

Varne komunikacije

Študijsko gradivo 2011/12

Anton Umek
anton.umek@fe.uni-lj.si

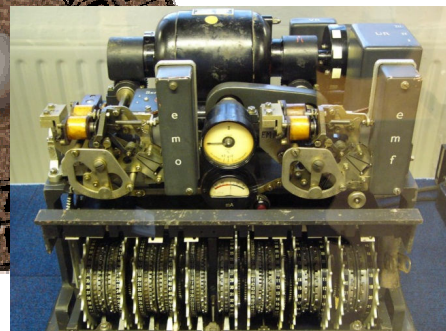
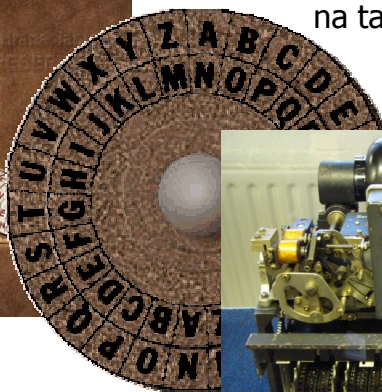
Zgodovina šifriranja

Varnost je dolgo temeljila
na tajnosti postopka:



Šparta, 500 p.n.š.

Julij Cezar, 100 p.n.š.



Enigma, 1920-1940

Moderno šifriranje

- Šifrirni algoritem je javen, varnost temelji na tajnosti ključev !
- **namen** šifriranja, varnostni vidiki:
 - tajnost
 - verodostojnost
 - avtentičnost
 - neovrgljivost
- Najbolj znani šifrirni **algoritmi**:
 - DES, IDEA, **AES**
 - **RSA**, DH
 - MD5, SHA-1,..**SHA-3**



1. Uvod v varne komunikacije

- Elektronski dokumenti
- Izmenjava datotek
- Zasebnost in zaupnost
- Celovitost sporočil
- Šifriranje dokumentov





Elektronski in tiskani dokumenti

- Skoraj vsi dokumenti nastajajo s pomočjo računalnika.
- Elektronski dokument ima veliko prednosti:
 - kadarkoli ga lahko ponovno natisnemo
 - **lahko ga tudi spreminjamo**: spremenimo naslovnika, datum...
- Zakaj se potem velik del dokumentov še vedno tiska na papir ?
- Vprašljiva je originalnost elektronskega dokumenta
- Tiskani dokument vsebuje lastnoročne podpise in časovne žige
- Elektronski dokument brez varnostnih mehanizmov ni pravno veljaven:
 - ne more služiti za arhiv ali kot pogodba

5



Razvoj izmenjave elektronskih dokumentov

- dokument natisnemo na papir in po pošti pošljemo naslovniku
- dokument pošljemo iz računalnika direktno na telefaks naslovnika
- dokument pošljemo v elektronski obliki na fizičnem mediju (kurir, pošta, DHL..)
- dokument posredujemo v elektronski obliki na primer preko elektronske pošte

- zadnji način je od vseh naštetih najbolj učinkovit vendar hkrati tudi najbolj ranljiv !

6



Zasebnost in zaupni dokumenti

- Govorimo o zasebnosti ali tajnosti komunikacije.
- Zaupni dokument je namenjen samo naslovniku, zato želimo preprečiti vpogled tretje osebe.
- Če zaupni dokument pride v napačne roke je zasebnost komunikacije izgubljena.
- Verjetnost takšnega dogodka je omejena s stopnjo varovanja zasebnosti. Zelo zaupne dokumente varujemo z najvišjo možno stopnjo varovanja zasebnosti (tajnosti).
- Pri pismu je zasebnost udeležencev v komunikaciji slabo varovana z vlaganjem tiskanega dokumenta v ovojnico. Zaupnost tiskanega dokumenta je lahko posebej označena, kar pa lahko še dodatno pritegne pozornost.

7



Zagotavljanje celovitosti sporočil

Poznamo več vidikov celovitosti sporočil:

- zasebnost ali tajnost (privacy , confidentiality)
 - Ali je vsebina sporočila res dostopna samo naslovniku ?
- verodostojnost :
 - Ali je sprejeto sporočilo res enako oddanemu sporočilu ?
- avtentičnost (authentication) zagotavlja izjavljeno identiteto pošiljatelja:
 - Ali nam sporočilo res pošilja predstavljeni pošiljatelj ?
- neovrgljivost (nonrepudiation):
 - Ali lahko pošiljatelj zanika avtorstvo sporočila ?
- časovna opredeljenost: časovne omejitve veljavnosti, časovni žig, trajnost !

8



Šifriranje dokumentov

- Varovanje zasebnosti zagotovimo s šifriranjem dokumentov tako, da velja:
 - iz šifriranega dokumenta ni mogoče razbrati vsebine in
 - samo naslovnik zna dešifrirati dokument.
- Zgodovina šifriranja sporočil:
- Veda o šifriranju (kriptologija) je bila zelo dolgo na seznamu najstrožje varovanih skrivnosti
 - Grki: kryptos "skrite", logos "besede", angl: cryptology
 - Cezarjev postopek šifriranja : CESARUS->FHVDUAV
 - Nemški šifrirni stroj iz II. svetovne vojne: Enigma
- Javno uporabo kriptografije je omogočila iznajdba asimetričnega postopka šifriranja pred približno 30. leti
 - junija 1991 je Philip Zimmerman objavil programski paket za varno izmenjavo sporočil PGP (Pretty Good Privacy)
 - Danes uporabljamo vrsto standardnih postopkov šifriranja sporočil v privatnih in poslovnih komunikacijah.

9



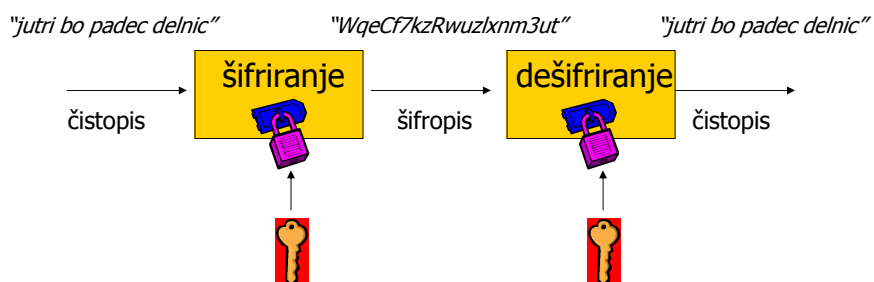
Šifrirni postopek

- Lastnosti dobrega šifrirnega postopka:
 - Zasebnost ne sloni na tajnosti postopka pač pa na tajnosti ključa za dešifriranje.
 - Postopek šifriranja mora biti izvedljiv na računalniku v realnem času.
 - Postopek dešifriranja mora izvedljiv na računalniku v realnem času za tistega, ki pozna dešifrirni ključ.
 - Postopek dešifriranja ne sme biti izvedljiv v realnem času za napadalca, ki ne pozna ključa, čeprav razpolaga z zelo zmogljivim računalnikom.
- Glede na smernost šifrirnega postopka ločimo:
 - Simetrično šifriranje (dvosmerno šifriranje)
 - Asimetrično šifriranje (enosmerno šifriranje)

10

Simetrično šifriranje

- Za šifriranje in dešifriranje uporabimo isti ključ:

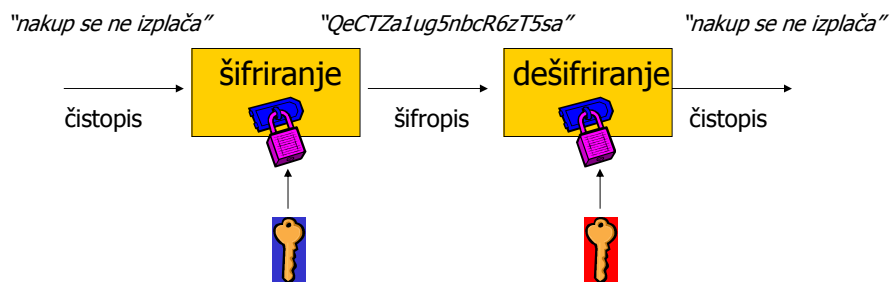


- Pošiljatelj in prejemnik morata uporabiti enak **tajni** ključ !

11

Asimetrično šifriranje

- Ključa za šifriranje in dešifriranje nista enaka:

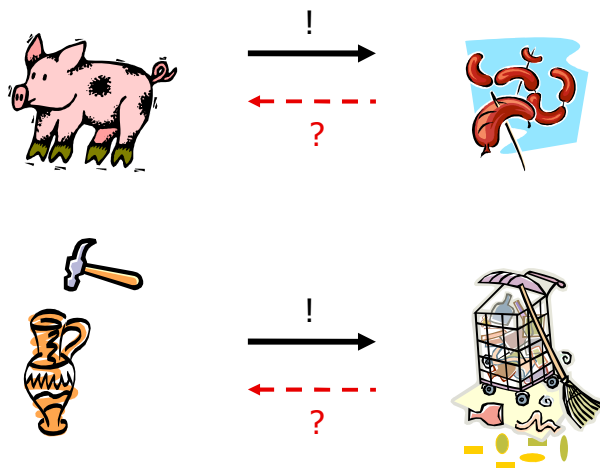


- Pošiljatelj šifrira sporočilo z **javnim** ključem prejemnika.
- Prejemnik dešifrira sporočilo z zasebnim (privatnim) **tajnim** ključem.
- Asimetrični postopki šifriranja so zasnovani na **enosmerni funkciji** s stranskim vhomom (one way trapdoor function).

12

Enosmerne funkcije

- Značilnost enosmerne funkcije: preslikava v nasprotni smeri je praktično nemogoča:



13

Mešani postopek šifriranja

- Uporabimo simetrični in asimetrični postopek šifriranja:
 - Asimetrični postopek uporabimo za izmenjavo začasnega **sejnega ključa**.
 - Po simetričnem postopku s sejnim ključem šifriramo in dešifriramo sporočilo.
- Pošiljatelj pošlje simetrično šifrirano sporočilo in zraven še asimetrično šifriran ključ, s katerim je bilo sporočilo šifrirano:
 - Pošiljatelj naključno generira **sejni ključ** in z njim šifrira sporočilo.
 - Ključ s katerim je sporočilo šifrirano se šifrira z javnim ključem naslovnika.
- Prejemnik prejme šifrirano sporočilo in šifriran sejni ključ.
 - Prejemnik dešifrira **sejni ključ** s svojim privatnim tajnim ključem.
 - Prejemnik na osnovi sejnega ključa dešifrira sporočilo.

14

Varne komunikacije

Digitalni podpis in digitalno potrdilo:

- Elektronski prstni odtis dokumenta



- Digitalni podpis



- Upravljanje s ključi



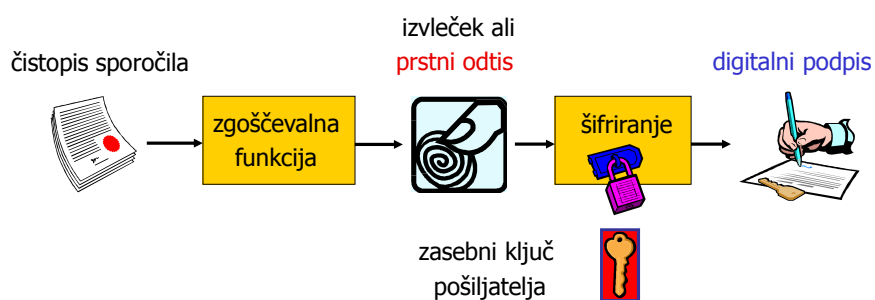
- Digitalno potrdilo



15

Digitalni podpis

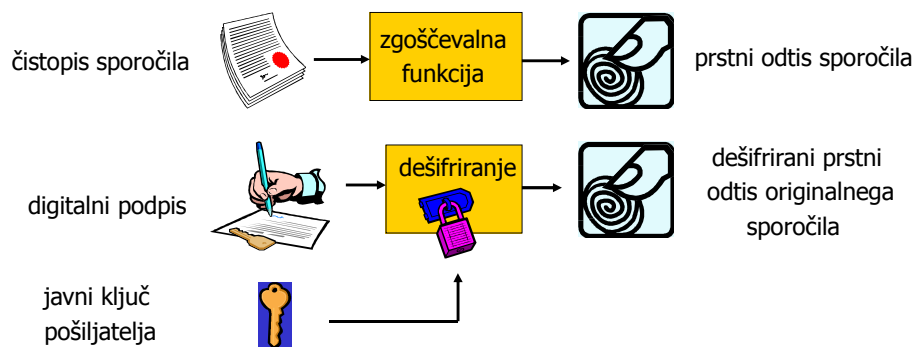
- Digitalni podpis je s tajnim ključem šifrirani **prstni odtis** sporočila:



- Zgoščevalna funkcija je enosmerna funkcija in vsaka sprememba čistopisa spremeni tudi prstni odtis sporočila.
- Napadalec bi lahko spremenil sporočilo in dodal nov prstni odtis !
- Pošiljatelj zaščiti prstni odtis s šifriranjem!

16

Preverjanje digitalnega podpisa



- Prejemnik preveri ujemanje prstnih odtisov in če sta enaka
 - je **sporočilo verodostojno**,
 - potrjena je **identiteta pošiljatelja** in
 - **pošiljatelj ne more zanikati** sporočila.

17

Namen digitalnega podpisa



- Digitalni podpis dodajamo nešifriranemu sporočilu in zato ne zagotavlja tajnosti komunikacije.
- Pošiljatelj z digitalnim podpisom zagotovi:
 - verodostojnost sporočila,
 - potrjuje svojo identiteto in s tem
 - sprejme tudi odgovornost za sporočilo.
- Prejemnik lahko hkrati preveri verodostojnost in avtentičnost:
 - Ali je sprejeto sporočilo res enako oddanemu sporočilu ?
 - Ali nam sporočilo res pošilja predstavljeni pošiljatelj ?
- Če prejemnik potrди verodostojnost sporočila in avtentičnost pošiljatelja, potem tudi pošiljatelj ne more sporočila zanikati:
 - Če se prstna odtisa ujemata, potem sporočilo ni bilo spremenjeno in podpisal ga je lahko le pošiljatelj, ki ima edini pravi zasebni ključ.
- Digitalni podpis omogoča zagotavljanje verodostojnosti, avtentičnosti in neovrgljivosti sporočil.

18

Uporaba zasebnih in javnih ključev

- Digitalni podpis temelji na asimetričnem šifrirnem postopku, ki uporablja parov imetnikovih ključev: javni ključ + zasebni ključ



- Vsak uporabnik nosi odgovornost za uporabo in varovanje **zasebnega ključa**. Dostop do tajnega ključa varujemo z dolgim geslom, ki ga imenujemo fraza. Uporabnik ne sme zaupati nikomur svojega zasebnega ključa. Če to stori, potem nosi tudi vso odgovornost za zlorabe.

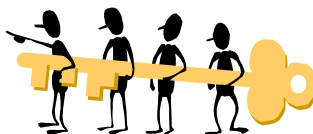


- **Javni ključ** mora biti vsakomur dostopen z jamstvom, da pripada navedenemu uporabniku. V nasprotnem primeru lahko pride do problemov:
 - Problem lažne identitete: napadalec podtakne lažni javni ključ in dešifrira vsa prestežena sporočila.
 - Problem zanikanja identitete: pošiljatelj zanika lastno sporočilo.

19

Upravljanje s ključi

- Javni ključ mora nositi garancijo, da res pripada navedenemu uporabniku. **Overjanje javnih ključev** opravlja posebna služba (podobno notarju), ki skrbi tudi za upravljanje s ključi.
- **Urad za overjanje (CA=Certification Authority)** potrjuje verodostojnost javnih ključev z digitalnim podpisom odgovorne osebe. Imetnik javnega ključa se mora ob **registraciji** identificirati in s tem prevzema odgovornost za uporabo zasebnega ključa. Identifikacijo izvrši uradna oseba (**RA=Registration Authority**).
- Na zahteve imetnikov opravlja CA tudi **razveljavitve javnih ključev**. Potreba po preklicu javnega ključa nastopi v primeru izgube tajnosti zasebnega ključa.



20

Digitalno potrdilo

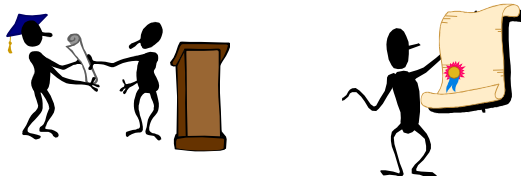
- Digitalno potrdilo (digital certificate) je overjena kopija javnega ključa.
- Digitalno potrdilo vsebuje:
 - kopijo javnega ključa
 - identifikacijske podatke imetnika
 - digitalni podpis [Urada za overjanje \(CA\)](#)
 - datum začetka veljavnosti potrdila
 - datum poteka veljavnosti potrdila
 - serijsko številko
 - ...
- Veljavnost digitalnega potrdila je odvisna od zaupanja uporabnikov v [CA](#).
- Neznani CA ne smemo zaupati !!



21

Pridobitev digitalnega potrdila

- Glavni overitelj digitalnih potrdil za pravne in fizične osebe je [SIGEN-CA](#) (Slovenian General Certification Authority)
- Spletno kvalificirano digitalno potrdilo pridobimo nekaj dni po oddaji izpolnjenega formularja na Upravni enoti ob identifikaciji z osebnim dokumentom.
- Digitalno potrdilo lahko med drugim uporabimo tudi za različne storitve na portalu [e-uprava](#)
 - oddaja vlog za upravne storitve,
 - oddaja obrazcev za dohodnine,
 - vpogled v osebne podatke centralnega registra prebivalstva ..

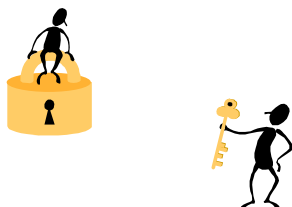


22

Varne komunikacije

3. Šifrirni algoritmi

- Klasično šifriranje
 - transpozicijska in substitucijska šifra
 - pretočno in blokovno šifriranje
- Simetrični šifrirni algoritmi
 - DES
 - AES
 - ostali simetrični šifrirni algoritmi



23

Klasične metode šifriranja

- Pri **transpozicijskem** šifriranju **premešamo** znake v sporočilu
 - izberemo tajno "ključno besedo" npr. *blisk* :

geslo →

čistopis:

napad ob šesti uri

	<i>b</i>	<i>l</i>	<i>i</i>	<i>s</i>	<i>k</i>
1	1	4	2	5	3
n	n	a	p	a	d
o	o	b	š	e	s
t	t	i	u	r	i

šifropis:

notpšudsiabiaer

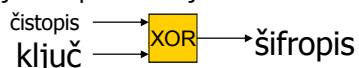
- Pri **substitucijskem** šifriranju **zamenjamo** znake v sporočilu
 - zamenjava je lahko s pomočjo tabele ali algoritma
 - Primer je Cezarjeva šifra: $C(P) = (P+3) \bmod 25$

24

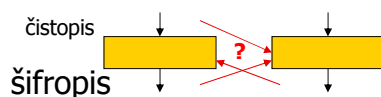
Pretočno in blokovno šifriranje

- Glede na dolžino sporočila, ki ga naenkrat šifriramo ločimo:
 - **Pretočno šifriranje** preslika sproti vsak znak ali zelo majhno število znakov, primer je substitucijsko šifriranje.
 - **Blokovno šifriranje**, kjer veliko število znakov čistopisa šifriramo v blok znakov šifropisa, primer je transpozicijsko šifriranje.
- **Pretočno šifriranje** je mnogo hitrejše od blokovnega.

- Primer zelo hitrega pretočnega šifriranja je množenje bitnega zaporedja čistopisa z naključno izbranim ključem po modulu 2.



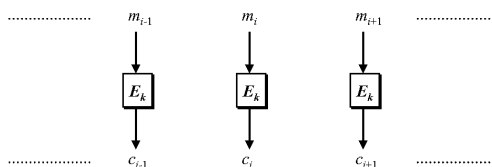
- **Blokovno šifriranje** je tipično z dolžino 64 bitov (primer: DES). Glede na soodvisnosti čistopisov in šifropisov med bloki ločimo več načinov blokovnega šifriranja (ECB, CBC, CFB, OFB).



25

Načini blokovnega šifriranja

- **ECB = Electronic Code Book**

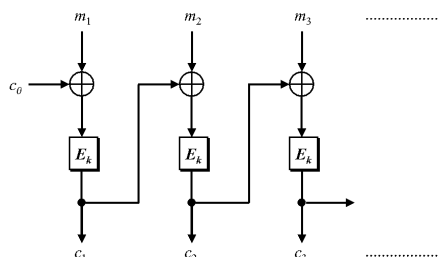


- Med bloki ni povezav, vsak blok šifriramo ločeno z 64 bitnim ključem
- Slabost: pri ponavljajočih blokih čistopisa dobimo tudi ponavljajoči vzorec v šifropisu.

26

Načini blokovnega šifriranja

- CBC = Cipher Block Chaining

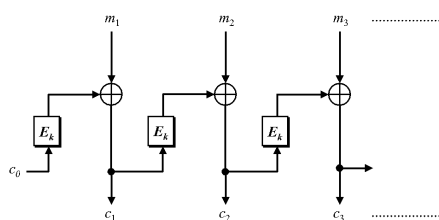


- Bloki so verižno povezani tako, da vedno šifriramo mešani (XOR) čistopis bloka in šifropis predhodnega bloka.
- Veriženje preprečuje pojav ponavljajočih vzorcev v šifropisu.
- Začetno stanje določa inicializacijski vektor c_0

27

Način blokovnega šifriranja

- CFB = Cipher Feedback Mode

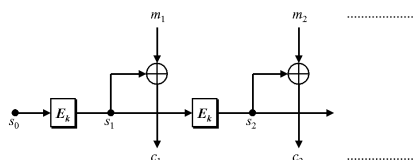


- Šifropis bloka dobimo z mešanjem (XOR) čistopisa in šifropisa predhodnega bloka.
- Začetno stanje določa inicializacijski vektor c_0 .

28

Načini blokovnega šifriranja

- OFB = Output Feedback Mode

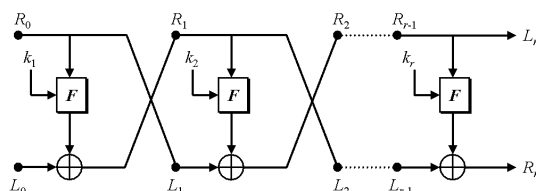


- Šifropis bloka dobimo z mešanjem (XOR) čistopisa in zaporednega stanja s_k .
- Zaporedja podatkovnih blokov s_k dobimo s šifriranjem predhodnih blokov s_{k-1}
- Začetno stanje s_0 je naključno število

29

Blokovni šifrirni postopek s ponavljanjem

- Proces blokovnega šifriranja lahko poteka v več krogih z enako transformacijsko funkcijo in različnimi ključi. Nabor ključev v tem primeru izhaja iz istega tajnega ključa. Varnost algoritma se povečuje s številom krogov, žal pa tudi računska kompleksnost.
- Feistel - ovo šifriranje je **večkrožno blokovno šifriranje**, kjer podatkovni blok razpolovimo na levi in desni del:



- Šifriranje se izvaja samo nad polovico podatkovnega bloka, druga polovica pa se diagonalno prepisuje v naslednji krog.
- Dešifriranje poteka na enak način, vendar s ključi v nasprotnem vrstnem redu (k_r, \dots, k_2, k_1)

30

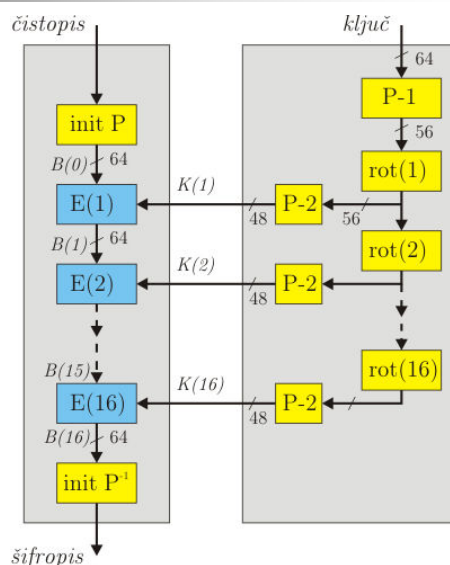
Simetrični šifrirni algoritem DES

- **DES = Data Encryption Standard**
 - prvič je objavljen 1977, vlada ZDA ga je izbrala kot standard za podatkovne komunikacije,
 - ključ ima dolžino 56 bitov, dolžina bloka je 64 bitov,
 - največkrat se uporablja CBC način bločnega šifriranja, možni pa so tudi vsi ostali načini (CFB in OFB).
 - s poskušanjem (brute force) je danes mogoče z zelo dobro opremo dešifrirati DES v razmeroma kratkem času.
 - **Triple-DES (3 DES)** : trikrat šifrira 64 bitni blok podatkov z DES algoritmom z različnimi ključi. Obstajajo tri verzije 3 DES:
 - $E(K1, E(K2, E(K3, P)))$, 3 ključi,
 - $E(K1, E(K2, E(K1, P)))$, 2 ključa,
 - $E(K1, D(K2, E(K1, P)))$, 2 ključa,
 - vse tri verzije so enako varne (efektivni 112 bitni ključ)

31

Šifrirni algoritem DES

- Po začetni permutaciji poteka šifriranje v 16 krogih
- na osnovi 56 bitnega sejnega ključa generiramo 16 podključev $K(r)$
- jedro DES algoritma je enkripcijski modul $E(1) - E(16)$
- zadnja operacija nad blokom šifriranih podatkov je inverzna začetna permutacija



32

Permutacije bitov

- blok N bitov preslikamo tako, da zamenjamo lego bitov v bloku
- takšni preslikavi sta permutaciji bloka podatkov **IP** in **IP⁻¹** pri DES algoritmu:

44 -> 35

Bit	0	1	2	3	4	5	6	7
1	40	8	48	16	56	24	64	32
9	39	7	47	15	55	23	63	31
17	38	6	46	14	54	22	62	30
25	37	5	45	13	53	21	61	29
33	36	4	44	12	52	20	60	28
41	35	3	43	11	51	19	59	27
49	34	2	42	10	50	18	58	26
57	33	1	41	9	49	17	57	25

35 -> 44

Bit	0	1	2	3	4	5	6	7
1	58	50	42	34	26	18	10	2
9	60	52	44	36	28	20	12	4
17	62	54	46	38	30	22	14	6
25	64	56	48	40	32	24	16	8
33	57	49	41	33	25	17	9	1
41	59	51	43	35	27	19	11	3
49	61	53	45	37	29	21	13	5
57	63	55	47	39	31	23	15	7

33

Permutacije bitov v manjše ali večje bloke

- Blok N bitov preslikamo z zamenjavo lege v manjši blok bitov. Primer **redukcije** pri DES algoritmu je permutacija bitov po tabeli P-2:

Bit	0	1	2	3	4	5
1	14	17	11	24	1	5
7	3	28	15	6	21	10
13	23	19	12	4	26	8
19	16	7	27	20	13	2
25	41	52	31	37	47	55
31	30	40	51	45	33	48
37	44	49	39	56	34	53
43	46	42	50	36	29	32

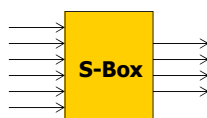
- Blok N bitov preslikamo z zamenjavo lege v večji blok bitov. Primer **ekspanzije** pri DES algoritmu je permutacija bitov po tabeli E:

Bit	0	1	2	3	4	5
1	32	1	2	3	4	5
7	4	5	6	7	8	9
13	8	9	10	11	12	13
19	12	13	14	15	16	17
25	16	17	18	19	20	21
31	20	21	22	23	24	25
37	24	25	26	27	28	29
43	28	29	30	31	32	1

34

Substitucijsko šifriranje

- Blok N bitov preslikamo v nov blok z zamenjavami. Primer substitucijskega kodiranja pri DES algoritmu so **S- škatle**
- S_1 do S_8 pretvarjajo 6-bitne bloke v 4-bitne bloke.
- Zgornji in spodnji bit določata vrstico v tabeli, srednji štirje biti pa določajo stolpec v tabeli:

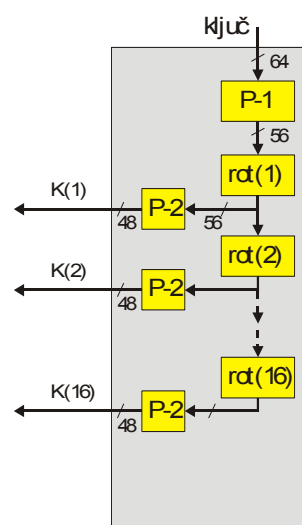
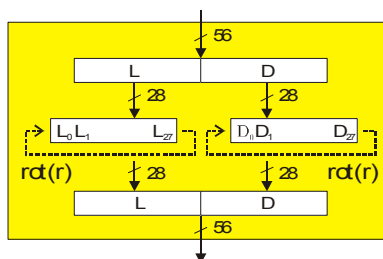


Row / Column	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6

35

Generacija nabora 16 ključev

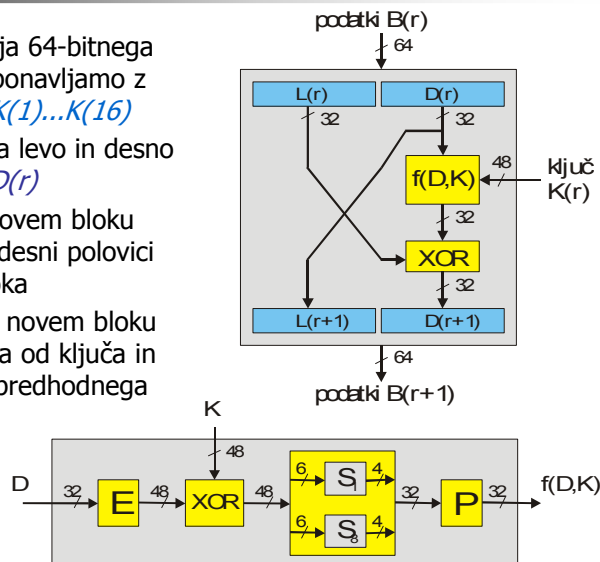
- P-1 preslika 64-bitni ključ v 56-bitni ključ
- P-2 je zamenjava bitov z redukcijo bloka iz 56 v 48 bitov
- rotacija ali krožna preslikava leve in desne polovice 56-bitnega ključa se ponavlja z različnim številom premikov (1 ali 2)



36

Jedro DES algoritma

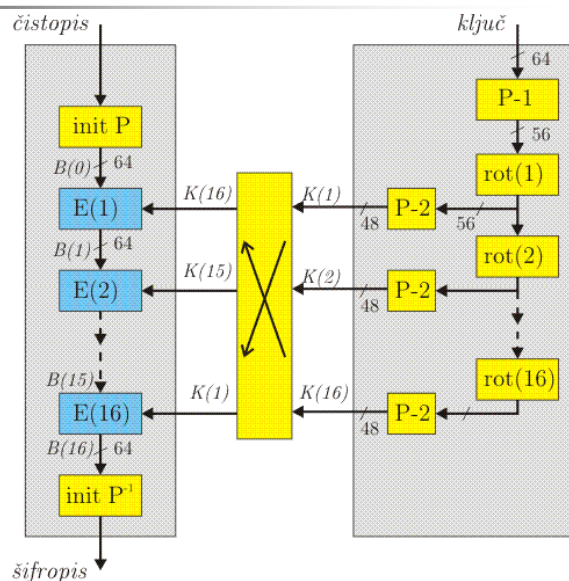
- postopek šifriranja 64-bitnega bloka podatkov ponavljamo z različnimi ključi $K(1) \dots K(16)$
- blok razdelimo na levo in desno polovico $L(r)$ in $D(r)$
- leva polovica v novem bloku $L(r+1)$ je enaka desni polovici predhodnega bloka
- desna polovica v novem bloku $D(r+1)$ je odvisna od ključa in celotne vsebine predhodnega bloka:



37

Dešifrirni algoritem DES

- DES je simetrični šifrirni postopek
- algoritem za dešifriranje je enak kot za šifriranje,
- Pri dešifriranju se zamenja le vrstni red pri uporabi podključev:
 - $K(16)$,
 - $K(15) \dots$
 - $K(1)$



38




AES = Advanced Encryption Standard

- NIST je v natečaju za AES leta 1997 postavil zahtevo za javni simetrični blokovni algoritem, ki deluje z 128-bitnimi blokom in lahko uporablja tri dolžine ključa: 128, 192 in 256 bitov.
- **RIJNDAEL algoritem** izpolnjuje postavljene zahteve z najboljšo oceno analiz v času javnega ocenjevanja (l. 2000). Izbiramo lahko med devetimi kombinacijami parov dolžine bloka in dolžine ključa (128, 192 in 256). Ime algoritma izhaja iz priimkov avtorjev iz Belgije: **Rijmen** in **Daemen**.
- **RIJNDAEL** je ponavljajoč (večkrožni) blokovni algoritem, število krogov šifriranja (od 10 do 14) pa je odvisno od dolžine bloka in dolžine ključa. V vsakem krogu šifriranja se izvaja štiri različne matematične operacije (ByteSub, ShiftRow, MixColumn, AddRoundKey).
- **AES (RIJNDAEL)** nudi najbolj zanesljiv simetrični šifrirni postopek in služi kot zamenjava za zastareli DES in 3DES.

39



Ostali simetrični šifrirni postopki

- **IDEA - International Data Encryption Algorithm** uporablja 128 bitni ključ, ki ga razdelimo na 52 ključev dolžine 16 bitov.
-  **Blowfish** algoritem uporablja različno dolge ključke od 32 do 448 bitov,
- **Skipjack** algoritem uporablja 80-bitni ključ. Implementiran je v šifrirnih napravah Clipper, zato je bil zelo dolgo tajen,
- **CAST** uporablja ključ z dolžino od 40-128 bitov,
- Kot finalni kandidati natečaja za **AES** (1997-98) so poleg zmagovalca **RIJNDAEL** nastopili še:
 - **MARS** (IBM)
 - **RC6** (RSA Security)
 - **TWOFISH** (Counterpane Systems)
 - **SERPENT**

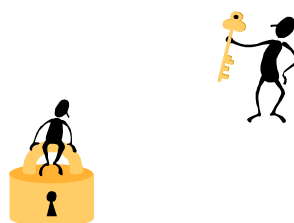


40

Varne komunikacije

4. Asimetrični šifrirni algoritmi

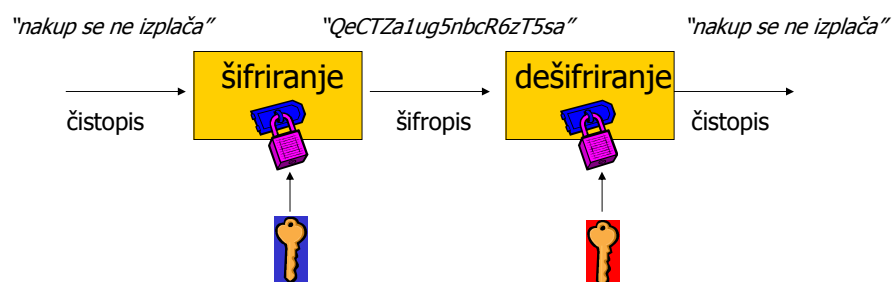
- Asimetrični šifrirni algoritem RSA
 - generacija parov ključev
 - šifriranje in dešifriranje
- Diffie-Hellmanov algoritem (DH)
- ElGamal-ov algoritem



41

Asimetrično šifriranje

- Ključa za šifriranje in dešifriranje nista enaka:



- Pošiljatelj šifrira sporočilo z **javnim** ključem prejemnika.
- Prejemnik dešifrira sporočilo z zasebnim (privatnim) **tajnim** ključem.
- Asimetrični postopki šifriranja so zasnovani na **enosmerni funkciji** s stranskim vhomom (one way trapdoor function).

42

Asimetrični šifrirni algoritem RSA

- RSA algoritem se imenuje po prvih črkah priimkov avtorjev (Ronald Rivest, Adi Shamir, Leonard Adleman), ki so razvili algoritem l. 1977
- Kot vsi asimetrični šifrirni postopki tudi RSA temelji na principu enosmerne funkcije.
- RSA izkorišča težavnosti faktorizacije velikih števil:
 - $15 = 3 * 5$
 - ali znate faktorizirati veliko število ?
109417386415705274218097073220403576120037329454492059909138421314763499842889
34784717997257891267332497625752899781833797076537244027146743531593354333897
=
102639592829741105772054196573991675900716567808038066803341933521790711307779
*
106603488380168454820927220360012878679207958575989291522270608237193062808643



43

RSA algoritem

- **Generacija ključev:**
 - izberemo dve veliki praštevili (p, q)
 - izračunamo produkt $n = p q$ (modul) ,
 - izračunamo produkt $\phi = (p-1) (q-1)$
 - izberemo število e , ki nima skupnega faktorja z ϕ
 - poiščemo število d tako, da daje produkt ($e d$) ostanek 1 pri deljenju z ϕ : $(e d) \bmod \phi = 1$
 - **javni ključ:** (n, e)
 - **tajni ključ:** (n, d)
- **Enkripcija in dekripcija:**
 - šifriramo dolge bloke čistopisa m , šifropis označimo z $E(p)$
 - šifriranje: $E(m) = m^e \bmod n$
 - dešifriranje: $D(E(m)) = E(m)^d \bmod n$
- Na osnovi znanega javnega ključa e , čistopisa m , šifropisa $c = E(m)$ v realnem času ni mogoče ugotoviti tajnega ključa d !!


44

Zgled generacije RSA ključev

- izberemo dve (veliki) praštevili (p, q) $p=17, q=31$
- izračunamo $n = p q$, $\phi = (p-1)(q-1)$ $n=527, \phi=480$
- izberemo število e , ki nima skupnega faktorja z ϕ , veljati mora $\gcd(e, \phi) = 1$; izberemo $e=61$ 
- poiščemo število d tako, da ima produkt $(e d)$ ostanek 1 pri deljenju z ϕ : $(e d) \bmod \phi = 1$
 - veljati mora enačba: $e d = k \phi + 1 \rightarrow d = (k \cdot 480 + 1) / 61$, pri tem pa mora biti k celo število:
 - $d = 7k + (53k + 1) / 61 = 7k + k_1 \rightarrow d = 181$ 
 - $k = (61k_1 - 1) / 53 = k_1 + (8k_1 - 1) / 53 = k_1 + k_2 \rightarrow k = 23$
 - $k_1 = (53k_2 + 1) / 8 = 6k_2 + (5k_2 + 1) / 8 = 6k_2 + k_3 \rightarrow k_1 = 20$
 - $k_2 = (8k_3 - 1) / 5 = k_3 + (3k_3 - 1) / 5 = k_3 + k_4 \rightarrow k_2 = 3$
 - $k_3 = (5k_4 + 1) / 3 = k_4 + (2k_4 + 1) / 3 = k_4 + k_5 \rightarrow k_3 = 2$
 - $k_4 = (3k_5 - 1) / 2 = k_5 + (k_5 - 1) / 2 = k_5 + k_6 \rightarrow k_4 = 1$
 - $k_5 = (2k_6 + 1)$, izberemo $k_6 = 0 \rightarrow k_5 = 1$

45

RSA šifriranje sporočila

- javni RSA ključ: $(n, e) = (527, 61)$ 
- sporočilo v čistopisu: $m = 40$
- šifriranje sporočila: $c = E(m) = m^e \bmod n$

$$c = 40^{61} \bmod 527$$

upoštevamo lastnost: $e = 61 = 1 + 4 + 8 + 16 + 32$

$$40^1 \bmod 527 = 40$$

$$40^2 \bmod 527 = (40^1 \bmod 527)^2 \bmod 527 = 19$$

$$40^4 \bmod 527 = (40^2 \bmod 527)^2 \bmod 527 = 361$$

$$40^8 \bmod 527 = (40^4 \bmod 527)^2 \bmod 527 = 152$$

$$40^{16} \bmod 527 = (40^8 \bmod 527)^2 \bmod 527 = 443$$

$$40^{32} \bmod 527 = (40^{16} \bmod 527)^2 \bmod 527 = 205$$

$$40^{64} \bmod 527 = (40^{32} \bmod 527)^2 \bmod 527 = 392$$

$$40^{128} \bmod 527 = (40^{64} \bmod 527)^2 \bmod 527 = 307$$


$$c = ((40^1 \bmod 527)(40^4 \bmod 527)(40^8 \bmod 527)(40^{16} \bmod 527)(40^{32} \bmod 527)) \bmod 527$$

$$c = (40 \cdot 361 \cdot 152 \cdot 443 \cdot 205) \bmod 527 = 350$$

- šifrirano sporočilo: $c = 350$

46

RSA dešifriranje sporočila

- tajni RSA ključ: $(n, d)=(527,181)$ 
- sporočilo v šifropisu: $c = 350$
- dešifriranje sporočila: $m = D(c) = c^d \bmod n$
 $m = 350^{181} \bmod 527$

upoštevamo lastnost: $d=181=1+4+16+32+128$

$$350^1 \bmod 527 = 350$$

$$350^2 \bmod 527 = (350^1 \bmod 527)^2 \bmod 527 = 236$$

$$350^4 \bmod 527 = (350^2 \bmod 527)^2 \bmod 527 = 361$$

$$350^8 \bmod 527 = (350^4 \bmod 527)^2 \bmod 527 = 152$$

$$350^{16} \bmod 527 = (350^8 \bmod 527)^2 \bmod 527 = 443$$

$$350^{32} \bmod 527 = (350^{16} \bmod 527)^2 \bmod 527 = 205$$

$$350^{64} \bmod 527 = (350^{32} \bmod 527)^2 \bmod 527 = 392$$

$$350^{128} \bmod 527 = (350^{64} \bmod 527)^2 \bmod 527 = 307$$

$$c = ((350^1 \bmod 527)(350^4 \bmod 527)(350^{16} \bmod 527)(350^{32} \bmod 527)(350^{128} \bmod 527)) \bmod 527$$

$$c = (350 * 361 * 443 * 205 * 307) \bmod 527 = 40$$

- dešifrirano sporočilo: $m = 40$

47

Diffie - Hellmanov algoritem

- glavna parametra sta lahko enaka za vse uporabnike:
 - (g, p) : "generator" g in veliko praštevilo p
- varnost DH algoritma temelji na težavnosti računanja diskretnega logaritma:
 - za vsako število $0 < n < p$ lahko najdemo potenco k , tako da velja:
 - $n = g^k \bmod p$
 - število k pa zelo težko poiščemo iz (n, g, p) !!
- A in B najprej vsak na svoji strani izbereta tajni ključ tk_a in tk_b
- A in B izračunata javna ključa jk_a in jk_b :
 - $jk_a = g^{tk(a)} \bmod p$
 - $jk_b = g^{tk(b)} \bmod p$
- A in B si izmenjata ključa $n_a \leftrightarrow n_b$ in izračunata skupni ključ:
 - uporabnik A izračuna sejni ključ: $tk_{Ab} = jk_b^{tk(a)} \bmod p$
 - uporabnik B izračuna sejni ključ: $tk_{Ba} = jk_a^{tk(b)} \bmod p$
- oba sejna ključa sta enaka: $g^{tk(b)tk(a)} \bmod p = g^{tk(a)tk(b)} \bmod p$

48

DH - primer generacije sejnega ključa

- parametra g in p sta: $g=23$, $p=31$
- A in B izbereta tajni ključ tk_a in tk_b
 - $tk_a = 9$
 - $tk_b = 3$
- A in B izračunata javna ključa jk_a in jk_b :
 - $jk_a = g^{tk(a)} \bmod p = 23^9 \bmod 31 = 27$
 - $jk_b = g^{tk(b)} \bmod p = 23^3 \bmod 31 = 15$
- A in B izmenjata javna ključa jk_a , jk_b in izračunata skupni ključ:
 - A: $tsk_{ab} = jk_b^{tk(a)} \bmod p = 15^9 \bmod 31 = 29$
 - B: $tsk_{ba} = jk_a^{tk(b)} \bmod p = 27^3 \bmod 31 = 29$

49

ElGamal-ov algoritem

- imenuje se po avtorju: Taher ElGamal
- varnost ElGamal algoritma temelji na težavnosti računanja diskretnega logaritma:
 - skupina uporabnikov izbere veliko praštevilo p in naključno število g
 - vsak uporabnik naključno izbere število x in izračuna par y :
 $y = g^x \bmod p$
 - tajni ključ sestavljajo števila (x, g, p)
 - javni ključ so števila (y, g, p)
 - če je p zelo veliko število, potem iz (y, g, p) zelo težko izračunamo eksponent x !!
- ElGamal algoritem za enkripcijo
- ElGamal algoritem za digitalni podpis

50

ElGamal-ovo šifriranje

- veliko praštevilo p in naključno število g sta javna
- uporabnika A in B naključno izbereta tajna ključa in generirata javna ključa:

- $jk_a = g^{tk(a)} \bmod p$
- $jk_b = g^{tk(b)} \bmod p$

- pošiljatelj A:



- izbere naključno število k , $\gcd(k, p-1)=1$
- na osnovi čistopisa m , naključnega števila k in javnega ključa prejemnika jk_b izračuna dvodelni šifropis (a, b) :

$$a = g^k \bmod p$$

$$b = (jk_b^k m) \bmod p$$

- prejemnik B:

- dešifriranje sporočilo na osnovi tajnega ključa prejemnika x :

$$m = b a^{p-1-tk(b)} \bmod p$$

51

ElGamal-ovo šifriranje - zgled

- javni števili za več uporabnikov: $p=31$ in $g=9$
- uporabnik B naključno izbere tajni ključ in generirata javni ključ:

- $tk_b = 3$
- $jk_b = g^{tk(b)} \bmod p = 16$

- pošiljatelj A:

- izbere naključno število $k=7$, $\gcd(7, 30)=1$
- na osnovi čistopisa $m=23$, naključnega števila k in javnega ključa prejemnika jk_b izračuna dvodelni šifropis (a, b) :

$$a = g^k \bmod p = 9^7 \bmod 31 = 10$$

$$b = (jk_b^k m) \bmod p = (16^7 23) \bmod 31 = 29$$

- prejemnik B:

- dešifriranje sporočilo na osnovi tajnega ključa prejemnika x :

$$m = b a^{p-1-tk(b)} \bmod p = 29 10^{27} \bmod 31 = 23$$

52

ElGamal-ov digitalni podpis

- izbran in objavljen je par števil (g, p)
- podpisnik izračuna dvodelni podpis sporočila m na osnovi svojega tajnega ključa (x, g, p) in naključnega števila k tako da:
 - izbere naključno število k , ki izpolnjuje pogoj $\gcd(k, p-1)=1$
 - izračuna prvi del podpisa a :
$$a = g^k \bmod p$$
 - In izračuna drugi del podpisa b :
$$m = (x a + k b) \bmod (p-1)$$
 - javni ključ podpisnika je $y = g^x \bmod p$
- prejemnik preverja digitalni podpis (a, b) sporočila m s pomočjo javnega ključa pošiljatelja y :
$$y^a a^b \bmod p = g^m \bmod p \quad ?$$



53

ElGamal-ov digitalni podpis - zgled

- izbran in objavljen je par števil $(g=2, p=11)$
- podpisnik izračuna dvodelni podpis sporočila $m=5$ na osnovi svojega tajnega ključa $(x=8, g=2, p=11)$ tako da:
 - izbere naključno število $k=9$, $\gcd(9, 10)=1$
 - izračuna prvi del podpisa a :
$$a = g^k \bmod p = 2^9 \bmod 11 = 6$$
 - izračuna drugi del podpisa b :
$$m = (x a + k b) \bmod (p-1) \rightarrow b=3$$
 - javni ključ podpisnika je $y = g^x \bmod p = 2^8 \bmod 11, y=3$
- prejemnik preverja digitalni podpis $(a=6, b=3)$ sporočila m s pomočjo javnega ključa pošiljatelja $y=3$:
$$y^a a^b \bmod p = g^m \bmod p \quad ?$$

levo: $y^a a^b \bmod p = 3^6 6^3 \bmod 11 = 10$

desno: $g^m \bmod p = 2^5 \bmod 11 = 10$

54

Varne komunikacije

5. Zgoščevalne funkcije (HASH)

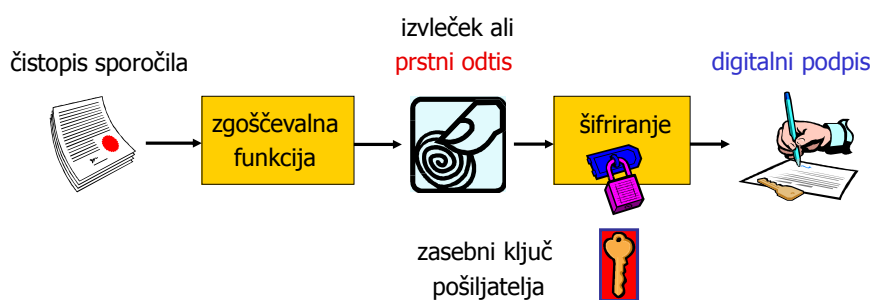
- Digitalni podpis in prstni odtis
- Zgoščevalne funkcije
 - MDC in MAC
 - razredi MDC
- Zgoščevalna funkcija na osnovi DES
- MD4
- MD5
- SHA-1, SHA-2 ...



55

Digitalni podpis

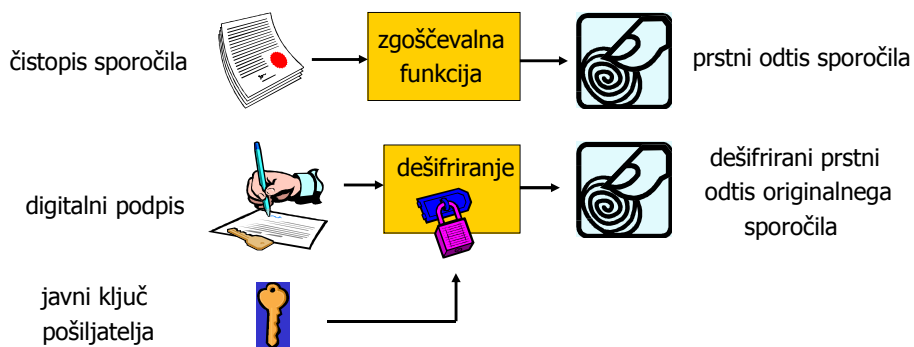
- Digitalni podpis je s tajnim ključem šifrirani **prstni odtis** sporočila:



- Zgoščevalna funkcija je enosmerna funkcija in vsaka sprememba čistopisa spremeni tudi prstni odtis sporočila.
- Napadalec bi lahko spremenil sporočilo in dodal nov prstni odtis !
- Pošiljatelj zaščiti prstni odtis s šifriranjem!

56

Preverjanje digitalnega podpisa

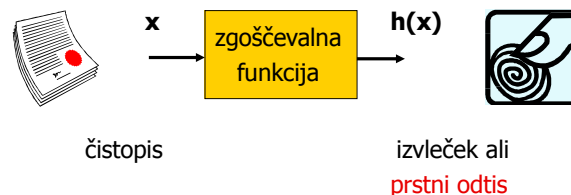


- Prejemnik preveri ujemanje prstnih odtisov in če sta enaka
 - je **sporočilo verodostojno**,
 - potrjena je **identiteta pošiljatelja** in
 - če velje oboje, potem **pošiljatelj ne more zanikati** sporočila.

57

Zgoščevalna funkcija

- Zgoščevalna funkcija (**hash function**) preslika poljubno dolgo sporočilo v blok podatkov končne dolžine. Izvleček (**digest**) imenujemo tudi **prstni odtis** (**digital fingerprint**) sporočila.
- Zgoščevalna funkcija je enosmerna funkcija.
- Verjetnost, da najdemo sporočilo z enakim prstnim odtisom mora biti zelo majhna $\Pr(h(x_1)=h(x_2)) \rightarrow 0$.



58

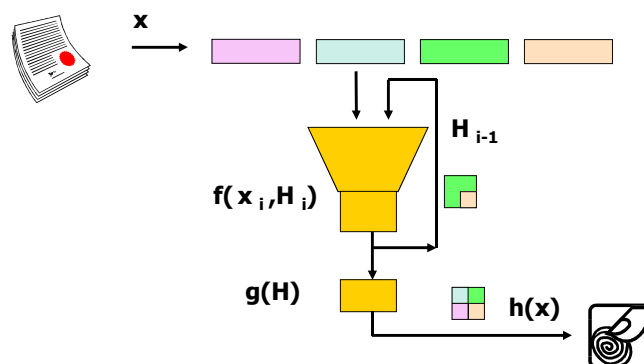
Funkcionalna delitev zgoščevalnih funkcij

- **MDC** zgoščevalne funkcije (**m**odification **d**etection **c**odes) omogočajo prepoznavo sprememb v sporočilu . Ključa ne potrebujemo .
- **MAC** zgoščevalne funkcije (**m**essage **a**uthentication **c**odes) zagotavljajo verodostojnost sporočila in avtentičnost pošiljatelja. Zgoščevalna funkcija uporablja tajni ključ zato se uporablja za MAC tudi ime keyed hash function.

59

Model iteracijske zgoščevalne funkcije

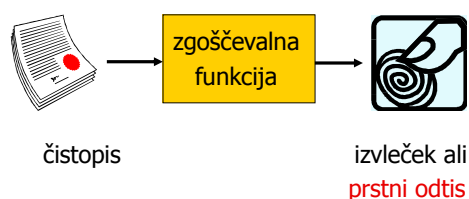
- Sporočilo razdelimo na bloke dogovorjene dolžine.
- Postopek zgoščevanja ponavljamo in vsakič uporabimo izvleček predhodnih blokov.



60

Razredi MDC zgoščevalnih funkcij

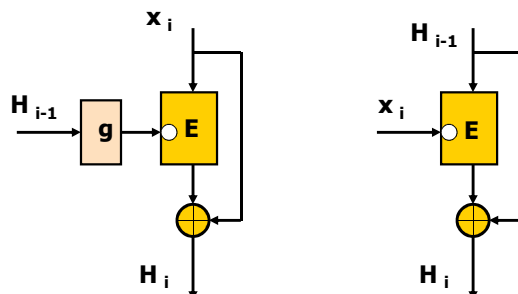
- MDC zgoščevalne funkcije razdelimo v tri razrede:
 - zgoščevalne funkcije, ki uporabljajo blokovne šifrirne postopke; (npr. DES - Davies-Meyer)
 - zgoščevalne funkcije, ki uporabljajo modularno aritmetiko (MASH 1 - Modular Arithmetic Secure Hash - algorithm 1);
 - posebej prilagojene (namenske) zgoščevalne funkcije odlikuje mnogo manjša računska zahtevnost; (najbolj znane so MD4, MD5, SHA1)



61

Zgoščevalna funkcija z blokovno šifro

- Množica zgoščevalnih funkcij temelji na enem od blokovnih šifrirnih postopkov, pogosto na DES-u.
- primer sta zgoščevalni funkciji z blokovno šifro:
 - (Matyas-Meyer-Oseas)
 - (Davies-Meyer)



62

Zgoščevalne funkcije MD

- Namenske MDC zgoščevalne funkcije so računsko bistveno bolj učinkovite od blokovnih. Najbolj pogoste funkcije temeljijo na MD algoritmu, ki ga je razvil Ron Rivest:
 - MD2 (message digest algorithm 2) , 1989
 - MD4 (message digest algorithm 4), 1990
 - MD5 (message digest algorithm 5), 1991
 - SHA - 1 (Secure Hash Algorithm 1) je bil razvit v NITS v sodelovanju z NSA in objavljen 1994
- Pogosto sta uporabljena zgoščevalna algoritma MD5 in SHA-1:
 - dolžina sporočil je "omejena" na 2^{64} bitov.
 - MD5 je računsko manj zahteven od SHA-1 (razmerje hitrosti 7 : 3)
 - algoritem SHA-1 ima daljši izvleček (160bit vs 128 bit)
 - algoritem SHA-1 je del standarda (DSS)
- Danes se uporablja varnejši SHA-2 , dolžina je 256-512 bitov
- Na poti je še novejši SHA-3 !!

63

Posodobitve SHA-1, 2, ..

Algorithm and variant	Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)	Rounds	Operations	Collisions found	Example Performance (MiB/s) ^[1]	
SHA-0	160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rot	Yes	-	
SHA-1	160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rot	Theoretical attack (2^{51}) ^[2]	153	
SHA-2	SHA-256/224	256/224	256	512	$2^{64} - 1$	32	64	+,and,or,xor,shr,rot	None	111
	SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80	+,and,or,xor,shr,rot	None	99

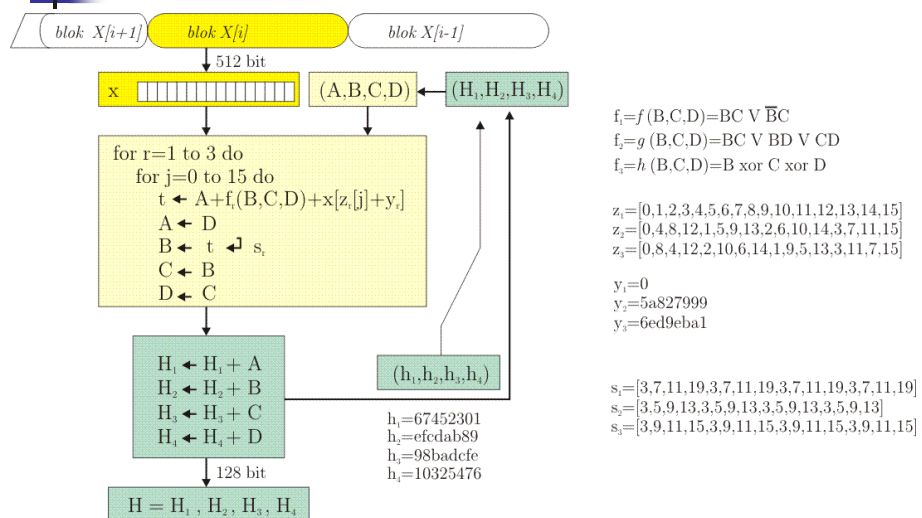
64

MD4

- b - bitno sporočilo X razdelimo na 512- bitne bloke $X[i]$
- izveček H ima dolžino 128 bitov (štiri 32- bitne besede)
- zgoščevanje vsakega bloka poteka v treh krogih
- v vsakem krogu uporabimo
 - različne funkcije: f, g, h
 - različni vrstni red branja besed v bloku $z(x)$
 - različne rotacije $s(x)$
- delni izveček $H[i]$ uporabimo pri obdelavi naslednjega bloka, v zadnjem bloku pa je to hkrati izveček celotnega sporočila H

65

MD4 algoritem



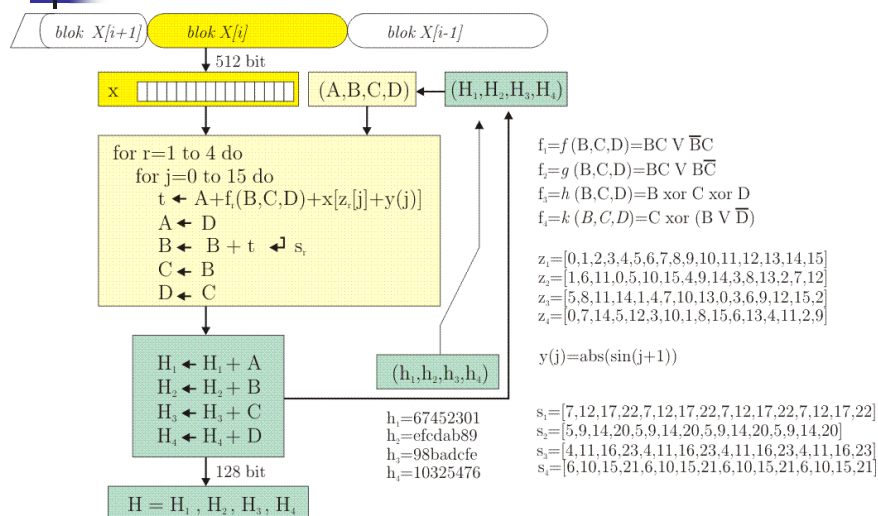
66

MD5

- b - bitno sporočilo X razdelimo na 512- bitne bloke $X[i]$
- izvaleček H ima dolžino 128 bitov (štiri 32- bitne besede)
- zgoščevanje vsakega bloka poteka v štirih krogih
- v vsakem krogu uporabimo
 - različne funkcije: f, g, h, e
 - različni vrstni red branja besed v bloku $z(x)$
 - različne rotacije $s(x)$
- delni izvaleček $H[i]$ uporabimo pri obdelavi naslednjega bloka, v zadnjem bloku pa je to hkrati izvaleček celotnega sporočila H

67

MD5 algoritem



68

Zgoščevalne funkcije na zgledu

testno sporočilo x : izvleček $h(x)$:

MD4	“” “a” “abc” “abcdefghijklmnopqrstuvwxy”	31d6cfe0d16ae931b73c59d7e0c089c0 bde52cb31de33e46245e05fbd6fb24 a448017aaf21d8525fc10ae87aa6729d d79e1c308aa5bbcdeea8ed63df412da9
MD5	“” “a” “abc” “abcdefghijklmnopqrstuvwxy”	d41d8cd98f00b204e9800998ecf8427e 0cc175b9c0f1b6a831c399e269772661 900150983cd24fb0d6963f7d28e17f72 c3fcd3d76192e4007dfb496cca67e13b
SHA-1	“” “a” “abc” “abcdefghijklmnopqrstuvwxy”	da39a3ee5e6b4b0d3255bfef95601890afd80709 86f7e437faa5a7fce15d1ddcb9eaeaea377667b8 a9993e364706816aba3e25717850c26c9cd0d89d 32d10c7b8cf96570ca04ce37f2a19d84240d3a89



69

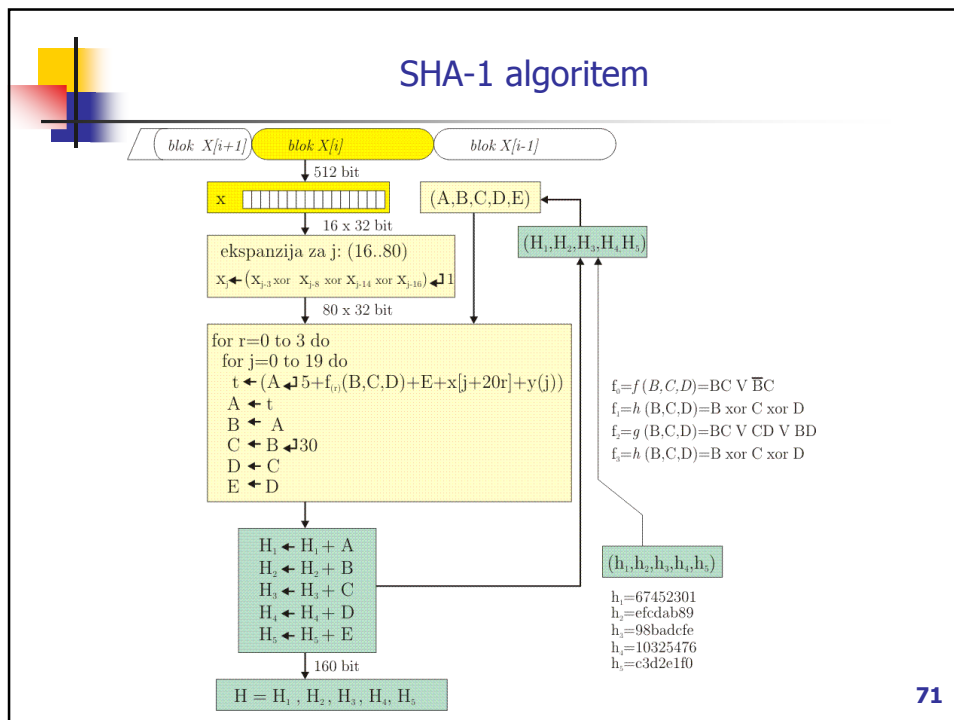
SHA-1

- b - bitno sporočilo X razdelimo na 512-bitne bloke $X[i]$
- blok 16 32-bitnih besed ekspaniramo na 80 besed (2560 bitov)
- zgoščevanje ekspaniranega bloka poteka s štirimi različnimi funkcijami: f, g, h, e
- delni izvleček $H[i]$ uporabimo pri obdelavi naslednjega bloka, v zadnjem bloku pa je to hkrati izvleček celotnega sporočila H
- izvleček H ima dolžino 160 bitov (pet 32-bitnih besed)

- SHA-1 (Secure Hash Algorithm 1) je posodobitev prvotnega algoritma SHA

70

SHA-1 algoritem



71

Digitalni podpis

- Digitalni podpis je šifrirani izveček sporočila. Potrebujemo ustrezno zgoščevalno funkcijo in asimetrični postopek šifriranja.
- Za digitalni podpis zelo pogosto uporabljamo kombinacijo:
 - zgoščevalna funkcija SHA-1
 - asimetrični šifrirni algoritem RSA
 - V standardnem formatu digitalnega certifikata X-509 je v enem polju zapisan tudi tip zgoščevalne funkcije in šifrirni postopek. V spletnem potrdilu sigen-ca je algoritem podpisa [sha1RSA](#).
- NITS je leta 1991 postavil standard za digitalni podpis [DSS](#).
- [DSS](#) predpisuje
 - zgoščevalno funkcijo [SHA-1](#) in
 - šifrirni algoritem [DSA](#) (Digital Signature Algorithm).
- [DSA](#) je poseben primer ElGamal algoritma in varnost prav tako temelji na težavnosti računanja diskretnega logaritma.

72

DSA generacija ključev

- števila (p,q,g) so lahko skupna za več uporabnikov:
- izberemo veliko praštevilo p
 - dolžina zapisa L je od 512 do 1024, L je mnogokratnik 64
- število q izberemo tako, da je $p-1$ deljiv z q (160-bitno število)
- generiramo število $g=h^{(p-1)/q} \bmod p$,
 - izberemo $h < p-1$, veljati mora $g > 1$
- javni ključ y generiramo s pomočjo naključno izbranega tajnega ključa x :
$$y = g^x \bmod p$$
- javni ključ sestavlja (p,q,g,y)
- tajni zasebni ključ je (p,q,g,x)

73

DSA digitalni podpis

- Pošiljatelj A želi podpisati sporočilo m
- $H(m)$ je izvleček sporočila m (SHA-1, 160 bit)
- A izbere naključno število $k < q$ in s pomočjo tajnega ključa (p,q,g,x) izračuna dvodelni digitalni podpis:
$$r = (g^k \bmod p) \bmod q$$
$$s = (k^{-1} (H(m) + x r)) \bmod q$$
- A pošlje podpisano sporočilo (m, r, s) prejemniku B
- Prejemnik B ima tudi dostop do javnega ključa pošiljatelja (p,q,g,y) in izračuna verifikacijsko sporočilo v po korakih:
$$w = s^{-1} \bmod q$$
$$u_1 = (H(m) w) \bmod q$$
$$u_2 = (r w) \bmod q$$
$$v = (g^{u_1} y^{u_2} \bmod p) \bmod q$$
- Prejemnik B preveri, če velja $v=r$?

74

DSA na zgladu

- Generirana so števila
 - $p=1291$,
 - $q=215$, $p-1=6q$
 - $g=1003$, $(h=1230)$
- Pošiljatelj A želi podpisati izvleček sporočila $H(m)=1155$
- A izbere naključno število $k=973$ in izračuna podpis:
 - $r=(g^k \bmod p) \bmod q = 52$
 - $s=(k^{-1} (H(m)+x r)) \bmod q = 118$
- A pošlje sporočilo in podpis ($r=52$, $s=118$) prejemniku B
- Prejemnik B izračuna $H(m)$ in verifikacijsko sporočilo v po korakih:
 - $w=s^{-1} \bmod q = 82$
 - $u_1=(H(m) w) \bmod q = 110$
 - $u_2=(r w) \bmod q = 179$
 - $v=(g^{u_1} y^{u_2} \bmod p) \bmod q = 52$
- Prejemnik B potrdi veljavnost sporočila, saj velja $v=r=52$

75

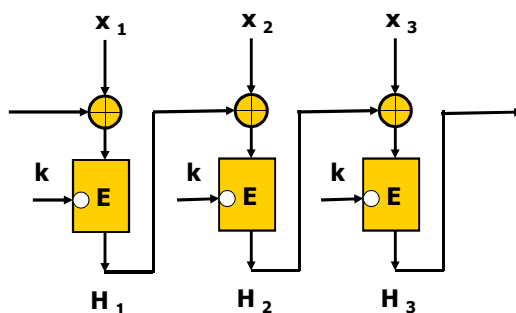
Zgoščevalne funkcije s ključem (MAC)

- Namen zgoščevalnih funkcij s ključem (keyed hash functions) je avtentikacija sporočila, ki nam zagotavlja integriteto sporočila in avtentikacijo izvora za datoteke poslane med dvema uporabnikoma (primer uporabe je SSL);
- Poznamo dve vrsti MAC funkcij:
 - MAC funkcije, ki temeljijo na bločnih šifrah (CBC-MAC)
 - MAC funkcije ki temeljijo na MD funkcijah. V praksi se v glavnem uporabljata dve vrsti teh funkcij:
 - HMAC-SHA-1 in
 - HMAC-MD5
 - Glavna razlika v primerjavi z bločnimi MAC funkcijami je prilagoditev funkcije stiskanja, ki zavisi od ključa k . Vse posredovalne iteracije vključujejo skrivni ključ. To zagotavlja dodatno varnost v primeru odkritja slabosti osnovne zgoščevalne funkcije

76

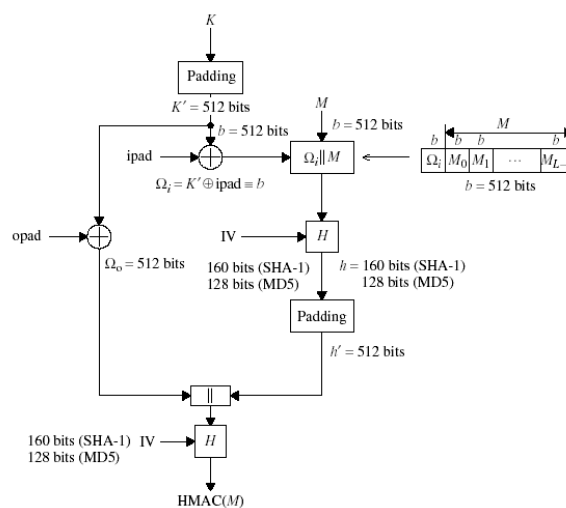
MAC zgoščevalne funkcije z bločno šifro

- Avtentikacijske zgoščevalne funkcije s ključem (MAC) pogosto uporabljajo verženje blokov - CBC (Cipher Block Chaining).
- Šifrirni algoritem je lahko DES:



77

HMAC



$$HMAC = H \left[(K \oplus opad) \parallel H \left[(K \oplus ipad) \parallel M \right] \right]$$

78

Varne komunikacije

- Šifriranje z javnimi ključi



- Digitalno potrdilo

- PGP
- X-509



79

Namen digitalnega podpisa

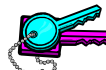
- Digitalni podpis dodajamo nešifriranemu sporočilu in zato ne zagotavlja tajnosti komunikacije.
- Pošiljatelj z digitalnim podpisom zagotovi:
 - verodostojnost sporočila,
 - potrjuje svojo identiteto in s tem
 - sprejme tudi odgovornost za sporočilo.
- Prejemnik lahko hkrati preveri verodostojnost in avtentičnost:
 - Ali je sprejeto sporočilo res enako oddanemu sporočilu ?
 - Ali nam sporočilo res pošilja predstavljeni pošiljatelj ?
- Če prejemnik potrdi verodostojnost sporočila in avtentičnost pošiljatelja, potem tudi pošiljatelj ne more sporočila zanikati:
 - Če se prstna odtisa ujemata, potem sporočilo ni bilo spremenjeno in podpisal ga je lahko le pošiljatelj, ki ima edini pravi zasebni ključ.
- Digitalni podpis omogoča zagotavljanje verodostojnosti, avtentičnosti in neovrgljivosti sporočil.



80

Uporaba zasebnih in javnih ključev

- Digitalni podpis temelji na asimetričnem šifrirnem postopku, ki uporablja parov imetnikovih ključev: javni ključ + zasebni ključ



- Vsak uporabnik nosi odgovornost za uporabo in varovanje **zasebnega ključa**. Dostop do tajnega ključa varujemo z dolgim geslom, ki ga imenujemo fraza. Uporabnik ne sme zaupati nikomur svojega zasebnega ključa. Če to stori, potem nosi tudi vso odgovornost za zlorabe.



- **Javni ključ** mora biti vsakomur dostopen z jamstvom, da pripada navedenemu uporabniku. V nasprotnem primeru lahko pride do problemov:
 - Problem lažne identitete: napadalec podtakne lažni javni ključ in dešifrira vsa preštržena sporočila.
 - Problem zanikanja identitete: pošiljatelj zanika lastno sporočilo.

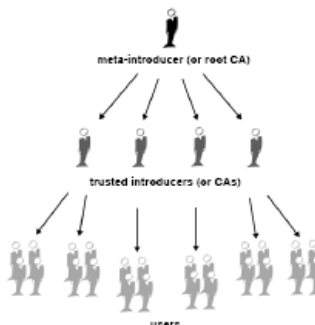
81

Model zaupanja

- Neposredno zaupanje:
 - uporabniki si v parih izmenjajo certifikate



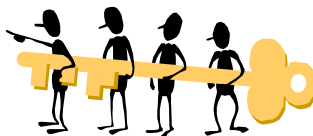
- Hierarhični model ima obrnjeno drevesno strukturo:
 - vrhovni overitelj - root CA upravlja digitalna potrdila overiteljev, ki ležijo en nivo nižje v strukturi
 - najnižji CA upravljajo s potrdili uporabnikov



82

Upravljanje s ključi

- Javni ključ mora nositi garancijo, da res pripada navedenemu uporabniku. **Overjanje javnih ključev** opravlja posebna služba (podobno notarju), ki skrbi tudi za upravljanje s ključi.
- **Urad za overjanje (CA=Certification Authority)** potrjuje verodostojnost javnih ključev z digitalnim podpisom odgovorne osebe. Imetnik javnega ključa se mora ob **registraciji** identificirati in s tem prevzema odgovornost za uporabo zasebnega ključa. Identifikacijo izvrši uradna oseba (**RA=Registration Authority**).
- Na zahteve imetnikov opravlja CA tudi **razveljavitve javnih ključev**. Potreba po preklicu javnega ključa nastopi v primeru izgube tajnosti zasebnega ključa.



83

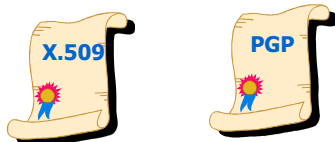
Digitalno potrdilo

- **Digitalno potrdilo** (digital certificate) je kopija javnega ključa, ki je overjena od tretje osebe ali institucije.
- Imetnik javnega ključa se mora ob registraciji identificirati in s tem prevzema tudi odgovornost za uporabo zasebnega ključa. Identifikacijo izvrši uradna oseba **RA** (Registration Authority).
- Urad za overjanje potrdil **CA** (Certification Authority) je nevtralna organizacija, ki ji uporabniki zaupajo.
- **Upravljanje z javnimi ključi** ne zajema samo shranjevanje digitalnih potrdil na strežniku, pač pa celoten postopek posrednih overjanj izdajateljev potrdil, razveljavitve javnih ključev itn.
- Infrastruktura javnih ključev **PKI** (Public Key Infrastructure) določa protokole in storitve pri upravljanju z javnimi ključi.

84

Format digitalnega potrdila

- Digitalno potrdilo vsebuje poleg javnega ključa tudi množico identifikacijskih podatkov uporabnika in izdajatelja potrdila.
- Najbolj znana formata sta X-509 in PGP:
 - ITU-T mednarodni standard predpisuje **X-509** format digitalnih potrdil. V opisu je določeno katere informacije so vsebovane v poljih potrdila in kakšen je njihov format zapisa.
 - X-509 v1 1988, osem polj
 - X-509 v2 1993, + dodani dve identifikacijski polji = 10 polj
 - X-509 v3 1996, + dodano polje za razširitve
 - PGP format digitalnega potrdila se uporablja v programskem paketu za varno izmenjavo podatkov **PGP** (Pretty Good Privacy). PGP je v začetku devetdesetih let ustvaril Phil Zimmerman.



85

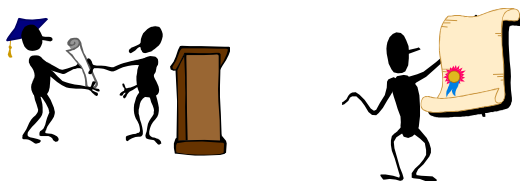
X.509 - demo

- Version: 3 (0x2)
- Serial Number: 0 (0x0)
- Signature Algorithm: sha1WithRSAEncryption
- Issuer: OU=Demo SI-CA, O=SETCCE, C=SI
- Validity
 - Not Before: Dec 20 11:49:01 2004 GMT
 - Not After : Dec 18 11:49:01 2014 GMT
- Subject: OU=Demo SI-CA, O=SETCCE, C=SI
- Subject Public Key Info:
 - Public Key Algorithm: rsaEncryption
 - RSA Public Key: (4096 bit)
 - **modulus (4096 bit): JAVNI KLJUČ**
 - X509v3 extensions:
 - X509v3 Basic Constraints: critical
 - CA:TRUE
 - X509v3 Subject Key Identifier:
B6:16:5E:27:5B:B2:2E:E4:CF:3A:83:71:7C:AF:4E:B8:EB:F6:22:3E X509v3
 - Key Usage: critical Certificate Sign, CRL Sign
- Signature Algorithm: sha1WithRSAEncryption

86

Pridobitev digitalnega potrdila

- Glavni overitelj digitalnih potrdil za pravne in fizične osebe je **SIGEN-CA** (Slovenian General Certification Authority)
- Spletno kvalificirano digitalno potrdilo pridobimo nekaj dni po oddaji izpolnjenega formularja na Upravni enoti ob identifikaciji z osebnim dokumentom.
- Digitalno potrdilo lahko med drugim uporabimo tudi za različne storitve na portalu **e-uprava**
 - oddaja vlog za upravne storitve,
 - oddaja obrazcev za dohodnine,
 - vpogled v osebne podatke centralnega registra prebivalstva ..



87

Kvalificirani izdajatelji certifikatov v Sloveniji ?

- <http://www.si-ca.si/>
- <http://www.si-ca.si/kripto/kr-cert.htm>
- <http://e-uprava.gov.si/e-uprava/dogodkiPrebivalci.euprava?zdid=780&sid=244>
- <http://postarca.posta.si/>
- .
- .

88



Varne komunikacije

7. Varnostni mehanizmi na Internetu

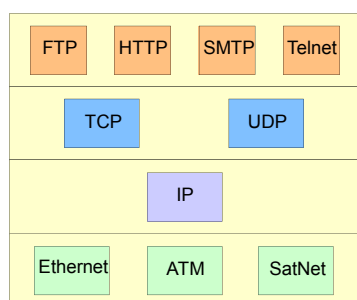


Varne komunikacije po Internetu

- **Plasti** in protokoli
- Varnost na **omrežni plasti**
 - IPsec
 - VPN
- Varnost na **transportni plasti**
 - SSL
 - TSL
- Varnost na **aplikacijski plasti**
 - varna elektronska pošta
 - S/MIME
 - PGP

Plasti in protokoli

osnovni internetni protokoli:



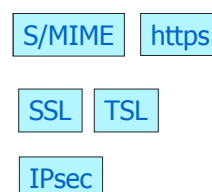
Aplikacijska plast

Transportna plast

Internetna plast

Računalnik/omrežje

dodani varnostni protokoli:



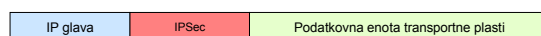
91

IPsec

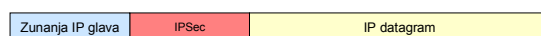
IP Security (IPsec) omogoča **varovanje na omrežni plasti**.

IPsec deluje na dva možna načina:

- IPsec **transportni način** ohranja glave IP paketov nespremenjene, šifrira se samo vsebina paketa
- IPsec **tunelski način** dodaja novo glavo IP paketom, stara glava in vsebino paketa pa se prenašata v šifrirani obliki. Varovana komunikacija poteka med parom prehodov (gateway to gateway), ki jih naslavljaajo dodane glave IP paketov.



Transportni način



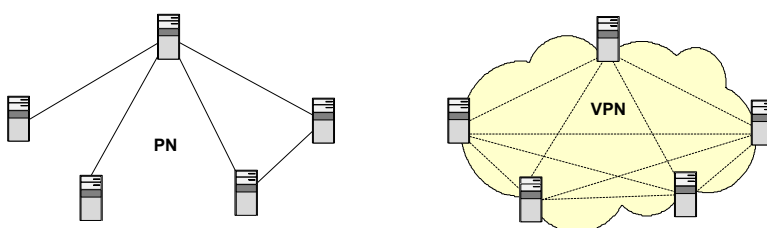
Tunelski način

92

VPN

VPN je navidezno zasebno omrežje (Virtual Private Network)

- Povezave v navideznih zasebnih omrežjih so dinamične in navidezne. Potekajo kot tuneli po javni TK infrastrukturi.
- Varo komunikacijo zagotavlja tunelski protokol, na primer IPsec.



93

Svetovni splet

- Svetovni splet = WWW (World Wide Web) je porazdeljen informacijski sistem. Sestavlja ga množica spletnih strani na strežnikih, ki so povezani v Internet.
- HTTP (Hypertext Transfer Protocol) je protokol za prenos spletnih strani med spletnim strežnikom in brskljalnikom. HTTP deluje na aplikacijski plasti. HTTP deluje podobno kot FTP in SMTP:
 - prenaša datoteke podobno kot FTP
 - sporočila med strežnikom in klientom so podobna kot pri SMTP,
 - HTTP prenaša sporočila direktno, SMTP pa po principu shrani in pošlji naprej
- HTML (Hypertext Markup Language) je programski jezik, ki ga uporabljamo za opis spletnih strani. HTML uporablja samo ASCII znake, kar omogoča največjo možno prilagodljivost.

94



Varna komunikacija po spletu

- **SSL (Secure Socket Layer)** je razvil Netscape za varno komunikacijo med spletnim klientom in strežnikom. SSL podpira preverjanje identitete strežnika. V komunikaciji se za vsako sejo ustvari varni kanal. SSL zagotavlja varno komunikacijo **na transportni plasti**.
- **TLS (Transport Layer Security)** je standardizirana (IETF) zamenjava za SSL.
- TLS_v1 in SSL_v3 sta si zato zelo podobna, razlike so sicer zelo majhne, vendar nista interoperabilna

95



SSL, TLS

- SSL in TLS imata dve plasti: handshake in record
 - **Handshake** protokol določa vrsto sporočil za dogovor varnostnih parametrov, ki se bodo uporabili za podatkovni prenos v seji.
 - Klient in strežnik dogovorita **uporabo varnostnih mehanizmov** ob izmenjavi prvih sporočil: Client Hello , Server Hello

Poziv klienta= Client Hello:

ClientVersion 3,1

ClientRandom[32]

SessionID: None (new session)

Suggested Cipher Suites:

TLS_RSA_WITH_3DES_EDE_CBC_SHA TLS_RSA_WITH_DES_CBC_SHA Suggested

Compression Algorithm: NONE

Odgovor strežnika = Server Hello:

Version 3,1

ServerRandom[32]

SessionID: bd608869f0c629767ea7e3ebf7a63bdcffb0ef58b1b941e6b0c044acb6820a77

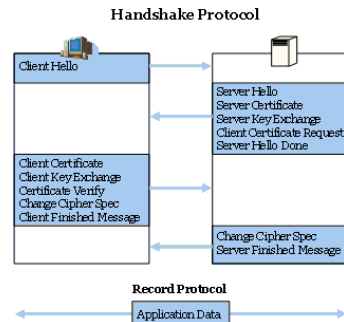
Use Cipher Suite: **TLS_RSA_WITH_3DES_EDE_CBC_SHA**

Compression Algorithm: NONE

96

SSL, TLS

- Strežnik pošlje digitalno potrdilo z javnim ključem, ki služi za avtentikacijo strežnika in za šifriranje sporočil. Klient tako lahko preveri tudi ujemanje imena strežnika. Strežnik zahteva avtentikacijo klienta (opcija).
- Klient na zahteva pošlje svoj certifikat in za tem še z javnim ključem strežnik šifrirano sporočilo ki omogoča generacijo glavnega ključa. Klient podpiše izveček vse komunikacije in s tem omogoči lastno avtentikacijo = Certificate verify. Klient potrdi, da bo nadaljna komunikacija potekala šifrirano z dogovorjenimi parametri.
- Strežnik obvesti klienta, da bo nadaljna komunikacija šifrirana v skladu z dogovorom
- **Record protokol** obdeluje podatkovni niz aplikacijske plasti. Podatkovni niz se razdeli na označene podatkovne bloke in obratno združuje v pravi vrstni red. Sledi lahko kompresija blokov, za tem pa šifriranje blokov. Verodostojnost podatkov se preverja z računanjem MAC.



97

Varna komunikacija po spletu

- <https> ni poseben protokol, pač pa pomeni da HTTP poteka preko varne transportne plasti: SSL, TLS !
- **S-HTTP** (Secure Hypertext Transfer Protocol) je poseben protokol, ki deluje na aplikacijski plasti. Namesto ustvarjanja varnega kanala kot pri SSL se pri S-HTTP šifrira vsako sporočilo posebej. S-HTTP podpira dvosmerno preverjanje identitete.

98



Elektronska pošta

- **SMTP** je protokol za izmenjavo elektronske pošte med poštnimi strežniki (**S**imple **M**ail **T**ransfer **P**rotocol)
- SMTP modul (strežnik) sprejme sporočilo o naslovu prejemnika in ga preko FTP pošilja SMTP modulu naslovljenega strežnika. Ko se identificira naslov prejemnikovega poštnega strežnika, se poštno sporočilo usmerja po internetnem protokolu (TCP/IP).
- Na strani prejemnikovega strežnika se poleg SMTP uporablja še dodatni protokol, ki omogoča delovanje uporabnikovega poštnega nabiralnika:
 - **POP3** (**P**ost **O**ffice **P**rotocol ver.3) ali
 - **IMAP** (**I**nternet **M**essage **A**ccess **P**rotocol)
- Uporabnik elektronske pošte uporablja SMTP protokol za pošiljanje in POP3 ali IMAP za sprejemanje poštnih sporočil.
- **MIME** (**M**ultipurpose **I**nternet **M**ail **E**xtension) je dodatni protokol za izmenjavo podatkov, ki niso v ASCII formatu. MIME določa nabor funkcij za pretvorbo v ASCII in obratno.

99



Varna elektronska pošta

- **PEM** (**P**rivacy **E**nhanced **M**ail) je standard za varno elektronsko pošto. PEM določa način šifriranja pri izmenjavi e-pošte. PEM uporablja CA certifikate.
- **MOSS** (**M**IME **O**bject **S**ecurity **S**ervice) je zamenjava standarda PEM, ki nudi povezavo med poštnimi naslovi in certifikati. MOSS omogoča tudi varno izmenjavo prionk v elektronski pošti.
- **S/MIME** (**S**ecure/**M**ultipurpose **I**nternet **M**ail **E**xtentions) je varna izboljšava standarda za format elektronske pošte MIME.
 - S/MIME uporablja X-509 infrastrukturo javnih ključev
 - S/MIME je zelo prilagodljiv in omogoča uporabo različnih simetričnih in asimetričnih šifrirnih postopkov
- **PGP** (**P**retty **G**ood **P**rivacy) zagotavlja
 - tajnost s šifriranjem sporočil
 - avtentičnost z digitalnim podpisom

100



Varne komunikacije

8. Varnost v mobilnih komunikacijah

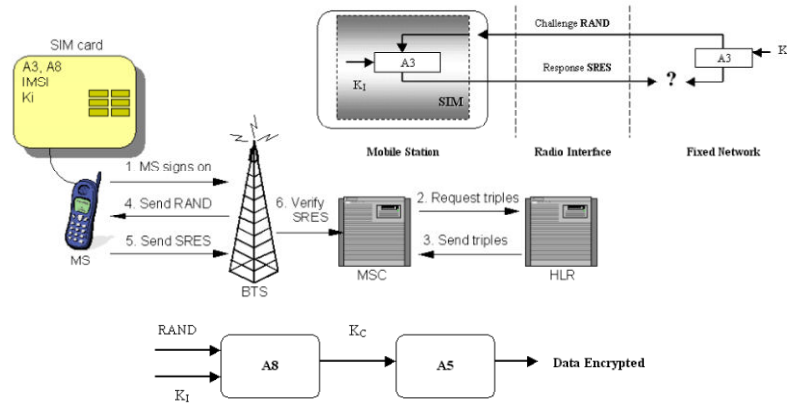


Varnost v mobilnih komunikacijah

- Varnost komunikacij v komercialnih mobilnih omrežjih
 - GSM
- Varnost v profesionalnih mobilnih omrežjih
 - TETRA

Varnostni mehanizmi v GSM

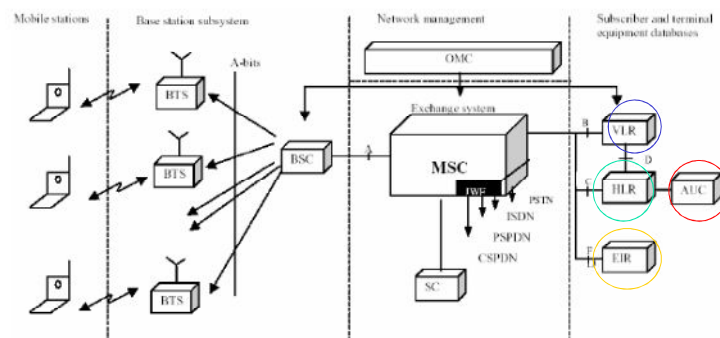
- tajni ključ K_i je shranjen na SIM kartici in varovan z dostopnimi kodami (PIN, PUK) in se ne prenaša po radijskem kanalu
- identiteta uporabnika je v komunikaciji prikrita: TMSI - IMSI
- avtentikacija mobilne postaje s strani omrežja
- komunikacija na radijskem delu zveze je šifrirana



103

Podatkovne baze v arhitekturi GSM

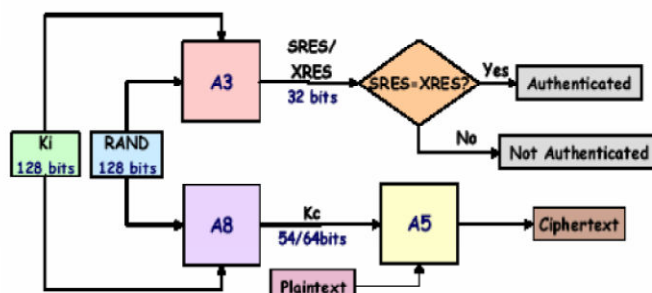
- HLR : vsi admin. podatki reg. naročnikov vključno s trenutno lokacijo
- VLR : podatki uporabnikov ki so zunaj domačega omrežja
- EIR : sezname vseh IMEI, ki imajo dovoljen, opazovan ali prepovedan dostop do omrežja
- AUC : baza identifikacijskih podatkov hrani IMSI, TMSI, LMI in tajni šifrirni ključ K_i



104

Algoritmi v GSM

- Varnostni mehanizmi so bili razviti v tajnosti in algoritmi A3, A5 in A8 niso bili javno objavljeni (security by obscurity ?)
- Ob vsakem klicu se generira nov šifrirni ključ Kc.
- A5-1 je pretočna šifra, ki uporablja tri LFSR z dolžinami 19,22 in 23 bitov ☹



105

Možni napadi v GSM

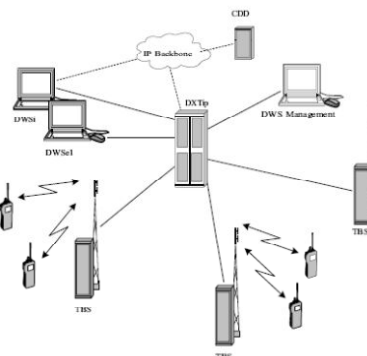
- Kraja ključa ? kloniranje SIM je sicer težavno vendar mogoče s "specialno" opremo.
- Tajnost algoritmov poraja upravičene dvome o varnosti ...
- Napad na pretočni šifrirni algoritem A5, ki se uporablja za šifriranje komunikacij na radijskem linku je dokazano uspešen! Ključ je dolg samo 54 bitov ...
- Ni vzajemne avtentikacije: omrežje preverja identiteto uporabnika, uporabnik pa ne preverja identitete omrežja ! Mogoč je napad na sredini z lažno BS: (TE <-> LBS <-> BS)
- Varovana je le komunikacija po radijskem linku med terminalom in bazno postajo ! Signalizacijsko omrežje operaterja ni zavarovano (SS7). Najbolj nevaren je napad z dostopom do omrežja operaterja (npr. uspeli nepooblaščeni dostop do administrativne baze uporabnikov = HLR ☹)
- *Za povprečnega uporabnika je GSM dovolj varen, saj našete pomankljivosti zaenkrat še ne pomenijo resnih možnosti prisluškovanja s strani radovednih sosedov !!*

106

Profesionalno radijsko omrežje TETRA

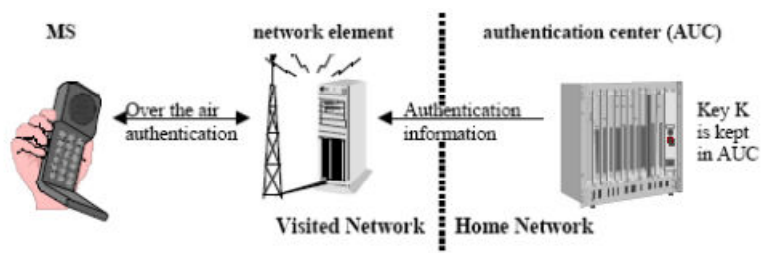
Varnostni mehanizmi:

- zagotavljanje avtentičnosti :
vzajemna avtentikacija
 - avtentikacija terminala (uporabnika)
 - avtentikacija omrežja
- zagotavljanje tajnosti : šifriranje komunikacij
 - na radijskem kanalu
 - šifriranje med koncema zveze



107

Vzajemna avtentikacija uporabnika in omrežja



Ali je pravo omrežje ?



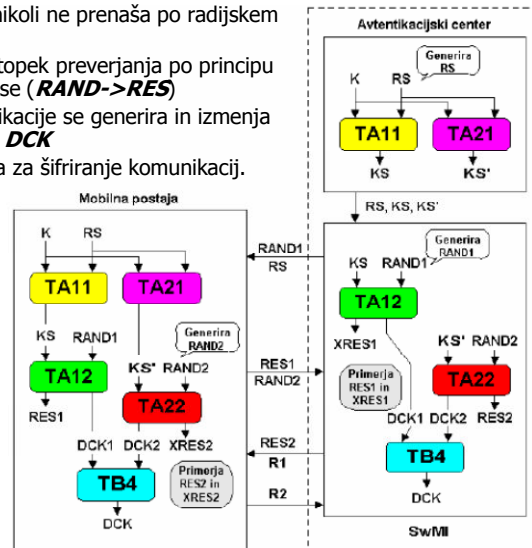
Ali je pravi uporabnik ?



108

Algoritmi za avtentikacijo

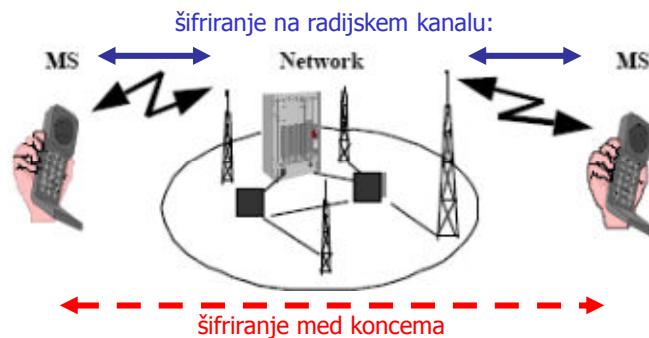
- Tajni ključ K se nikoli ne prenaša po radijskem kanalu!
- Uporablja se postopek preverjanja po principu challenge-response ($RAND \rightarrow RES$)
- V procesu avtentikacije se generira in izmenja skupni tajni ključ DCK
- DCK se uporablja za šifriranje komunikacij.



109

Varovanje tajnosti komunikacij v TETRA

- Šifriranje radijskega vmesnika A/E je zaščita pred zunanjim napadalcem. Uporablja se pretočne šifrirne algoritme TEA.
- Omrežje TETRA podpira tudi šifriranje med koncema zveze $E2E$. Za šifriranje med koncema se uporablja blokovne šifrirne algoritme (npr. IDEA, AES ..)
- Razpoložljivost, zanesljivost in **varnost** so glavne odlike profesionalnih celičnih omrežij !



110