Titkosítás

Uhlár László

1. Miért?

Talán egy idős lehet az emberiséggel az igény arra, hogy bizonyos személyes dolgainkat mások elől elrejtsünk. Titkosírások tömkelege alakult ki a történelem során, amelyek mind azt a célt szolgálták, hogy üzeneteinket, leveleinket, vagy csak saját használatra készített feljegyzéseinket illetéktelenek ne tudják elolvasni. Nincs ez másként a mai modern időkben sem. Elektronikus üzeneteinket, fájljainkat szeretnénk mások elől elrejteni.

2. Módszerek

Kezdetben megfelelőnek számított egy egyszerű karaktercserés titkosítás: minden karaktert ugyan arra a másik karakterre cserélünk. Ezt „kézzel" megfejteni eléggé nehézkes volt, de a mai, számítógépekkel támogatott megfejtőknek ez már nem okozna gondot.

2.1. „Jó algoritmus"

Mitől lesz egy titkosítási eljárás jó? Talán elfogadhatjuk válasznak azt, hogy a megfejtéséhez szükséges energia befektetés nem áll arányban a megfejtéssel szerezhető nyereséggel.

2.2. Szimmetrikus titkosítás

Szimmetrikusnak nevezünk egy titkosítási módszert, ha ugyan azzal a kulccsal kell visszafejteni egy kódolt állomány, mint amivel titkosították. Lehetőleg titkolni kell magát az algoritmust, mindenképpen titkolni kell a kulcsot. Ez felveti a kulcs terjesztésének a kockázatát. Hogyan lehet úgy megegyezni a titkosítási algoritmusban, kulcsban, ha erre csak a nyílt hálózat ad lehetőséget, személyes találkozóra nincs lehetőség, hogy mások elől ez rejtve maradjon?

2.3. Aszimmetrikus titkosítás

Olyan eljárás, amely során más kulcsot használunk a titkosításhaz és megint másikat a visszafejtéshez. Jó példa lehet erre a fordítás: magyarról angolra egy magyar-angol szótárral fordítunk szöveget (kódolunk), visszafejtéshez viszont az angol-magyar szótárt használjuk.

2.4. Brute force

Egy tikosítás feltörési módszer („Brutális erő"), melynek során az algoritmus ismeretében sorra az összes szóba jöhető kulcsot kipróbáljuk. Átlagosan nyilván a kulcsok felénél várható a siker, ezért ellene például olyan módszerrel lehet megpróbálni védekezni, ahol olyan mennyiségű a kulcsok száma, hogy gyakorlatilag feltörhetetlennek tekinthető. (Ha az én egyik titkosított e-mail-emet a Föld összes számítógépének „ráállításával" is csak év ezredek alatt lehetne feltörni, akkor ez egy feltörhető eljárás ugyan, de számomra kellő biztonságot nyújt: gyakorlatilag feltörhetetlen). A mai modern eljárásokban a kulcsok valamilyen számok, amelyeknél a lehetséges kulcsok száma (és ezzel az átlagos próbálgatási idő) a szám nagyságától függ. Ezt általában bitben adják meg, így láthatunk 56 bites, 128 bites stb. kulcsokat leírva.

3. Ismertebb szimmetrikus titkosítások

3.1. DES

Data Encryption Standard nevű titkosítás ma már korszerűtlennek számít. A mai technikai eszközökkel már emberi időben is, azaz hatékonyan feltörhető.
A DES tehát egy szabványosított matematikai algoritmus, amelyet számítógépes adatok kriptográfiai védelmére vezettek be. Az algoritmust bináris adatokhoz fejlesztették ki, amelyeket 64 bites blokkokra osztva titkosította és ehhez 64 bites kulcsot használt. A 64 bit elsőrendű fontossággal bír, mert vele egy 64 bites nyílt szövegből ugyancsak 64 bites titkosított szöveget állít elő egy 64 bites kulccsal, de ez nem jelenti azt, hogy a titkosítási eljárás során mindvégig csak 64 bites szóhosszúsággal dolgozik. Mivel a DES mint eljárás teljesen nyilvános, ezért a titkosítás kriptográfiai biztonsága a kulcs védelmének a biztonságától függ. A titkosított szöveg megfejtésekor az algoritmust fordított sorrendben kell lefuttatni ugyanazzal a kulccsal, amellyel az eredeti nyílt szöveget titkosították.

3.2. Az IDEA

IDEA (International DataEncryption Algorithm - nemzetközi adat titkosító eljárás) szintén 64 bites adatblokkokkal dolgozik, de ehhez 128 bites kulcsot használ. A kilencvenes években fejlesztették ki kifejezetten titkosított adatátvitel megvalósítására.

3.3. Az AES

AES (Advanced Encryption Standard). A kilencvenes években a DES felváltására kiírt pályázat győztese a Rijndael algoritmus lett, ezt nevezik ma AES-nek. A hivatalosan kiadott AES szabvány a Rijndael algoritmus 128 bites változatát tartalmazza.
A fenti módszerek részletes tárgyalása matematikai tudásunkat meghaladja.

4. Nyilvános kulcsú titkosítás (PKE)

4.1. Az elv

A titkosított kommunikációban résztvevő minden félnek két kulcsa van: egy nyilvános és egy titkos, ezek egyenrangúak: amit az egyikkel titkosítunk, azt a másikkal lehet visszafejteni. A nyilvános kulcs ismeretében a titkos kulcs nem (vagy csak nagyon nehezen) tudható meg. A nyilvános kulcsokat mindenki nyilvánosságra hozzá (ezért nyilvános). Tételezzük fel, hogy A és B kommunikálni szeretne. A(ny) és B(ny) lesznek a megfelelő nyilvános kulcsok, A(t) és B(t) a titkos kulcsok. Ha A szeretne írni B-nek, akkor B nyilvános kulcsával titkosítja az üzenetet. Mivel a tikos és a nyilvános kulcsok összetartoznak, ezért visszafejtéshez a titkos kulcsot kell használni. Így tehát B-nek írni bárki tud, de azt megfejteni csak a B(t) tulajdonosa, jelesül B tudja. Ha találunk ilyen algoritmust, akkor az a kulcsok kiosztásának a gondját is megoldja, ami a szimmetrikus tikosításnál komoly probléma, vagyis a két fél hogyan egyeztessen a titkosítás kulcsáról?
Sőt! A fenti algoritmus sok egyéb pozitív következménnyel is jár. Ha B bizonytalan abban, hogy tényleg A-tól jött az üzenet (bárki ráírhatja, hogy A vagyok !), akkor megkérheti A-t, hogy az üzenet elküldése előtt titkosítsa azt még egyszer az ő titkos kulcsával (A(t)). Ezt a kétszeresen titkosított üzenetet A nyilvános kulcsával lehet először kibontani, ezt bárki megteheti A(ny) ismeretében, és ez ugye nyilvános. Ebből a sikeres visszafejtésből következik, hogy csak olyan valaki „zárhatta be", aki A(t) birtokában van, azaz csakis A küldhette azt. Ez a digitális aláírás elve. Utána B(t) segítségével B visszafejti a neki szóló üzenetet. (ezt a lépést már csak B tudja megtenni,

mert csak neki van meg B(t)). A gyakorlatban gyakran csak az aláírás történik meg, ezzel hitelesítve az iratot. Általában nem az egész dokumentumot tikosítja A az ő tikos kulcsával, hanem a titkosítatlan dokumentum mellé (akár külön fájlban) helyezik el az aláírást, amely a dokumentum valamilyen „lenyomatának" a tikosításával jön létre.

4.1.1. A hash

A hash (hasító) függvények tetszőleges méretű adatból azonos méretű lenyomatot készítenek. Természetesen a bemenet tetszőleges hosszúságú lehet, a kimenet fix méretű. Ezek a függvények, algoritmusok igyekeznek a lehetséges ütközéseket minimalizálni, azaz minimálisra csökkenteni annak az esélyét, hogy egy üzenetet megváltoztatva annak hash értéke ne változzon meg. Legelterjedtebbek az SHA és az MD algoritmusok különböző típusai.
Tehát digitális aláírásnál lehetséges, hogy csak az üzenet valamely eljárással készült hash értékét kódoljuk A(t)-vel, ezt illesztjük a nyilvános üzenet mellé. Az ellenőrzés során A(ny)-vel „kinyitjuk" a kódolt aláírást, majd összevetjük az eredeti üzenet hash értékét a kódoltan érkezővel. Ha megegyezik, akkor az eredeti üzenet bár kódolatla, de nem változott meg (változtatták meg szándékosan!), és tényleg A küldte.

4.1.2. A tanúsítványok

Még egy „kis" problémát kell megoldani: ha valaki küld nekem egy nyilvános kulcsot, hogyan győződhetek meg arról, hogy tényleg tőle kaptam? Ha személyes találkozás során történik ez, akkor minden rendben, de ekkor már egy szimmetrikus módszerben, kulcsban is megegyezhetnénk, ezeknek általában lényegesen kisebb a számítási teljesítmény igényük. Több megoldás is felmerülhet. Ha személyről van szó, akkor felhívom telefonon és megkérem, hogy a kulcsának valamilyen részletét olvassa be. Más lehetőség, hogy egy harmadik féltől kérem a hitelesség igazolását: ha C nyilvános kulcsában abszolut megbízok (pl. személyesen adta át), és C ismeri A-t, akkor ha C aláírná a saját tikos kulcsával (C(t)) A nyilvános kulcsát, ezzel garanciát vállalván arra, hogy az A-tól van (lehet, hogy ők személyesen találkoztak), én is megbízhatnék abban, hogy A-é a kulcs. Tehát megjön valahogyan A(ny) C(t)-vel titkosítva, aláírva. Ezt C(ny) birtokában vissza tudom fejteni, mivel sikerült, ez bizonyítja, hogy C írta alá, hiszen C nyilvános kulcsában megbízok. Az aláírásával C kezeskedik arról, hogy a nyilvános kulcs A-tól van, ezek után már A(ny)-t is hitelesnek fogadhatom el. A jövőben ha kapok egy nyilvános kulcsot D-től, amit A aláírt, mivel már A hitelességéről meggyőződtem, D

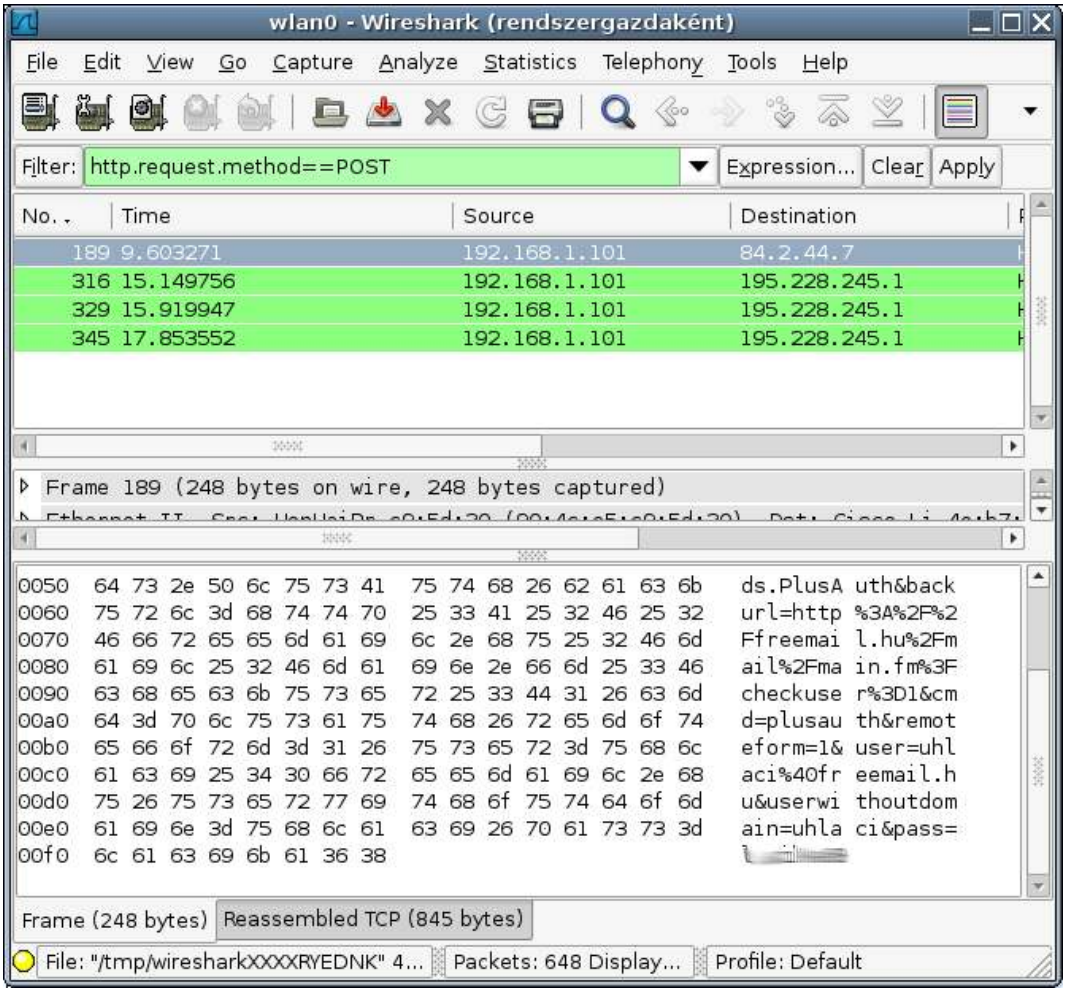
nyilvános kulcsában is megbízok. A fenti elv megvalósítása a bizalmi háló. Ennek egy másik megvalósulása a tanúsítvány kiszolgálók használata: bizonyos hivatalok, cégek aláírását hitelesnek fogadjuk el (miért is???), majd pedig ha egy hozzám küldött nyilvános kulcsot valamelyik hitelesítés szolgáltató aláírta, akkor elfogadom, hogy a kapott nyilvános kulcs tényleg a feladótól származik. (ez az aláírás természetesen pénzbe kerül, az ilyen aláírók a CA-k, a Certificate Authority-k). A tikosított csatornán folyó böngészés (https) ilyen „megbízható harmadik fél" által aláírt tanúsítványokon alapszik.

4.2. A megvalósítás

A huszadik század végéig nem hitte senki, hogy a fenti feltételeknek eleget tevő algoritmus létezhet. Legismertebb megvalósítása az RSA algoritmus, amelynek kidolgozása három matematikus nevéhez fűződik (Rivest, Shamir és Adleman) a hetvenes évekből. Ugyan ezt valósítja meg a DSA és a Diffie-Hellmann algoritmus is.

5. Az SSL

Az SSL (Secure Socket Layer) egy protokoll réteg, amely a szállítási rétegbeli protokoll (pl. TCP/IP) és valamely alkalmazási rétegbeli protokoll (pl. HTTP) között helyezkedik el, az OSI terminológia szerinti viszony- és megjelenítési réteg feladatait látva el. Webböngészésnél például az SSL biztosítja a biztonságos kommunikációt a kliens (böngésző) és a szerver (webszerver) között. Autentikációhoz digitálisan aláírt tanusítványokat használ, a kommunikáció titkosítva zajlik (az SSL handshake során közösen megegyeznek egy kulcsban, ebből generálják azután az egy folyamat erejéig használatos session keyt, és ezt használják valamely szimmetrikus titkosító algoritmussal, pl. DES, AES, stb). személyes adataink védelmében jó tudni, hogy bejelentkezési neveink, jelszavaink „bárki" által könnyedén megszerezhető módon haladnak a hálózat eszközein, hacsaknem tikosítjuk azokat. Ezért biztonsági szempontból óvatosan használjuk az ftp, telnet, rsh, stb programokat
Webes bejelentkezésnél hacsak lehet, válasszuk a biztonságos opciót, ahol erre mód van. Álljon itt példának a freemail.hu oldalra történő bejelentkezés. Egy kattintással beállíthatom, hogy biztonságosan szeretnék bejelentkezni, azaz a hitelesítő adataim tikosítva kerülnek elküldésre. Ha nem ezt választom, akkor a belépési nevem, jelszavam gond nélkül ellopható. Több program is van erre, egyszerűen használható, grafikus felületű a Wireshark nevű. A mellékelt kép e program egyik kimenetét mutatja, ahol az alsó sorban bizony a freemai fiókhoz tartozó jelszavamat lehetne olvasni.



6. A PGP

Fontos megemlíteni ezt a megvalósítást is. Valójában a nyilvános kulcsú titkosítás elterjedése a polgári életben a pgp kidolgozójának, Zimmermannak köszönhető. Aki szereti a krimibe illő történeteket, olvasson utána az interneten. A pgp jelentése: Pretty Good Privacy.

7. A GPG

A GNU által létrehozott megvalósítás, jelentése GNU Privacy Guard. Szinte minden linux disztribúcióban alapból települ. Ezzel közelebről is megismerkedünk a gyakorlatokon. Addig is: man gpg.

8. Az SSH

Az ssh voltaképpen egy hálózati protokoll, amely biztonságos távoli bejelentkezést tesz lehetővé. Az ssh szerver és kliens hostok a kommunikáció elején aszimmetrikus titkosítás használatával megegyeznek egy kulcsban, amelyet e kommunikáció folytatásában a szimmetrikus titkosításhoz használnak majd. A távoli bejelentkezésen túl lehetőség van tunnelezésre is, azaz egy biztonságos csatornát lehet így kialakítani más protokollok számára.
Használatához a távoli szerveren szükség van egy accountra (név, jelszó). Az utasítás: `ssh valaki@ip.vagy.domain.név`. Ez után kell beírnunk a jelszavunkat. Windowsra is létezik ssh kliens program, ez a putty. Szabadon letölthető, használható.

9. Véletlen számok

A fentebb vázolt digitális tikosítási folyamtok fontos lépése a titkosításhoz használt kulcsok előallítása. Alapvető fontosságú, hogy ezek véletlenül legyenek kiválaszva. A gyakorlati életben véletlen eseménynek szoktuk elfogadni például a kockadobás eredményét. Rávehető arra egy program, hogy „kockázzon”?

9.1. Valódi véletlen

Valódi véletlen sorozatnak nevezzük az olyan bitsorozatot, amelynek a következő eleme az előző elemek ismeretében nem következtethető ki – vagyis nem létezik olyan algoritmus, amivel több, mint 50% valószínűséggel meg tudjuk jósolni a következő elemet. (Erre a kikiötésre azért van szükség, mert ha teljesen véletlenszerűen tippelünk, akkor nagy átlagban 50% valószínűséggel leszünk sikeresek – ennél jobb algoritmusnak nem szabad léteznie.) Azaz a valódi véletlen sorozat megjósolhatatlan.
A definícióból természetesen következik, hogy valódi véletlen sorozatokat nem lehet kizárólag algoritmikus módszerekkel (szoftveresen) előállítani, azaz kiszámítani. Valódi véletlen sorozat előállításához szükségünk van olyan jelenségre, amelynek kimenetelét nem tudjuk előre megjósolni; ilyen lehet például egy pénzérme feldobása (fej vagy írás), vagy ha egy zárt térben rádioaktív atomokat helyezünk el és egy Geiger-Müller-számláló figyeljük hogy a számláló kattanásakor óránk másodpercmutatója páros vagy páratlan szám volt-e. Mivel a véletlen számokra sok számítógépes alkalmazásnál is szükség van, újabban kifejlesztettek olyan eszközöket is, amelyek számítógéphez csatlakoztathatók vagy abba beépítettek, és például a levegő páratartalma vagy

nyomása alapján állítanak elő kimeneti értéket. Az ilyen eszközök azonban a kriptográfiai algoritmusok sebességéhez képest sokkal lassabbak, valamint felhasználásuknak gátat szab, hogy a legtöbb felhasználónál általában nem állnak rendelkezésre. Ezért helyettük a gyakorlatban inkább olyan módszereket alkalmaznak, amelyek ugyan nem valódi véletlent állítanak elő, ellenben szoftveresen megvalósíthatóak.

9.2. Ál-véletlen generátorok

Az ál-véletlen generátorok (pszeudo-random generátorok) olyan bitsorozatot állítanak elő, amelyek reális időn belül megkülönböztethetetlenek egy valódi véletlenszám generátortól – vagyis nem létezik olyan algoritmus,amely több, mint 50% valószínűséggel kizárólag a kimenetek ismeretében el tudja dönteni, hogy két generátor közül melyik a valódi véletlenszám generátor és melyik az álvéletlen generátor.
Ez a feltétel egyben azt is jelenti, hogy az álvéletlen generátor által előállított sorozat következő eleme az addigiak ismeretében reális időn belül megjósolhatatlan.
A gyakorlatban használt ál-véletlen generátorok a véletlen előállításához valamilyen elegendően „véletlennek tűnő" forrást használnak fel (pl. a rendszeridő), majd algoritmikus módszerekkel ebből készítenek szükséges hosszúságú ál-véletlen sorozatot. Természetesen fontos kritérium, hogy a felhasznált véletlen „mag" ne legyen utólag kikövetkeztethető, mert az visszaélésekre adhat módot (például ha a generátort kriptográfiai kulcs előállításra használjuk).

9.2.1. Linuxban

A /dev/random és a /dev/urandom eszközök
Ezek a rendszermag véletlenszám-generátorai. A /dev/random a rendszer hardverelemeinek entrópiáját használja fel a számok generálására. Ha elfogy a használható entrópia, akkor várnia kell, amíg újra összegyűlik annyi, hogy újabb számokat olvashasson ki belőle. A /dev/urandom hasonló elv szerint működik. Kezdetben ez is a rendszerhardver entrópiáját használja, de ha ez elfogy, akkor is folytatja a számok küldését, egy ál-véletlenszám-generátor képlet segítségével. Ezt a módszert kevésbé biztonságosnak tartják, ha olyan életbe vágóan fontos célokról van szó, mint például titkos kulcspárok generálása.
adjuk ki a következő utasítást: `cat /dev/random`, illetve `cat /dev/urandom`, megszakítani Ctrl-c-vel lehet. (man urandom!!!)

10. Hardveres véletlenszám generátorok

Napjaink hardveres megoldásai a kvantumfizika eredményeit használják fel valódi véletelen események előállítására. A kvantumfizika a véletlenről szól! Vannak eszközök, amelyek a hőmérsékleti sugárzás mért értékeiből dolgoznak, mások a radiokatív bomlás véletlenszerűségét használják ki. Akik „nagyon nagy" biztonságra vágynak és nem elégednek meg a pszeudo-random generátorok nyújtotta „véletlennel", azok ilyen eszközöket vásárolhatnak.