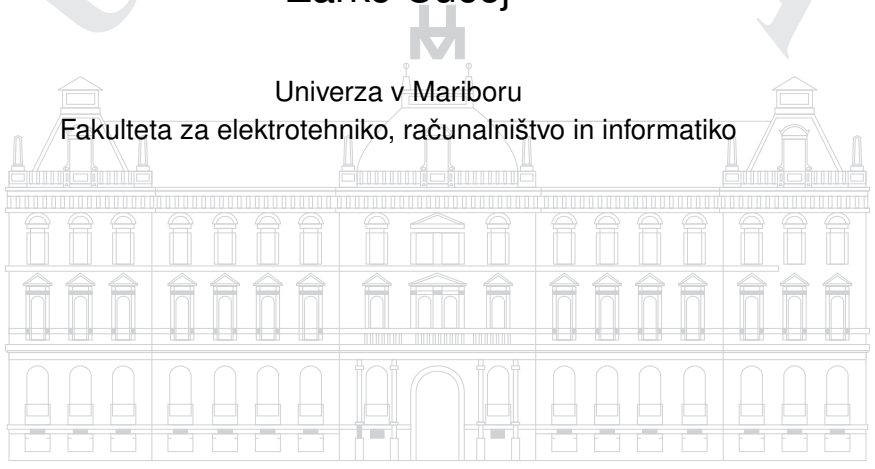
Internetne tehnologije

Tehnike šifriranja sporočil

Žarko Čučej

Univerza v Mariboru
Fakulteta za elektrotehniko, računalništvo in informatiko



Maribor 10. april 2009

Vsebina

Tehnike šifriranja

Internet se je iz preprostega sistema za prenos datotek razvil v sofisticirani sistem, ki ga uporabljamo na primer pri nakupih avtomobilov, izpolnjevanju receptov, vlogah za bančna posojila, plačilo računov itd, na kratko, postaja ena najpomembnejših infrastruktur tudi za poslovanje.

Podjetja so kmalu spoznala, da za poslovanje preko Interneta morajo zagotoviti varnost svojega omrežja in varnost komunikacije preko Interneta, da jim bodo stranke zaupale.

Tudi vlade so kmalu spoznale, da morajo svojim državljanom zagotoviti zasebnost tudi pri uporabi sodobnih informacijsko komunikacijskih tehnologij, predvsem pri uporabi njihovih osebnih podatkov, s katerimi razpolagajo in jih uporabljajo institucije kot so banke, bolnice, zavarovalnice itd.

Uporabljane kratice

CBS-DES	Cipher Block Chaining – Data Encryption Standard
DES	Data Encryption Standard
HMAC	Hash-based Message Authentication Code
MD5	Message Digest 5
RADIUS	Remote Authentication Dial-In User Service

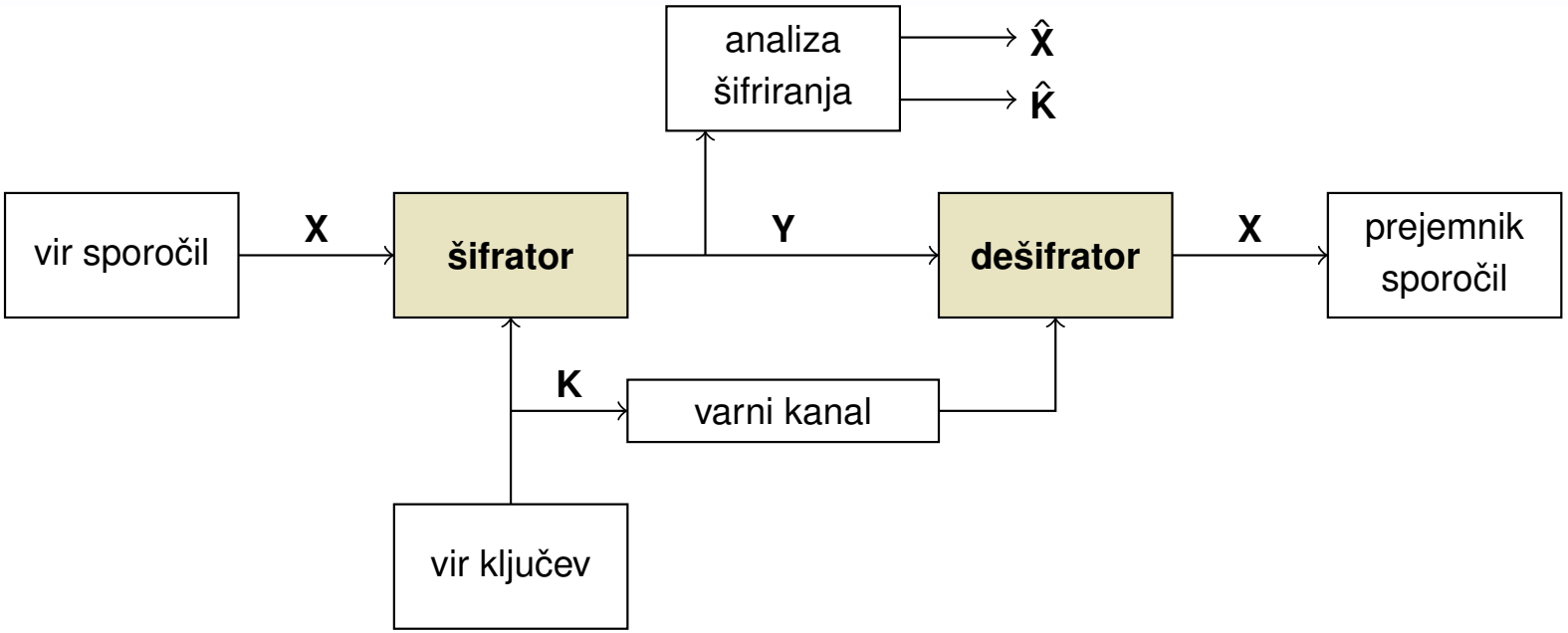
Zasebnost s klasično enkripcijo

- slovar:

originalno sporočilo:	**plaintext**
šifrirano sporočilo:	**ciphertext**
šifrator:	**encrypter**
dešifrator:	**decrypter**
šifrirni ključ:	**key**

- delovanje (slika 1):
  - šifrator šifrira sporočilo skladno z instrukcijami, ki jih dobi od vira šifrirnih ključev
  - po varnem kanalu/poti se ključ prenese/dostavi sprejemni strani, kjer dešifrator skladno z instrukcijami v sprejetem ključu sporočilo dešifrira

Zasebnost s klasično enkripcijo (2)



Slika: Model klasičnega ali simetričnega šifriranja. **X**: sporočilo, **Y**: šifrirano sporočilo, **K**: šifrirni ključ, $\hat{\mathbf{X}}$ ocena sporočila, $\hat{\mathbf{X}}$ ocena ključa.

Zasebnost s klasično enkripcijo (3)

- simbolični zapis algoritma:

$$\mathbf{Y} = E_K(\mathbf{X}) \tag{1}$$
$$\mathbf{X} = D_K(\mathbf{Y}) \,, \tag{2}$$

kjer sta:

E_K: algoritem šifriranja
D_K: algoritem dešifriranja

- Postopek šifriranja določa ključ **K**, ki mora biti znan tudi prejemniku, da lahko šifrirano sporočilo dešifrira.
- Postopke dešifriranja mora biti inverzen postopku šifriranja:

$$E_K = D_K^{-1} \,. \tag{3}$$

Algoritem DES

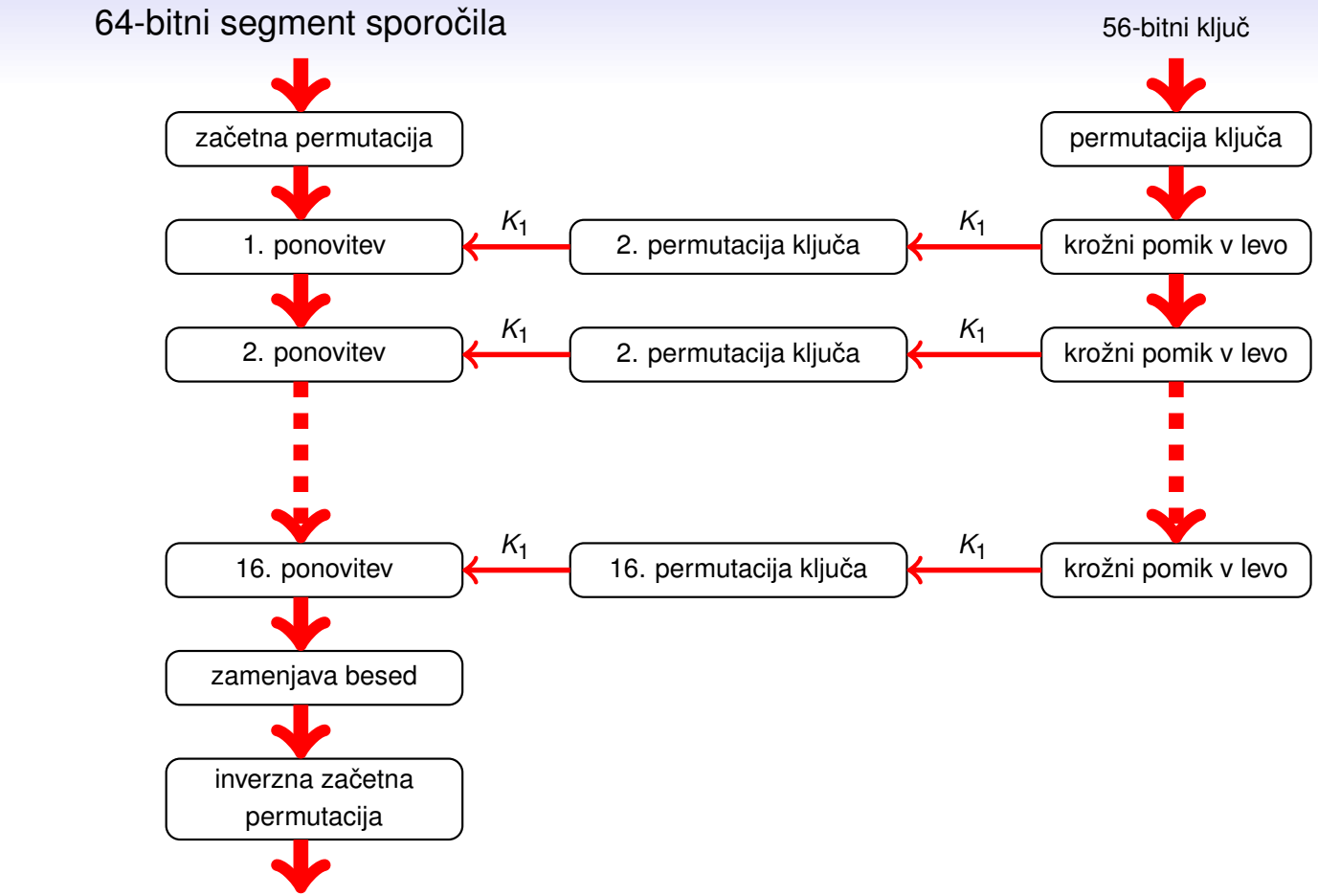
- Standard za šifriranja podatkov je leta 1997 sprejel **National Bureau of Standards**, danes **National Institut of Standards and Technology** (NIST), njegova uradna oznaka je FIPS PUB46 (Federal Information Processing Standard 46)
- Prvotni standard FIPS PUB46 določa 56 bitni ključ, novejše verzije standarda pa 128-bitne, 512-bitne in daljše ključe.
- šifriranje poteka v treh fazah (slika 2):

  prva faza: 64 bitni segmenti grejo skozi začetni permutacijski proces, ki ga določa ključ 56 bitov dolg ključ.

  druga faza: Sledi 16 ponovitev prve faze, pri katerih se pri vsaki ponovitvi ključ krožno premakne v levo. V dobljenem rezultatu še zamenjata prva in druga polovica segmenta svoji mesti.

  tretja faza: Izvede se inverzna začetna permutacija.

Algoritem DES (2)



Slika: Blokovna shema delovanja algoritma DES.

Izvajanje algoritma DES

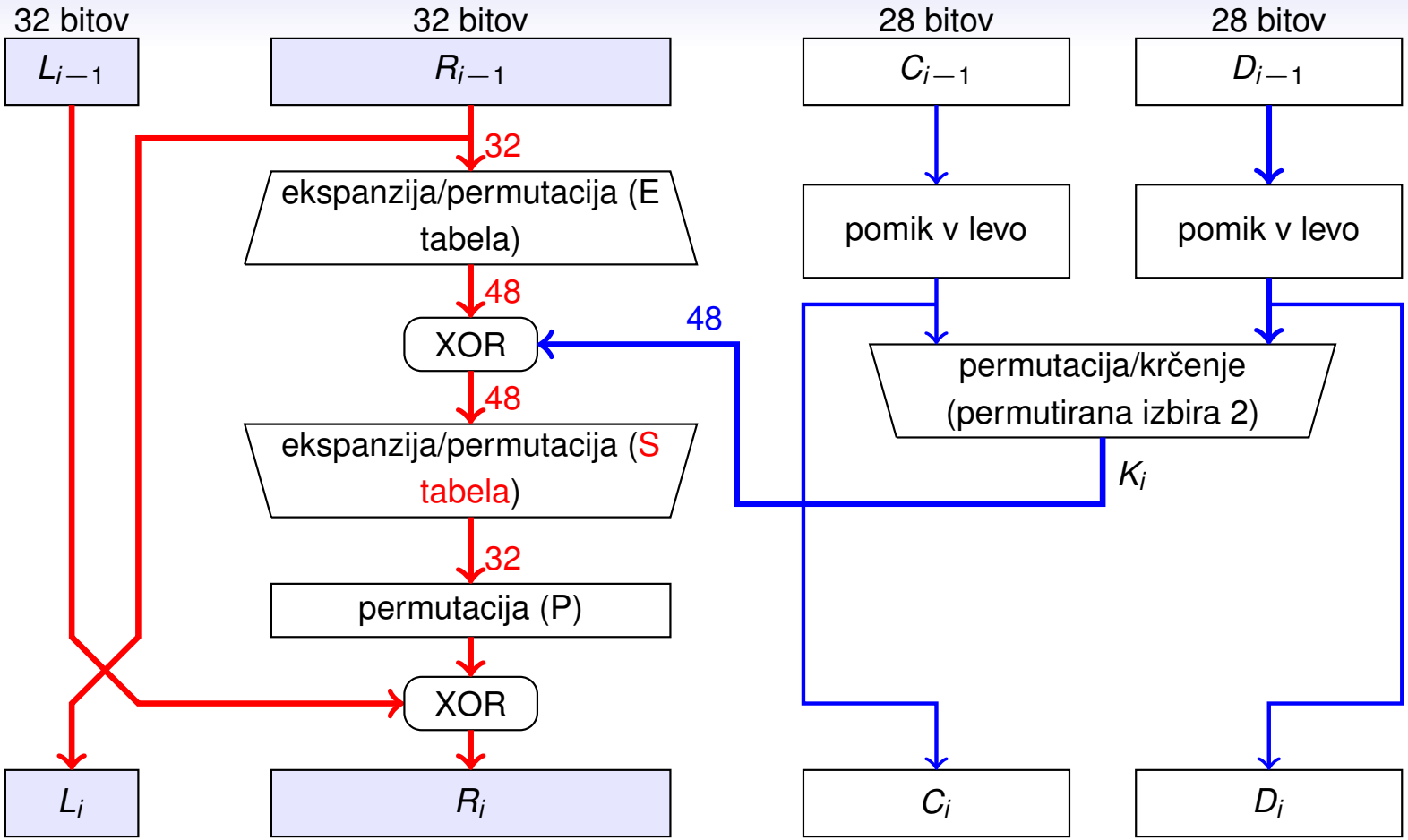
- Vsaka iteracija v algoritem DES je kompleksna funkcija (slika 3), formalno jo opišemo z:

$$L_i = R_{i-1} \tag{4}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \tag{5}$$

- Leva polovica rezultata permutacije (L_i) je enaka desni polovici predhodne permutacije (R_{i-1}).
- Vse permutacije ključev se naredijo na isti način, vendar se dobljeni sub-ključi K_i medsebojno razlikujejo zaradi krožnih premikov začetne permutacije ključa (slika 2).

Izvajanje algoritma DES (2)



Slika: Ena iteracija algoritma DES.

Moč DES

- Na moč šifriranja vplivata narava algoritma in dolžina ključa:
  - permutacije v DES se izvedejo s pomočjo posebnih permutacijskih tabel (“S-škatel), ki so (bile) tajne
  - leta se je sumilo, da imajo tabele slabost, ki omogočajo obstoj “super ključa” ki odklene vse šifre . . .
  - dolgoletne raziskave so pokazale sicer marsikatero neregularnost in (in s tem) nepričakovano obnašanje tabel, vendar do sedaj nobenemu ni uspelo dokazati očitanih jim slabosti
- Resnejša skrb je dolžina ključa:
  - 56-bitni ključ generira $7,6 \times 10^{16}$ ključev
  - za razbitje ključev so potrebni zelo veliki resursi
  - Diffi ter Hellman (izumitelja javnih ključev) in kasneje Wiener so pokazali, kako se naredi stroj, ki razbije šifro, in koliko bi to stalo (v času in denarju)

Trojni DES

- Uporablja dva ključa in tri šifriranja z algoritmom DES:

$$C = E_{K_1}[D_{K_2}[E_{K_1}[P]]] \tag{6}$$

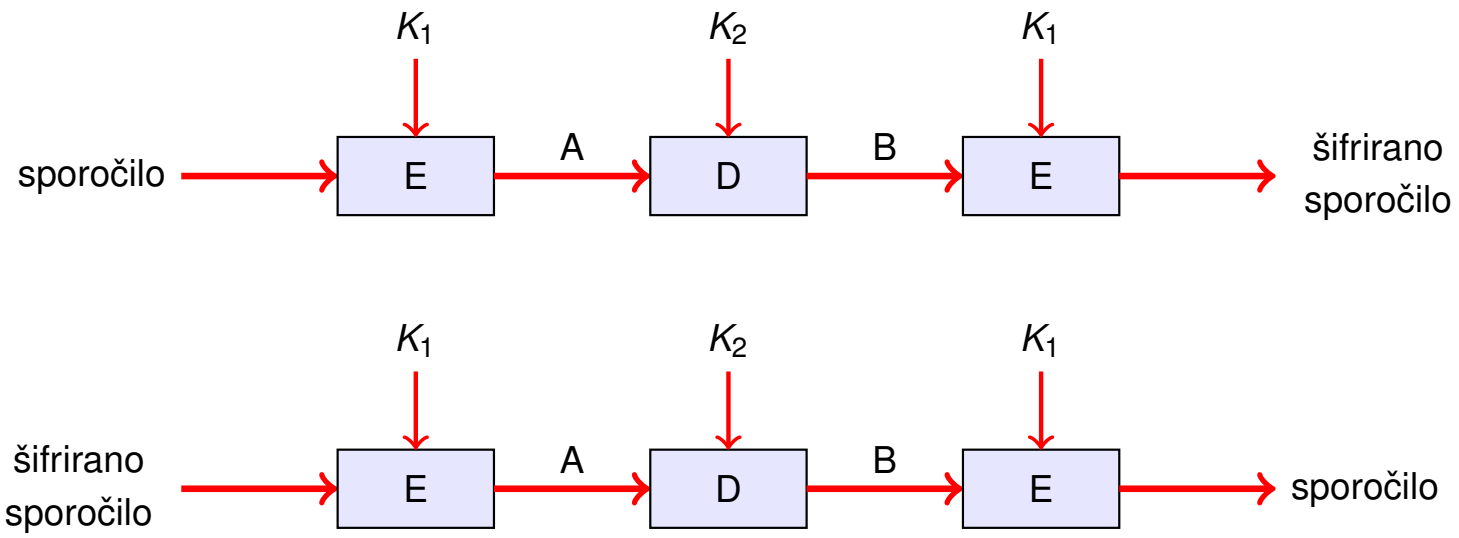
kjer so E_K: algoritem šifriranja s prvim ključem
D_K: algoritem dešifriranja z drugim ključem
P: sporočilo, ki se ga šifrira

- Vmesno dešifriranje z drugim ključem le malo prispeva k moči šifriranja, edina prednost je, da omogoča uporabnikom trojnega DES dešifriranje sporočil šifriranih z enojnim DES:

$$C = E_{K_1}[D_{K_1}[E_{K_1}[P]]] = E_{K_1}[P] \tag{7}$$

- Čeprav sta le dva ključa, se mora algoritme DES izvesti trikrat (slika 4)

Trojni DES (2)

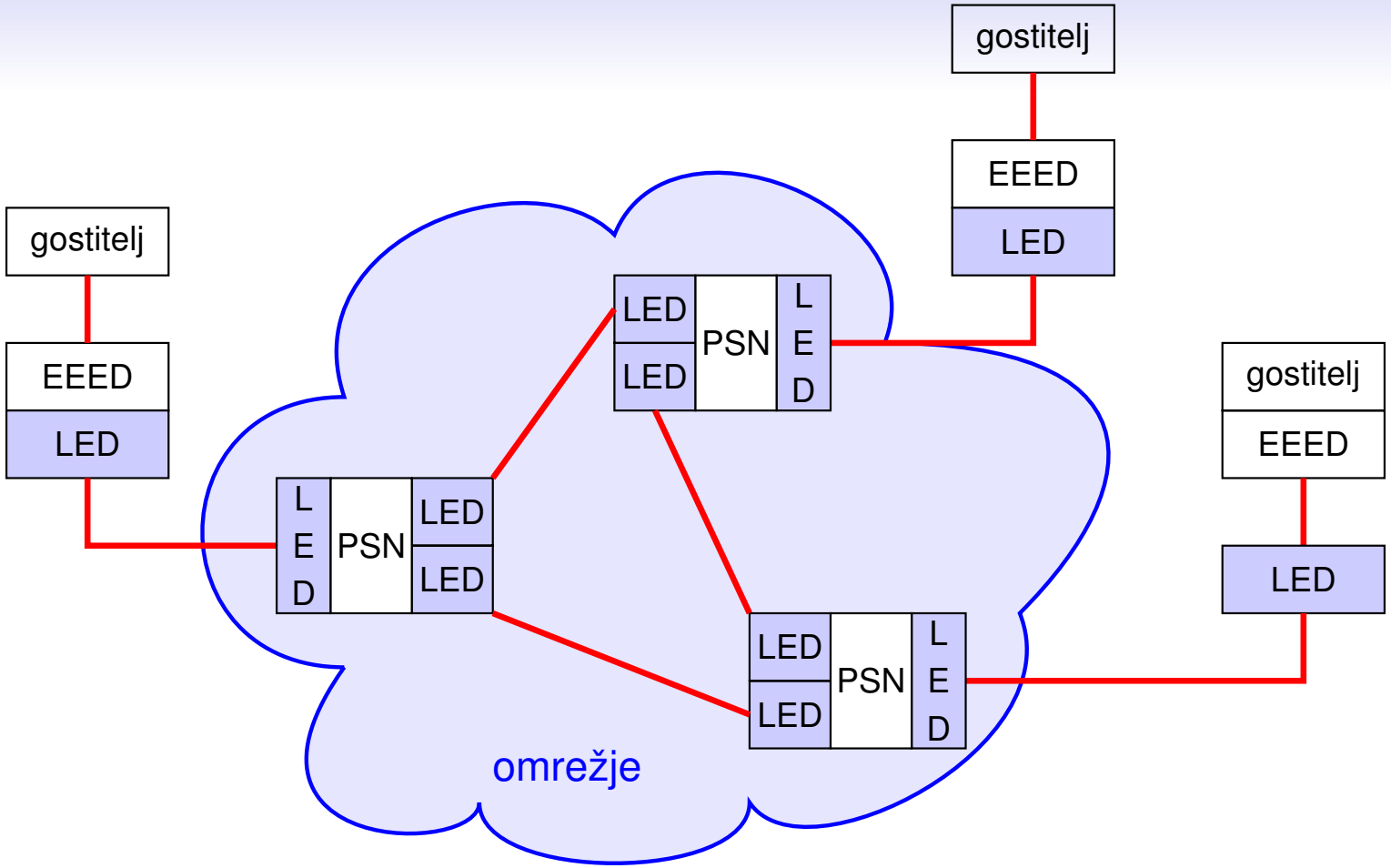


Slika: Avtomatsko razdeljevanje ključev pri povezavno orientiranih protokolih. E: šifriranje (enkripcija), D: dešifriranje (dekripcija)

Lokacije šifratorjev

- Obstajata dve osnovni možnosti namestitve šifratorjev:
  - v povezave med stikali ali usmerjevalniki
  - na izhod gostitelja uporabnika
- Čeprav prva možnost zahteva veliko število naprav – posebej pri velikih omrežjih – so njene prednosti očitne. Slabosti:
  - zaradi dešifriranja v stikalih/usmerjevalnikih so paketi v stikalih/usmerjevalnikih ranljivi
  - v javnih omrežjih uporabnik nima nadzora nad varnostjo v stikalih
- Druga možnost potrebuje manj naprav, sporočila so varovana tudi v stikalih/usmerjevalnikih, za varnost skrbijo uporabniki.
- Slabosti druge rešitve je mehanizem usmerjanja paketov po omrežju, ki zahteva, da šifriranje sporočila ne sme zajeti glav paketov. Zato se je vzorec prometa nezaščiten.

Lokacije šifratorjev (2)



Slika: Šifriranje sporočil v omrežju s stikali L2. EEED: End-to-End Encryption Device, LED: Line Encryption Device, PSN: Packed Switching Network.

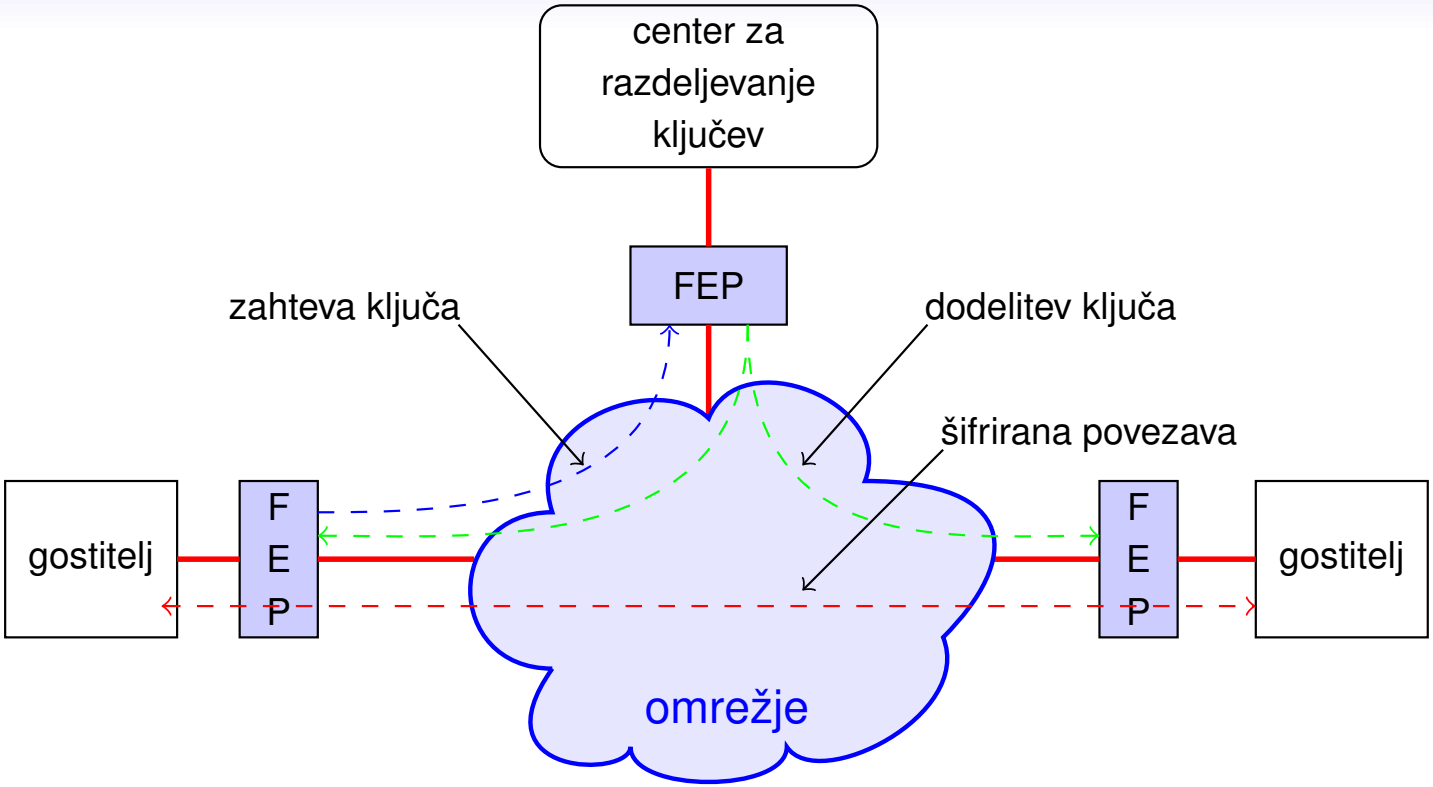
Razdeljevanje ključev

- Moč šifriranega sistema je odvisna od tehnike razdeljevanja ključev med komunikacijskimi partnerji.
- A in B lahko prideta do istega ključa na naslednje načine:
  1. ključ izbere A in ga fizično pošlje B
  2. ključ izbere tretji partner in ga fizično dostavi A in B
  3. partnerja lahko stare ključe zamenjata z novimi tako eden pošlje s starim ključem šifrirano sporočilo, ki vsebuje novi ključ
  4. partnerja A in B imata šifrirano povezavo s tretjim partnerjem, ki jima po tej poti pošlje ključa za medsebojno komunikacijo
- Ročno dostava ključev v 1 in 2 ni primerna za razprostranjena omrežja, ne za sisteme s pogostimi menjavami ključev.
- Način 3 ni dovolj varen. Če je razkrit stari ključ, zamenjava ključa ni več zaščitena.
- Uporabno razdeljevanje ključev (tako) nudi le način 4.

Stalni in začasni ključi

- Za razdeljevanje ključev preko s šifriranjem zaščitene povezave s tretjim partnerjem potrebujemo dve vrsti ključev (slika 6):
  - Začasni ključ Za izvedbo seje, v kateri partnerja izmenjata podatke, se vzpostavi navidezno zasebno omrežje, ki je varovano z začasnim le za to sejo. Po seji se ključ uniči.
  - Stalni ključ ki se uporablja le za izmenjavo začasnih ključev.
- Sistem omogoča:
  - Center za razdeljevanje ključev. Ta center določa, kateri partnerji v omrežju medsebojno lahko medsebojno vzpostavljajo varne (navidezno zasebne) povezave. Tako povezavo jim odobri z dodelitvijo začasnega ključa .
  - Čelni procesor. Čelni procesor ima dve nalogi: (i) na zahtevo uporabnika pri centru za razdeljevanje ključev pridobi začasni ključ, (ii) izvaja šifriranje sporočil, ki se izmenjujejo med čelnimi procesorji.

Stalni in začasni ključi (2)



Slika: Avtomatsko razdeljevanje ključev pri povezavno orientiranih protokolih.

Avtentikacija

- Za sporočilo, datoteko, dokument ali drugo zbirka podatkov rečemo, da je avtentično (originalno), kadar resnično in dejansko pride iz vira, za katerega trdi, da prihaja.
- Avtentikacija sporočila je postopek, ki omogoča komunicirajočima partnerjema preveriti, da je sprejeto sporočilo avtentično.
- Pri avtentikaciji ugotavljamo:
  - ali se je (med prenosom) vsebina sporočila spremenila
  - ali je vir sporočila avtentičen
- Pomembna je tudi časovna sled sporočila – iz nje ugotavljamo, ali je bilo sporočilo med prenosom umetno zakasnjeno oziroma ponovno poslano.

Avtentikacijski kod

Avtentikacijski kod

- Avtentikacijski kod je blok podatkov, ki ga ustvarimo iz sporočila s postopkom, ki je odvisen od tajnega ključa. Ta blok podatkov imenujemo tudi avtentikacijski kod sporočila.
- Za ugotavljanje avtentičnosti sporočila moramo poznati ključ s katerim je bil narejen avtentikacijski kod.

MAC: Message Authentication Code

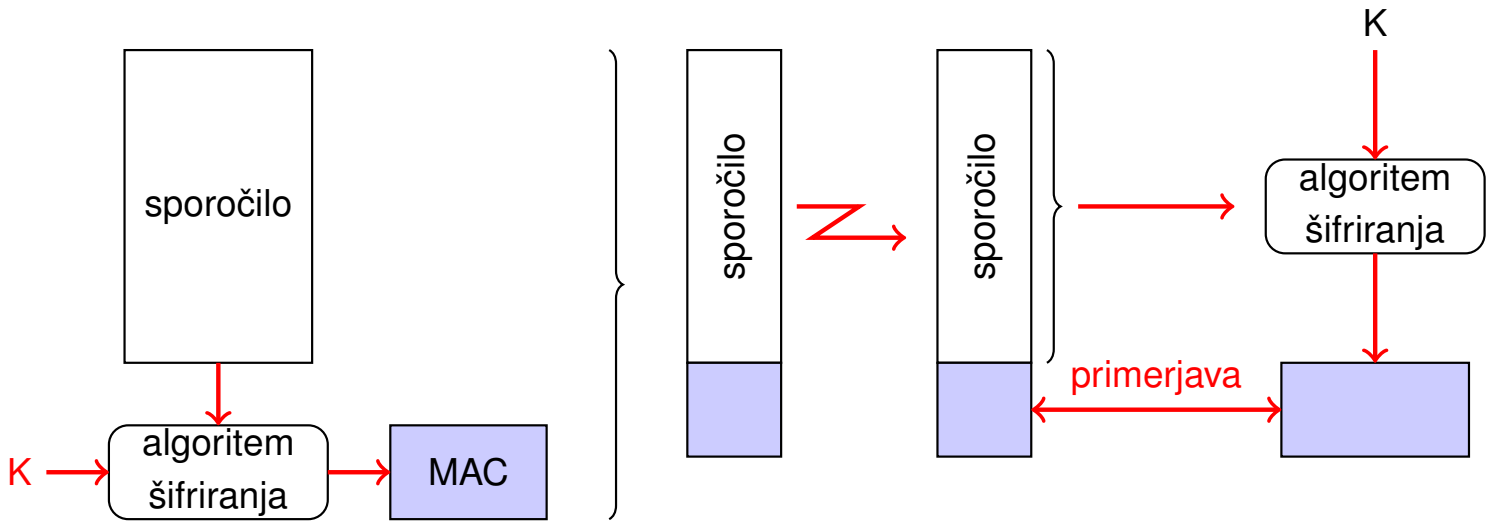
- Postopek avtentikacije z MAC (slika 7):
  - s tajnim ključem izračunamo MAC:

$$\mathrm{MAC}_M = F(\mathrm{K}_{AB}, M) \tag{8}$$

    kjer sta K_{AB} tajni ključ, ki ga morta poznati pošiljatelj (A) in prejemnik (B) in M sporočilo
  - sporočilu pripnemo MAC in ga pošljemo prejemniku sporočila

Avtentikacijski kod (2)

- prejemnik z enakim ključem izračuna MAC sprejetega sporočila ter ga primerja s sprejetim
- če se oba MAC ujemata, je sporočilo avtentično



Slika: Avtomatsko razdeljevanje ključev pri povezavno orientiranih protokolih.

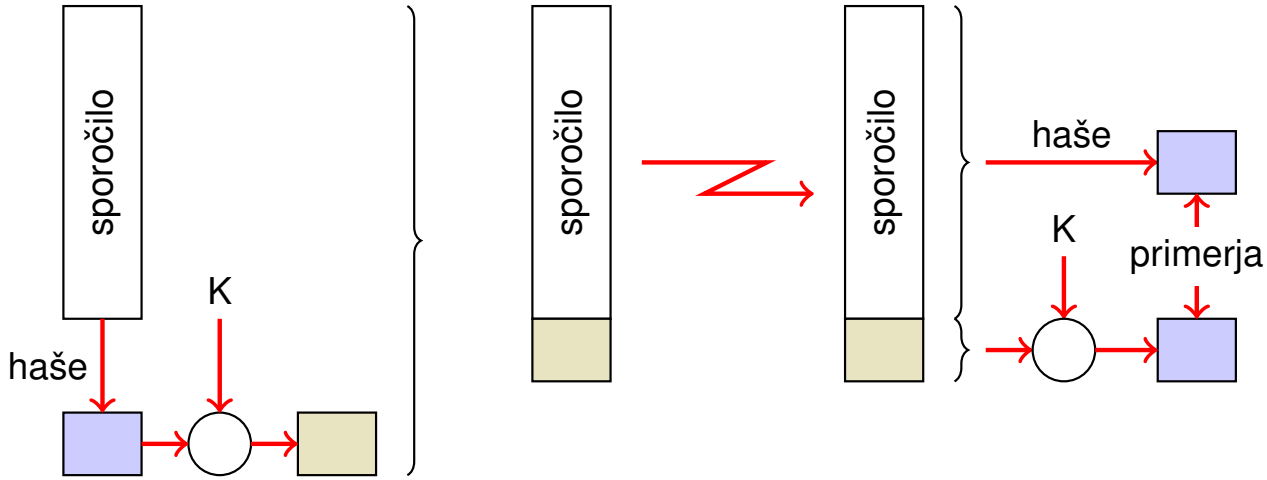
- tipična dolžina MAC je 16 ali 32 bitov

Avtentikacijski kod (3)

- Za generiranje MAC se uporabljajo številni algoritmi, ameriški biro za standardizacijo priporoča uporabo algoritma DES:
  - z DES se šifrira sporočilo
  - zadnji del šifriranega teksta se uporabi za MAC
- Avtentikacije z razliko šifriranja razlikuje ne zahteva, da je avtentikacijski algoritem reverzibilni, kar je pogoj pri za dešifriranje.
- Odsotnost reverzibilnosti daje avtentikacijskim algoritmom večjo moč kot jo imajo algoritmi sa šifriranje.

Enosmerne haše funkcije

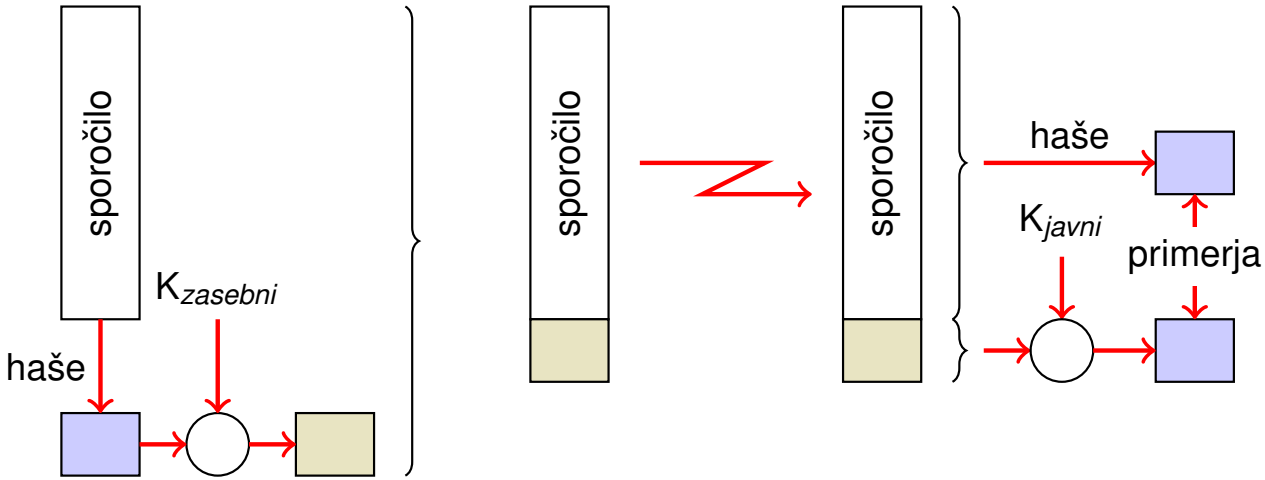
- Haše funkcije – ime imajo po rezultatu svojega učinka: sesekljanju sporočila – lahko ustvarijo šifrirani povzetek sporočila, ki ga uporabimo pri avtentikaciji sporočila.
- Avtentikacijo s haše funkcijo lahko naredimo s klasičnim postopkom šifriranja (slika 8).



Slika: Avtentikacije sporočil z enosmerno haše funkcijo in klasičnim šifriranjem.

Enosmerne haše funkcije (2)

- Uporaba javnih ključev pri avtentikaciji ima dve pomembni prednosti pred klasičnim šifriranjem:
  - poleg avtentikacije omogoča še digitalni podpis
  - ne zahteva razdeljevanje ključev med komunicirajočimi partnerji



Slika: Avtentikacije sporočil z enosmerno haše funkcijo in uporabo javnih ključev.

Enosmerne haše funkcije (3)

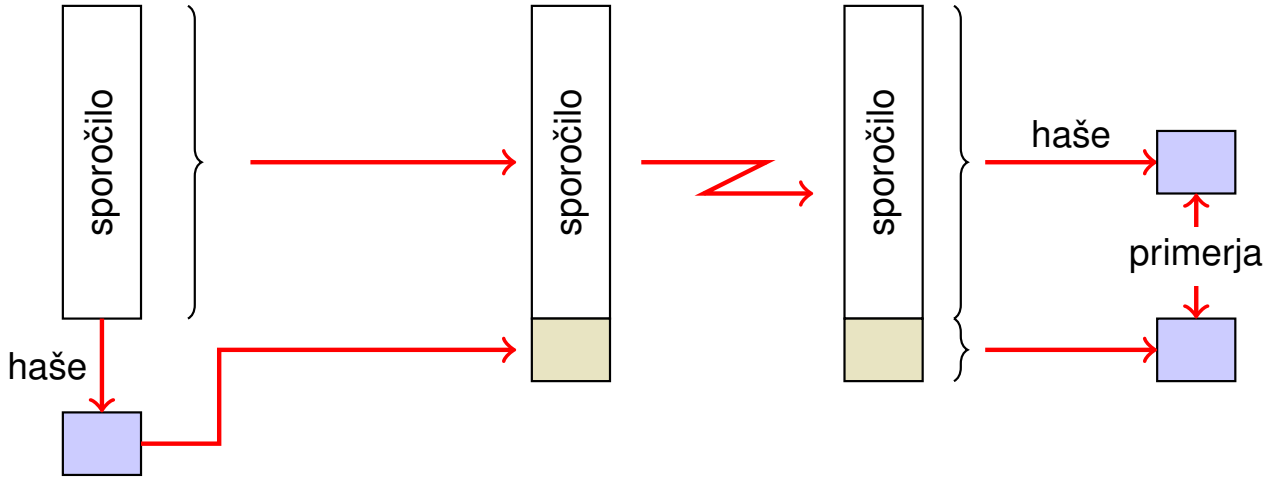
- Avtentikacija s haše funkcijo ne šifrira sporočila, vendar so razlogi proti avtentikaciji s šifriranjem:
  - šifriranje je počasno, tudi pri šifriranju izvlečkov
  - cene naprave za šifriranje niso zanemarljive
  - naprave za šifriranje so optimizirane za velike količine podatkov
  - mnogi šifrirni algoritmi so patentirani . . .
  - nekateri – iz strateških razlogov – nimajo izvoznega dovoljenja.
- Rešitev – uporabe skrivne vrednosti:
  - Partnerja (A in B) imata skupno skrivnost – vrednost S_{AB}.
  - A izračuna haše:

$$MD_H = H(S_{AB} \| M) \,, \tag{9}$$

kjer $\|$ označuje, da je bil pri računanju k sporočilu M pripeta skrivna vrednost S_{AB}.

Enosmerne haše funkcije (4)

- A pošlje k B sporočilo s pripetim izračunanim izvlečkom $[M||MD_M]$ (slika 10).



Slika: Avtentikacije sporočil z enosmerno haše funkcijo in tajne vrednosti.

  - Ker B pozna S_{AB}, lahko izračuna $H(S_{AB}||M)$ in preveri MD_M.

- Dokler S_{AB} ostane skrivna, sporočila ni mogoče ponarediti.

Zahtevane lastnosti haše funkcij

- Haše funkcija H ustvari “prstni odtis” sporočila . . .
- Za uporabo v avtentikaciji mora H imeti naslednje lastnosti:
  1. uporablja se lahko na poljubni velikosti bloka podatkov
  2. rezultat mletja (delanja hašeja) ima fiksno dolžino
  3. za katerokoli kod m je neizvedljivo odkriti x tako, da velja $\mathrm{H}(x) = m$
  4. za katerikoli blok x je neizvedljivo odkriti $y \neq x$ z lastnostjo $\mathrm{H}(x) = \mathrm{H}(y)$
  5. neizvedljivo je odkriti par (x, y) za katerega velja $\mathrm{H}(x) = \mathrm{H}(y)$
- Prve tri lastnosti so pomembne za praktično izvedbo haše funkcije.
- Četrto lastnost imenujemo enosmerna lastnost, ki je lahko pri avtentikaciji s skrito vrednostjo, resna pomanjkljivost:

Zahtevane lastnosti haše funkcij (2)

- Iz prestreženega promet se lahko izlušči sporočilo M in haše

$$\mathrm{MD}_H = \mathrm{H}(S_{AB} \| M) \; . \tag{10}$$

  Z obratom haše funkcije dobimo

$$S_{AB} \| M = \mathrm{H}^{-1}(\mathrm{MD}_H) \; . \tag{11}$$

  Z znanima $S_{AB} \| M$ in M je trivialno odkriti še S_{AB}.

- Haše funkcije z lastnostmi 1 – 5 so šibke haše funkcije
- Haše funkcije z lastnostmi 1 – 6 so močne haše funkcije
- Šesta lastnost ščiti pred sofisticiranimi napadi **birthday attack**.
- Izvleček ščiti tudi integriteto podatkov, podobno kot preveritvene sekvence (CRC).

Enostavne haše funkcije

- Vse haše funkcije delujejo po istem splošnem principu:
  - vhod se obravnava kot zaporedje n-bitnih blokov
  - v iterativnem postopku se obdela blok za blokom
- primer preproste haše funkcije:

$$C_i = b_{i1} \oplus b_{i2} \oplus \cdots \oplus b_{im} \tag{12}$$

kjer so C_i: i-ti bit v haše kodu, $1 \leq i \leq n$

m: število vhodnih n-bitnih blokov

b_{ij}: i-ti bit v j bloku

$\oplus$: operacijo XOR

blok 1	b_{11}	b_{21}		b_{n1}
blok 2	b_{12}	b_{22}		b_{n2}
	⋮	⋮		⋮
blok m	b_{1m}	b_{2m}		b_{nm}
haše	C_1	C_2		C_n

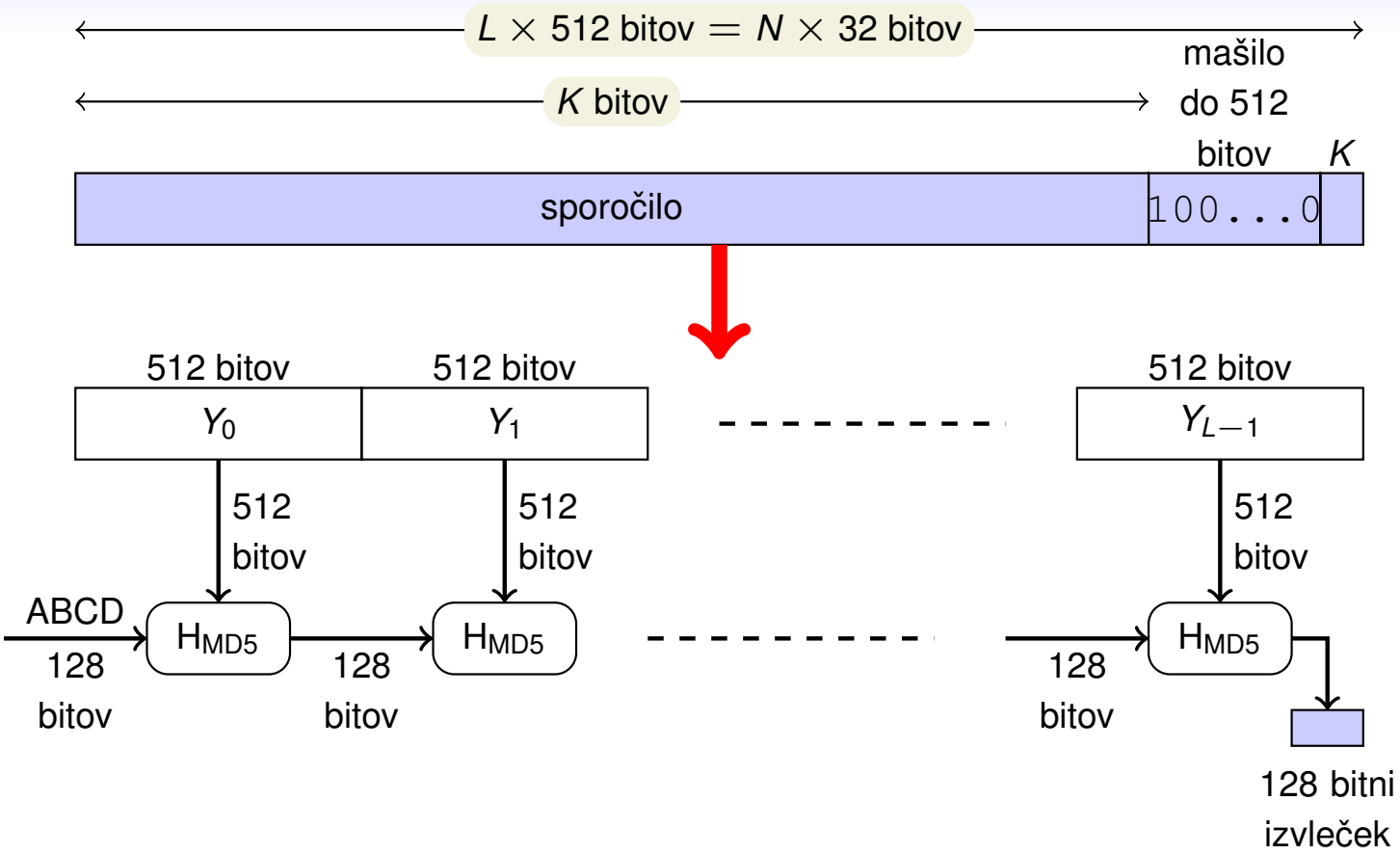
Algoritem MD5

MD5: Message Digest alogorithm, no. 5 (RFC 1321)

- Algoritem generira 128-bitni izvleček (“digest”) iz poljubno dolgih sporočil.
- Pri izdelavi izvlečka sporočila procesira v 512-bitnih blokih.
- Delovanje algoritma se izvaja v petih korakih:

1. dodajanje bitnega mašila sporočilo se z dodajanjem mašila (bitnega vzorca `1000...` (slika 11) vedno razširi (tudi ko je take dolžine) na dolžino 488 mod 512 bitov (dolžina mašila je 64 bitov manjša od celoštevilčnega mnogokratnika 512)
2. dodajanje dolžine za mašilnimi biti še doda 64-bitni zapis dolžine sporočila (slika 11). S tem se oteži tako imenovani “padding attack”

Algoritem MD5 (2)



Slika: Generiranje izvlečka sporočila s postopkom MD5.

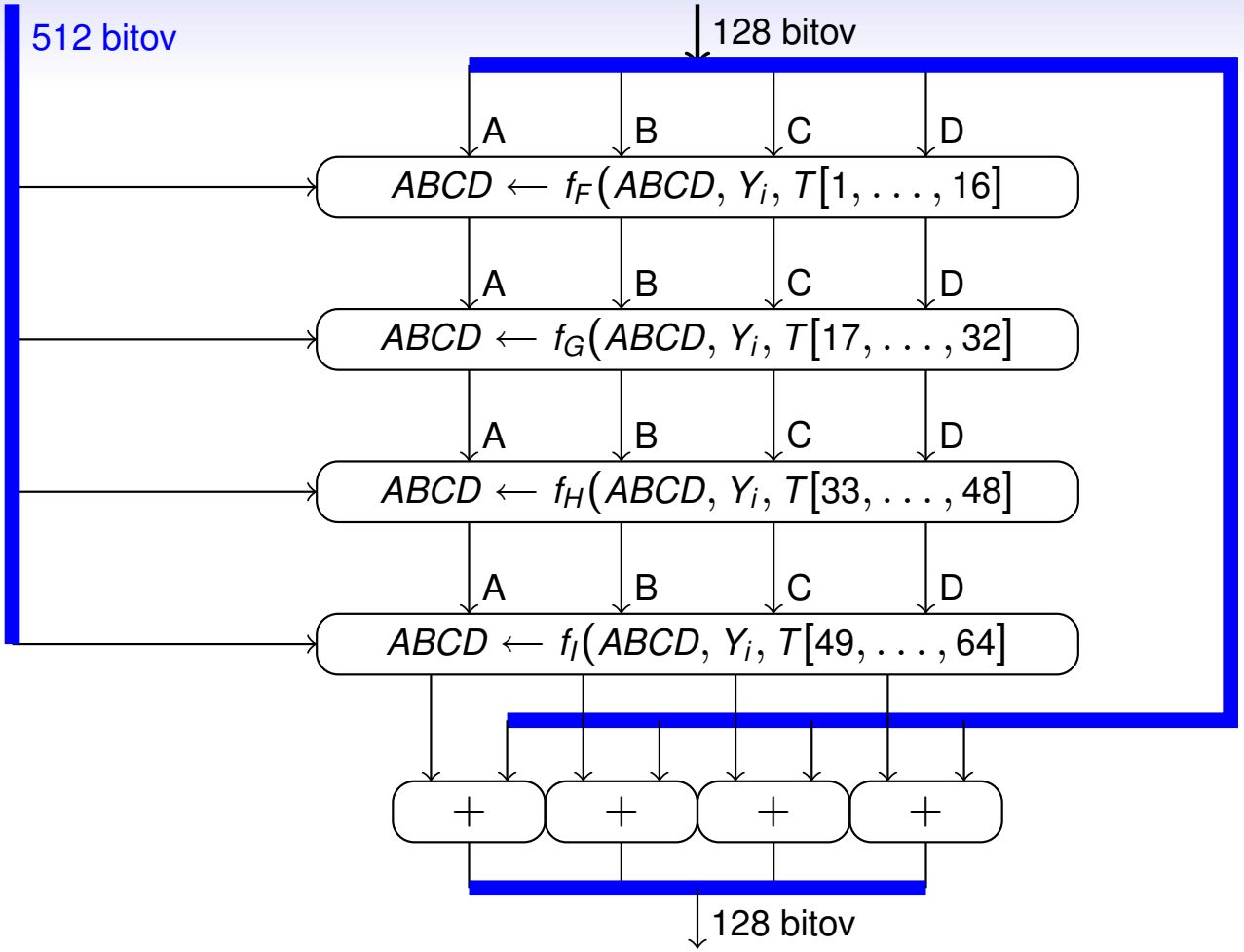
Algoritem MD5 (3)

3. inicializacija vmesnega pomnilnika za MD V izvajanju algoritma se uporabljajo 128-bitni vmesni pomnilniki za hranjenje vmesnih rezultatov izvlečka. Ob inicializaciji se v nje vpišejo vrednosti (hexadecimalno):

   A = 01234567
   B = 89ABCDEF
   C = FEDCBA98
   D = 76543210

4. obdelava sporočila 512-bitnih blokih Srce algoritma do modulih H_{MD5}, ki vsebujejo štiri primitive podobne strukture vendar z različnimi logičnimi funkcijami: f_F, f_G, f_H in f_I (slika 12).

Algoritem MD5 (4)



Slika: Procesiranje sporočil v modulu H_{MD5}.

Šifriranje z javnimi ključi

- Šifriranje z javnimi ključi sta predlagala Diffie in Hellman leta 1976

  W. Diffie and M. Helman: “A new Directions in Cryptography”. IEEE Transaction on Information Theory, November 1976.

- Ta predlog je bil dobesedno prvi revolucionarni napredek v šifriranju po tisočih letih . . .
- Algoritmi javnih ključev temeljijo na matematičnih funkcijah in ne na zamenjavah ali permutacijah kot vsi ostali algoritmi.
- Najpomembnejša lastnost algoritmov je asimetričnost, ki uporablja dva ključa, z razliko od simetričnih algoritmov, ki uporabljajo le enega.
- Uporaba dveh ključev ima (še vedno) nedoumljive posledice pri zagotavljanju zaupnosti, distribuciji ključev in avtentikaciji.

Nekaj napačnih predstav o javnih ključev

- Algoritmi z javnimi ključi so bolj varni od ključev, ki se uporabljajo pri klasičnem šifriranju . . .
  - dejstvo je, da varnost kateregakoli sistema šifriranja je odvisna od dolžine ključa in potrebnega računskega časa za razkritje ključa
  - v principu algoritmov klasičnega šifriranja in javnih ključev ni ničesar, kar bi dajalo večjo varnost enemu od njih
- Z javni ključi ni problemov z distribucijo ključev . . .
  - za distribucijo je še vedno potrebni neke vrste protokoli, običajno so v obliki agentov,
  - agenti niso bolj enostavni ali zelo različni od postopkov znanih iz distribucije simetričnih ključev

Princip uporabe javnih ključev

1. Vsak končni sistem v omrežju (uporabnik) generira par ključev, ki jih uporablja pri šifriranju
2. Enega od ključev izbere za javni ključ in ga objavi v javnem registru ključev, drugega pa ustrezno shrani kot zasebni ključ
3. Ko uporabnik A želi poslati šifrirano sporočilu uporabniku B, sporočilo šifrira z javnim ključem B.
4. Uporabnik B dešifrira sprejeto sporočilo s svojim zasebnim ključem. To sporočilo lahko dešifrira le on, saj drugi nimajo njegovega zasebnega ključa.

RSA algoritem javnih ključev

- Prvi sistem javnih ključev so leta 1977 razvili Ron Rivest, Adi Shamir in Len Adleman iz MIT in ga objavili leta 1978

> R. Rivest, A. Shamir and L. Adleman: "A Method for Obtaining Digital Signatures and Public Key Cryptosystems". Communications of ACM, February 1978.

- Algoritem RSA je od takrat vodilni in splošno sprejeti algoritem za generiranje javnih ključev. Opišemo ga z:

$$C = M^e \mod n \tag{13}$$

$$M = C^d \mod N = \left(M^e\right)^d \mod n = M^{ed} \mod n \tag{14}$$

- Oddajnik in sprejemnik morata poznati n.
- Oddajnik še pozna e in samo sprejemnik še d

RSA algoritem javnih ključev (2)

- Ključa sestavljata para:

$$\text{javni ključ:} \quad KU = \{e, n\}$$
$$\text{zasebni ključ:} \quad KR = \{d, n\}$$

- Algoritem RSA:
  - izberemo dve praštevili p in q
  - določimo količino $\phi(n)$ (Eulerjevo sprotno vrednost n), ki pove koliko pozitivnih celih števil je manjših od n
  - izberemo število d, ki j največji skupni delitelj (gcd) n in $\phi(n)$
  - izračunamo e, ki je enaka inverzni vrednosti d po modulu $\phi(n)$

RSA algoritem javnih ključev (3)

Generiranje ključev

izberi p, q	p in q sta praštevili
izračunaj $n = pq$	
izberi celo število d	$\gcd(\phi(n), d) = 1, 1 < d < \phi(n)$
izračunaj e	$e = d^{-1} \mod \phi(n)$
javni ključ:	$\mathrm{KU} = \{e, n\}$
zasebni ključ:	$\mathrm{KR} = \{d, n\}$

Šifriranje

sporočilo:	$M < n$
šifrirano sporočilo:	$C = M^e \mod n$

Dešifriranje

šifrirano sporočilo:	C
sporočilo:	$M = C^d \mod n$

RSA algoritem javnih ključev (4)

- Primer izračuna ključev:
  1. izberemo praštevili, na primer $p = 7$ in $q = 17$
  2. izračunamo $n = pq = 7 \times 17 = 119$
  3. izračunamo $\phi(n) = (p-1)(q-1) = 96$
  4. izberemo d tako, da bo najbližji celoštevilčni mnogokratnik k $\phi(n)$ in manjše od $\phi(n)$, v tem primeru je $d = 5$
  5. določimo e tako, da je $de = 1 \mod 96$ in $e < 96$. Pravilni vrednost je $e = 77$, ker $77 \times 5 = 385 = 4 \times 96 + 1$

- Ključa sta:

javni ključ: $KU = \{5, 119\}$
zasebni ključ: $KR = \{77, 119\}$

Več informacij najdemo v . . .

- William Stallings: Data and Computer Comunications.
  Fifth editions, Prentice Hall, 1997, ISBN 0-13-571274-2
- Cisco Systems, Inc: Internetworking Technologies Handbook.
  Fourth Edition, Cisco Systems™ 2004, ISBN 1-58705-119-2