Kriptografija i kriptoanaliza

doc. dr. sc. Ante Đerek i prof. dr. sc. Marin Golub

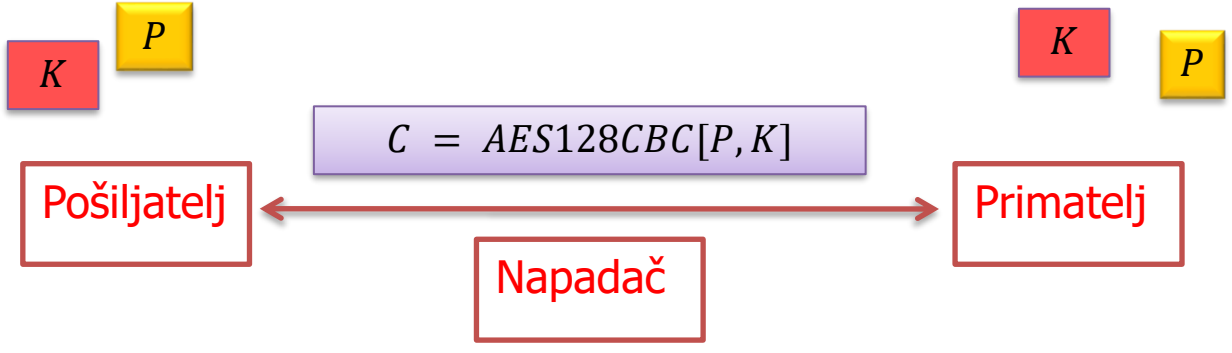
6.
Asimetrični kriptosustavi

ili sustavi s javnim ključem

Asimetrični kriptosustavi

- Sustavi kriptiranja javnim ključem
- Metode razmjene ključeva
- Digitalni potpisi

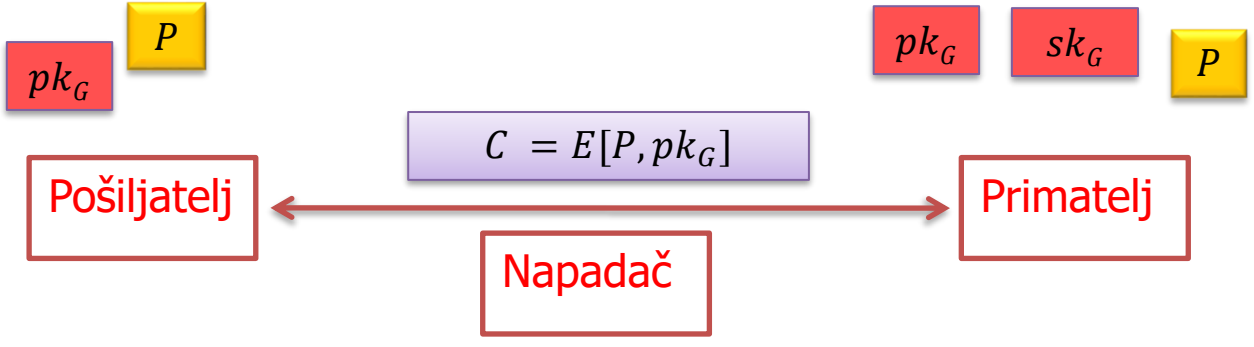
Simetrična enkripcija



- Distribucija ključeva?
- Što ako je primatelj na drugom kontinentu?
- Kako uspostaviti zajednički ključ s poslužiteljem na internetu?

Enkripcija javnim ključem

- Nova ideja: Primatelj ima dva ključa
  - Javni ključ pk_G: Javno poznat (npr. telefonski imenik)
  - Privatni ključ sk_G: Poznat samo Primatelju
  - Jasni tekst se kriptira s javnim ključem
  - Skriveni tekst se dekriptira s privatnim ključem



Primjena – PGP/GPG

```
$ gpg --list-keys Gledec
pub   1024D/800D81AC 1999-12-03
uid                  Gordan Gledec <gordan.gledec@tel.fer.hr>
uid                  Gordan Gledec <gordan@tel.fer.hr>
uid                  Gordan Gledec <gordan.gledec@fer.hr>
uid                  Gordan Gledec <gordan@kaktus.tel.fer.hr>
uid                  Gordan Gledec <gordan.gledec@zg.hinet.hr>
sub   1024g/7EBABF31 1999-12-03

$ cat poruka.txt
Napadamo u zoru

$ gpg --armor --encrypt --recipient 0x800D81AC  --output poruka.pgp poruka.txt
gpg: 7EBABF31: There is no assurance this key belongs to the named user

pub  1024g/7EBABF31 1999-12-03 Gordan Gledec <gordan.gledec@tel.fer.hr>
 Primary key fingerprint: 8294 3615 5220 2F8A 9A70  3F7A 8B1B 4606 800D 81AC
      Subkey fingerprint: A240 91E6 80BB BFD0 920E  B7AE CF09 55B2 7EBA BF31

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
$ cat poruka.pgp
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1

hQEOA88JVbJ+ur8xEAP+PhWVbvpYFuAVLoCBmkid8hPXUTInN0oYSafZ0rSFRQzo
JS+/qHMu24C8QzSQkyyV/9wLWQeyak6ApzCZozov3TlpFk9qlOHqwfvY+F70T2Uz
3jVJKI0c9Y0k8AiFLYgqNogZZ84J45ra00KHSse7vhoJto3j1Rm+1qsTFChjGx0D
/ijAFyg+KdKKfjAUKc0Sm8GH3XHQVVrAjCz5Q4KfnTD90QgOOPgeAOMsqCri+gc9
5gx+dH3Ko9S9pjWPYNUAkD74evfrOL5cGUgI50wV7Xhf4eWWjXmewylaZxECBjkv
NWDWFQRLm42oZ9wk7KkqDfzjALBV/BjDR7RYzH3m7XbQyTAgyQHuzsEP4kI6FjCU
P8TKqXci1Ug8eAnSCMkONdqMNYoLrSOzBwwPI4IHTj3RI7o=
=HlYL
-----END PGP MESSAGE-----
$
```

Search results for '0x8b1b4606800d81ac'

Type bits/keyID cr. time exp time key expir

pub 1024D/800D81AC 1999-12-03
      Fingerprint=8294 3615 5220 2F8A 9A70 3F7A 8B1B 4606 800D 81AC

uid Gordan Gledec <gordan@tel.fer.hr>
sig sig 800D81AC 1999-12-03 _____ _____ [selfsig]

uid Gordan Gledec <gordan.gledec@fer.hr>
sig sig 800D81AC 2001-01-31 _____ _____ [selfsig]

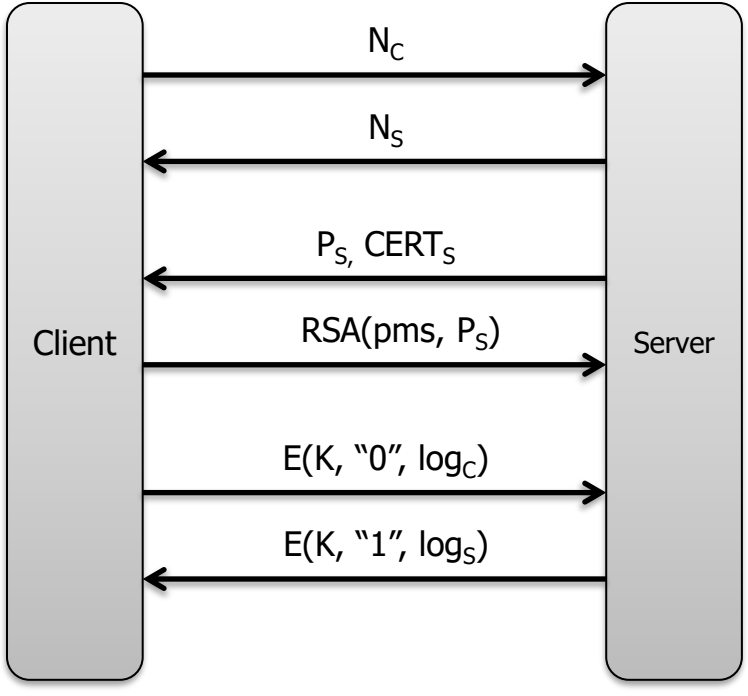
uid Gordan Gledec <gordan.gledec@tel.fer.hr>
sig sig 800D81AC 2001-01-31 _____ _____ [selfsig]

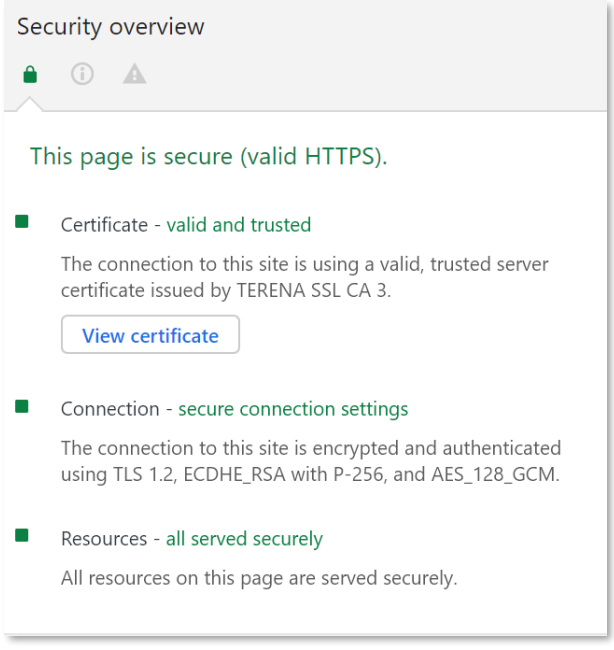
uid Gordan Gledec <gordan@kaktus.tel.fer.hr>
sig sig 800D81AC 2001-01-31 _____ _____ [selfsig]

uid Gordan Gledec <gordan.gledec@zg.hinet.hr>
sig sig 800D81AC 2001-01-31 _____ _____ [selfsig]

sub 1024g/7EBABF31 1999-12-03
sig sbind 800D81AC 1999-12-03 _____ _____ []

Primjena – TLS protocol (jedna od puno varijanti uspostave ključeva)





Povijest asimetrične kriptografije

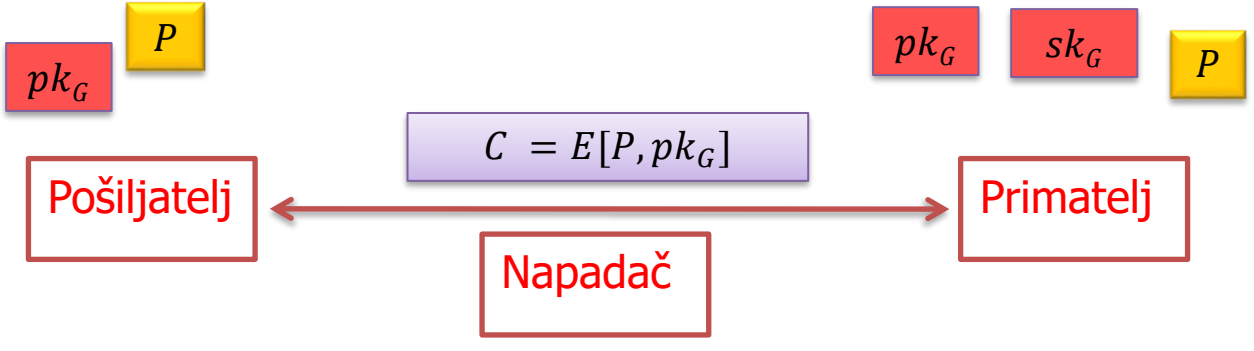
- 1973 – Cocks, „A Note on Non-Secret Encryption”
- 1975 – Merkle, „Secure Communications over Insecure Channels”
- 1976 – Diffie, Hellman, „New Directions in Cryptography”
- 1977 – Rivest, Shamir, Adelman, „A method for obtaining digital signatures and public key cryptosystems”
- 1985 – ElGamal, „A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”
- 2020 – NIST Post-Quantum Cryptography Standardization Process

Sustav kriptiranja javnim ključem

- Trojka *efikasnih* algoritama G, E i D
  - G – algoritam koji generira par ključeva pk, sk
  - $E(m, pk)$ – algoritam enkripcije
  - $D(c, sk)$ – algoritam dekripcije
- Za svaki par ključeva (pk, sk) generiranih algoritmom G i za svaki jasni tekst p vrijedi $D(E(p, pk), sk) = p$

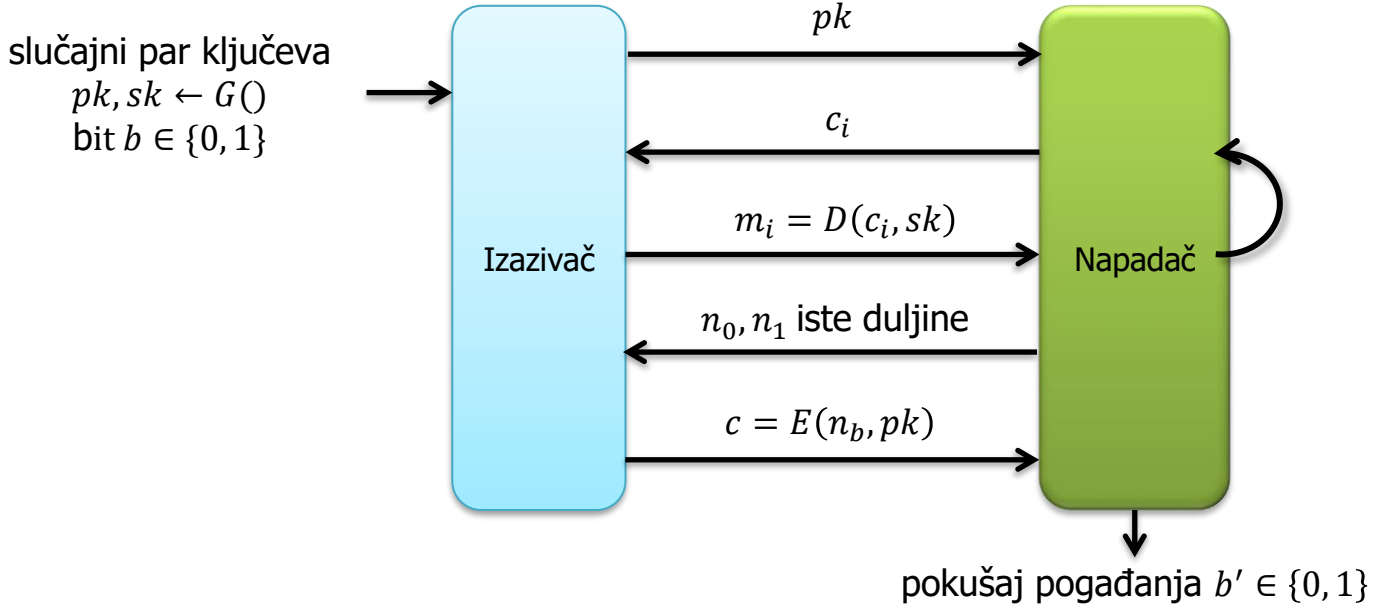
Sustav kriptiranja javnim ključem – sigurnost

- SKJK je *siguran* ako je teško na temelju kriptiranog teksta odrediti bilo što o jasnom tekstu ...
- ... čak i ako napadač ima na raspolaganju:
  - Javni ključ kojim je jasni tekst kriptiran
    - (chosen-plaintext attack).
  - Mogućnost da dobije $p = D(c, sk)$ za proizvoljni c
    - (chosen-ciphertext attack)



Primjer definicije sigurnosti SKJK

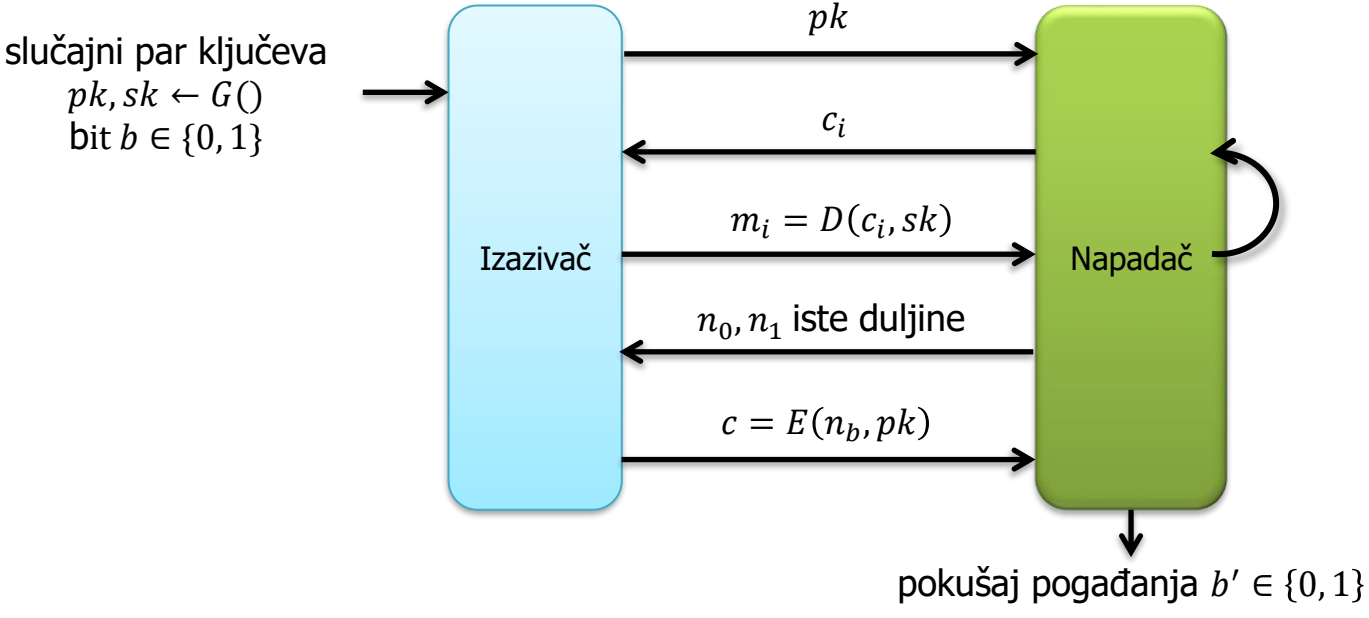
- Semantička sigurnost od napada odabranim skrivenim tekstom (semantic security under chosen-ciphertext attack): Svaki efikasan algoritam ima zanemarivu prednost u sljedećoj igri.



$$\mathrm{Adv}_{SS-CCA1}(A) = |P(W_0) - P(W_1)|$$

Zadatak: Sigurnost determinističkog SKJK

- Može li deterministički SKJK biti semantički siguran od napada odabranim skrivenim tekstom? Od napada odabranim jasnim tekstom?



Primjeri sustava kriptiranja javnim ključem

- Merkleove slagalice (1974)
  - građene pomoću simetrične šifre, nepraktično
- RSA (1978)
  - teorija brojeva, sigurnost povezana s problemom faktorizacije
- McEliece (1978)
  - teorija kodiranja, sigurnost povezana s problemom dekodiranja općenitog linearnog koda
- ElGamal (1985)
  - teorija brojeva ili eliptičke krivulje, sigurnost povezana s problemom diskretnog logaritma

Ne znamo izgraditi dobar sustav kriptiranja javnim ključem pomoću supstitucija, permutacija, operacije XOR ¯_(ツ)_/¯

Teorija brojeva – notacija

- N – prirodni broj
- p, q – prosti brojevi
- $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ – *prsten* u kojemu se zbraja, oduzima i množi modulo N
- Pišemo $a = b$ u $\mathbb{Z}_N$ umjesto $a \equiv b \pmod{N}$

Aritmetika u $\mathbb{Z}_N$

$$9 + 8 = 5 \text{ u } \mathbb{Z}_{12}$$
$$5 \cdot 7 = 11 \text{ u } \mathbb{Z}_{12}$$
$$7 - 9 = 10 \text{ u } \mathbb{Z}_{12}$$

Propozicija: Za aritmetiku u $\mathbb{Z}_N$ vrijede uobičajena svojstva komutativnosti, asocijativnosti i distributivnost (za sada nema dijeljenja u $\mathbb{Z}_N$).

Najveći zajednički djelitelj

Propozicija: Neka su x i y cijeli brojevi i neka je k njihov *najveći zajednički djelitelj*, $k = nzd(x,\ y)$. Postoje cijeli brojevi a i b tako da vrijedi $ax + by = k$.

- Brojevi a, b i k se mogu efikasno odrediti *proširenim Euklidovim algoritmom*

Dijeljenje u $\mathbb{Z}_N$

- *Inverz* elementa $x \in \mathbb{Z}_N$ je element $y \in \mathbb{Z}_N$ takav da vrijedi $x \cdot y = 1$ u $\mathbb{Z}_N$.
- Inverz od x označavamo s x^{-1} (ako postoji)

Inverz od 2 u $\mathbb{Z}_{17}$? 9

Inverz od 4 u $\mathbb{Z}_{10}$? Ne postoji.

Kriterij invertibilnosti

Propozicija: Broj x ima inverz u $\mathbb{Z}_N$ ako i samo ako je $nzd(x, N) = 1$.

- Ako je $nzd(x, N) = 1$ onda vrijedi da postoje a i b takvi da je $ax + bN = 1$ pa je a inverz od x u $\mathbb{Z}_N$.
- Obratno ako je $xy = 1 + kN$ onda x i N moraju biti relativno prosti zato što svaki broj koji dijeli x i N mora dijeliti i broj 1.

Grupa $\mathbb{Z}_N^*$

- $\mathbb{Z}_N^*$ je skup svih invertibilnih elementa $x \in \mathbb{Z}_N$
- Drugim riječima $\mathbb{Z}_N^* = \{x \in \mathbb{Z}_N, nzd(x, N) = 1\}$
  - Svaki element u $\mathbb{Z}_N^*$ ima inverz koji je također u $\mathbb{Z}_N^*$
  - Ako su x i y u $\mathbb{Z}_N^*$ onda je i xy u $\mathbb{Z}_N^*$
  - Stoga je $\mathbb{Z}_N^*$ *grupa* u odnosu na operaciju množenja

$$\mathbb{Z}_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$$

Ako je p prost $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$

Eulerova funkcija

- *Eulerova funkcija* $\varphi(N) = |\mathbb{Z}_N^*|$ je broj prirodnih brojeva manjih od N i relativno prostih s N.

$$\varphi(15) = 8$$

Ako je p prost onda $\varphi(p) = p - 1$

Eulerova funkcija

Propozicija: Ako su p i q različiti prosti brojevi onda je $\varphi(pq) = (p-1)(q-1)$.

- Razmatram sve brojeve $1, 2, \ldots, pq-1$
- Trebam izbaciti sve višekratnike od p
  - To su $p, 2p, 3p, \ldots, (q-1)p$
- Trebam izbaciti sve višekratnike od q
  - To su $q, 2q, 3q, \ldots, (p-1)q$
- Zašto nisam niti jedan broj izbacio dvaput?
- $\varphi(pq) = pq - 1 - (p-1) - (q-1) = (p-1)(q-1)$

Eulerov teorem

Teorem (Euler): Za svaki prirodni broj N i za svaki $a \in \mathbb{Z}_N^*$ vrijedi $a^{\varphi(N)} = 1$ u $\mathbb{Z}_N$.

Teorem (Fermat): Za svaki prosti broj p i za svaki $a \in \mathbb{Z}_p^*$ vrijedi $a^{p-1} = 1$ u $\mathbb{Z}_p$.

Računanje s velikim brojevima

- Radimo s brojevima veličine 1024-4096 bitova (300-1200 dekadskih znamenki)
- Broj veličine n bitova najčešće pohranjujemo u $\frac{n}{32}$ 32-bitna bloka

```
typedef struct bignum_st BIGNUM;

struct bignum_st
       {
       BN_ULONG *d;    /* Pointer to an array of 'BN_BITS2' bit chunks. */
       int top;        /* Index of last used d +1. */
       /* The next are internal book keeping for bn_expand. */
       int dmax;       /* Size of the d array. */
       int neg;        /* one if the number is negative */
       int flags;
       };
```

The integer value is stored in **d**, a malloc()ed array of words (**BN_ULONG**), least significant word first. A **BN_ULONG** can be either 16, 32 or 64 bits in size, depending on the 'number of bits' (**BITS2**) specified in `openssl/bn.h`.

Izvor: openssl dokumentacija

Aritmetika

- Zbrajanje/oduzimanje?
  - *Školski* algoritam: $O(n)$
- Množenje?
  - *Školski* algoritam: $O(n^2)$
  - Karatsuba: $O\left(n^{\log_2 3}\right) \approx O(n^{1.58})$
  - Asimptotski bolji algoritmi?
- Dijeljenje s ostatkom?
  - *Školski* algoritam: $O(n^2)$
  - Optimizacijski trikovi – estimacija kvocijenta, normalizacija
  - Asimptotski bolji algoritmi?

Modularna aritmetika

- Zbrajanje/oduzimanje modulo N?
  - Školski algoritam: $O(n)$
- Množenje modulo N?
  - Pomnoži pa izračunaj ostatak: $O(M) + O(D)$
  - Montgomery: $O(n^2)$
- Eksponenciranje modulo N?
  - Računamo b^a mod N, gdje su a, b, i N n-bitni brojevi
  - For petlja: $O(a\ M)$
  - Uzastopno kvadriranje: $O(n\ M)$

Uzastopno kvadriranje

$$b^{2k} = \left(b^k\right)^2$$
$$b^{2k+1} = \left(b^k\right)^2 b$$

Neka je $a_m, a_{m-1}, \ldots, a_1, a_0$ binarni prikaz od a.

```
d = 1;
i = m;
dok je (i >= 0) {
    d = (d * d) mod n;
    ako je (a[i] == 1) {
        d = (d*b) mod n;
    }
    i --;
}
```

Uzastopno kvadriranje – primjer

Primjer 11.4.

Neka su:

$a = 635 \Rightarrow a = 1001111011_{(2)}$

$n = 734$

$b = 5$

$5^{635} \bmod 734 = ?$

```
d = 1;
i = m;
dok je (i >= 0) {
    d = (d * d) mod n;
    ako je (a[i] == 1) {
        d = (d * b) mod n;
    }
    i --;
}
```

Postupak izračunavanja $b^a \bmod n$:

i	9	8	7	6	5	4	3	2	1	0
a[i]	1	0	0	1	1	1	1	0	1	1
d	5	25	625	685	261	29	535	699	253	21

Kriptosustav “obični RSA”

- *Textbook RSA / Schoolbook RSA*
- Moramo definirati
  - Algoritam G
  - Algoritam E
  - Algoritam D

RSA – generiranje ključeva

Algoritam G:

1. Odaberem velike slučajne proste brojeve p i q
2. Izračunam $N = p \cdot q$
3. Izračunam $\varphi(N) = (p-1)(q-1)$
4. Odaberem proizvoljni $e \in \mathbb{Z}^*_{\varphi(N)}$ (u praksi $e = 65537$)
5. Izračunam $d = e^{-1}$ u $\mathbb{Z}^*_{\varphi(N)}$
6. Javni ključ: $pk = (e, N)$
7. Privatni ključ: $sk = (d, N)$

RSA – generiranje ključeva

Algoritam G:

1. Odaberem velike slučajne proste brojeve p i q
2. Izračunam $N = p \cdot q$
3. Izračunam $\varphi(N) = (p-1)(q-1)$
4. Odaberem proizvoljni $e \in \mathbb{Z}^*_{\varphi(N)}$ (u praksi $e = 65537$)
5. Izračunam $d = e^{-1}$ u $\mathbb{Z}^*_{\varphi(N)}$
6. Javni ključ: $pk = (e, N)$
7. Privatni ključ: $sk = (d, N)$

Ako je moguće N efikasno rastaviti na faktore onda je RSA nesiguran
Ako je moguće efikasno izračunati $\varphi(N)$ onda je RSA nesiguran

Obični RSA – enkripcija i dekripcija

Algoritam E:

- $E\big(m,(e,N)\big) = m^e \text{ u } \mathbb{Z}_N$

Algoritam D:

- $D\big(c,(d,N)\big) = c^d \text{ u } \mathbb{Z}_N$

e zovemo *javni eksponent*
d zovemo *privatni eksponent*
N zovemo *modul*
Jasni i skriveni tekst su brojevi u $\mathbb{Z}_N$

RSA – Korektnost

Algoritam G:

1. Veliki slučajni prosti brojeve p i q
2. Izračunam $N = p \cdot q$
3. Izračunam $\varphi(N) = (p-1)(q-1)$
4. Odaberem $e \in \mathbb{Z}^*_{\varphi(N)}$
5. Izračunam $d = e^{-1}$ u $\mathbb{Z}^*_{\varphi(N)}$
6. Javni ključ: $pk = (e, N)$
7. Privatni ključ: $sk = (d, N)$

Algoritam E:

- $E\big(m, (e, N)\big) = m^e$ u $\mathbb{Z}_N$

Algoritam D:

- $D\big(c, (d, N)\big) = c^d$ u $\mathbb{Z}_N$

$$D\big(E(m,(e,N)),(d,N)\big) = D\big(m^e,(d,N)\big) = (m^e)^d = m^{ed}$$
$$= m^{1+k\varphi(N)} = m \cdot (m^{\varphi(N)})^k = m \cdot (1)^k = m \text{ u } \mathbb{Z}_N$$

RSA – poruke

- Jasni i skriveni tekst moraju biti u $\mathbb{Z}_N^*$
- Zašto ovo nije problem?
  - Možete li za veliki RSA modul N pronaći neki broj između 1 i $N-1$ koji nije u $\mathbb{Z}_N^*$?

RSA – Implementacija

Algoritam G:

1. Veliki slučajni prosti brojeve p i q
2. Izračunam $N = p \cdot q$
3. Izračunam $\varphi(N) = (p-1)(q-1)$
4. Odaberem $e \in \mathbb{Z}^*_{\varphi(N)}$
5. Izračunam $d = e^{-1}$ u $\mathbb{Z}^*_{\varphi(N)}$
6. Javni ključ: $pk = (e, N)$
7. Privatni ključ: $sk = (d, N)$

Algoritam E:

- $E\big(m, (e, N)\big) = m^e$ u $\mathbb{Z}_N$

Algoritam D:

- $D\big(c, (d, N)\big) = c^d$ u $\mathbb{Z}_N$

- Množenje, zbrajanje, inverz, modularno eksponenciranje

RSA – Implementacija

Algoritam G:

1. Veliki slučajni prosti brojeve p i q
2. Izračunam $N = p \cdot q$
3. Izračunam $\varphi(N) = (p-1)(q-1)$
4. Odaberem $e \in \mathbb{Z}^*_{\varphi(N)}$
5. Izračunam $d = e^{-1}$ u $\mathbb{Z}^*_{\varphi(N)}$
6. Javni ključ: $pk = (e, N)$
7. Privatni ključ: $sk = (d, N)$

Algoritam E:

- $E\big(m, (e, N)\big) = m^e$ u $\mathbb{Z}_N$

Algoritam D:

- $D\big(c, (d, N)\big) = c^d$ u $\mathbb{Z}_N$

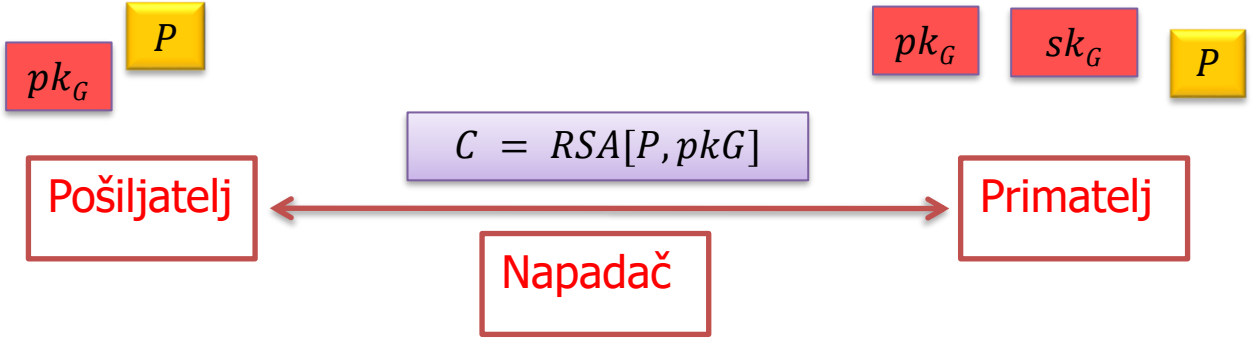
- Složenost enkripcije i dekripcije? Općenito $O(n^3)$
- $e = 63357$?

Generiranje prostih brojeva

- Algoritam
  - Izaberi slučajni broj zadane veličine
  - Provjeri je li izabrani broj prost
- Prosti brojevi su dovoljno *gusti*
- Postoje efikasni algoritmi koji određuju je li broj prost ili složen (npr. Miller-Rabin)

RSA – sigurnost

- Obični RSA *nije* siguran sustav kriptiranja javnim ključem ☹



Zadatak: Obični RSA 1

- Kriptiramo glasove na izborima
  - sudjeluju dva kandidata označena s 1 i 2
  - izborno povjerenstvo objavi svoj javni ključ pk.
  - glasač A izračuna $c_A = E(g_A, pk)$ gdje je $g_A \in \{1, 2\}$
  - glasač A šalje c_A izbornom povjerenstvu

- $E(1, pk) = 1$
- $E(2, pk) \neq 1$
- Napadač može zaključiti za koga je glasač glasao!

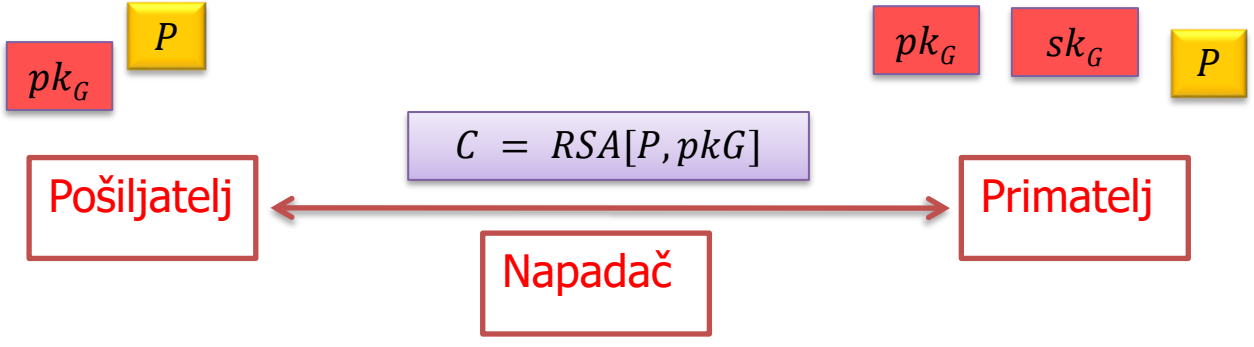
Zadatak: Obični RSA 2

- Kriptiramo datoteku
  - Datoteka se sastoji se od n bajtova $b_1, b_2, \ldots, b_n$
  - kriptiramo svaki bajt zasebno $c_k = E(b_k, pk)$
  - šaljemo $c_1, c_2, \ldots, c_n$ Wi-Fi mrežom

- Napadač može za svaki mogući bajt $b = 0, 1, \ldots, 255$ izračunati $c = E(b, pk)$
- Kada vidi $c_1, c_2, \ldots, c_n$ lagano nalazi $b_1, b_2, \ldots, b_n$

Sustav kriptiranja javnim ključem – sigurnost

- Ako je algoritam enkripcije deterministički onda sustav kriptiranja javnim ključem nikako ne može biti siguran



Zadatak: Obični RSA 3

- Šaljemo 128-bitni AES ključ K koristeći RSA
  - neka je $e = 3$
  - šaljemo $c = E(K, pk)$

- K^3 ima oko manje od 400 bitova
- Prilikom enkripcije se ne dogodi redukcija modulo N
- Napadač može izračunati $K = \sqrt[3]{c}$

Zadatak: Obični RSA 4

- Šaljemo 64-bitni DES ključ K koristeći RSA
  - neka je $e = 65537$
  - šaljemo $c = RSA(K, pk)$

- Ponekad će se slučajno dogoditi da je $K = K_1 \cdot K_2$ gdje su K_1 i K_2 32-bitni brojevi
- $c = K^e = K_1^e \cdot K_2^e$ u $\mathbb{Z}_N$
- $c \cdot (K_1^e)^{-1} = K_2^e$ u $\mathbb{Z}_N$
- *Meet-in-the-middle* algoritam nalazi K_1 i K_2 u 2^{32} koraka