

## 11. SIGURNOST RAČUNALNIH SUSTAVA

- uz obvezne na sigurnosti operativnih sustava
- osl. zemljiš. fer. nr — algoritmi i protokoli, simulacije
- na što će sigurnost oduzeti - više sve je dno (vojska, banke...) te vanjski upitci (potres, raspod EES-9, ...), od kojih, u utrviči nastavi mehanizmi (od samih korisnika - poslužit se za sebe same...) itd.

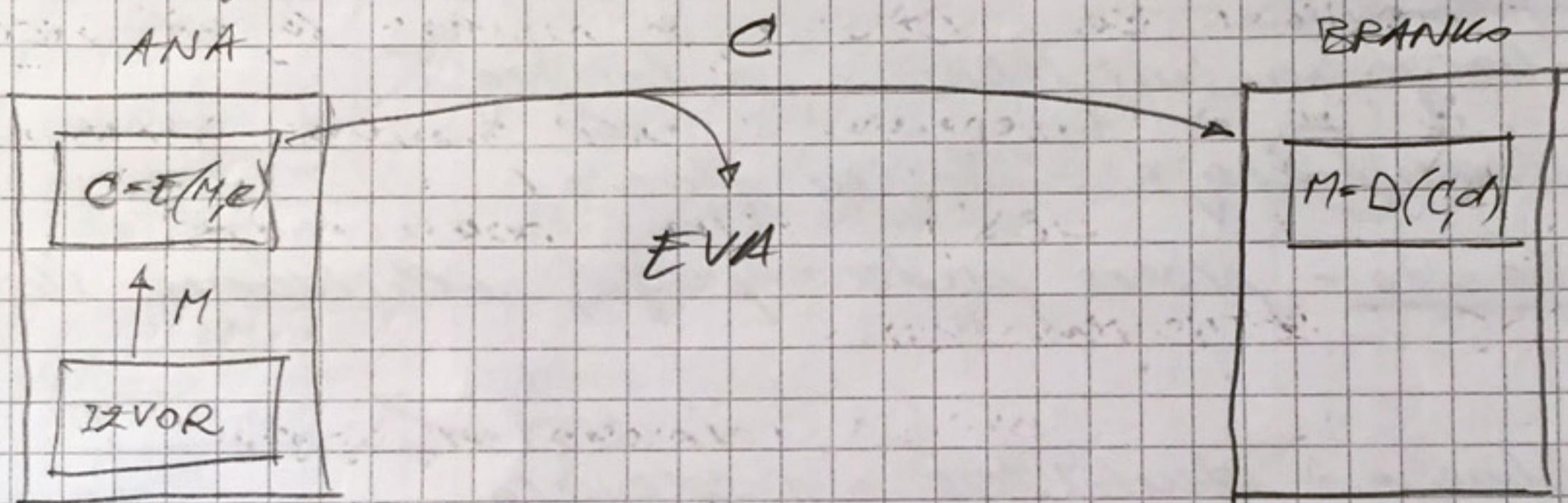
### Osnovni pojmovi

- identifikacija, autentifikacija, autorizacija
  - autentifikacija — koji su fac's
- identifikacija — proces predstavljanja, označavanja lica po asobi
- autentifikacija — identifikacija + provjera
  - autentifikacija i identifikacija o stvarnosti, mogu razlikati individualno - ovde je o datim na logiji se identifikacija vrsti; aps. obzale preta (od jednom obog), logini i password je druge
- autorizacija — autentifikacija + provjera ovlasti

### Napadi (proj. etape)

ZAHVETI	NAPADI (PROJETAJTE)	ANALIZA	BRANKO
1. Autenticnost (N3.)	1. Preključivanje		
2. Integritet (N5.)	2. Prekidanje		
3. Tajnost (N1.)	3. Ložno predstavljanje		
4. Nepovećivost (N6.)	4. Ponavljajuće stare poruke (izmjenjene poruke)		
5. Kontrola pristupa (N2.)	5. Modifikacija paketa		
6. Raspoloživost (N2.)	6. Poticanje		

- pouzivá slouje staré poručky - upr. vložením slouje zahýbavá za elektronické placičky
- poricany - napadací je poslal poruku, ne je  $\checkmark$
- zahýbaví 1.-4. na osnově sigurnosti zahýbaví
- dodatki sú 5. a 6.
- možné se diskutovat → rovnice da zahýbaví správovanou vise od jednoho napada
- N4. - potřebu jde druhým posláním mechanizma (upr. time stamp)



- předpokládáme da ANA má risti sigurno ráčunalo
  - napadací se spouští týkají kódů - může risti ráčunalo odkryvat slabosti da ráčunálka; ráčunalo da nemá svá ráčunala zabezpečit instalace da výrobcu
  - ráčunalo definitivně n'je sigurno
  - M - poručka a jasnom obecně
  - C-E → kryptiraj, M - poručka, K - klíč
  - Salje se C (pp da jde same kanal nesiguran)
  - M-D → dekryptiraj
  - mye dolno? (symetrický systém), e=d=k

## Tipovi kriptografskih algoritama (zavisnosti su o operatori)

### 1. Simetrični algoritmi

- $e = d = K \rightarrow$  tajni ključ, simetrični ključ ili sjednički (session) ključ
- dva ključa ključ i tajni dake samo dale tajne sprednja (može se prepoštaviti da je moguće probiti ključ)
- generira se slučajno
- DES (Data Encryption Standard) - prvi standard (danas se smatra nesigurnim)
  - problem kod DES-a je prenali ključ - moguće je pogoditi tajnu (izpad grubom silom)
- 3DES (Triple DES) - troput pokreće DES
- DESX - izbijegavši DES
  - 3DES se smatra potpuno sigurnim, i danas je savrem u redu (DESX je -fiksno desiran)
- AES (Advanced ES) - nov algoritam, siguran
- IDEA - siguran
- sigurnost. fer. hr - nisuće algoritama (veli sigurni, neli ne)
- AES i DES su standardni na zapadu (Rusija i u njoj privatila - GOST, Gosudarstveni Standard)

### 2. Asimetrični algoritmi

- dva ključa,  $e \neq d$  ( $P$ -public ili javni ključ,  $S$ -secret ili privatni ključ)
- $S$  - nije secret ili tajni,  $P$  (za public) je bilo zašeto
- RSA - siguran (i jedan od rijetkih)
- asimetrični ključ se može generirati protokolom,

apr. Diffie - Hellman

- ECCES (Elliptic Curve Encryption System)
- ElGamal

3. Funkcije za izračunavanje sažetka poruke (hash functions)

- MD5, SHA-0, SHA-1, 2, 3 →
  - pre 2 god. proglašen SHA-3 (njegova četvrta proučavanja)

- naprave za šifriranje
  - simetrični sustav
- Kirchhoffov princip - kripto sustav mora biti siguran i onda bude sa sve informacije o kripto sustavu javno poznate osim tajnog ključa.
- ovaj princip zadovoljava vise modernih kripto sustava, u sklopu su uporabljaju
- funkcije se na tor funkcij, princip:

$$\text{XOR } C = M \oplus K$$

$$M = C \oplus K$$

$$M = 1011001$$

$$K = 1101110$$

$$C = 0110111$$

$$K = 1101110$$

$$M = 1011001$$

- uveigurajući simetrični kripto sustav se naziva ono time pad (kript. reduksivna bolješnica) - matematički dokazano

- zašto se ne koristi samo uveiguraji? drugi su i znatno komplikiraniji; dva razloga:

1. Siguran je jedino onda ako je generiran brž slučajno (ne pseudo slučajno)

- uveiguraji nadiju je upr. vaoanje novčića (pismo - 0, glava - 1)

2. Ključ mora biti jednake duljine 64 i  
poruka

- danas - 128 do 256 bita
- konkretno, file od 10 GB zahtjeva ključ  
od 10 GB - postaje besmisleno sigurna  
razmjena ključa (Bilo je dada loša vrednost  
poruka jer su iste veličine)
- asimetrični algoritam je 4-5 redova  
većine sporije od simetričnog - postupak  
generiranja ključa je uaulta
- Vigenereova řeč - upr. tipkovnica klasičnih mrežtela  
one time pod - da je raspodjeljiv potom razumijec  
generator i primljanju ključa bilo kriptiranje, inace řečitom

### DES (Data Encryption Standard)

- IBM 1977., nastao iz Lucifera
- 56 bita, veliki ključ + 8 bitova za zaštitu (redundanciju),  
obiveni paritetni bitovi (štite od greške pri kom prijenosu),  
odnosno ukupno 64 bita
  - napredna manja DES-a je upravo prekratka ključ  
(za danas užiće pojmove)
  - Microsoft je zbog politike mogao izvjesti Windows  
sa samo 40-bitnim DES-om
  - Cerberos - za autentifikaciju (MIT)
    - razvijan na DES-u, opet 40 bita
    - u Evropi upravljen konzorcijem za sličan proizvod  
je nastaje Sesam; potom se ulazi uga zbrajan  
izvoza kriptografskog ključa i Sesam restuje  
s ključem

## - 1998. - DES Challenge II

- RSA - DES Challenge s nagradom od \$10000
- racunalno vrijedno \$250000, i.e. za manje od tri dana našlo了解
- rečeno je 3DES s dvostrukim ili trostrukim了解, odnosno 112 ili 168-bitnim了解em
- DES64 usugližen u ažin rada DES-a: (sigurnost: 2<sup>56</sup>, fuz. 40):
  - kriptira se blok od 64 bita, već poruka se razbijaju, manje se nadopunjaju (padding)
  - operi dio-kriptiranje, desni, i generiranje potkrijenja
  - DES - 1. faza je generiranje 16 podkrijenja iz jednog 56-bitnog; veličina podkrijenja je 64 bita, to je priprema za kriptiranje koje ima 16 koraka
  - kriptiranja poruka se mora biti jednako veličine kao i originalna poruka
  - prije kriptiranja - padding, dodaju se nule
  - ukloni inverzne permutacije se 64-bitni podatci dopeli na operi "desni" od 32 bita
  - desni dio se expandira na 48, ukloni tog XOR; razbijanje na  $8 \times 6$  bita, i tih rad kreira kriptiranje
  - S-tablica (S-box, S-kutija, i.e. substitucijska tablica) je SLE algoritma
  - blokovi se zamjenjuju; postupak se ponavlja 16 puta te slijedi zadnja permutacija uklon oega se dobiva kriptirani podatak

- DES: funkcija  $F$  ("bijeli dio")

- samo XOR bit rezultirao sustavom koji se može prividno razbiti

- siguran kripto sustav je onaj kripto sustav za koji nije poznat ni jedan algoritam koji je složnost manja od složnosti pretrage u prostoru rješenja, a slati se premačuje tajnog ključa; kolobar - uspešan napad je napad koji je manji od uvedene složnosti

- np. AES - 128-bitni ključ, prostor  $2^{128}$

po bit uspešan napad bio, primjerice,  $2^{128} - 1$  ili  $2^{125}$

- protiv uspešnog napada broj se poveća 8 S-tablica

- imaju četiri retka i 16 stupaca:

	0	1	2	...	15
00					
01					
10					
11					

→ vrijednosti u tablici

- svi su broj će se ponoviti 4 puta

- ulaz je 6-bitni podatak np.

001011

gdje srednja četiri predstavljaju adresu stupca,  
a prvi i zadnji adresu retka te se dobrova  
vrijednost iz S-tablice

- osnovna namjena S-tablice je nelinearnost,  
uvijek je jednозначно doći do ulaza

- dekriptiranje pomoću DES-a je jednako kao  
enkripcija, ali se postupkuven ustojaju obrnutim  
redoslijedom

- najteži dio je onaj funkcije  $F$

- jedina "komplikacija" DES-a su S-tablice
- ideja simetričnih kripto sustava - kriptiranje blok po bloku
- dva tipa simetričnosti:

1. kriptiranje blok po bloku

2. kriptiranje cijela podataka

- kriptiranje cijela podataka - rotacije je kriptirati bit po bitu, i na ovaj način se razlikuje od blok po bloku metoda

- 12 blok po blok algoritma je moguće prepravljati  
dolži varijantu koga kriptira cijeli podatak (vrši o  
vome kasnije)

- kriptiranje i dekriptiranje igrom Solitaire

- među jednostavnim algoritmima - danas popularan  
za manje zaštitnim sustavima

- RC4 - temelj za napade na routere

- WEP, WPA - jedan je nesiguran

- AES - mobilni, prihvaćen u Evropi, Sjevernoj Amerikama

- inače DES-a:

- 3-DES:

$$3\text{-DES}(M, K_1, K_2, K_3) = \overbrace{\text{DES}}^{\text{3-kući}} \left\{ \overbrace{\text{DES}}^{\text{3-kući}} \left[ \overbrace{\text{DES}}^{\text{3-kući}}(M, K_1); K_2 \right]; K_3 \right\}$$

3-kući, svaki 56-bitni

- par klučeva, npr. K<sub>1</sub>; K<sub>3</sub> mogu biti isti

- ne radi DES tri puta, već je DES<sup>-1</sup> dekriptiranje  
tj. kriptira se uz K<sub>1</sub>, de-uz K<sub>2</sub> i kriptira  
uz K<sub>3</sub>; obrnuti postupak

$$3\text{-DES}^{-1}(M, K_1, K_2, K_3) = \overbrace{\text{DES}}^{\text{3-kući}} \left\{ \overbrace{\text{DES}}^{\text{3-kući}} \left[ \overbrace{\text{DES}}^{\text{3-kući}}(M, K_3); K_2 \right]; K_1 \right\}$$

- izgrijevni DES ili DESX:

- također  $M$  uključa

$$DESX(M, K_1, K_2, K_3) = DES(M \oplus K_3, (K_1)) \oplus K_2 = C$$

- pouzane u kriptiranju, ali ne provise

- dekriptiranje:

$$DESX^{-1}(M, K_1, K_2, K_3) = DES^{-1}(C \oplus K_3, (K_1)) \oplus K_2$$

### IDEA

- kripto sustav koji se i danas koristi; potpuno siguran

- dovršen 1992.

- kљуč 128 bita

- blokovi dugi ne 64 bita se dijele na 4 16-bitne

- postupak u 9 koraka; trik:

- 8 istih koraka, 1 je drugačiji (zadnji)

- 8 koraka - sudjeluje 4 podbloka i 6 podlijubčeva  
dugme 16 bita

- 9. korak - 4 podlijubča

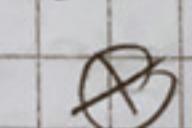
- dakle generira se  $8 \cdot 6 + 4 = 52$  podlijubčeva

- nelinearnost je u čakarom konstrukcije operatora po  
modulu, konkretno zbrajanje; množenje

- množenje po modulu  $2^{16} + 1$

- zbrajanje po modulu  $2^{16}$

- XOR (bit po bit)



operacije logičke

IDEA koristi

- sadržaj S-taulja je fiksan:

- bilo kakva promjena S-taulja bitno oslabljuje  
DES; ako svojetvo nelinearnosti nije do postignuti, a  
suprotstavno je činjenicom diferencijskim  
kriptoanalizom može smanjiti prostor rješenja

- ne postoji uspešan napad na DES, ali je kriptiranje danasne pojmove pre maleu
- zato 3-DES ne valja - prepor!

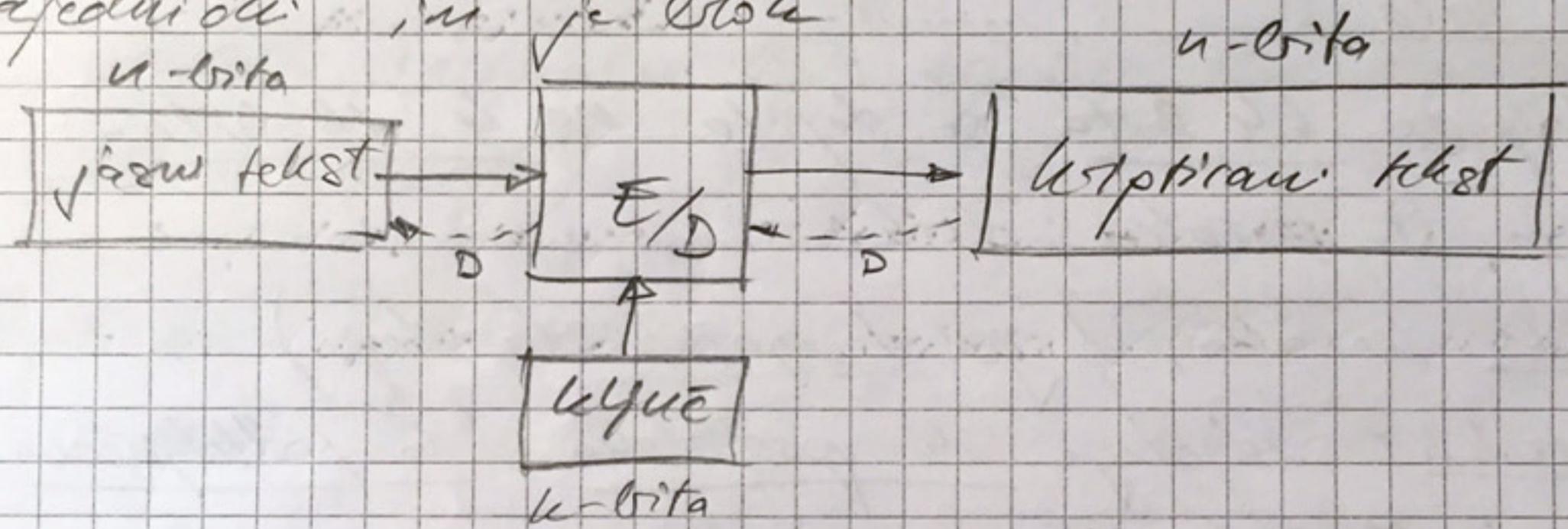
- AES je orži od DES-a

- 3-DES traje tri puta duže, a DES-a koliko ;  
DES

### Algoritam kriptiranja bloka (Block Ciphers)

- doc. dr. sc. Ante Đereš, konzultacije utorak 11-12,  
DSSB

- zadanički im je blok



D : AKB je par algoritama

$$E : D \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$$

t.d. vrijedi:

$$\forall k \exists p D(E(p, k), k) = p$$

- ako je poznato puno parova jasnoog i skrivnog teksta  
i da je se on može dobiti ključ :

Definicija AKB je siguran ako je test? no tenuelje  
kriptiranog teksta pronaći jasni tekot za nepoznat  
ključ K čak ; ako napadač ima mogućnost

$$p_i, q_i = E(p_i, K)$$

pronaći mnogo parova jasnoog i skrivnog teksta.  
Napadač može imati mogućnost dobivanja

$$q'_i = E(p'_i, K)$$

gdje je  $\pi$  podatke po napadacu izboru.

Obez mogućnosti napadaca prethode davanju algoritma  
odnosno slavu su koristiti postupci o pitanju.

Napadac može imati mogućost dobitanja pi za  
proizvoljni  $q$ .

- primjeri algoritama:

- 3-DES -  $n=64$ ,  $k=168$  (DES -  $n=64$ ,  $k=56$ )

- AES -  $n=128$ ,  $k=128, 192, 256$

- razbijanje algoritma kriptiranja bloka

- osnovni algoritam kriptoanalize (gruba sila, brute  
force)

- po da je poznato veliko broj parova  $(\pi_i, c_i) = E(\pi_i, k)$ ,  
na kojima se isprobavaju sve moguće kombinacije ključa  $K$

- za svaku ključ  $K'$  izračunava se:

$$c'_i = E(\pi_i, K') == c_i ?$$

- dovođe se 3-S parova (ne veći redovi veličine)

- moguće je i već poznatih parova, ipak po  
svojim rezultata (e-mail)

- 1997: Internet search: 3 mjeseca (DES Challenge)

- gornji algoritam se vrlo lako može distribuirati

- 1998: EFF machine: 3 dana, cijena stroja \$250000

- 1999: 1+2 (zbog goruge metode) - 22 sata

- 2006: COPACABANA - 120 GPU-a, 7 dana uz malo  
trošak od \$10000

- počela - ne koristi DES

- AES - Gruba sile  $\rightarrow$  ve  $10^9$  računala i  $10^3$  klijenata po sekundi na računalu treba vise napravo godina
- kriptografija ne implementirati samostalno!
- 3-DES je bio samo prvi standard, ali je bio previše spor (48 putova, DES 16)
- $\rightarrow$  kako AES radi:
- općenito:
  - fejzolovo mreža (?)
  - substitucije - permutacija mreža - prvo subst., zatim permut.
  - prvo XOR, zatim rastavljanje u dva polja te substituijanje S-tablicama, potom permutacija
  - postupak se u slučaju AES128 ponavlja 10 puta
  - za S-kutije treba vrijediti inverzibilnost (jer je postupak dobroj početku obrnut)
  - permutacija je po definiciji inverzibilna
- od ulaza se radi matica  $4 \times 4 \times 3$  ( $= 128$  bita)
- postupkom ekspansije klijenca se od gl. klijenca dobita 11 128-bitnih podklijenaca
- Byte Sub - 1B S-tablica
- ShiftRows - posmicanje (rotacija klijenca) svih 8 redaka, prvi redak ne mijenja
- MixColumns - 4x4 srednje redake prenosi na linearnu funkciju

→ Konacna polje  $\text{GF}(2^8)$

- elementi polya scrieri:

$$q_7x^7 + q_6x^6 + \dots + q_1x + q_0, \quad q_i \in \{0, 1\}$$

np:  $x^6 + x^3 + x + 1 = (01001011)_2 \Rightarrow (GB)_{16}$   
oblik  $q_7q_6q_5q_4q_3q_2q_1q_0$

- cele polje se uobičajeno zbrojuje, množi se

- reducirni polinom - de može se izvaditi pravzapravo

koji unosi se dva polinoma (analogno prošlim programima)

$$g(x) = x^8 + x^4 + x^3 + x + 1 = (11B)_{16}$$

(opisan reducirni polinom)

- zbrojne polinoma na modulu 2

$$\begin{array}{r} x^6 + x^3 + x + 1 & 01001011 \\ + x^6 + x^5 + x^2 + x & 01100110 \\ \hline x^5 + x^3 + x^2 + 1 & 00101101 \end{array} \rightarrow \boxed{\text{XOR}}$$

R  
ekvivalentno

- ne raditi direktno polinoma kao u srednjoj

$$\begin{array}{r} x^6 + x^3 + x + 1 & 01001011 = a \\ \cdot \quad x^2 + 1 & 00000101 \\ \hline x^6 + x^3 + x + 1 + & \\ + x^8 + x^5 + x^3 + x^2 = & a \cdot 1 = 01001011 \\ = x^8 + x^6 + x^5 + x^2 + x + 1 & ax^3 = 0100101100 \\ \hline \end{array}$$

pouzdro modulo 2

Boji  
nacin,  
jednostavniji  
za implem.  
hardverski  
implementaci

↓ redukcija modulo  $g(x)$

- prilikom redukcije  $g(x)$  promatra se ostatak:

$$\begin{array}{r} 57 \\ - 17 \\ \hline 40 \\ - 23 \\ \hline 6 \end{array}$$

(konstanta oduzimanja je 17)

$\rightarrow$  ostatak

- sljedeće, sa  $g(x)$  - oduzimamo ga dok ne dobijemo manje stupanj od  $g(x)$

- procedura:

$$\begin{array}{r} 01|0110|0111 ? \\ 10001|1011 \quad \text{[XOR]} \\ \hline 00|0111|1100 \end{array}$$

- da je blok  $x^8 b_7$  se oduzima s  $x \cdot g(x)$

- dijelye:

- grupa sva - sprobaće sve inverze npr. za  
 $(11)_10 \cdot \text{inverz} = 1$

$\rightarrow$  AES blok

- postoji korak naziva Mix Column blok

- Intel - aesenc, aesenclast uz 2 128-bitne registrе  
(u jedan ido ulje, a drugi podatak)

- redukcija AES - manje runda (npr. 8 runda)

što ako postoji više blokova podataka?

- poti pokusaj - razbijanje na blokove ; poselma  
enkripcija svihog bloka (Electronic Codebook)

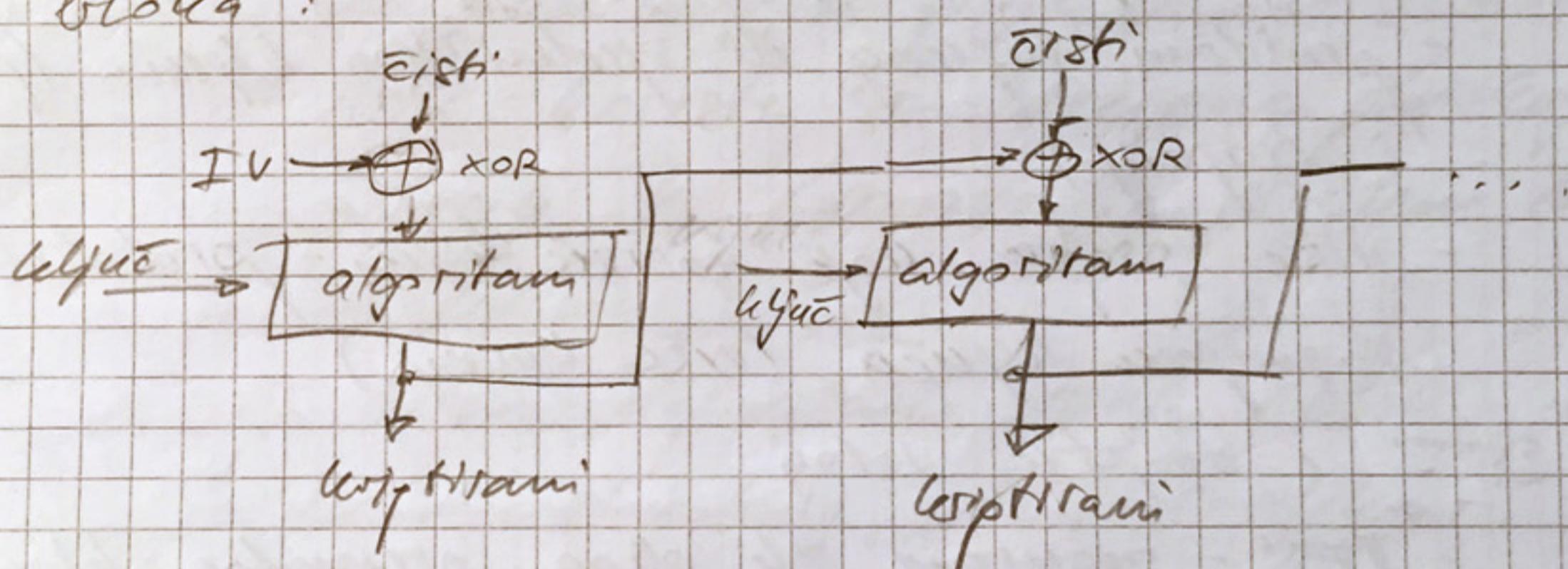
- sa strane korektnosti je dobro

- postoji problem toga što će isti blokovi svih  
teksta imati isti enkripciju teško → moguće  
je otkriti informacije o originalnom tekstu  
(Linux povezana!)

- jedno rješenje - Cipher Block Chaining

- pouzno se daje na blokove

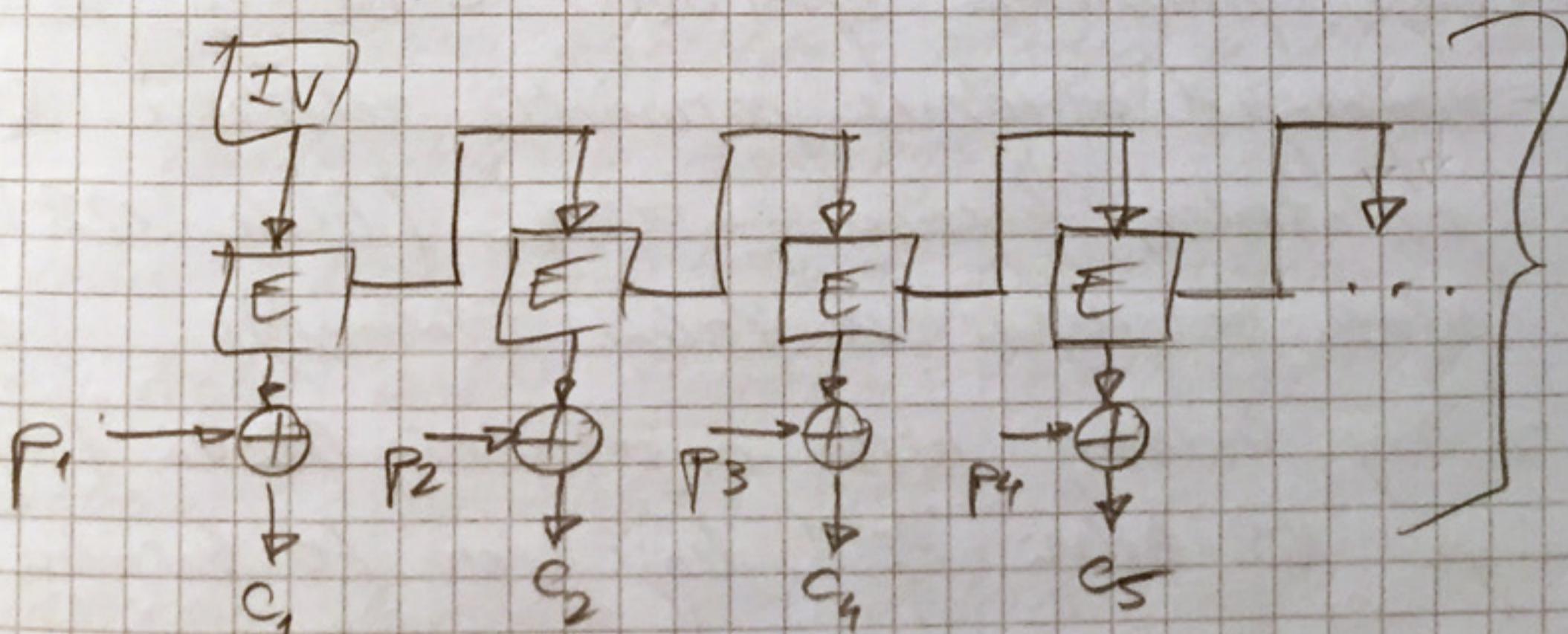
- inicijalizacijski vektor - svečan uz veličinu  
bloka :



- za dekripciju je potreban inicijalizacijski vektor;  
oni je dio skrivene poruke

- dekripciju obrnuto (prvo prvi kriptirani blok)

- slične ideje koristi i Cipher Feedback (CFB) i Output  
Feedback (OFB)



Paketi, ali ne slike  
na isprav, znati:  
drug, obave

- ako užiće nije narančan, kriptografski sustav nije siguran

- kriptografski generatori slučajnih brojeva - sigurniji od standardnih (fks: Crypto Library i Python)

- Electronic Code Book zaključeva blokove koji su viselikanci broja 16, ostali provode padding  
- uklad u koristi ECB!

→ posavljajuće

- očluje se posavljajuće (fks svedoci fpedom - hash i AES)

- smotrijuci algoritmu - važni je bog očuvanja sigurnosti  
- problem rapirog iti spodnje člog ključa (prema i druge)

- više "osoba" koje koriste ključe - protokoli razmjene ključa (više ključa)

- DES i svojstva istog

- DES - siguran, ali zato prenalož ključa nesiguran

- taj problem rešava 3-DES, ali dolazi problem sporosti (trput daje vremena izvodenja)

- rješenje problema sporosti 3-DES-a je AES

- unaprjed određeni parametri relevantne bloku podataka, ključa ; skica (blok 128 biti, uklj. se dodeli na talice blokova)

- broj koraka aussi o veličini bloka (default je 128-bitni AES koji ima 10 koraka)

- funkcije 128-bitog AES-a:

- zamena znakova (superstvujuca tablica)
- pozmali redova
- mješanje stupaca (jedna vrstica u drugom stupcu)
- XOR (za podleguću)

- broj podlegućevih je  $N+1$  gdje je  $N$

broj koraka algoritma

- znak operacije zbrajanja i množenja u mješanu stupacu (za MT i za blisc)

- proces encrpatije i dekriptije

- DES učima razlike modu postupcima  
encrpatije i dekriptije

- AES - koristi se inverzne funkcije  
goruge 4 (poznata je problem inverz  
mixcolumns)

- DES učimo definisanje stampa (blok je  
producentualan); AES - razlika:

- blok AES-a : 16 B

- 16 B 80 "seli" a doye dimenzije (kvadrat  
4x4)

- manipulacije nad stampom u doye  
dimenzije

- novi hash algoritmi - stampa u tri dimenzije  
(producent 4 dimenzije)

- učima potrebe za novim poznavanjem mixcolumns,  
ali treba znati funkcije (operacije) koje koristi

- kakav je podleguć AES-a?

- inverzne funkcije za dekriptiranje

- CBC način kriptiranja je samo sredstvo za sigurnost
- greska jednog podatka rezultira jednom potpunom kriptu bloku, u odnosu na problem jednog bita, a naku bog je u redu (za dekriptiranje)
- rasploštenje u lancu varanje
- put načina kriptiranja, od apisa do same OFB i CTR napravljeni za kriptiranje točki podataka
- OFB - Obezstojični ključ predstavlja višestruko kriptiranje inicijalizujućeg vektora sljedeće, za CTR - dobiva se aproksimacija upoređenje podata
- povećava se veličina klijenta jednostavnija implementacija je garantna za kriptiranje bloka; točka podataka
- "praktična" implementacija algoritma za točku podataka; konzisti sa AES s OFB-om i CTR-om

### Funkcije za izračunavanje sažetka poruke (hash)

- MD5, SHA: SHA-0, SHA-1, SHA-2, SHA-3
- MD5 - danas se smatra nesigurnim, problem je veličina hasha (128 bita)
  - ujedno i ta logika lako je za AES!
  - rotacioni napad - smanjuje se prostor pretrazivanja po polu (korijen), tj:  $2^{64}$  što se razmerno lako provodi

## → MD5

- 5. verzija algoritma, 170 - Message Digest (sazetak poruke; od 128 bita)
- MD5 deli na 512-bitne blokove; zadnji blok se nadopunjava (padding) prema:
  - doda se jedna jedinica u poslednju poziciju svake sljedeće mije t.d. se zadnji 64 bita sastavljaju iz veličine poruke a 0 bitima
  - ako je veličina poruke t.d. nema mesta za jedinicu s vrednjom "1" 64 bita dodaje se novi blok
    - $80_{(16)}$  - dodaje se novi blok sa crnim vulanom osim posljednjeg 64-bitnog
- Blok od 512 bita se deli u podblokove  $M_0, M_1, M_2, \dots$
- stari podblokovi 32-bitni (zbrog 32-bitne arhitekture) - ukupno 16 blokova ( $M_0, \dots, M_{15}$ )
- svaki hash algoritam pate od redudanskog napada (tak se radi i da je smatraju sigurnim)
- MD5 postupak (sljedesu sa SHA-0, SHA-1, SHA-2):
  - 64 podjedinica u 4 grupe
  - početne konstante  $A_0, B_0, C_0$  i  $D_0$ 
    - upitova konstanta je jednodim. podatak od 128 bita (zravi 32 bita; arhitektura!)
  - prava datoteka - hash će biti upravo  $A, B, C, D$
  - a prvi kružci prva funkcija, drugi druga itd.
  - jedan ulaz konstante, drugi ulaz prvi blok; naku nog pri drugi blok itd.

- 64 konstante - sačinjuju se jedinice i daju memoriju
- od ovog svakog do su konst. A<sub>0,1,..,D<sub>0</sub></sub> 32-bitne i da postupale imaju 64 koraka
- 864a približava jedan od 64 koraka
- prvi korak - inicijalne vrijednosti A<sub>0,1,..,D<sub>0</sub></sub>
- A u 2. koraku je prošli D i sljede za ostale - počinje 29. jedna 32-bitna konstanta
- funkcija F<sub>i</sub>(B, C, D) ; izrajava po modulu na izlazu
- M - 512-bitni podatak M je dodjeljen u 32-bitni podatak
- K<sub>i</sub> - konstanta je 2<sup>32</sup>abs(sin(i))
- S rotacija je S rotacija
- zapravo se sve konst. osim jedne pozivaju, dok se jedna pozivajuva u "kompleksnim" način
- postupak se ponavlja dok se ne proteže 512-bitne podatke

## SHA

→ SHA-0 i SHA-1

- 1995. NSA predstavlja SHA-1 kao zamjenu za SHA-0

- 1998. - uspješan napad na SHA-0, ali de na SHA-1
- 2004. - napad na MD4, MD5, SHA-0 i ostale, ali de i na SHA-1
- 2005. - uspješan napad na SHA-1

- obog proizvode 160 - bitni sažetek
- test/LSB/c/MJ - postri uo 0/1/10/0
- podjela na blokove ista kao i kod MD5 (16 32-bitna bloka)
- padding isti kao kod MD5
- sažetak od 5 32-bitnih konstanti (umesto 4 kod MD5):  $A_0, B_0, C_0, D_0, E_0$  (prve četiri se razlikuju od MD5)
- funkcije uisu (uzivo) iste kao i kod MD5 ali ponovo koristi osm. logičke funkcije (AND, OR, NOT, XOR ...)
- ovdje se fja koristi u 20 koraka (umesto 16 prije) zbog većeg sažetka; opet 4 stupnja, ali uključuju 80 koraka
- primjer u podacima koji se koriste iz blokova
  - u svakom od koraka se koristi drugi podatak (razliku u odnosu na MD5 - tamo je u prvoj koraci koristen  $M_1$ , druga 4  $M_2$  itd.)
  - u prvih 16 se koristi  $M_i$ , u novim 16-tog koraka klijentnica (XOR sastava rijeci)
  - ovaj XOR se rotira za 1 bit u slucaju SHA-1 time se dice mala sigurnost (SHA-1 se rotira i zove zbog rotacije sa 1 bit)
  - razlike u odnosu na SHA-0
- koraci su sljedeci:  $490$ ;  $490$ ;  $490$ ;  $490$ ;  $A$  se razvija počevši  $W_0, K_0$  (isto kao i kod MD5)
- hash je konkatenacija  $S = A_{80} B_{80} C_{80} D_{80} E_{80}$ , gdje 80 označava rezultat završnog koraka

→ Vazne svojstva hash funkcija

- tri osnovna i jedan dodatni

- (1) Oporust na izračunavanje originala (prva domaćinska oporust; engl. preimage resistance)
- funkcija hashiranja je:

$$H = h(M)$$

sazetak

poruka je jedno užaz (konec su dva algoritma)

- sivojstvo kaže da ne postoji fja  $h^{-1}$  t.d.

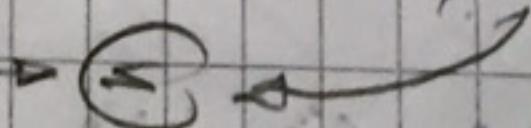
$$M = h'(H)$$

- da postoji takva funkcija ne bi bilo potrebe da veliki skladostima podataka, odnosno sivojstvo je dobro za velike podatke
- ipak, za upr. hashiranje lipica (128-bitnu lipicu u 160-bitnu poruku) bi se takvo čuo moglo, ali razmatrane fje pokrivaju i takav slučaj

- (2) Oporust na izračunavanje poruke koja daje isti sazetak (druga domaćinska oporust; engl. second preimage resistance)

- poznata poruka  $M$  saznati kime i hash  $H$
- ne moguće je pronaći drugu poruku  $M'$  koja daje isti hash  $H$

$$M, H = h(M) \quad \text{ne moguće } M', H = h(M')$$



- Čuđevo pitanje, koliko postoji istih poruka koje daju isti hash? Odgovor - beskonačno mnogo

- dva sasige je rupe organizirane oko svog centra
- (3.) odgovorst na koliziju (engl. collision resistance)
- neusvojive je poruke druge poruke  $M_1$ ,  $M_2$  za koje se dolazi u isti sažetak

$$M_1, M_2 \text{ t.d. } h(M_1) = h(M_2)$$

- SHA-0, SHA-1 nemaju ovo svojstvo
- razlika između (2) i (3)

- (2.) - zadana poruka; hash

- (3.) - mista nije zadana

- (2.) - napad  $2^{160}$ , (3.) - napad  $2^{80}$

## (4.) Difuzija

- svelca, pa; organizirana raspodjeljenja ulaznih podataka rezultira velikom i naištegled sluzajnom propusnjom u sažetku
  - drastična promjena hasha uslijed male promjene poruke
  - broj hashova je ograničen, ali broj poruka nije; stoga je beskonačno moglo poruka koje dan isti sažetak
  - za  $2^{160}$  poruka moraju postojati barem 2 ista hasha
  - redundansi napad - potrebno je  $2^{80}$  hashova (analogija s druge osobe koje mogu redudans na isti dan - isti proračun)
  - (3.) - no podrazumijeva neprćan redundansi napad, već 1 logičku druga napade koji daljuje suvremenim dozvlasti
- 860 ako nadamo  $M_1, M_2$  t.d.  $h(M_1) = h(M_2)$ ?

- odnosno koga koristi od collision resistance svojstva?
- problem izuzene ugovore (čista kriptografija) je bila napad na (2.) - nemoguće
- (3.) - problem certifikata
  - svaki certifikat sadrži vrijednost - moguće je preusmjeriti samo vrijednost tako da učini 'stog ostane neprouzvjetan u adresi na original - tada predstavljaće postope moguće
  - bolja proba (3.) - porečati hash (SHA-2), uklanjajući mogućnosti dodatnih napada (nadeudajući uvjete postopa)
  - SHA-0 i SHA-1 - kroz vremeno se sve više smatraju prekršljivima
  - 2007. - NIST raspisuje natječaj za SHA-3
    - u međarasporedu se prepričava SHA-2 (NSA)

→ SHA-2

- prepravak SHA-1: osnovno je porečana relativa saschka (224, 256, 384, 512-bitni skupetci)
- osnake poput SHA-224
- 224 i 256-bitni optimizirani za 32-bitnu arh.
- 384 i 512-bitni optimizirani za 64-bitnu arh.
- u 64-bit arhitekturi nema sumnje konstituujici hash; u 32-bit verzi ušporava
- 256-bitna ne zvuči puno, ali je embedded softver je svaki bit bitan - uvođi se 128-bitna verzija
- broj rundi - 64 za 224/256, 80 za 384/512

- funkcije: SHA-1 +, and, or, xor, rot; SHA-2  
koristi ; shift

- razlika između rot i shift - shiftom se gube bitovi!

- razlika u broju konstanti (blok je S12 za 824/055,  
odnosno 1024. za 384/512)

- koristi istu SHA-2 uz S12-om: sačinje  
jez označen SHA-S12 mada ovi i SHA-2 i  
SHA-3

- razlika SHA-1 i SHA-0 je u rotaciji za 1 bit  
(dovoljno da odgovori!)

- novi specijalni SHA-2 - koristi se funkcije za  
proračun MD5

- 156-bitne sačetale - 8 konstanti (il. 32-bitne, 8  
bitne)

- pouzno je jednako princip - (steo) sve konstante  
su poznavajuće, samo jedna sadrži komplikacije

→ SHA-3

- 2.10.2012. - poljedulik NIST-ovog natjecanja je Krocak (autor Daemen)

- potpisno drukčiji algoritam od dosadašnjih (drug  
finašti su bili temeljeni na AES-u)

- padding oblicem i, konkretnije, izmjenjuju  
na način da se dodaje 1000...001 / 1 na  
početku i kraju, između 0)

- znati ali je verzija paddinga! (i predloženu i  
prihvaccnu)

- spužvasta konstrukcija algoritma:

BLIC

- faza upravlja i faza cijedanja

- dio upravlja i po veličini cijedanje je sedam jedini brojaci

- cijedanje je rezultat zadnje iteracije (nao  
i kod desadašnjih algoritama)

- prvi je algoritma:

- učinak tečest se dijeli na dva dijela:  $c$  i  $r$   
(content i residualum)

BLIC - stane u 3D - ploča  $5 \times 5$  duljine koja odgovara  
duljini slike

- Ort - jedinicna kočka

- row -  $1 \times 5$ , column -  $5 \times 1$ , same ide u =  
dimenziji

- pet osnovnih funkcija:  $\theta$ ,  $\rho$ ,  $\pi$ ,  $\chi$ , i

- u svakoj ruci (kočici) algoritma se koristi  
svoj 5 funkcija (redom, nizce, i paralelno)

- tko pet funkcija su slike nao i propse  
ab predstavljaju transformacije ortova u  
tri dimenzije

BLIC

## ASIMETRICKI KRIPTOSUSTAVI

- asimetrični kriptosustavi - napotnja ideja kriptosustava
- Cezarova sifra - poslali su 3 slova, dekriptiranje poslali su 3 u suprotnom smjeru
- asimetrični - ne mogu se pouzdanuti i supotrivali

### Simetrično kriptiranje

np. AES u CBC modu

A

$k_{AB}$

m

$$c = E(m, k_{AB})$$

B

$k_{AB}$

$m' = D(c, k_{AB})$

→ zadovoljivo simetrično  
kriptiranje  
(bezveštice)

- dva eksperta koja zadovoljavaju sustav:

- korelirnost

- dobrota (sigurnost)

- korelirnost

- ako je poslani m uz  $k_{AB}$  dekriptiranje uz  $k_{AB}$  mora rezultirati istom porukom

$$m = D(E(m, k_{AB}), k_{AB})$$

- dobrota (sigurnost):

- definirana ranije

- problem: raspoznaća klijenca, kako voditi briga

• mogućim parovima klijenca

## Ašumetskius kriptiranje

A

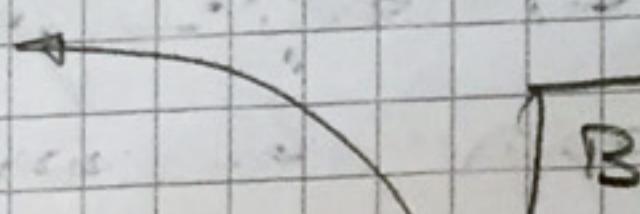
P<sub>B</sub>

m

$$c = E(m, P_B) \xrightarrow{e} m' - D(c, S_B) : \boxed{m' = m}$$

B

|P<sub>B</sub>|, S<sub>B</sub>



- nova prethodnog dogovora o klijentu
- B generira par klijentera P<sub>B</sub>, S<sub>B</sub> (javi se i privatni)
- P<sub>B</sub> svr smjer zvati
- doliva ga A, kriptira i salje B
- dekripcija privatnog klijentera S<sub>B</sub>
- da padač kopijima parni klijent ne može saznati informaciju o poruci
- čini se nemoguće, ali je ipak izvedivo
- kol se dojaviti upri:
  - "Mirko je prepisao labos"
  - komunikacija kopja je jasna samo osome komu je poruka namijenjena
- povijest:
  - 1976: Diff, Hellman - moguće postojanje ovakvog sustava
  - 1977: RSA - MIT
  - 1997: ipak je RSA otvoreno 1973. (C. Cocks)
  - vratna kočija knjige

definicija Sustav kryptiranja javnog ključa sa rečima  
algoritama  $(G, E, D)$  gdje je  $G$  randomizator  
algoritam generiranja ključeva  
 $G() \rightarrow (p, s)$

$$G() \rightarrow (\rho_k, s_k)$$

Tje faloder rando mit rani algoritam ujpravaq

$$E(u, p_e)$$

D je algoritam dekriptiranja

$$D(c, s_k)$$

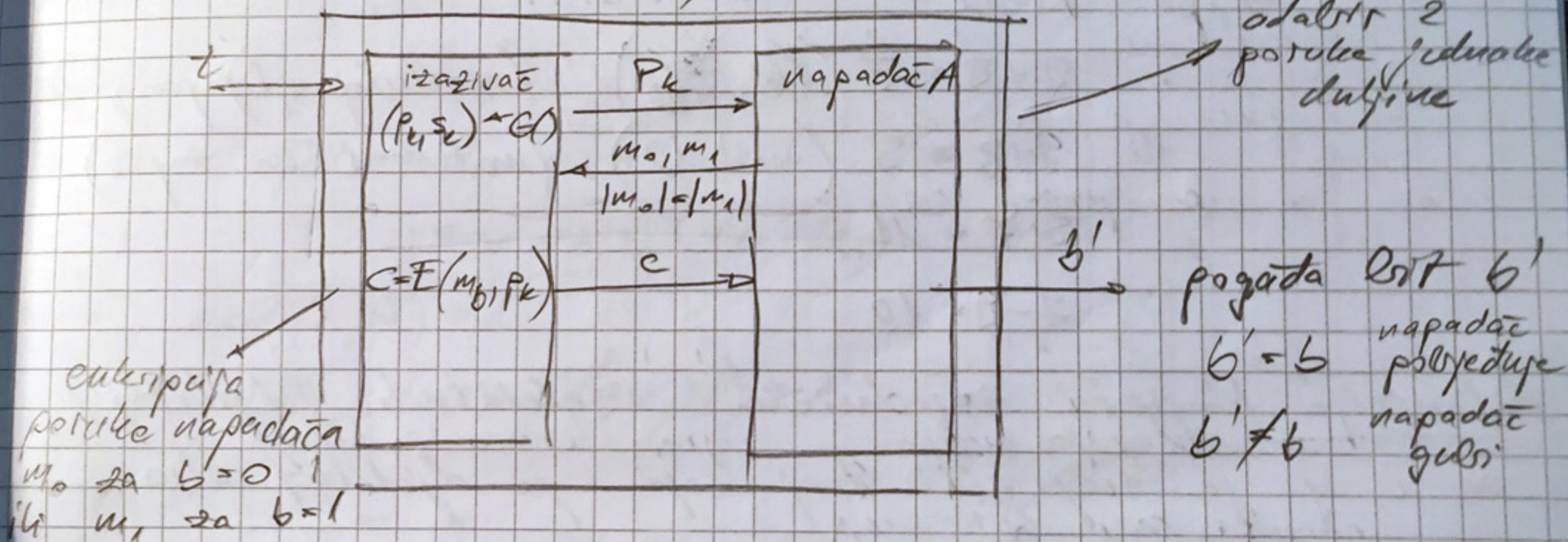
talors da vrispedi

$$D(E(u, p_k), s_k) = m \quad \text{from } t(p_k, s_k) \leftarrow G$$

Sigurðarst:

~~Expt~~(b)

→ eksperiment



- sustar y c. siguran also napadač ne može poljpedati  
(osim rasunivo), tj. sustav je siguran ako je  
bacanje novčića vačka strategija

definicja konsystencji ( $G, \emptyset \vdash D$ ) w semanticzny sposób  
albo w swącej epikratycznej algorytmie A wojedynie.

$$| P[\text{EXP}(0)=1] - P[\text{EXP}(1)=1] | \text{ je tauemarivs}$$

Korolaris uverospatnos polypode il. poraz napadaca je  
(skoro pa) jednaka

- RSA ("olsorci") nije siguran algoritam (nije semantički siguran sastav javnim ključem)
  - ali se pomoću neke može napraviti siguran
  - treba dodati određene stvari kako bi postao siguran

### TEORIJA BROJEVA

- notacija:
  - $N$ -prodaci broj
  - $p, q$  - prosti brojevi
  - $\mathbb{Z}_N = \{0, 1, 2, \dots, N-1\}$  - prostani cijeli brojevi modulo  $N$  uz operacije  $+, -, \times$
  - npr.  $N=12$

$$9+8=5 \quad (\text{u } \mathbb{Z}_{12}) \quad (\text{modulo zbrajanje})$$

$$\text{li } 9+8=5 \quad (\text{mod } 12) \quad (\text{matematički zapis})$$

$$5 \cdot 7 = 11$$

$$7-9=10$$

definicija - Najveći zajednički učestalnik (djelitelj)

$x, y$  je najveći broj koji je djelitelj oba broja ( $\text{nzd}(x, y), \text{gcd}$ )

$\text{nzd}(x, y) = 1 \Rightarrow x, y$  relativno prosti

- tražimo  $\text{nzd}(x, y)$ :

teorem Euclidov algoritam

$\forall x, y \in \mathbb{Z} \quad \exists a, b \in \mathbb{Z} \quad \text{t.d.} :$

$$ax + by = \text{nzd}(x, y)$$

Dodatako, postoji efikasan? algoritam kojim proučavati  $a, b$  i  $\text{nzd}(x, y)$  - poštano Euclidov algoritam.

- cifrasan - ali za  $x \in \mathbb{Z}$  u-om brojem doda  
prostimi - Euklidski algoritam (ma slожnost  $O(n^2)$ )

→ Modularni inverz

- problem: dajanje  $x \in \mathbb{Z}_N$

definicija Inverz od  $x \in \mathbb{Z}_N$  je element  $y \in \mathbb{Z}_N$  t.d.

$$x \cdot y = 1 \quad (\text{u } \mathbb{Z}_N)$$

- primjerice!

$$2^{-1} = 8 \quad (\text{u } \mathbb{Z}_{17})$$

$4^{-1}$  ne postoji u  $\mathbb{Z}_{10}$

teorem  $x$  ima inverz u  $\mathbb{Z}_N \iff \text{nsd}(x, N) = 1$ .

Dokaz

⇒  $\exists a, b$  t.d.  $ax + bN = 1$

$$\Rightarrow ax = 1 \quad (\text{u } \mathbb{Z}_N)$$

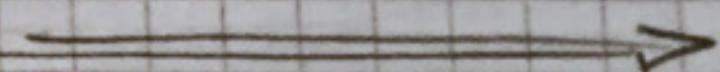
⇒  $\text{nsd}(x, N) = 1$

$a \in \mathbb{Z}_N \quad \text{nsd}(ax, N) = 1$

Q.E.D.

definicija  $\mathbb{Z}_N^* = \{x \in \mathbb{Z}_N : \text{nsd}(x, N) = 1\}$  (skup svih  
elementata koji imaju inverz u  $\mathbb{Z}_N$ )

- nisu svih brojevi u  $\mathbb{Z}_N$  invertibilni



- lako - lagam, ali za dodatne 3 Rada treba raditi

- može se izračunati s radom, ali nedostaje Digitalni  
potpis - online

- pogledati upute:
    - dati su izbor DES ili AES i sljede
    - treba napraviti snocje kog omogucava kompjutaciju, deljenju i sljede
    - Ovis je pogodnosti snocje koristenu za generiranje ključeva
  - 2. liders
  - 3. liders - napraviti neli algoritam it pogodujuog skupa
    - ostvariti ga i spozetka
    - treba postojati mogućnost odabira vlastitog ili Pythonovog AES-a, ali tada rezultat treba biti identičan
- 

- uveo nije realizirao asimetrični kryptosustav temeljen na S-tablema, XOR-ovima i sl.
  - učka  $v \in \mathbb{F}$  prost broj,  $\mathbb{F}^*$  je
 
$$\mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}$$

$$|\mathbb{Z}_p^*| = p-1 \quad \text{broj elemenata}$$

$$|\mathbb{Z}_{15}^*| = 8 \quad \text{zasto}$$
- 

### teorem (Fermat)

$x \in \mathbb{Z}_p^*$ , gdje  $p$  je prost broj vrijedi:

$$x^{p-1} = 1 \quad (u \mathbb{Z}_p)$$

- Korolar:

$$x^{p-1} = 1$$

$$x^{p-2} x = 1$$

$$\Rightarrow x^{p-2} = x^{-1} \quad (u \mathbb{Z}_p)$$

Teoreum (Eulerov teoreum)

$\mathbb{Z}_p^*$  je cikločka grupe, tj: postoji element  $g \in \mathbb{Z}_p^*$  t.d.:  
 $\{1, g, g^2, g^3, g^4, \dots, g^{p-2}\} = \mathbb{Z}_p^*$

Takav element  $g$  se naziva generator grupe.

upr.  $p=7$   $g=3$   $\{1, 3, 9, 2, 4, 5\}$

pa je 3 generator grupe  $\mathbb{Z}_7$

$g=2$   $\{1, 2, 4, 1, 2, 4\} \rightarrow$  nije generator

Acl  $\text{ord}_p(g) = |\{1, g, g^2, \dots\}|$ , tj: cardinalitet grupe.

definicija  $\varphi(n) = |\mathbb{Z}_n^*|$  red uduale svake elementarne grupe  $\mathbb{Z}_n$  koja su rel. prosti s n.

$$\varphi(p) = p-1$$

$$\varphi(15) = 8$$

- upravi:

$$N = p \cdot q, \quad p \text{ i } q \text{ prosti}$$

$$\varphi(N) = N - 1 - (q-1) - (p-1) =$$

$$= N - p - q + 1 = \frac{(p-1)(q-1)}{1}$$

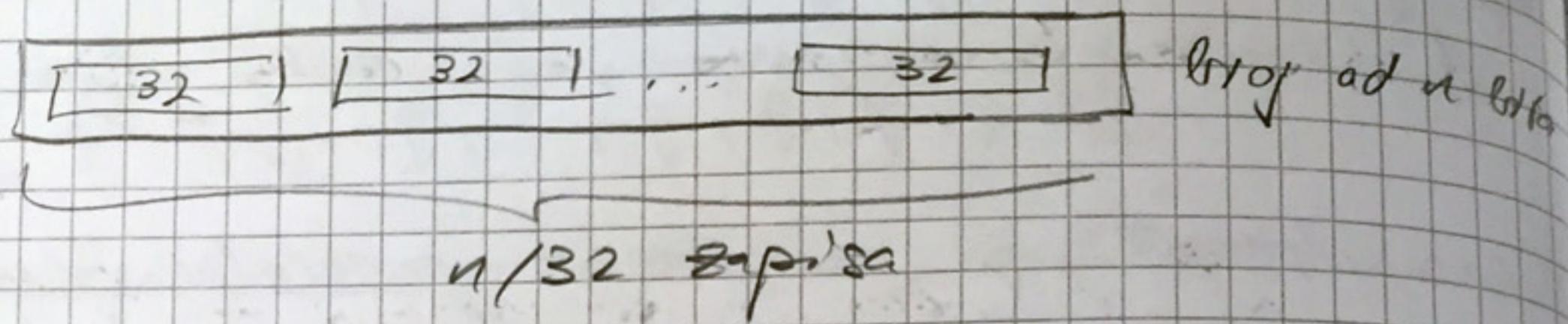
Teoreum (Eulerov teoreum - poopćenje Fermatovog)

$$\forall x \in \mathbb{Z}_N^* \quad x^{\varphi(N)} = 1 \quad (\text{u } \mathbb{Z}_N)$$

(Fermat je specijalni slučaj jer je  $\varphi(p) = p-1$ )

## Reprezentacija velikih brojeva

- ideja:



- zapisi ovim 64-bitniim blokima izbjegavamo preljeva

1. Zbrojanje:  $\mathcal{O}(n)$

2. Množenje:  $\mathcal{O}(n^2) / \mathcal{O}(n^{1.58}) / \mathcal{O}(n \log n)$

- množenje i množenje: Karatsuba  $\mathcal{O}(n^{1.58})$ ; ideja:

$$\boxed{x_2 \quad | \quad x_1}$$

$$\boxed{y_2 \quad | \quad y_1}$$

$$(2^6 x_2 + x_1) (2^5 y_2 + y_1)$$

- kako izvesti uz samo tri množenja?

-  $\mathcal{O}(n \log n)$  - FFT

- problem: konstantni faktori su ogromni, postupak je učinkovit (cca. 10<sup>5</sup> bitova)

3. Dijeljenje s ostatkom  $\mathcal{O}(n^2)$

- dva koraka: traženje inverzni i množenje

4. Potenciranje:  $\mathcal{O}(n^3)$

- N - n-bitni broj

-  $x^y \in \mathbb{Z}_N$

$$x^y \pmod{N} = ?$$

- algoritam nestopnog kvadriranja:

17 <sup>256</sup>  
module 19

$$(((17^2)^2)^2)^2 \Rightarrow \log_2 256 = 8 \text{ množenja}$$

$$x^{53} = x^1 \cdot x^4 \cdot x^{16} \cdot x^{32}$$

$$53 = (110101)_2 = 1 + 1 \cdot 4 + 1 \cdot 16 + 1 \cdot 32$$

- užastopn. kvadratrange za potencje celika  $2^n$

RSA

$\approx 1024$  bit

1.  $p, q$  sl. veliki prosti brojevi

2.  $N = pq$ ,  $\varphi(N) = (p-1)(q-1)$ ,  $N \approx 2048$  bit

3.  $e$ ,  $n \text{ mod } \varphi(N) = 1$ , ( $e = 65537$ )

4.  $d = e^{-1} \text{ mod } \varphi(N)$       ↳ brusne  
100...001

-  $d$  nije presan;  $\varphi(N)$  je pravougliv

- generirati:

$$pk = (e, N)$$

$$sk = (d, N)$$

sljedeći put

[G]

- prepostavimo  $m \in \mathbb{Z}_N$

- izracunava se (euklipsija):

$$c = m^e \quad (\text{u } \mathbb{Z}_N), \quad e - \text{javi eksponent}$$

[E]

- dekripcija:

$$m' = c^d \quad (\text{u } \mathbb{Z}_N), \quad d - \text{tajni eksponent}$$

- ispravnost:

$$(m^e)^d = m^{ed} \quad | \quad ed = 1 \quad (\text{u } \mathbb{Z}_{\varphi(N)})$$

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

$$\Rightarrow m^{1+k\varphi(N)} = m \cdot (m^{\varphi(N)})^k = m \cdot 1^k = m \quad (\text{u } \mathbb{Z}_N)$$

l u  $\mathbb{Z}_N$  prema Eulerovom teoremu

čime je dokazana ispravnost kryptosustava

- siguran je jer nije poznat nadim razlozanje
- kad bi mogli raspaliti  $N$  na prosti faktore  
doljni bi se pao iz dega sljedici  $\varphi(N)$   
na samom tome i c' id
- ne postoji efikasan algoritam za razlozanje  
tako velikih brojeva na prosti faktore
- zašto RSA nije securan?
- bezog determinističkog postupka je moguće  
uz poznavanje konteksta poruke pogodati
- kod AES-a ne vrijedi napad nemoguće, dok kod RSA napad može dobiti vremena klijenata
- opisan sustav je skoro jedini primjer kada je  
lopuća dama poznat
- još jedan mogući napad:
- pomoću RSA se salje 64-bitni klijent (stegan)

$k$ , 64-bitni klijent

$$c = E(k, p_k) = k^e \pmod{N}$$

gruba sota:  $2^{64}$  koraka

meet-in-the-middle napad:

- pp da je:

$k = k_1 k_2$  i  $k_1$  i  $k_2$  32-bitni brojevi

↳ → vjerojatnost 2%

(preveliko za kriptosustav)

- tada vrijedi:

$$c = k^e = (k_1 \cdot k_2)^e$$

$$\frac{c}{k_1^e} = k_2^e \pmod{N}$$

- tablica veličine  $2^{32}$ ,

$2^{32}$	$k_1$	$k_2$

- ovakav napad ✓  
nugac 2 lošog ↓

jer je  $k_1$  32-bitni broj (dijeli se na  
množenje s množenjem inverzom)

- potrebito  $2^{32}$  koraka i uspa memorije

- izracunava se  $k_2$ ; traži odgovarajući  $k_1$   
u tablici; postupak završava kada se  
prouđe par  $(k_1, k_2)$

- složenost napada je  $2^{40}$  koraka, odnosno  
puno je bolji od brute force napada

- zahtijevale su konstanti RSA ovisno o dliku  
(uz par iznimki)

→ RSA u praktici (ISO...)

- (G, E, D) - RSA

- (E<sub>s</sub>, D<sub>s</sub>) - simetrična enkrptacija = integritetom, upr.  
AES-HMAC - SHA256

- H - hash funkcija, upr. SHA256

- ideja:

$E(k, p_k)$	$E_s(m, k)$
-------------	-------------

$x \leftarrow$  slučajni element  $\mathbb{Z}_N$

$y = E(x, p_k)$

$k \leftarrow H(x)$

$c \leftarrow E_s(m, k)$

→ Šalje se par  
(y, c)



- dekripcija:

$$x = D(y, s_k) \quad RSA \text{ dekripcija}$$

$$k = H(x)$$

$$m = D_s(c, k)$$

- RSA - kripto sustav zasnovan na teoriji brojeva

- u praktici nije dobro da je m kod RSA poruka, jer se sama oliscavan broj je kog je se hasi funkcionom izvođi. Upravo kog je se cifrica poruka

- najbolji napad krenjen na nastava u prosti faktore

- AES - 100 B - 83 procesorska ciklusa

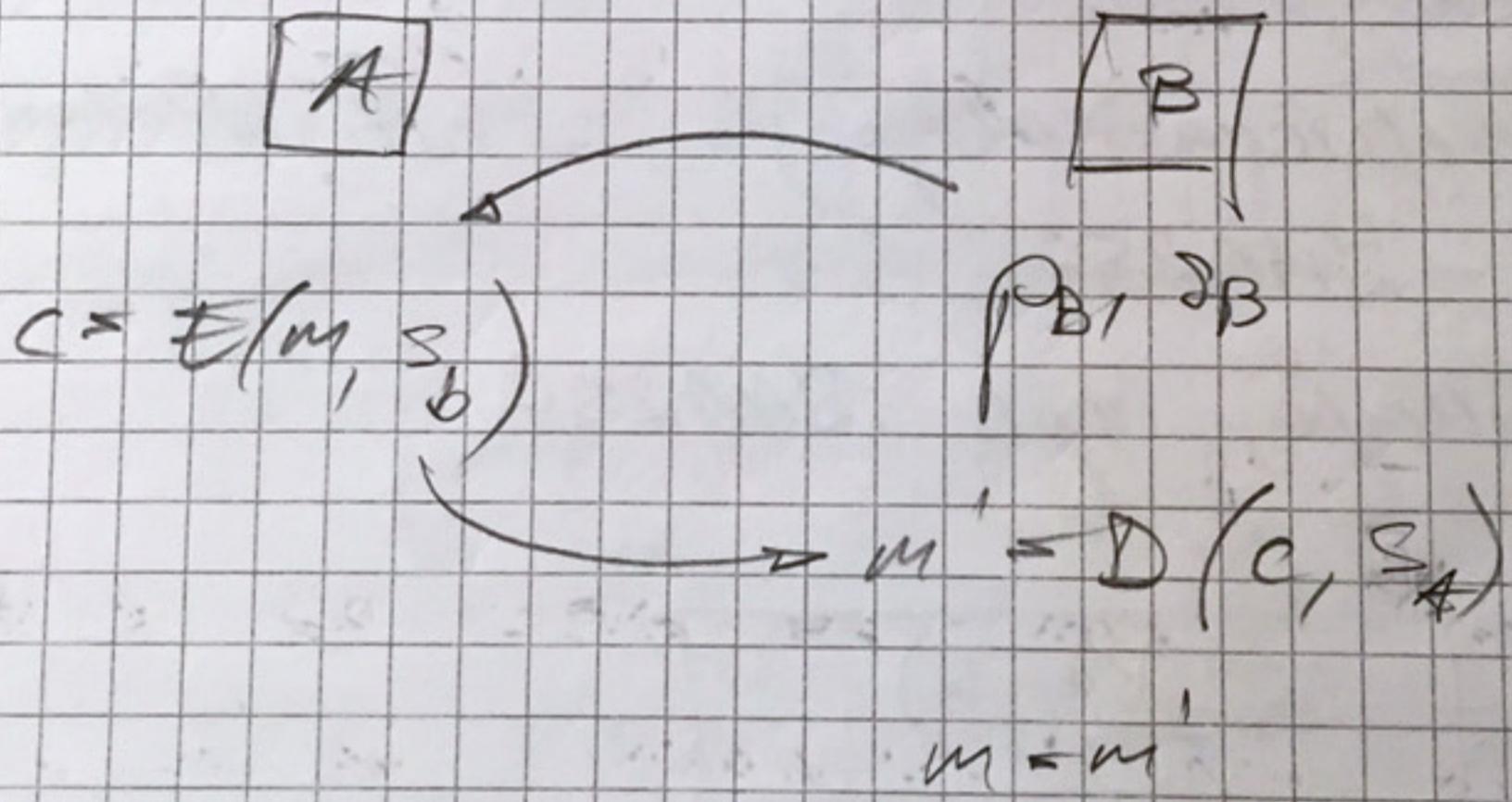
RSA - 1024 b -  $\approx 2 \cdot 10^8$  procesorska ciklusa

- simetrična je znatno brza

- veličina ključa govori o granici uspešnosti napada, same po sebi nista ne znaju

- granica vrpolnosti - en kriptiranje

nacijon MI:



G

- p i q veliki su, prosti brojevi ~ 1024 bit

$$N = pq, \varphi(N) = (p-1)(q-1)$$

$$e, \text{mod}(e, \varphi(N)) = 1, e = 65537 \text{ (veliko fiksiran)}$$

$$d = e^{-1} \text{ mod } \varphi(N)$$

- $P(e, N)$  - jarić klijent  $S(d, N)$  - tajni ključ
- način je da je  $N$  testo rastaviti na proste faktore

E

- $m^e \bmod N$  sifriranje

D

- $m^d \bmod N$  desifriranje

- RSA kriptosustav je siguran jer je  $\varphi(N)$  velika u sklopu uvođenja uvođenja
- dokazati da je RSA siguran  $\iff N$  jest do rastavljivo
- no proste faktore

Zadatak 5: modulacija

$$m_A \sim 10^{12}, (3, N)$$

$$E(m_A, P) = m_A^3 \bmod N$$

- druga možina: jedan je deterministički RSA, tj.  
velike isprobati  $60^{th}$  vrijednosti (nije puno!)

$$c_A = m_A^2 \bmod N$$

$$(c_A, N) \rightarrow m_A$$

- ovaj način ne je rabi za AES jer je tajni ključ klijent, odnosno neće biti grubac slobodan ovog tipa niti moguće kod simetričnih kriptosustava
- drugi način: izvlači se preček konjuna iz  $m_A$  funkcionalna je  $m_A$  mala, tj.

$$m_A^3 \bmod N = m_A^3$$

odnosno RSA nije siguran - to malu poruku i

malou e

- postupak jednog muog napada na RSA zbog dega se nikad ne koristi za kriptiranje poruke
- drugi dio zadatka:

$$C_A = S_A \oplus K \quad \rightarrow \quad K = C_A \oplus S_A$$
$$C_B = S_B \oplus K$$

XOR je idempotentna operacija

- jednokratna vrijednost se naziva jednokratnom jer se isti uljepši smije koristiti samo jednom
- druga vrsta napada na jednokratna vrijednost, tj. postane su dvije poruke

$$C_A = S_A \oplus K$$

$$C_B = S_B \oplus K$$

$$\rightarrow C_A \oplus C_B = S_A \oplus S_B \quad (\text{za dojamu zadacu?})$$

- uz informaciju o  $S_A$  i  $S_B$  može ih se otkriti

- 
- za labos skriptu pyCrypto

- enkripcija je ujedno i de RSA, složenost  $O(n^3)$

- velina se je  $e=65537$  koji je binarnom zapisu ima oblik 1000...001, tj. za enkripciju faktor je potreban 17 množenja

- RSA.generate, RSA.construct

- ne koristi se safer Nefran na jazilici koju je deterministički

- kako generirati velike slučajne proste brojeve p. i.g?  
osnovni algoritam:

gen. prost(n) // n - br. emitova

```
while (true)
    p = sl. broj;
    if (prost(p))
        return p;
```

- frekvencija pojavljivanja prostih brojeva, f:  
funkcija gustoće prostih brojeva

$$f(n) \approx \frac{1}{\ln(n)}$$

f: koliko ima prostih brojeva manjih od n  
- prost(p) - raznjava li taj broj na faktore?

nije učinkovit  
- učinkoviti algoritmi i polinomijalni  
deterministički, učim raditi s učinkovit-  
stvom;

prost(p):

0 onda je p sigurno složen  
1 onda je p vrlo učinkovito prost

prost(p)

pocinj u puta

```
a = slučajni broj < p
if slijedeci za-složenost(p, a)
    return 0;
return 1;
```

- ako idući broj slijedila je složenost, ipr  
 $\frac{1}{2}$  br.  $< p \Rightarrow$  učinkovit broj  
rezultata je  $\left(\frac{1}{2}\right)^k \Rightarrow$

- Fermat:

$$a^{p-1} \equiv 1 \pmod{p}$$

$\Rightarrow$  svjedok Fermat ( $p, a$ )

if  $a^{p-1} \pmod{p} \neq 1$

return 1

return 0;

- u praksi "spada da je svjedok Fermat odličan" za veliku većinu brojeva; problem su Carmichaelovi brojevi (kojih je, načinju, mno)
- drugi način: rješenje jednadžbe  $x^2 = 1$

$$x^2 = 1 \pmod{p} \rightarrow \text{u } \mathbb{Z}_p \text{ jednadžba } (x-1)(x+1) = 0 \text{ ima samo dva rješenja}$$

- neprimjekov rješenje jednadžbe je svjedok za složnost

- potpun algoritam:

$$a^{p-1} \equiv 1 \pmod{p}$$

$$(a^{\frac{p-1}{2}})^2 \equiv 1 \pmod{p}$$

- 1 - nastavljamo dalje, -1 prestaje algoritam

$$(a^{\frac{p-1}{4}})^2 \equiv 1 \pmod{p}$$

- za proste  $p$ :  $a^{\frac{p-1}{4}} = 1$

-  $x^2 = 1$  samo triv. rješenja

$$b = p-1 = 2^s d / 2^{s-1} d / 2^{s-2} d / \dots / 2^1 d / 1$$

$$a^b \pmod{p} = 1 \quad 1 \quad 1 \quad -1 \quad x \quad \dots$$

$\pm 1$



ako se pojavlji prije -1 postoji

svjedok za složnost

- sumarno: traži se vetrinjsko ( $\neq \pm 1$ ) prekucje  
 $x^2 = 1$

### teorem Bez dokaza

Vise od polovice brojeva a su razpredeljeni u sluzbenost  
gorajim testom ( $a=p$ ), tj. po Miller-Rabinu  
da je p slozen.

- verovatnost da prost broj nije dekomponisan rese-  
nanim algoritmom je zanevarena

- velika broj ceta odabran slozeni lepci p i q (tj. nisu  
prosti) sto ce se dogodi?

- ne vrijedi  $\varphi(n) = (p-1)(q-1)$  pa stoga  
ekriptacija, dekriptacija ne bi bili inversni  
postupci, tj. korektnost RSA ne postoji

- moze li se provesti da je broj prost ponovo  
RSA? - test za primalnost?

- multipulne RSA

→ - enkripcija i dekripcija - provjeri?

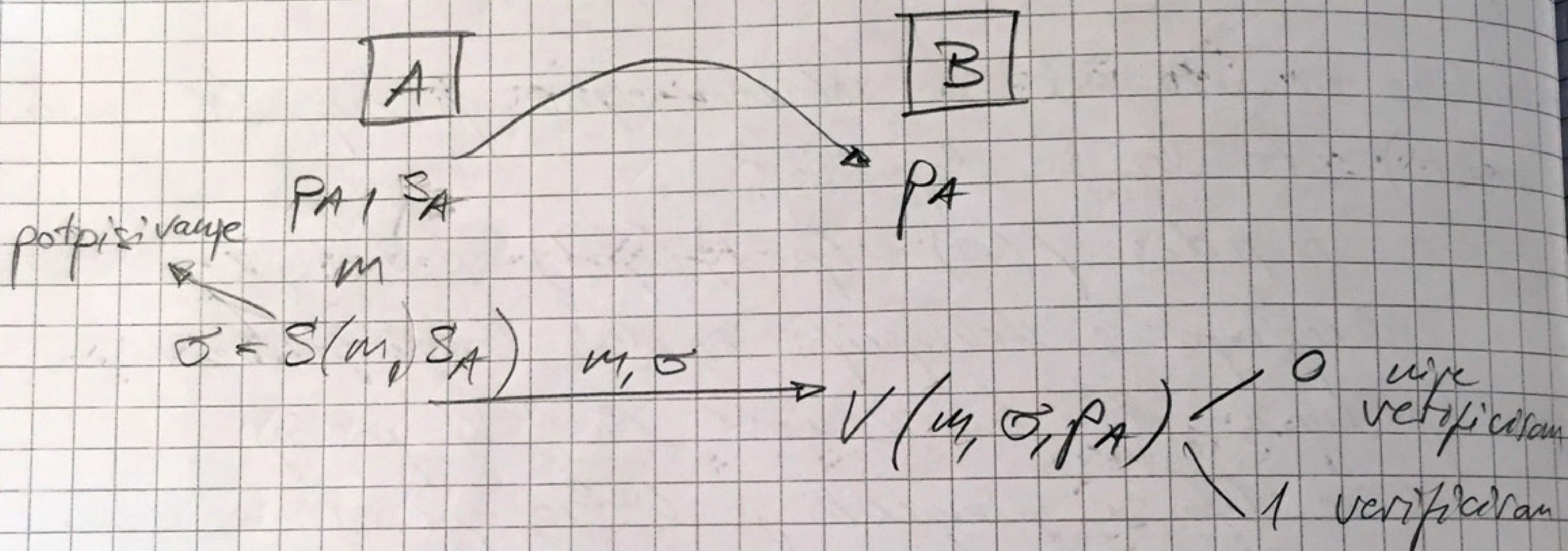
### Digitalna anotacija

nesigurnost za klijent ne koristi direktno, već  
uzgrijati? - pogledati

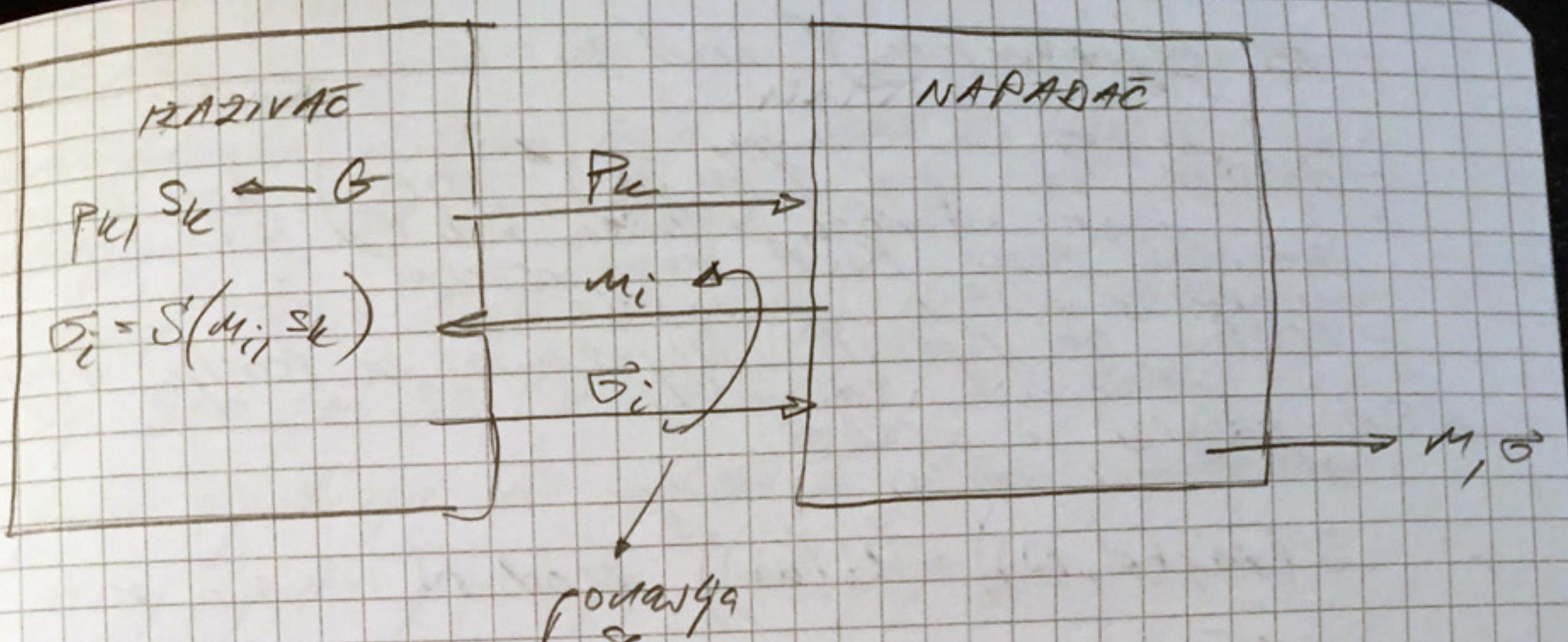
- Philip R. Zimmermann - izvor koda u formi knjige  
(sloboda govora)

## Digitalni potpis

- obican potpis ima sljedeća svojstva:
  - mogućnost prouzore (ponosu osoblje)
  - rezultost za dokument
  - nemogućnost liovitvorenja
  - nemogućnost policanja
- Želja: elektronička skeniranja sa skenom ugovornim
- e-mail, potpisivanje softvera...



- potpis je rezultat za poruku  $m$
- svojstva:
  - korektnost:  $V(m, S(m, s_A), p_A) = 1$
  - sigurnost
  - postoji puno korektnih potpisa, tako i  $V$  koji vrši vrlo mala 1 je korektan
  - sigurnost: test je generiran par  $(m, \sigma)$  bez da je poznat nijedan drugi



- napadac je uprješam da

$$V(m, \sigma, P_K) = 1, \quad m_i \neq m \quad \text{ti}$$

odnosno

$P(A \vee \neg \sigma_i)$  je zanemariva i vjerojatnost algoritma je

- obični RSA algoritam signiran

$$S(m, (d, N)) = m^d \bmod N$$

$$V(m, \sigma, (e, N)) = \sigma^e = m ? \quad 1:0$$

1. napad:  $(e, N)$

$\sigma - \sigma_i$  je je sl. broj

$$\begin{aligned} m &= \sigma^e \bmod N \\ \sigma &= m^d \bmod N \end{aligned} \quad \left\{ \begin{array}{l} (m, \sigma) \end{array} \right.$$

- takav par  $(m, \sigma)$  je dovoljan za proglašavanje sustava nesigurnim

2. napad:  $(e, N)$

$$m_1 \dots \sigma_1 = m_1^d \bmod N$$

$$m_2 \dots \sigma_2 = m_2^d \bmod N$$

$$\sigma_1 \cdot \sigma_2 = (m_1 \cdot m_2)^d \bmod N$$

- dobit će par  $(m_1, m_2), (\sigma_1, \sigma_2)$  koji u jeku snimka čini RSA nesigurnom
- treba se raspisati dovođenja iz teorije brojeva
- RSA potpis u praksi:

$$S(m, (d, N)) = (H(m))^d \bmod N$$

$$V(m, \sigma, (e, N)) = \sigma^e = H(m) ? 1:0$$

- $H$  je kriptografski sigurna hash funkcija  
za čiji rezultat ne može da se pogrešno o  
muženu i sljemo
- digitalni potpis osigurava:
  - autentičnost (samo uz prethodno poznat identitet)
  - integritet
  - neporeduost

### Digitalni potpis

- kombinacija digitalne omotnice i potpisa
- Python 2.7 i 3.3 uključuju default Crypto  
Library; instalirati za pyCrypto
  - alternativa - Anaconda
- pod Lab2 uploadati samo drugi labos

## → SHA-3

- sažeci jednake duljine kao i SHA-2
- padding - 10...01 do veličine bloka:
  - stane 3D:  $5 \times 5 \times$  duljina sažeci u binarnim  
A. 800 (za 32-bit racunalac), i.e. 1600 (+64)
  - faza upredjivanja de sadrži  $f_1$  i  $f_2$  se potrebno  
korijenimo ponavlja kod upredjavanja

$$LSW = C + r \rightarrow \text{residuum}$$

$\downarrow \quad \downarrow$   
 $S \times S$  kapacitet

-  $w$  je duljina "speci"

800      32-bit

$$LSW = C + r = 1600 \text{ (za 64-bitnu arh.)}$$

$$C = 2 \times \text{veličina sažetka} \quad (224/256/384/512)$$

- $r$  je rezim, uz prenavanje arhitekture, tako izračunati
- blok koji se hashira, redukuje se u dve  
kao i  $r$ , te se zadajući blok paddingom  
nadopunjuje do veličine  $r$
- embedded sustav - sažeci od brojnih entiteta  
majući u svakoj zadovoljavaju  $LSW = C + r$   
uz  $C = 2^l \cdot v \cdot s \otimes$
- Mr. korala algoritma =  $12 + 26$

$$2^l = w$$

- npr. za 64-bitnu arh.:

$$64 = 2^l \Rightarrow l = 6 \Rightarrow \# \text{korala} = 2^6$$

- SHA-3 konzisti pet osnovnih funkcija

- Ø - XOR određenih bitova
- P - rotacija bitova
- π - posmatrati redakciju (ne svih da jedan način; dodatne transformacije)
- 2 - obratan XOR
- X - NOT, AND ; XOR

### Preporuke

- najbolji simetrični - AES, u DES
- DES se stvarno uvoli slobog 3-DES-a (sporija od AES-a)
  - IDEA je u redu
- asimetrični - RSA
  - nedostatak - dugotrajan generisanje ključeva, pogotovo uz velike kreditne vrijednosti
  - brojera p i q (dodatako usporava)
  - eksploite krvulje - takvi kriptosustavi generiraju nesuvremene ključeve
    - ključevi su red veličine manji od RSA su jednako sigurni - manje memorije, brž rad
  - Elgamal - visoko prolažljiv
- hash - SHA-256 (SHA-2)
  - izbjegavati SHA (mali hash, redudanski napadovi)
  - FINA u svome kartice tek uvede SHA-2, tj. stariji algoritmi su i dalje u uporabi
- ključevi
  - simetrični - minimum 96, optimalno 128

(normale preporučuje 256)

- RSA - 1024 minimum (1024 niste sigurni), 4096 preporuka
- 1K i ma veću karticu, 2K tako u pogodno
- elliptičke - 192/256
- hash - minimalno 224, preporuka SHA-256

### SIGURNOSNI PROTOKOLI

#### Diffie-Hellmanov protokol

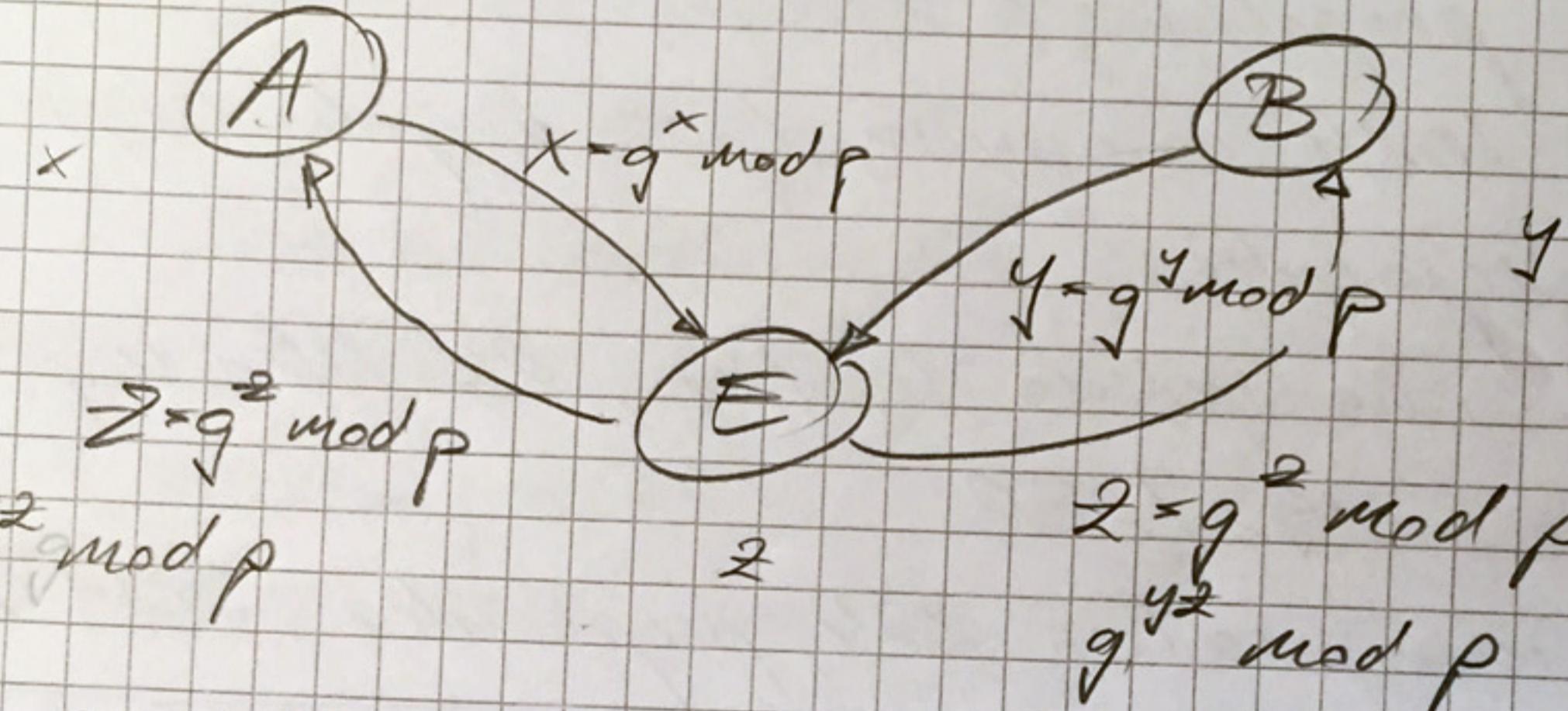
- SSH - prilikom logiranja na udaljeni računarski naredbe se salje kriptirani kanalom
- DH protokol služi za razmjenu ključa:

$$\begin{array}{ccc} \textcircled{A} & \xrightarrow{\quad X = g^x \bmod p \quad} & \textcircled{B} \\ \xrightarrow{\quad Y = g^y \bmod p \quad} & \end{array}$$

$\text{uzd}(g, n) = 1$

$$\begin{array}{ccc} x & & y \\ \downarrow & & \downarrow \\ p & & p \\ \hline g^x \bmod p & & g^y \bmod p \\ = g^{xy} \bmod p & & = g^{xy} \bmod p \end{array}$$

- javno se dogovaraju za dva broja  $n$  i  $g$ ;
- $n$  je prost broj
- odabira nasumični veliki broj  $x$ ,  $y$
- A računa  $X$  i salje B
- B računa  $Y$  i salje A
- A računa  $Y^x \bmod p$ , B računa  $X^y \bmod p$ .
- preostalo samo (javni) dogovor o broju bitova izljeđenja se RSA (tako je postupak siguran)
- ogromna mala - nije mala na manim i u middle napad (napad ovođen u sredini)



- E generira svoj broj  $z$ , izračunava  $z = g^z \text{ mod } p$  i salje A i B
- rezultat  $g^{xz} \text{ mod } p$  i  $g^{yz} \text{ mod } p \rightarrow$  proizit
- protiv man u sredini uobičajeno je da se koristi autentifikacija
- kako se certifikati (zadnji kript.) - zašto?
- u praksi odabrat će Diffie-Hellman uz obvezu autentifikaciju

- Diffie-Hellman - zabitjera dodatna autentifikacija
- u sudionika - koliko D-H ključeva treba generirati?
  - svaki od sudionika treba imati  $N-1$  ključ, pa je potreban

$$\frac{N(N-1)}{2}$$

odnosno vek duplicita

$$\frac{N(N-1)}{2}$$


---

Raspodjela ključeva u zaključenom simetričnom kryptosustavu

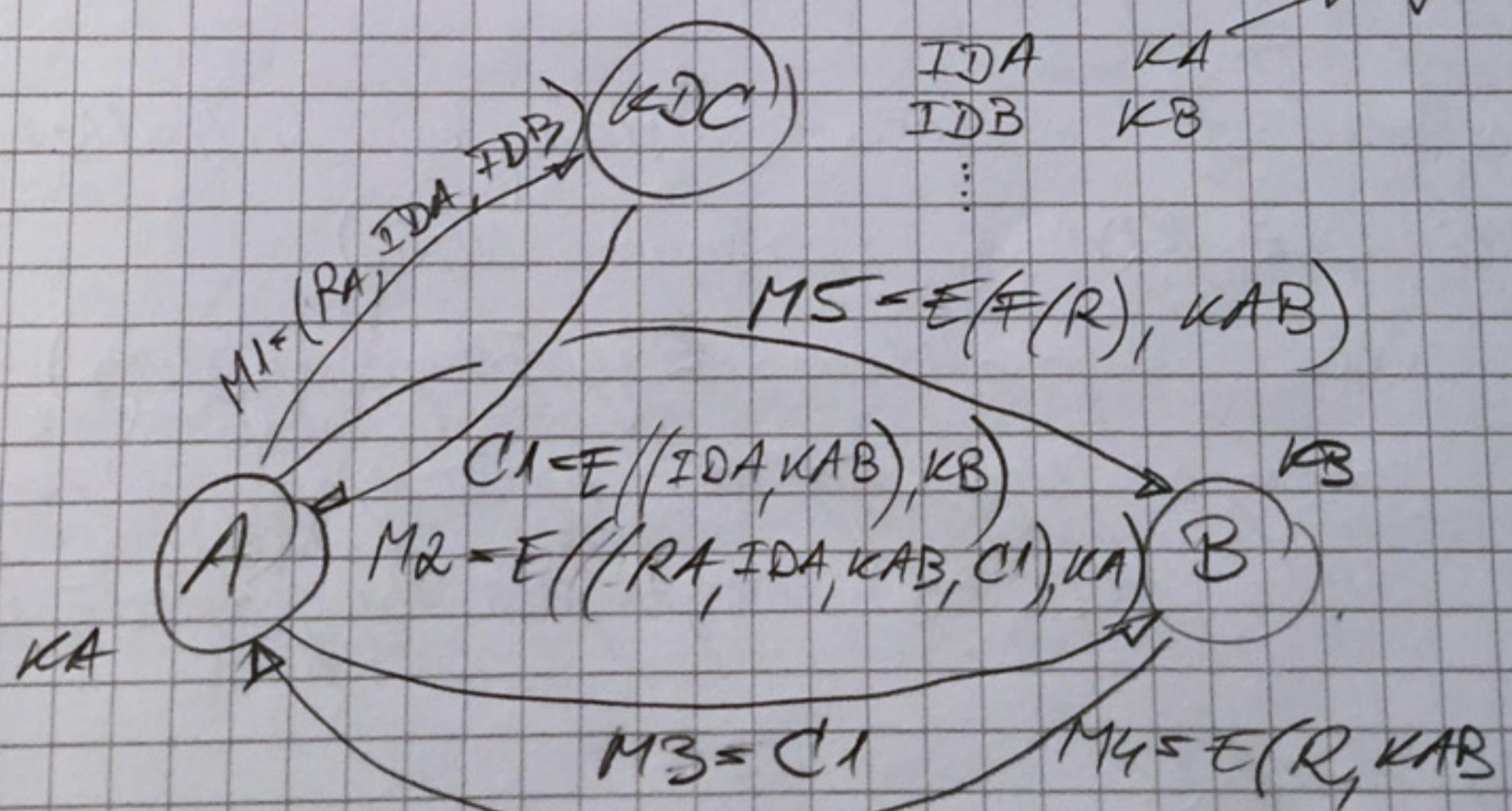
je danje problema provođenog poliranjenog broja ključeva jer suvremenim postupitim odlog se dolazi do ključeva

- dodatni problem - ključevi se mijenjaju na dnuvoj bazi

- KDC - Key Distribution Center ili centar za raspodjelu ključeva

- postupitim gara sve sudionice - zaključen sustav (broj sudionika se ne mijenja)

tajni ključ



- tajni ključ se ne mijenja prečesto te je broj ključeva  
ki jednake  $N$

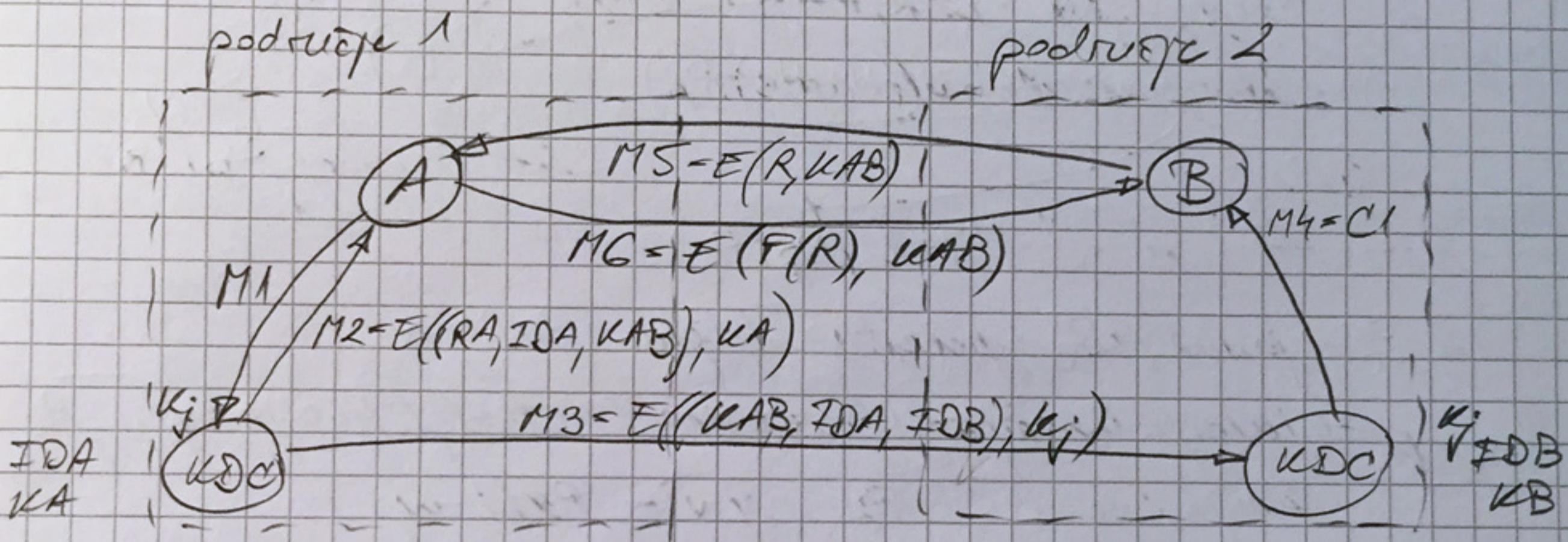
- A salje  $M_1$  KDC-u;  $P_A$  - tajni zaključevi; žoli razg.  
S B

- KDC proverava jesu li A i B na popisu
- KDC salje unštirane poruke  $C_1$ , sluci gov. kada se A i B komuniciraju između A i B koji je katalog uključi; šifrirano uključen KB
  - $C_1$  je dio poruke  $M_2$
- A raspalisa  $M_2$  i salje B poruku  $M_3 = C_1$ 
  - i dolje postoji (osigad) problem autentifikacije
  - ipak, E da pozvane KDC te ne može dešifrirati
- B salje  $M_4 = E(R, KAB)$  gdje je  $R = \text{NONCE}$  (Number used only once)
- $E(R)$ :  $E(\cdot)$  je općepoznata (ne tačna) funkcija
- NONCE - džec sigurnost
  - slučajno generisan broj
- problemi sustava (pr. 11.9.)
  - za  $N=100$  treba 4950 klijenata (bez KDC-a)
  - KDC treba da ih 100 (ukupno 200 s duplikatima kod A, B, ...)
  - pouzdanost je loša - sustav je centraliziran, ne radi bez KDC-a (njegove UPS)
    - veći je problem zagonjenje (ping) - KDC postaje veliko golo - počinje je raspodijeljeni sustav KDC-a (raspadela na područja)
- raspodjeljeni KDC
- delat je 10 područja ;  $N=100$
- međusobna komunikacija KDC-ova zaliđena međusobne uključeve

$$N_{\text{KDC}} = \frac{N_{\text{pod}}(N_{\text{pod}}-1)}{2}$$

Učinkovito je potrebno  $100 + 45 = 145$  ključeva (za KDC-ove)

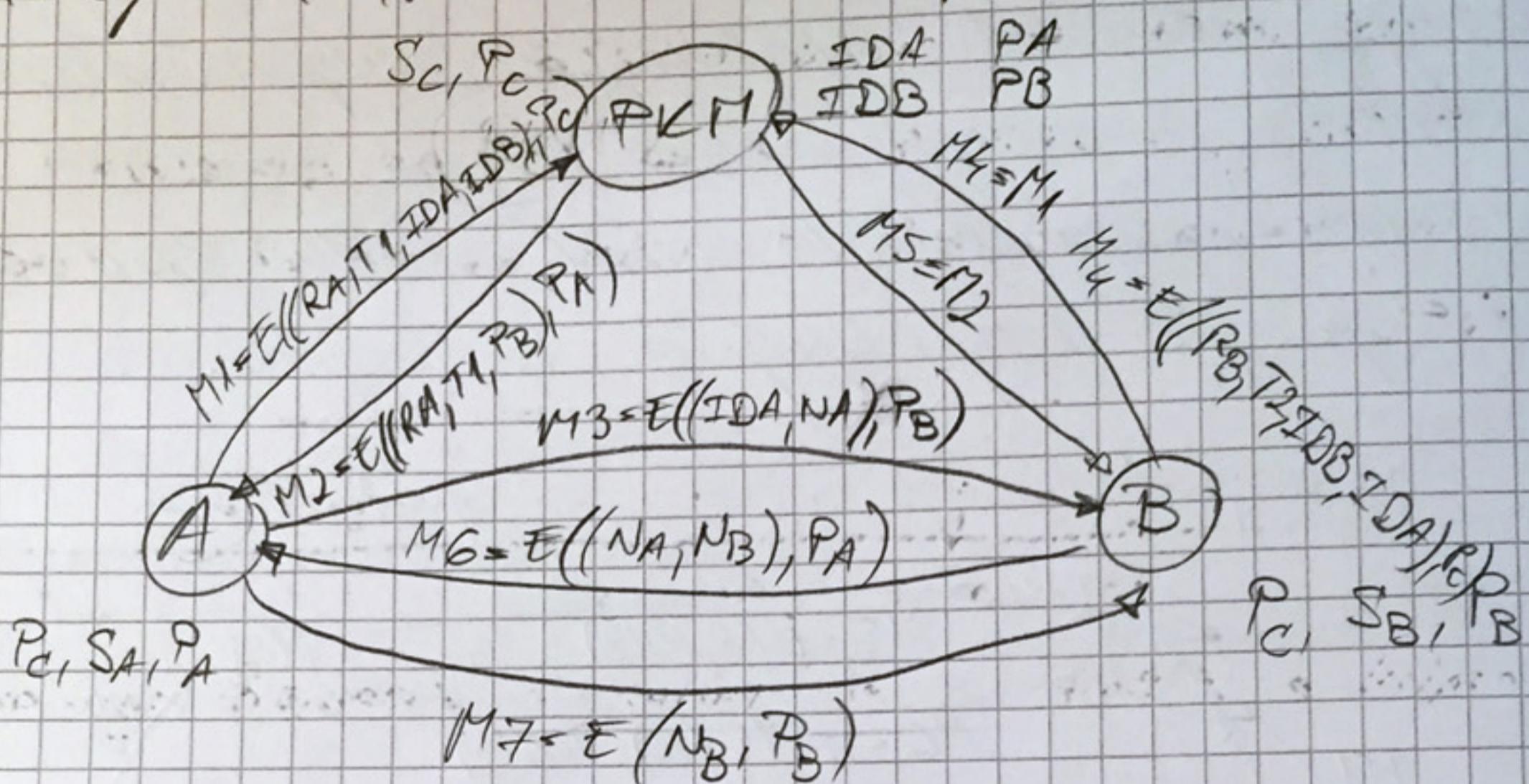
- proizlazi iz razine formule  $10 \cdot 10 + \frac{10 \cdot 9}{2} = 145$
- ali su sudsudski u istom području učina prethodnog algoritma
- u slučaju razvojih potreba je razvijena mada KDC-ova
- početno minimum  $10 + 2$  (između KDC-ova)
- posljedje je raspodjeljeni sat. kriptos.



- M1 je išli kaš i prije
- od KDC-a A prima poruku M2 koja je gotovo ista do na C1 (za drugu KDC-ovu)
- istovremeno KDC1 salje KDC2 poruku M3 (uključi se ključevi Kj)
- uključi Kj služi za komunikaciju KDC-a
- KDC2 salje B istu poruku C1 kas ; razine
- B salje A M5 ekv. poruci M4 iz prethodnog algoritma
- A vrada B poruku M6 te komunikacija može poceti
- M4 ; M5 iz prethodnog su tu M5 ; M6

## Raspodjela ključeva u zaključku asimetričnog kriptosustava

- u problemu sastavlja se problem autentifikacije i povećanja pouzdanosti (mobilizacija)
- takođe postoji zapredučki server - Central za raspodjelu javnih ključeva (PKM - Public Key Manager)



- $P_C$  - javni,  $S_C$  - privatni ključ PKM-a
  - mobilizacija obostražuju autentifikaciju pa A, B znaju da je  $P_C$  opravio PKM-ov
- A salje šifriranu poruku  $M_1$  (konsticijem  $P_C$ )
  - $T_1$  je vrijeme - da se ne radi o starijim porukama
  - $M_1 = E((RA, T_1, IDA, IDB), P_C)$
- PKM vraca A  $M_2$ , posusva je uključen  $T_1$  (da se zna da je jači, i zaštiti odgovor poslano)
  - $M_2 = E((RA, T_1, P_B), P_A)$
  - šifriran javni ključ  $P_B$  - delujiči ne treba biti tačan, ali se šifrira sa  $P_A$  (može i  $S_C$  - komunikacija)
  - Se bri znaci da su ti podaci sigurni, pouzdati jer samo PKM može uspostaviti  $S_A$  sa  $S_C$
- A salje B  $M_3$ 
  - $NA$  jeopravio NONCE

- kriptiranje s  $P_B$
- B preporučava poruku kod PKM-a porukom  $M_4$   

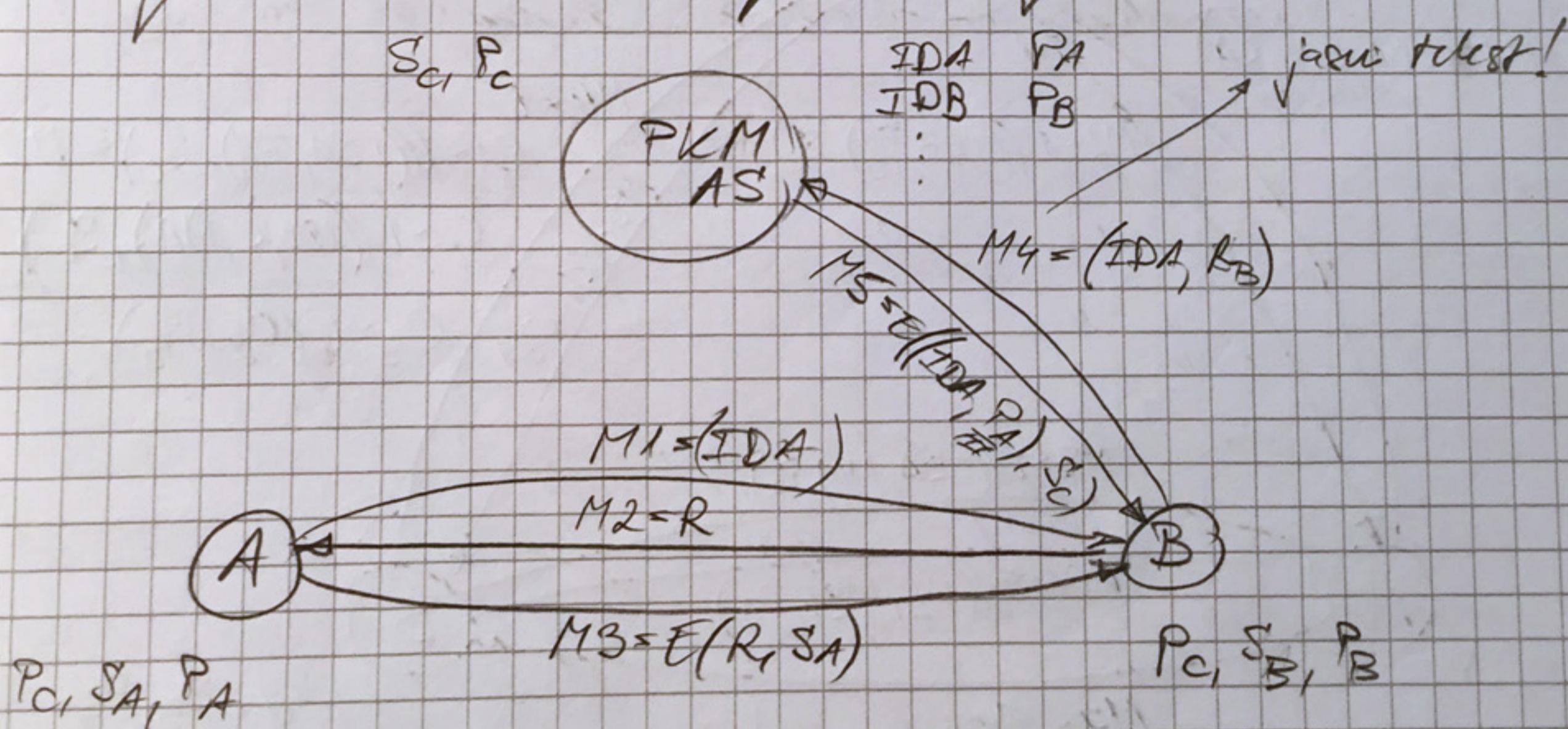
$$- M_4 = E((R_B, T_2, ID_B, ID_A), P_C)$$
- PKM odgovara B porukom  $M_5 \equiv M_2$   

$$- M_5 = E((R_B, T_2, P_A) P_B)$$
- B odgovara A porukom  $M_6$   

$$- NB \checkmark \text{ je NONCE od } B \text{ (bu ga generira)}$$
- A usporava B da je stvorio to A porukom  
 $M_7$

Tednostrau autentifikacija u zaklopcu asimetričnom kriptosustavu

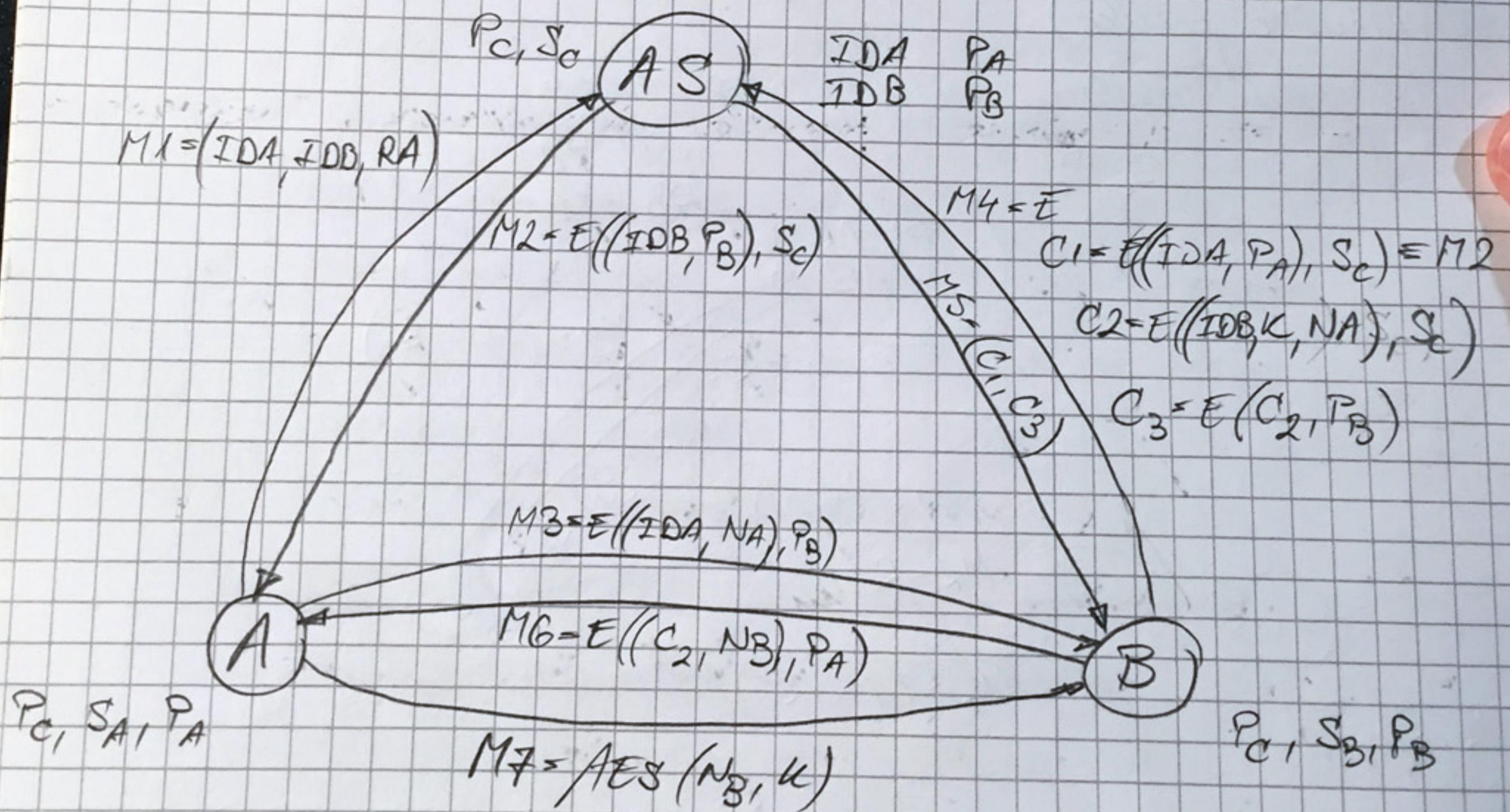
- usporjava S protokolom protokolom
- ove protokole treba znati! slavat (načini logiranja, postopek ekvalentne poruke)
- same jedna osoba autentificira drugu



- AS - autentifikacijski server (može biti isti kao i PKM)
- A salje B  $M_1$ ; B će usporiti te preporučiti; Salje poruku  $M_2 = R$ , gdje je  $R$  NONCE
- A salje kriptirani NONCE svojim priv. ključem  $S_A$

- B salje M<sub>4</sub> PKM-u (šavni ključ!, nekriptiran)
- PKM salje B poruku M<sub>5</sub> kriptiranu privatnim ključem (vlastitim) S<sub>C</sub>
- M<sub>5</sub> sadrži PA koji B zahtijeva porukom M<sub>4</sub> od PKM-a
- B zatim generira novi ključ za novi kript. te ga salje A kriptirano uz njegov javni ključ P<sub>A</sub>
- Prilikom A da se da je u potrazi B!; treba li A provjeriti je li njezina B na isti način - obrostrana autentifikacija

Obostrana autentifikacija u zatvorenum asimetričnum kriptosustavima



- cilj ovih algoritama je samo autentifikacija; prethodno razvijena ključeva
- algoritam:
  - A salje AS-u M<sub>1</sub>; vrši P<sub>B</sub>

- AS odgovara s M<sub>2</sub> (koji raspadač može dekript.)
- A salje M<sub>3</sub> B
- B salje AS-u poruku M<sub>4</sub> (kriptiranu)
  - M<sub>4</sub> = E((IDA, NA, IDB, RB), PC)
  - RB - zaključak za klijenta
- AS odgovara M<sub>5</sub> = M<sub>2</sub>
- C<sub>2</sub> - sigurnost; C<sub>3</sub> - autentičnost
- AS generira zapadničku tajnu klijenta K
  - C<sub>3</sub> - tajni klijenta za kom. s A; NONCE od A
  - A ne zna K
- B salje M<sub>6</sub> prema A
  - C<sub>2</sub> je opet kriptiran
- A salje M<sub>7</sub> prema B gdje dolazeće da je primila K
  - AES označava simetrično criptiranje, E asimetrično
  - B se uvjerava da je A ispravno primila K u pretvoru algoritma (je alg razvijena teoretski, dok se ovde dolazi autentifikacija i klijent A za simetrični kriptosustav)
- radi se o sljedećim algoritmima koji imaju sljedeći oblik, ali razvoj su surbi
- RSA - kriptiranje bazirajući asimetričnim algoritmom (kriptira se jasni tekst) uve pomoći
- Baptori su mali  $\rightarrow$  bilo potenciran da je ne mora rezultirati brojem redom od N  $\rightarrow$  se provodi se mod N te dekripcija za napadajuća postaje jednostavna
- praktični kriptosustavi se dozvoljavaju kriptiranje asim. alg. biti čega osim klijenta (koji ima vise dajući; općenito poznavaju formu)

## PRIJAVA ZA RAD

- prijava za rad apr. otklonom preta uve se zadržala zbog  
nugostva problema
  - broj znacajki kod informacija
  - je li ih (zajedno sli) dobro postupiti
  - veličina baze podataka?
  - brzina pristupa bazi?
- u praksi bio su znacajke ušu stroko prihvocene već se  
koristi išme konsultacija i ložnica
  - UID - user ID (proracun na temelju e-maila)
  - tako je sustav stari i ima predurosti
  - password nikad ne smije putovati među u mrežu u niti  
klijentu obliku
  - problemi - korisnik upotrebu ložnice
- zašticeće datoteku - kriptiranje ložnica P na način:

$$E(P, P)$$

- starija metoda, po novome H(P) (hash)
- napad rječnikom - račun hasha za sve rječi iz  
rječnika (treba otkriti samo jedan password kako bi  
usao u sustav; netko koristi kriptiran password)
- današnjem računalu za proracun hasha engleskog  
rječnika (ili usporedbeni gotovi hashevi) treba  
svega nekoliko sekundi
- pravila regularnih izraza (npr. rječi-brig); za  
jedan broj uključenih rječi treba 10 puta više  
vremena (fj. dvadesetak sekundi)

COVER REINFORCED

Autorizacija - zaštita pristupa posredniku

- subjekti su korisnici ili procesi (korisnici procesi  
objekti)

- osoba koja je potražila proces ili proces sam  
za sebe

objekti - objekti zaštite, datoteka, procesi

subjekti } zaštita pravila  
objekti } zaštita pravila

zaštita pravila - za svaku par definirati što su sve  
radni (nijenyati i sljed.)

- jedan od mogućih rešenja - matrica pristupa  
(operaciju sustav)

objekti

r, w	...	x	r - read
x	...	w	w - write
...	...	...	x - execute
r	...		

- iz matrice se stvaraju liste (matrična je  
prethodne reprezentacija)

liste:

- Osta prava pristupa objektu - za svaku se objekt  
def. što što su sve radni - generalno se  
iz stupaca (npr sati elementi) - access control

list

- Osta dozvola za pristup objektima (capability  
tickets) - npr sati elementi redaka

- za svaki se subjekat definiraju EPO snage raditi

- cešća lista: #subjekata - #objekata

→ Autentifikacijski protokol Kerberos

- 1988. - MIT

- autentifikacija studenata

- pretpostavke: svaki se student va početku autentificira → sustav koristi i dva (dvojni logon uk password sustav)

- problem: mreža nije sigurna - prva pretpostavka  
- kriptirajuće nije dobro operirajuće  
- password u mreži ljepe oblike ne mogu putovati mrežom

- druga pretpostavka: racionala su sigurna  
- nije boljka realna pretpostavka

- tad je OTO pozvao DES, da upani se zaštita Kerberos autentifikacije

- treća pretpostavka: u sustavu postoji poslužitelj (Kerberos poslužitelj) koji predstavlja treću stranu koja surađuje

- zahtev: single sign-on - samo jedna autentifikacija je dovoljna za sve sisteme unutra

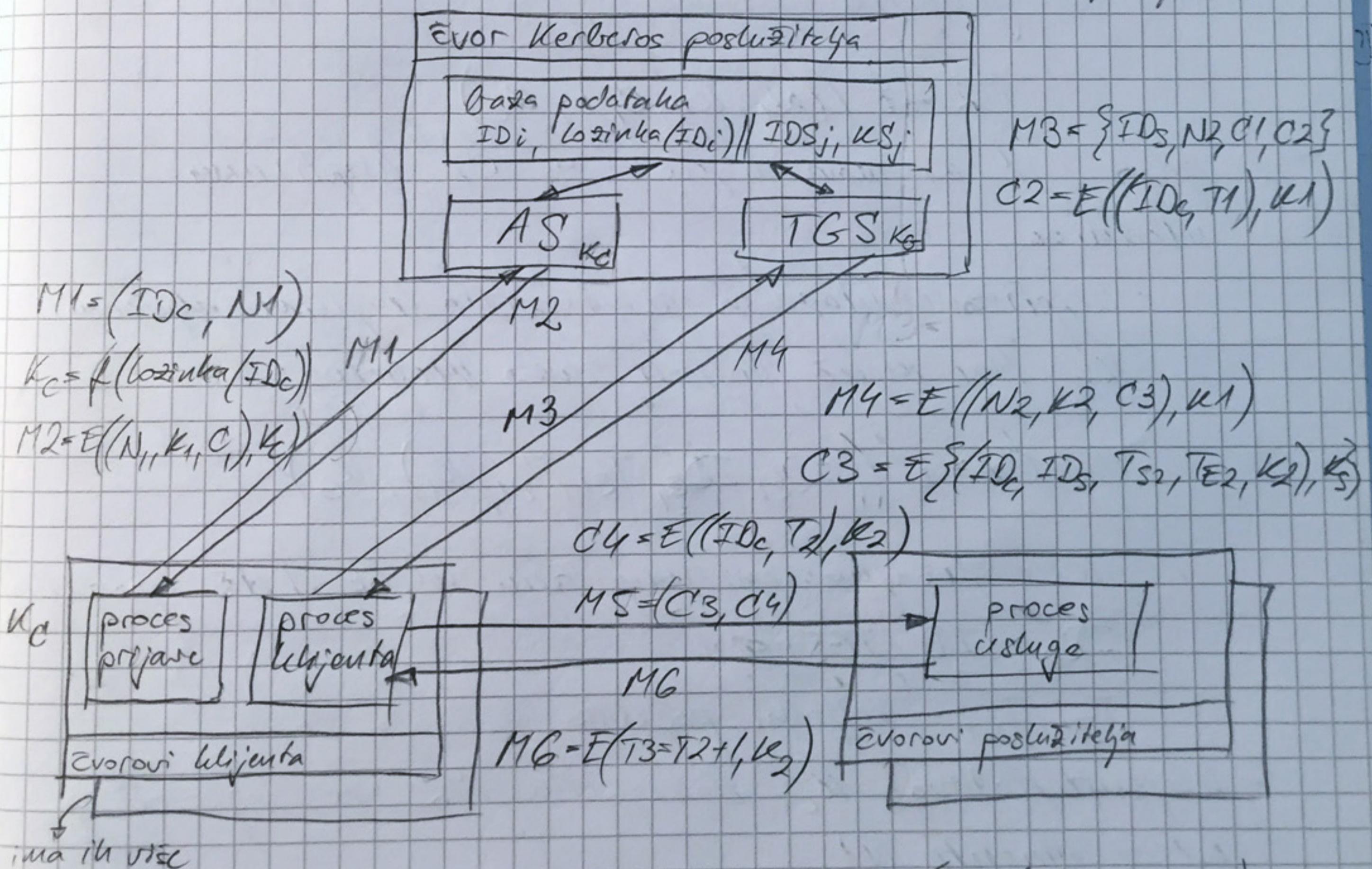
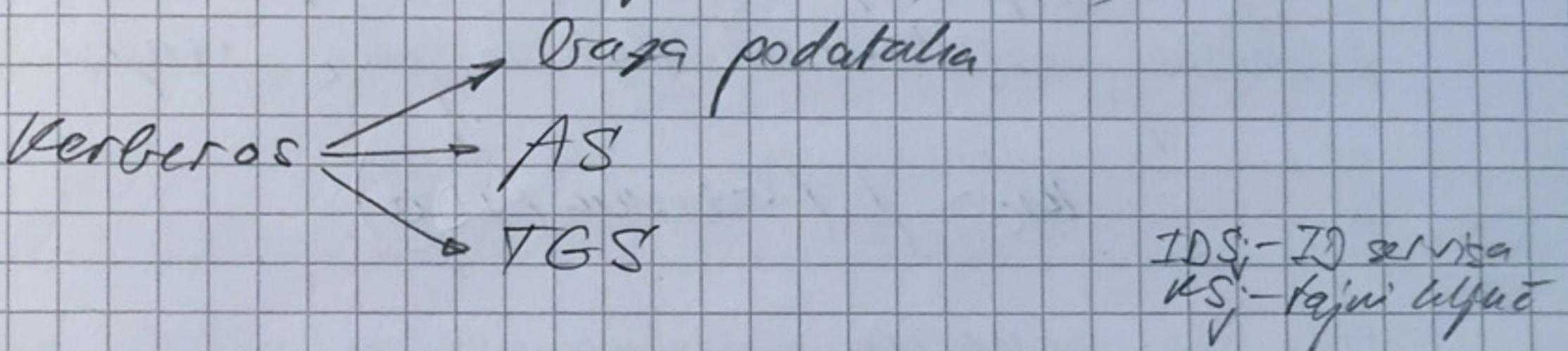
- Kerberos je propisane EPO treba valjati autentifikacije

sustav s m' vrste čvor:

- čvor ulijepšava (mra ih ovise)

- čvor poslužitelja (može im biti ovise, npr. web poslužitelj)

- Čvor Kerberos poslužitelja (samo jedan)
- sastoji se od tri komponente: Baza podataka u kojoj su zapravo user-id, lozinka; tajni ključovi svih poslužitelja u sustavu (zadnji za server-id-en)
- druga komponenta: AS (Authentication Server, autentifikacijski poslužitelj)
- treća komponenta: poslužitelj za dodjelu ulaznica za pristup pojedincim uslugama (Ticket Granting Server, TGS)



- proces prijava - u pravilu jednom (single sign-on)
- nakuć koga se aplikacije, procesi klijenta - ipr. Firefox, printer...

- proces usluge, vpr. mail ; st.
- početak - proces autentifikacije sačitava dobre poruke
- sačitava aplikacija - četiri poruke
  - dvije TBS- u, dvije poslužitelja

- proces pripreme

- prvo su unosi ime iz kojeg se proracunava (pronalazi) VID (user ID), odredjuje IDc te se generira NONCE N1  $\rightarrow M_1$

- u jedom mjestu ide VID ; N1

- M2 - AS predaje IDc te uz ponad obra pronalazi lozinku te dobitnu Kc, dajući klijenću

$$K_C = f(\text{lozinka}(ID_C))$$

te se generira

$$M_2 = E((N_1, K_1, C_1), K_C)$$

gdje je  $K_1$  novi klijent, a  $C_1$  kriptirana ulaznica

- kriptirana ulaznica se kriptira sačinjući klijentom  $K_G$  taj klijent morati same kerberos :

$$C_1 = E((ID_C, ID_G, T_{S1}, T_{E1}, K_1), K_G)$$

$K_1$  - slučajno generiran tapci klijent (AS generira)

$ID_G$  - ID TGS-a

$T_{S1}, T_{E1}$  - vrijeme početna i kraja rada

- klijent dobitna  $K_C$ , dekriptira  $M_2$  te usporoduje  $N_1$  i vlastiti  $N_1$

- sprva kriptiranu ulaznicu te ju koristi po potrebi

- u sljepu potrebe za uslugu sa M3

$$M3 = \{ ID_S, N2, C1, C2 \}$$

ID\_S - svaki identifikator

N2 - novi NONCE

C1 - krypt. ulaznica

- C2 je kriptirani autentifikator

$$C2 = E((ID_C, T_1), K_1)$$

autentifikator

ID\_C - vlastiti ID

T\_1 - vrijeme stanja zahtjeva

K\_1 - ključ ekspresije

- TGS dekriptira C1 te dobiva ID\_C, ID\_S (da  
zna da je usmjerjen)

- provjerava  $ID_C = ID_C \wedge C2$

- vraca poruku M4 ako je sve u redu:

$$M4 = E((N2, K_2, C3), K_1)$$

K\_2 - skret za komunikaciju s procesom poslužitelja,

• K\_1 je skret za TGS

N2 - prethodni NONCE

- C3 je pravom kriptirana dozvola

$$C3 = E\{(ID_C, ID_S, T_{S2}, T_{E2}, K_2), K_S\}$$

ID\_S - identifikator usluge

T\_{S2}, T\_{E2} - vrijeme usluge

$$(ID_C, ID_S, T_{S2}, T_{E2}, K_2)$$

dозвola

jednake oblike  
K\_S i C1

- klopenut řádce MS = (C3, C4)

C4 = E1(IDc, T2), K2

- odpovídá je MG:

MG = E(T3 = T2 + 1, K2)

- generátora sl. klopena koženitelního zrcadlovam - IRB

- tvr. uvozova záhlite Kerberos systému:

- prověra autentiknosti samo na počítaču (opracování protokolu) - první uvoz

- druhý uvoz - sigurné poslánky

- uz svalku poslánku kde se řádce (i klopena a protokolu) se dodaje kryptovaný autentifikator

- "privátní" poslánka - 1 poslánka je (uz autentifikator) kryptovaná - třetí uvoz

- išti klopeni has i autentifikator

- nedostatci / propusti Kerberosa:

- sváří program musí být "kerberosem", snad komun. & procesor klopena

- nějak autorizace (velké počítání)

- uživ. dikt. predvídání

- ostanění opačná nedostřednost - treba proštítací

- Kerberos poslání může být fyzicky zasílán

- sigurné poslání například klopena?

- Kerberos podleježde strojím amesického založení o

izvozu kryptografie (danas vložen propust)

→ PKI (Public Key Infrastructure)

šta treba pružiti?

1. Integrirat - proujera proujene poruke (hash)

2. Sigurnost uz identitet

3. Pouzdanost vremena : datuma - time stamp  
authenticity

4. Formalna - planu valjanost - sudski procesi, ne ma veze s tehničkim aspektima

- mora biti zakonski reguliran - RH Zakon o elektroničkom poslovanju

PKI - skup tehnologija, protokola, normi i usluga

koji zajedno omogućuju sigurnu komunikaciju temeljenu na sustavu javnih ključeva preko nesigurnih mrež

- zadatā se PKI-a prava zaštjtivina treba svrstati na dva dijela:

1. Nedvojbeno potvrđivanje javnih ključeva s klijentima te proujera još li važeći

2. offline proujera identiteta certifikatima

- certifikati služe za offline proujera identiteta

- problem PKI-a - kada je certifikat nevažeći?

- kudi gube ključeve, kartice i druge; sustav mora moći blokirati kartice

- tako se spesava i opoziv certifikata, tj. treba početati lista opozvanih certifikata

- PKI nije zaštitio upravo bog offline proujere; niti su online makar se i taj problem spesava

- certifikat - jači supodoblja usta potvrđuje da je određeni korisnik u trenutku izdavanja certifikata posredovao