

Računalniške komunikacije

2023/24



<https://alison.com/course/advanced-diploma-in-cryptography>

kriptografija
osnovne metode
simetrična kriptografija
kriptografija z javnimi ključi
principi varovanja
bločna, asimetrična, principi varovanja

operativna varnost
požarni zidovi
sistemi za preprečevanje vdorov
napadi in grožnje

Omrežna varnost



- varnosti izzivi pri elektronski komunikaciji:
 - **zaupnost:** samo pošiljatelj in prejemnik naj bi lahko brala sporočilo
 - preprečevanje **prisluškovanja:** prestrezanje sporočil (pasivni napad)
 - **integriteta:** pošiljatelj in prejemnik želita zagotoviti, da sporočilo med prenosom ni bilo spremenjeno
 - prečevanje dodajanja, brisanja in **spreminjanje** sporočil (aktivni napad)
 - **identifikacija in avtentikacija:** pošiljatelj in prejemnik želita preveriti in potrditi medsebojni identiteti
 - preprečevanje **kraje identitete** (impersonation): napadalec lahko ponaredi izvorni naslov v paketu
 - preprečevanje **ugrabitve seje** (hijacking): napadalec odstrani pošiljatelja in prevzame njegovo vlogo
 - omogočanje **avtorizacije:** katere aktivnosti lahko uporabnik izvaja v sistemu
 - preprečevanje **zanikanja komunikacije**
- primeri aplikacij: elektronske transakcije preko spletja (npr. nakupi), elektronsko bančništvo, DNS strežniki, usmerjevalniki, ...

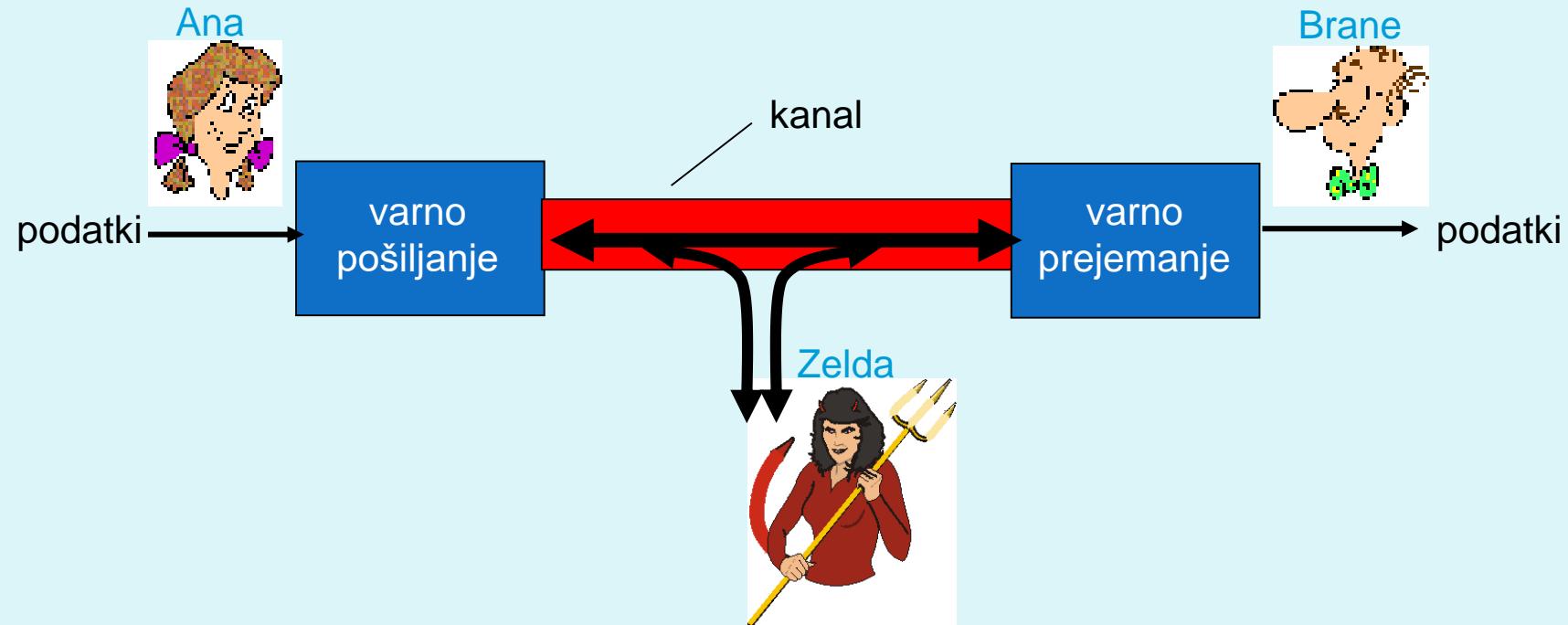
Omrežna varnost



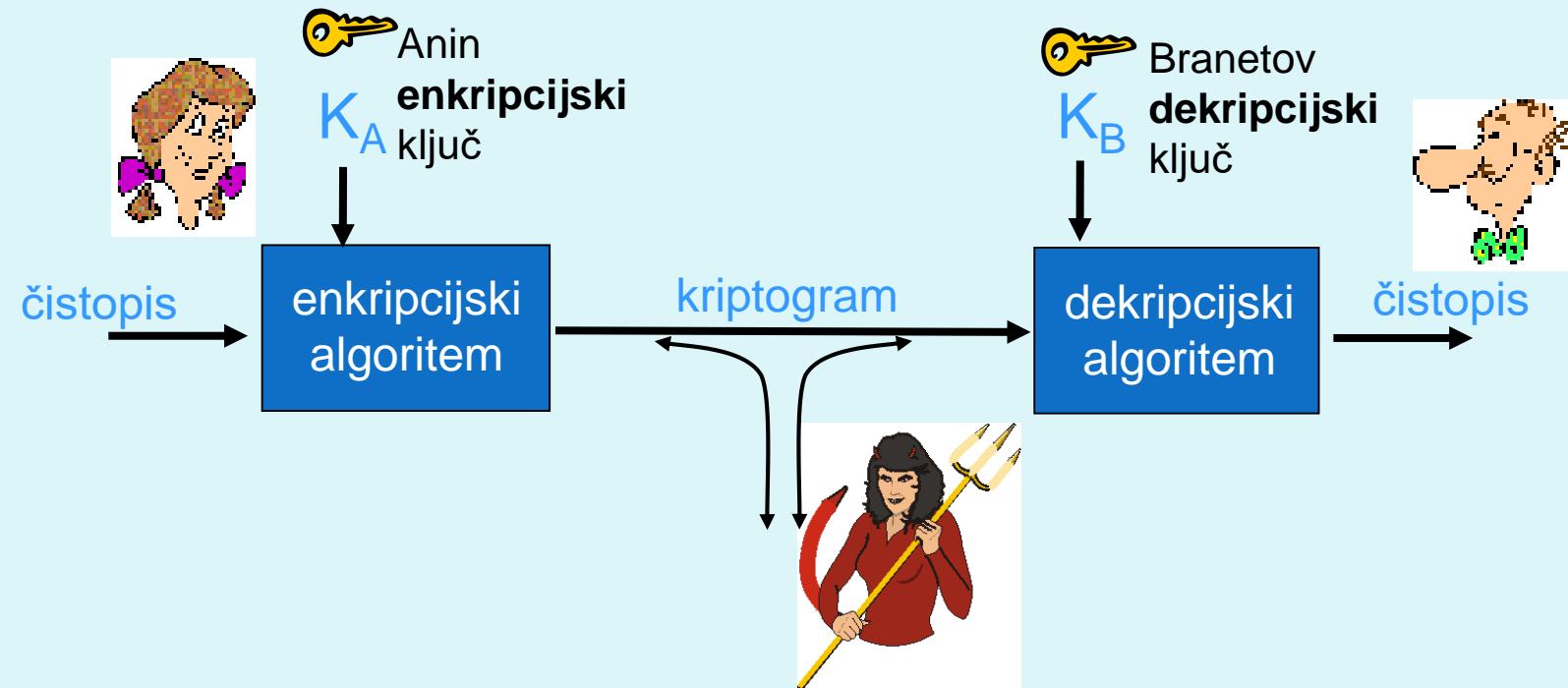
- **varnost v praksi (operativna varnost):**
 - naprave: **požarni zidovi**, sistemi za **zaznavanje/preprečevanje vdorov**,
 - dodaten cilj: preprečevanje **onemogočanja storitev** (denial of service): onemogoči uporabo storitev (npr. s preobremenitvijo virov)
- varnost je **potrebna na vseh plasteh**: na aplikacijski, transportni, omrežni in povezavni plasti
 - **fizična plast**: kriptiranje povezave
 - **omrežna plast**: filtriranje paketov, opazovanje prometa
 - **transportna plast**: kriptiranje povezav med dvema procesoma
 - **aplikacijska plast**: avtentikacija na podlagi preverjanja identitet

"Prijatelji" in "sovražniki"

- pošiljatelj in prejemnik (Ana in Brane) želita **varno** komunicirati
- za zagotovitev varne komunikacije uporabita postopke, s katerimi zavarujeta sporočilo na kanalu pred sovražniki (Zelda, Sauron, Tracy, ... ?)



Terminologija



m

$K_A(m)$

$m = K_B(K_A(m))$

čistopis

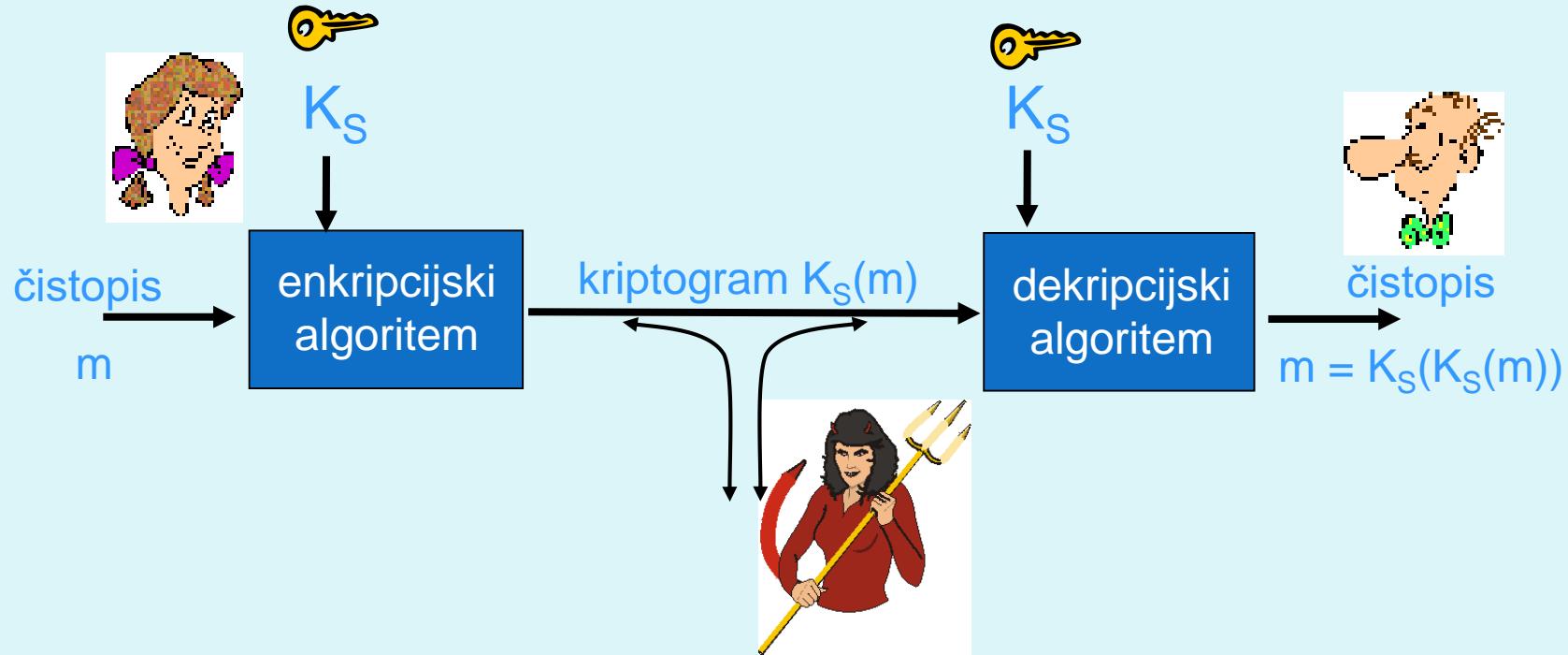
kriptogram, kriptiran s ključem K_A

odkriptiran čistopis

Kriptografija

- običajno se uporablja
 - algoritmom, ki je **znan** vsem
 - **tajni** so samo ključi
- **vrste kriptografije glede na ključe:**
 - **simetrični ključi:** enkripcijski in dekripcijski ključ sta enaka
 - **javni ključi:** uporaba dveh ključev, enega javnega in drugega tajnega
- zgoščevalne funkcije (*hash functions*)
 - so tudi enkripcijski algoritmom, vendar ne uporablja ključev
 - kdaj je to lahko koristno?

Kriptografija s simetričnim ključem



- pošiljatelj in prejemnik uporablja isti (simetričen) ključ
- primer ključa: dogovor o zamenjavi znakov v sporočilu (substitucijski vzorec)
- *kako si pošiljatelj in prejemnik varno izmenjata ključ?*

Metode kriptiranja

- glede na način kriptiranja (algoritem)
 - **substitucija** (zamenjava znakov z drugimi)
 - Cezarjev Kriptogram
 - Vigenèr-jev kriptogram
 - Porterjev kriptogram
 - **transpozicija** (zamenjava vrstnega reda znakov)
 - **sodobne (kombinirane) metode**
- glede na velikost sporočila (podatkov)
 - znakovna
 - bločna (kriptiramo zaporedja znakov)
- glede na uporabljene ključe
 - **simetrična** kriptografija (oba udeleženca uporabljata enak ključ)
 - znakovna (bit po bit)
 - bločna (sporočilo razbijemo na bloke, vsakega kriptiramo neodvisno)
 - **kriptografija z javnimi ključi** (ključa za enkripcijo in dekripcijo sta različna)



Klasične metode: Cezar

- Cezarjev kriptogram: substitucija z zamikom za k črk
 - ključ je k (velikost zamika)
 - kriptogram JULUA = čistopis ??? ($k = 21$)

a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	z	ž
u	v	z	ž	a	b	c	č	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t

- imamo 25 možnih ključev
 - kriptogram razbijemo v največ 25 poskusih



Kriptoanaliza substitucijskih kriptogramov

Kriptoanaliza (razbijanje kriptogramov) na osnovi:

- **poznanega besedila** (npr. "please login" ali "HTTP/1.1")
 - zato kriptiramo le vsebino, ne cele komunikacije
- **statistike jezika** (črke, besede, dvo- ali tročrkovni sklopi – potrebno je daljše besedilo).
- **poznavanja vsebine** (semantika) olajša razbijanje – iščemo pričakovane korene besed ipd.
- Glej:
 - primer: <https://www.crypto.org/en/cto/>



Vigenèr-jev kriptogram – večabecedno kriptiranje

- Viegnerjeva matrika: vse Cesarjeve abecede.

izbira abecede
glede na ključ

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z				
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z					
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z						
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z							
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z								
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z									
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z										
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z											
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z												
N	O	P	Q	R	S	T	U	V	W	X	Y	Z													
O	P	Q	R	S	T	U	V	W	X	Y	Z														
P	Q	R	S	T	U	V	W	X	Y	Z															
Q	R	S	T	U	V	W	X	Y	Z																
R	S	T	U	V	W	X	Y	Z																	
S	T	U	V	W	X	Y	Z																		
T	U	V	W	X	Y	Z																			
U	V	W	X	Y	Z																				
V	W	X	Y	Z																					
W	X	Y	Z																						
X	Y	Z																							
Y	Z																								
Z																									

črke sporočila

- ključ = niz D črk, vsaki pripada ena vrstica (enaka 1. črka).
- z abecedo n-te črke gesla kriptiramo n -to, $n+D$ -to, $n+2D$ -to ... črko sporočila.
- prost ključ
- statistika jezika in semantika postaneta nemočni



Vigenèr-jev kriptogram (primer)

- Geslo: računalniške komunikacije
- Sporočilo: Junija vsi izpiti na žalost odpadejo, razen pri ekonomiki, septembra pa bo spet vse po starem

r	a	č	u	n	a	l	n	i	š	k	e	k	o	m	u	n	i	k	a	c	i	j	e	
j	u	n	i	j	a	v	s	i	i	z	p	i	t	i	n	a	ž	a	l	o	s	t	o	
d	p	a	d	e	j	o	r	a	z	e	n	p	r	i	e	k	o	n	o	m	i	k	i	
S	e	p	t	e	m	b	r	a	p	a	b	o	s	p	e	t	v	s	e	p	o	s	t	
a	r	e	m																					

- dolžina ključa: D=24
- Prvi stolpec črk (torej 1., 25. (1+24), 49. (1+48) črko) kriptiramo z 18. abecedo (r), itd.

Porterjev kriptogram

- Kriptiramo po 2 znaka hkrati.
- Simboli so v tabeli – vrstica za en, stolpec za drugi znak.

	a	b	c	č	d	e	f	g	h	i	j	k	l	m
a	⌚	✂️	☎️	⏳	⌚	💣	😊	✡️	🕷️	🏡	😢	🕸️		
b	👓	🔔	☺️	👀	🖱️	🌲	🌐	🎗️	▣					
c	✉️	🕯️	❄️	👋	🍽️	*								
č	▶	⟳	🦋											... itd ...
d														

- npr. KAČA = ⌚️⏳

Kodiranje

- cel znak ali besedo nadomestimo z drugo
- ni splošnega pravila za zamenjave
- ključ predstavlja cela kodna tabela



"Bugger! I was just about to crack his code, when
he burnt his blanket."

Metode kriptiranja

- glede na način kriptiranja (algoritem)
 - **substitucija** (zamenjava znakov z drugimi)
 - Cezarjev Kriptogram
 - Vigenèr-jev kriptogram
 - Porterjev kriptogram
 - **transpozicija** (zamenjava vrstnega reda znakov)
 - **sodobne (kombinirane) metode**
- glede na velikost sporočila (podatkov)
 - znakovna
 - bločna (kriptiramo zaporedja znakov)
- glede na uporabljene ključe
 - **simetrična** kriptografija (oba udeleženca uporabljata enak ključ)
 - znakovna (bit po bit)
 - bločna (sporočilo razbijemo na bloke, vsakega kriptiramo neodvisno)
 - **kriptografija z javnimi ključi** (ključa za enkripcijo in dekripcijo sta različna)



Transpozicijski kriptogram

- spremenimo zaporedje znakov ali delov besedila
- uporabimo ključ, oštevilčimo črke po abecedi
- zapišemo stolpce glede na oštevilčenje črk

k	o	p	r	i	v	a
3	4	5	6	2	7	1
J	u	n	i	j	a	n
a	ž	a	l	o	s	t
v	s	i	i	z	p	i
t	i	o	d	p	a	d
e	j	o	b	l	a	b

Sodobna simetrična kriptografija

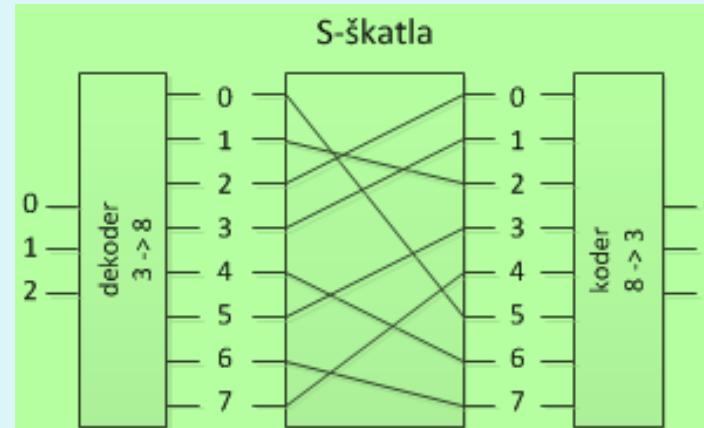
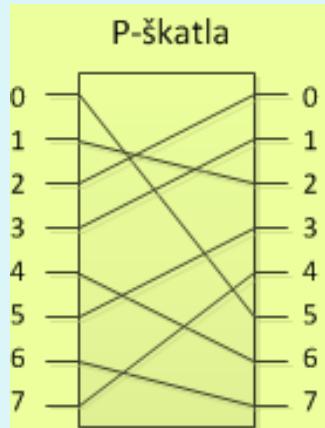
- **bločna kriptografija:** sporočilo m kriptiramo tako, da obdelamo posamezne bloke k bitov (npr. bloke po 64 bitov)
- preslikava med bloki sporočila in kriptograma je bijektivna (enolična)
- primer, če $k=3$

vhod	izhod
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Kakšen je kriptogram čistopisa 010110001111?

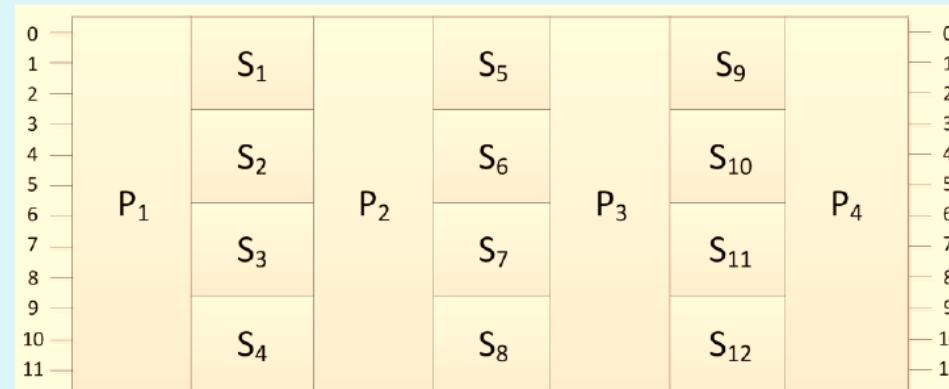
Bločna kriptografija

- permutacijska škatla (s ključem 23157046)
- substitucijska škatla (dekoder-permutacija-koder)



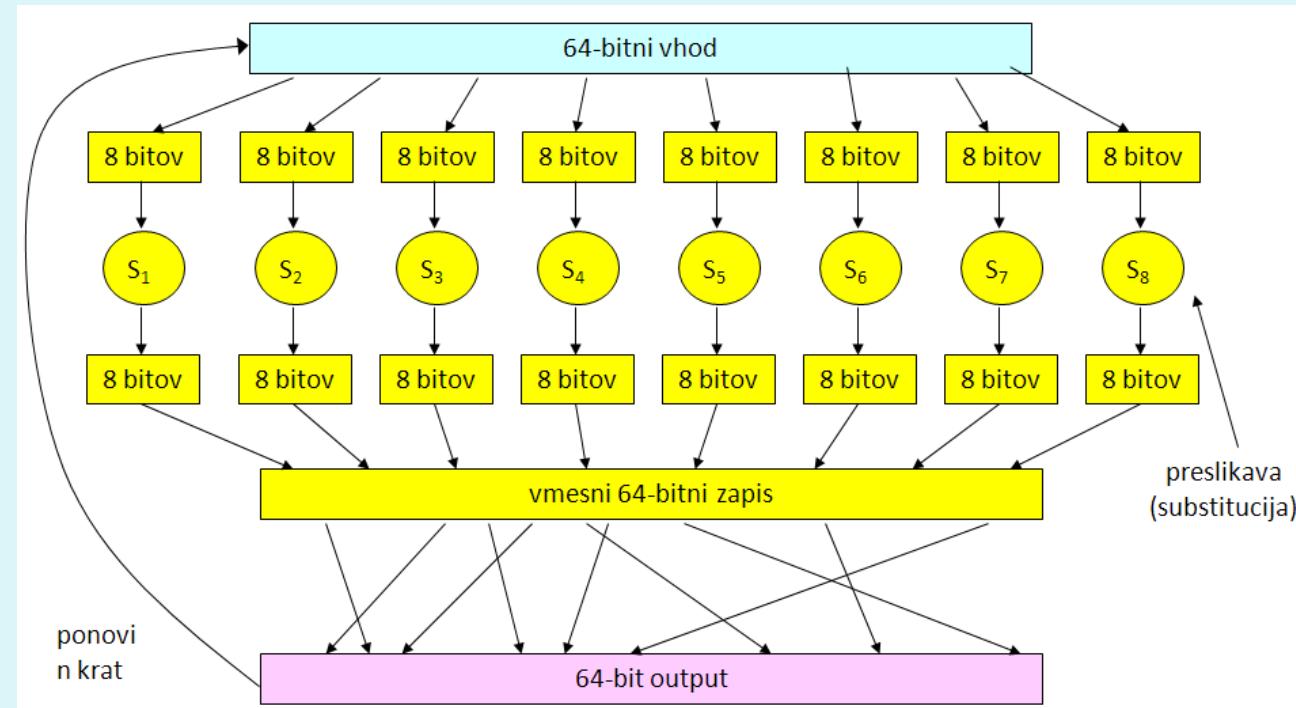
čistopis	kriptogram
000	101
001	010
010	000
011	001
100	110
101	011
110	111
111	100

- možno je kombiniranje škatev v preslikovalno kaskado za poenostavitev logike



Bločna kriptografija

- **problem:**
 - če $k=3$, imamo $40320 (=2^k!=8!)$ permutacij vseh možnih vhodov (možno razvozlati kodo na domačem PC računalniku)
 - če $k=64$, je permutacij ogromno ($=2^{64}!$) in je težko hraniti tabelo permutacij (težka tudi izmenjava ključev)
- **rešitev:** uporabimo preprosto funkcijo za kriptiranje, ki pa simulira veliko tabelo

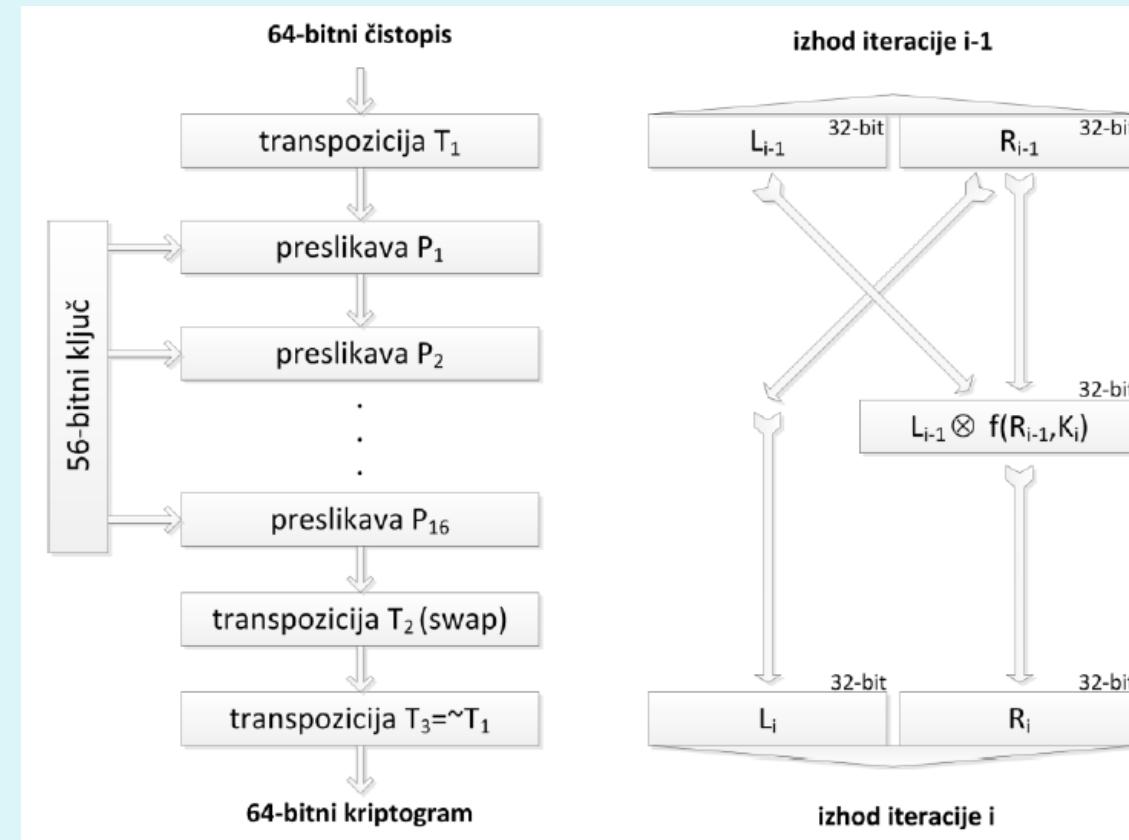


Bločna kriptografija

- **primeri algoritmov** za bločno kriptografijo
 - DES (*Data Encryption Standard*)
 - 3DES (3-kratni DES s 3 različnimi ključi)
 - AES (*Advanced Encryption Standard*)
- zgornji algoritmi uporabljajo KLJUČE
 - DES 64-bitne bloke in 56-bitni ključ;
 - AES 128-bitne bloke in 128/192/256-bitne ključe

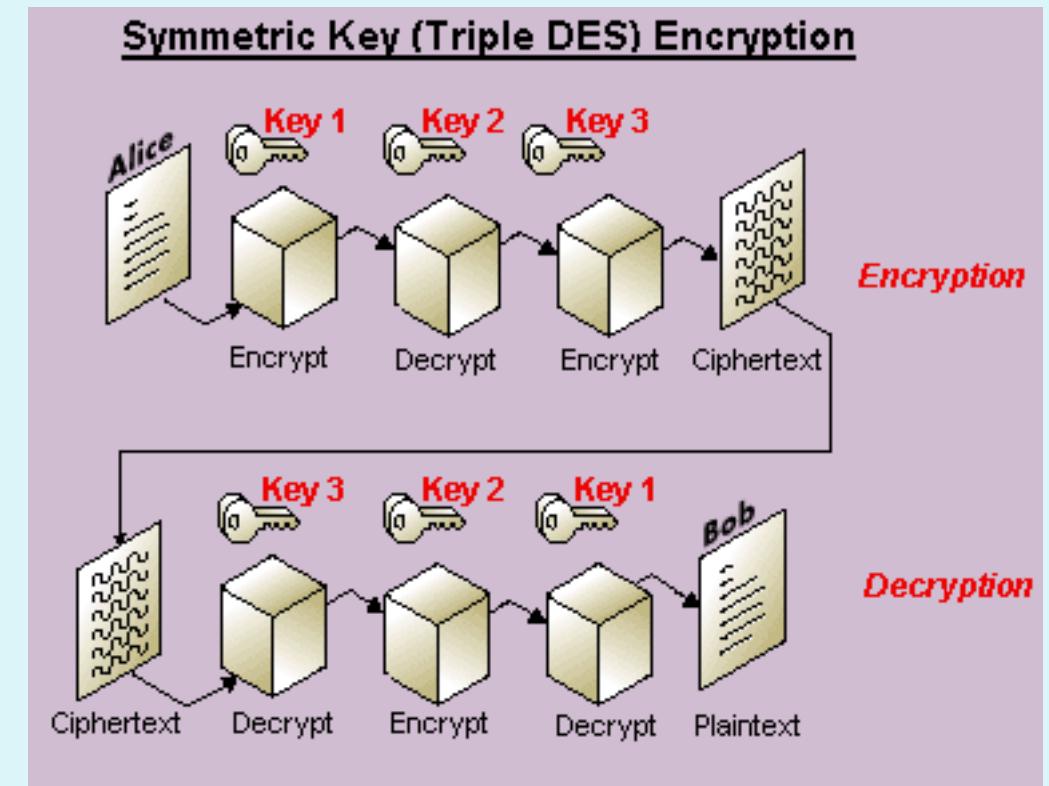
DES (Data Encryption Standard)

- DES je kombinacija transpozicijskih in substitucijskih metod
- transpozicija -> 16 preslikav (ključ!) -> SWAP -> transpozicija



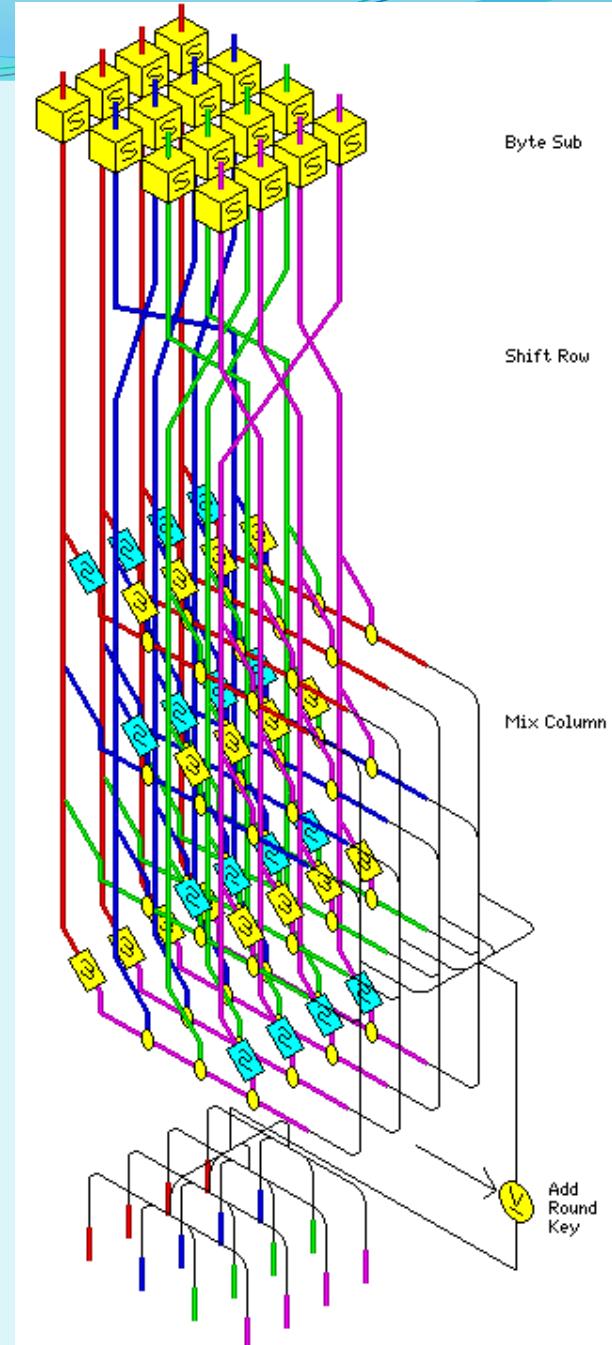
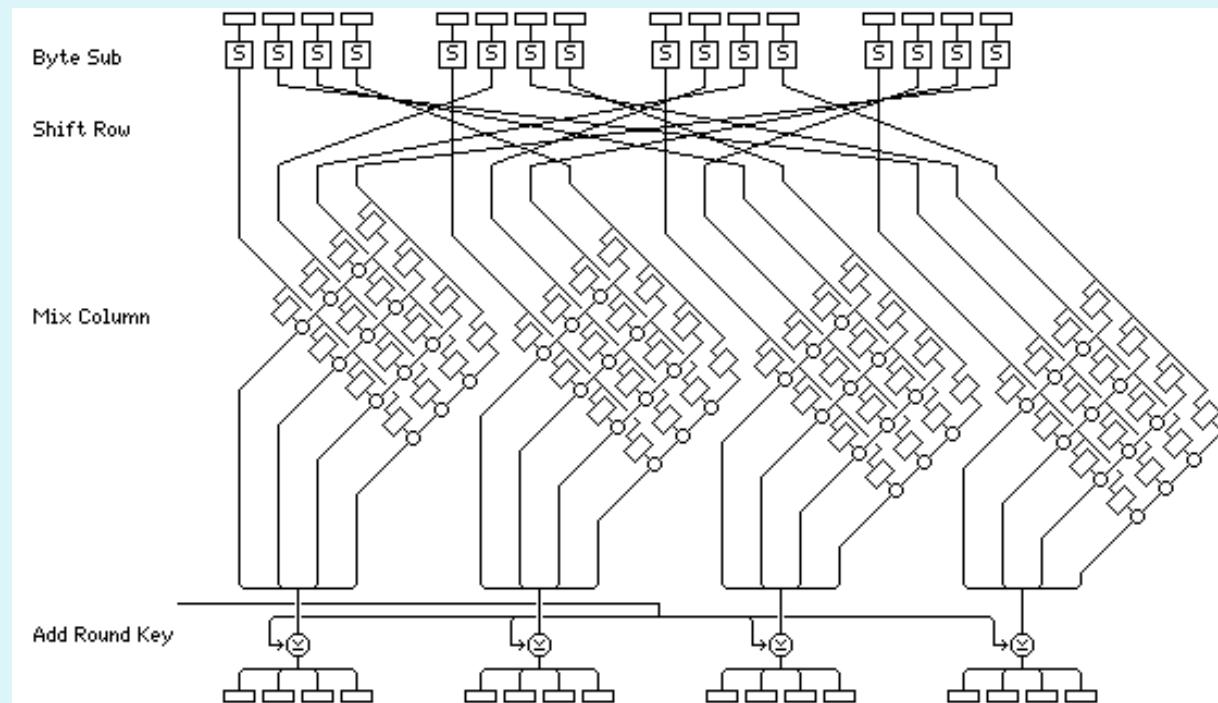
3DES (trojni DES)

- 3 x kriptiranje (kriptiranje, dekriptiranje, kriptiranje)
- 3 x počasnejši
- $2^{2 \cdot 56}$ krat varnejši za napad z grobo silo, ker uporablja 2 dodatni kriptiranji s 56-bitnim ključem
- združljivost z DES (če $K_2 == K_3$)



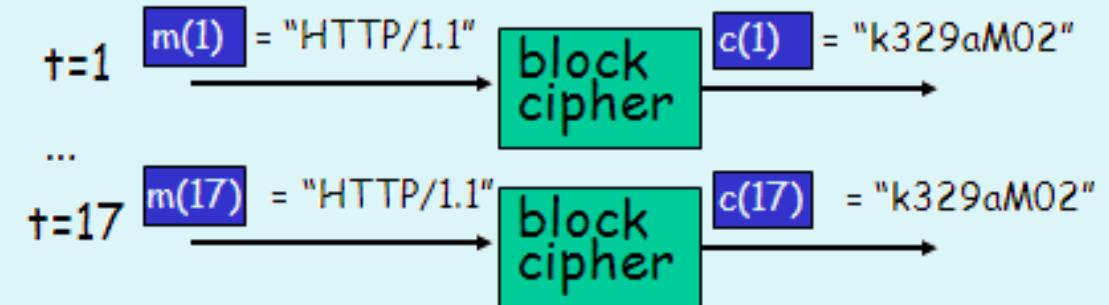
AES

- algoritem Rijndael, najbolj učinkovita in varna metoda
- bloki 128 bitov, ključ 128/192/256 bitov
- dober računalnik bi potreboval 10^{10} let za razbijanje
- različne AES operacije: zamikanje vrstic, zamenjave stolpcev, izpeljave ključev

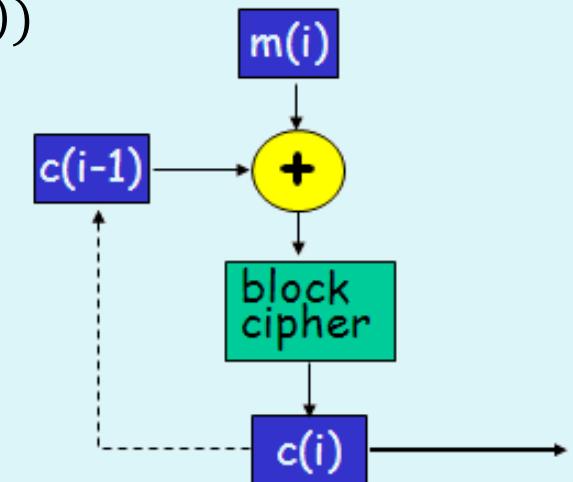


Verižno kriptiranje blokov (*Cipher Block Chaining*)

- bločna kriptografija ni praktična za pošiljanje velikih sporočil, ker dajejo ponavljajoči se bloki (npr. HTTP/1.1) iste kriptograme (možna kriptoanaliza sporočila)



- REŠITEV:** verižno kriptiranje
 - pošiljatelj s prvim sporočilom pošlje neko naključno vrednost (inicializacijski vektor) $c(0)$
 - pošiljatelj kriptira vsako sporočilo v kriptogram: $c(i) = K_S(m(i) \otimes c(i-1))$
 - prejemnik odkriptira sporočilo: $m(i) = K_S^{-1}(c(i)) \otimes c(i-1)$
- ni potrebno, da je IV tajen
- omogoča, da se isto sporočilo kriptira v različne kriprograme!



Verižno kriptiranje blokov (*Cipher Block Chaining*)

- pošiljatelj kriptira: $c(i) = K_S(m(i) \otimes c(i - 1))$
- prejemnik odkriptira: $m(i) = K_S^{-1}(c(i)) \otimes c(i - 1)$

PRIMER

- $m=010010010$, $\text{IV}=c(0)=001$, $k=3$
- pošiljatelj:
 - $c(1) = K_S(m(1) \text{ XOR } c(0)) = K_S(010 \text{ XOR } 001) = K_S(011) = \textcolor{blue}{100}$
 - $c(2) = K_S(m(2) \text{ XOR } c(1)) = K_S(010 \text{ XOR } \textcolor{blue}{100}) = K_S(110) = \textcolor{red}{000}$
 - $c(3) = K_S(m(3) \text{ XOR } c(2)) = K_S(010 \text{ XOR } \textcolor{red}{000}) = K_S(010) = 101$

\underline{m}	$K_S(\underline{m})$
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

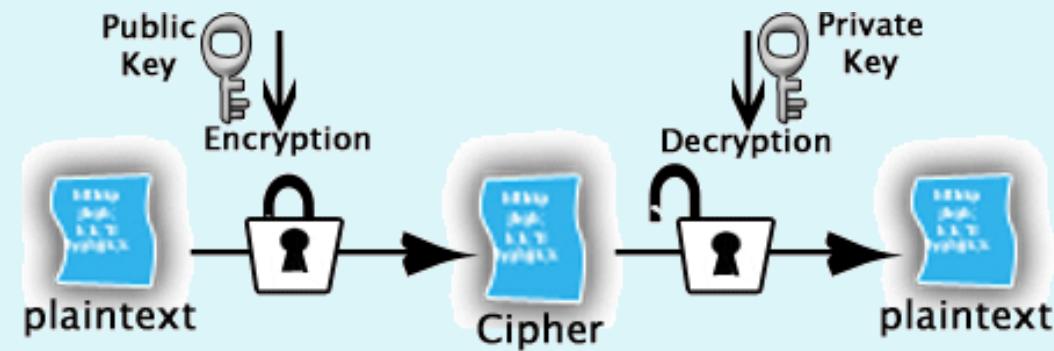
Razbijanje bločnih kriptosistemov z grobo silo

- ocena glede na stanje tehnologije v letu 2013

Ključ	Oseba	Mala skupina	Razisk. omrežje	Veliko podjetje	Vojska
40	sekunde	milisekunde	milisekunde	mikrosekunde	nanosekunde
56	dan	ure	ure	sekunde	mikrosekunde
64	tedni	tedni	dnevi	ure	milisekunde
80	tisočletja	tisočletja	stoletja	leta	minute
128	∞	∞	∞	∞	∞

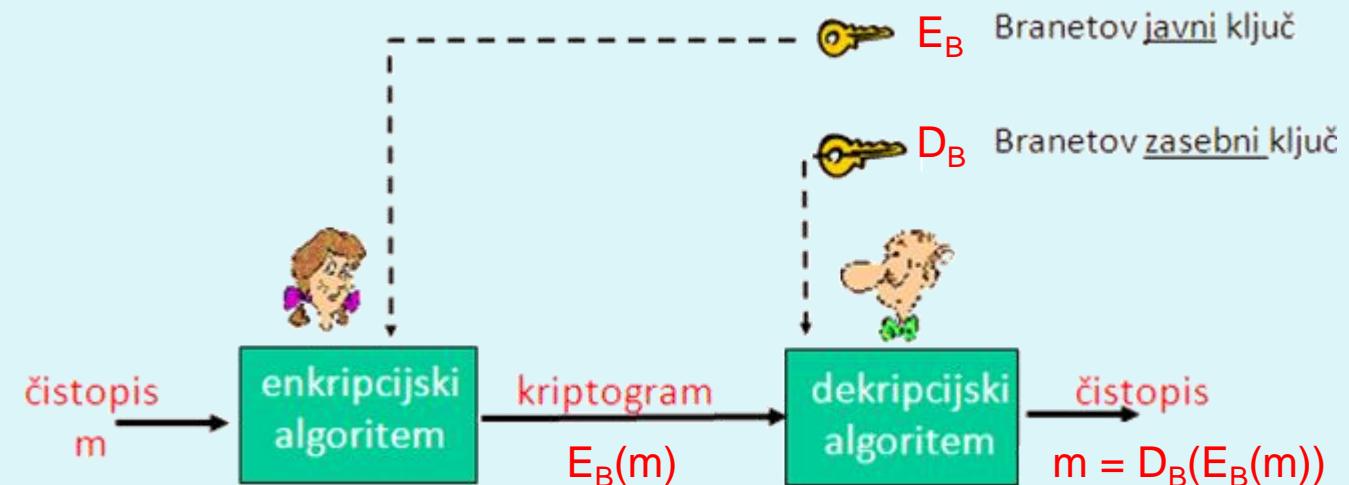


Kriptografija z javnimi ključi (= asimetrična kriptografija)



Kriptografija z javnimi ključi (asimetrična kriptografija)

- **enkripcijski (E)** in **dekripcijski (D)** ključ sta lahko različna
- E je lahko javen, D mora biti tajen. Zahteve:
 - $D(E(m)) = m$
 - iz m in $E(m)$ je nemogoče ugotoviti D
 - iz E je nemogoče ugotoviti D
- primer: algoritem RSA (Rivest, Shamir, Adleman)



Algoritem RSA

- izberemo p, q : veliki praštevili (1024 bitov)

$$n = pq$$

$$z = (p-1)(q-1)$$

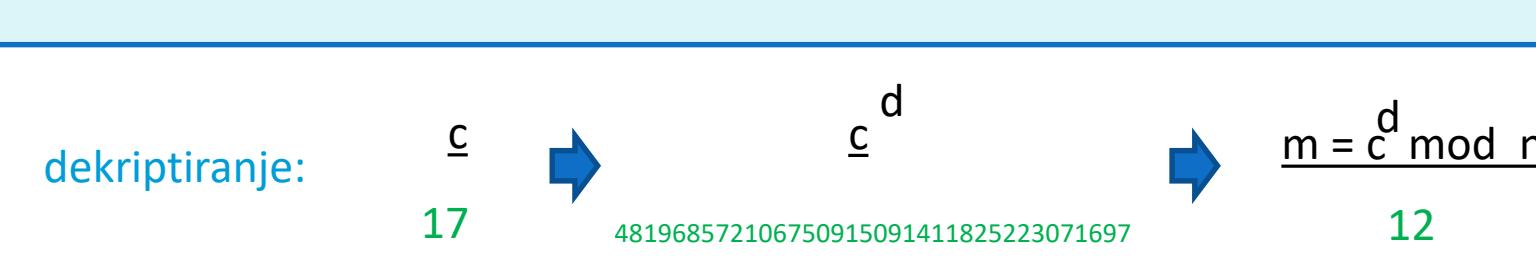
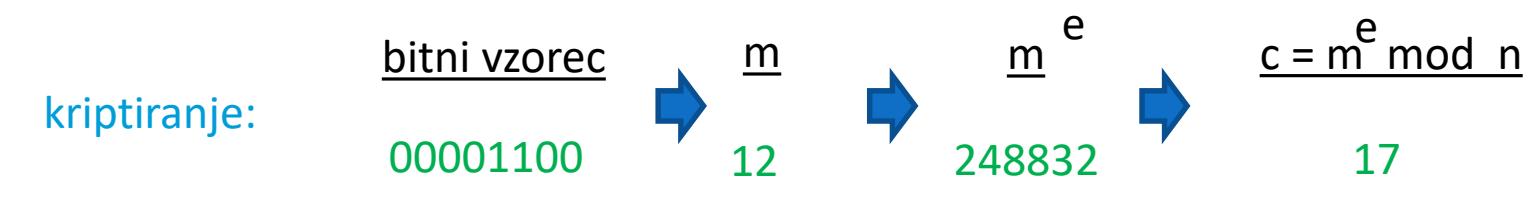
- izberemo e tako, da nima skupnih deliteljev z z.
- izberemo d tako, da $ed \bmod z = 1$
- definiramo:
 - javni ključ: $E = (n, e)$
 - zasebni ključ: $D = (n, d)$

$m \rightarrow c = m^e \bmod n$ *criptiranje*

$c \rightarrow m = c^d \bmod n$ *dekriptiranje*

RSA: primer

- Brane izbere: $p=7$, $q=5$
 - sledi $n=35$, $\varphi=24$
 - izberemo $e=5$ (e in φ sta si tuji števili) in $d=29$ (da je $e-1$ deljivo z φ)



Dejstva o RSA

- Zakaj je RSA varen?
 - Denimo, da poznamo javni ključ (n, e) . RSA je varen, ker je v praksi težko poiskati delitelja p in q števila n , saj za to ne obstaja učinkovit algoritem.
 - iskanje deliteljev 500-mestnega števila bi trajalo 10^{25} let
- ker je RSA počasen, ga običajno uporabljamo le za izmenjavo simetričnega ključa za izvedbo nadaljnje seje K_s (*session key*)

Principi varne komunikacije: avtentikacija, integriteta, elektronski podpis



Contracts Financial statements
Company strategy Human resources
documents User names and passwords
Social Security numbers Credit card details
Invoices Market research Legal advice
Sales figures Customer information

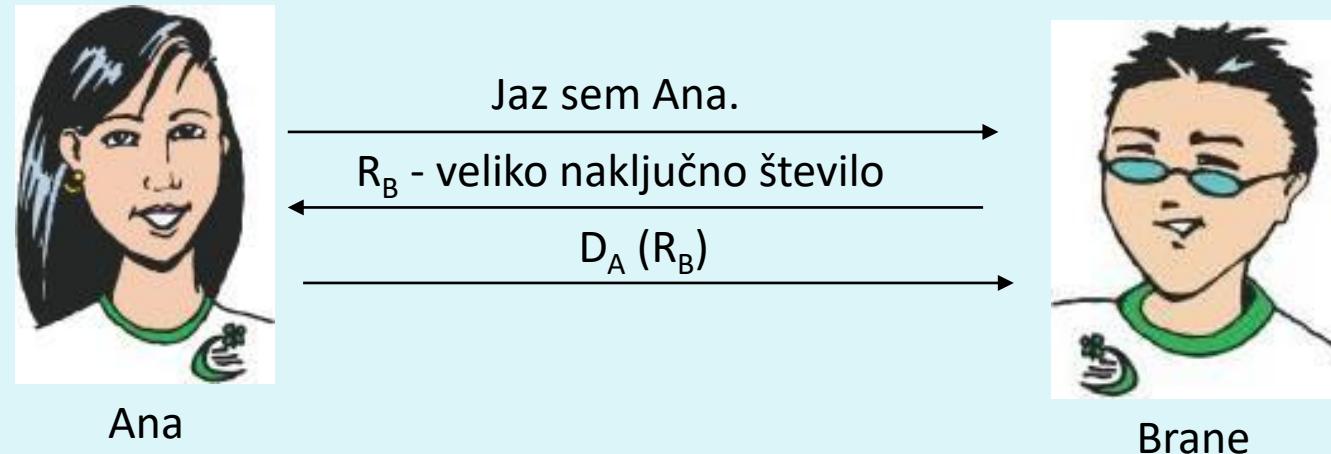
Avtentikacija udeležencev

- želimo imeti komunikacijsko okolje, v katerem lahko udeleženci preverijo, da:
 - je pošiljatelj dejansko oseba, za katero se predstavlja
 - je prejemnik zagotovo prava oseba in ne prisluškovalec
- mehanizmi za avtentikacijo z RSA:
 - imamo pošiljatelja A in prejemnika B
 - avtentikacija **pošiljatelja**:
 - sporočilo m kriptiramo v $D_A(m)$, prejemnik odkriptira v $E_A(D_A(m)) = m$
 - avtentikacija **prejemnika**:
 - sporočilo m kriptiramo v $E_B(m)$, prejemnik odkriptira v $D_B(E_B(m)) = m$
 - **vzajemna** avtentikacija:
 - sporočilo m kriptiramo v $D_A(E_B(m))$, prejemnik odkriptira v $D_B(E_A(D_A(E_B(m)))) = m$



Avtentikacija pošiljatelja in prejemnika

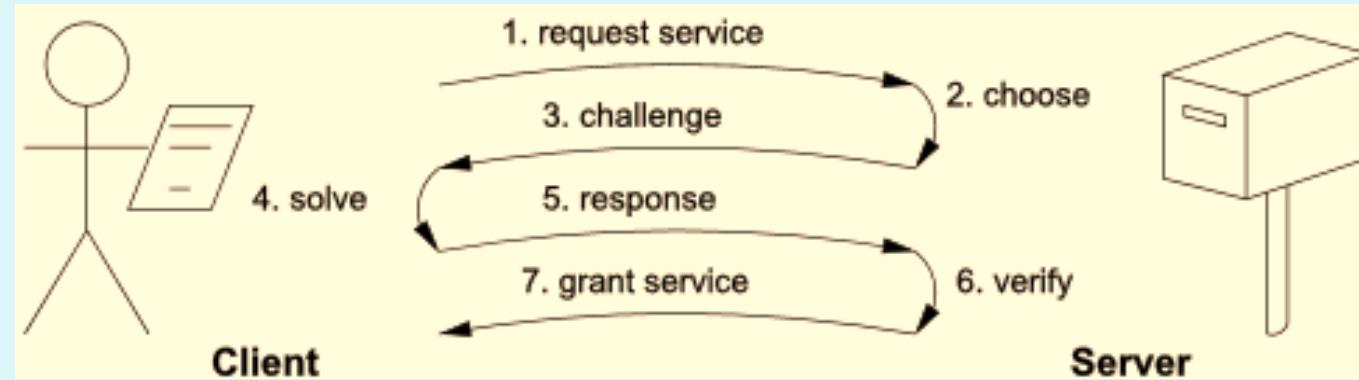
- kako preveriti, s kom zares komuniciramo?
- mehanizem izziv-odgovor (challenge-response)



- Brane dobi dokaz, da komunicira z Ano tako, ker lahko z Aninim javnim ključem odkriptira kodirano naključno število (tega je kriptirala Ana s svojim zasebnim ključem), ki ga je definiral on in ga torej pozna.
- varnostna luknja: nekdo se lahko vrine v zgornjo komunikacijo (man-in-the-middle attack) in se obnaša kot posrednik
 - REŠITEV?

Avtentikacija udeležencev

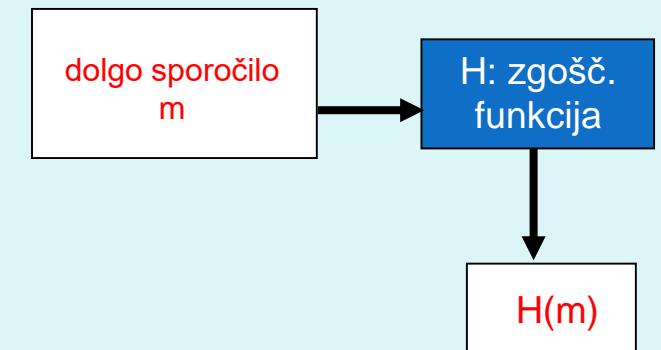
- dvostranska avtentikacija?
- učinkovitost postopka?



- drugi avtentikacijski protokoli? napadi?
 - glej učbenik, poglavje 7.6.2

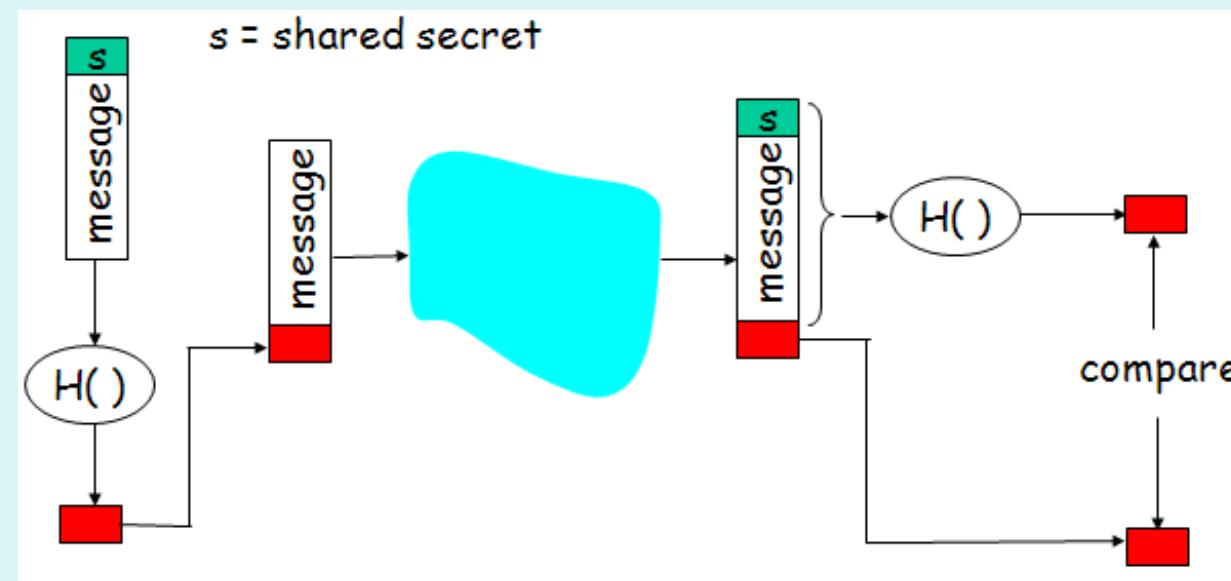
Integriteta komunikacije

- želimo imeti komunikacijsko okolje, v katerem lahko udeleženci preverijo, da:
 - sporočilo ni bilo **spremenjeno**,
 - je sporočilo "sveže" (**ni ponovljeno** posneto staro sporočilo),
- uporabljam **zgoščevalne kriptografske funkcije**:
 - preprosto izračunljive,
 - iz $H(m)$ ne moremo ugotoviti m
 - težko je najti m in m' , da velja $H(m)=H(m')$
 - izgleda kot naključen niz
 - primer: internet checksum (zakaj je slab?)
 - MD5, SHA-1
 - primer: <http://hash.online-convert.com/>



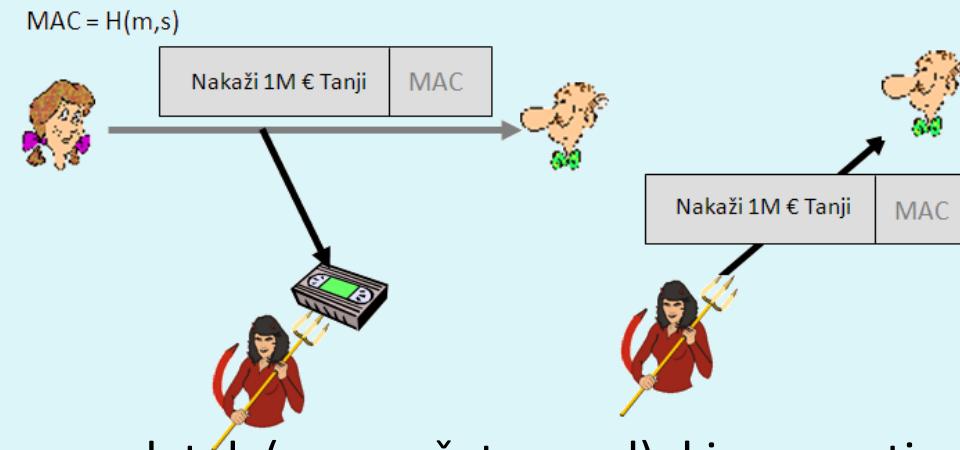
Zgoščena vrednost sporočila

- s sporočilom m pošljemo tudi $H(m)$, prejemnik preveri, ali se ujemata (dokaz o integriteti sporočila, ne pa o avtentikaciji)
- za preverjanje avtentikacije uporabimo še **avtentikacijski ključ (shared secret)**. Pošljemo m in $H(m+s)$ = MAC (*message authentication code*)
 - problem: distribucija avtentikacijskega ključa

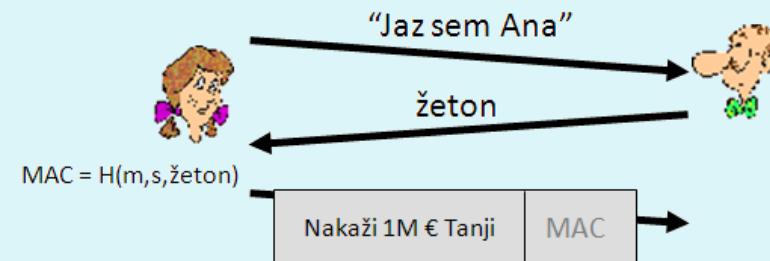


Napad s ponavljanjem komunikacije

- napadalec lahko shrani celotno (avtenticirano!) komunikacijo in jo kasneje ponovno izvede

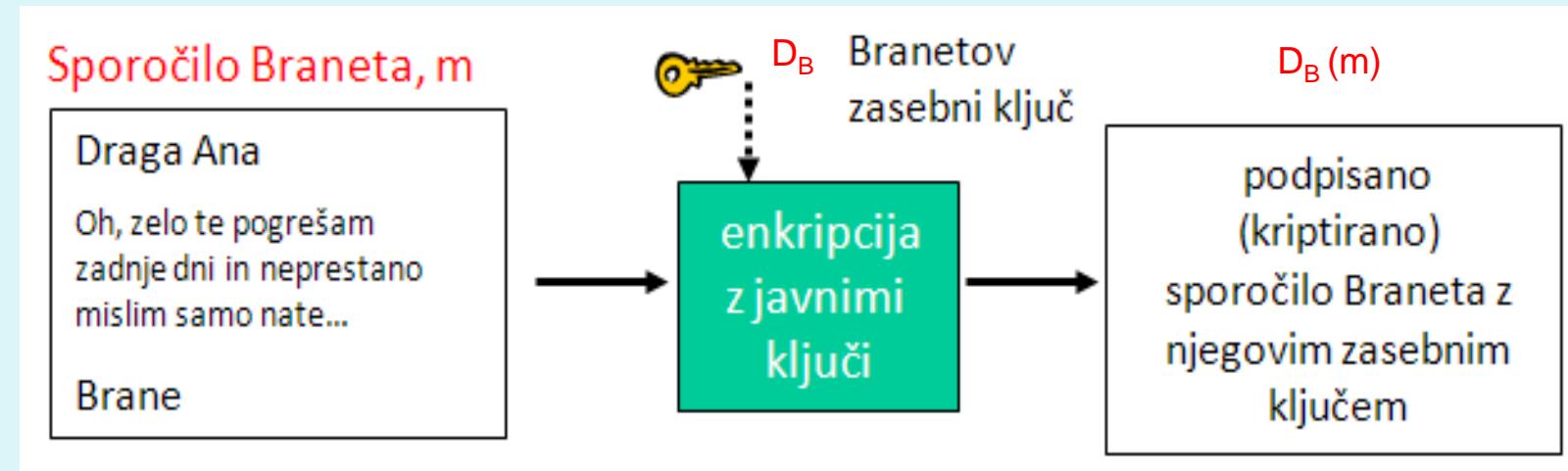


- REŠITEV: uporabimo enkraten podatek (*nonce*, žeton, sol), ki ga zgostimo skupaj s sporočilom in ključem: $H(m, s, \text{žeton})$



Digitalni podpis

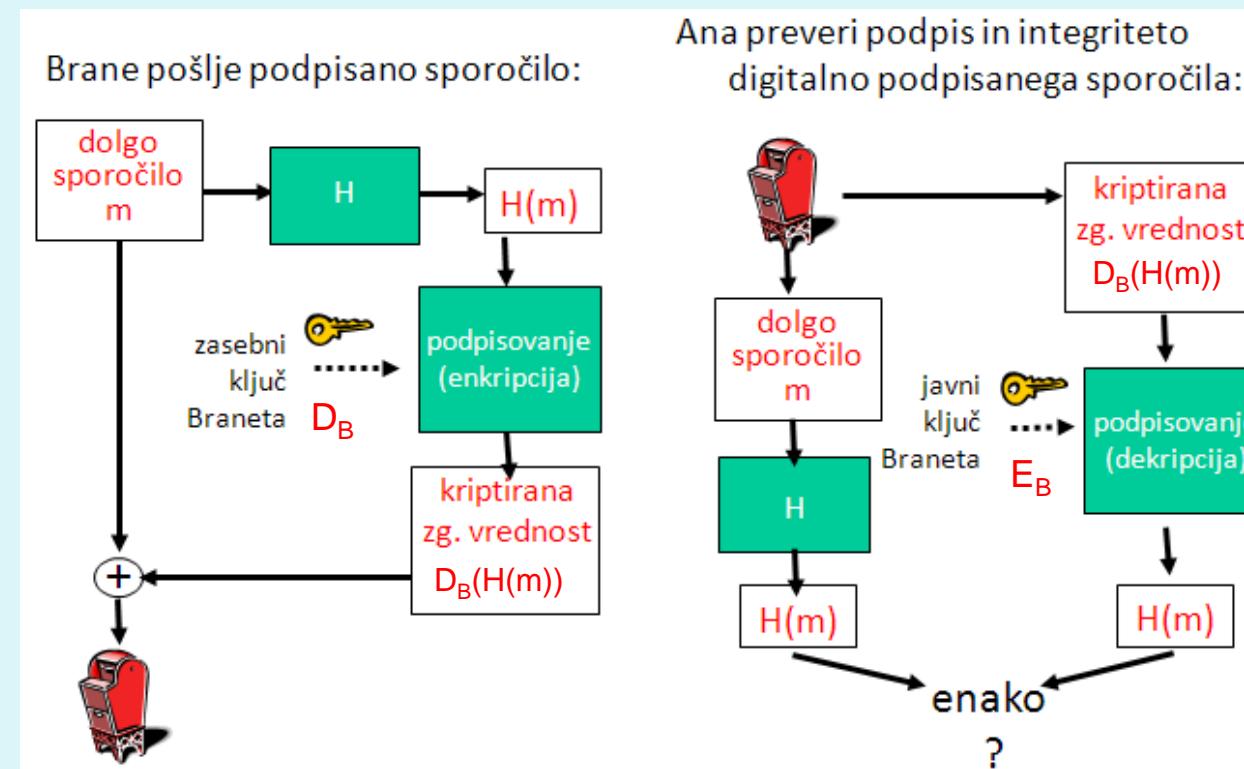
- način za zamenjavo osebnega podpisa (informacija, ki enolično potrjuje identiteto posameznika)
- avtentikacija z MAC ni enolična, uporabimo kriptografijo z javnimi ključi
- (slaba, neustrezna) IDEJA:
 - pošiljatelj B lahko izračuna $D_B(m)$, kar lahko predstavlja podpisan dokument
 - prejemnik izračuna $E_B(D_B(m)) = m$
 - časovno zahtevno pri dolgih sporočilih



Digitalni podpis

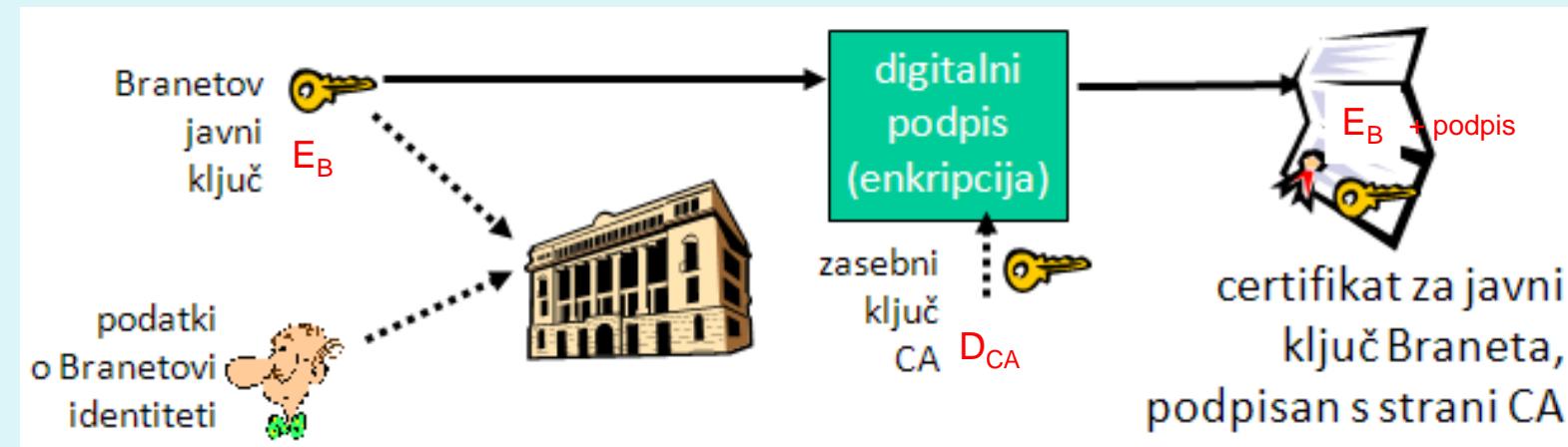
- DIGITALNI PODPIS:

- pošiljatelj B uporabi zgoščevalno funkcijo H in kriptira le zgoščeno vrednost s svojim zasebnim ključem
- digitalni podpis = $D_B(H(m))$
- skupaj z originalnim (nekriptiranim) sporočilom pošlje tudi kriptirano zgoščeno vrednost



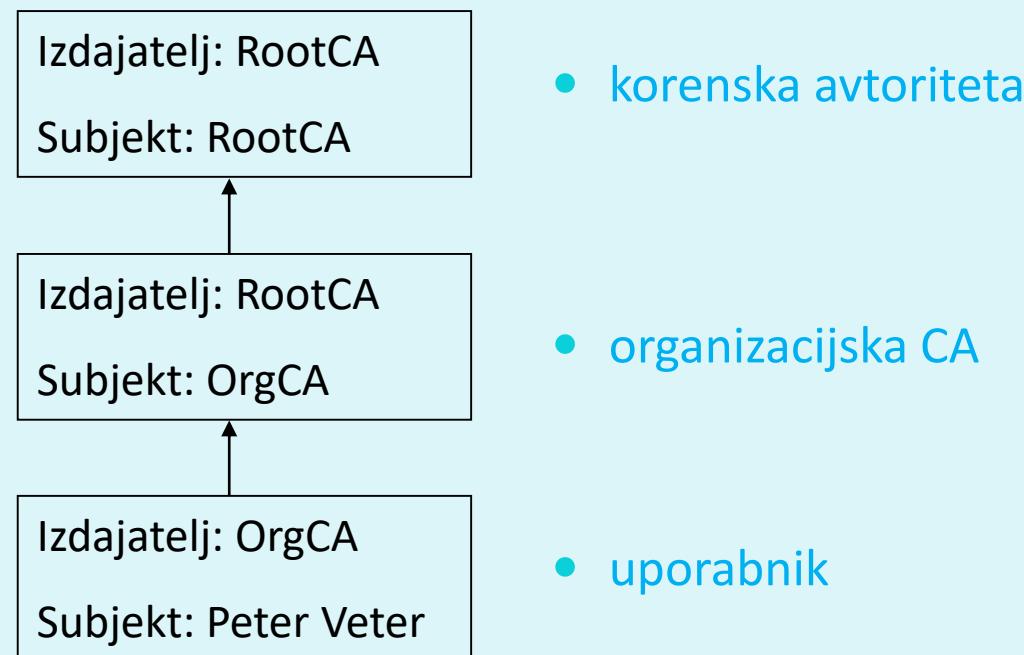
Certifikacijska agencija

- pri digitalnem podpisovanju lahko vdiralec podpiše sporočilo s svojim zasebnim ključem in se pretvarja, da je to ključ od nekoga drugega
- **REŠITEV:**
 - certifikacijske agencije (*certification authority*) preverjajo povezavo med javnim ključem in identiteto osebe.
 - certifikacijska agencija shrani povezavo med ključem in identiteto v CERTIFIKAT (tega agencija podpiše s svojim zasebnim ključem)



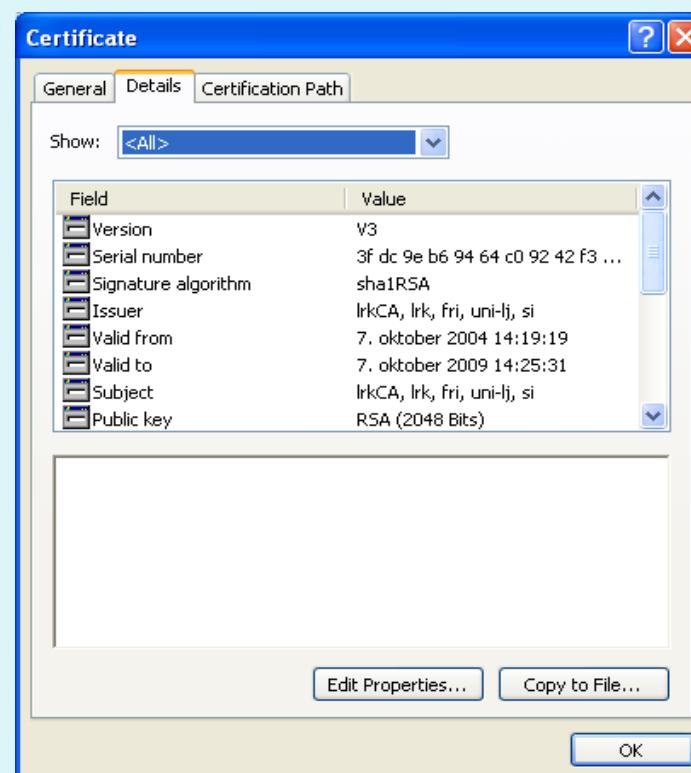
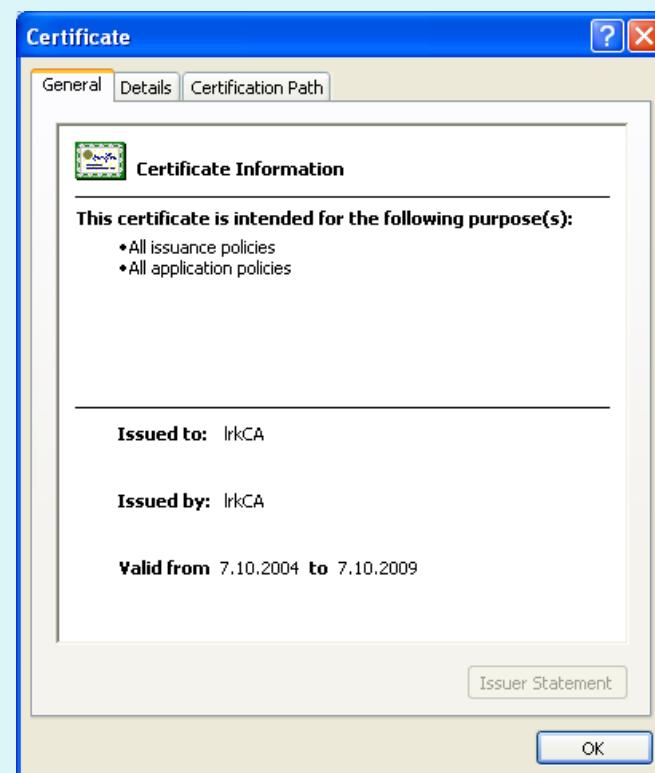
Veriga zaupanja certifikatov

- veljavnost certifikatov preverimo z verigo zaupanja
- korenske avtoritete so znane



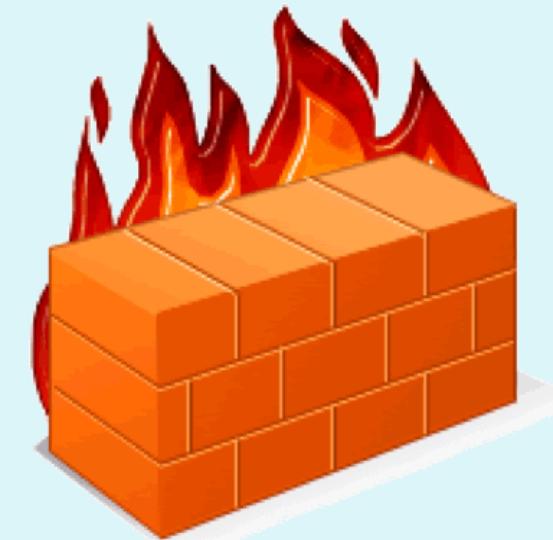
Certifikati

- certifikat vsebuje: ime izdajatelja, ime osebe, naslov, ime domene, javni ključ osebe, digitalni podpis (podpis z zasebnim ključem izdajatelja)
- uveljavljen standard za zapis certifikatov je X.509



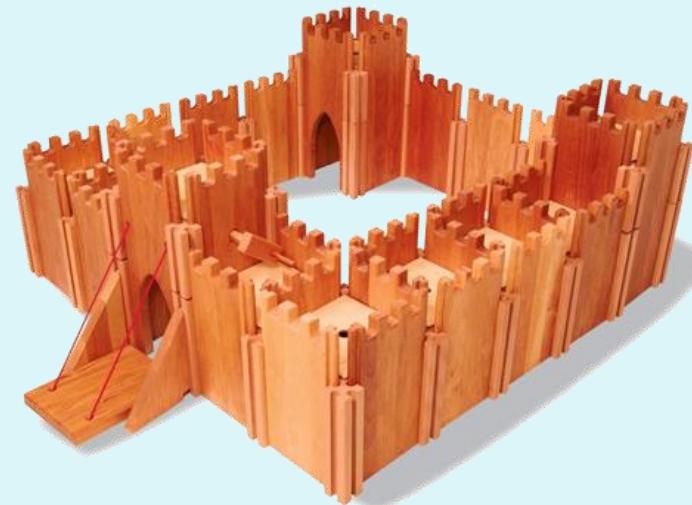
Operativna varnost:

požarni zidovi in sistemi za zaznavanje vdorov



Varnost v omrežju

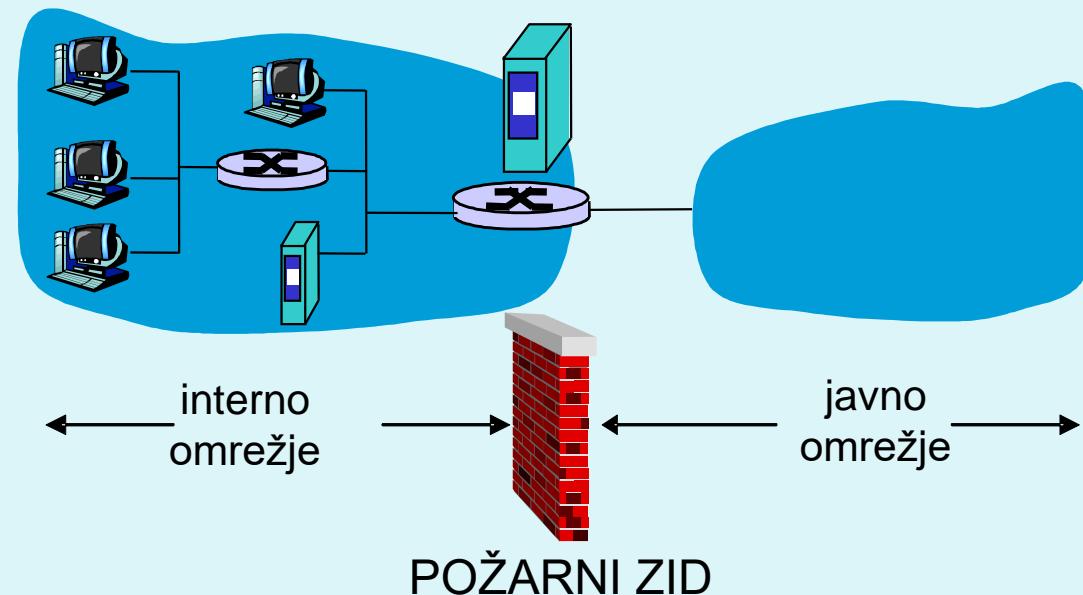
- Administrator omrežja lahko uporabnike deli na:
 - **good guys**: uporabniki, ki legitimno uporabljajo vire omrežja, pripadajo organizaciji,
 - **bad guys**: vsi ostali, njihove dostope moramo skrbno nadzorovati
- Omrežje ima običajno eno samo točko vstopa, kontroliramo dostope v njej:
 - **požarni zid (firewall)**
 - **sistem za zaznavanje vdorov (IDS, intrusion detection system)**
 - **sistem za preprečevanje vdorov (IPS, intrusion prevention system)**



Požarni zid (firewall)

izolira interno omrežje od velikega javnega omrežja, določenim paketom dovoli prehod, druge blokira. Ima 3 naloge:

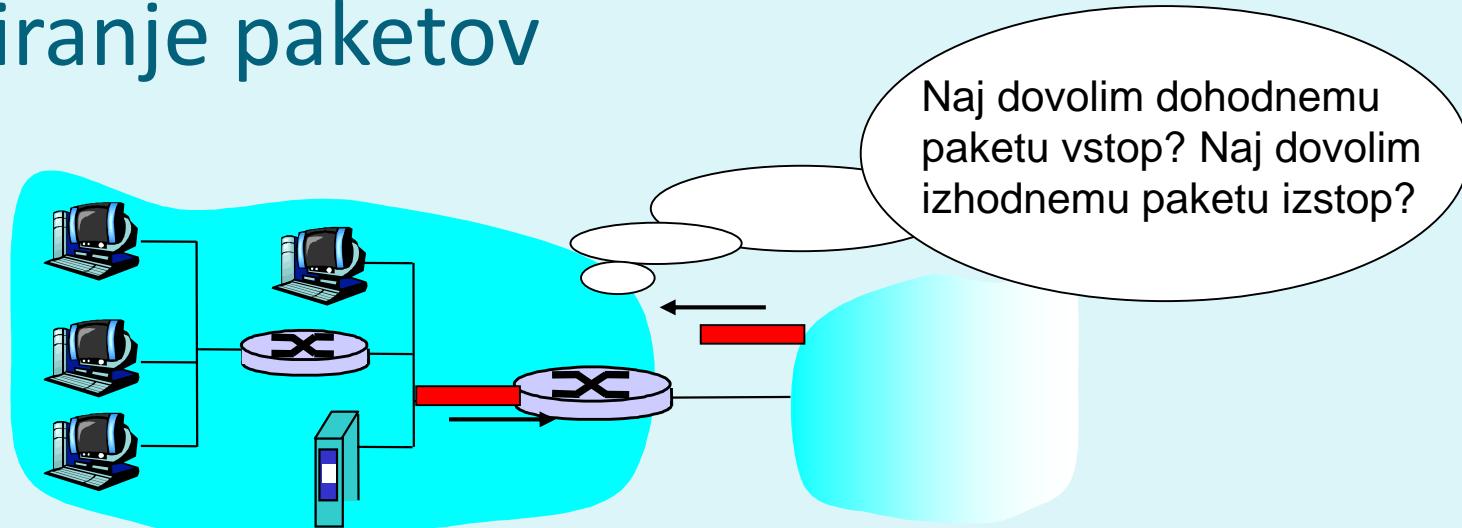
- filtrira VES promet,
- prepušča samo promet, ki je DOPUSTEN glede na politiko,
- je IMUN na napade



Požarni zid: vrste filtriranj

1. **izolirano filtriranje paketov** (angl. *stateless, traditional*)
 - pretežno filtriranje na podlagi podatkov v glavi: izvorni in ponorni naslovi ter vrata
2. **filtriranje paketov v kontekstu** (angl. *stateful filter*)
 - nadzoruje vzpostavljenost povezave
3. **aplikacijski prehodi** (angl. *application gateways*)
 - filtriranje z vpogledom v podatke aplikacijske plasti (vsebina, aplikacijski protokol, uporabniško ime, ...)

Izolirano filtriranje paketov



- filtriranje običajno izvaja že usmerjevalnik, ki meji na javno omrežje. Na podlagi vsebine paketov se odloča, ali bo posredoval **posamezen paket**,
- odločitev na podlagi:
 - IP izvornega/ponornega naslova
 - številke IP protokola: TCP, UDP, ICMP, OSPF itd.
 - TCP/UDP izvornih in ciljnih vrat
 - tip sporočila pri protokolu ICMP
 - zastavice TCP: SYN in ACK bit (sta aktivni za prvi segment pri povezovanju, nadzorujemo dopustnost vzpostavljanja povezave)

Izolirano filtriranje: dostopovni seznam

- dostopovni seznam (angl. access control list, ACL)
- tabela pravil, upošteva (procesira) se jo od vrha proti dnu
- zapisi so par (*pogoj, akcija*)
- primer: onemogoči ves promet razen WWW navzven in DNS v obe smeri

izvorni naslov	ciljni naslov	protokol	izvorna vrata	ciljna vrata	zastavica	akcija
222.22/16	izven 222.22/16	TCP	> 1023	80	any	dovoli
izven 222.22/16	222.22/16	TCP	80	> 1023	ACK	dovoli
222.22/16	izven 222.22/16	UDP	> 1023	53	---	dovoli
izven 222.22/16	222.22/16	UDP	53	> 1023	----	dovoli
all	all	all	all	all	all	zavrzi

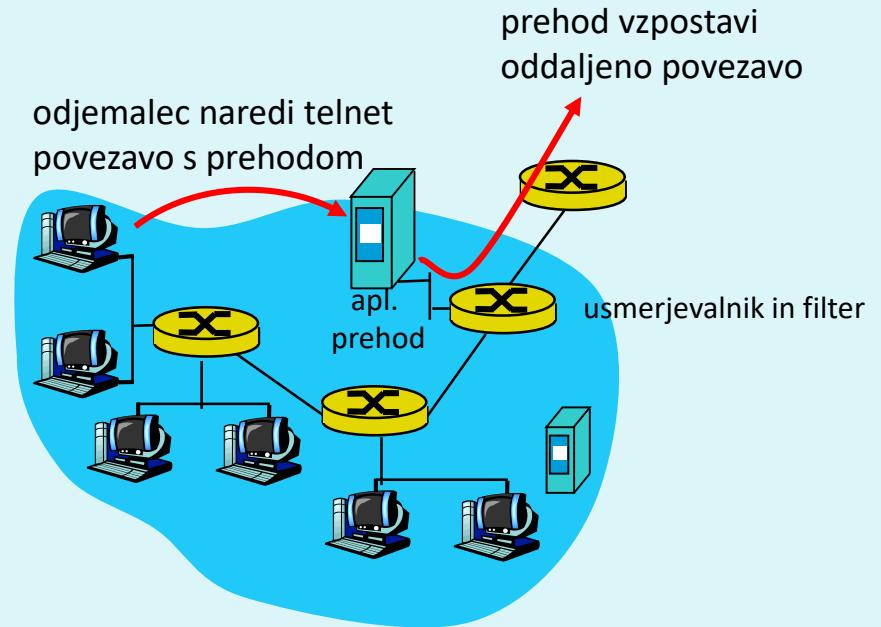
Filtriranje paketov v kontekstu

- angl. *stateful filter*, upošteva povezavo
 - izolirano filtriranje lahko dovoli vstop nesmiselnim paketom (npr. vrata = 80, ACK =1; čeprav notranji odjemalec ni vzpostavil povezave)
- IZBOLJŠAVA: **filtriranje paketov v kontekstu** spremišča in vodi evidenco o vsaki vzpostavljeni TCP povezavi
 - zabeleži vzpostavitev povezave (SYN) in njen konec (FIN): na tej podlagi odloči, ali so paketi smiselni
 - po preteku določenega časa obravnavaj povezavo kot neveljavno (timeout)
 - uporabljam podoben dostopovni seznam, ki določa, kdaj je potrebno kontrolirati veljavnost povezave (angl. *check connection*)

Filtriranje paketov v kontekstu

Aplikacijski prehodi

- omogočajo dodatno filtriranje glede na izbiro uporabnikov, ki lahko uporabljajo določeno storitev
- omogočajo filtriranje na podlagi podatkov na aplikacijskem nivoju poleg polj IP/TCP/UDP.



1. vsi uporabniki vzpostavljajo povezavo preko prehoda
2. samo za avtorizirane uporabnike prehod vzpostavi povezavo do ciljnega strežnika
3. prehod posreduje podatke med 2 povezavama
4. usmerjevalnik blokira vse povezave razen tistih, ki izvirajo od prehoda

Aplikacijski prehodi

Tudi aplikacijski prehodi imajo omejitve:

- če uporabniki potrebujejo več aplikacij (telnet, HTTP, FTP itd.), potrebuje vsaka aplikacija svoj aplikacijski prehod,
- klient je potrebno nastaviti, da se znajo povezati s prehodom (npr. IP naslov medstrežnika v brskalniku)



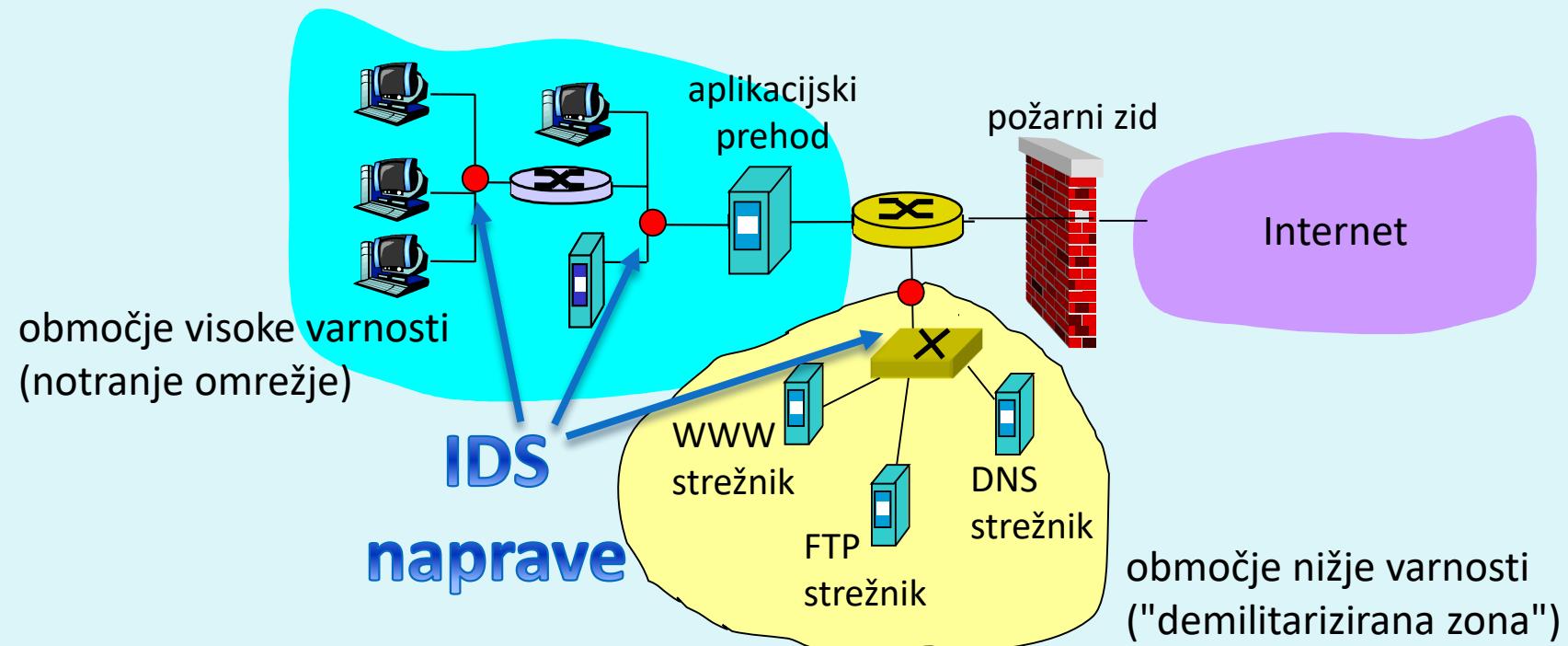
Sistemi za zaznavanje vdorov

- dodatna naprava - IDS, ki izvaja **poglobljeno analizo paketov**. Na podlagi vstopa sumljivih paketov v omrežje lahko naprava prepreči njihov vstop ali razpošlje obvestila.
 - sistem za zaznavanje vdorov (IDS) pošlje sporočilo o potencialno škodljivem prometu
 - sistem za preprečevanje vdorov (IPS) ukrepa pri pojavitvi sumljivega prometa
- primeri: Cisco, CheckPoint, Snort IDS

Načini zaznavanja vdorov

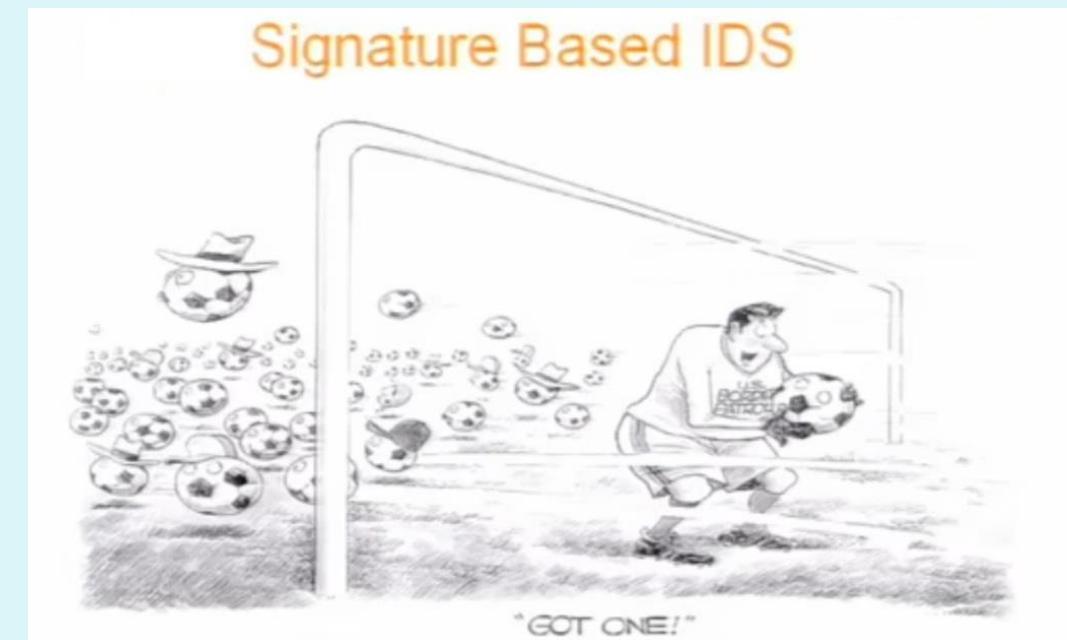
Kako deluje IDS/IPS?

- primerjava s shranjenimi vzorci napadov (angl. **signatures**)
- opazovanje netipičnega prometa (angl. **anomaly-based**)



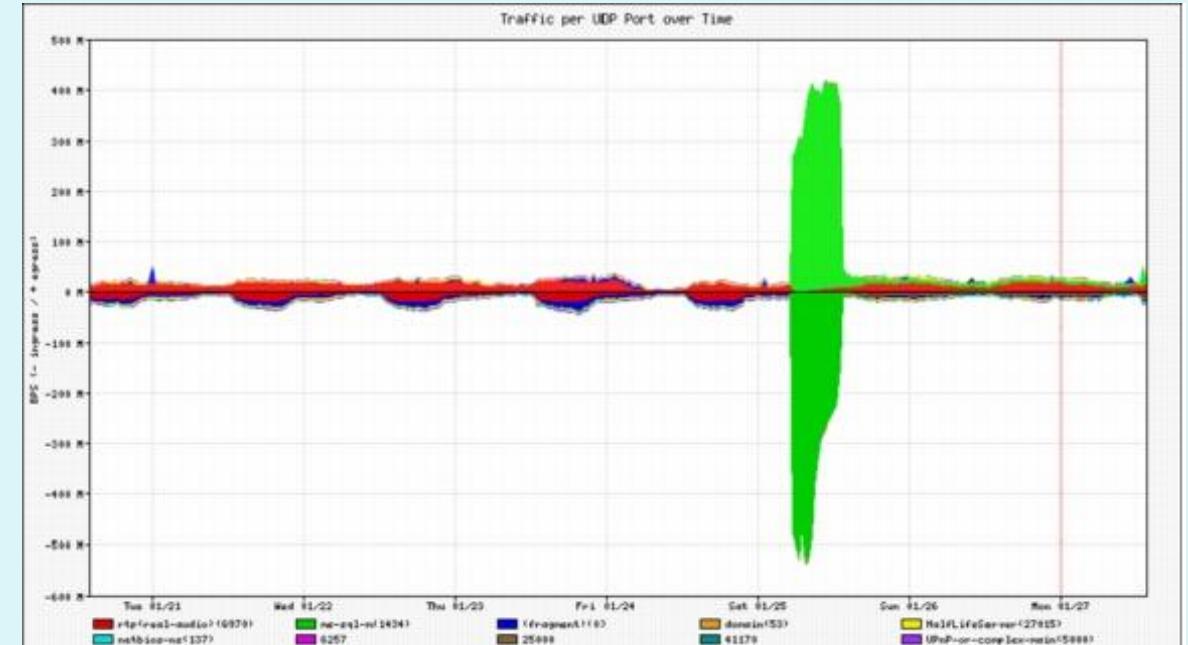
Zaznavanje z vzorci napadov

- vzorci napadov lahko hranijo izvorni IP, ponorni IP, protokol, zaporedje bitov v podatkih paketa, lahko so vezani na serijo paketov
- varnost je torej odvisna od baze znanih vzorcev; IDS/IPS slabo zaznava še nevidene napade
- možni lažni alarmi
- zahtevno procesiranje (lahko spregleda napad)



Zaznavanje netipičnega prometa

- sistem opazuje običajen promet in izračuna statistike, vezane nanj
- sistem reagira na statistično neobičajen promet (npr. nenadno velik delež ICMP paketov)
- možno zaznavanje še nevidenih napadov
- težko ločevanje med normalnim in nenavadnim prometom



Primer IDS/IPS sistema

- Snort IDS

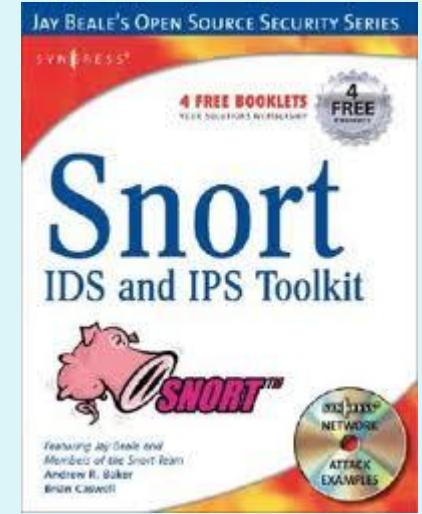
- public-domain, odprtokodni IDS za Linux, UNIX, Windows (uporablja isto knjižnico za branje omrežnega prometa kot Wireshark)
- primer vzorca napada

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any  
(msg:"ICMP PING NMAP"; dsiz: 0; itype: 8; )
```

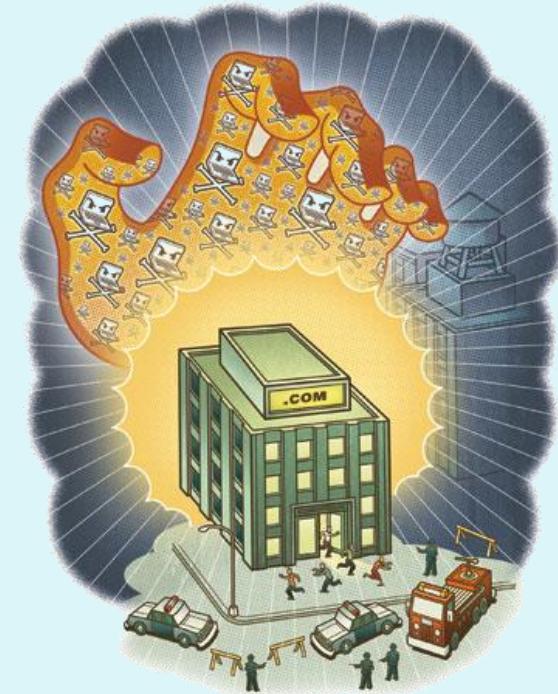
sporočilo za administratorja

prazen paket (dolžina 0) in
ICMP tip 8 (=PING) sta
lastnosti NMAP napada

reagiraj na VES DOHODNI
ICMP promet



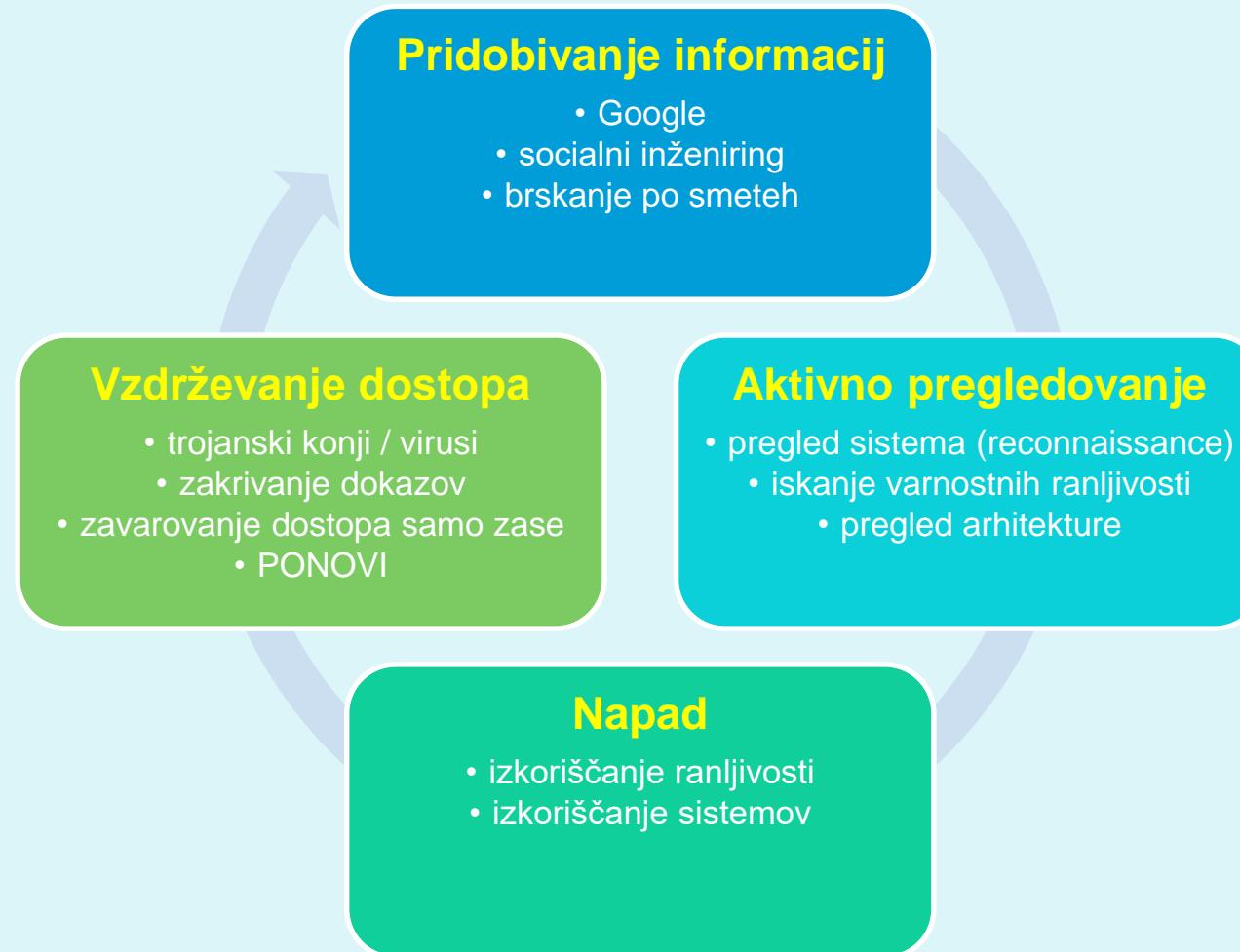
Napadi in grožnje



Pogosti napadi na omrežne sisteme

- **NAMEN?** Namenjeni so škodovanju ali obhodu računalniških in omrežnih funkcij.
- **ZAKAJ?** Denarna dobrobit, škodovalnost, poneverbe, ekonomske dobrobiti, čast in slava?
- **KAKO?** Ogrožanje zaupnosti, integritete in razpoložljivosti omrežnih sistemov
 - napadi s spreminjanjem informacij (*modification attack*)
 - zanikanje komunikacije (*repudiation attack*)
 - odpoved delovanja sistema (*denial-of-service attack*)
 - nepooblaščen dostop (*access attack*)
 - ...

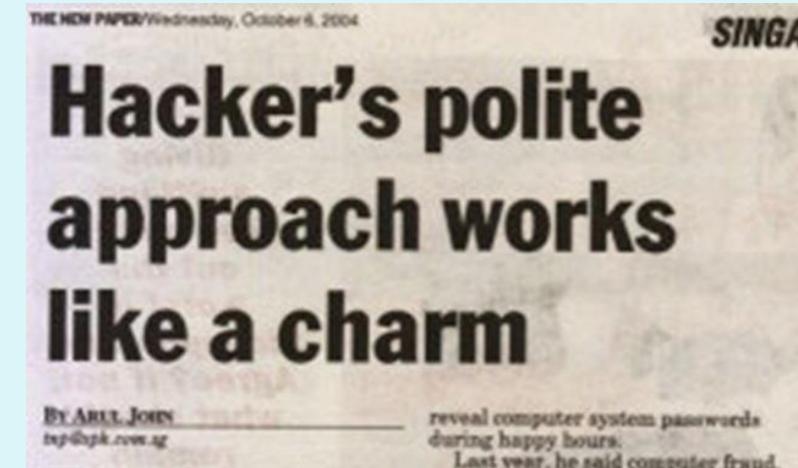
Pogosti napadi na omrežne sisteme



Pogosti napadi

1. prisluškovanje in ponarejanje sporočil
2. matematični napadi na kriptografske algoritme in ključe
3. ugibanje gesel (brute force, napad s slovarjem)
4. virusi, črvi, trojanci
5. izkoriščanje šibkosti v programski opremi
6. socialni inženiring (preko e-maila, telefona, servisov)

SOCIAL ENGINEERING SPECIALIST
Because there is no patch for human stupidity

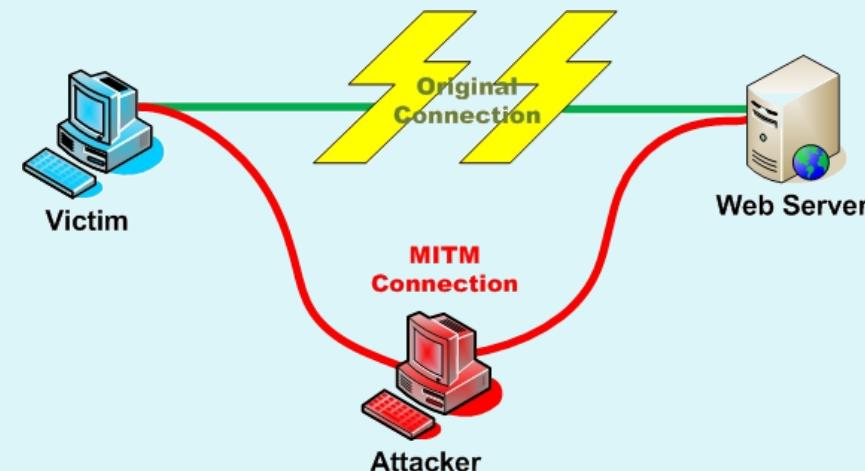


Pogosti napadi

7. **pregled vrat** (*port scan*): napadalec testira, kateri strežniki so delajoči (npr. ping) in katere storitve ponujajo. Napadalec lahko pridobiva podatke o sistemu: DNS, storitve, operacijski sistemi)
8. **brskanje po smeteh** (*dumpster diving*): način, s katerim lahko napadalci pridejo do informacij o sistemu (navodil za uporabo, seznamov gesel, telefonskih številk, opisa organizacije dela)
9. **rojstnodnevni napad** (*birthday attack*): je napad na zgoščevalne funkcije, za katere zahtevamo, da nobeni dve sporočili ne generirata iste zgoščene vrednosti. Pri slabših funkcijah napadalec išče sporočilo, ki bo dalo isto zgoščeno vrednost.

Pogosti napadi

10. **zadnja vrata (back door)**: napadalec zaobide varnostne kontrole in dostopi do sistema preko druge poti,
11. **ponarejanje IP naslovov (IP spoofing)**: napadalec prepriča ciljni sistem, da je nekdo drug (poznan) s spremenjanjem paketov,
12. **presteganje komunikacije (man-in-the-middle)**: napadalec prestreže komunikacijo in se obnaša, kot da je ciljni sistem (pri uporabi certifikatov lahko žrtev uporablja tudi javni ključ napadalca)



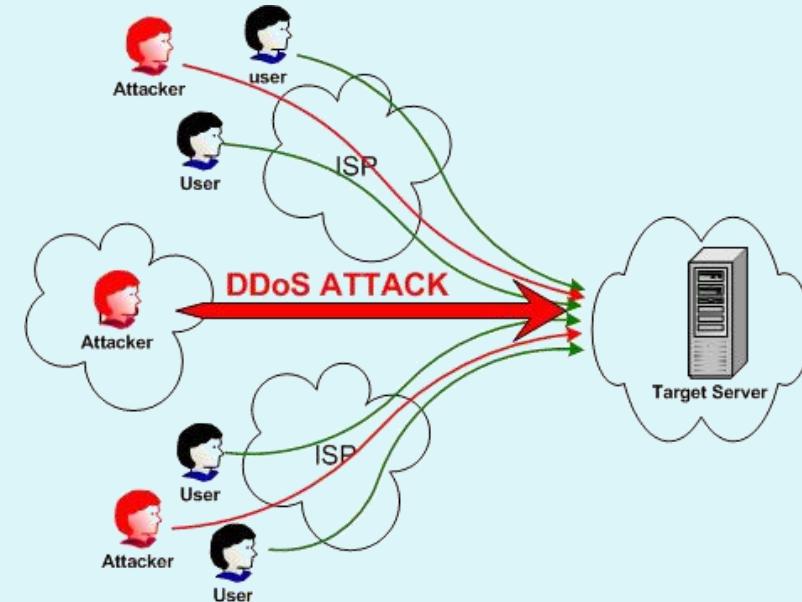
Pogosti napadi

13. **ponovitev komunikacije (replay)**: napadalec prestreže in shrani stara sporočila ter jih ponovno pošlje kasneje, predstavljač se kot eden izmed udeležencev
 - kako preprečimo napade s ponovitvijo komunikacije?
14. **ugrabitev TCP sej (TCP hijacking)**: napadalec prekine komunikacijo med uporabnikoma in se vrine v mesto enega od njiju; drugi verjame, da še vedno komunicira s prvim
 - kaj napadalec pridobi s tem?
15. **napadi s fragmentacijo (fragmentation attack)**: z razbijanjem paketa na fragmente razdelimo glavo paketa med fragmente tako, da jih požarni zid ne more filtrirati
 - tiny fragment attack: deli glavo prvega paketa
 - overlapping fragment attack: napačen offset prepiše prejšnje pakete

Napadi DoS (1/5)

16. odpoved delovanja sistema (*Denial-of-Service*)

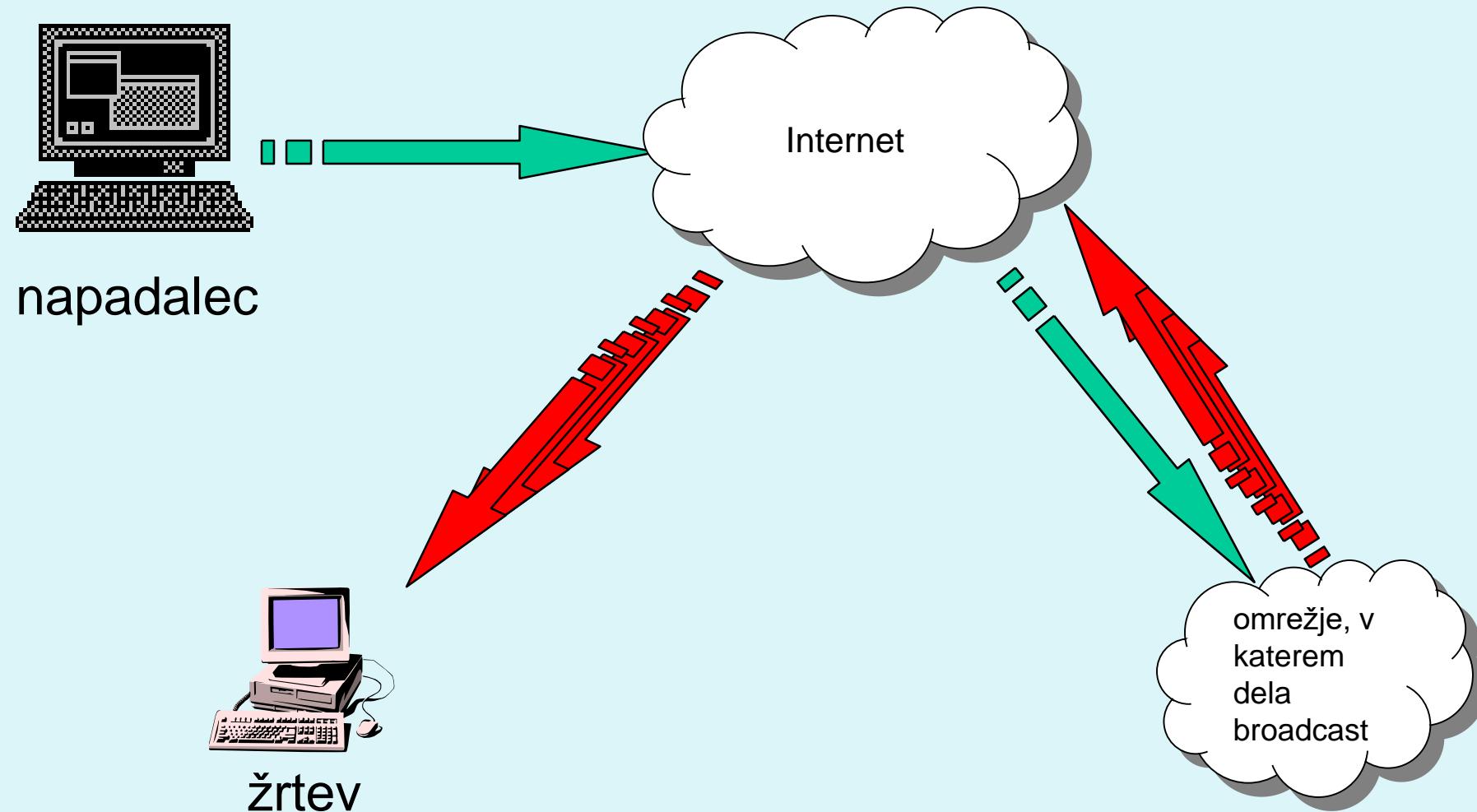
- cilj napadalca: obremeniti omrežne vire tako, da se ne lahko odzivati zahtevam regularnih uporabnikov (npr. vzpostavitev velikega števila povezav, zasedanje diskovnih kapacitet, ...)
- lahko je porazdeljen (distributed DOS = DDoS)



Napadi DoS (2/5)

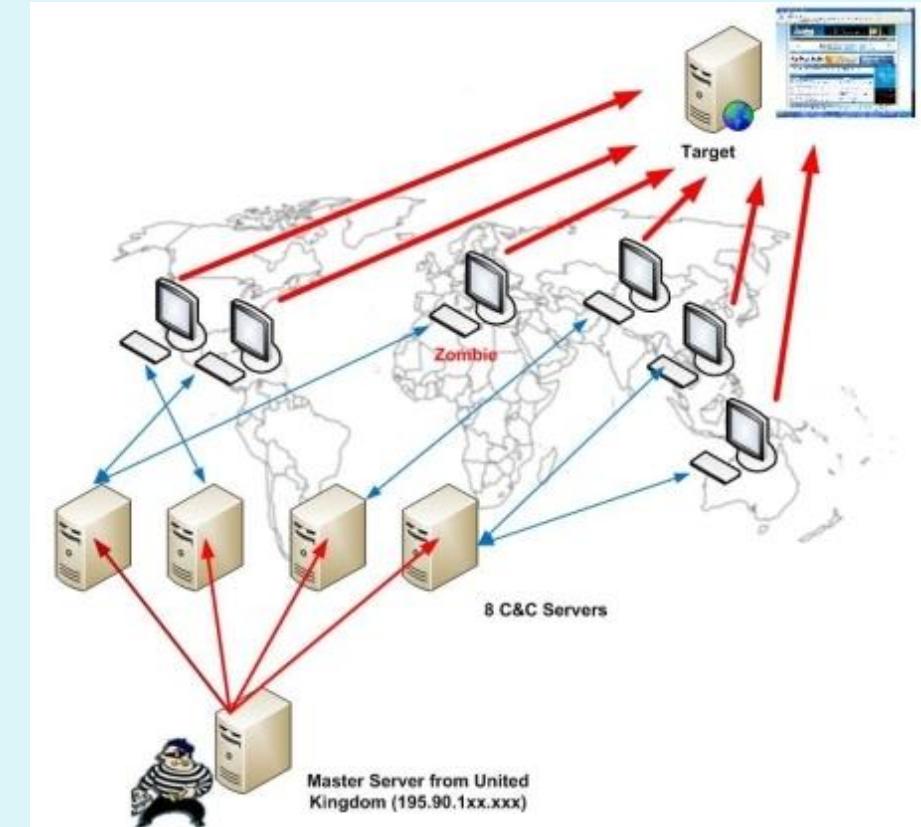
- primeri:
 - **prekoračitev medpomnilnika (buffer overflow)**: procesu pošljemo več podatkov, kot jih lahko sprejme (Ping of death: ICMP z več kot 65K podatkov je povzročil sesutje sistema)
 - **SYN napad**: napadalec pošlje veliko število zahtev za vzpostavitev povezave in se na odgovor sistema ne odzove; pride do preobremenitve vrste zahtev v sistemu
 - rešitev: omejitev števila odprtih povezav, timeout
 - **napad Teardrop**: napadalec spremeni podatke o številu in dolžini fragmentov v IP paketu, kar zmede prejemnika
 - **napad Smurf** (naslednja prosojnica): uporaba posrednega broadcasta za preobremenitev sistema
 - porazdeljen DDoS
 - uporabniki porazdeljenih omrežnih sistemov lahko da ne vejo, da je napadalna oprema nameščena pri njih

DoS Smurf (3/5)



Napadi DoS (4/5)

- Uporaba *bot*-ov (*web roBOT*) za organizacijo napadov na ciljni sistem
 - boti so lahko računalniki, okuženi s trojanskimi konji
 - njihovi uporabniki običajno ne vejo, da sodelujejo v napadu



Napadi DoS (5/5)

- subjekti v napadu: **napadalec**, centralni računalnik za **krmiljenje botov (herder)**, **boti (zombie)**, **cilj**

