Óbudai Egyetem

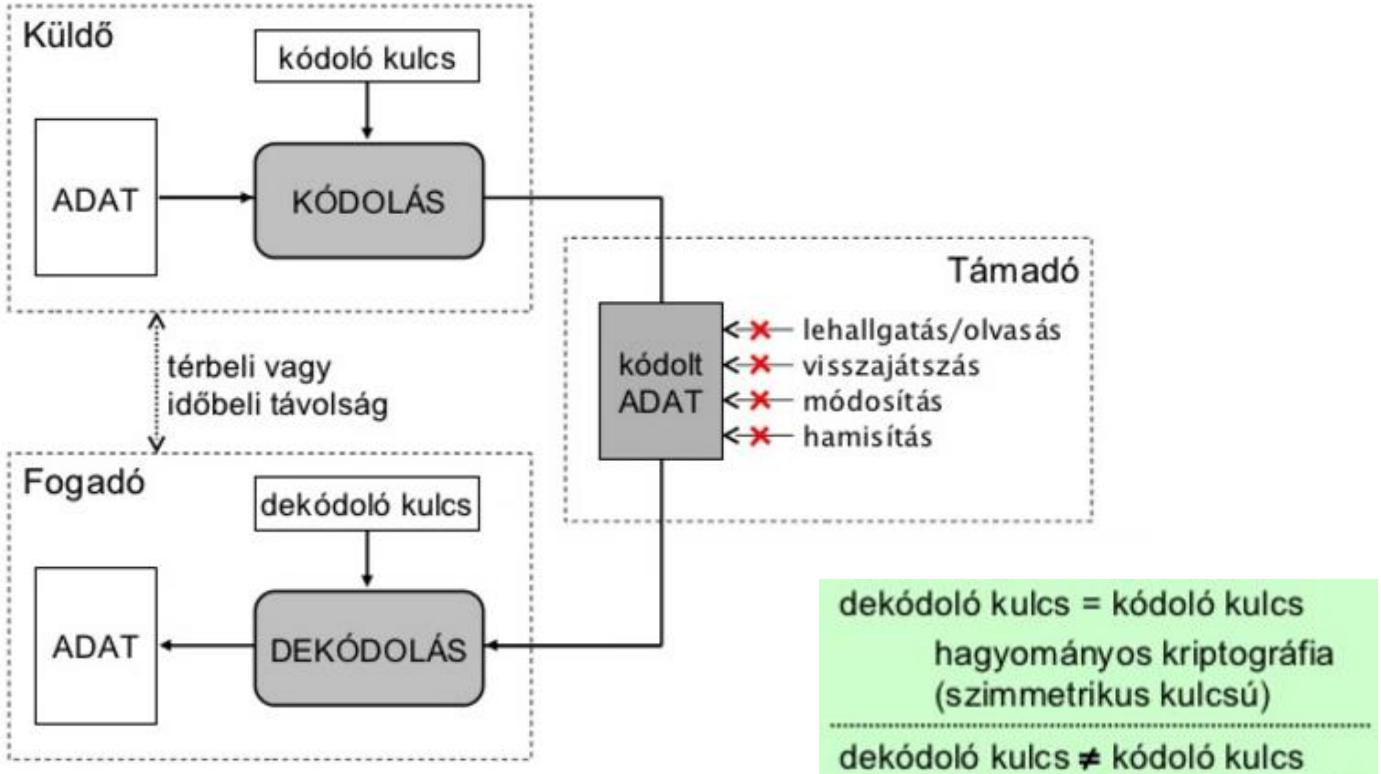
Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar

Műszaki területek informatikai biztonsága
Nyilvános kulcsú kriptográfia

ZENTAI DÁNIEL

ZENTAI.DANIEL@BGK.UNI-OBUDA.HU

A titkosítás alapmodellje



dekódoló kulcs = kódoló kulcs
hagyományos kriptográfia
(szimmetrikus kulcsú)

dekódoló kulcs ≠ kódoló kulcs
nyílvános kulcsú kriptográfia
(aszimmetrikus kulcsú)

Az RSA titkosítás

Ronald Rivest, Adi Shamir, és Leonard Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, 1978

Ronald Rivest

Adi Shamir

Leonard Adleman

Az RSA titkosítás

1969-ben Ellis rájött, hogy nyilvános kulcsú kriptográfia lehetséges (nem-titkos kódolásnak nevezte).
1973-ban Cocks kitalálta a később RSA néven ismertté vált kódolást.
1974-ben Williamson (Cocks barátja) felfedezi a később Diffie-Hellman kulcscsere néven ismertté vált eljárást.
1975-re Ellis, Cocks, és Williamson a nyilvános kulcsú kriptográfia összes alapvető tételét kidolgozta, de hallgatniuk kellett 1997-ig.


James Ellis


Clifford Cocks


Malcolm Williamson

Az RSA titkosítás

Definíció (kongruencia): a, b, n egész számok $a \equiv b \pmod{n}$, ha a és b ugyanazt a maradékot adja n-nel osztva. (ejtsd: a kongruens b-vel modulo n)

Pl.: $16 \equiv 4 \pmod{6}$, $11 \equiv 3 \pmod{4}$

Definíció (multiplikatív inverz): x multiplikatív inverze mod n az az y, amelyre teljesül, hogy $x \cdot y \equiv 1 \pmod{n}$

Ilyen csak akkor létezik, ha lnko(x,n)=1 (pl. 4-nek nincs „reciproka” mod 6)

Pl.: $3 \cdot 2 \equiv 1 \pmod{5}$, ezért 3-nak 2 a multiplikatív inverze mod 5

Az RSA titkosítás

Definíció (Euler-féle φ függvény) n természetes számhoz φ(n) azt mutatja meg, hogy hány olyan n-nél nem nagyobb szám van, ami n-hez relatív prím (azaz lnko-juk 1).

Pl.: 6-hoz relatív prímek: 1, 5, azaz φ(6)=2

A φ két fontos tulajdonsága:
Ha lnko(a,b)=1, akkor φ(a)·φ(b)=φ(a·b).
Ha p egy prímszám, akkor φ(p)=p-1.

Pl.: lnko(3,5)=1, φ(3)=2, φ(5)=4, φ(15)=8 de lnko(2,4)=2, φ(2)=1, φ(4)=2, φ(8)=4

Egy fontos következmény: Ha p és q prímek, akkor φ(p·q)=(p-1)·(q-1)

Az RSA titkosítás

Kulcsgenerálás:

1. Választunk 2 nagy prímet: p és q
2. Kiszámoljuk az $n=p \cdot q$ szorzatot.
3. Kiszámoljuk $\varphi(n)$-et: $\varphi(n)=\varphi(p \cdot q)=(p-1) \cdot (q-1)$
4. Választunk egy olyan e számot, amire $1<e<\varphi(n)$ és lnko$(e,\varphi(n))=1$
5. Kiszámoljuk e multiplikatív inverzét mod $\varphi(n)$, legyen ez a szám d (ezt meg lehet csinálni kiterjesztett euklideszi algoritmussal)

- A nyilvános kulcs: e, n
- A titkos kulcs: d

Az RSA titkosítás

Titkosítás: A feladó az m nyílt szöveget titkosítja a címzett nyilvános kulcsával, e-vel, c a rejtett szöveg: $c \equiv m^e \pmod{n}$

Dekódolás: A címzett a c rejtett szöveget dekódolja a saját titkos kulcsával, d-vel: $m \equiv c^d \pmod{n}$

Az RSA titkosítás

Miért működik?

Euler-Fermat tétel: Ha lnko(a,n)=1, akkor $a^{\varphi(n)} \equiv 1 \pmod{n}$.

Miért lesz $m \equiv c^d \pmod{n}$?

d-t úgy választottuk, hogy $e \cdot d \equiv 1 \pmod{\varphi(n)}$, azaz valamilyen k egész számra $e \cdot d = k \cdot \varphi(n) + 1$.

Ezért $\mathbf{c^d = (m^e)^d = m^{e \cdot d} = m^{k \cdot \varphi(n)+1} = (m^{\varphi(n)})^k \cdot m}$, és az Euler-Fermat tételből következik (feltéve hogy lnko(m,n)=1), hogy $\mathbf{(m^{\varphi(n)})^k \cdot m \equiv 1^k \cdot m \equiv m \pmod{n}}$, azaz $c^d \equiv m \pmod{n}$

Az RSA titkosítás

Miért biztonságos?

Tegyük fel, hogy egy támadó meg akarja tudni m-et c ismeretében. A támadó d-t nem ismeri, csak e-t és n-et. Azt tudja, hogy d az e multiplikatív inverze mod φ(n). Ezt ki lehetne számolni φ(n) ismeretében. Tudjuk, hogy $\varphi(n)=(p-1)\cdot(q-1)$, de p és q sem publikus.

Tehát a támadónak p-t és q-t kell kiszámolnia n ismeretében, azaz n prímtényezős felbontását kell kiszámolnia: ez a faktorizációs probléma

Egy számról gyorsan el lehet dönteni, hogy prím –e. Viszont nem ismerünk gyors algoritmust egy szám faktorizálására. 1024 bites (kb. 300 decimális jegy) számoknál ez a számítógépeknek is csillagászati ideig tart, ha jól választjuk meg a paramétereket.

Az RSA aláírás

Kulcsgenerálás:

1. Választunk 2 nagy prímet: p és q
2. Kiszámoljuk az $n=p \cdot q$ szorzatot.
3. Kiszámoljuk $\varphi(n)$-et: $\varphi(n)=\varphi(p \cdot q)=(p-1) \cdot (q-1)$
4. Választunk egy olyan e számot, amire $1<e<\varphi(n)$ és lnko$(e,\varphi(n))=1$
5. Kiszámoljuk e multiplikatív inverzét mod $\varphi(n)$, legyen ez a szám d (ezt meg lehet csinálni kiterjesztett euklideszi algoritmussal)

- A nyilvános kulcs: e, n
- A titkos kulcs: d

Az RSA aláírás

Aláírás: A feladó az m üzenetet aláírja a saját titkos kulcsával, d-vel, s az aláírás: $s \equiv m^d \pmod{n}$

Verifikáció: A címzett a feladó nyilvános kulcsával, e-vel ellenőrzi az aláírást: Ha $s^e \equiv m \pmod{n}$, akkor az aláírás valid.

Az RSA aláírás

Hash függvény: A hash függvények tetszőleges hosszúságú bitsorozatot képeznek le egy rögzített hosszúságú bitsorozatba.

A hash függvények legelterjedtebb használata: jelszó tárolás, digitális aláírás

A gyakorlatban nem magát az üzenetet szokás aláírni, hanem annak a hash értékét. Tehát ha h egy hash függvény, akkor az aláírás $s \equiv h(m)^d \pmod{n}$ és a verifikáció $s^e \equiv h(m) \pmod{n}$.

Fontos, hogy h ütközés ellenálló hash függvény legyen (azaz nehéz legyen olyan m-et és m'-t találni, amire h(m)=h(m')).

RSA paraméterválasztás

Rosszul választott paraméterek esetén a faktorizációs probléma gyorsan megoldható!

Ha p és q egymáshoz közeli prímek, akkor ezek közel vannak [√n]-hez is (egészrész). Próbáljuk n-et elosztani a [√n]+1, [√n]+2, … számokkal. Ha valamelyik szám megvan n-ben maradék nélkül, akkor sikerült faktorizálni n-et.

Kis d esetén d gyorsan kiszámolható (teljes feltörés): Wiener tétel: Ha $d < (1/3)\cdot n^{1/4}$, akkor a támadó ki tudja számolni d-t.

Pl.: Ha $n=10^{300}$, akkor $d=3\cdot 10^{74}$ esetén már törhető az RSA.

RSA paramétervválasztás

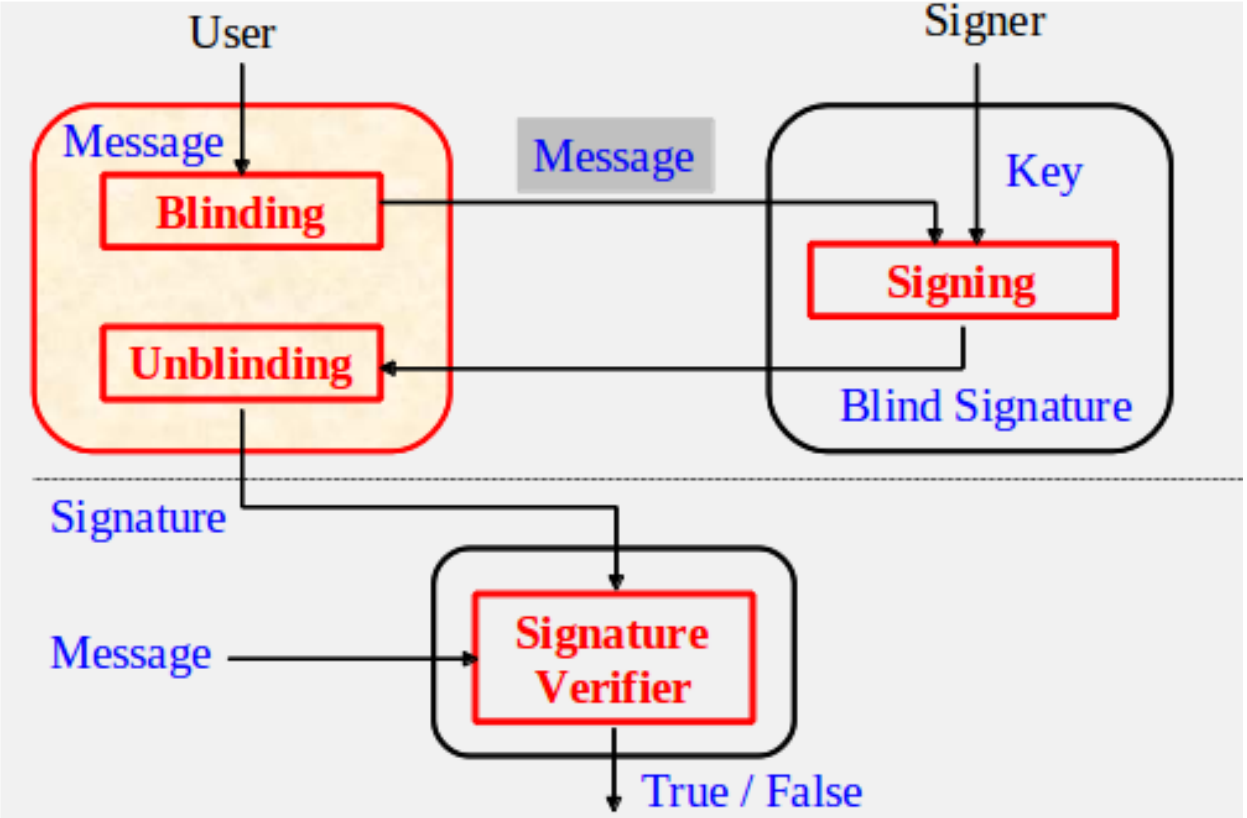
Kis e eset bizonyos üzeneteket meg lehet fejteni (de nincs teljes törés):

Coppersmith tétel: Meg lehet fejteni d ismerete nélkül az olyan üzeneteket, melyek hossza kisebb, mint $n^{1/e}$.

Ha e nagy, akkor csak az 1-2 bit hosszúságú üzeneteket lehet feltörni, de pl. e=3 és $n=10^{300}$ esetén minden legfeljebb 100 jegyű (kb. 300 bites) üzenetet meg tudunk fejteni.

Ha e-vel megegyező számú üzenetet kiküldünk ki ugyanazzal a nyilvános kulcscsal titkosítva, akkor visszafejthető az üzenet.

RSA vak aláírás



RSA vak aláírás

Legyenek n, e, d a szokásos RSA paraméterek, de most e és d nem a feladó, hanem az aláíró kulcspárja.

A feladó választ egy r véletlen számot, amire lnko(r,n)=1 teljesül, és kiszámolja az $m' \equiv m \cdot r^e \pmod n$ értéket, és elküldi ezt az aláírónak.

Az aláíró aláírja ezt a saját titkos kulcsával: $s' \equiv (m')^d$ és visszaküldi ezt a feladónak.

A feladó kiszámolja $s \equiv s' \cdot r^{-1} \pmod n$-et, azaz s'-t beszorozza r multiplikatív inverzével. Ekkor $s \equiv m^d \pmod n$, ugyanis:

$s \equiv s' \cdot r^{-1} \equiv (m')^d \cdot r^{-1} \equiv (m \cdot r^e)^d \cdot r^{-1} \equiv m^d \cdot r^{ed} \cdot r^{-1} \pmod n$, de azt tudjuk, hogy $r^{ed} \equiv r \pmod n$, ezért $m^d \cdot r^{ed} \cdot r^{-1} \equiv m^d \cdot r \cdot r^{-1} \equiv m^d \pmod n$.

Tehát a feladó birtokában van az aláírásnak úgy, hogy az aláíró nem tudta elolvasni z üzenetet.

Köszönöm a figyelmet!