RSA algoritmus

Smidla József

Rendszer- és Számítástudományi Tanszék
Pannon Egyetem

2012. 3. 27.

Tartalom

Aszimmetrikus kódolók

Szimmetrikus kódolók

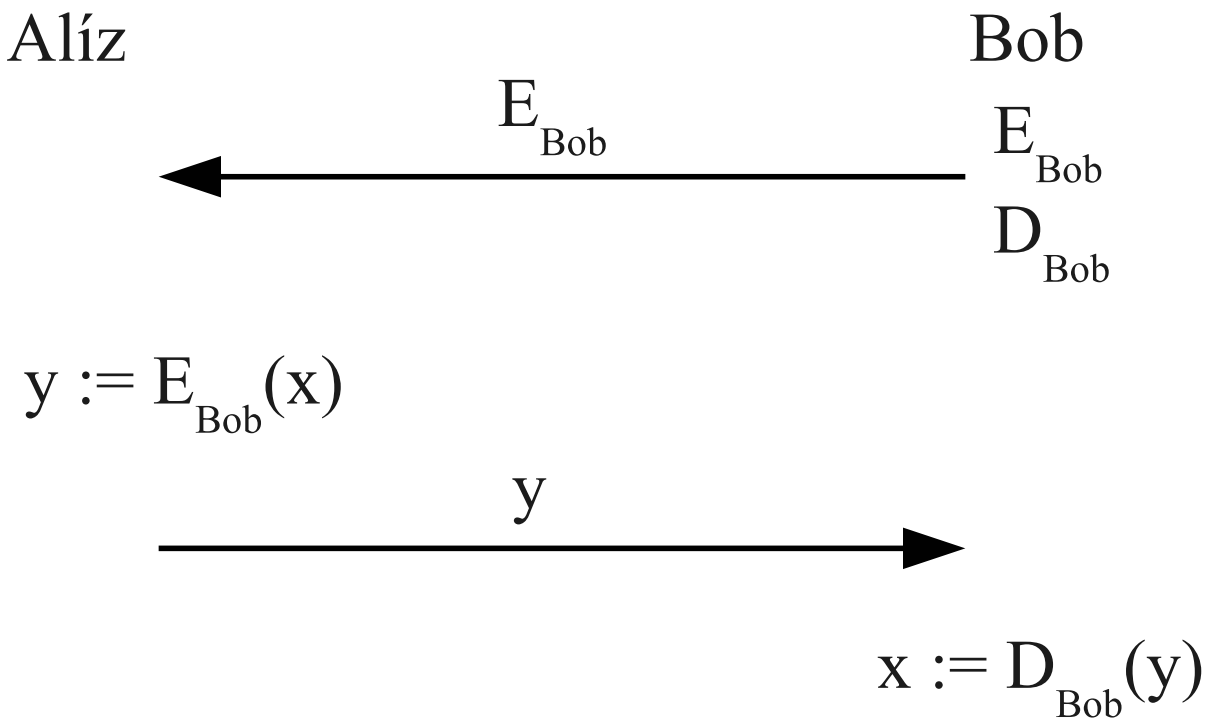
- Ugyanazt a kulcsot használja a kódoló, és a dekódoló is
- Gyors kódolás és dekódolás
- Probléma: A közös kulcsban meg kell egyezni

Aszimmetrikus kódolók

- A kódoláshoz és dekódoláshoz használt kulcsok eltérőek
- Mindenki rendelkezik egy privát és egy publikus kulccsal
- A privát kulcsot nehezen határozhatjuk meg a publikus kulcs alapján
- Több nagyságrendel lassabbak, mint a szimmetrikus kódolók

Aszimmetrikus kódolók

- Alíz szeretne üzenetet (x) küldeni Bobnak
- Bob nyilvános, azaz kódoló kulcsa: E_{Bob},
- Bob privát, azaz dekódoló kulcsa: D_{Bob}



Matematikai alapok

Legnagyobb közös osztó

Definíció

A és B egész számok legnagyobb közös osztója az a legnagyobb szám, amely osztója A-nak és B-nek is

Példa

- lnko(84, 30) = ?
- 84 = 2 * 2 * 3 * 7
- 30 = 2 * 3 * 5
- A közös osztók: 2 és 3, azaz lnko(84, 30) = 2 * 3 = 6
- A prímtényezős felbontás nagy számok esetén gyakorlatilag kivitelezhetetlen

Euklideszi algoritmus

Bemenet: A és B egész számok, $A > B$

1. $R := A \bmod B$
2. Ha $R = 0$, akkor végeztünk, a legnagyobb közös osztó $= B$
3. $A := B$
4. $B := R$
5. Vissza az 1. pontra

Példa

$A = 84$, $B = 30$

1. $R = 84 \bmod 30 = 24$
   - $A = 30$, $B = 24$
2. $R = 30 \bmod 24 = 6$
   - $A = 24$, $B = 6$
3. $R = 24 \bmod 6 = 0$
   - Vége, legnagyobb közös osztó $= 6$

Multiplikatív inverz

Definíció

Egy A szám multiplikatív inverze az a B szám modulo M-ben, amire igaz, hogy:

$$AB \equiv 1 \ (\ \mathrm{mod}\ \mathrm{M}\ )$$

Példa

- Legyen A = 23, M = 120
- Ha B = 47, akkor: $AB = 1081$
- 1081 mod 120 = 1
- Tehát 23 multiplikatív inverze modulo 120 esetén 47

Multiplikatív inverz meghatározása

- Az inverzet a kiterjesztett euklideszi algoritmussal határozhatjuk meg
- a és b adottak, az algoritmus x és y egészek értékét határozza meg az alábbi egyenletben:

$$ax + by = \text{lnko}(a, b)$$

- Legyen x az a multiplikatív inverze modulo m-ben:

$$ax \equiv 1 \ (\ \text{mod m}\ )$$

- Azaz m osztója az $ax - 1$-nek, az osztás eredménye legyen q:

$$ax - 1 = qm$$

- Ezt átrendezve:

$$ax - qm = 1$$

- Ha lnko(a, m) = 1, akkor a kiterjesztett euklideszi algoritmus megadja x-et és q-t, x lesz a keresett inverz

Kiterjesztett euklideszi algoritmus

Határozzuk meg az lnko($a = 120$, $b = 23$)-at:

Osztandó	Osztó	Hányados (q)	Maradék (r)
120	23	5	5
23	5	4	3
5	3	1	2
3	2	1	1
2	1	2	0

Fejezzük ki az i. maradékot, azaz r_i-t a és b alapján:

$$r_i = ax_i + by_i$$

A táblázatból kiolvasható, hogy az $i > 2$ esetén

$$r_i = r_{i-2} - q_i r_{i-1}$$

Az előző kettő összefüggésből a következőt kapjuk:

$$r_i = (ax_{i-2} + by_{i-2}) - q_i(ax_{i-1} + by_{i-1})$$

Ezt átrendezve:

$$r_i = a(x_{i-2} - q_i x_{i-1}) + b(y_{i-2} - q_i y_{i-1})$$

Kiterjesztett euklideszi algoritmus

Tudjuk tehát, hogy

$$r_i = a(x_{i-2} - q_i x_{i-1}) + b(y_{i-2} - q_i y_{i-1})$$

Legyen $x_1 = 1$, $y_1 = 0$, $x_2 = 0$ és $y_2 = 1$, és $i > 2$-re pedig:

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

Valamint

$$r_1 = a * x_1 + b * y_1 = a$$

$$r_2 = a * x_2 + b * y_2 = b$$

Kiterjesztett euklideszi algoritmus

Példa

Keressük x és y értékét, ha $a = 120$, $b = 23$

i	q_i	r_i	x_i	y_i	$r_i = ax_i + by_i$
1		120	1	0	$120 = 120 * \underline{1} + 23 * \underline{0}$
2		23	0	1	$23 = 120 * \underline{0} + 23 * \underline{1}$
3	5	5	1	-5	$5 = 120 * \underline{1} + 23 * (\underline{-5})$
4	4	3	-4	21	$3 = 120 * (\underline{-4}) + 23 * \underline{21}$
5	1	2	5	-26	$2 = 120 * \underline{5} + 23 * (\underline{-26})$
6	1	1	-9	47	$1 = 120 * (\underline{-9}) + 23 * \underline{47}$

Kiterjesztett euklideszi algoritmus

Példa

Tehát $1 = 120 * (-9) + 23 * 47$, azaz $x = -9$ és $y = 47$
A korábbi megállapítások alapján :

$$47 * 23 \equiv 1 \ (\ \mathrm{mod}\ 120\ )$$

$$120 * (-9) \equiv 26 * 38 \equiv 1 \ (\ \mathrm{mod}\ 47\ )$$

$$120 * (-9) \equiv 14 * 5 \equiv 1 \ (\ \mathrm{mod}\ 23\ )$$

Euler-függvény

Relatív prímek: Definíció

A és B egész számok relatív prímek, ha lnko(A, B) = 1

Euler-függvény: Definíció

$$\varphi(n) = |\{a : \text{ lnko(n, a)} = 1 \text{ és } a < n\}|$$

Azaz azon egészek száma, amelyek n-nél kisebbek, és n-el relatív prímek.

Példa

$\varphi(6) = ?$

lnko(6, 1) = 1 ✓ lnko(6, 2) = 2

lnko(6, 3) = 3 lnko(6, 4) = 2

lnko(6, 5) = 1 ✓

$\varphi(6) = 2$

Euler-függvény

Legyen $m = p_1^{e_1} * p_2^{e_2} \ldots * p_n^{e_n}$
$\varphi(m)$ Kiszámítása:

$$\varphi(m) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i - 1})$$

Példa

$\varphi(240) = ?$

- $m = 16 * 15 = 2^4 * 3^1 * 5^1 = p_1^{e_1} * p_2^{e_2} * p_3^{e_3}, n = 3$
- $\varphi(240) = (2^4 - 2^3) * (3^1 - 3^0) * (5^1 - 5^0) = 8 * 2 * 4 = 64$

A kiszámításához ismerni kell az m szám osztóit...

A kis Fermat-tétel

Legyen a egész, p pedig prím, ekkor igaz a következő állítás:

$$a^p \equiv a \ (\mod p)$$

másképp, ha $a \neq 0$:

$$a^{p-1} \equiv 1 \ (\mod p)$$

Példák

- Legyen a = 34, p = 37
- $34^{37} = 46208293553660808371261784090350221471870358813721042944$
- $46208293553660808371261784090350221471870358813721042944 \mod 37 = 34$
- Legyen a = 34, p = 7
- $34^7 = 52523350144$
- $52523350144 \mod 7 = 6 = 34 \mod 7$

RSA

RSA

- 1976-ban Ron Rivest, Adi Shamir és Len Adleman fejlesztették ki
- A módszer azt használja ki, hogy egy nagy szám prímtényezőkre bontására nem ismert gyors algoritmus

Lépései:

1. Kulcsválasztás
   1. Legyen p és q nagy prímszámok, $p \neq q$
   2. $N = p * q$
   3. $\varphi(N) = (p-1) * (q-1)$
   4. Legyen e olyan szám, hogy lnko(e, $\varphi(N)$) $= 1$
   5. Legyen d olyan szám, hogy $e * d \equiv 1$ (mod $\varphi(N)$)
   6. Nyilvános kulcs: (N, e), privát kulcs: d
2. Kódolás: $y = x^e \pmod{N}$
3. Dekódolás: $x = y^d \pmod{N}$

A támadónak a privát kulcs meghatározásához faktorizálni kell N-t

Példa

1. Legyen $p = 73$, $q = 151$
2. $N = p * q = 11023$
3. $\varphi(N) = (p - 1) * (q - 1) = 10800$
4. e legyen 11, mert lnko(10800, 11) = 1
5. d értéke 5891, mert $e * d \equiv 1 \pmod{10800}$
6. A nyilvános kulcs tehát: (11023, 11), a privát kulcs pedig 5891
7. Titkosítsuk az $x = 17$ üzenetet
8. $17^{11} \pmod{11023} = 1782$
9. Dekódolás: $1782^{5891} \pmod{11023} = 17$

Egy életszerűbb példa

1. $p = 14155210439216370810694657014 97$
2. $q = 171827468423467208701103275 3331$
3. $N = 243225397477198435382244489307218274890236227899958127843 6507$
4. $\varphi(N) = 243225397477198435382244489306904895317420596983150077998 1680$
5. $e = 7$
6. $d = 138985941415541963075568279603945654467097483990371473141810 3$
7. Titkosítsuk az $x = 7462836432632683268432732$ üzenetet:
8. $y = 576984127400318320790120869935160506898233978757302023931375$

A fenti példa bár látványos, még mindig túl kicsi számokat tartalmaz

Bizonyítás

Állítás: Bármely x egész számra igaz, hogy

$$(x^e)^d \equiv x \ (\mod N)$$

Tudjuk, hogy d az e-nek mod $\varphi(N)$ inverze, azaz

$$ed \equiv 1 \ (\mod \varphi(N))$$

Tehát: ed egy olyan szám, amit ha elosztok $\varphi(N)$-el, akkor 1-et kapok maradékul, az egész osztás eredménye legyen v:

$$ed = v\varphi(N) + 1$$

Tehát a következőt kell belátnunk:

$$x^{v\varphi(N)+1} \equiv x \ (\mod N) \tag{1}$$

Bizonyítás

Bizonyítsuk be, hogy tetszőleges x és s értékekre igaz az alábbi állítás, ha u prím:

$$x^{s(u-1)+1} \equiv x \ (\ \mathrm{mod\ u}\ ) \tag{2}$$

Nézzük azt az esetet, mikor x-et nem osztja u, ekkor a kis Fermat-tétel szerint:

$$x^{u-1} \equiv 1 \ (\ \mathrm{mod\ u}\ ) \rightarrow (x^{u-1})^s \equiv 1 \ (\ \mathrm{mod\ u}\ ) \rightarrow x(x^{u-1})^s \equiv x \ (\ \mathrm{mod\ u}\ )$$

Tehát (2) ebben az esetben igaz.
A másik eset az, mikor x-et osztja u. Ekkor A maradék 0, tehát:

$$x = 0 = x^{s(u-1)+1} \equiv x \ (\ \mathrm{mod\ u}\ )$$

Azaz (2) ekkor is igaz lesz.

Bizonyítás

- Ezt akarjuk belátni: $x^{v\varphi(N)+1} \equiv x \ (\mod N)$
- Tudjuk, hogy $x^{s(u-1)+1} \equiv x \ (\mod u)$ ha u prím
- Valamint $ed = v\varphi(N) + 1 = v(p-1)(q-1) + 1$
- Legyen $s_1 = v(p-1)$, valamint $s_2 = v(q-1)$

Ezek alapján igazak a következők:

$$x^{s_1(q-1)+1} \equiv x \ (\mod q)$$

$$x^{s_2(p-1)+1} \equiv x \ (\mod p)$$

Vagyis q osztja $x^{s_1(q-1)+1} - x$-et és p osztja $x^{s_2(p-1)+1} - x$-et

$$\Downarrow$$

$pq = N$ osztja $x^{v\varphi(N)+1} - x$-et $\rightarrow x^{v\varphi(N)+1} \equiv x \ (\mod N)$ $\square$

GNU MP

GNU MP

- Nagy számok használata
- Szabadon elérhető
- gmplib.org
- Szükséges header file: gmp.h
- C kód fordítása: gcc myprogram.c -lgmp
- C++ kód fordítása: g++ mycxxprog.cpp -lgmpxx -lgmp

Telepítés linux alatt:

- Csomagoljuk ki a honlapról letölthető fájlt, majd lépjünk be a könyvtárába
- ./configure –enable-cxx –disable-shared
- make
- sudo make install

GNU MP

Példakód:

```
#include <iostream>
#include <gmp.h>
int main() {
    mpz_t a;
    mpz_init_set_str(a, "12345678910323232", 10);
    std::cout << a << std::endl;
    mpz_clear(a);
}
```

Egész számok GNU MP-ben

- mpz_t : egészek tárolására alkalmas típus
- használat előtt inicializálni kell:
- mpz_init(mpz_t a): 0-ra inicializálja a változót
- mpz_init_set_str(mpz_t rop, char * str, int base): A rop változót inicializálja az str-ben megadott szám alapján, mely a base számrendszerben adott

Értékadás:

- mpz_set(mpz_t rop, mpz_t op) : rop = op
- mpz_set_ui(mpz_t rop, unsigned long int op) : rop = op

Összeadás, kivonás, szorzás:

- mpz_add(mpz_rop, mpz_t op1, mpz_t op2) : rop = op1 + op2
- Hasonlóan mpz_sub, mpz_mul

Számelméleti függvények

- Modulo hatványozás
- mpz_powm(mpz_t rop, mpz_t base, mpz_t exp, mpz_t mod): rop = base^{exp} (modulo mod)
- Prímszám keresés:
- mpz_nextprime(mpz_t rop, mpz_t op): rop = op utáni következő prímszám
- Legnagyobb közös osztó:
- mpz_gcd(mpz_t rop, mpz_t op1, mpz_t op2): rop = lnko(op1, op2)
- Modulo inverz:
- mpz_invert(mpz_t rop, mpz_t op1, mpz_t op2): rop = op1 inverze modulo op2

Véletlen számok

- gmp_randstate_t randstate: random generátor állapotát tárolja
- gmp_randinit_default (gmp_randstate_t state): inicializálja a randomgenerátor állapotát
- gmp_randseed (gmp_randstate_t state, mpz_t seed): a randomgenerátor belső változóit állítja be
- mpz_urandomb (mpz_t rop, gmp_randstate_t state, mp_bitcnt_t n): rop = véletlen szám 0 és 2^{n-1} között