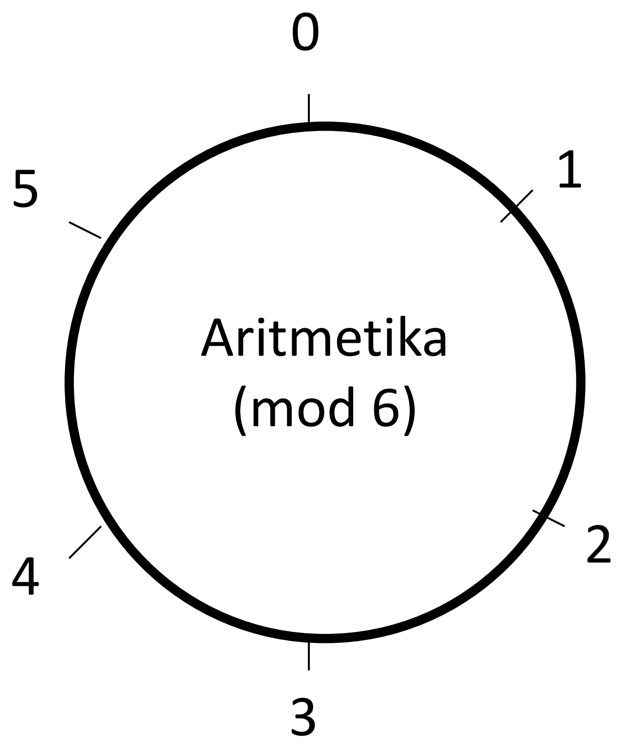
Šifarski sistemi sa javnim ključem

Sadržaj

- Matematičke osnove
- Problem razmene ključa
- Difi-Helmanov algoritam za razmenu ključeva
- Kriptografija sa javnim ključevima
- RSA algoritam
- Tajnost i neporecivost
- Infrastruktura javnih ključeva

Modularna aritmetika

- Za cele pozitivne brojeve x i n, x po modulu n predstavlja ostatak deljenja $x \div n$.
- $x \pmod{n} = r \;\Rightarrow\; x = kn + r$, gde je k neki ceo broj.
- Primeri:
  - 7 (mod 6) = 1 ili 7 = 1 (mod 6)
  - 33 (mod 5) = 3 ili 33 = 3 (mod 5)
  - 33 (mod 6) = 3 ili 33 = 3 (mod 6)
  - 51 (mod 17) = 0 ili 51 = 0 (mod 17)
  - 17 (mod 6) = 5 ili 17 = 5 (mod 6)



Modularna aritmetika

- Zapis i neka svojstva:
  - 7 (mod 6) = 1
  - 7 (mod 6) = 13 (mod 6) = 1
- Primeri sabiranja:
  - (3 + 5) (mod 6) = 2
  - (2 + 4) (mod 6) = 0
  - (3 + 3) (mod 6) = 0
  - (7 + 12) (mod 6) = 19 (mod 6) = 1
  - (7 + 12) (mod 6) = (1 + 0) (mod 6) = 1

Modularna aritmetika

- Primeri množenja:
  - $(3 \cdot 4) \pmod 6 = 0$
  - $(2 \cdot 4) \pmod 6 = 2$
  - $(5 \cdot 5) \pmod 6 = 1$
  - $(7 \cdot 4) \pmod 6 = 28 \pmod 6 = 4$
  - $(7 \cdot 4) \pmod 6 = (1 \cdot 4) \pmod 6 = 4$
- Stepenovanje:
  - $a \pmod N = b \Rightarrow a^k \pmod N = b^k$

- **Adtitivna inverzija** x po modulu n (označava se sa $-x$) je broj koji treba sabrati sa x da bi moduo tog zbira bio 0.
  - $-2 \pmod 6 = 4$ jer je $(2+4) \pmod 6 = 0$
  - Nije negativan broj, samo oznaka.
- **Multiplikativna inverzija** x po modulu n (označava se sa x^{-1}) je broj koji treba pomnožiti sa x da bi moduo tog proizvoda bio 1.
  - $3^{-1} \pmod 7 = 5$ jer je $(3 \cdot 5) \pmod 7 = 1$
  - Nije broj manji od 1, samo oznaka.

- Primeri:
  - Koliko je -3 (mod 6)?
    - Odgovor: 3.
  - Koliko je -1 (mod 6)?
    - Odgovor: 5.
  - Koliko je 5^{-1} (mod 6)?
    - Odgovor: 5.
  - Koliko je 2^{-1} (mod 6)?
    - Odgovor: nema rešenja!
  - Multiplikativna inverzija ne postoji za svaki broj!

Prosti brojevi

- **Prost broj** je ceo broj koji ima samo dva delioca: 1 i samog sebe.
  - Po dogovoru, smatra se da 1 nije prost broj.
- Primer prostih brojeva: 2, 3, 5, 7, 11, 13, 17, 19, 23, ...
- Nema pravila na osnovu kojeg su raspoređeni prosti brojevi u skupu celih brojeva.
- Prostih brojeva ima beskonačno mnogo.
- Za broj koji nije prost kaže se da je složen.

Uzajamno prosti brojevi

- Neka su x i y dva cela broja.
- **Najveći zajednički delilac** (*gcd*) brojeva x i y je najveći broj d takav kojim se mogu prodeliti x i y.
  - gcd(3, 16) = 1
  - gcd(28, 8) = 4.
- Brojevi x i y su **uzajamno prosti** ako je gcd(x, y) = 1.
  - Dva prosta broja su istovremeno i uzajamno prosta.
- x^{-1} (mod y) postoji samo ako su x i y uzajamno prosti.
- x^{-1} (mod y) se lako nalazi (ako postoji) korišćenjem Euklidovog algoritma.

Uzajamno prosti brojevi

```
int gcd(int a, int b) {
  if(a==0)return b;
  while(b!=0) {
    if(a > b) a=a-b;
    else b=b-a;
  }
  return a;
}
```

Primer: gcd(6,15) = 3

Kineska teorema o ostatku

- Ako su p i q prosti brojevi i ako je:
  - $m \pmod{p} = a$
  - $m \pmod{q} = a$
- onda je:
  - $m \pmod{pq} = a$

Ojlerova funkcija φ

- Leonard Ojler (1707-1783), švajcarski matematičar.
- $\varphi(n)$ je broj pozitivnih celih brojeva manjih od n, koji su uzajamno prosti u odnosu na n.
- Primeri:
  - $\varphi(4) = 2$ jer je 4 uzajamno prost sa 1 i 3.
  - $\varphi(5) = 4$ jer je 5 uzajamno prost sa 1, 2, 3 i 4.
  - $\varphi(12) = 4$ jer je 12 uzajamno prost sa 1, 5, 7 i 11.
  - ...

Ojlerova funkcija φ

- Ako je p prost broj, onda je:
  - $\varphi(p) = p - 1$
- Ako je p prost broj, onda je:
  - $\varphi(p^n) = p^n - p^{n-1}$
- Ako su m i n uzajamno prosti, onda je:
  - $\varphi(mn) = \varphi(m)\,\varphi(n)$
- Ako su p i q prosti brojevi, onda je:
  - $\varphi(pq) = \varphi(p)\,\varphi(q) = (p-1)(q-1)$
- Za svaki pozitivan broj n i svako x koje je uzajamno prosto sa n važi:
  - $x^{\varphi(n)} = 1 \pmod{n}$

Faktorizacija broja

- Osnovna teorema aritmetike:
- Svaki pozitivan ceo broj N >1 može da se predstavi kao proizvod jednog ili više prostih brojeva u sledećem obliku:

$$N = p_1^{e_1} \times p_2^{e_2} \times \ldots \times p_n^{e_n}$$

- Ovaj postupak se naziva faktorizacija broja N.
- Faktorizacija broja je jedinstvena.
- Primeri:
  - $6647 = 17^2 \cdot 23$
  - $90 = 2 \cdot 3^2 \cdot 5$
- Problem faktorizacije broja je u opštem slučaju težak problem.
- Jedan od načina određivanja *gcd* dva broja svodi se na faktorizaciju oba broja.
- Donošenje odluke da li je broj prost ili nije, lakši je problem od faktorizacije.

Problem diskretnog logaritma

- Za date cele brojeve *b*, *c* i *n* problem je kako naći *x* takvo da je: $b^{x} = c \pmod{n}$.
- Ovaj problem je vremenski (procesno) zahtevan i smatra se teškim.
- Nije poznato da postoji efikasan algoritam za rešavanje preoblema diskretnog logaritma!

Kriptosistemi sa javnim ključem

- Kriptosistemi sa javnim ključem koriste **dva ključa**:
  - **Javni** (za šifrovanje)
  - **Privatni** (za dešifrovanje).
- Tri načina upotrebe:
  - Razmena simetričnog ključa.
  - Šifrovanje/dešifrovanje (poverljivost).
  - Digitalni potpis (autentifikacija, a ukoliko se koriste heš funkcije i integritet).

Problem razmene ključa

- Banka treba da obavi šifrovani prenos sa klijentom, kako dostaviti ključ?
  - Najbezbednije: lično; vreme, ljudi, ...
  - Manje bezbedno: kurirskom službom.
    - Da li je to nezavisna organizacija?
    - Da li je to slaba karika?
- Dostava ključa vojnim jedinicama u ratnim uslovima?
- Dostava ključa (nuklearnim) podmornicama koje se nalaze (skrivene) na 1000-de km od baze?
- **Država** raspolaže novcem i resursima, može da izađe na kraj sa ovakvim problemima.
- Za **civilni sektor** je ovo bio gotovo nerešiv problem.

Problem razmene ključa

- Uprkos opšte prihvaćenom mišljenju da je ovaj problem nerešiv, jedna grupa entuzijasta je krajem 70-ih ponudila rešenje.
- Istraživanja u ovom pravcu su dovela do razvoja kripto sistema sa javnim ključem.
- **Vitfild Difi** (Whitfield Diffie)
  - Rođen 1944. godine, Njujork.
  - 1965. diplomirao na MIT.
  - http://en.wikipedia.org/wiki/Whitfield_Diffie
- **Martin Helman** (Martin Hellman)
  - Rođen 1945. godine, Bronks.
  - 1967. doktorirao na Stanford univrzitetu.
  - http://en.wikipedia.org/wiki/Martin_Hellman
- **Ralf Merkl** (Ralph Merkle)
  - Kasnije se pridružio grupi

Jednosmerne funkcije

- Difi i Helman su tražili matematičke funkcije za koje **redosled šifrovanja i dešifrovanja nije bitan**, npr: $f(g(x)) = g(f(x))$
- Ovakve funkcije postoje.
- Većina ih je dvosmerna (mogu se lako izračunati ali je lako naći i njihovu inverznu vrednost).
- Primer dvosmernih funkcija:
  - $f(x) = 2x$
  - $f(x) = x^2$
  - Uključivanje/isključivanje prekidača
  - …
- Međutim, ovakve funkcije **nisu poželjne** u kriptografiji.
- Od interesa su **jednosmerne funkcije** (*one way*), tačnije neki oblici ovih funkcija.

Jednosmerne funkcije

- Jednosmerne funkcije relativno lako mogu da se izračunaju, ali njihova inverzna vrednost može da se odredi samo izuzetno složenim postupkom.
  - Za dato x lako se računa $f(x)$, ali je za dato $f(x)$ teško izračunati x.
  - Šta se podrazumeva pod pojmom "teško" izračunati?
    - Ovaj pojam se odnosi na probleme koji se ne mogu rešiti u prihvatljivom vremenskom periodu koristeći:
      - Najbolji poznati algoritam
      - Najbolju raspoloživu tehnologiju.
  - U čemu je njihov značaj?
  - Poruka šifrovana jednosmernom funkcijom ne može da se dešifruje!
  - Čemu služe?
- Za kritografiju sa javnim ključem značajne su **jednosmerne funkcije sa zamkom** (*trapdoor one way function*).

Jednosmerne funkcije

- Jednosmerne funkcije sa zamkom su **poseban oblik jednosmernih funkcija**.
  - Lako ih je izračunati u jednom (direktnom) smeru.
  - Teško je izračunati inverznu vrednost.
  - Ako je poznata tajna vrednost – zamka, onda se lako može izračunati i direktna i inverzna vrednost.
- Za dato x:
  - Lako je izračunati $f(x)$
  - Teško je izračunati x iz $f(x)$.
  - Ako je poznata tajna vrednost y, lako se računa x iz f(x) i y.
- Modularna aritmetika obiluje jednosmernim funkcijama.

Jednosmerne funkcije

- Problem:
  - Strogo matematički gledano, nije dokazano da postoje:
    - Jednosmerne funkcije
    - Jednosmerne funkcije sa zamkom.
- Uprkos tome, postoje dve funkcije koje se smatraju kandidatima za funkcije sa pomenutim svojstvima:
  - **Proizvod celih brojeva**, čija je inverzna funkcija faktorizacija dobijenog broja.
  - **Diskretni eksponent**, čija je inverzna funkcija diskretni logaritam.
- Ove dve funkcije su lake za izračunavanje, dok se veruje da to nije slučaj sa njihovim inverznim funkcijama.

Difi-Helmanov (DH) algoritam za razmenu ključa

- Razvijen nezavisno na dva mesta:
  - Government Communications Headquarters – GCHQ
    - Džejms Elis, Kliford Koks i Malkom Vilijamson.
  - Stanford univerzitet
    - Difi i Helman.
- Predstavlja algoritam za **razmenu ključeva**.
- Koristi se za razmenu zajedničkog simetričnog ključa.
- Nije namenjen za šifrovanje ili digitalno potpisivnje.
- Sigurnost ovog algoritma se zasniva na računskoj složenosti izračunavanja (jednosmerne funkcije) **diskretnog logaritma**.

Difi-Helmanov algoritam za razmenu ključa

- Za poznato g i x, gde je $x = g^n$, može da se odredi n: $n = \log_g(x)$,
- Ako je $x = g^n \ (mod\ p)$, n se takođe određuje preko logaritma, ali u ovom slučaju diskretnog.
- Primer:
  - Ako je poznato $3^n = 81$, relativno lako se može doći do rezultata ($n = 4$).
  - Ako je $3^n = 1 \pmod 7$, kako doći do n ?
    - Napraviti tabelu:

n	1	2	3	4	5	6
3^n	3	9	27	81	243	729
$3^n \pmod 7$	3	2	6	4	5	1

    - Dobro rešenje za ovu funkciju ali je za npr $328^n \pmod{23713}$ teško izvodljivo!

Difi-Helmanov algoritam za razmenu ključa

- Neka je p veliki prost broj i g takvo da se za svako $x \in \{1, 2,\ldots, p\text{-}1\}$ može naći n tako da je:
  - $x = g^n \pmod p$
- Vrednosti p i g su javne.
  - Alisa bira tajnu vrednost a (veliki slučajan ceo broj).
  - Bob bira tajnu vrednost b (veliki slučajan ceo broj).
  - Alisa javno šalje vrednost $g^a \pmod p$ Bobu.
  - Bob javno šalje vrednost $g^b \pmod p$ Alisi.
  - Oboje računaju zajedničku tajnu vrednost $g^{ab} \pmod p$.
- Ta zajednička tajna vrednost može da se koristi kao simetrični ključ.
- Napomene:
  - $(g^a)^b \pmod p = g^{ab} \pmod p$
  - $g^a g^b \pmod p = g^{a+b} \pmod p \neq g^{ab} \pmod p$

Difi-Helmanov algoritam za razmenu ključa

- Pretpostavimo da Alisa i Bob koriste g^{ab} (mod p) kao simetrični ključ.
- Trudi može da sazna vrednosti: g^a (mod p) i g^b (mod p).
    - Ove vrednosti su poslate su javno.
- Ako Trudi nađe vrednosti a ili b, sistem je razbijen.
- Ako Trudi reši problem diskretnog logaritma, mogla bi da nađe vrednosti a ili b.

- **Javno:** g i p
- **Tajno:** Alisin eksponemt a i Bobov eksponent b.

$g^a \pmod{p}$ →

← $g^b \pmod{p}$

Alisa, a Bob, b

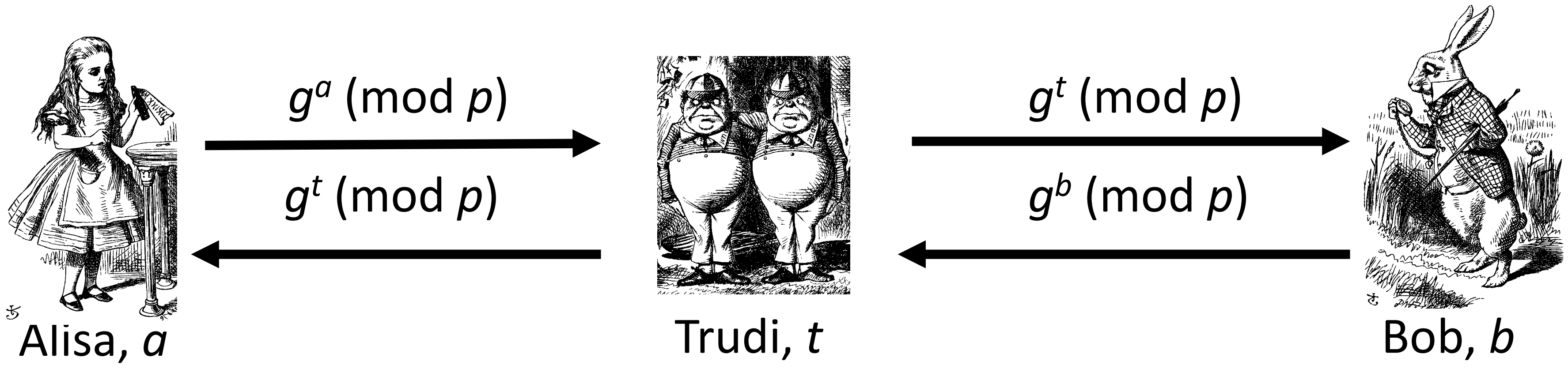
- Alisa računa: $(g^b)^a \pmod{p} = g^{ba} \pmod{p} = g^{ab} \pmod{p}$
- Bob računa: $(g^a)^b = g^{ab} \pmod{p}$
- Kao simetrični ključ može da se koristi razmenjena tajna vrednost $K = g^{ab} \pmod{p}$

Difi-Helmanov algoritam za razmenu ključa (primer)

- Primer: 7^n (mod 11)
  - Alisa bira *a*=3.
    - Računa 7^3 (mod 11) = 343 (mod 11) = 2
    - Šalje Bobu A=2.
  - Bob bira *b*=6.
    - Računa 7^6 (mod 11) = 117649(mod 11) = 4
    - Šalje Alisi B=4.
  - Alisa uzima Bobov rezultat i računa: B^a (mod 11) = 4^3 (mod 11) = 64 (mod 11) = 9.
  - Bob uzima Alisin rezultat i računa: A^b (mod 11)= 2^6 (mod 11) = 64 (mod 11) = 9.
- U praksi se za *p*, *g*, *a* i *b* koriste veliki brojevi!
  - Konačan rezulat je mnogo veći broj koji može da se koristi kao ključ za simetrično šifrovanje.

Difi-Helmanov algoritam za razmenu ključa (primer)

- DH algoritam je osetljiv na napad tipa čovek u sredini (*man-in-the-middle*).



- Trudi deli tajnu g^{at} (mod p) sa Alisom.
- Trudi deli tajnu g^{bt} (mod p) sa Bobom.
- Alisa i Bob ne znaju da Trudi postoji!
- Potreban je **mehanizam autentifikacije**.
  - Potrebno je da obe strane budu sigurne u poreklo poruka!

Kriptografija sa javnim ključevima

- Difi i Helman su **predložili primenu asimetričnog šifarskog sistema**.
  - Asimetrični ili sistem sa javnim ključem.
  - Ideja je njihova, ali nisu predložili funkciju koja bi radila na ovaj način.
- Ključ za šifrovanje i dešifrovanje su različiti.
  - Alisa **ima javni ključ** koji je svima dostupan i koji se koristi za šifrovanje poruke.
  - Samo Alisa ima **tajni ključ** koji je neophodan za dešifrovanje poruke.
  - Privatni i javni ključ su **povezani** odgovarajućim matematičkim relacijama.
- Kriptografija sa javnim ključevima nije "bolja" od kriptografije sa simetričnim ključem.
  - U opštem slučaju ona je i do 1000 puta sporija od simetričnih sistema.
  - Sistemi sa javnim ključem mogu da se korsite za šifrovanje, autentifikaciju, digitalni potpis i razmenu simetričnih ključeva.
  - Simetrični sistemi se najčešće koriste za šifrovanje veće količine podataka.

RSA algoritam

- RSA algoritam je nastao 1978. godine
- Autori su tri istraživača sa MIT Univerziteta: Ronald Rivest, Adi Šamir i Leonard Adelman.
- Kliford Koks, Britanski matematičar koji je radio za GCHQ, objasnio je isti sistem u internoj dokumentaciji 1973.
  - Njegov pronalazak nije objavljen do 1997. godine jer je bio državna tajana.
- MIT je zaštitio algoritam patentnim pravom 1983.
  - Patentno pravo je isteklo 2000. godine.

RSA algoritam

- Potrebno je naći funkciju C=E(M, K_e) koja menja poruku M (otv. tekst) u šifrat C.
  - Funkcija E(M, K_e) treba da bude **jednosmerna**.
- Alisa (ili bilo ko drugi) koristi tu funkciju da šifruje svoju poruku pre nego što je pošalje Bobu.
- Bob treba da ima mogućnost da uz poznavanje tajne vrednosti K_d primeni **inverznu funkciju** M=D(C, K_d) kako bi od šifrata C dobio poruku M.
- Za svako M treba da važi: M= D(E(M, K_e), K_d).
  - Dakle, potrebna je jednosmerna funkcija sa zamkom.

- Funkcija koja zadovoljava iznete pretpostavke ima oblik: $f(x) = x^{e} \pmod{N}$.
    - Uz odgovarajući izbor vrednosti e i N, ova funkcija je jednosmerna.
        - Uz poznavanje tajne vrednosti može da se nađe njena inverzna vrednost.
        - Bez poznavanja tajne vrednosnti ona je praktično nerešiva.

RSA algoritam

- **Postupak šifrovanja**:
  - $C = M^{e} \pmod{N}$
  - C je šifrat, M je poruka (otvoreni tekst).
  - Šta su i kako odrediti vrednosti e i N ?
- **Postupak dešifrovanja**:
  - $M = C^{d} \pmod{N} = (M^{e})^{d} \pmod{N} = M^{ed} \pmod{N}$
  - Šta je i kako odrediti vrednost d ?
- Obe strane u komunikaciji znaju vrednosti N *i* e.
- Samo prijemna strana zna vrednost d.

RSA algoritam

- Javni ključ: (N ,e)
- Privatni ključ: d
- Zahtevi:
  - e, d i N treba da su takvi da je $M^{ed} = M$ (mod N) za svako $M < N$
  - Relativno lako za izračunavanje: M^{e} za $M < N$
  - Praktično nemoguće izračunati d za dato e и N
    - Računski sigurno za dovoljno veliko e i N.

- **Generisanje ključeva**:
  - Izabrati 2 velika prosta broja p i q.
  - Formirati proizvod $N=pq$.
  - Izračunati $\varphi(N) = (p-1)(q-1)$.
  - Izabrati eksponent e takav da je uzajamno prost sa $\varphi(n)$ i manji od $\varphi(N)$.
  - Naći eksponent d takav da je $d = e^{-1} \pmod{\varphi(N)}$.
    - Ili $de = 1 \pmod{\varphi(N)}$.
  - **Javni ključ:** (N,e)
  - **Privatni ključ:** d

- **Da li su zahtevi zadovoljeni**?
  - Za dato $C = M^{e}$ (mod N) treba pokazati
    - $M = C^{d}$ (mod N) = M^{ed} (mod N)
  - Iskoristićemo Ojlerovu teoremu:
    - Ako je x uzajamno prost u odnosu na N tada je $x^{\varphi(N)} = 1$ (mod N)
  - Činjenice:
    - $ed = 1$ (mod $\varphi(N)$) = 1 (mod $(p - 1)(q - 1)$)
    - Po definiciji modula $ed = k(p - 1)(q - 1) + 1$
    - $\varphi(N) = (p - 1)(q - 1)$
    - Tada je $ed - 1 = k(p - 1)(q - 1) = k\,\varphi(N)$
  - $M^{ed} = M^{(ed - 1) + 1} = M \cdot M^{ed - 1} = M \cdot M^{k\,\varphi(N)}$
  - $M \cdot (M^{\varphi(N)})^{k}$ (mod N) = $M \cdot 1^{k}$ (mod N) = M (mod N)

- **Primer**:
  - Izabrati "velike" proste brojeve: $p = 11$, $q = 3$.
  - Odrediti $N = pq = 33$.
  - Odrediti $(p\text{-}1)(q\text{-}1) = 20$.
  - Izabrati $e = 3$ (uzajamno prost sa 20).
  - Naći d takvo da je ed = 1 (mod 20)
    - d = 7, odgovara zahtevu
  - Javni ključ: $(N, e) = (33, 3)$
  - Privatni ključ: d = 7
- Neka je poruka $M = 8$.
  - Šifrovanje:
    - $C = M^e \text{ (mod } N) = 8^3 = 512 = 17 \text{ (mod 33)}$
  - Dešifrovanje:
    - $M = C^d \text{ (mod } N) = 17^7 = 410{,}338{,}673 = 12{,}434{,}505 * 33 + 8 = 8 \text{ (mod 33)}$

RSA algoritam

- Postupak šifrovanja i dešifrovanja obuhvata celobrojne računske operacije sa porukom M.
- Poruku M prvo treba pretvoriti u broj (a=01, b=02, ..., z=26).
  - Primer: sat predstavljamo kao 030120.
- Računa se po modulu N.
  - Potrebno je da je N > M, da bi proces šifrovanja bio jednoznačan.
  - Ako je i pored izbora velikog N, M > N, onda poruka M treba da se rastavi na manje celine (blokove).

- **Sigurnost RSA algoritma**.
  - Eva može da zna $C \equiv M^e \pmod{N}$, e i N.
    - To su javne vrednosti.
  - Može li ona da rekonstruiše poruku M ?
  - Ne postoji dokaz da Eva može efikasno da rekonstruiše M na osnovu poznavanja C, e i N, a da pri tome ne zna $\varphi(N)$.
    - Veruje se da ne može.
  - Dokazano je da problem pronalaženja $\varphi(N)$ podjednako složen kao i faktorizacija broja N.
  - Veruje se, mada nije dokazano, da je problem faktorizacije velikih brojeva praktično nerešiv.

- Ako je *N* = 27997833911221327870829467638722601621070446786955428537560009
  92932612840010760934567105295536085606182235191095136578863710595448200
  6576775098580557613579098734950144178863178946295187237869221823983
  - *N* je dužine: 200 dekadskih cifara ili 663 binarne cifre
- Kako odrediti *p* i *q* (*N*=*pq*)?
- Problem je rešen 9.5.2005. (tim sa Univerziteta u Bonu).
  - p = 3532461934402770121272604978198464368671197400197625023649303468776
    121253679 423200058547956528088349
  - q=7925869954478333033347085841480059687737975857364219960734330341455767872818 152135381409304740185467

RSA algoritam

- Povećanje granice sigurnosti zahteva povećanje dužine ključa.
- Razlog: **algoritmi za faktorizaciju** broja se unapređuju (kriptoanaliza).
- Vreme potrebno za šifrovanje i dešifrovanje je proporcionalno **trećem stepenu dužine ključa**.
  - Rezultat: RSA postaje sve sporiji sa povećanjem zahteva bezbednosti.
  - Osnovna primena RSA je za šifrovanje kriptografskih ključeva.
    - Simitrični kripto sistemi se koriste za zaštitu podataka.
- RSA algoritam je oko 1500 puta sporiji od DES algoritma.
  - Razlog: računanje eksponenta i modula.
  - Generisanje brojeva koji se koriste u RSA algoritmu zahteva vreme.
- Testiranje vrednosti N u odnosu na poznate metode faktorizacije je i dalje otovreno pitanje.
  - Savremena saznanja preporučuju dužinu N > 1000 bita (1024, 2048,....)

Upotreba kriptografije sa javnim ključem

- Kriptografijom sa javnim ključem može da se postigne:
  - Poverljivost.
    - Prenos podataka
    - Skladištenje podataka.
  - Autentifikacija.
  - Digitalni potpis, koji obezbeđuje **integritet** i **neporecivost** (non-repudiation).
    - Nema servisa neporecivosti u sistemima sa simetričnim ključem!

RSA digitalni potpis

- Javni ključ je (N, e), privatni je d.
- **Digitalno potpisivanje** poruke M: $S = M^d$ (mod N)
  - Napomena: kod RSA, dešifrovnje i potpisivanje su iste operacije.
  - Za računanje S je neophodno poznavanje tajnog ključa d.
- **Potvrda ispravnosti digitalnog potpisa** na poruci M: S^{e} (mod N) =(M^d)^{e} (mod N)= M
  - Napomena, verifikacija potpisa je ista operacija kao i šifrovanje..
- Svako ko zna javni ključ (N, e) može da potvrdi ispravnost digitalnog potpisa.

Neporecivost: sistemi sa simetričnim ključem

- Alisa izdaje nalog za kupovinu 100 akcija svom brokeru Bobu.
- Alisa izračuna MAC primenom simetričnog ključa K_{AB} (obezbeđen je servis integriteta).
- Vrednost akcija se smanjila za 80%, Alisa tvrdi da nije izdala nalog za kupovunu.
- Može li Bob da dokaže da je Alisa izdala nalog za kupovinu akcija?
- **Ne**!
    - Kako Bob, takođe, zna simetrični ključ K_{AB}, on je mogao sam da napiše poruku!
    - Problem: Bob ne može da dokaže da je Alisa izdala nalog za kupovinu.

Neporecivost: sistemi sa javnim ključem

- Alisa izdaje nalog za kupovinu 100 akcija svom brokeru Bobu.
- Alisa digitalno potpisuje nalog svojim privatnim ključem (obezbeđen je servis integriteta).
- Vrednost akcija se smanjuje za 80%, Alisa tvrdi da nije izdala nalog za kupovunu.
- Može li Bob da dokaže da je Alisa izdala nalog za kupovinu akcija?
- **Da**!
  - Samo neko ko poseduje Alisin privatni ključ je mogao da digitalno potpiše nalog.
  - Podrazumeva se da Alisin privatni ključ nije ukraden.

Oznake.

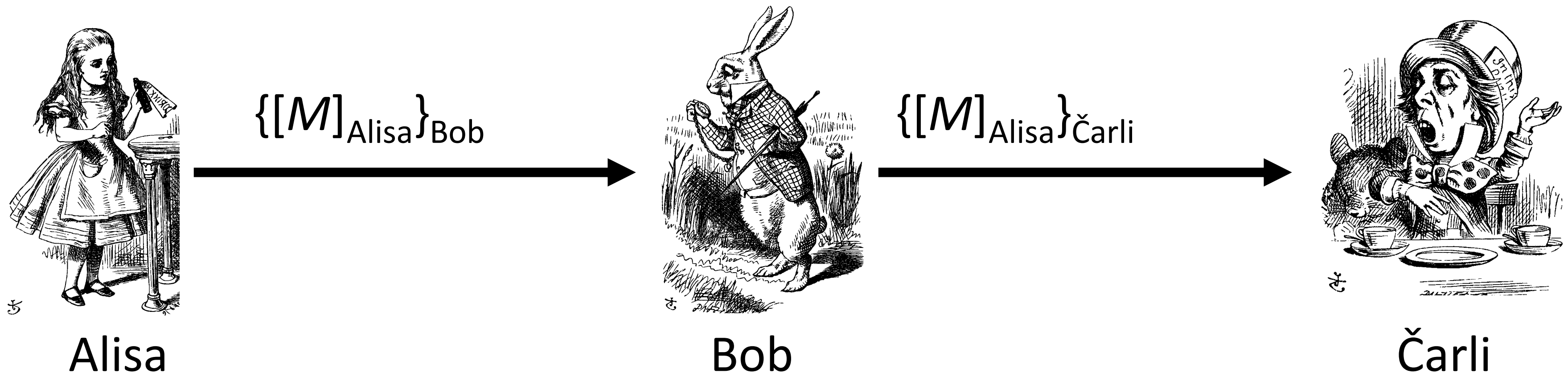
- **Šifrovanje** poruke M Alisinim **javnim ključem**: $\boldsymbol{C} = \{M\}_{\text{Alisa}}$
- **Dešifrovanje** šifrata Alisinim **privatnim ključem**: $\boldsymbol{M} = [C]_{\text{Alisa}}$
- **Digitalno potpisivanje** poruke *M* Alisinim **privatnim ključem**: $\boldsymbol{S} = [M]_{\text{Alisa}}$
  - *S* je digitalno potpisana poruka.
  - Formalno se poklapa sa dešifrovanjem.
- Sledi:
  - $\{[M]_{\text{Alisa}}\}_{\text{Alisa}} = M$
  - $[\{M\}_{\text{Alisa}}]_{\text{Alisa}} = M$

Tajnost i neporecivost

- Pretpostavimo da je potrebno da se istovremeno ostvare servisi tajnosti i neporecivosti.
- Da li sistemi sa javnim ključevima obezbeđuju oba zahteva?
- Alisa šalje poruku Bobu:
  - Potpiše je pa potom šifruje $\{[M]_{Alisa}\}_{Bob}$
  - ili
  - Šifruje je pa potom potpiše $[\{M\}_{Bob}]_{Alisa}$
- Da li je **redosled** bitan?

Tajnost i neporecivost

- **Potpiši pa šifruj**.
  - Alisa šalje poruku Bobu M = “Volim te ...”
  - Bob se kasnije naljutio, dešifruje poruku da bi dobio $[M]_{\text{Alisa}}$ i šifruje je sa Čarlijevim javnim ključem.



  - Čarli misli da je Alisa njemu uputila poruku!
  - **Pitanje**: U čemu je problem?
  - **Odgovor**: Čarli ne razume kriptografiju javnog ključa!

Tajnost i neporecivost

- **Šifruj pa potpiši**.
  - Alisa šalje poruku Bobu M = “Moj novi izum …”
  - Čarli je ljut na Alisu i Boba, presreće poruku, dešifruje Alisinim javnim ključem, potpisuje svojim tajnim ključem i šalje Bobu

Alisa

$\xrightarrow{[\{M\}_{\text{Bob}}]_{\text{Alisa}}}$

Čarli

$\xrightarrow{[\{M\}_{\text{Bob}}]_{\text{Čarli}}}$

Bob

  - Bob misli da je poruku poslao Čarli!
  - **Napomena**: Čarli <u>ne može</u> da dešifruje M.
  - **Pitanje**: U čemu je problem?
  - **Odgovor**: Bob ne razume kriptografiju javnog ključa!

Tajnost i neporecivost

- **Napomene**.
  - Ne treba zaboraviti da je javni ključ svima dostupan.
    - Svako može da izračuna $\{M\}_{\text{Alisa}}$.
  - Privatni ključ je tajan.
    - Samo Alisa može da izračuna $[C]_{\text{Alisa}}$ ili $[M]_{\text{Alisa}}$.
  - Drugim rečima:
    - Svako može da šifruje poruku za Alisu ali samo ona može da je dešifruje.
    - Samo Alisa može da digitalno potpiše poruku svojim privatnim ključem, ali svi mogu da provere ispravnost potpisa ukoliko znaju javni ključ.

Sertifikati javnih ključeva

- **Sertifikat** sadrži podatke o korisniku (ime, ...) i njegov **javni ključ**.
  - Može da sadrži i druge informacije.
- Izdavač sertifikata **digitalno potpisuje sertifikat**.
  - Time se obezbeđuje integritet podataka.
  - Potpis na sertifikatu se može proveriti pomoću javnog ključa izdavača sertifikata.

Sertifikaciono telo

- **Sertifikaciono telo** (*Certificate authority*, CA) je treća strana od poverenja (TTP) koja izdaje i potpisuje sertifikate.
  - Proverom potpisa na sertifikatu se istovremeno utvrđuje i identitet vlasnika odgovrajućeg javnog (privatnog) ključa.
  - Međutim, na ovaj način se ne može utvrditi identitet izdavača sertifikata!
  - Sertifikati su javni!
  - Šta ako CA napravi grešku?
    - (CA izda nekom drugom već izdati sertifikat)
  - Zajednički format za sertifikate je X.509

PKI

- **Infrastruktura javnog ključa** (Public Key Infrastructure, PKI) sastoji se od svih podsistema koji su neophodni za bezbednu upotrebu kriptografije sa javnim ključevima:
    - Generisanje i upravljanje ključevima
    - Serifikaciona tela
    - Povlačenje sertifikata
    - ...
- Ne postoji opšti standard za PKI.
- Razmotrićemo nekoliko “modela poverenja”.
    - Osim tri koja ćemo pomenuti, postoje i drugi modeli.

PKI modeli poverenja

- **Monopolski model**.
  - Jedinstvena organizacija od poverenja je sertifikaciono telo za sve korisnike.
  - Model je predložila VeriSign iz razumljivih razloga jer je najveće komercijalno CA.
  - Veliki problem nastaje ako se takvo CA bilo kada kompromituje.
  - Šta će se desiti ako to sertifikaciono telo ne ostvari šire poverenje?
- **Oligarhijski model**.
  - Postoji više sertifikacionih tela.
  - Korisnik može sam da odluči kojim sertifikacionim telima će ukazati poverenje a kojima ne.
- **Anarhijski model**.
  - Svako može da bude CA a korisnik sam odlučuje kome će verovati.
  - Ovaj pristup se koristi u PGP.
  - Zašto se ovaj model naziva “anarhijski”?
    - Pretpostavimo da je sertifikat potpisao Frenk (ne poznajemo ga) ali verujemo Bobu koji kaže da je Alisa od poverenja i da ona garantuje za Frenka. Da li treba da verujemo Frenku?

Prednost kriptografije sa javnim ključevima

- **Poverljivost bez deljenja tajni**.
  - Veoma korisno u komercijalnom svetu.
  - Nema problema sa razmenom ključeva
    - … kao kod simetričnih kripto sistema.
- **Autentifikacija može da se obavi bez deljenja tajni**.
  - Koristi se digitalni potpis kao dokaz o poreklu poruke.
  - Nema potrebe da se sakriva javni ključ, ali je potrebno da se zna da je Alisin javni ključ stvarno njen javni ključ.

Nedostaci kriptografije sa javnim ključevima

- Algoritmi su za 2-3 reda veličine **sporiji**.
  - Modularna (eksponencijalna) aritmetika je računarski zahtevna.
- Tipična upotreba: **hibridni sistemi**.
  - Sistemi sa javnim ključem se koriste za razmenu simetričnog ključa
  - Potom se prelazi na simetričnu kriptografiju.
  - IPsec i SSL
- Ključevi su duži.
  - 1024-2048 bita (RSA) prema 128-256 bita (AES).
- Sigurnost se zasniva na **pretpostavkama koje nisu dokazane**.
  - Šta ako se reši problem faktorizacije?

Literatura

1. M. Stamp: *Information Security*. John Wiley and Sons.
2. M. Veinović, S. Adamović: Kriptologija 1. Univerzitet Singidunum, Beograd. *

* Može se besplatno preuzeti sa portala: www.singipedia.com

Hvala na pažnji

Pitanja su dobrodošla.