, -logia *gör-lat* **1.** utótagként jelöl: vmilyen tudományt; -tan, -tudomány (pl. *geológia*) **2.** utótagként jelöl: (számnevekkel) az összetevők számát (pl. *tetralógia*) **3.** utótagként jelöl: vmilyen beszéd- v. előadásmódot (pl. *tautológia*)
λόγος (logosz) *szó*, *beszéd*, *magyarázat*, *fogalom*, *tudomány* szóból eredeti görög képzésű utótag
λόγιος, ***λογία***, ***λόγιον*** (logiosz, logia, logion) *értelmes*; *tudománnyal kapcsolatos*
λέγω (lego) (főnévi igenév: ***λέγειν*** legein) *mond*

kriptológia *gör el.* a rejtjelfejtés elmélete és gyakorlata
rejtett dolgok tudománya

-gráfia (-graphia) *gör* **1.** utótagként jelöl: vmely tudományágat (pl. *geográfia*) **2.** utótagként jelöl: vmely írás- v. más rögzítési módot (pl. *fotográfia*) **3.** utótagként jelöl: vmely nyomdászati eljárást; -nyomás (pl. *litográfia*)
γραφή (grafé) *írás* szóból görög képzésű utótag
γράφω (grafo) (főnévi igenév: ***γράφειν*** grafein) *vés*; *ír*

kriptográfia *gör el.* titkosírás, rejtjeles írás, ill. ennek rendszere, kulcsa
fn *Tud*|Titkosírások készítésének és megfejtésének módszertana.|Titkosírás. [nk: gör el.]
titkosírás (mestersége)

analízis *gör* **1.** elemzés; részekre, elemekre való bontás mint tudományos kutató módszer **2.** *mat* a matematika azon ágainak összessége, amelyek a függvény, a határérték, a differenciál és az integrál fogalmával szervesen összefüggnek, arra épülnek **3.** vegyelemzés **4.** lélekelemzés, pszichoanalízis
ανάλυσις (analüszisz) *feloldás, megoldás, darabokra szedés, megfejtés*
αναλύω (analüo) (főnévi igenév: ***αναλύειν*** analüein) *feloszt, feldarabol*
ανα- (ana-) *föl* + ***λύω*** (lüo) (főnévi igenév: ***λύειν*** lüein) *old*

kriptoanalízis *gör* **cryptanalysis** *fn* titkosírás megfejtése
titkos írás, titkos jelek megfejtése

-gram, -gramma *gör* **1.** utótagként jelent: vmilyen *-gráf* utótagú műszer által lerajzolt görbét (pl. *szeizmogram*) **2.** utótagként jelent: vmilyen ábrát v. görbét (pl. *diagram*) **3.** utótagként jelent: vmely írásművet (pl. *epigramma*)
γράμμα (gramma) *vésett*, *betű*; *rajzolás*, *írás*, *feljegyzés* ***γράφ|ω***-ból a ***-μα*** képzővel

kriptogramma *gör el.* titkosírásos v. rejtett értelmű felirat, szöveg(részlet)

entrópia: *Shannon* javaslatára entrópiának nevezzük az információ átlagos hírértékét. Az entrópia eredetileg a hőtanban használt állapotjelző neve, melyet *Rudolf Clausius* vezetett be a termodinamikai folyamatok megfordíthatóságának mértékeként. Az *εντρεπειν* (*entrepein*) görög szó, jelentése: *megfordít*. Az információelméleti és termodinamikai entrópia rokonsága a matematikai modell azonosságára vezethető vissza.

entrópia *gör* **1.** *fiz* anyagi rendszerek molekuláris rendezetlenségi fokának, ill. állapotuk termodinamikai valószínűségének mértéke **2.** *fiz* az energia hasznosíthatóságának, munkavégző képességének mértéke termikus folyamatokban **3.** *inf* a bizonytalanságnak a kapott információkkal csökkenő

arányszáma **4.** *fil* a termodinamikai állapotfüggvények hatályának hibás kiterjesztése révén létrehozott elmélet a világ hőhaláláról

εντροπία (entropia) *fordulat*

εντροπή (entropé) *fordulat*, *belefordulás*, *meghajlás, lealacsonyodás* szóból valószínűleg latin szavak mintájára képzett szó

εν- (en) *-ban*, *-ben* + **τροπή** (tropé) *fordulat* a ***τρέπω*** (trepo) (főnévi igenév:***τρέπειν*** trepein) *fordít* igéből

redundancia: 1. hétköznapi értelemben felesleg, vagyis az a többlet, amelyet a cél eléréséhez mindenképpen szükséges eszközökön túl használnak. 2. Számítástechnikai vonatkozásban az információ ábrázolására rendelkezésre bocsátott, de fel nem használt terület. Ha egy karakterlánc például 256 karakteres, de aktuális értéke csak 6 bájt hosszú, 250 redundáns bájtot tartalmaz, hiszen az eredetileg a karakterlánc számára lefoglalt tárterület nagysága nem változik. 3. Az információforrás redundanciája az egyenletes eloszlású forrás maximális entrópiájához viszonyított relatív entrópia komplementere: $R_S = 1 - H(S)/\log V$. Szemléletesen a forrás egy hírében, üzenetében rejlő, de információt, tehát újdonságot nem tartalmazó közlés arányát, a hír banalitásának fokát fejezi ki. 4. A kód redundanciája az átlagos szóhossznak az információt ténylegesen nem hordozó, vagyis a feltétlenül szükséges minimális szóhosszt meghaladó része. A szeparálható kódok esetében a minimális szóhosszt pontosan ismerjük, ilyenkor ennek és a tényleges szóhossz arányának a komplementere a kód redundanciája. Mivel $L_0 = H(S)/\log B$ (*Shannon* II. tétele), ezért $R_E = 1 - L_0/L = 1 - H(S)/(L \log B)$.

redundancia *lat* **1.** *inf* újabb információt nem adó felesleg a közleményben, amely nélkül azonban a megértés nehezebbé válna **2.** vminek redundáns volta

fn *Távk* Közlésben az egyértelmű megértéshez elvileg elegendő minimumom felüli többlet. [nk: lat]

redundantia (*túlzott*) *bővelkedés*

redundáns *lat* **1.** *inf* új információt, érdemleges közlést már nem tartalmazó **2.** terjengős, fölösleges elemeket tartalmazó

redundant-, ***redundans*** (*lat*) jelenidejű melléknévi igenév a *redundo*, *redundare túlcsordul* igéből

re-, ***red***- (fokozás)- + ***unda*** *hullám*

sifre (*fr→ném*) titkosírás, rejtjel

sifríroz *fr→ném* titkosírással ír, rejtjelez
encipher egy szöveg titkosítása

desifríroz *fr→ném* kibetűz, titkos- v. rejtjelezett írást megfejt
decipher titkosított szöveg visszafejtése

chiffre *h fn* **1.** szám(jegy) **3.** titkos írásjel, rejtjel, sifrírozás; **en chiffres** sifrírozva; rejtjelben; **le Chiffre** a külügyminisztérium rejtjelosztálya **4.** rejtjel- v. sifre-kódex; titkosírás rendszere *[ábécéje]*

chiffre *n. m.* (XV^{e}, «écriture secrète»; *cifre*, 1220; lat. médiév. *cifra* «zéro», de l'arabe *sifr* «vide», *ch-* d'apr. it. *cifra*) II. 1. Caractères numériques de convention employés dans une écriture secrète (V. **Cryptographie**). *Écrire en chiffres* (*opposé à* écrire en clair). – *Par anal.* Tout signe de convention servant à correspondre secraitement, et absolt. *Le chiffre*, l'ensemble de ses signes. V. **Code**. *Changer de chiffre*. *Avoir le secret*, *la clef du chiffre*. V. **Chiffrer**, **déchiffrer**. *Service du chiffre*: bureau civil ou militaire où l'on chiffre et déchiffre les dépêches secrètes. *Être affecté au chiffre*.

cifra, **ziphra**, **zifera** (*jel*, *számjel*, *nulla*, *titkos írásjel*) késő-/középlatin szó. A klasszikusban érthetően nem létezik, mert az arab *ṣifr* XIII. századi átvétele a latin matematikai műnyelvbe. Eredetileg az arab szó a szanszkrit *sūnya* tükörfordítása. Számos európai nyelvben jelen van: ***Ziffer*** (ném.,

ciffer), ***chiffre*** (fr., sifr), ***cifra*** (ol., csifra)... A *titkos írásjel* értelme a diplomácia köreiből fejlődött, ugyanis itt gyakran alkalmaztak számjegyeket titkosított írásokban. A magyarba is eredetileg hasonló értelemben került be, majd a zérus, a kis kör forma díszítőelemként való alkalmazása elvezetett a *cifra*, *díszes*, *tarka* értelemhez is.

cipher, **cypher** 1. a 0 számjegy, zérus 2. bármely decimális számjegy 3. egy jelentéktelen személy 4. titkos írás(i rendszer); kód

zéró, zérus (*arab→ol*) **1.** nulla, semmi **2.** *biz* senki; jelentéktelen ember

صِفْر (szifr) **صُفْر** (szufr) **صَفِر** (szafir) **صُفُر** (szufur) **صَفْر** (szafr) többesszáma: **أصْفار** (’ászfár) *üres*, *haszontalan*, *értéktelen*, *mentes* (**مِن** (min) *vmitől*)

صِفْر (szifr) *zérus*, *zéro*, *nulla*, *semmi*

kommunikáció *lat* **1.** tájékoztatás, (hír) közlés **2.** *inf* információk közlése v. cseréje vmilyen erre szolgáló eszköz, ill. jelrendszer (nyelv, média, gesztusok stb.) útján **3.** *ritk* közlemény **4.** összeköttetés, kapcsolat, érintkezés

communicatio *közlés, a közlés folyamata*

communis, communæ *közös*

communico, communicare *megoszt, közöl, közössé tesz*

kommunikál *lat* közöl (vmit vmivel, vkivel); értesít (vkit)

információ *lat* **1.** felvilágosítás, tájékoztatás **2.** hírközlés **3.** értesülés, adat **4.** híranyag, a közlés tárgya **5.** *inf* elektronikus úton továbbított jel; hír

informatio *formába öntés*; *közlés átvitele*

informo, ***informare***, ***informavi***, ***informatum*** az ***in*** (*-ba*, *-be*, *-ban*, *-ben*) praepositio – igekötő – és a ***forma***, ***formae*** f(emininum) (*alak*) összetétele. A *forma* szó a ***fero ferre tuli latum*** (*hoz*, *visz*) ige gyökének minőségi hangmásulás (qualitatív ablaut – gyakori jelenség az indoeurópai nyelvekben) szenvedett alakja és egy főnévképző (*ma* suffixum) egyesüléséből van. *Informo = alakot ad*, *formába önt*, képletesen *szavakban*, *szavakkal formál meg*, azaz *közöl*.

kód (*lat→fr*) **1.** *inf* megállapodás szerinti jelek v. szimbólumok rendszere, amellyel vmely információ továbbítható és visszaalakítható **2.** *biol* → genetikai kód **3.** rejtjeles ábécé kulcsa **4.** *inf* jelábécé (sürgönynél, távírónál stb.)

(fn) **1**. *Tud* Megállapodás szerinti jelek v. szimbólumok rendszere, amellyel vmely információ egyértelműen visszaadható. **2**. *ritk* Jelkulcs [nk:fr<lat]

code *n. m.* (1220; lat. jur. *codex* «planchette, recueil»). **1**. Recueil de lois. ... **4**. Recueil de conventions ; dictionnaire des équivalences entre deux langages (*spécialt*. un langage naturel et un langage non naturel). *Code de signaux*. *Code secret*. V. **Chiffre**

caudex, codex (*fa*)*törzs*, *rönk*, *tuskó*, *tönk*; *dokumentum*, eredetileg *fa írótábla*-ból. Ebből származik a magyar ***kódex*** szó.

code *h fn*, **1.** *jog* törvénykönyv, jogszabálygyűjtemény, kódex ... **2.** kód, jel-/betűkódex; **code binaire** bináris kód; **code biquinaire** bikvináris kód; **code cyclique** ciklikus kód; **code détecteur d'erreurs** hibakereső kód; **code génétique** genetikai kód; **code points-traits** Morze-ábécé; **code télégraphique** sürgönyjel-ábécé; *vill* **code temporel** időkód; **code à adresses multiples** többcímű kód; **code à redondance** redundáns kód; **code de contrôle** ellenőrző kód; **code de correction d'erreurs** hibajavító kód; **code de graph** gráfkód; **code de signaux** jelzési utasítás; **télégramme en code** rejtjeles, sifrírozott sürgöny; **code d'instructions** utasításrendszer; **mettre en code** kódol, rejtjelez **3.** kód(szám); **code-barre**, **code à barres** termékkód; **code génétique** genetikai kód; **code postal** irányítószám; **code de comptes** folyószámla (kód)száma; **code pour carte bancaire** PIN-kód **4. le code** a szabályzat

Vissza a tartalomhoz

Tárgymutató

F

G

Gy

H

I,Í

J

K

L

M

N

Ny

O,Ó

Vissza a tartalomhoz

Irodalom

Beutelspacher, A. — *Cryptology*
The Mathematical Association of America, 1994

Brassard, G. — *Modern Cryptology*
Springer Verlag, 1989

Buttyán, L.-Vajda, I. — *Kriptográfia és alkalmazásai*
Typotex, 2004

Diffie, B. W.-Hellman, M. E. — *New directions in cryptography*
IEEE Trans. on Info. Theory, IT-22 (1976), pp. 644-654

Kahn, D. — *The Codebreakers The story of secret writing.*
MacMillan Publishing Company, 1967

Ködmön, J. — *Kriptográfia*
Computerbooks, 1999/2000

Menezes, A. J.-Oorschot, P. C. van-Vanstone, S. A. — *Handbook of Applied Cryptography*
CRC Press LLC, 1997

Nemetz, T.-Vajda, I. — *Algoritmusos adatvédelem*
Akadémiai Kiadó, 1991

Poe, E. A. — *Az aranybogár*
Magyar Helikon, 1972

Révay, Z. — *Titkosírások*
Lazi Könyvkiadó, 2001

Salomaa, A. — *Public-key Cryptography*
Springer Verlag, 1990

Shannon, C. E. — *Communication Theory of Secrecy System*
Bell Syst. Tech. J., vol. 28, 1948., pp. 656-715
(Először „A Mathematical Theory of Cryptography", Sept. 1, 1946, egy bizalmas jelentésben)

Shannon, C. E. — *A Mathematical Theory of Communication*
Bell System Technical Journal, vol 27 (1948), pp. 379-423, 623-656
(magyarul egy javított változat „*A hírközlés matematikai elmélete*" címmel A kommunikáció matematikai elmélete-ben)

Singh, S. — *Kódkönyv. A rejtjelezés és a rejtjelfejtés története*
Park Könyvkiadó, 2001

Tilborg, H. C. A. van — *An Introduction to Cryptology*
Kluwer, 1989

Virasztó, T. — *Titkosítás és adatrejtés*
NetAcademia Kft, 2004