



Kriptografija i kriptoanaliza

izv. prof. dr. sc. Ante Đerek i prof. dr. sc. Marin Golub

listopad 2025.

Sadržaj

1. Uvod u kriptografiju i kriptoanalizu
2. Simetrični kriptosustavi (DES, 3DES, IDEA, AES)
3. Funkcije za izračunavanje sažetka poruke (MD5, SHA)
4. Autentifikacijsko kriptiranje
5. Napadi na kriptosustave, kriptoanaliza
6. Asimetrični kriptosustavi (RSA, ECC)
 - Digitalni potpis: RSA digitalni potpis i DSA
7. Kriptografija prilagođena računalima s ograničenim mogućnostima (Lightweight Crypto)
8. Kvantna i post-kvantna kriptografija

Literatura

- [1] Christof Paar, Jan Pelzl, ***Understanding Cryptography***, Springer-Verlag Berlin Heidelberg, 2009.
- [2] L. Budin, M. Golub, D. Jakobović, L. Jelenković, *Sigurnost računalnih sustava*, poglavlje u knjizi ***Operacijski sustavi***, Element, Zagreb, 3. izdanje 2013.
- [3] ***Sigurnost računalnih sustava, zbirka studentskih radova***, dostupno na Internet adresi: <http://sigurnost.zemris.fer.hr>

1.

Uvod u kriptografiju i kriptoanalizu

- Osnovni pojmovi
- Prijetnje i napadi
- Podjela kriptografskih algoritama
- Jesu li i koliko su kriptografski algoritmi sigurni?

Osnovni pojmovi

Kriptologija = kriptografija + kriptoanaliza

Kriptografija

- znanstvena disciplina (ili umjetnost?) sastavljanja poruka sa ciljem skrivanja sadržaja poruka

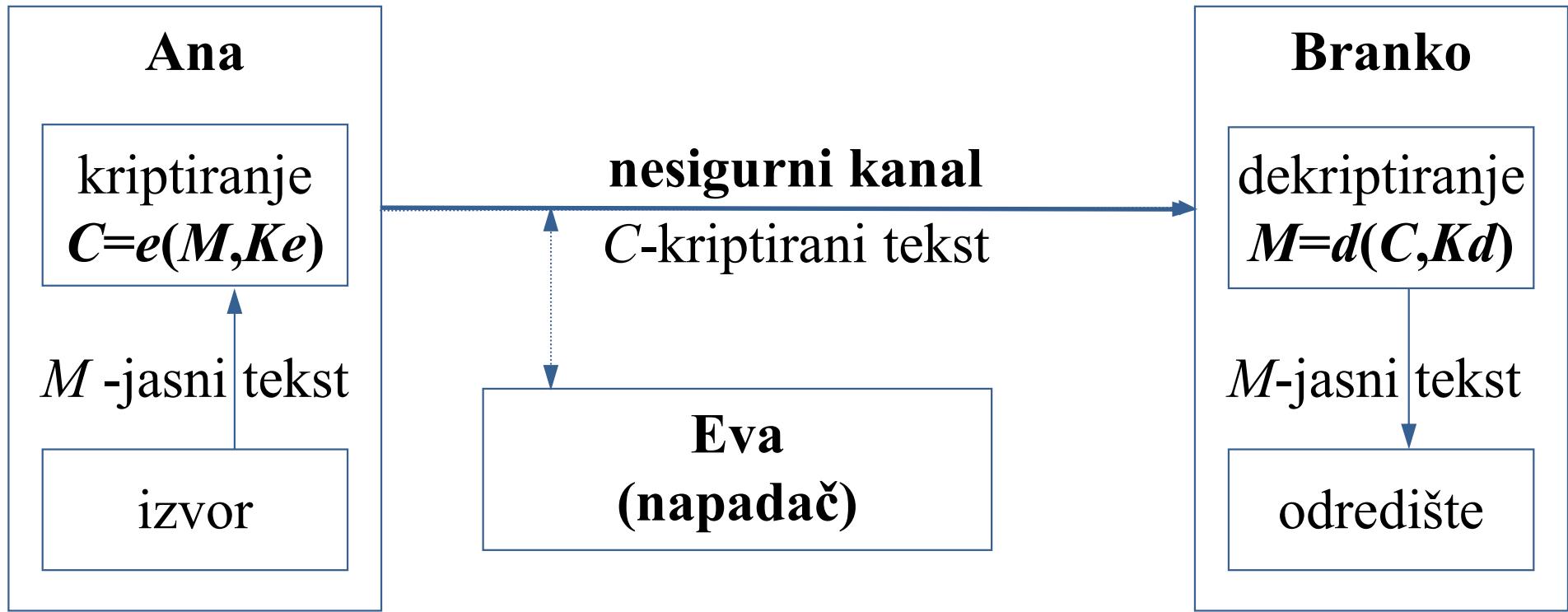
Kriptoanaliza

- znanstvena disciplina koja se bavi analizom skrivenih aspekata sustava i koristi se kako bi se ispitala (ili narušila) sigurnost kriptografskog sustava

Kriptoanaliza

- izvedenica iz grčkih riječi
 - *kryptós* – skriven i
 - *analyein* – rastavljati
- analiza informacijskog sustava u svrhu pronađaska **skrivenih aspekata sustava**
- obuhvaća primjerice
 - diferencijalnu kriptoanalizu
 - linearnu kriptoanalizu
 - analizu propusta u implementaciji
- uspješnost kriptoanalize ocjenjuje se **uspoređivanjem s napadom grubom silom** odnosno ispitivanjem svih mogućih ključeva

Osnovni pojmovi i oznake



Na ovom će se predmetu pojmovi kriptiranje i šifriranje koristiti na sljedeći način:

- šifriranje/dešifriranje - klasična kriptografija
- kriptiranje/dekriptiranje - moderna kriptografija

Kratko ponavljanje gradiva iz predmeta Sigurnost računalnih sustava

Osnovni pojmovi

identifikacija = predstavljanje

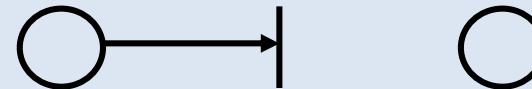
autentifikacija = identifikacija + verifikacija

autorizacija = autentifikacija + provjera prava pristupa

Prijetnje i napadi

1. prisluškivanje

2. prekidanje

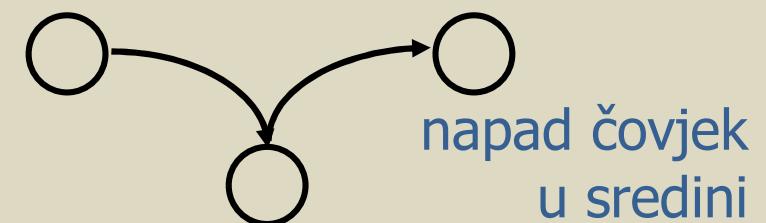
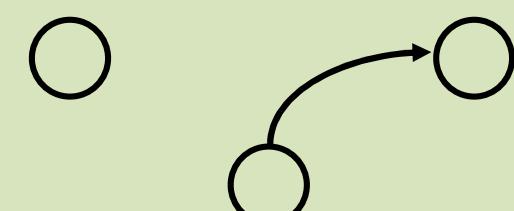
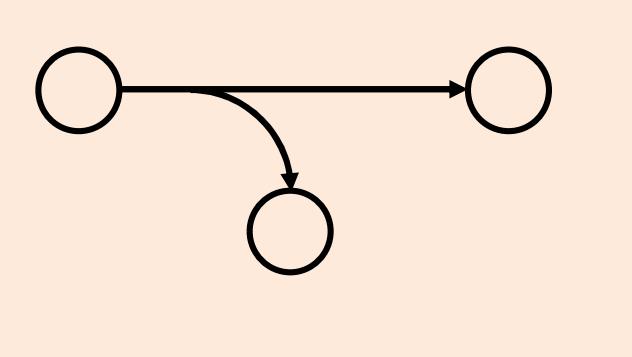


3. lažno predstavljanje

4. ponovno odašiljanje
snimljenih starih
paketa

5. modifikacija paketa

6. poricanje



napad čovjek
u sredini

Prijetnje i napadi

1. prisluškivanje
2. prekidanje
3. lažno predstavljanje
4. ponovno odašiljanje snimljenih starih paketa
5. modifikacija paketa
6. poricanje

Sigurnosni zahtjevi

1. tajnost
2. autentičnost
3. neporecivost
4. integritet
5. kontrola pristupa
6. raspoloživost

Sigurnosni zahtjev: **povjerljivost**

- pojam koji se koristi primjerice u kratici *CIA* (engl. *Confidentiality, Integrity and Availability*)
- sigurnosni zahtjev koji štiti informacije od neautoriziranog pristupa
 - tj. samo autorizirani korisnici smiju pristupiti osjetljivim podacima
- ostvaruje se kombinacijom autentičnosti i tajnosti ali i provjerom prava pristupa
- **povjerljivost ≠ tajnost**
 - mada se ta dva pojma često poistovjećuju
 - povjerljivost se može ostvariti uz pomoć **tajnosti**
- ali, može se ostvariti i bez tajnosti, npr.:
 - samo uz pomoć osiguravanja **autentičnosti** i provjere **prava pristupa**

Podjela kriptografskih algoritama

- Klasični
 - supstitucija
 - transpozicija
- Mehanički strojevi
- Moderni
 - simetrični
 - blok (AES, DES, PRINCE, PRESENT, Twofish, RC6, ...)
 - protočni ili kriptiranje toka podataka
(Salsa20, Trivium, Mickey, Grain, Achterbahn, Rakaposhi, ...)
 - $Ke = Kd = \mathbf{K}$ (simetrični, sjednički ili tajni ključ)
 - asimetrični
 - $Ke \neq Kd$ (\mathbf{P} - javni i \mathbf{S} - privatni ključ)
 - funkcije za izračunavanje sažetka poruke (*hash*)

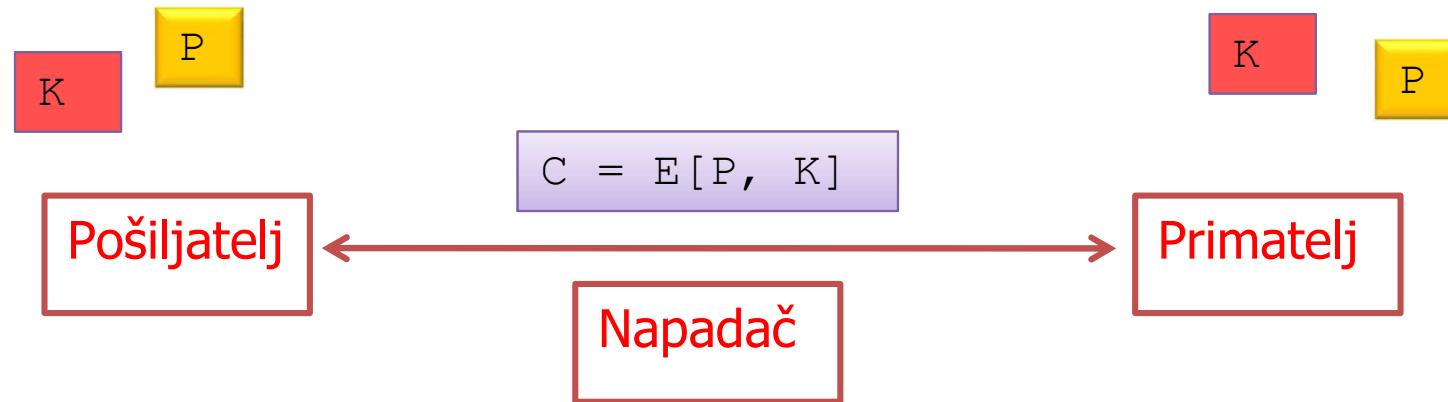
Jesu li i koliko su kriptoalgoritmi sigurni?

- postoje specijalizirana računala za napad grubom silom na DES
kriptosustav: COPACOBANA (*A Cost-Optimized PArallel COde Breaker*)
- 12.12.2009. faktoriziran RSA-768
- na kvantnom računalu je riješen problem faktoriziranja velikih brojeva i problem diskretnog logaritma
- 17.8.2004. - kineski i francuski znanstvenici su objavili članak pod naslovom: "*Kolizija za hash funkcije: MD4, MD5, Haval-128 i RIPEMD*"
- 13.2.2005. - kineski znanstvenici: "*Collision Search Attacks on SHA-1*"
- napadi koji koriste sporedna svojstva uređaja (*Side-Channel Attacks, SCA*)

2.

Simetrični kriptosustavi

Simetrična enkripcija



Kerckhoffov princip

- Kriptosustav mora biti siguran i onda kada su sve informacije o kriptosustavu javno poznate, osim tajnog ključa.
- Simetrični kriptosustavi temelje se na jednostavnoj logičkoj operaciji isključivo ILI (XOR):
$$C = M \oplus K \quad M = C \oplus K$$
$$M = (M \oplus K) \oplus K$$
- ONE TIME PAD – jednokratna bilježnica

Savršena povjerljivost

- Claude Shannon, 1946
- jednokratna bilježnica pruža *savršenu povjerljivost*:
 - za svaku poruku $m \in \{0, 1\}^n$ i šifrat $c \in \{0, 1\}^n$ i vrijedi:

$$P_{k \leftarrow \{0,1\}^n}(E(m, k) = c) = \frac{1}{2^n}.$$

- jednokratna bilježnica u praksi:



Izvor: www.cryptomuseum.com

Kriptografija i kriptoanaliza 17

Jednokratna bilježnica – nedostaci

- Ključ
 - mora se generirati potpuno i uistinu slučajno!
 - mora biti jednako velik kao i poruka!
 - smije se koristiti najviše jednom!

$$c_1 = m_1 \oplus k$$

$$c_2 = m_2 \oplus k$$

$$c_1 \oplus c_2 = m_1 \oplus m_2$$

- Moguće je na predvidiv način izmijeniti poruku (engl. *malleable encryption*)!
 - naravno, samo ako nam je poznata kriptirana poruka

$$c_1 = OTP(m_1, k) = m_1 \oplus k$$

$$c_2 = c_1 \oplus m_1 \oplus m_2 = m_1 \oplus k \oplus m_1 \oplus m_2 = m_2 \oplus k = OTP(m_2, k)$$



Klasična kriptografija

- nećemo se baviti klasičnom kriptografijom, no, ipak ćemo navesti nekoliko primjera

Šifriranje uz pomoć papira i olovke

- supstitucijske šifre
 - Cezarova šifra
 - Vigenèreova šifra (1586.)
 - Playfairova šifra (1854.)
 - Hillova šifra (1929.)
- transpozicijske šifre

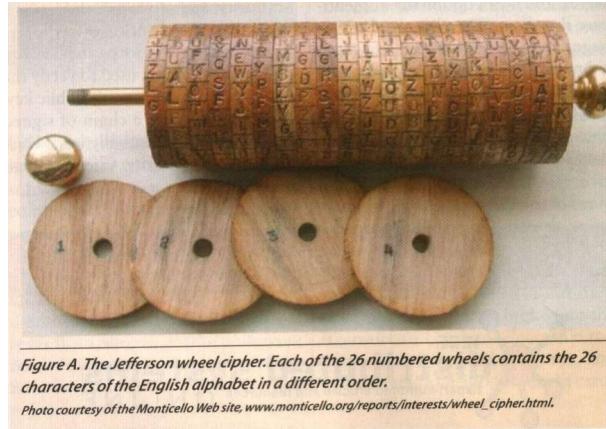
Šifriranje uz pomoć naprava

- Jeffersonov kotač (krajem 18. stoljeća Thomas Jefferson)
- električni stroj za kodiranje (1915. Edward Hugh Hebern)
- Enigma (1918. Artur Scherbius)
- C-36 (1936. Boris Hagelin)
 - u američkoj vojsci ta je naprava nosila naziv M-209

Naprave za šifriranje



Enigma
(1918. Artur Scherbius)



Jeffersonov kotač
(krajem 18. stoljeća Thomas Jefferson)



električni stroj za kodiranje
(1915. Edward Hugh Hebern)



C-36
(1936. Boris Hagelin)
• u američkoj vojsci ta je
naprava nosila naziv M-209

Primjer supstitucijske šifre: Cezarova šifra

- najjednostavnija i najčešće korištena šifra
- monoalfabetska šifra: svako slovo se mijenja drugim slovom za jednak pomak, gdje je pomak ključ

$$\text{Pomak} = 3$$

Jasni tekst: a b c č ď e f g h i j k l m n o p r s š t u v z ž
Šifra: Č Ć D Ď E F G H I J K L M N O P R S Š T U V Z Ž A B C

Jasni tekst: A U T O
Šifra: Č Ž Z S

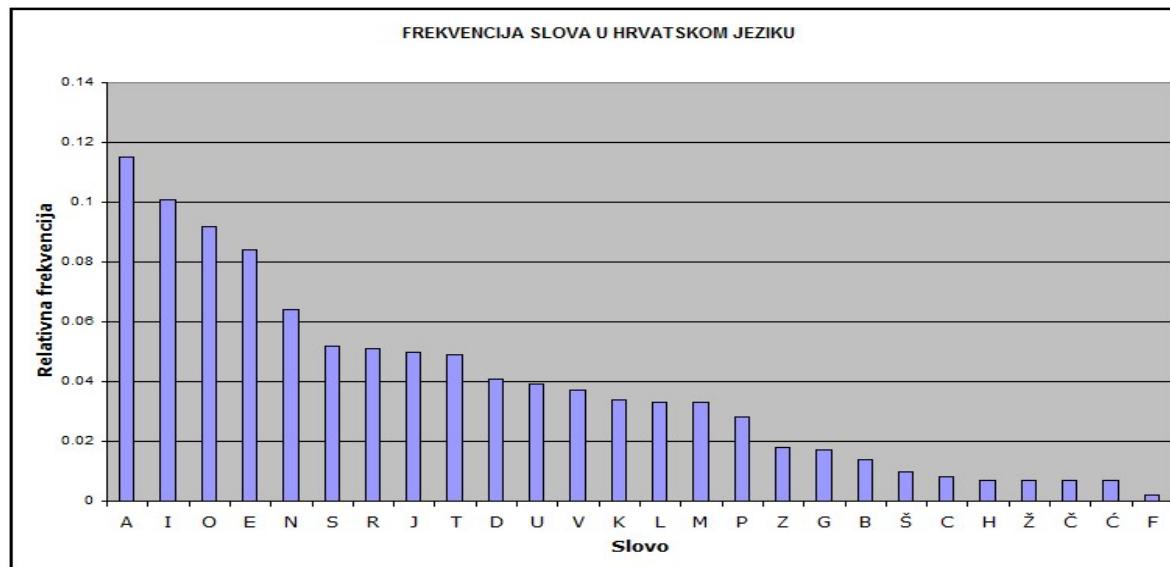
- jednostavan napad frekvencijskom analizom ako je poznat jezik

Frekvencijska analiza (1/4)

- uzeti u obzir frekvenciju slova (u promilima):

Tablica 1. Frekvencija slova u hrvatskom jeziku

A	I	O	E	N	S	R	J	T	U	D	K	V	L	M	P	C	Z	G	B	H	F
115	98	90	84	66	56	54	51	48	43	37	36	35	33	31	29	28	23	16	15	8	3



Izvor: wikipedia.org

... a prebrojali smo
najfrekventnije znakove u
nekom tekstu koji je šifriran
supstitucijskom šifrom:

Z 51
P 54
D 56
R 60
W 78
O 91

Frekvencijska analiza (2/4)

- ako znamo da je poruka šifrirana Cezarovom šifrom možemo uz pomoć frekvencijske analize iz šifrata:

VWZWUO UG FOEQCPGPO: RAJOMRIOPW ZPQSGDPG ERUW XG DRZWPW
PGKDRCRZEW W SJQZPIGDW JOMIRU KJIOPZEG EJRM DOZPOIQ GCGEPJRPKGDWEG,
JOXQDOJZPIO PG WDFRJVOWXUZEG W ERVQDWEOXWUZEG PGKDRCRHWUG
MOZDRIODQ DO JGMQCPOPWVO WZPJOMWIODUO, ZPIOJOPW DRIO MDODUO EJRM
VGSUQDOJRSR L JWMDOP WZPJOMWIODUO, JOMIWUOPW HRZLRSOJZPIR W
UOIDW ZGEPRJ EJRM WDRIOXWUG PG SRLJWDRZWPW QEQLDRV JOMIRUQ SJQZPIO,
AWPW QZPODARIO IWZREWK OEOSGVZEWK IJWUGSDRZPW W GPWXEWK EJWPGJWUO,
VUGZPR EJWPWXERH JOMVWZCUODUO W LJRLWPWIODUO PG UGSDOERZPW ZIWK
DUGDWK XCODARIO W AWPW LREJGPOXEO ZDOHO KJIOPZERH SJQZPIO.
Q WZLQDUGDUQ VWZWUG FOEQCPGPO RZCODUOVR ZG DO DOZG PGVGCUDG
IJWUGSDRZPW ERUG SOCUG JOMIWUOVR: IRSGXO ZVR DOXWRDOCDO
IWZRERZERCZEO W WZPJOMWIOXEO QZPODARIO Z WMIJZDWV DOZPOIDWXWVO W
ZPQSGDPWVO, XIJZPR LRIGMODO Z HRZLRSOJZPIRV, WMIJZDR RJHODWMWJODO W
VGSUQDOJRSR LJGLRMDOPCUWIO.

Frekvencijska analiza (3/4)

- dobiti nešto što nalikuje na tekst na hrvatskom jeziku:

.I.I.A .. A...N.NA: O..A.O.ANI .N...EN. .O.I .. EO.INI
N..EO.O..I IN..EI .A...O.AN.... O. EA.NA...N.ON..EI...,
.A..EA..N.A N. IE.O..A.I.... I .O..EI.A.I.... N..EO.O.I..
.A.EO.AE. EANANI.A I.N.A.I.AE.A, .N.A.ANI EO.A .EAE.A ..O.
.....EA.O.EO ..I.EANA I.N.A.I.AE.A, .A..I.ANI .O..O.A..N.O I
.A.EI ...NO. ..O. IEO.A.I.. N. .O..IEO.INIEO. .A..O..N.A,
.INI ..NAEO.A .I.O.I. A.A.....I. ..I....EO.NI I .NI..I. ..IN..I.A,
....NO ..INI..O. .A..I....AE.A I ..O.INI.AE.A N.EA.O.NI ..I.
E..EI. ..AEO.A I .INI .O...NA..A .EA.A ...AN..O.N.A.
. I....E..E.. .I.I.. .A...N.NA O..AE.A.O .. EA EA.. N.....E.
.I....EO.NI .O.. .A.... A..I.A.O: .O....A ..O EA.IOE.A.EA
.I.O.O..O....A I I.N.A.I.A..A ..NAEO.A . I....EI. EA.NA.EI.I.A I
.N....ENI.A,NO .O...AEA . .O..O.A..N.O., I....EO O..AEI.I.AEA I
.....EA.O.EOO.EAN..I.A.

- nisu sva slova pogodjena jer je premali tekst

– što ima više teksta, lakše ga je dešifrirati

Frekvencijska analiza (4/4)

- frekvencija slova na hrvatskom i engleskom jeziku:

A	I	O	E	N	S	R	J	T	D	U	V	K	L	M	P	Z	G	B	Š	C	H	Ž	Ć	Ć	F
117	104	94	85	64	53	50	49	47	42	40	38	37	35	34	27	18	17	15	10	7	6	5	4	3	1

E	T	A	O	I	N	S	H	R	D	L	C	U	M	W	F	G	Y	P	B	V	K	J	Q	X	Z
127	91	82	75	70	67	63	61	60	43	40	28	28	24	23	22	20	20	19	15	10	8	2	1	1	1

- frekvencija bigrama:

HR: 2.8% **je**; 1.5% **na**; 1% **an st an ni ko os ti ij no en**

EN: 3.2% **th**; 2.5% **he**; 1.2% **an in er re on es ti at**

- frekvencija trigrama:

HR: 0.6% **ije**; 0.3-0.4% **sta ost jed koj oje jen**

EN: 3.5% **the**; 1.1% **ing**; 1% **and**; 0.7% **ion tio ent ...**

Drugi primjer supstitucijske šifre: Vigenèrova šifra

- polialfabetska šifra: niz od nekoliko Cezarovih šifri s različitim pomacima
- postupak šifriranja: $S_i = A_i + K_i \text{ mod(broj_slova=27)}$

	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	
Alfabet (A):	a	b	c	č	ć	d	đ	e	f	g	h	i	j	k	l	m	n	o	p	r	s	š	t	u	v	ž

Ključ = „ključ“ = {13, 14, 12, 23, 3}

Jasni tekst: A U T O M O B I L L=14 u=23

Ključ (**K**): k l j u č k l j u (14 + 23) mod 27 = 37 mod 27 = 10 = H

Šifra (**S**): K H E K P Č M U H

Treći primjer supstitucijske šifre: Playfairova šifra

- polialfabetska, bigramska šifra: šifriraju se parovi znakova
- ako je neparan broj slova, dodaje se neko slovo npr. Ž
- koristi se matrica 5x5 slova (ako ima više slova, neka se poistovjećuju, npr. ĐĐ i SŠ) koja se stvara na temelju ključa (neka je ključ = „OVOJEKLJUČ“):

Alfabet: a b c č ď ē f g h i j k l m n o p r s š t u v z ž

O	V	J	E	K	Pravila:
L	U	Č	A	B	1. par slova u istom retku posmiču se udesno (npr. AU=BČ)
C	Ć	ĐĐ	F	G	2. par slova u istom stupcu posmiču se dolje (OR=LO)
H	I	M	N	P	3. par slova čine pravokutnik i mijenjaju se sa slovima na
R	SŠ	T	Z	Ž	suprotnim stranama pravokutnika (npr. TO=RJ ili BI=UP)

Jasni tekst: AU TO MO BI LŽ

Šifra: BČ RJ HJ UP BR

Četvrti primjer supstitucijske šifre: Hillova šifra

- poligramska šifra: šifrira se m znakova
 - ako duljina poruke nije djeljiva s m zadnji blok treba nadopuniti
 - ključ K je matrica $m \times m$

Alfabet (**A**): 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

Neka je $m = 3$ i $K = K^{-1} = \begin{bmatrix} 5 & 8 & 22 \\ 2 & 5 & 24 \\ 10 & 20 & 17 \end{bmatrix}$ Jasni tekst: subota (18, 20, 1, 14, 19, 0)

$$\text{Šifriranje: } (18 \ 20 \ 1) \begin{bmatrix} 5 & 8 & 22 \\ 2 & 5 & 24 \\ 10 & 20 & 17 \end{bmatrix} = (140 \ 264 \ 893) \bmod 26 = (\textcolor{red}{10 \ 4 \ 9}) = \text{ k e j}$$

$$(14 \ 19 \ 0) \begin{bmatrix} 5 & 8 & 22 \\ 2 & 5 & 24 \\ 10 & 20 & 17 \end{bmatrix} = (108 \ 207 \ 764) \text{ mod } 26 = (4 \ 25 \ 10) = e \ z \ k$$

Šifra: kejezk (10, 4, 9, 4, 25, 10)

Hillova šifra – postupak dešifriranja

- za dešifriranje koristi se inverz matrice K , tj. K^{-1}
- izvorno autor predlaže da je $K = K^{-1}$
 - ali se time značajno smanjuje prostor svih mogućih ključeva

1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

Alfabet (**A**) : 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

$$K = K^{-1} = \begin{bmatrix} 5 & 8 & 22 \\ 2 & 5 & 24 \\ 10 & 20 & 17 \end{bmatrix}$$

Šifra: kejezk (10, 4, 9, 4, 25, 10)

Dešifriranje: (10 4 9) $\begin{bmatrix} 5 & 8 & 22 \\ 2 & 5 & 24 \\ 10 & 20 & 17 \end{bmatrix} = (148 \ 280 \ 469) \ mod \ 26 = (18 \ 20 \ 1) = \text{s u b}$

(4 25 10) $\begin{bmatrix} 5 & 8 & 22 \\ 2 & 5 & 24 \\ 10 & 20 & 17 \end{bmatrix} = (170 \ 357 \ 858) \ mod \ 26 = (14 \ 19 \ 0) = \text{o t a}$

Jasni tekst: subota (18, 20, 1, 14, 19, 0)

Kriptografija i kriptoanaliza 29

Primjer transpozicijske šifre: Stupčana transpozicija

- umjesto zamjene znakova koristi zamjenu položaja elemenata otvorenog teksta
- frekvencije znakova šifrata su jednake kao i kod jasnog teksta
- ključ je permutacijski niz od m elemenata
- jasni tekst se upisuje u pravokutnik po retcima, a jedan redak ima m znakova
- zadnji redak se nadopunjuje proizvoljnim znakovima

Jasni tekst: danas je lijep dan

Ključ: 3 2 5 1 4
Jasni tekst: d **a** n **a** s
 j **e** l **i** j
 e **p** d **a** n
Šifra: **aia****aep****djesjnld**

- u stupcu 1 piše „**aia**” i tako počinje šifrirani tekst
- u stupcu 2 piše „**aep**” pa se tako nastavlja šifrat
- slijedi „**dje**”, itd.

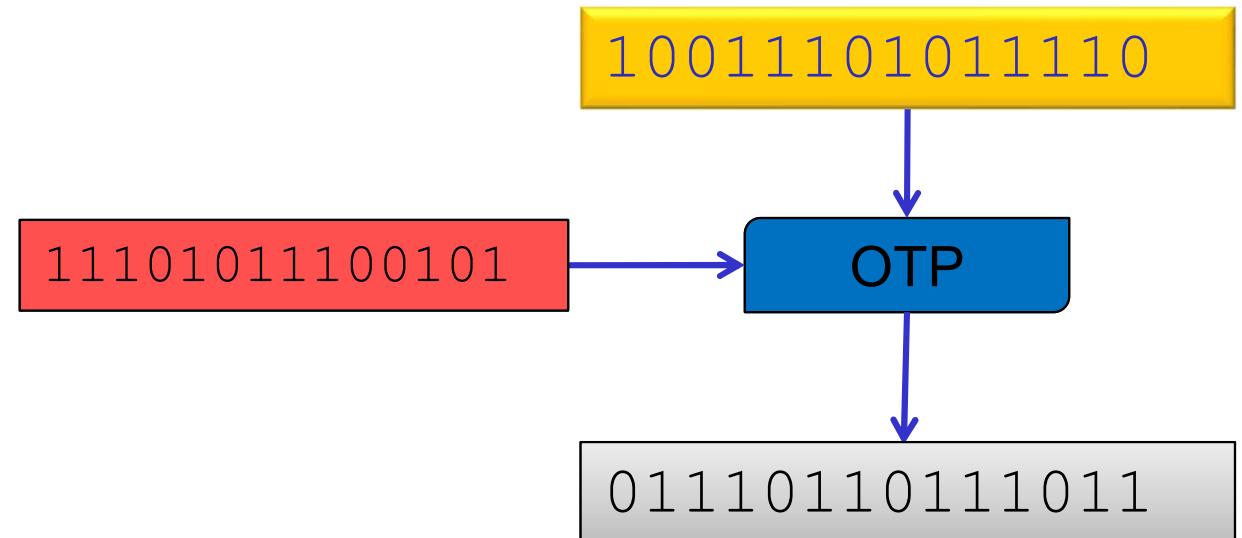
Simetrična enkripcija – definicija

- Neka su K , M i C konačni skupovi:
 - K - prostor ključeva,
 - M - prostor jasnih tekstova i
 - C - prostor skrivenih tekstova.
- Simetrična enkripcija je par algoritama E i D ($E: M \times K \rightarrow C$, $D: C \times K \rightarrow M$) gdje za svaki $k \in K$ i $m \in M$ vrijedi

$$D(E(m, k), k) = m.$$

Primjer: jednokratna bilježnica

- $M = K = C = \{0, 1\}^n$
- $E(m, k) = m \oplus k$
- $D(c, k) = c \oplus k$



Sigurnost

- Neformalno:

Simetrična enkripcija je sigurna ako je napadaču *jako teško* na temelju skrivenog teksta

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

odrediti *bilo što* o jasnom tekstu M .

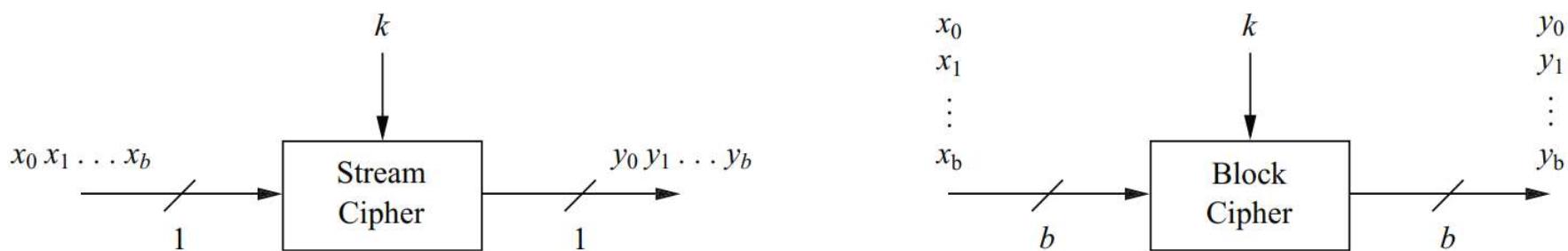
Različite vrste sigurnosti

prema tome što napadač ima na raspolaganju:

- samo jedan skriveni tekst
- jedan par (m_i, c_i)
- puno parova (m_i, c_i) gdje je $c_i = E(m_i, k)$
 - Napad poznatim izvornim tekstrom / *known plaintext attack*
- mogućnost da dobije $c_i = E(m_i, k)$ za m_i po izboru
 - Napad odabranim izvornim tekstrom / *chosen plaintext attack*
- mogućnost da dobije $m_i = D(c_i, k)$ za c_i po izboru
 - Napad odabranim skrivenim tekstrom / *chosen ciphertext attack*
- ...

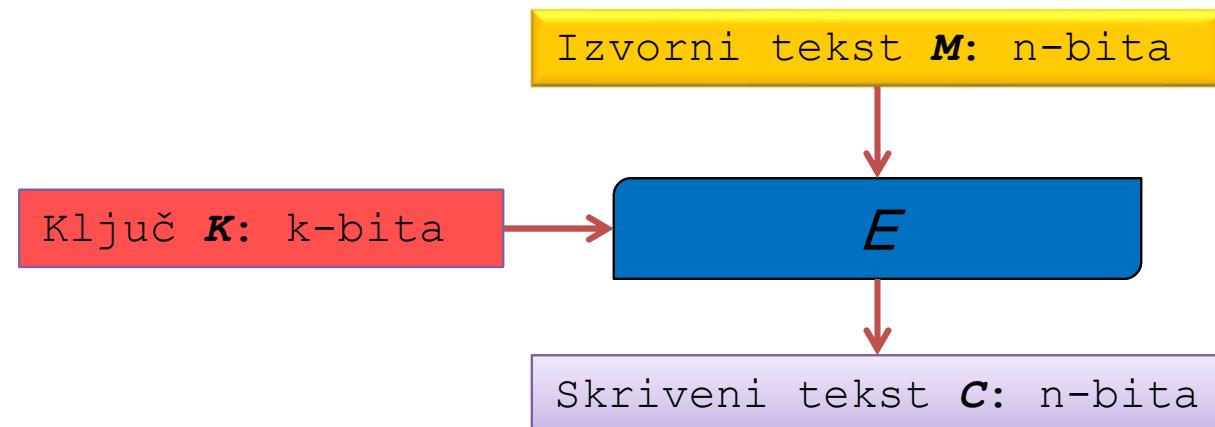
Vrste simetrične enkripcije

- Protočna enkripcija (*eng. stream cipher*) ili kriptiranje toka podataka
 - kriptira se jedan po jedan bit
- Sustavi kriptiranja bloka podataka (*eng. block cipher*)
 - kriptiraju se blokovi fiksne duljine



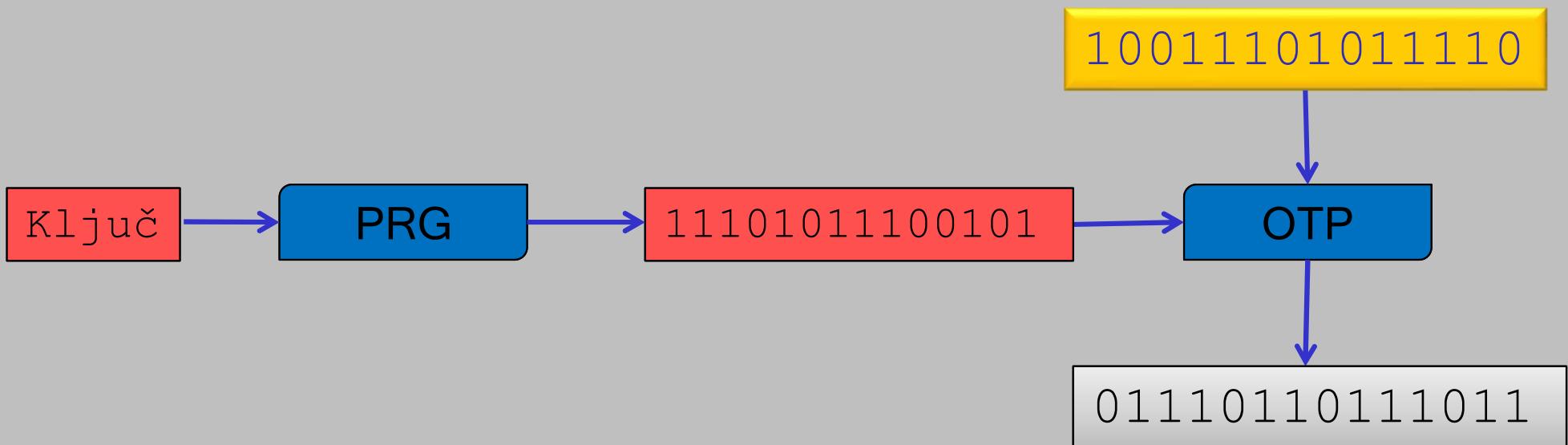
Sustav kriptiranja bloka

- $M = C = \{0, 1\}^n$
- $K = \{0, 1\}^k$
- E i D su deterministički algoritmi.



Sustav kriptiranja toka podataka

- Ideja: umjesto slučajnog ključa koristimo *pseudoslučajni ključ*.
- Generator pseudoslučajnih brojeva na temelju ključa generira niz bitova koji se XOR-a s izvornim tekstom.



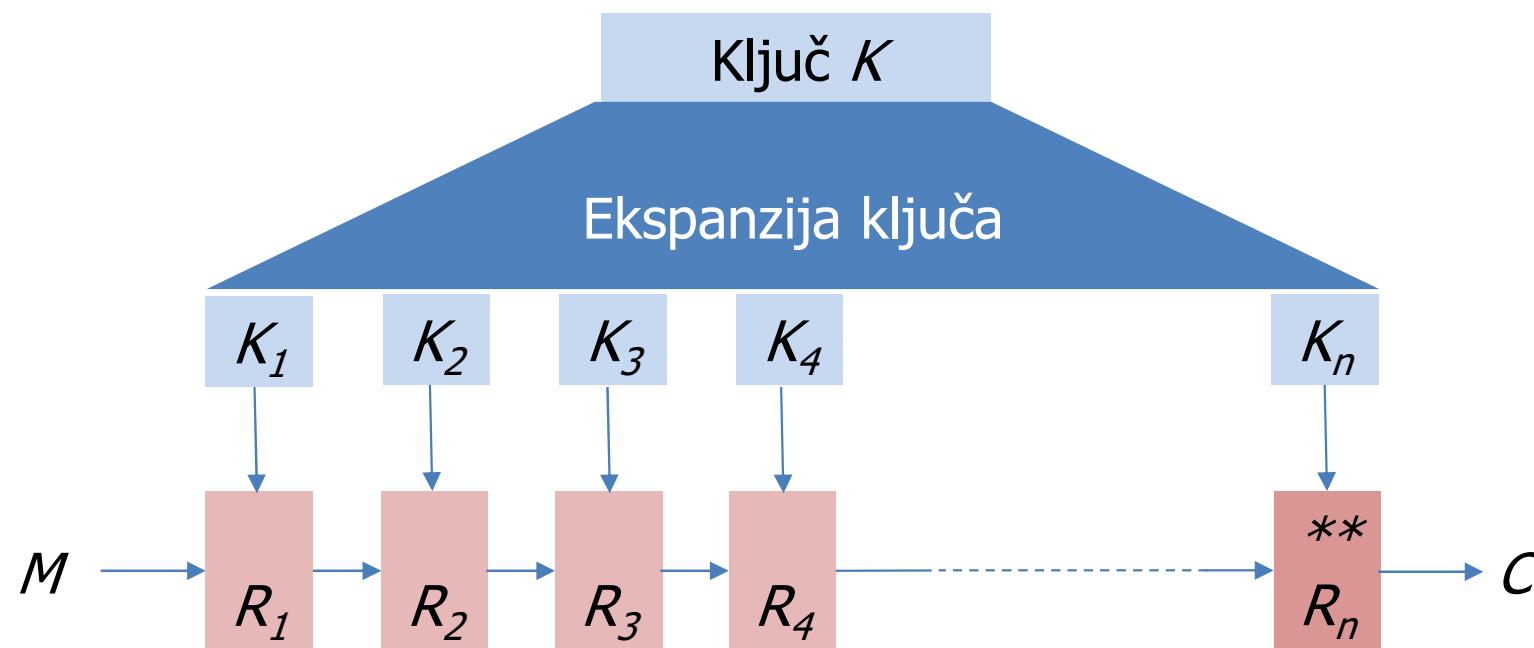
Primjeri sustava kriptiranja bloka

- DES (1970-te)
 - **n=64, k=56**, danas potpuno nesiguran zbog malog ključa
 - dugogodišnji standard, još uvijek se široko koristi
- 3DES, trostruki DES (1970-te)
 - **n=64, k=112 ili 168**
 - veća sigurnost s istim algoritmom kriptiranja
- IDEA (1991)
 - **n=64, k=128**
- Blowfish (1993)
 - **n=64, k=32–448**
- AES (1999)
 - **n=128 k=128, 192, 256**
 - standard od 2002., vrlo se široko koristi

Kako izgraditi sustav kriptiranja bloka?

- **Upozorenje:**
 - U ovom predmetu (kroz predavanja i vježbe) objašnjavamo kako iznutra rade ovakvi sustavi.
 - **Osmišljavanje novih kriptografskih sustava nije jedan od predviđenih ishoda znanja!**
 - **Ispravna i sigurna implementacija kriptografskih sustava nije jedan od predviđenih ishoda znanja!**
 - U praksi uvijek koristite dobro poznate sustave definirane standardima i implementirane u provjerenim bibliotekama!

Koraci* algoritma kriptiranja bloka



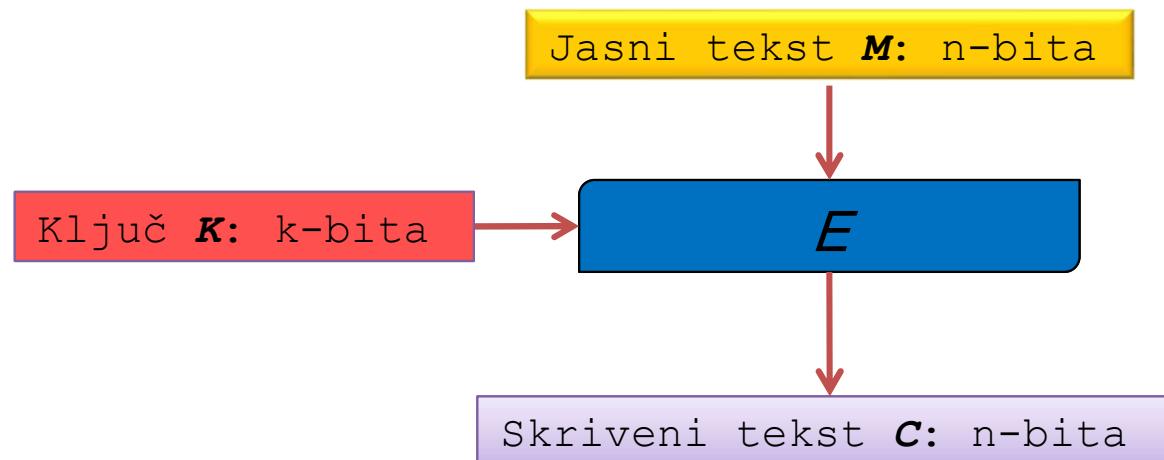
* iteracije ili *runde*

** zadnji korak je kod nekih algoritama drugačiji

Shannonova načela

- **Difuzija**

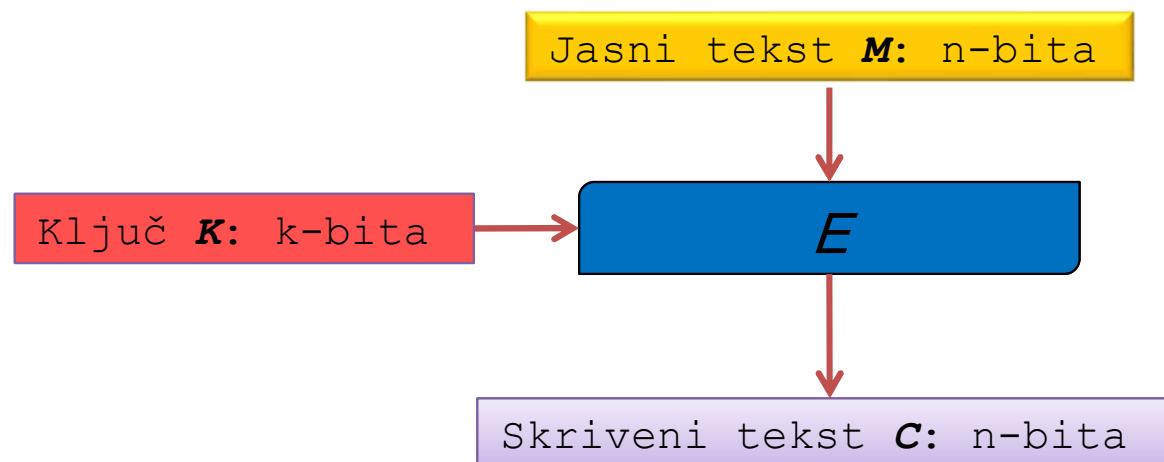
- svaki bit jasnog teksta kao i svaki bit tajnog ključa treba utjecati na mnogo bitova kriptiranog teksta
- promjena samo jednog bita jasnog teksta mora uzrokovati promjenu (statistički) polovicu bitova kriptiranog teksta
- ostvaruje se primjerice permutacijom i u više koraka algoritma



Shannonova načela

- **Konfuzija**

- međuzavisnost kriptiranog i jasnog teksta je previše složena da bi se mogla iskoristiti za razbijanje kriptosustava
- svaki bit kriptiranog teksta treba ovisiti o više bitova ključa ali tako da se pritom prikrije veza između njih
- ostvaruje se primjerice supstitucijom, tj. supstitucijskim tablicama



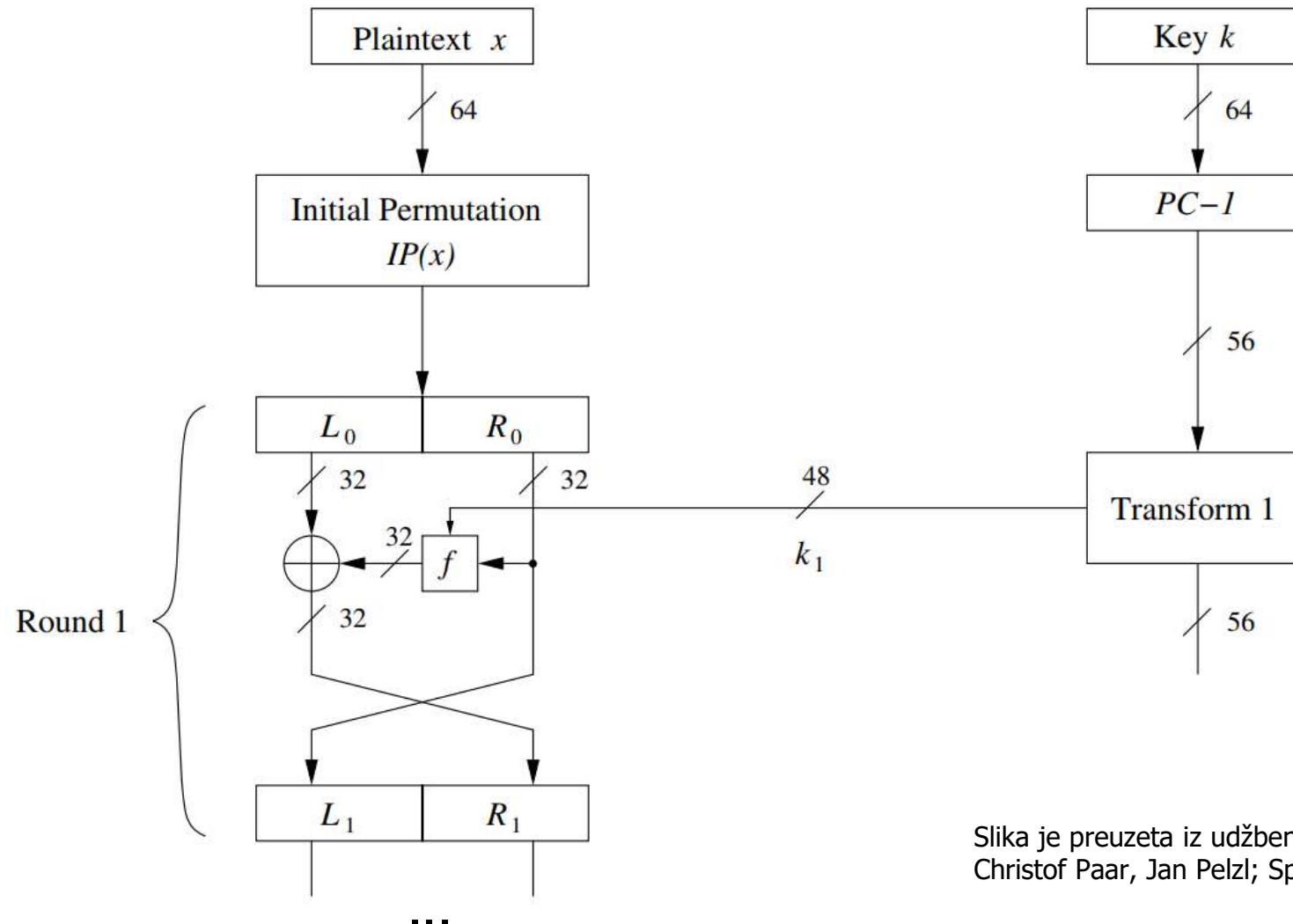
DES (Data Encryption Standard)

- veličina bloka: 64 bita
- veličina ključa: 56 bita
- struktura: Feistelova mreža
- razvijen u IBM-u, povijest:
 - Kasne **1960-te**: IBM razvija Lucifer
 - **1972**: US National Bureau of Standards (NBS) započinje proces standardizacije simetrične enkripcije
 - **1975, 1976**: National Security Agency (NSA) predlaže određene promjene NBS-u, IBM-u
 - **1977**: NBS objavljuje Data Encryption Standard (FIPS PUB 46)
 - **1990**: Diferencijalna kriptoanaliza DES-a (neuspješna)
 - **1994**: Linearna kriptoanaliza DES-a (donekle uspješna)
 - **1990-te**: DES Challenges – napadi grubom silom na DES
 - **2002**: NIST (preimenovani NBS) objavljuje novi standard (AES)

DES (Data Encryption Standard)

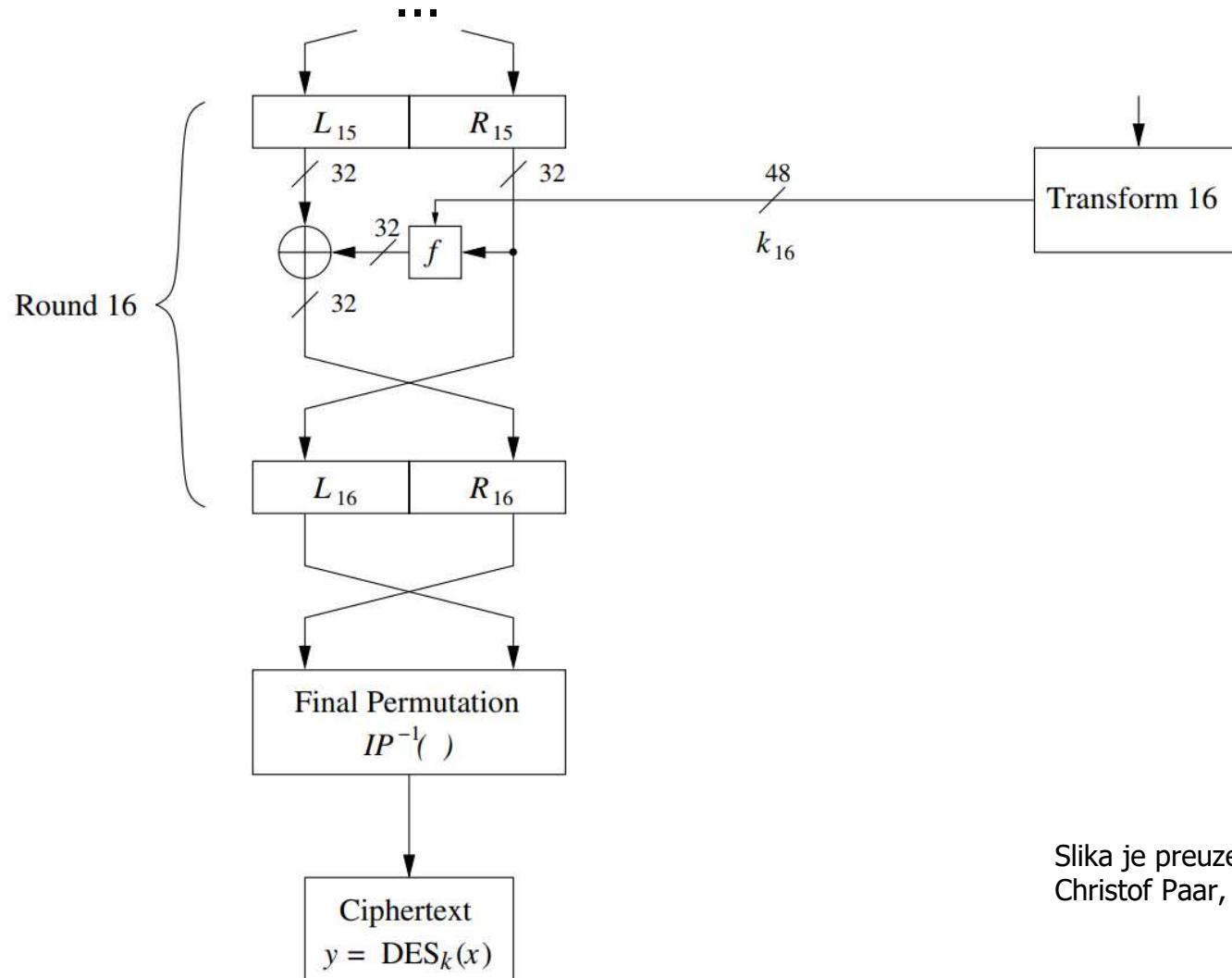
- nesiguran
 - 1998: "DES Challenge II" ostvareno računalom za \$250.000 koje za manje od 3 dana razbija DES poruku (nagrada je bila \$10.000)
- mala **veličina ključa** od 56 bita je najveći nedostatak koji se otklanja višestrukim kriptiranjem
 - **utrostručeni (*triple*) DES (3DES)** s ključem veličine
 - 112 (2x56) ili
 - 168 (3x56) bita
- unatoč svojoj *nesigurnosti* još uvijek se široko koristi

DES – Feistelova mreža



Slika je preuzeta iz udžbenika: ***Understanding Cryptography***,
Christof Paar, Jan Pelzl; Springer-Verlag Berlin Heidelberg, 2009.

DES – Feistelova mreža



Slika je preuzeta iz udžbenika: ***Understanding Cryptography***,
Christof Paar, Jan Pelzl; Springer-Verlag Berlin Heidelberg, 2009.

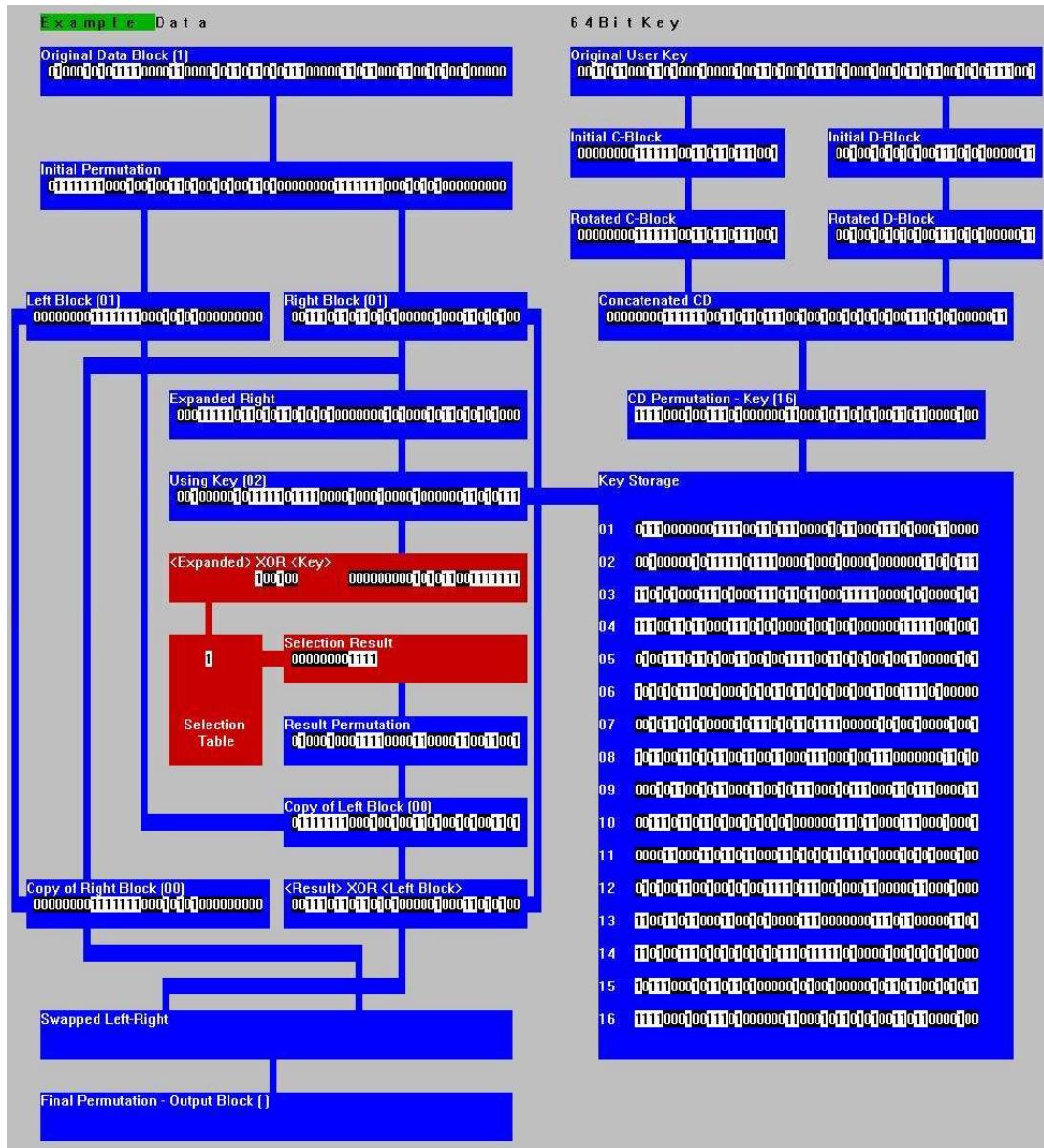
DES – postupak kriptiranja

- kriptiraju se blokove duljine 64 bita (8 bajtova)
- iz ključa K veličine 56 bita određuje se 16 podključeva K_i duljine 48 bita
- Postupak kriptiranja poruke M duljine 8 bajtova:
 - $L_0 \ R_0 = \text{IP} (M)$.
 - 16 koraka:
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i),$$
- $f(R_{i-1}, K_i)$ obavlja "preslagivanje" bitova u R_{i-1} ovisno o parametru K_i
- na kraju se obavlja inverzna permutacija od IP
$$C = \text{IP}^{-1} (R_{16} \ L_{16}).$$

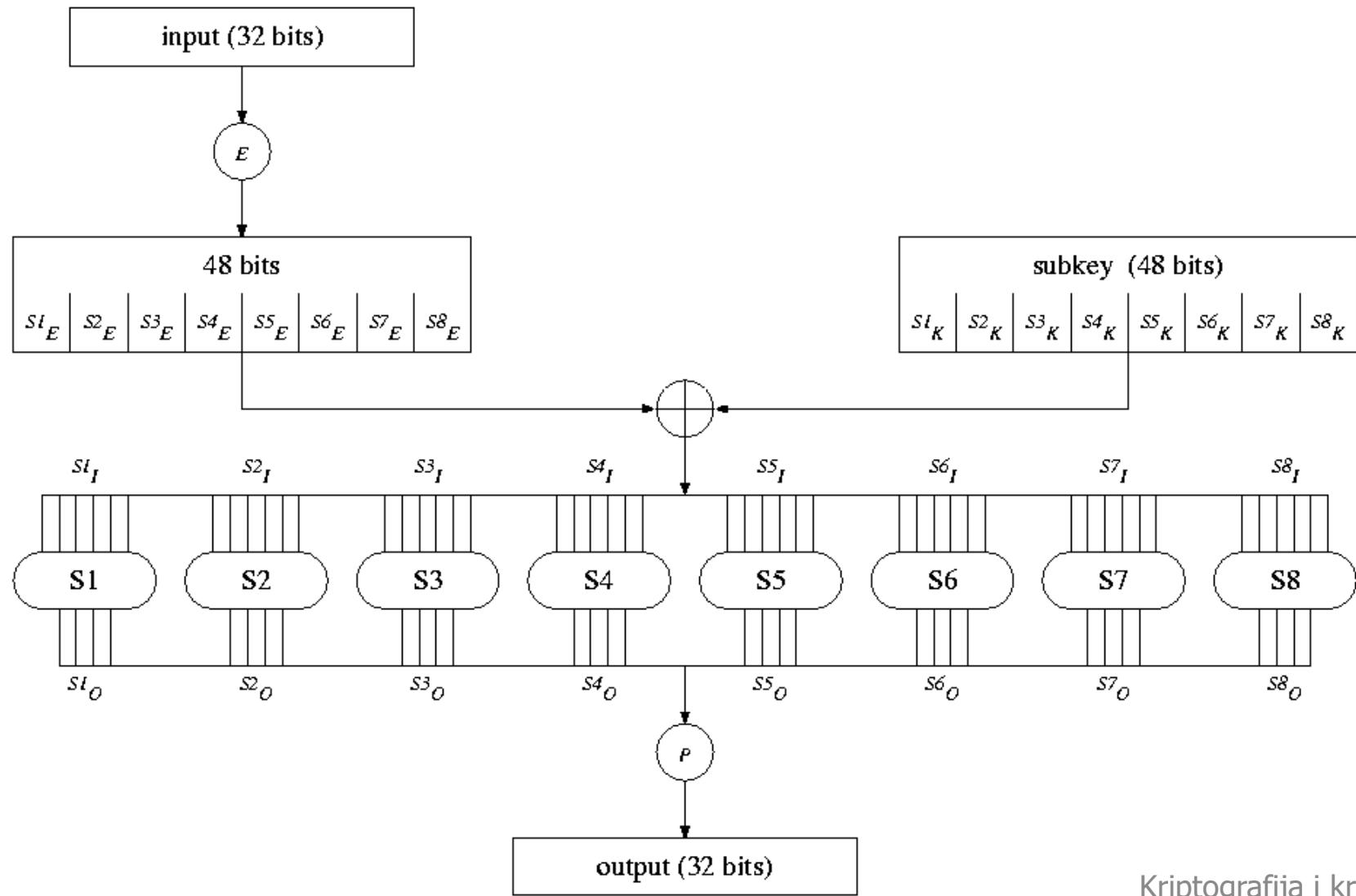
DES – detalji koji nisu jako bitni

- Permutacija IP nema utjecanja na sigurnost, nepoznato je zašto se koristi u DES-u.
- Ekspanzija ključa je jednostavna i nema velikog utjecaja na sigurnost.
 - Svaki podključ sadrži nekih 48 bitova ključa u nekom redoslijedu.
 - Svaki bit ključa se koristi u skoro svim rundama.

Simulacija kriptosustava DES



funkcija f



Funkcija f

- Funkcija E proširi ulaz tako da ponovi neke bitove dva puta.
 - Ne igra bitnu ulogu u sigurnosti.
- Supstitucijske tablice zamjenjuju 6-bitne blokove 4-bitnim blokovima.
 - Kritična uloga u sigurnosti, treba ih pažljivo odabrati.
- Funkcija P permutira bitove.
 - Bitna za difuziju.

Supstitucijske tablice

- ulaz u S tablicu je veličine 6 bita, a izlaz 4 bita
- supstitucijska tablica S1:

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

- prvi i zadnji bit svakog dijela ulaza predstavlja adresu retka
- srednja četiri bita određuju adresu stupca u tablici selekcije
- nakon primjene 8 supstitucijskih tablica od ulaznih 48 bita dobivamo 32 bita nakon supstitucije

Važne činjenice o supstitucijskim tablicama

- Kritične za sigurnost DES-a!
- Jedini nelinearni dio sustava.
- Kada bi supstitucijske tablice bile nasumično odabране, sustav bi bio potpuno nesiguran.

Dizajn supstitucijskih tablica za DES

- (S-1) Each S-box has six bits of input and four bits of output. (This was the largest size that we could accommodate and still fit all of DES onto a single chip in 1974 technology.)
- (S-2) No output bit of an S-box should be too close to a linear function of the input bits. (That is, if we select any output bit position and any subset of the six input bit positions, the fraction of inputs for which this output bit equals the XOR of these input bits should not be close to 0 or 1, but rather should be near 1/2.)
- (S-3) If we fix the leftmost and rightmost input bits of the S-box and vary the four middle bits, each possible 4-bit output is attained exactly once as the middle four input bits range over their 16 possibilities.
- (S-4) If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits. (That is, if $|\Delta I_{i,j}| = 1$, then $|\Delta O_{i,j}| \geq 2$, where $|x|$ is the number of 1-bits in the quantity x .)
- (S-5) If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits. (If $\Delta I_{i,j} = 001100$, then $|\Delta O_{i,j}| \geq 2$.)
- (S-6) If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same. (If $\Delta I_{i,j} = 11xy00$, where x and y are arbitrary bits, then $\Delta O_{i,j} \neq 0$.)
- (S-7) For any nonzero 6-bit difference between inputs, $\Delta I_{i,j}$, no more than eight of the 32 pairs of inputs exhibiting $\Delta I_{i,j}$ may result in the same output difference $\Delta O_{i,j}$.
- (S-8) Similar to (S-7), but with stronger restrictions in the case $\Delta O_{i,j} = 0$, for the case of three active S-boxes on round i . See the discussion below.

Izvor: D. Coppersmith, Data Encryption Standard (DES) and its strength against attacks, 1994

Svojstva Feistelove mreže

- Kako dekriptirati?
- Kriptiranje je invertibilno bez obzira na svojstva funkcije f .
- Isti sklop se može koristiti za kriptiranje i za dekriptiranje.
- Sigurnost ovisi o funkciji f , sve ostalo su XOR operacija i permutacije bitova.

DES: kriptiranje i dekriptiranje

generiraj_podključeve (ključ, K [16])

za svaki blok čini

p = perm_IP (blok [j])

L₁ = p [1:32] // lijevih 32 bita bloka

R₁ = p [33:64] // desnih 32 bita bloka

za i = 1 do 16 čini

L_{i+1} = R_i

R_{i+1} = L_i ⊕ f (R_i, K_i) // kriptiranje

R_{i+1} = L_i ⊕ f (R_i, K_{16-i+1}) // dekriptiranje

q [1:32] = R₁₆

q [33:64] = L₁₆

kriptirani_blok = perm_IP⁻¹ (q)

kraj

Zadatak: DES-bez-S

- Sustav DES-bez-S je identičan DES-u osim što nema S-tablice.
- Zadano je nekoliko stotina parova $M_i, C_i = DES\text{-bez-}S(M_i, K)$, odredite ključ K .

Utrostručeni DES, 3DES

$$3DES(M, K1, K2, K3) = DES(DES^{-1}(DES(M, K1), K2), K3)$$

$$3DES^{-1}(C, K1, K2, K3) = DES^{-1}(DES(DES^{-1}(C, K3), K2), K1)$$

- veličine ključeva:
 - varijanta ključa 1: tri nezavisna ključa $K1, K2$ i $K3 \Rightarrow 3 \times 56 = 168$ bitova
 - varijanta ključa 2: dva nezavisna ključa $K1 = K3$ i $K2 \Rightarrow 2 \times 56 = 112$ bitova
- nedozvoljene varijante ključa:
 - varijanta ključa 3: $K1 = K2 = K3$ (to je zapravo DES)
 - $K1 = K2$ (i to je zapravo DES jer se koristi samo jedan ključ $K3$)
 - $K2 = K3$ (i to je zapravo DES jer se koristi samo jedan ključ $K1$)

Zašto se ne koristi 2DES?

Zbog napada *susret u sredini* (eng. *Meet-in-the-middle attack*).

$$C = 2DES(M, K1, K2) = DES(DES(M, K1), K2)$$

Složenost napada grubom silom za poznati par (M,C)

2^{56} mogućnosti: $C' = DES(M, K1i), \quad i=1,2, \dots 2^{56}$

2^{56} mogućnosti: $M' = DES^{-1}(C, K2i), \quad i=1,2, \dots 2^{56}$

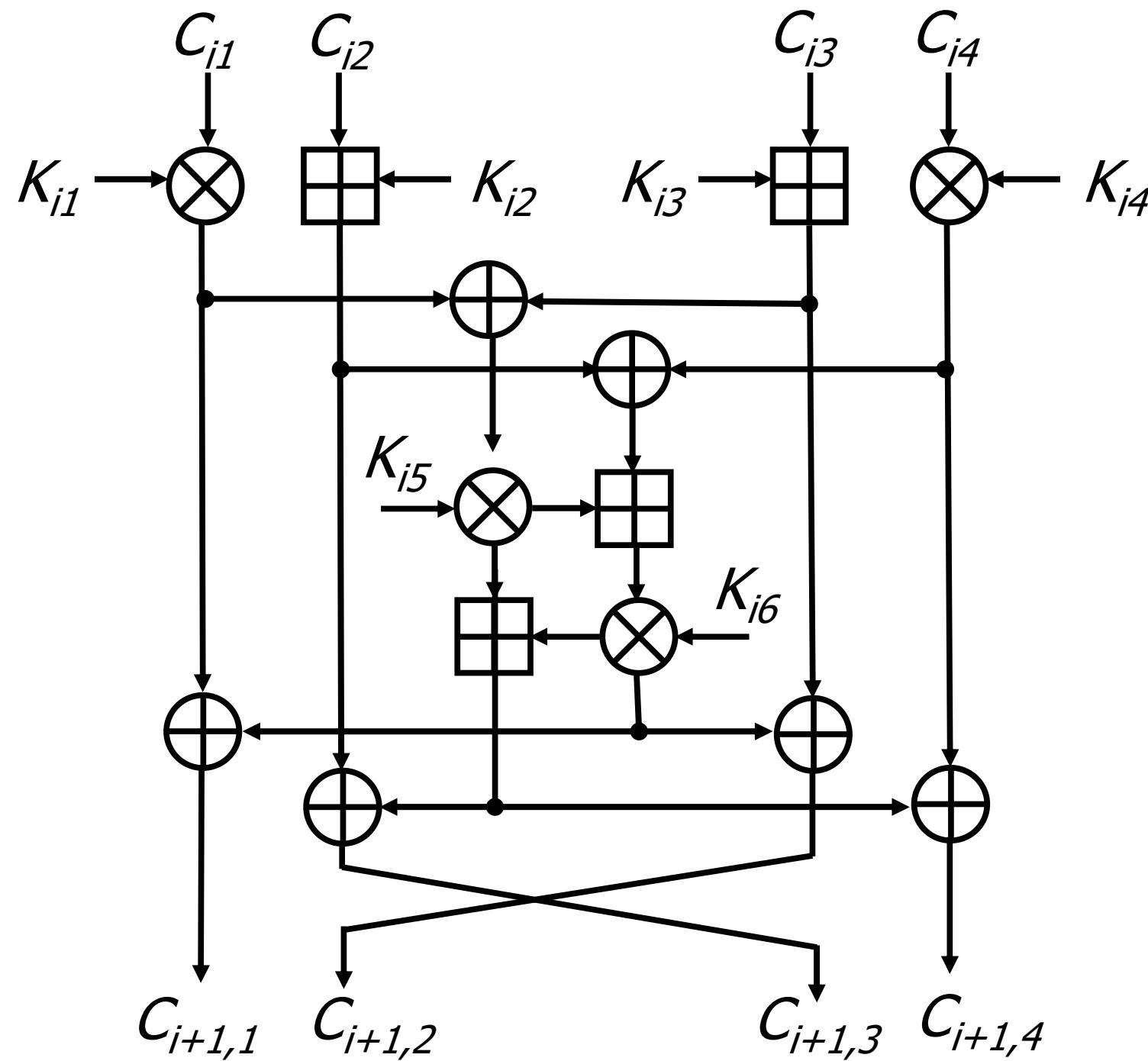
Ako je $M' = C'$ tada smo pronašli par ključeva $K1$ i $K2$

\Rightarrow složenost $2 \cdot 2^{56} = 2^{57}$, a ne 2^{112} !

IDEA (*International Data Encryption Algorithm*)

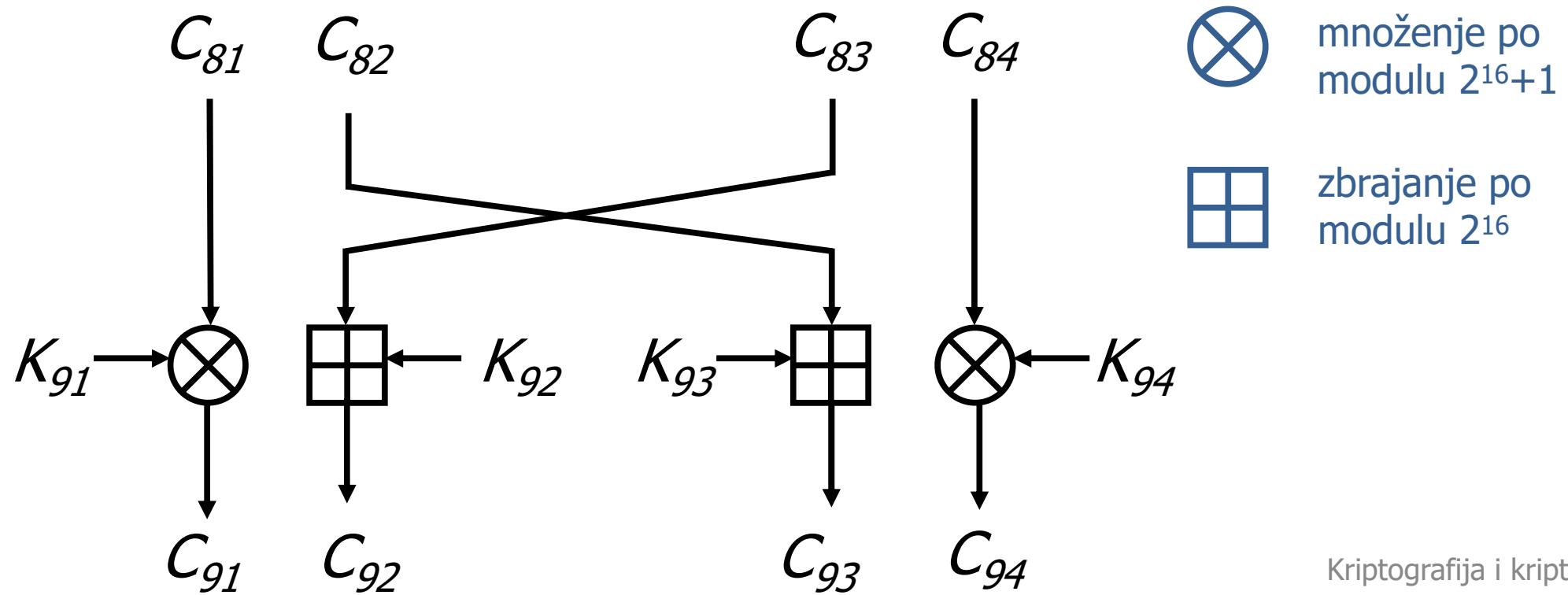
- dovršen 1992., siguran
- ključ je duljine 128 bita
- blokovi duljine 64 bita dijele se na 4 podbloka (16 b)
- postupak kriptiranja se provodi u 9 koraka
 - u svakom od prvih 8 koraka sudjeluju:
4 podbloka i 6 podključeva duljine 16 bita
 - u devetom koraku se koriste 4 podključa
- dakle, iz ključa K je potrebno generirati $8 \times 6 + 4 = 52$ podključeva

Prvih osam koraka algoritma IDEA

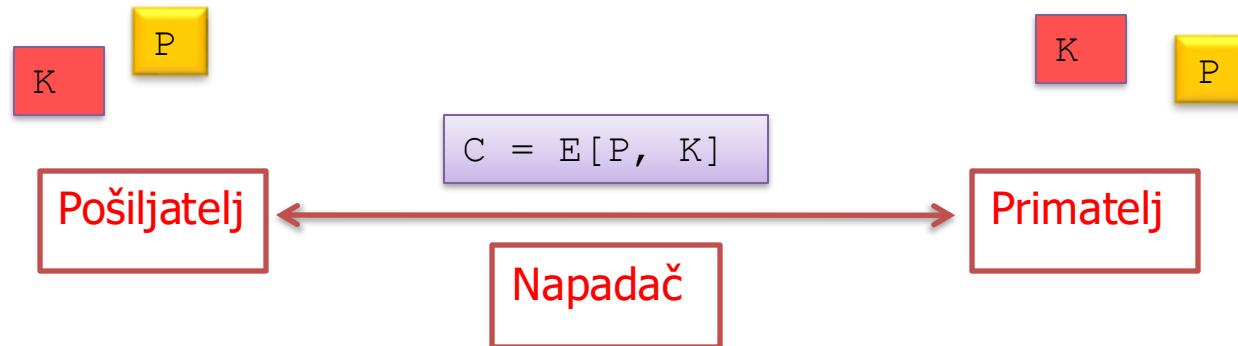


- \otimes množenje po modulu $2^{16}+1$
- \oplus zbrajanje po modulu 2^{16}
- \square XOR bit po bit

Deveti korak algoritma IDEA



Ponavljanje: Simetrična enkripcija

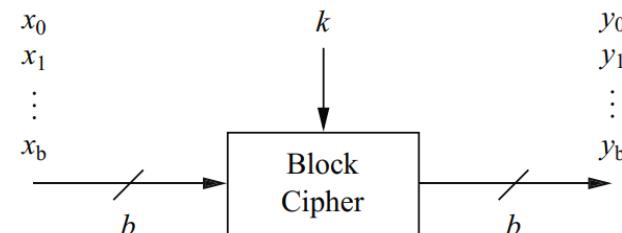
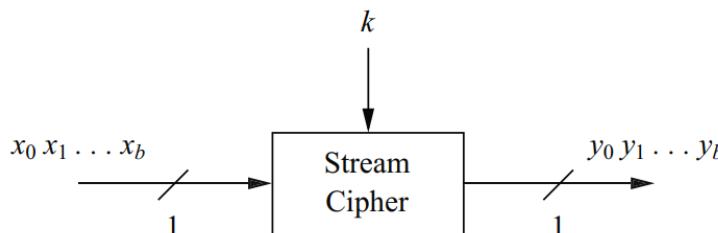


Ponavljanje: Simetrična enkripcija – definicija

- Neka su K , M i C konačni skupovi – prostor ključeva, prostor jasnih tekstova i prostor skrivenih tekstova.
- Simetrična enkripcija je par algoritama E i D ($E: M \times K \rightarrow C$, $D: C \times K \rightarrow M$) gdje za svaki $k \in K$ i $m \in M$ vrijedi
 - $D(E(m, k), k) = m.$

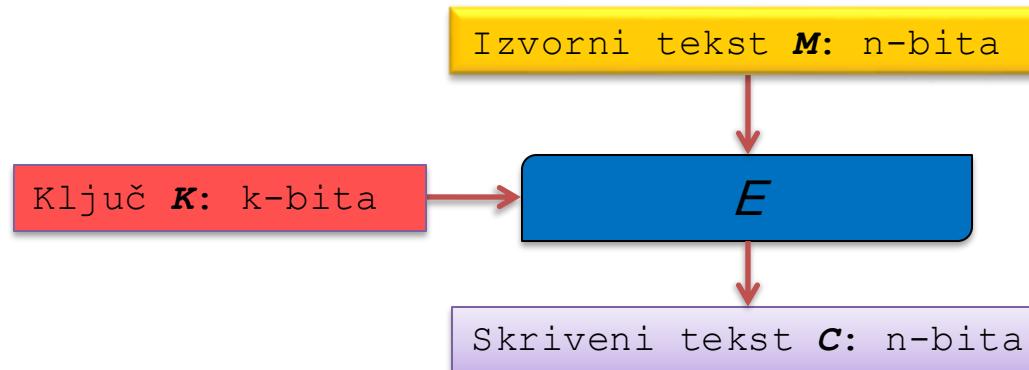
Ponavljanje: vrste simetrične enkripcije

- Protočna enkripcija (*eng. stream cipher*)
 - Kriptira se jedan po jedan bit.
- Sustavi kriptiranja bloka (*eng. block cipher*)
 - Kriptiraju se blokovi fiksne duljine.

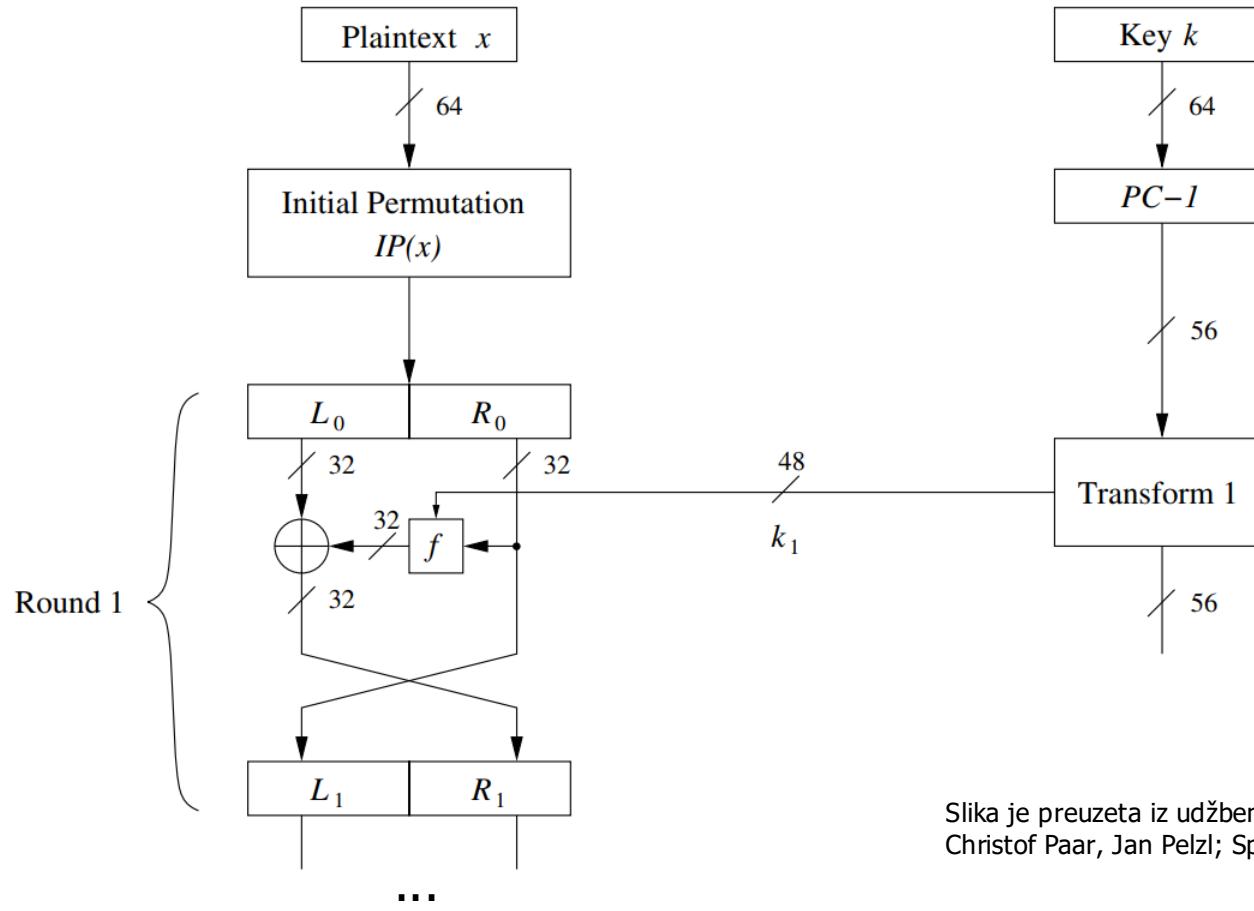


Ponavljanje: sustav kriptiranja bloka

- $M = C = \{0, 1\}^n$
- $K = \{0, 1\}^k$
- E i D su deterministički algoritmi.

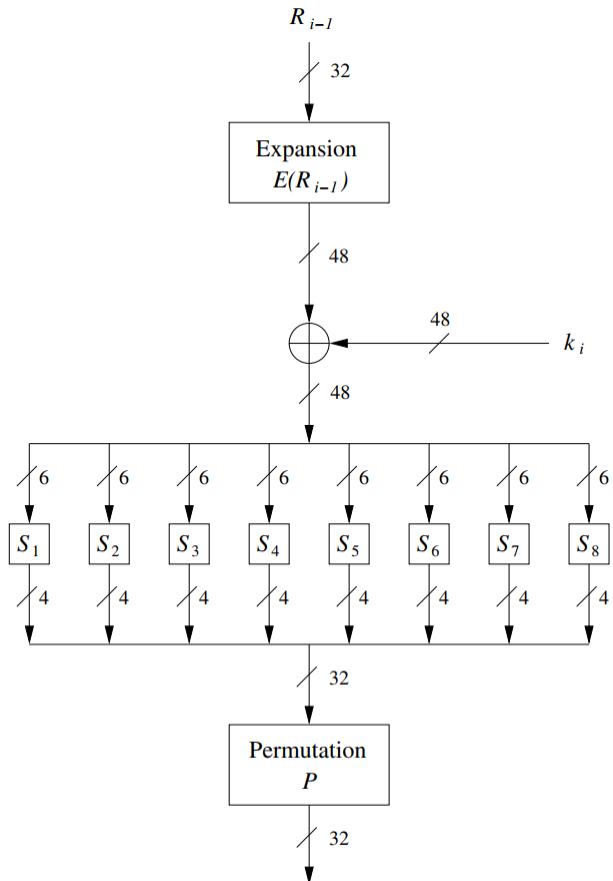


Ponavljanje: DES – Feistelova mreža



Slika je preuzeta iz udžbenika: ***Understanding Cryptography***,
Christof Paar, Jan Pelzl; Springer-Verlag Berlin Heidelberg, 2009.

Ponavljanje: DES – Funkcija f



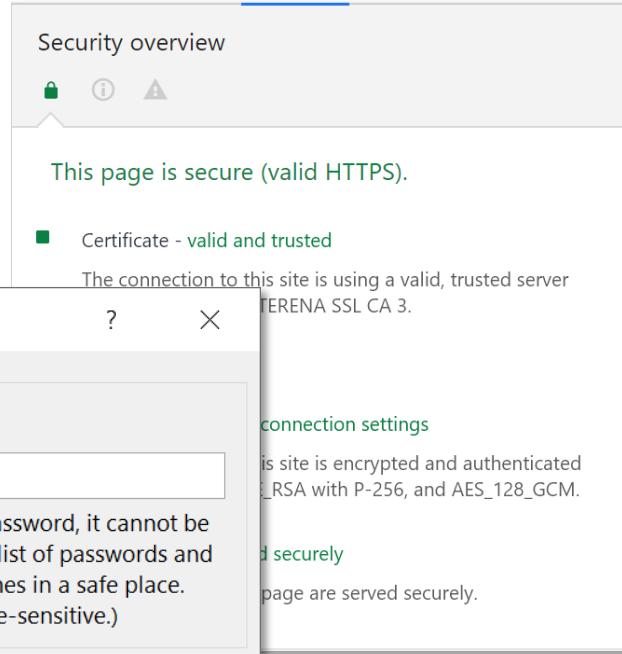
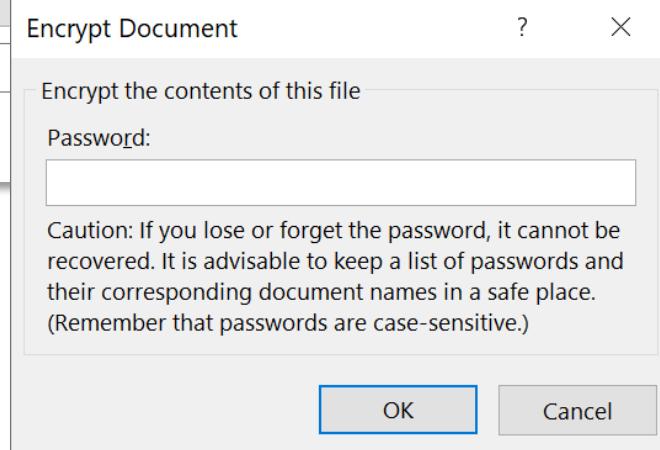
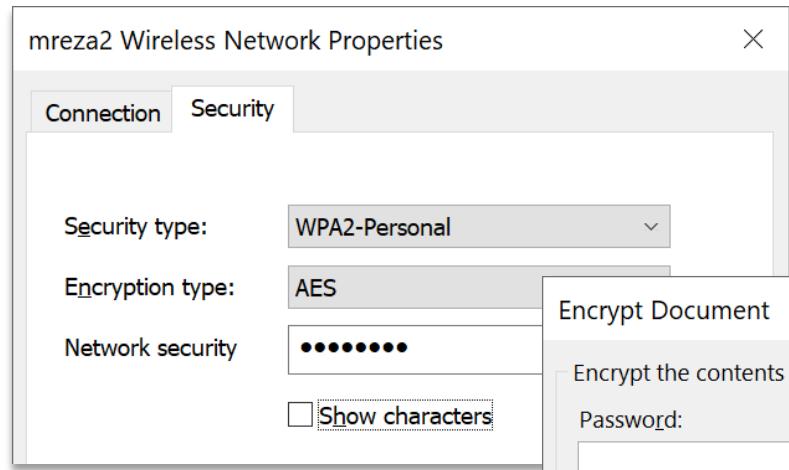
Slika je preuzeta iz udžbenika: ***Understanding Cryptography***,
Christof Paar, Jan Pelzl; Springer-Verlag Berlin Heidelberg, 2009.

Zadatak: DES-bez-S

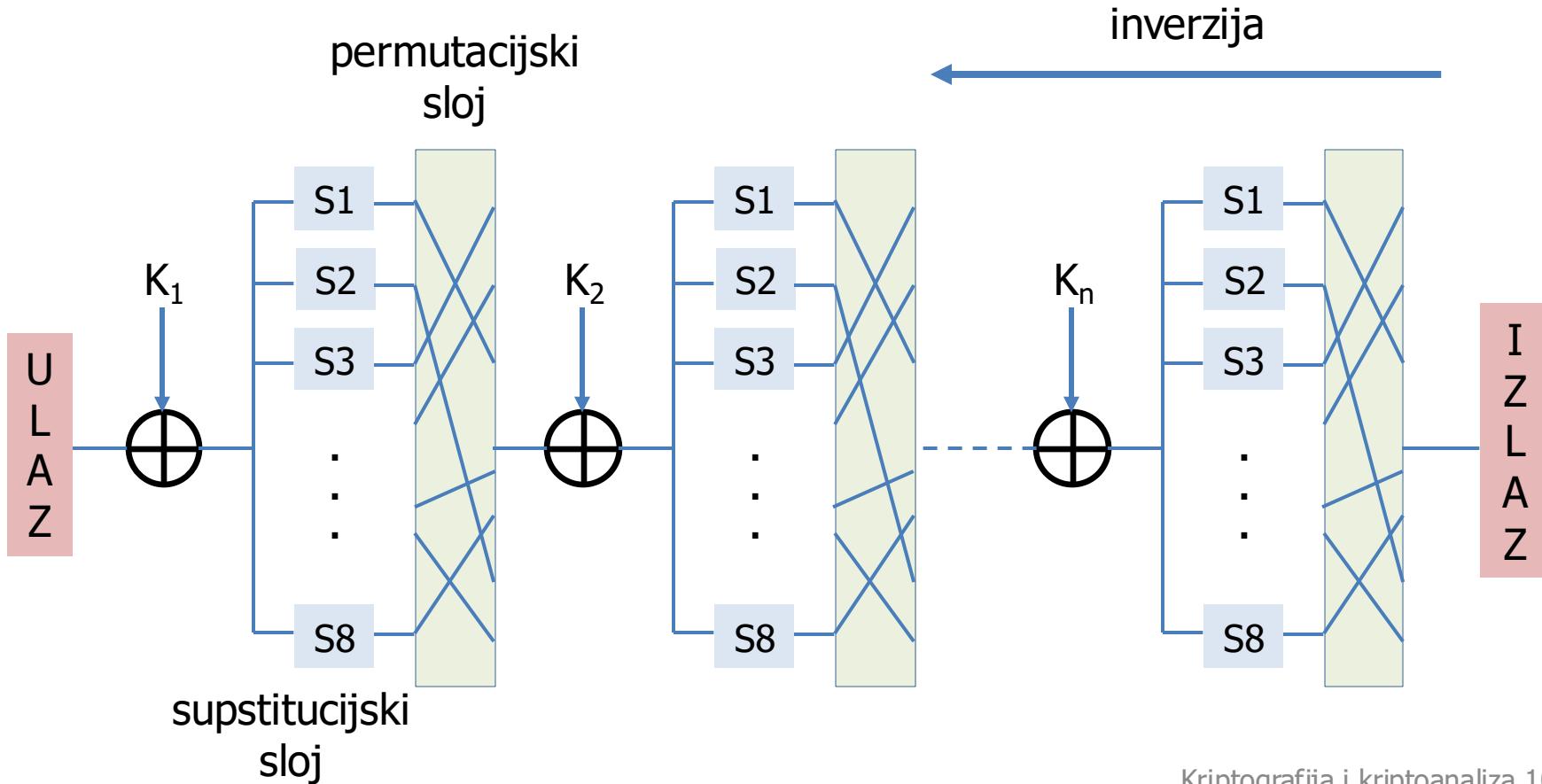
- Sustav DES-bez-S je identičan DES-u osim što nema S-tablice.
- Zadano je nekoliko stotina parova $M_i, C_i = DES\text{-bez-S}(M_i, K)$, pronađite način da dekriptirate nove poruke kriptirane ključem K .

Napredni kriptosustav (AES)

- Natječaj za novi standard je raspisao NIST 1997. godine
- Pobjednik sustav *Rijndael* (autori Vincent Rijmen i Joan Daemen)
- Jednostavna struktura!
- Parametri:
 - Veličina bloka: 128 bitova
 - Veličine ključa: 128, 192 ili 256 bitova



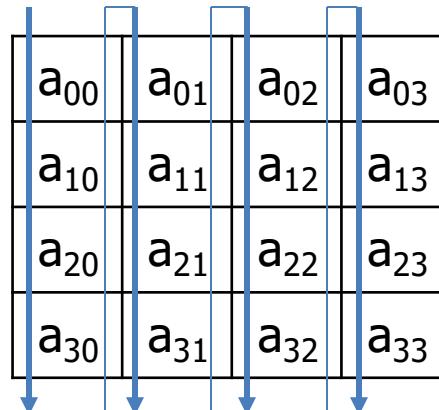
AES: supstitucijsko-permutacijska mreža



Blok

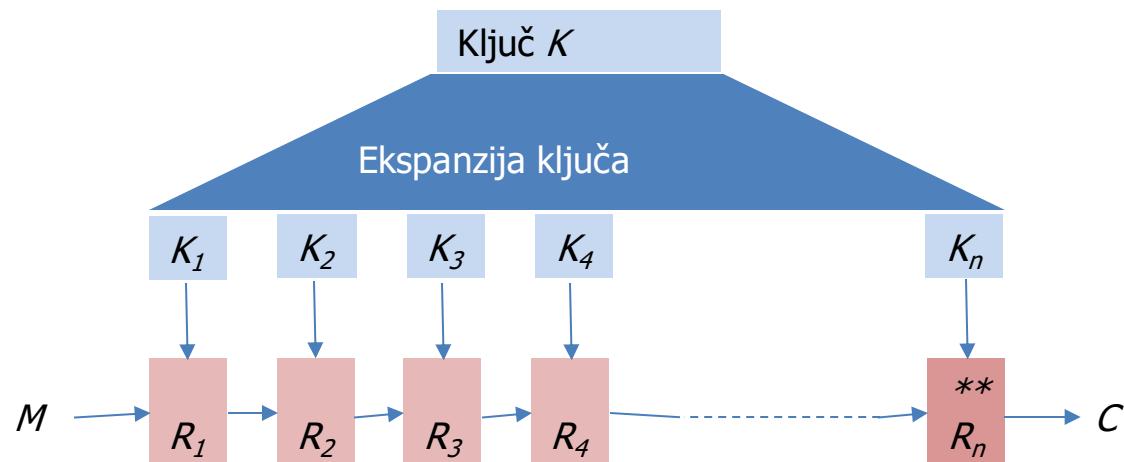
- veličina bloka: 128 bita (AES)
 - izvorni algoritam Rijndael dopušta veličine bloka od 128, 192 ili 256 bita nezavisno od veličine ključa
- pravokutni niz bajtova u četiri retka i četiri stupca $Nb = 4$
- na sličan način se tretira i ključ koji je također smješten u pravokutni niz bajtova u četiri retka, a broj stupaca ovisi o veličini ključa: $Nk = 4, 6$, ili 8
- broj koraka Nr :

Nr	$Nb = 4$
$Nk = 4$	10
$Nk = 6$	12
$Nk = 8$	14

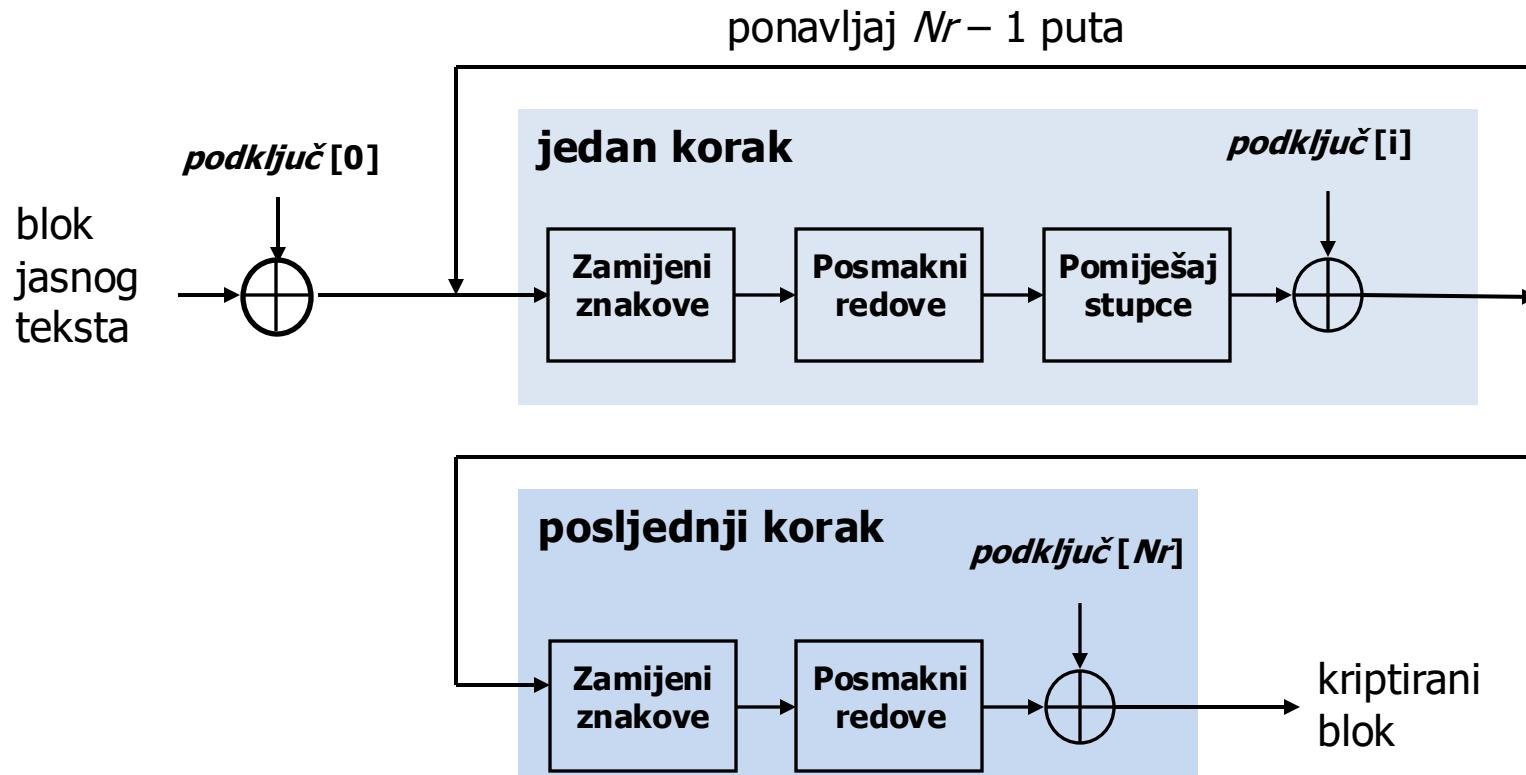


- redoslijed punjenja bloka
 - po stupcima

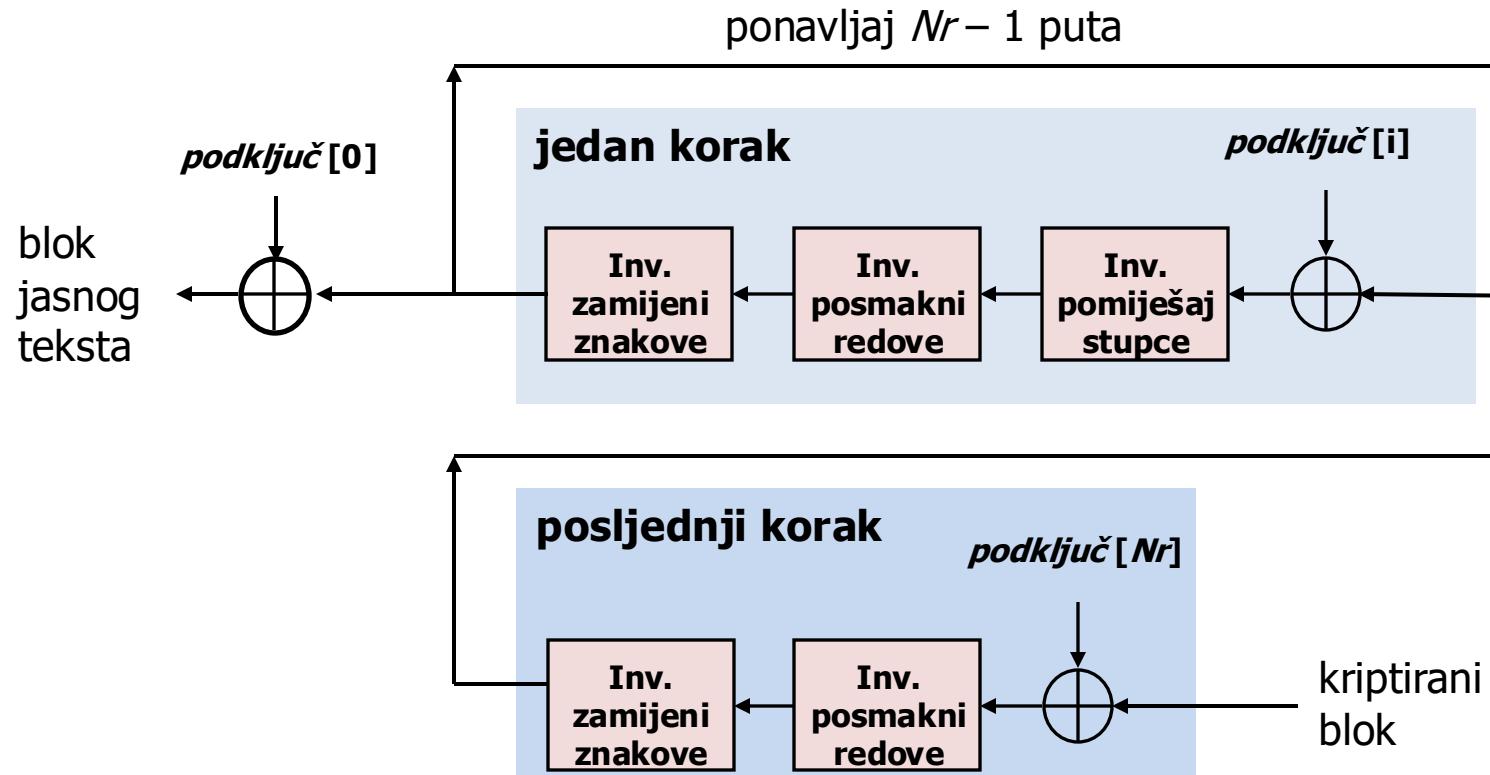
AES – runde



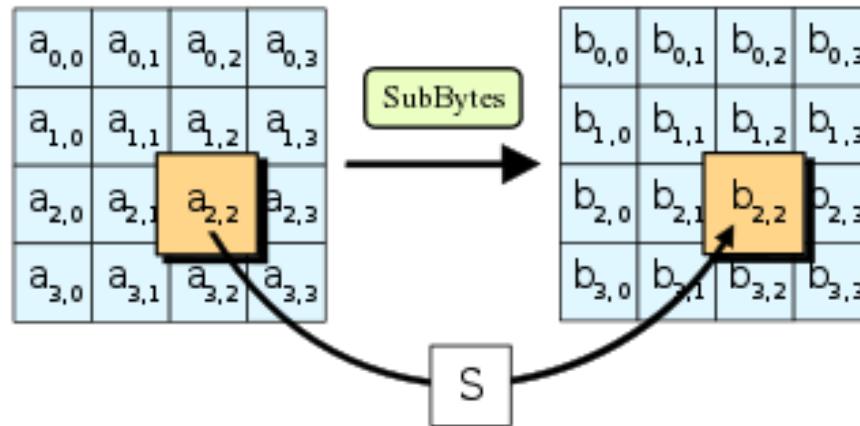
AES – postupak kriptiranja



AES – postupak dekriptiranja

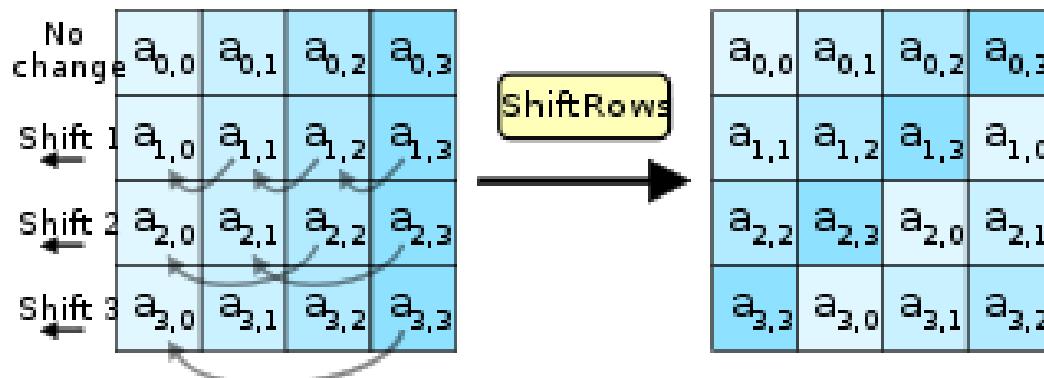


AES128 – Zamijeni znakove



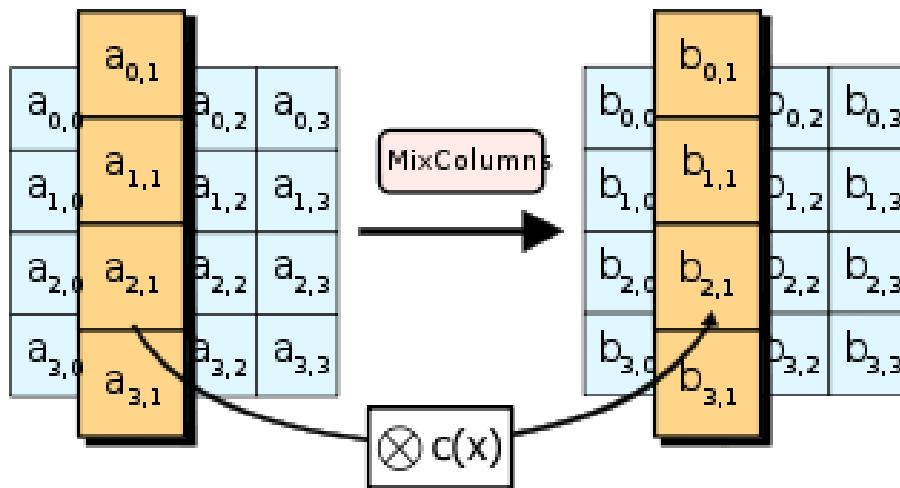
Izvor: [wikipedia.org](https://en.wikipedia.org)

AES128 – Posmakni redove



Izvor: [wikipedia.org](https://en.wikipedia.org)

AES128 – Pomiješaj stupce



Izvor: wikipedia.org

Funkcije koje koristi algoritam AES

- *zamijeni znakove*

$$\text{znak} = \text{Sbox}[\text{znak}]$$

- *dodaj podključ*

$$\text{blok} = \text{blok} \oplus \text{podključ}[i]$$

- *posmakni redove*

- rotira (kružno posmiče) znakove uljevo i to u drugom, trećem i četvrtom redu bloka (C_1 , C_2 i C_3) za unaprijed poznati broj mesta koji ovisi o N_b
- prvi red (C_0) se ne posmiče

N_b	C_1	C_2	C_3
4	1	2	3
6	1	2	3
8	1	3	4

Funkcije koje koristi algoritam AES

- *pomiješaj stupce*

- množi se stupac po stupac bloka (tako da se svaki stupac promatra kao četveročlani polinom) s fiksnim polinomom

$$a(x) = 03_H x^3 + 01_H x^2 + 01_H x + 02_H \text{ modulo } x^4 + 1$$

- odnosno, za svaki stupac bloka računa se stupac novog stanja:

$$\begin{bmatrix} s_{0i}' \\ s_{1i}' \\ s_{2i}' \\ s_{3i}' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0i} \\ s_{1i} \\ s_{2i} \\ s_{3i} \end{bmatrix}$$

AES – detalji koji nisu jako bitni

- Supstitucijske tablice imaju jednostavan matematični opis: inverz i afina funkcija u $GF(2^8)$.
- Ekspanzija ključa nešto složenija nego kod DES-a: XOR i supstitucijske tablice.
- Dizajn omogućuje vrlo efikasne softverske i hardverske implementacije.

Podsjetnik: Shannonova načela

- Difuzija
 - svaki bit jasnog teksta kao i svaki bit tajnog ključa treba utjecati na mnogo bitova kriptiranog teksta
 - promjena samo jednog bita jasnog teksta mora uzrokovati promjenu (statistički) polovicu bitova kriptiranog teksta
 - ostvaruje se primjerice permutacijom i u više koraka algoritma
- Konfuzija
 - međuzavisnost kriptiranog i jasnog teksta je previše složena da bi se mogla iskoristiti za razbijanje kriptosustava
 - svaki bit kriptiranog teksta treba ovisiti o više bitova ključa ali tako da se pritom prikrije veza između njih
 - ostvaruje se primjerice supstitucijom, tj. supstitucijskim tablicama

Zašto ovakav dizajn?

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

The linear mixing layer: guarantees high diffusion over multiple rounds.

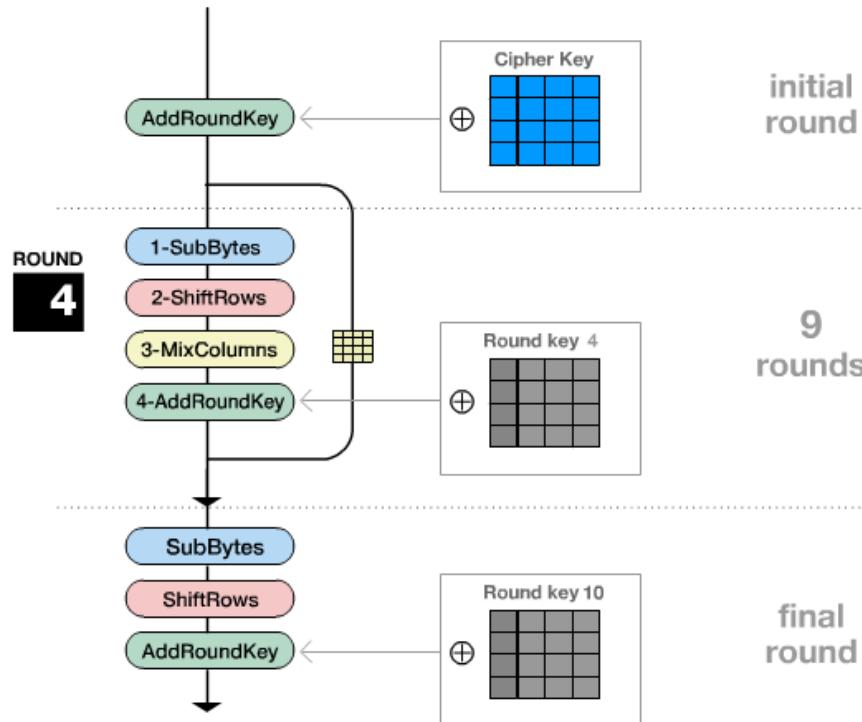
The non-linear layer: parallel application of S-boxes that have optimum worst-case nonlinearity properties.

The key addition layer: A simple EXOR of the Round Key to the intermediate State.

Izvor: AES Proposal: Rijndael
Joan Daemen, Vincent Rijmen, 2003.

Simulacija AES-a

Encryption process



[https://www.youtube.com/
watch?v=mlzxpkdXP58](https://www.youtube.com/watch?v=mlzxpkdXP58)

Programsko ostvarenje algoritma AES

- NE preporuča se vlastita programska implementacija zbog mogućih i vrlo vjerojatnih propusta
- koristiti raspoloživa i provjerena programska ostvarenja poput:
 - Openssl:
 - https://github.com/openssl/openssl/blob/master/crypto/aes/aes_x86core.c

Sklopovska potpora algoritmu AES

- Intel (slično i AMD)
 - aesenc, aesenclast: jedna runda AES-a
 - 128-bitni registri:
 - xmm1=state, xmm2=ključ za rundu
 - aesenc xmm1, xmm2 ; rezultat u xmm1
 - aeskeygenassist: stvaranje podključeva
 - 5 procesorskih ciklusa po bajtu, brzina se mjeri u GB/s

Primjer *uspješnog* napada na AES

- reducirani AES-128 na 8 rundi sa složenošću $2^{124.9}$
- potpuni AES-128 sa složenošću $2^{126.1}$
- potpuni AES-192 sa složenošću $2^{189.7}$
- potpuni AES-256 sa složenošću $2^{254.4}$

A. Bogdanov (KU Leuven), D. Khovratovich (MS Research Redmond), C. Rechberger (France Telecom), Biclique Cryptanalysis of the Full AES, ASIACRYPT, 2011.

Zadatak

- Razmatrajte 1AES – AES sa samo jednom rundom.
 - Pokažite da je nesiguran tako da opišete postupak koji će na temelju M i $C=1\text{AES}(M, K)$ odrediti ključ K .

Zadatak

- Za one koji žele više: Razmatrajte AES bez jedne od operacija i pokažite da je nesiguran.
 - Ako je dostupno puno parova M_i, C_i onda je moguće dekriptirati bilo koju poruku.

Ponavljanje? Grupe

Grupe. Grupa je matematička struktura koja se sastoji od nepraznog skupa G i binarne operacije $\circ : G \times G \rightarrow G$. To znači da je za svaka dva elementa $x, y \in G$ definiran njihov umnožak $x \circ y \in G$. Pri tome zahtjevamo da vrijede sljedeća svojstva

1) **Asocijativnost.** Za sve $x, y, z \in G$ vrijedi

$$(x \circ y) \circ z = x \circ (y \circ z).$$

2) **Postojanje neutralnog elementa.** Postoji element $e \in G$ takav da za svaki $x \in G$ vrijedi

$$e \circ x = x \circ e = x.$$

3) **Postojanje inverznog elementa.** Za svaki $x \in G$ postoji element $x^{-1} \in G$ takav da je

$$x \circ x^{-1} = x^{-1} \circ x = e.$$

Ako je k tome za svaka dva elementa $x, y \in G$ ispunjeno $x \circ y = y \circ x$, onda za G kažemo da je **komutativna** ili **Abelova** grupa.

Izvor: N. Elezović, Linearna algebra 1

Primjeri grupa

- $(\mathbb{Z}, +)$ je grupa
- $(\mathbb{Q} \setminus \{0\}, *)$ je grupa
- $(\mathbb{Z}_N, +)$ je grupa
- $(\mathbb{Z}_N^*, *)$ je grupa
- ...
- $(\mathbb{N}, +)$ nije grupa
- $(\mathbb{Q}, *)$ nije grupa
- $(\mathbb{Z}_N, *)$ nije grupa ako je N složen.
- ...

Ponavljanje? Polja

Polje. Sljedeća važna matematička struktura jest polje. Polje čini neprazni skup X na kojem su definirane dvije operacije i koje zadovoljavaju svojstva koja ćemo navesti u nastavku. Operacije ćemo označiti s $+$ i \cdot iako to ne moraju biti klasične operacije zbrajanja i množenja. Zahtijevamo da bude ispunjeno sljedeće:

- 1) $(X, +)$ je (aditivna) Abelova grupa,
- 2) (X^*, \cdot) je (multiplikativna) Abelova grupa, pri čemu je $X^* = X \setminus \{0\}$,
- 3) vrijede zakoni distribucije, za sve $x, y, z \in X$

$$(x + y) \cdot z = x \cdot z + y \cdot z$$

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

Izvor: N. Elezović, Linearna algebra 1

Primjeri polja

- $(\mathbb{Q}, +, *)$ je polje
- $(\mathbb{R}, +, *)$ je polje
- $(\mathbb{Z}_p, +, *)$ je polje ako je p prost broj
- ...
- $(\mathbb{Z}, +, *)$ nije polje
- $(\mathbb{Z}_n, +, *)$ nije polje ako je n složen
- ...

Zadatak: DES-bez-S

- Sustav DES-bez-S je identičan DES-u osim što nema S-tablice.
- Zadano je nekoliko stotina parova $M_i, C_i = DES\text{-bez-}S(M_i, K)$, odredite ključ K .

Konačno polje $GF(2^8)$

- elementi polja su polinomi oblika:

$$a_7x^7 + a_6x^6 + \dots + a_1x + a_0, \quad a_i \in \{0, 1\}$$

- svaki bajt $a_7a_6a_5a_4a_3a_2a_1a_0$ (niz od 8 bitova) je predstavljen odgovarajućim polinomom

- zbrajanje* - isključivo ILI
- množenje* - binarno množenje polinoma modulo fiksni ireducibilni polinom

$$g(x) = x^8 + x^4 + x^3 + x + 1 \equiv 11B_H$$

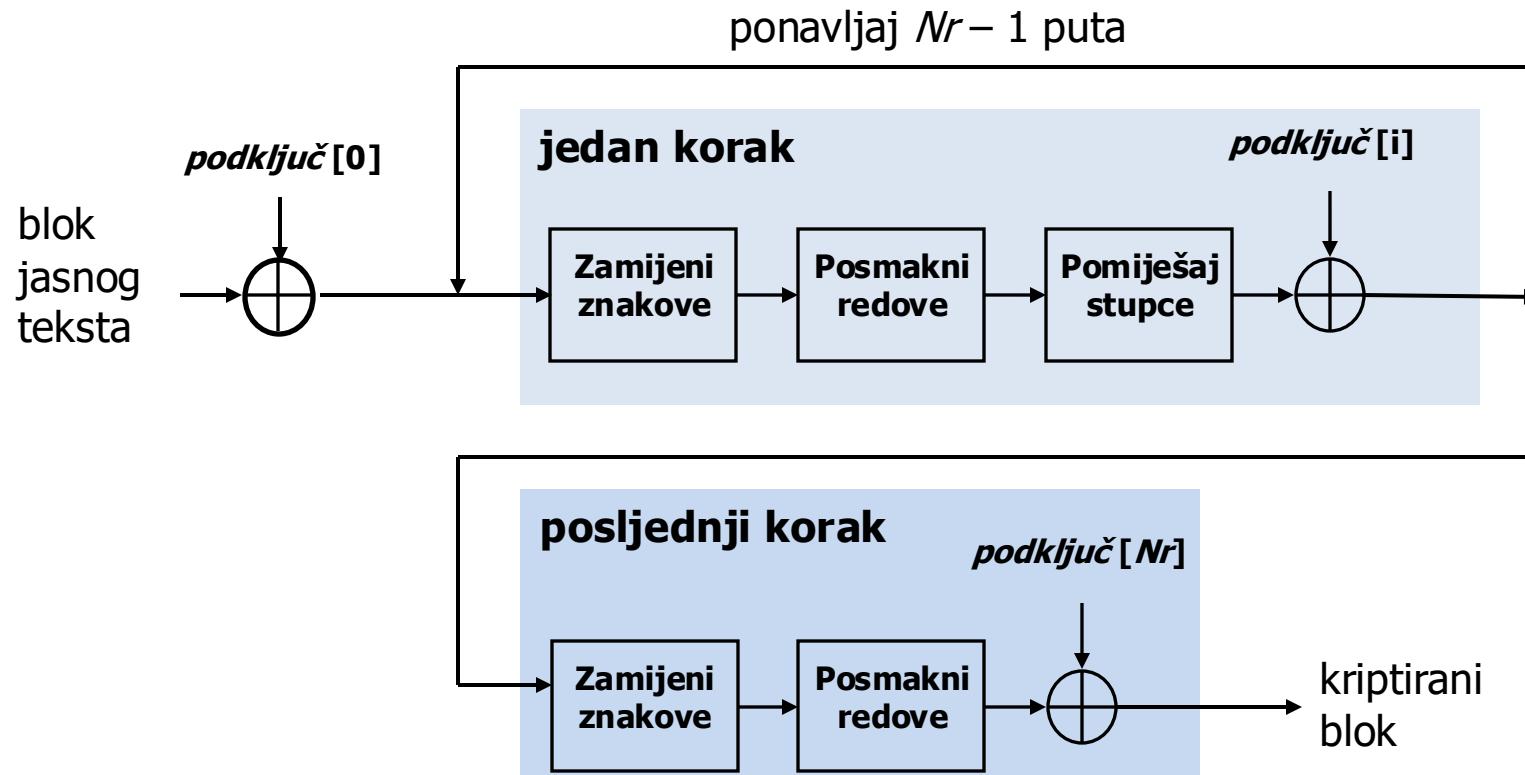


Kriptografija i kriptoanaliza

doc. dr. sc. Ante Đerek i prof. dr. sc. Marin Golub

listopad 2023.

Ponavljanje: AES – postupak kriptiranja



Ponavljanje: konačno polje $GF(2^8)$

- elementi polja su polinomi oblika:

$$a_7x^7 + a_6x^6 + \dots + a_1x + a_0, \quad a_i \in \{0, 1\}$$

- svaki bajt $a_7a_6a_5a_4a_3a_2a_1a_0$ (niz od 8 bitova) je predstavljen odgovarajućim polinomom

- zbrajanje* - isključivo ILI
- množenje* - binarno množenje polinoma modulo fiksni ireducibilni polinom

$$g(x) = x^8 + x^4 + x^3 + x + 1 \equiv 11B_H$$

Ponavljanje: Funkcije koje koristi algoritam AES

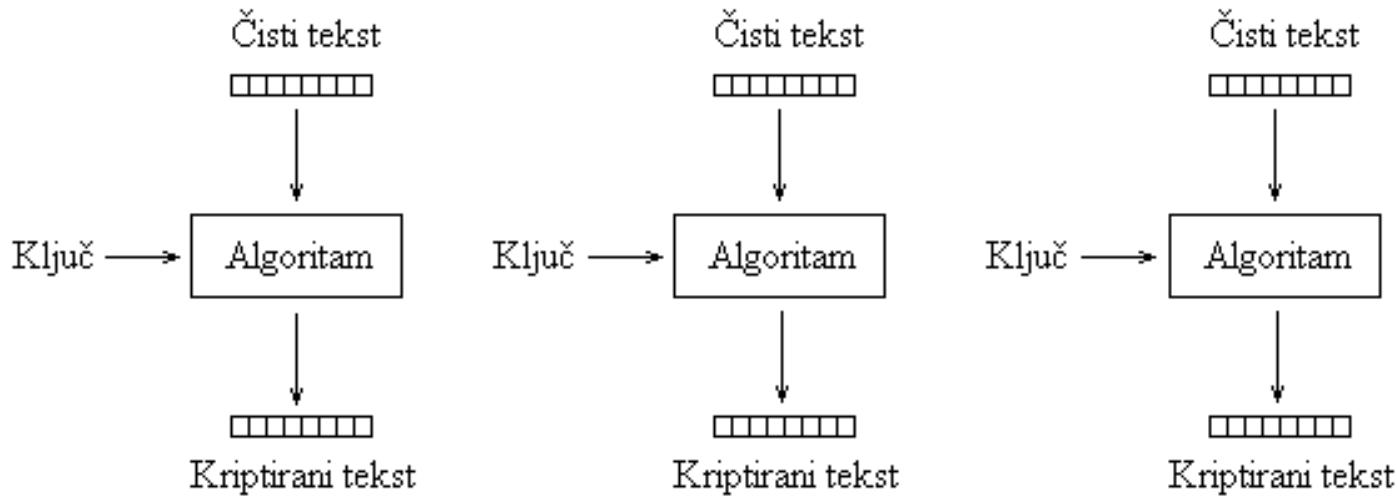
- *pomiješaj stupce*
 - Za svaki stupac bloka računa se stupac novog stanja:

$$\begin{bmatrix} s_{0i}' \\ s_{1i}' \\ s_{2i}' \\ s_{3i}' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0i} \\ s_{1i} \\ s_{2i} \\ s_{3i} \end{bmatrix}$$

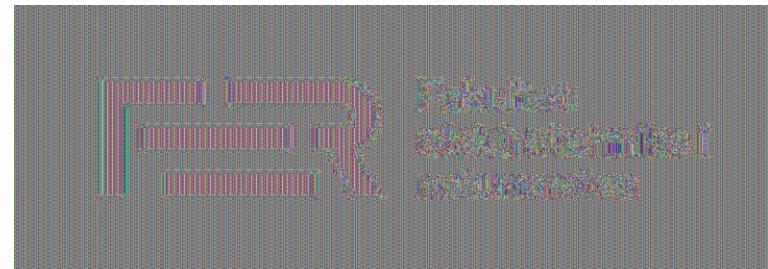
Načini kriptiranja

- Želimo kriptirati poruku proizvoljne duljine!
- Želimo jača sigurnosna svojstva koja nam ne može pružiti determinističko kriptiranje!

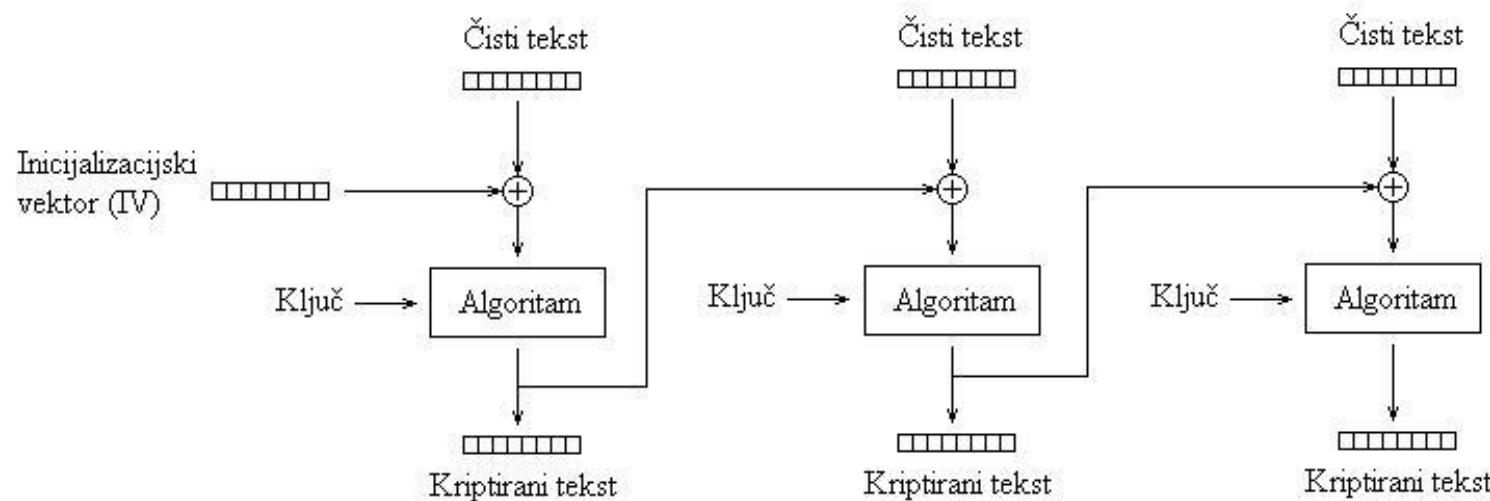
Načini kriptiranja: ECB - Electronic Codebook



Fakultet
elektrotehnike i
računarstva



Način kriptiranja: Cipher Block Chaining (CBC)

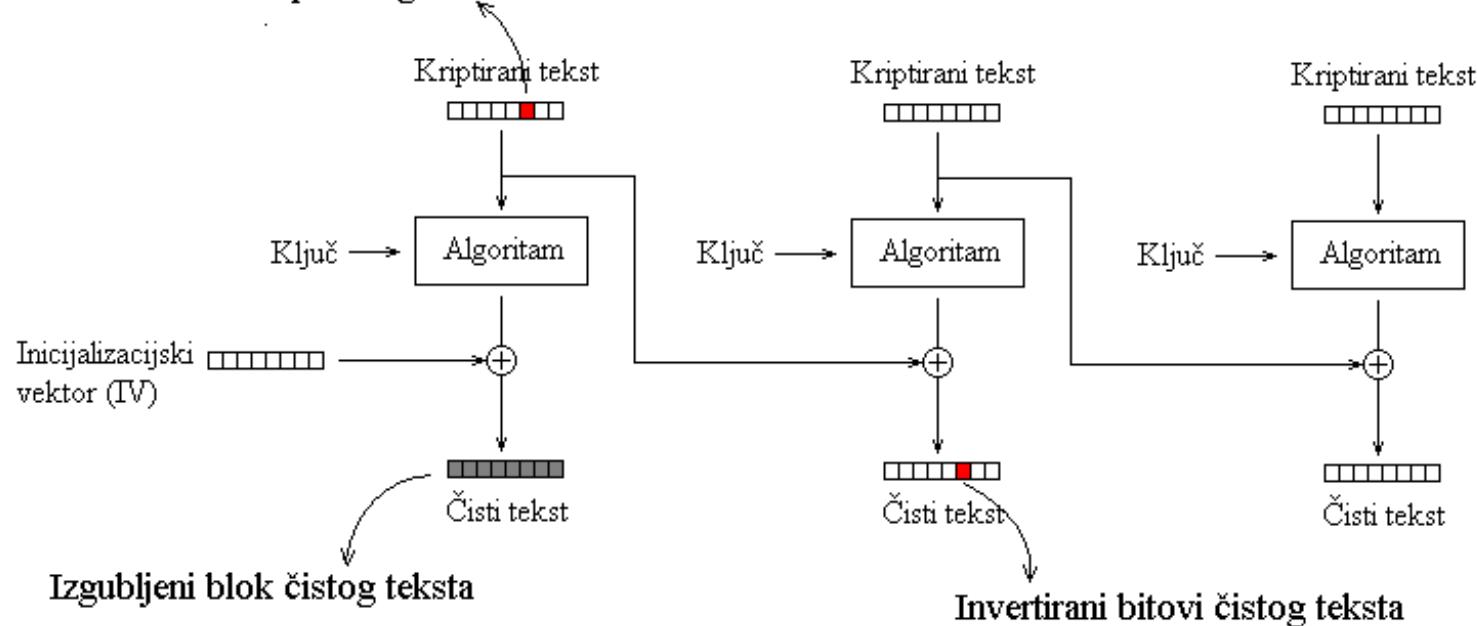


Cipher Block Chaining (CBC)

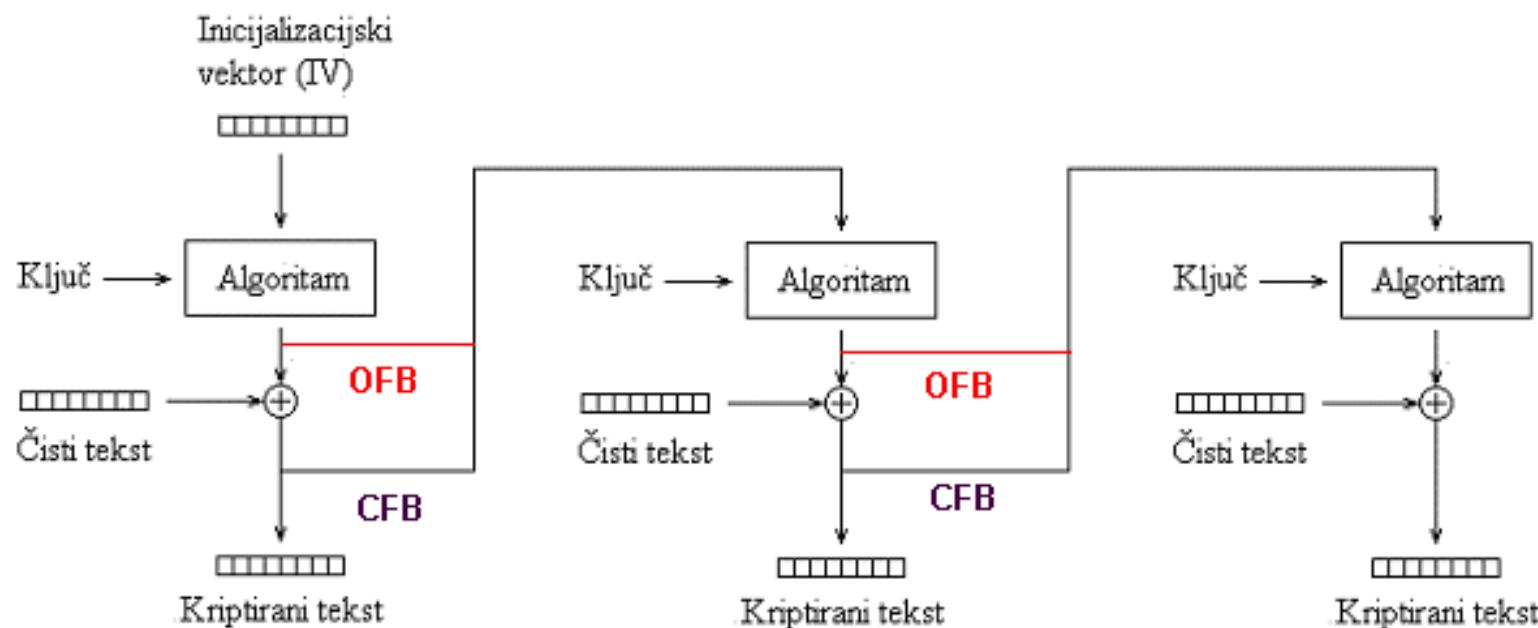
- Inicijalizacijski vektor IV se mora izabrati nasumično.
- Inicijalizacijski vektor IV se šalje zajedno s skrivenim tekstom (odnosno on je dio skrivenog teksta).
 - Posljedica: IV ne mora (i ne može) biti tajan.
- Potrebno je nadopuniti poruku tako da je duljina višekratnik veličine bloka.
- Kriptiranje: blok kriptiranog teksta ovisi o svim prethodnim blokovima jasnog teksta.
- Dekriptiranje: blok jasnog teksta ovisi o dva susjedna bloka kriptiranog teksta.
- Rezultat je sigurna enkripcija pod razumnim pretpostavkama.

Način kriptiranja: *Cipher Block Chaining (CBC)*

Invertirani bitovi kriptiranog teksta



Načini kriptiranja Cipher Feedback (CFB) i Output Feedback (OFB)



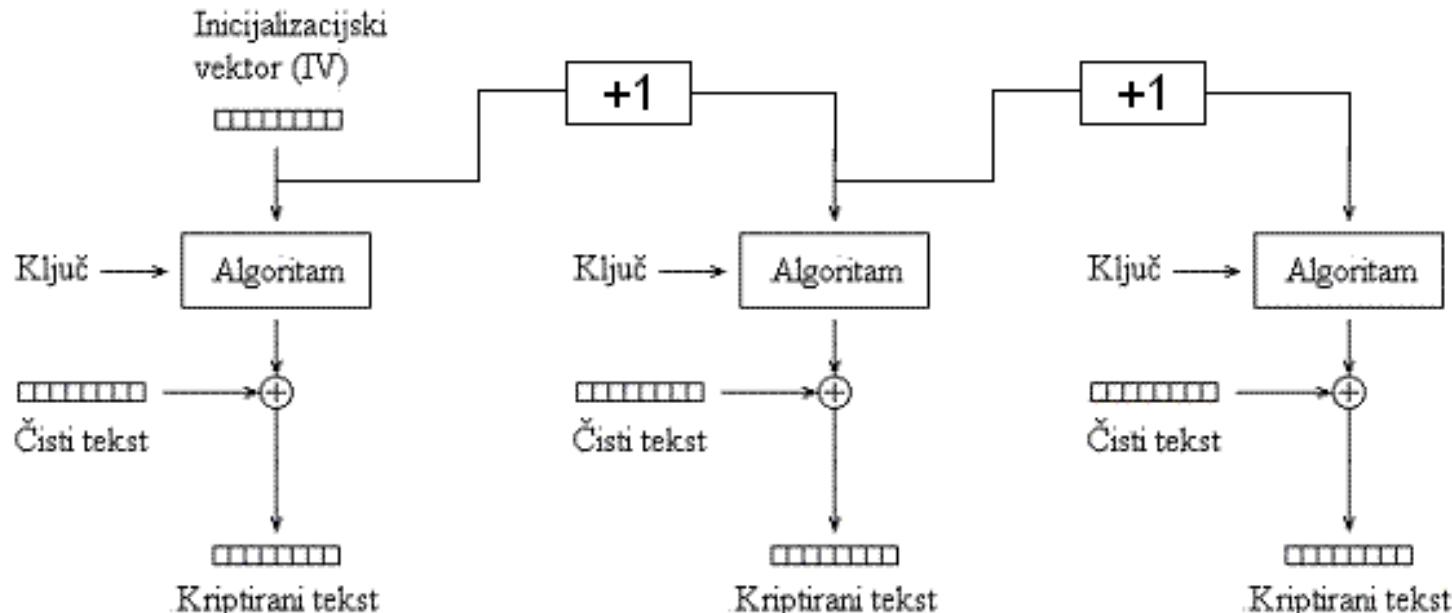
Output Feedback (OFB)

- Sličan protočnoj enkripciji odnosno jednostrukoj bilježnici: na temelju ključa i IV se izračuna niz bitova koji se XOR-a s jasnim tekstom.
- Dekripcija jednaka enkripciji.
- Zadatak: Što se događa ako se isti IV koristi dva puta?

Cipher Feedback (CFB)

- Sličan OFB načinu ali bitovi kojima se XOR-a dodatno ovise o jasnom tekstu.
- Dekripcija vrlo slična enkripciji.

Način kriptiranja *Counter Mode (CTR)*



CTR

- Sličan protočnoj enkripciji odnosno jednostrukoj bilježnici: na temelju ključa i IV se izračuna niz bitova koji se XOR-a s jasnim tekstom.
- Moguće paralelizirati.
- Nije potrebno nadopunjavati poruku.

Nadopunjavanje (padding)

- Kod CBC i nekih drugih načina kriptiranja je potrebno nadopuniti poruku do višekratnika duljine bloka.
- Nadopunjavanje mora biti invertibilno.
- Primjer: PKCSv7

```
01 -- if lth mod k = k-1  
02 02 -- if lth mod k = k-2  
.  
.  
.  
k k ... k k -- if lth mod k = 0
```

Izvor: <https://datatracker.ietf.org/doc/html/rfc5652>

Zadatak: *padding oracle* napad

- Jednostavna stvar poput nadopunjavanja može biti izvor sigurnosnih problema!
- Recimo da TLS poslužitelj koristi CBC način s PKCSv7 nadopunjavanjem. Kada TLS poslužitelj primi poruku, on je dekriptira te pokušava ukloniti nadopunjavanje.
 - Vraća „Invalid padding“ poruku ako nadopunjavanje nije ispravno.
- Napadač je video da klijent poslužitelju šalje poruku (IV, C_0, C_1) . Opišite način da napadač dekriptira zadnji blok.

Preporuke

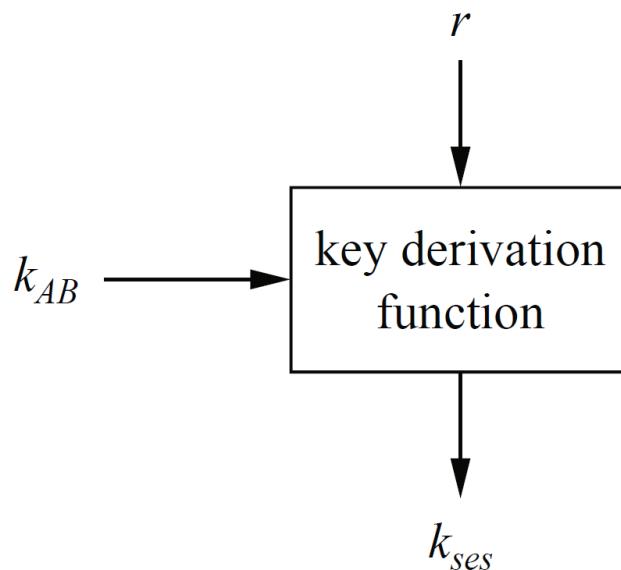
- Koristiti provjerena ostvarenja algoritama
 - NIKAKO se NE preporuča vlastita implementacija
- NE koristiti način kriptiranja ECB.
- IV se ne smije ponavljati i treba ga generirati slučajno (primjerice kod CBC načina kriptiranja)
- NE koristiti stalno isti simetrični ključ

Derivacija ključeva

- Problem: Kako općenito pretvoriti „tajne podatke“ u ključeve prikladane za simetričnu enkripciju?
 - Kako pretvoriti lozinku u ključ za simetričnu enkripciju?
 - Kako pretvoriti „matematičku“ dijeljenu tajnu u ključ za simetričnu enkripciju?
 - Što ako je “procurilo” pola 256-bitnog ključa?
 - Kako generirati više ključeva iz jednoga?

Funkcije za derivaciju ključa

- Funkcija za derivaciju ključa je deterministička funkcija koja kao ulaze prima:
 - Tajnu vrijednost k_{AB}
 - Javni parametar r
- Kao izlaz daje
 - Ključ za simetričnu šifru k_{ses}



Sigurnost funkcija za derivaciju ključa

- Neformalno, funkcija za derivaciju ključa je *sigurna* ako je napadač koji ne zna k_{AB} ne može odrediti nikakve informacije od $k_{ses} = KDF(k_{AB}, r)$
 - čak i ako zna r
 - čak i ako djelomično zna k_{AB}
 - čak i ako zna $KDF(k_{AB}, r_i)$ za mnoge $r_i \neq r$.

Sigurnost funkcija za derivaciju ključa

- Ako je KDF sigurna funkcija za derivaciju ključa onda je jednako teško
 - Saznati k_{AB} u potpunosti
 - Saznati bilo što o k_{ses}
- Posebno, funkcija za derivaciju ključa mora biti *jednosmjerna*.

Primjena: derivacija ključeva iz lozinki

- $k = KDF(\text{lozinka}, \text{salt})$
- *salt* se bira nasumično, ali je nakon toga javan (npr. pohranjuje se zajedno s skrivenim tekstom)
- Obično se koriste *password based* KDF koje su konstruirane tako da budu sporije.
 - PBKDF2
 - Scrypt
 - ...

Primjena: derivacija više ključeva iz jednog

- $k_1 = KDF(k_{master}, \text{"Encryption key 1"})$
- $k_2 = KDF(k_{master}, \text{"Encryption key 2"})$
- Alternativno koristi se KDF sa većim izlazom koji se podijeli.

Funkcije za derivaciju ključa – konstrukcije

- Bazirane na hash funkcijama: HKDF
- Bazirane na sustavima kriptiranja bloka
- ...

Funkcije za derivaciju ključa – česte greške

- Kriptografska hash funkcija često *nije* dobra funkcija za derivaciju ključa.
 - Npr. $KDF(k_{AB}, r) = SHA256(k_{AB} || r)$
- *Length extension* napad
 - Na temelju $SHA256(x)$ je ponekad moguće izračunati $SHA256(x || x')$

Laboratorijska vježba: double ratchet

- 1. vježba: symmetric key ratchet
- 2. vježba: Difflie-Hellman ratchet

Demo



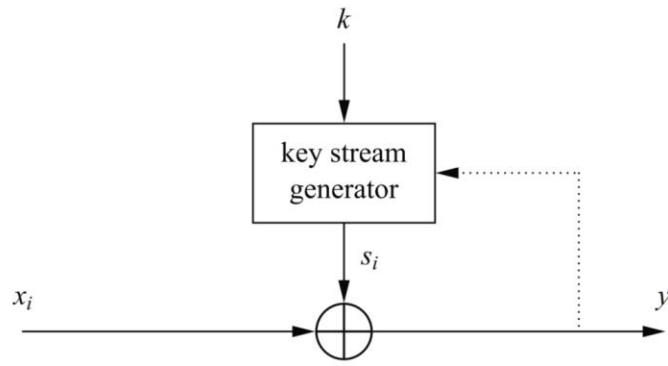
Kriptografija i kriptoanaliza

doc. dr. sc. Ante Đerek i prof. dr. sc. Marin Golub

Listopad 2023.

Algoritam kriptiranja toka podataka

- Protočna enkripcija / protočna šifra (*stream cipher*).
- Generira *tok ključa* koji se „zbraja” s jasnim tekstom operacijom XOR.
 - *Sinkrona protočna enkripcija* – tok ključa ovisi samo o ključu.
 - *Asinkrona protočna enkripcija* – tok ključa ovisi o ključu i prethodnim bitovima jasnog teksta.

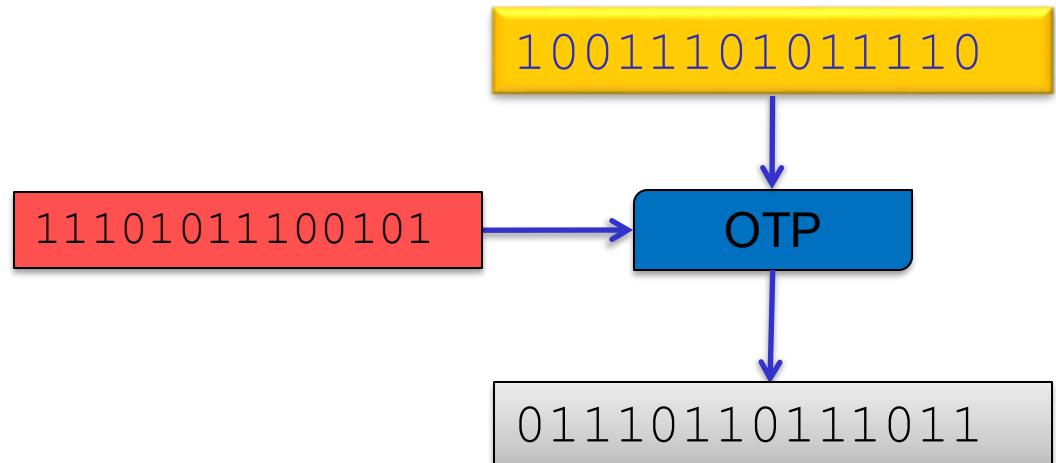


Primjeri protočne enkripcije

- RC4 (1987)
 - ključ veličine 40–2048 bitova
 - vrlo široko korišten, mnoštvo poznatih slabosti
- CSS (1996)
 - 40-bitni ključ
 - zaštita sadržaja na DVD-ovima
 - potpuno razbijen 1999. godine
- Salsa20/ChaCha (2005)
 - ključ 128 ili 256 bitova
 - podržan u TLS-u
 - alternativa AES-u zbog boljih performansi na uređajima gdje sklopoljje ne implementira AES

Ponavljanje: jednokratna bilježnica

- $M = K = C = \{0, 1\}^n$
- $E(m, k) = m \oplus k$
- $D(c, k) = c \oplus k$



Ponavljanje: Savršena povjerljivost

- Claude Shannon, 1946
- Jednokratna bilježnica pruža savršenu povjerljivost:
 - Za svaku poruku $m \in \{0, 1\}^n$ i šifrat $c \in \{0, 1\}^n$ i vrijedi:

$$P_{k \leftarrow \{0,1\}^n}(E(m, k) = c) = \frac{1}{2^n}.$$

- Alternativno, za svaku poruku $m \in \{0, 1\}^n$ izlazi sljedeća dva vjerojatnostna algoritma imaju jednake razdiobe:

$$[k \xleftarrow{R} \{0, 1\}^n; \text{output } E(m, k)]$$

$$[k \xleftarrow{R} \{0, 1\}^n; m' \xleftarrow{R} \{0, 1\}^n; \text{output } E(m', k);]$$

Ponavljanje: jednokratna bilježnica – nedostaci

- Ključ

- mora se generirati potpuno i uistinu slučajno!
- mora biti jednak velik kao i poruka!
- smije se koristiti najviše jednom!

$$c_1 = m_1 \oplus k$$

$$c_2 = m_2 \oplus k$$

$$c_1 \oplus c_2 = m_1 \oplus m_2$$

- Moguće je na predvidiv način izmijeniti poruku (engl. *malleable encryption*) !

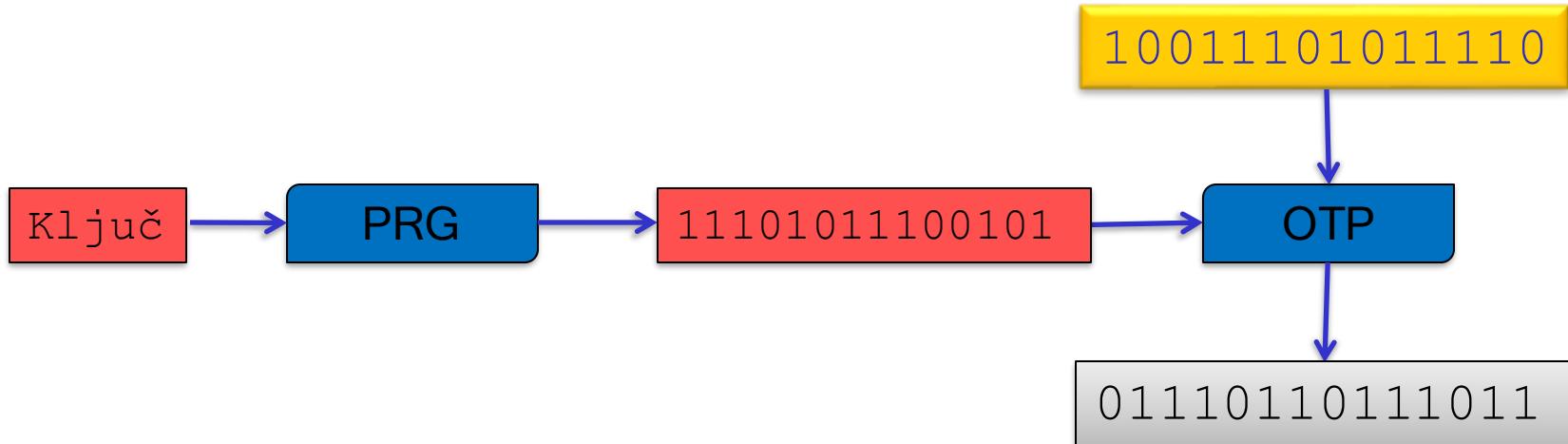
$$c_1 = OTP(m_1, k) = m_1 \oplus k$$

$$c_2 = c_1 \oplus m_1 \oplus m_2 = m_1 \oplus k \oplus m_1 \oplus m_2 = m_2 \oplus k = OTP(m_2, k)$$



Algoritam kriptiranja toka podataka

- Ideja: umjesto slučajnog ključa koristimo *pseudoslučajni ključ*.
- Generator pseudoslučajnih brojeva na temelju ključa generira niz bitova koji se XOR-a s izvornim tekstrom.



Generator pseudoslučajnih brojeva – definicija

- Generator pseudoslučajnih brojeva je efikasni deterministički algoritam $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$ gdje je $n \gg s$.
- Ideja: G na determinističan način od relativnog kratkog (npr. 128 bita) slučajnog *sjemena* (seed) generira vrlo dugačak (npr. 1GB) niz bitova koji *izgleda* slučajno.

Generator slučajnih brojeva – nomenklatura

- Generator slučajnih brojeva / *True random number generator* (TRNG)
 - Koristi prirodne slučajne procese ili ad-hoc informacije kako bi generirao *stvarno* slučajne brojeve.
- Generator pseudoslučajnih brojeva / *Pseudorandom number generators* (PRG)
 - Na temelju slučajnog *sjemena* deterministični računa vrijednosti koje *izgledaju slučajno*. Npr. rand() u programskom jeziku C.
- Kriptografski generator pseudoslučajnih brojeva / *Cryptographically secure pseudorandom number generator*
 - Generatori pseudoslučajnih brojeva s jakim sigurnosnim svojstvom nepredvidivosti.

Generator pseudoslučajnih brojeva – sigurnost

- Neformalno, generator pseudoslučajnih brojeva je *nepredvidiv* ako je napadaču (koji ne zna sjeme) jako teško predvidjeti izlaz generatora.
 - Čak i ako napadač vidi velik prefiks izlaza.

Protočna enkripcija

- Ako je $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$ generator pseudoslučajnih brojeva onda možemo definirati protočnu enkripciju na sljedeći način:
 - $E(m, k) = m \oplus G(k)$
 - $D(c, k) = c \oplus G(k)$

Zadatak: protočna enkripcija s previdivim generatorom

- Neka je G generator pseudoslučajnih brojeva koji je predvidiv. Neka je $E(m, k) = m \oplus G(k)$.
- Pokaži da E nije sigurna simetrična enkripcija pod pretpostavkom da napadač može pogoditi prefiks poruke koja se kriptira.

“Obični” generatori pseudoslučajnih brojeva

- *Linear congruential generator.*
 - S_0 = seed
 - $S_i = (aS_{i-1} + b) \text{ mod } m$
- Odlična statistička svojstva.
- Predvidiv i stoga neupotrebljiv u kriptografiji.

“Obični” generatori pseudoslučajnih brojeva

```
* This is a linear congruential pseudorandom number generator, as
* defined by D. H. Lehmer and described by Donald E. Knuth in
* <cite>The Art of Computer Programming, Volume 2, Third edition:
* Seminumerical Algorithms</cite>, section 3.2.1.
*
* @param bits random bits
* @return the next pseudorandom value from this random number
*         generator's sequence
* @since 1.1
*/
protected int next(int bits) {
    long oldseed, nextseed;
    AtomicLong seed = this.seed;
    do {
        oldseed = seed.get();
        nextseed = (oldseed * multiplier + addend) & mask;
    } while (!seed.compareAndSet(oldseed, nextseed));
    return (int)(nextseed >>> (48 - bits));
}
```

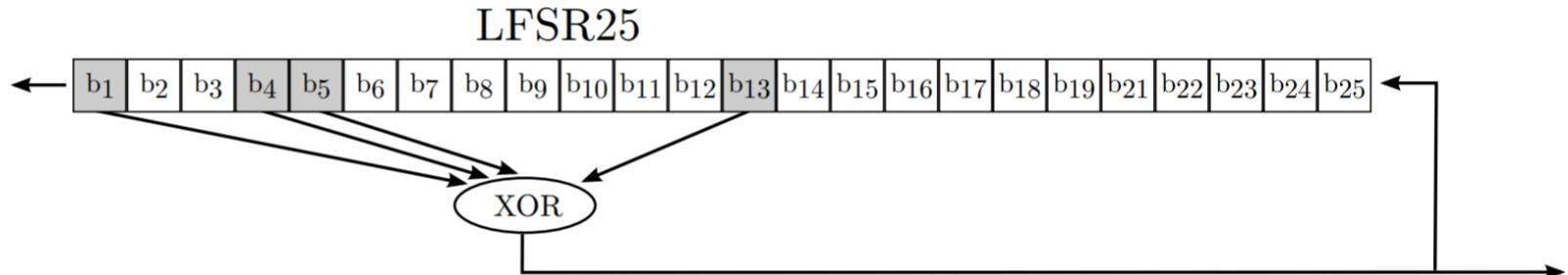
<https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/util/Random.java>

LFSR

- Linearni posmačni registar s povratnom vezom / *Linear Feedback Shift Register* (LFSR)
 - Stanje se sastoji od m bitova
 - U svakom ciklusu se izračuna XOR bitova na fiksnim (povratnim) pozicijama.
 - Rezultat b je izlaz ciklusa.
 - Stanje se posmiče i bit b se dodaje na kraj

Zadatak: Sigurnost LFSR-a

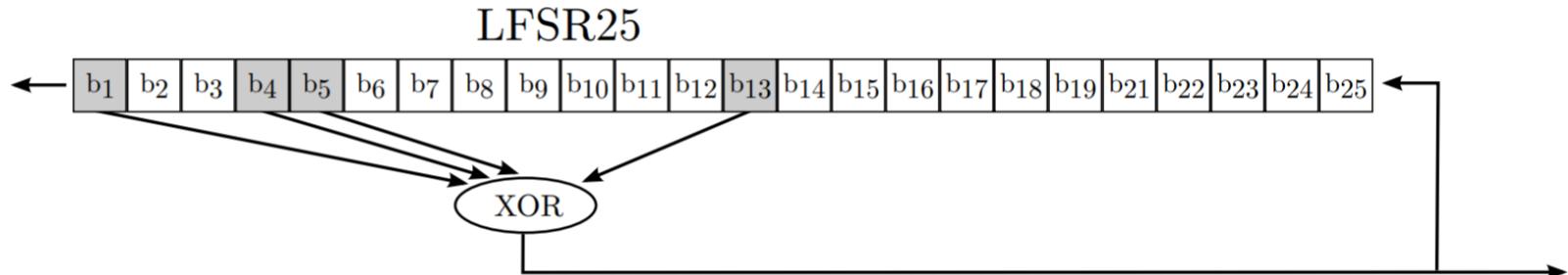
- Poznata je arhitektura 128-bitnog LFSR-a i poznato je prvih 1000 bitova njegovog izlaza. Predvidite ostale bitove.



Izvor: <https://hsin.hr/pripreme2016/zadaci/prvi/zadaci.pdf>

Zadatak: Sigurnost LFSR-a

- Poznata je arhitektura 128-bitnog LFSR-a i poznato je drugih 1000 bitova njegovog izlaza. Predvidite ostale bitove.



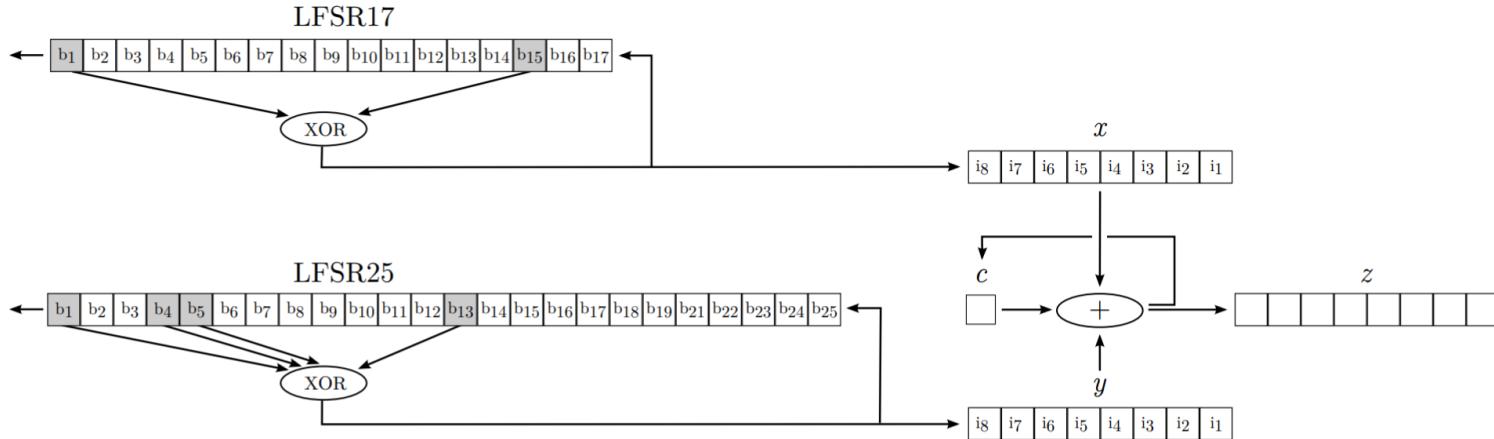
Izvor: <https://hsin.hr/pripreme2016/zadaci/prvi/zadaci.pdf>

LFSR

- Vrlo jednostavan i jeftin za sklopovsku implementaciju.
- Potpuno nesiguran za kriptografija.
- Baza za mnoge protočne enkripcije.

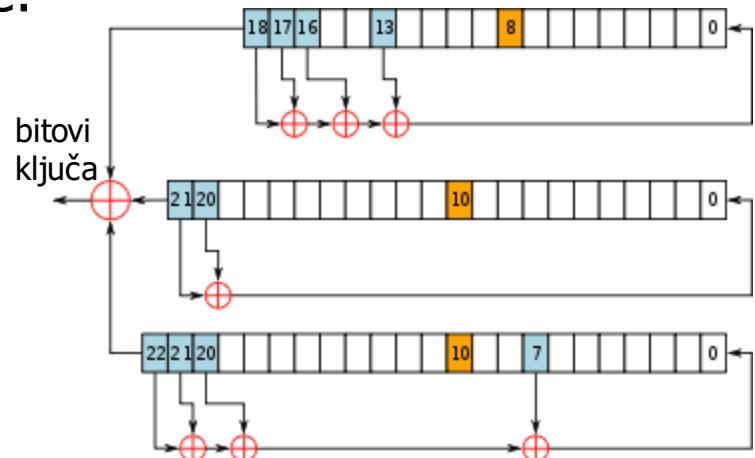
Primjer: CSS

- *Content Scramble System*
- Mehanizam zaštite sadržaja na DVD-ovima
- 40-bitni ključevi
- Jednostavni napad poznatim jasnim tekstom u 2^{17} koraka.



Primjer: A5/1

- U postupku generiranja enkripcijskog ključa koriste se 3 posmakačna registra duljine 19, 22 i 23 bita.
- Postupak kriptiranja odvija se u 3 faze:
 1. učitavanje 64-bitnog ključa u 3 registra u 64 koraka
 2. postavljanje početnog stanja registara u 100 koraka
 3. stvaranje niza bitova ključa
- interaktivni simulator je dostupan na
<https://733amir.github.io/a51-cipher-simulator/>



izvor: wikipedia.org

Primjer: A5/1

- Zabilježeni su brojni napadi.
- Eksperimentalno je utvrđeno da nakon postavljanja početnog stanja registara, koji se odvija u 100 koraka, stanje registara može poprimiti svega 15% svih mogućih stanja kojih ima 2^{64}
 - dakle, prostor pretraživanja se svodi na $2^{64} \times 0,15 \approx 2^{61,26}$
- 2006. g. kriptografi E. Barkan, E. Biham i N. Keller demonstrirali su napad na algoritam A5/1 koji omogućuje dekriptiranje razgovora u stvarnom vremenu.

eStream Contest

- 2004. – 2008.
- Profile 1: Stream ciphers for software applications with high throughput. Must support 128-bit key. Must support 64-bit IV and 128-bit IV.
- Profile 2: Stream ciphers for hardware applications with highly restricted resources. Must support 80-bit key. Must support 32-bit IV and 64-bit IV.

Profile 1 (software)	Profile 2 (hardware)
HC-128 [1] ↗	Grain [2] ↗
Rabbit [3] ↗	MICKEY [4] ↗
Salsa20/12 [5] ↗	Trivium [6] ↗
SOSEMANUK [7] ↗	

Primjer: Salsa20/ChaCha

- Ulaz u generator je ključ i *nonce* vrijednost.
 - $E(m, k, r) = m \oplus G(k, r)$
 - Nonce vrijednost omogućuje ponovno korištenje ključa. Ako se dva puta koristi isti nonce s istim ključem onda sustav više nije siguran.
- Generator (nezavisno) generira 512-bitne blokove uz pomoć brojača.
 - $G(k, r) = C(k, r, 0) || C(k, r, 1) || C(k, r, 2) || \dots$
 - Moguća paralelizacija i brzo dekriptiranje na proizvoljnoj lokaciji unutar kriptiranog teksta.

Primjer: Salsa20/ChaCha

```
void salsa20_block(uint32_t out[16], uint32_t const in[16])
{
    int i;
    uint32_t x[16];

    for (i = 0; i < 16; ++i)
        x[i] = in[i];
    // 10 Loops x 2 rounds/Loop = 20 rounds
    for (i = 0; i < ROUNDS; i += 2) {
        // Odd round
        QR(x[ 0], x[ 4], x[ 8], x[12]); // column 1
        QR(x[ 5], x[ 9], x[13], x[ 1]); // column 2
        QR(x[10], x[14], x[ 2], x[ 6]); // column 3
        QR(x[15], x[ 3], x[ 7], x[11]); // column 4
        // Even round
        QR(x[ 0], x[ 1], x[ 2], x[ 3]); // row 1
        QR(x[ 5], x[ 6], x[ 7], x[ 4]); // row 2
        QR(x[10], x[11], x[ 8], x[ 9]); // row 3
        QR(x[15], x[12], x[13], x[14]); // row 4
    }
    for (i = 0; i < 16; ++i)
        out[i] = x[i] + in[i];
}
```

Initial state of Salsa20

"expa"	Key	Key	Key
Key	"nd 3"	Nonce	Nonce
Pos.	Pos.	"2-by"	Key
Key	Key	Key	"te k"

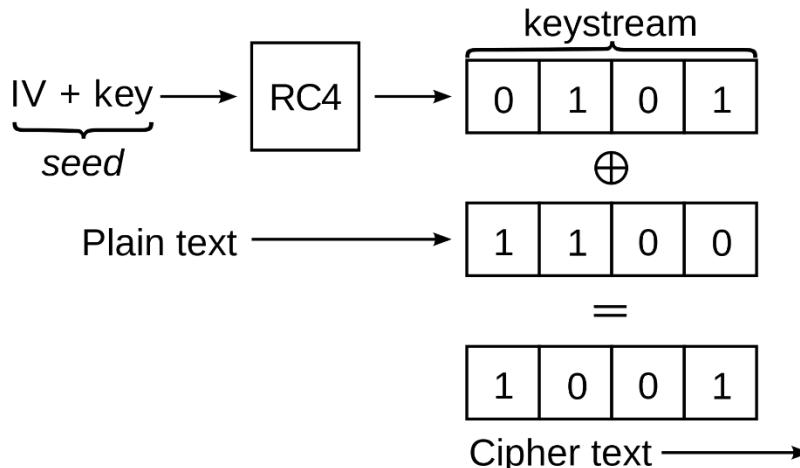
```
#define ROTL(a,b) (((a) << (b)) | ((a) >> (32 - (b))))
#define QR(a, b, c, d)(      \
    b ^= ROTL(a + d, 7),   \
    c ^= ROTL(b + a, 9),   \
    d ^= ROTL(c + b,13),  \
    a ^= ROTL(d + c,18))
```

Napadi na protočnu enkripciju

- Svi napadi na OTP su i napadi na protočnu enkripciju!
 - Ako isti ključ koristimo dva puta, nema nikakve sigurnosti!
 - Enkripcija ne pruža nikakvo svojstvo integriteta!

Primjer: WEP

- Ključ (40-bitni ili 104-bitni) se spaja s 24-bitnim IV i koristi kao ključ za RC4 protočnu enkripciju.
- Ključevi su povezani, a RC4 je ranjiv na takozvani *related key* napad.
- Potpuno razbijen u 2001.



izvor: wikipedia.org

Generiranje entropije

- Ad-hoc slučajne informacije sa sistema
 - /dev/random na Linux sustavima koristi vremenske informacije s tipkovnice i miša, vremenske informacije o prekidima, vremenske informacije o operacijama diska, podatke o samom uređaju (npr. serijski brojevi).
 - <https://github.com/torvalds/linux/blob/master/drivers/char/random.c>
- Specijalizirani hardver
 - Koristi prirodne slučajne procese (termalni šum, fotoelektrični efekt, ...)
- RDRAND (Intel)
 - Stvarna slučajnost i kriptografski generator pseudoslučajnih brojeva.
 - “The ES runs asynchronously on a self-timed circuit and uses thermal noise within the silicon to output a random stream of bits at the rate of 3 GHz.”

Ako nema slučajnih brojeva nema ni sigurnosti!

- Sigurnost svih sustava ovisi u pretpostavci da je moguće na slučajan način generirati:
 - Ključeve
 - Inicijalizacijske vektore
 - Nonce vrijednosti u protokolima
 - ...
- Ako napadač može predvidjeti ključeve, nikakva sigurnost nije garantirana!

Primjer: Netscape 1.1 (1995)

- Napad: Za sve moguće seed parametre
 - Generiraj ključ koristeći isti postupak
 - Provjeri je li moguće dešifrirati komunikaciju s dobivenim ključem

```
global variable seed;

RNG_CreateContext()
    (seconds, microseconds) = time of day; /* Time elapsed since 1970 */
    pid = process ID; ppid = parent process ID;
    a = mklcpr(microseconds);
    b = mklcpr(pid + seconds + (ppid << 12));
    seed = MD5(a, b);

mklcpr(x) /* not cryptographically significant; shown for completeness */
    return ((0xDEECE66D * x + 0x2BB862DC) >> 1);

MD5() /* a very good standard mixing function, source omitted */
```

Figure 2: The Netscape 1.1 seeding process: pseudocode.

```
RNG_GenerateRandomBytes()
    x = MD5(seed);
    seed = seed + 1;
    return x;

global variable challenge, secret_key;

create_key()
    RNG_CreateContext();
    tmp = RNG_GenerateRandomBytes();
    tmp = RNG_GenerateRandomBytes();
    challenge = RNG_GenerateRandomBytes();
    secret_key = RNG_GenerateRandomBytes();
```

Figure 3: The Netscape v1.1 key-generation process: pseudocode.

Izvor: Goldberg, Wagner, „Randomness and the Netscape Browser”, 1996.

Primjer: Debian OpenSSL (2005)

On May 13th, 2008 the Debian project [announced](#) that Luciano Bello found an interesting vulnerability in the OpenSSL package they were distributing. The bug in question was caused by the removal of the following line of code from *md_rand.c*

```
MD_Update(&m,buf,j);
[ .. ]
MD_Update(&m,buf,j); /* purify complains */
```

These lines were [removed](#) because they caused the [Valgrind](#) and Purify tools to produce warnings about the use of uninitialized data in any code that was linked to OpenSSL. You can see one such report to the OpenSSL team [here](#). Removing this code has the side effect of crippling the seeding process for the OpenSSL PRNG. Instead of mixing in random data for the initial seed, the only “random” value that was used was the current process ID. On the Linux platform, the default maximum process ID is 32,768, resulting in a very small number of seed values being used for all PRNG operations.

Izvor:
schneier.com

Druge konstrukcije PRG-ova

- Bazirane na sustavima kriptiranja bloka.
 - Npr. CRT_DBRG
- Bazirane na kriptografskim hash funkcijama
 - HASH_DBRG, HMAC_DBRG
- Specijalizirane konstrukcije.

Formalne definicije sigurnosti

- Cilj: pokazati kako se formalno može definirati sigurnost i kako se može dokazati sigurnost kriptografske konstrukcije na temelju sigurnosti pojedinih primitiva.

PRG

- Želimo na neki način definirati što znači da napadač ne može razlikovati generator pseudoslučajnih brojeva od generatora stvarno slučajnih brojeva.
- Neka je $G: K \rightarrow \{0, 1\}^n$ PRG, želimo da sljedeće dvije razdiobe budu „nerazlučive“ (*indistinguishable*).

$$[k \xleftarrow{R} K; \text{output } G(k)]$$

$$[r \xleftarrow{R} \{0, 1\}^n; \text{output } r;]$$

Statistički test

- *Statistički test* je bilo koji algoritam koji prima niz od n bitova i pokušava odrediti je li taj niz slučajan.
- $A: \{0, 1\}^n \rightarrow \{0, 1\}$
- Primjer:
 - $A(x) = 1$ ako je razlika broja jedinica i broja nula manja od \sqrt{n} , a $A(x) = 0$ inače.

Prednost

- Neka je $G: K \rightarrow \{0, 1\}^n$ PRG
- Neka je $A: \{0, 1\}^n \rightarrow \{0, 1\}$ statistički test
- Prednost (advantage) statističkog testa A definiramo kao:

$$\text{Adv}_{PRG}(A, G) = |P_{k \leftarrow K}(A(G(k)) = 1) - P_{x \leftarrow \{0,1\}^n}(A(x) = 1)|$$

- Prednost blizu nuli: test ne razlikuje G od slučajnih brojeva.
- Prednost blizu jedinici: test razlikuje G od slučajnih brojeva.

Zadatak: Prednost

- Neka je $G: K \rightarrow \{0, 1\}^n$ PRG koji ima svojstvo da je XOR prvih 10 bitova jednak 1 s vjerojatnošću $\frac{3}{4}$.
- Postoji li statistički test koji ima veliku prednost?

Definicija sigurnosti

- Neka je $G: K \rightarrow \{0, 1\}^n$ PRG. Kažemo da je G *siguran* ako ako svaki efikasni statistički test ima zanemarivo malu prednost.

Što znače efikasno i zanemarivo?

- Praktični pristup: fiksne vrijednosti.
- G je (t, ε) -siguran PRG ako svaki algoritam koji radi t koraka ima prednost strogog manju od ε .
- Konkretne poželjne vrijednosti ovise o situaciji, npr: $t = 2^{80}, \varepsilon = 2^{-80}$.

Što znače efikasno i zanemarivo?

- Teorijski pristup: funkcija sigurnosnog parametra.
- G_i je siguran PRG ako je svaki vjerojatnostni algoritam A koji radi u polinomom vremenu njegova prednost ε_i zanemariva funkcija.
 - Funkcija $f: \mathbb{N} \rightarrow \mathbb{R}$ je *zanemariva* ako za svaki broj $c \in \mathbb{R}^+$ postoji $n_0 \in \mathbb{N}$ takav da za sve $n > n_0$ vrijedi $|f(n)| < \frac{1}{n^c}$.
 - Intuicija: zanemariva funkcija raste sporije nego inverz svakog polinoma.

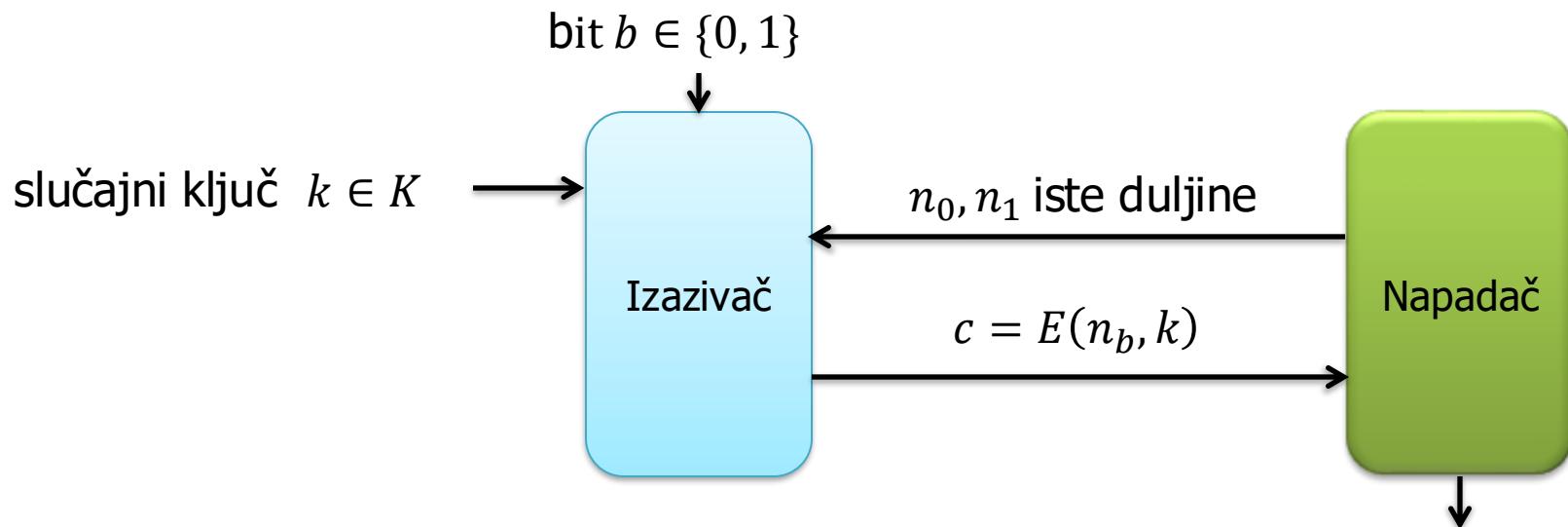
Zadatak: Sigurnost i nepredvidivost

- Neka je $G: K \rightarrow \{0, 1\}^n$ siguran PRG, pokažite da je G nepredvidiv.
- Postoji li statistički test koji ima veliku prednost?
- Vrijedi i obrat: svaki nepredvidiv PRG je siguran (Yao, 1982).

Semantička sigurnost

- Želimo definirati sigurnost simetrične enkripcije koja je općenitija od savršene povjerljivosti.
 - Ali dokle god se ključ koristi samo jednom, dakle ne uzima u obzir napade poznatim jasnim tekstom, odabranim jasnim tekstom, itd.

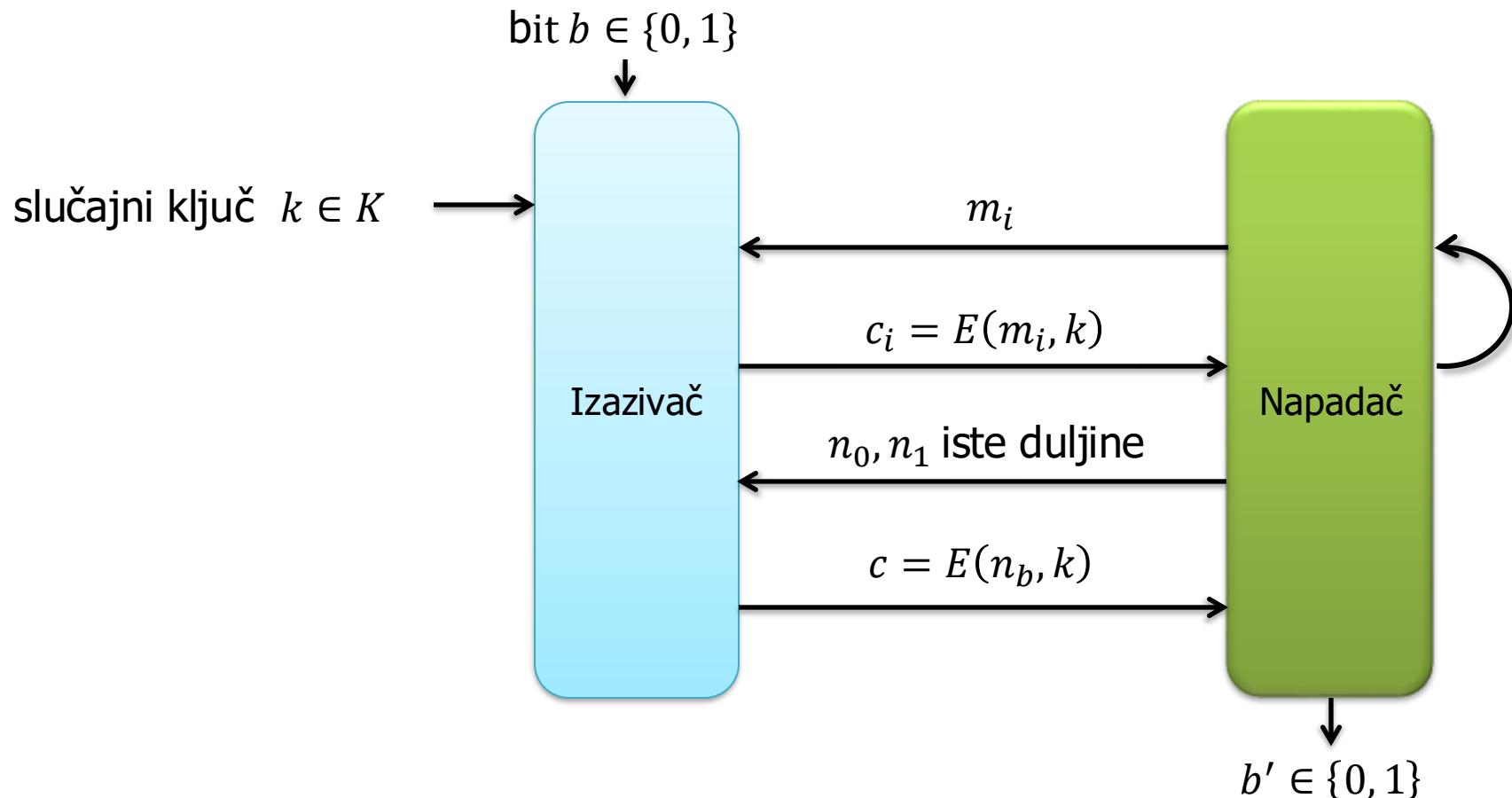
Semantička sigurnost



- W_0 : događaj u eksperimentu 0 kada A kaže 1
- W_1 : događaj u eksperimentu 1 kada A kaže 1

$$\text{Adv}_{SS}(A) = |P(W_0) - P(W_1)|$$

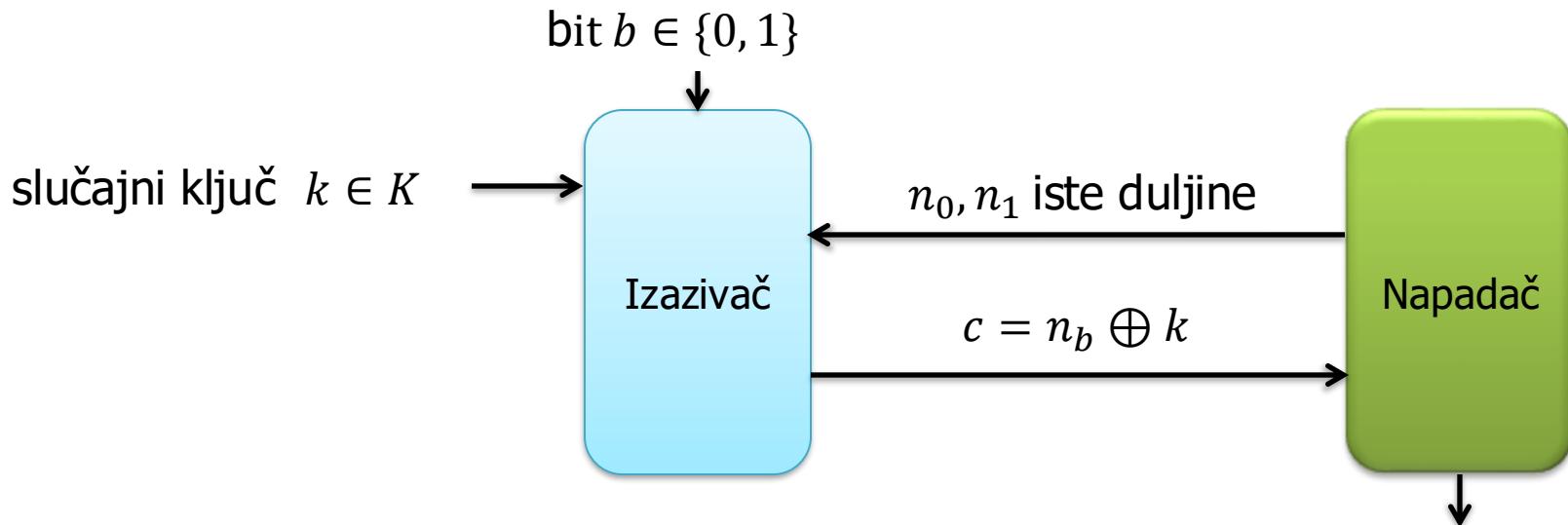
Digresija: Jače definicije sigurnosti



Semantička sigurnost

- Simetrična enkripcija je semantički sigurna ako je za svakog efikasnog napadača A njegova prednost $\text{Adv}_{SS}(A)$ zanemarivo mala.

Semantička sigurnost jednokratne bilježnice



- R_0 : događaj u eksperimentu 0 kada A kaže 1
- R_1 : događaj u eksperimentu 1 kada A kaže 1

$$\text{Adv}_{SS}(A) = |P(R_0) - P(R_1)| = 0$$

Sigurnost protočne enkripcije

- Teorem: Ako je G siguran PRG onda je $E(m, k) = m \oplus G(k)$ semantički sigurna enkripcija.
- Dokaz ćemo pokazati ali ga nije potrebno znati.

3. Funkcije za izračunavanje sažetka poruke

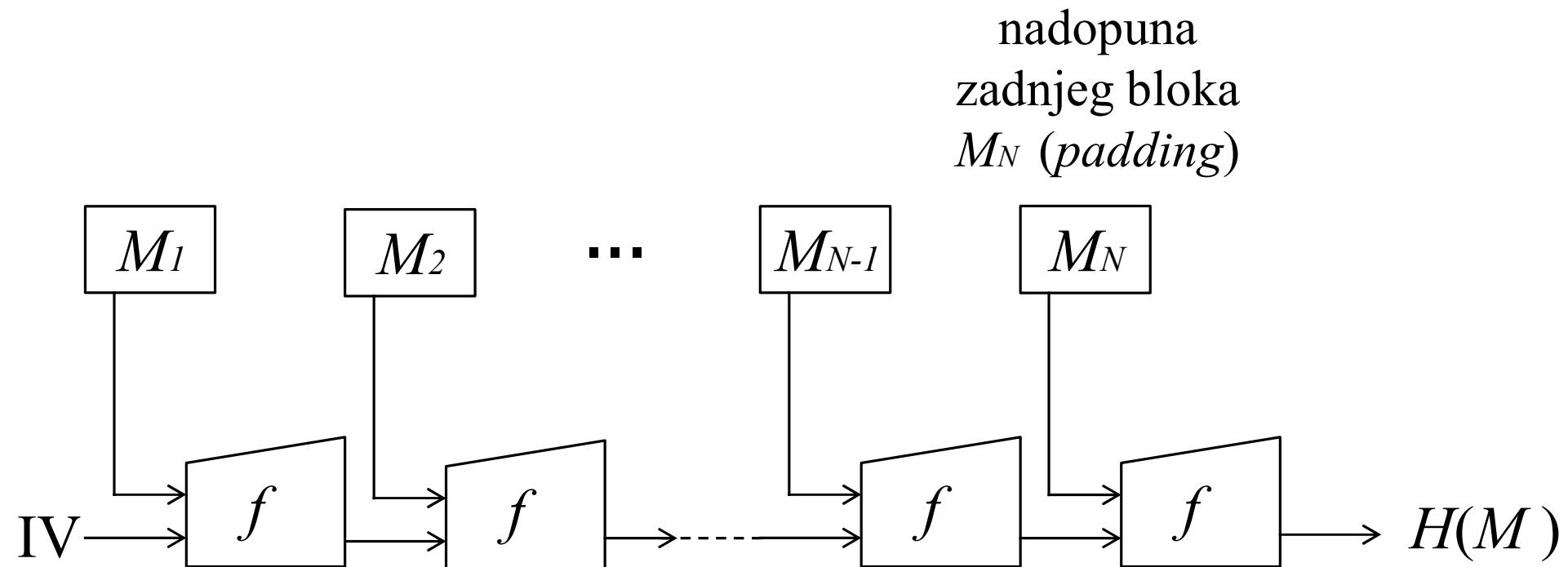
Funkcije sažimanja ili *hash* funkcije

Važna svojstva funkcija za izračunavanje sažetka poruke

- Otpornost na izračunavanje originala ili prva domenska otpornost (*preimage resistance*)
 - $H=h(M) \Rightarrow M=h^{-1}(H)$ – inverz ne postoji
 - za dani sažetak H teško je naći poruku M tako da je $H=h(M)$
- Otpornost na izračunavanje poruke koja daje isti sažetak ili druga domenska otpornost (*2-nd preimage resistance*)
 - za poznati M i $H=h(M)$ je vrlo teško pronaći M' koji daje isti H
- Otpornost na kolizije (*collision resistance*)
 - nemoguće je pronaći bilo koje dvije poruke M_1 i M_2 za koje se dobiva isti sažetak $h(M_1)=h(M_2)$
- Difuzija
 - svaka, pa i najmanja promjena ulaznog podatka rezultira velikom i naizgled slučajnom promjenom na izlazu

Konstrukcija Merkle–Damgård

- koriste je MD5, SHA-1, SHA-2 i druge funkcije za izračunavanje sažetka poruke



MD5

- *Message Digest* = sažetak poruke
- proizvodi 128-bitovni sažetak
- izvorni tekst dijeli se na blokove duljine **512** bitova
- zadnji blok teksta se nadopunjuje (engl. *padding*) do 512 bitova tako da se:
 - iza zadnjeg bita teksta dodaje jedna jedinica
 - nakon 1 upisuju se nule tako da u bloku preostanu 64 bita
 - u ta 64 bita se upisuje bitovna duljina izvorne poruke
- svaki blok se dijeli na 16 podblokova po 32 bita :

$$M_0, M_1, M_2, \dots, M_{15}$$

Funkcije i konstante algoritma MD5

- sažetak H od 128 bitova sastoji se od 4 nadovezanih 32-bitovnih varijabli koje se inicijaliziraju s vrijednostima:

$$A_0 = 01234567_{16} \quad B_0 = 89ABCDEF_{16}$$

$$C_0 = FEDCBA98_{16} \quad D_0 = 76543210_{16}$$

- postupak se obavlja u 64 koraka podijeljena u 4 kruga
 - ⇒ svaki krug se sastoji od 16 koraka
- u svakom krugu koristi se jedna od četiri funkcije

$$F_i(x, y, z) = (x \wedge y) \vee (\neg x \wedge z), \quad 1 \leq i \leq 16$$

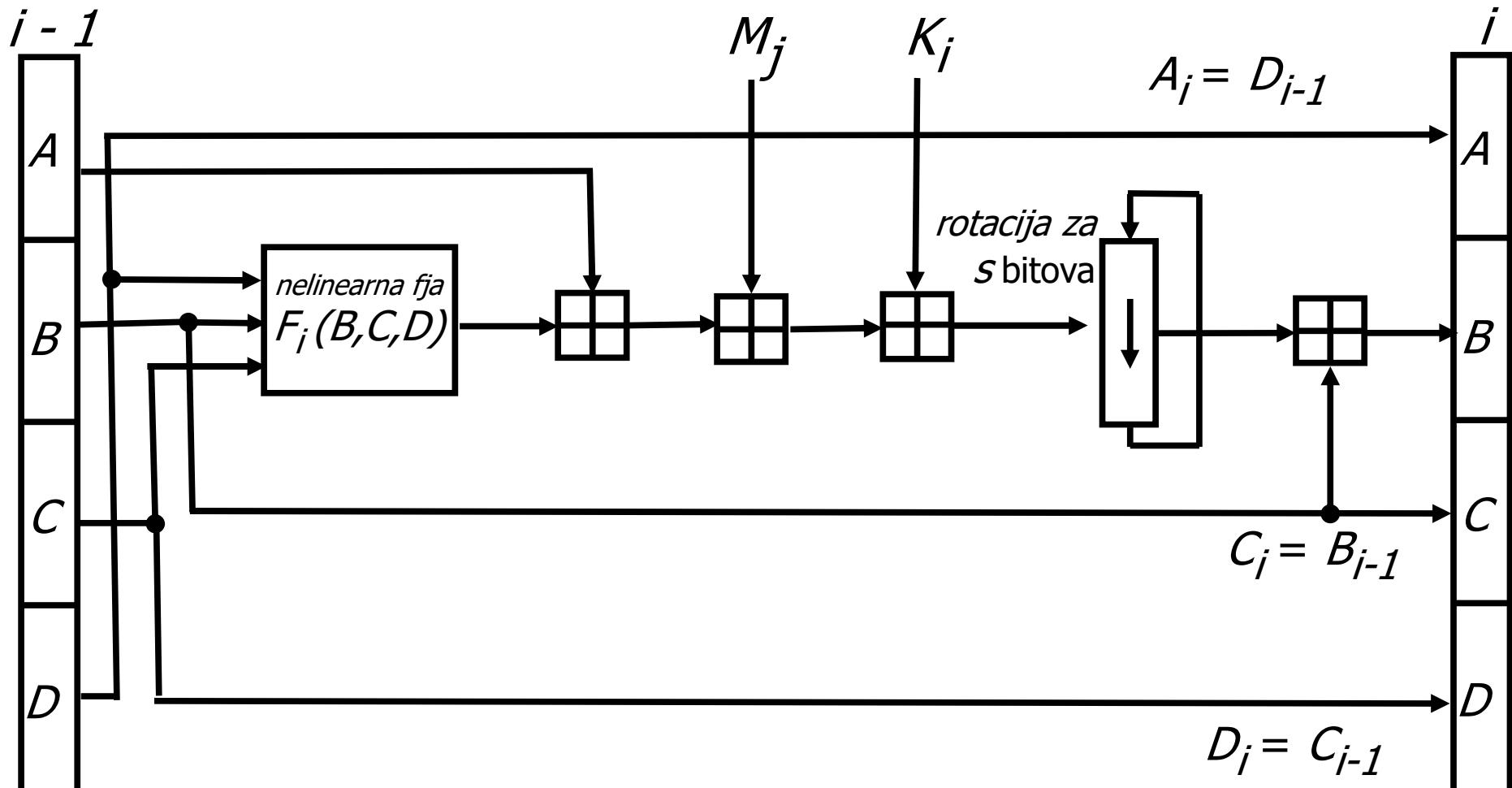
$$F_i(x, y, z) = (x \wedge y) \vee (x \wedge \neg z), \quad 17 \leq i \leq 32$$

$$F_i(x, y, z) = x \oplus y \oplus z, \quad 33 \leq i \leq 48$$

$$F_i(x, y, z) = y \oplus (x \wedge \neg z), \quad 49 \leq i \leq 64$$

- u svakom koraku koristi se sljedeća varijabla:

$$K_i = 2^{32} \times \text{abs}(\sin(i)), \quad 1 \leq i \leq 64$$



$$B_i = B_{i-1} + ((A_{i-1} + F_i(B_{i-1}, C_{i-1}, D_{i-1}) + M_j) \ll s)$$

$$\begin{aligned} S &= ABCD, \text{ gdje su } A = A_{64} + A_0 & B &= B_{64} + B_0 \\ C &= C_{64} + C_0 & D &= D_{64} + D_0 \end{aligned}$$

SHA-1

- proizvodi **160**-bitovni sažetak
- podjela jasnog teksta na blokove od 512 bitova i nadopuna zadnjeg bloka (*padding*) odvija se na jednak način kao i kod algoritma MD5
- sažetak H od 160 bitova sastoji se od 5 nadovezanih 32-bitovnih varijabli koje se inicijaliziraju s vrijednostima:

$$A_0 = 67452301_{16} \quad B_0 = EFCDAB89_{16}$$

$$C_0 = 98BADCFE_{16} \quad D_0 = 10325476_{16} \quad E_0 = C3D2E1F0_{16}$$

Funkcije i konstante algoritma SHA-1

- podblokovi M_0, \dots, M_{15} služe za stvaranje 80 riječi

$$W_i = M_{i-1}, \quad 1 \leq i \leq 16$$

$$W_i = (W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) \lll 1, \quad 17 \leq i \leq 80$$

- sažimanje svakog podbloka obavlja se u 4 kruga, svaki s 20 koraka, tj. ukupno 80 koraka, a u svakom krugu koristi se jedna od četiri funkcije i konstante:

$$F_i = (X \wedge Y) \vee (\neg X \wedge Z), \quad 1 \leq i \leq 20$$

$$F_i = X \oplus Y \oplus Z, \quad 21 \leq i \leq 40$$

$$F_i = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z), \quad 41 \leq i \leq 60$$

$$F_i = X \oplus Y \oplus Z, \quad 61 \leq i \leq 80$$

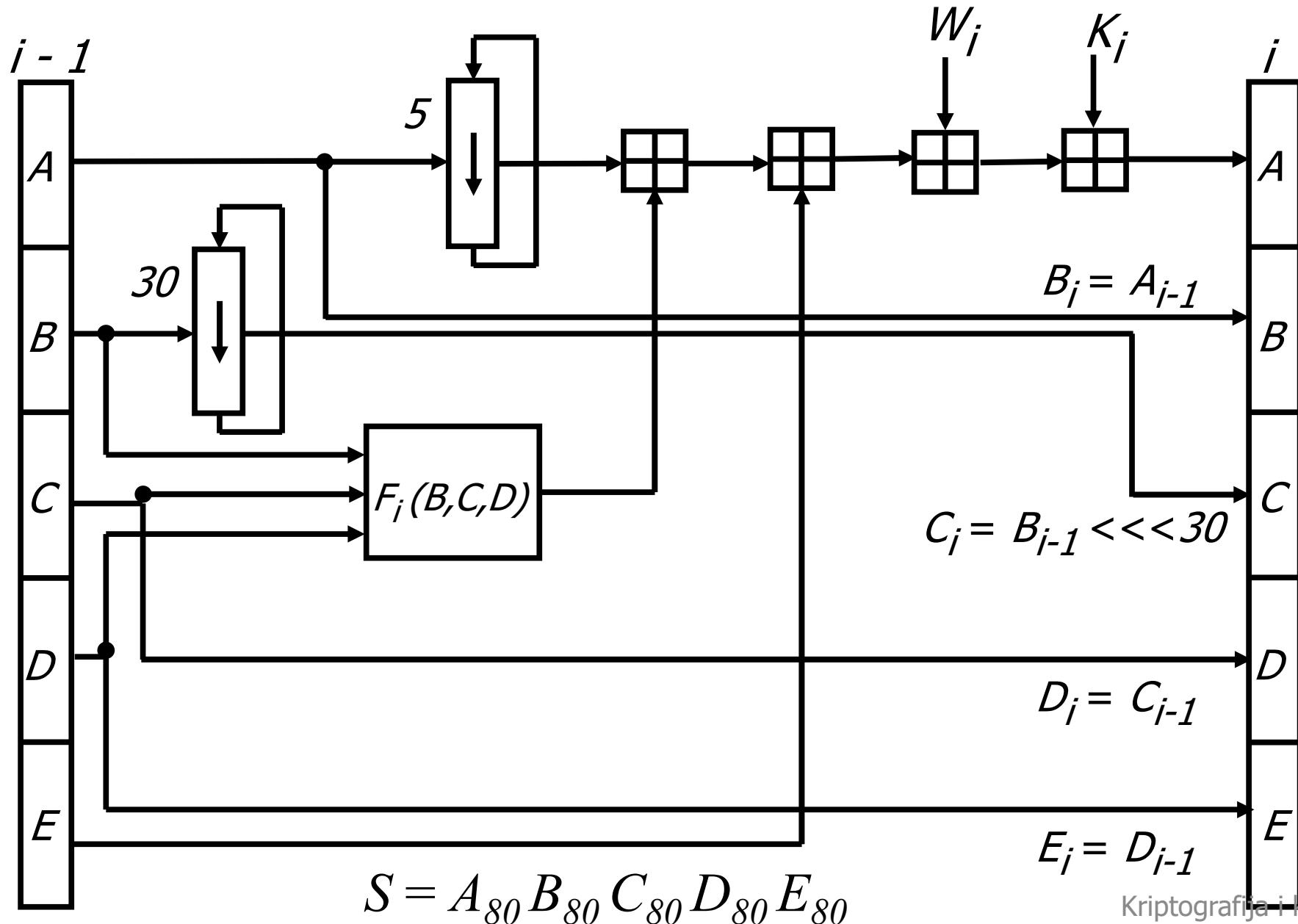
$$K_i = 5A827999_{16}, \quad 1 \leq i \leq 20$$

$$K_i = 6ED9EBA1_{16}, \quad 21 \leq i \leq 40$$

$$K_i = 8F1BBCDC_{16}, \quad 41 \leq i \leq 60$$

$$K_i = CA62C1D6_{16}, \quad 61 \leq i \leq 80$$

$$A_i = (A_{i-1} \lll 5) + F_i(B_{i-1}, C_{i-1}, D_{i-1}) + E_{i-1} + W_i + K_i$$



SHA-2

- osmislila NSA
- NIST publicirao 2001 u vrijeme natječaja za SHA-3
- skup funkcija:
 - SHA-224 (veličina bloka na ulazu je 512 bita = 64 bajta)
 - SHA-256 (veličina bloka na ulazu je 512 bita = 64 bajta)
 - SHA-384 (veličina bloka na ulazu je 1024 bita = 128 bajtova)
 - SHA-512 (veličina bloka na ulazu je 1024 bita = 128 bajtova)

Algoritam	Sažetak	Stanje	Blok	Poruka	Arhitektura	Broj rundi	Funkcije
SHA-1	160	160	512	$2^{64} - 1$	32	80	+ , and, or, xor, rot
SHA-256/224	256/224	256	512	$2^{64} - 1$	32	64	+ , and, or, xor, shift, rot
SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80	+ , and, or, xor, shift, rot

SHA-2

- <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>
- zadnji blok teksta se nadopunjuje do 512 bitova na isti način kao i SHA-1
- poruka se podijeli na blokove od po 512 bita:

$$M^{(1)}, M^{(2)}, \dots, M^{(N)}$$

- svaki blok se dijeli na 16 podblokova po 32 bita :

$$M_0, M_1, M_2, \dots, M_{15}$$

- $H^{(0)} = a_0 b_0 c_0 d_0 e_0 f_0 g_0 h_0$

$$a_0 = 6a09e667$$

$$e_0 = 510e527f$$

$$b_0 = bb67ae85$$

$$f_0 = 9b05688c$$

$$c_0 = 3c6ef372$$

$$g_0 = 1f83d9ab$$

$$d_0 = a54ff53a$$

$$h_0 = 5be0cd19$$

Funkcije i konstante algoritma SHA-2

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$S_0(x) = \text{ROTR}^2(x) \oplus \text{ROTR}^{13}(x) \oplus \text{ROTR}^{22}(x)$$

$$S_1(x) = \text{ROTR}^6(x) \oplus \text{ROTR}^{11}(x) \oplus \text{ROTR}^{22}(x)$$

$$F_0(x) = \text{ROTR}^7(x) \oplus \text{ROTR}^{18}(x) \oplus \text{SHR}^3(x)$$

$$F_1(x) = \text{ROTR}^{17}(x) \oplus \text{ROTR}^{19}(x) \oplus \text{SHR}^{10}(x)$$

$$K_t = 5a827999, \quad \text{za } 0 \leq t \leq 15$$

$$K_t = 6ed9eba1, \quad \text{za } 16 \leq t \leq 31$$

$$K_t = 8f1bbcd\bar{c}, \quad \text{za } 32 \leq t \leq 47$$

$$K_t = ca62c1d6, \quad \text{za } 48 \leq t \leq 64$$

- koristi se zbrajanje po modulu 2^{32}

SHA-2

za $i=1$ do N , tj. za svaki od N blokova računaj
Priprema (izračunavanje W_t)

$$W_t = M_t^{(i)}, \quad 0 \leq t \leq 15$$

$$W_t = F_1(W_{t-2}) + W_{t-7} + F_0(W_{t-15}) + W_{t-16}, \quad 16 \leq t \leq 63$$

Postavljanje početnih vrijednosti iz prošlog kruga
ili postavljanje konstanti ako se radi o prvom krugu

$$a = H_0^{(i-1)} \quad b = H_1^{(i-1)} \quad c = H_2^{(i-1)} \quad \dots \quad h = H_7^{(i-1)}$$

za $t=0$ do 63 radi // u 64 koraka

Računaj a, b, c, d, e, f, g, h

$$H_0^{(i)} = a + H_0^{(i-1)}$$

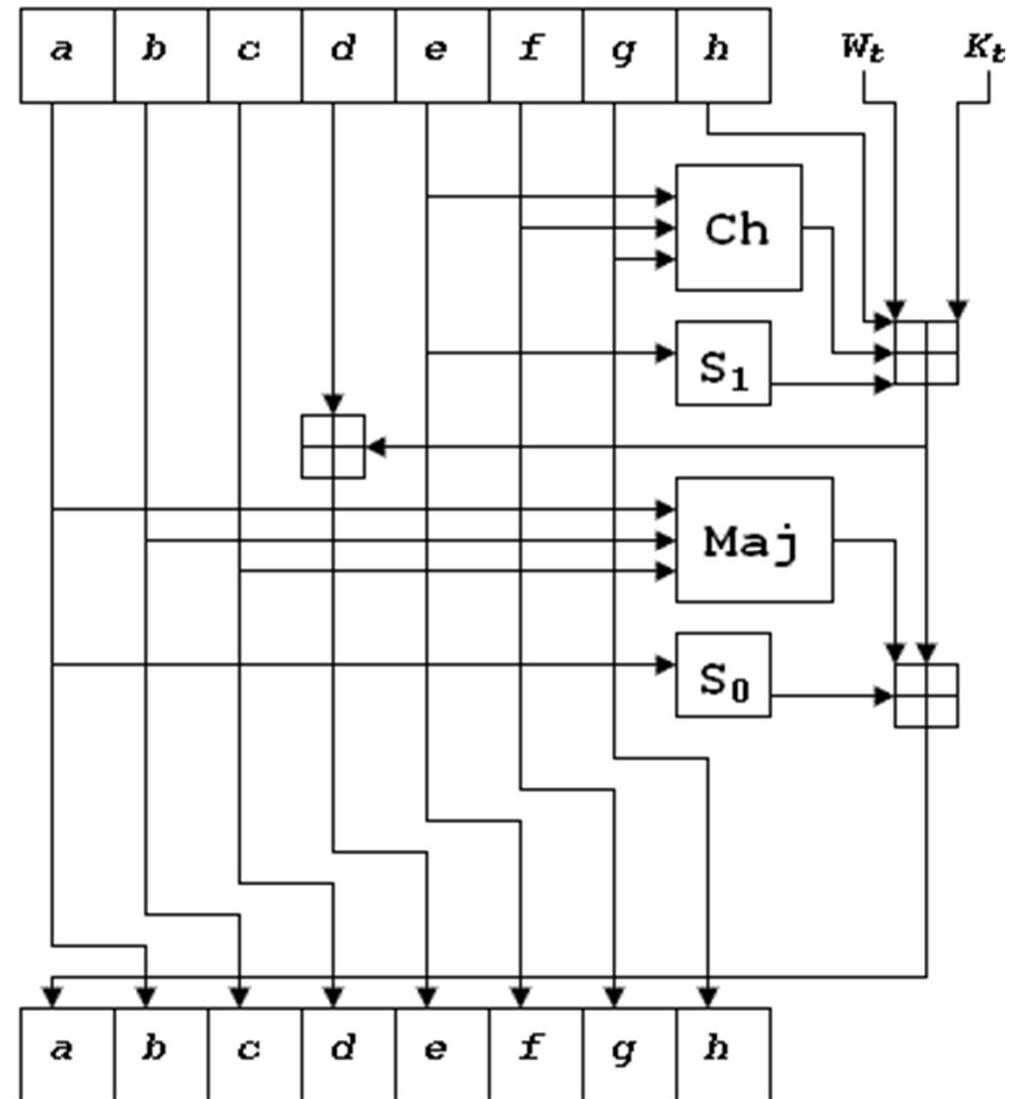
$$H_1^{(i)} = b + H_1^{(i-1)}$$

\dots

$$H_7^{(i)} = h + H_7^{(i-1)}$$

SHA-2

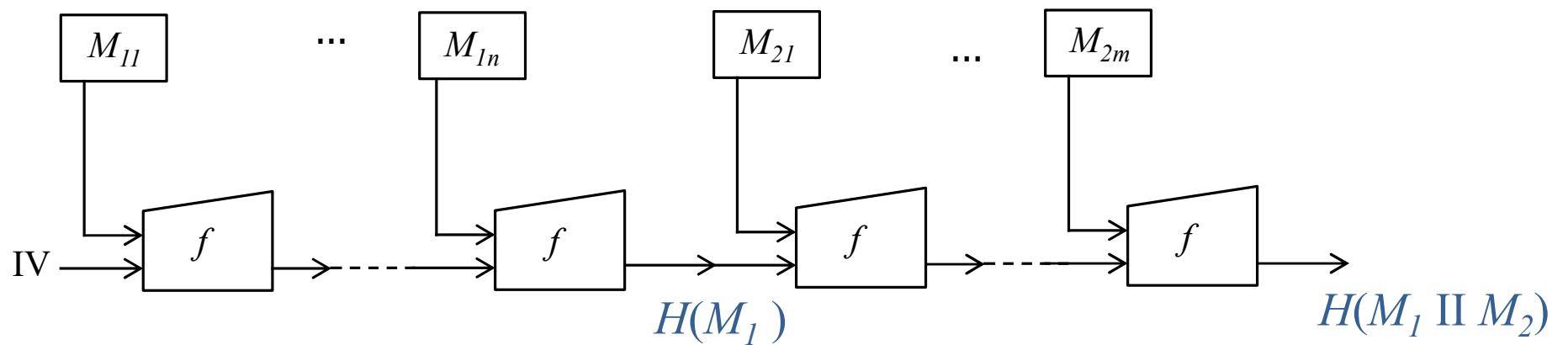
Računaj a, b, c, d, e, f, g, h :



Napad na tajni sufiks poruke

(engl. *Length Extension Attack* odnosno *Attack Against Secret Suffix*)

- algoritmi zasnovani na konstrukciji Merkle–Damgård su osjetljivi na tu vrstu napada
- napadač na temelju poznatog sažetka $H(M_1)$ i duljine poruke M_1 , a bez da poznaje poruku M_1 , može umetnuti dodatne podatke na kraj poruke M_1 , tj. može dodati proizvoljnu dodatnu poruku M_2 i izračunati $H(M_1 \text{ II } M_2)$



SHA-3

- 2.11.2007. – NIST raspisuje natječaj za SHA-3
- informacije o natječaju su dostupne na <http://csrc.nist.gov/groups/ST/hash/sha-3/index.html>
- konačni izbor 2.10.2012. godine
- do 31.10.2008. zabilježeno je 64 prijava:

Abacus	ARIRANG	AURORA	BLAKE	Blender	BMW	BOOLE	Cheetah
CHI	CRUNCH	CubeHash	DCH	Dynamic SHA	Dynamic SHA2	ECHO	ECOH
ENDO-R	EnRUPT	ESSENCE	FSB	Fugue	Groestl	Hamsi	HASH 2x
JH	Keccak	Khichidi-1	LANE	Lesamnta	Luffa	LUX	Maraca
MCSSHA-3	MD6	MeshHash	NaSHA	NKS 2D	Ponic	SANDstorm	Sarmal
Sgail	Shabal	SHAMATA	SHAvite-3	SIMD	Skein	Spectral Hash	StreamHash
SwiFFTX	Tangle	TIB3	Twister	Vortex	Wamm	Waterfall	ZK-Crypt
?	?	?	?	?	?	?	?

SHA-3

- 24.6.2009. objavljena je lista od 14 kandidata za drugi krug:

- ◆ BLAKE
- ◆ BMW - Blue Midnight Wish
- ◆ CubeHash ([Bernstein](#))
- ◆ ECHO (France Telecom)
- ◆ Fugue (IBM)
- ◆ Groestl ([Knudsen](#))
- ◆ Hamsi
- ◆ JH
- ◆ Keccak ([Daemen](#))
- ◆ Luffa
- ◆ Shabal
- ◆ SHAvite-3
- ◆ SIMD
- ◆ Skein ([Schneier](#))

	ARIRANG		BLAKE		BMW		Cheetah
CHI	CRUNCH	CubeHash			Dynamic SHA2	ECHO	
		ESSENCE	FSB	Fugue	Groestl	Hamsi	
JH	Keccak		LANE	Lesamnta	Luffa		
	MD6					SANDstorm	
	Shabal		SHAvite-3	SIMD	Skein		
SwiFFTX							

SHA-3

- 9.12.2010. objavljen je popis 5 finalista
- 2.10.2012. proglašen pobjednik

- ◆ BLAKE
- ◆ Groestl (Knudsen)
- ◆ JH
- ◆ Keccak (Daemen)
- ◆ Skein (Schneier)

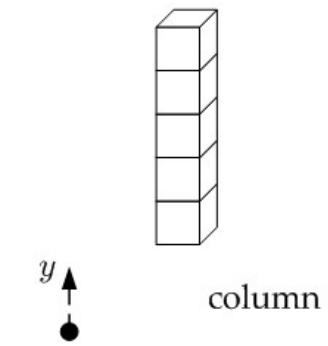
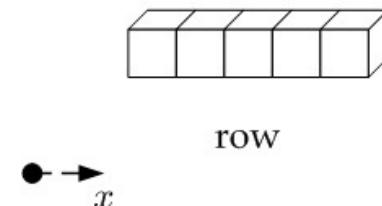
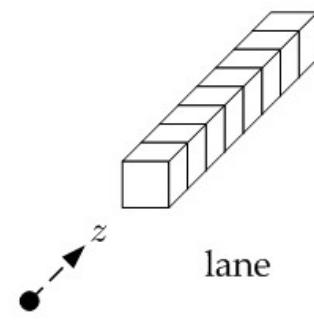
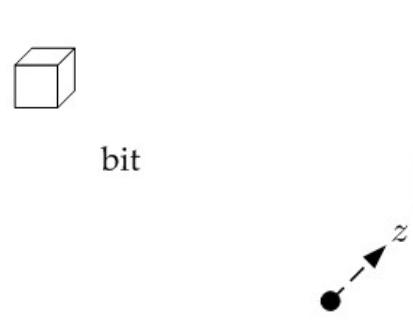
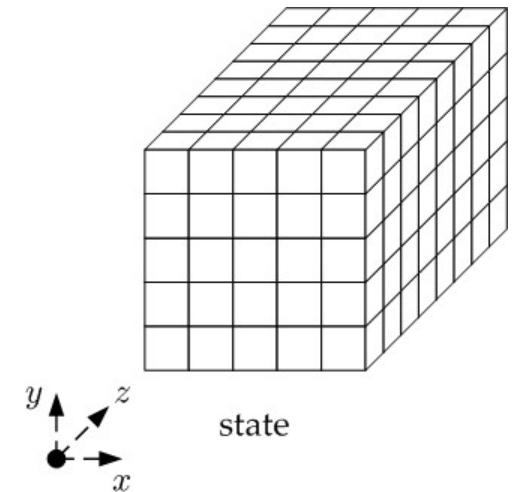
			BLAKE		BMW		
		CubeHash				ECHO	
				Fugue	Groestl	Hamsi	
JH	Keccak				Luffa		
	Shabal		SHAvite-3	SIMD	Skein		

SHA-3

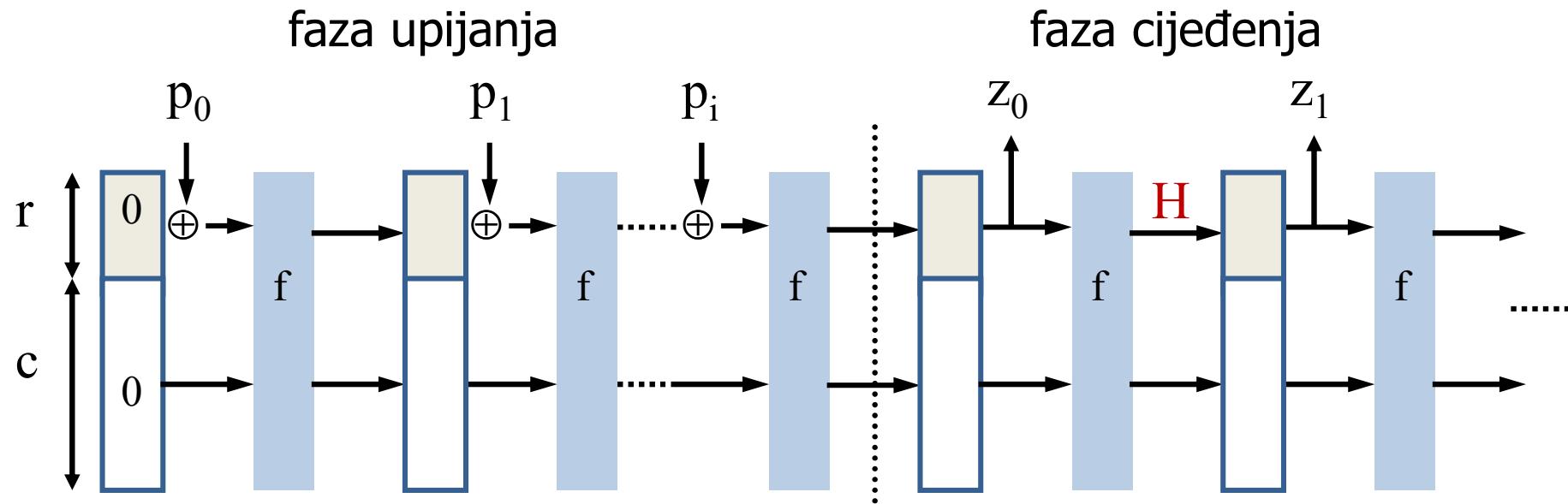
- autori su Guido Bertoni, Joan Daemen (autor AES-a), Michaël Peeters, and Gilles Van Assche
- značajno brži od ostalih finalista
- sažeci su jednake duljine kao i kod SHA-2, ali se veličine ulaznih blokova razlikuju:
 - SHA3-224 (veličina ulaznog bloka 1152 bitova)
 - SHA3-256 (veličina ulaznog bloka 1088 bitova)
 - SHA3-384 (veličina ulaznog bloka 832 bitova)
 - SHA3-512 (veličina ulaznog bloka 576 bitova)
- nadopunjavanje zadnjeg bloka teksta (*padding*) je izmijenjeno i obavlja se prema shemi $M \parallel 10^*1$
 - SHA-2: $M \parallel 10^* \parallel$ 64-bitna za duljinu poruke
 - izvorni prijedlog autora algoritma Keccak: $M \parallel 10^*1000000$

SHA-3: stanje, bit, traka, redak i stupac

- $X = Y = 5$
- duljina trake $Z = w \in \{1, 2, 4, 8, 16, 32, 64\}$
- w je duljina CPU riječi
- Keccak- $f[b]$ gdje je b broj bitova stanja $b = 25w$
 $b \in \{25, 50, 100, 200, 400, 800, 1600\}$
- slike su preuzete sa <http://keccak.noekeon.org/>



Spužvasta konstrukcija algoritma SHA-3



- **25w = c + r = 1600** za 64-bitne riječi ili 800 za 32-bitne riječi, itd.
- kapacitet $c = 2 \times$ veličina sažetka i **veličina bloka** (ostatak) $r = 25w - c$
 - SHA3-224: $c = 448$, $r = 800 - 448 = 352$ bitova = 44 bajta
 - SHA3-256: $c = 512$, $r = 800 - 512 = 288$ bitova = 36 bajta
 - SHA3-384: $c = 768$, $r = 832$ bitova = 104 bajta
 - SHA3-512: $c = 1024$, $r = 576$ bitova = 72 bajta

SHA-3: funkcija f

- obavlja se u n_r koraka: $n_r = 12+2l$, gdje je $2^l = w$
- za $w = 64 = 2^6$, $n_r = 24$ koraka

Keccak- $f[b](A)$

```
za i 0 do nr-1
    A = Round[b](A, RC[i])
return A
```

- funkcija f se sastoji od poziva pet osnovnih funkcija koje manipuliraju s bitovima *stanja*:

θ (*theta*)

ρ (*rho*)

π (*pi*)

χ (*chi*)

ι (*iota*)

SHA-3: funkcija f

```
Keccak-f[b] (A) {
    za i 0 do nr-1
        // (x, y) ∈ {0...4, 0...4}
        // funkcija θ
        C[x] = A[x, 0] xor A[x, 1] xor A[x, 2] xor
                A[x, 3] xor A[x, 4];
        D[x] = C[x-1] xor rot(C[x+1], 1);
        A[x, y] = A[x, y] xor D[x];

        // funkcije ρ i π
        B[y, 2*x+3*y] = rot(A[x, y], r[x, y]);

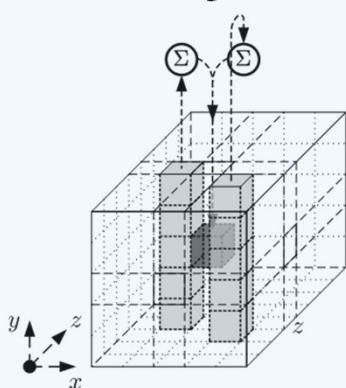
        // funkcija χ
        A[x, y] = B[x, y] xor ((not B[x+1, y]) and
                B[x+2, y]);
}

// funkcija ℓ
A[0, 0] = A[0, 0] xor RC
return A;
```

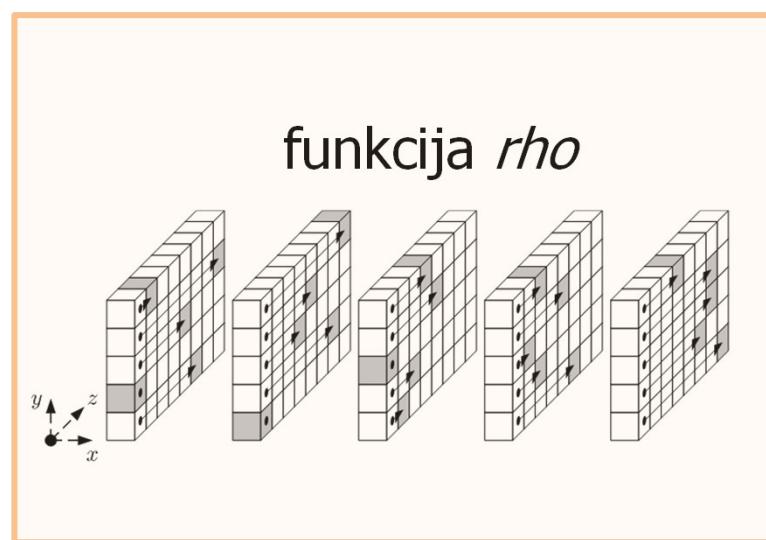
Funkcije θ (*theta*), ρ (*rho*), π (*pi*), χ (*chi*) i ι (*iota*)

- manipuliraju bitovima stanja

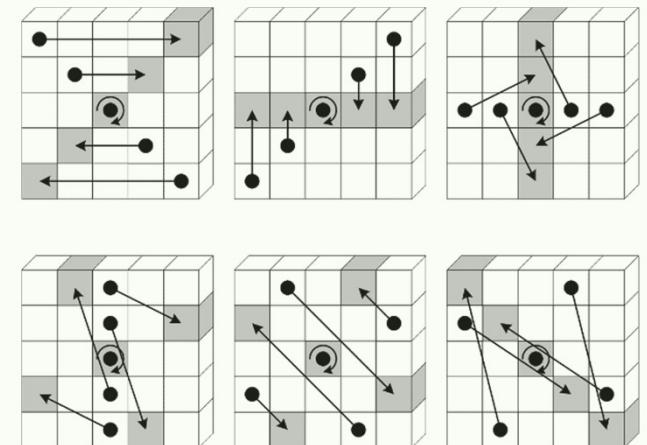
funkcija *theta*



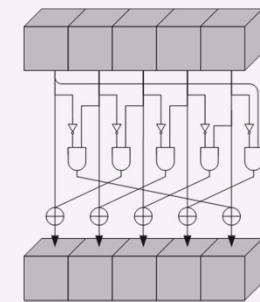
funkcija *rho*



funkcija *pi*



funkcija *chi*

$$\begin{aligned} a[i][j][k] &\oplus= \\ \neg a[i][j+1][k] \\ \& \& a[i][j+2][k] \end{aligned}$$


funkcija *iota*

$$\begin{aligned} A[0,0] &\oplus= RC[i] \\ RC[i] &- konstante \end{aligned}$$

Napadi na funkciju sažimanja SHA

- 1993. – objavljen SHA-0
- 1995. – NSA je predložila SHA-1 kao zamjenu za SHA-0
- 1998. – objavljen uspješan napad na SHA-0, ali ne i na SHA-1
- 2001. – NSA predlaže SHA-2
- 2005. – uspješan napad na SHA-1
- 2007. – NIST raspisuje natječaj za SHA-3 i preporuča SHA-2
- 2012. – proglašen pobjednik natječaja SHA-3: Keccak
- 2017. – uspješan napad na SHA-1:
 - dva različita PDF dokumenta daju isti sažetak (Marc Stevens ispred svih u suradnji s tvrtkom Google)
 - <https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html>
- za pronađak kolizije potrebno je:
 - MD5 → 1 pametni telefon i 30 s
 - SHA-1 → grubom silom i 12 000 000 GPU godina
 - SHA-1 → algoritam Shattered i 110 GPU godina



Primjer napada na *hash* funkcije

Rođendanski napad

- engl. *birthday attack*
- vjerojatnost da dvije osobe u dvorani u kojoj je ukupno $k=1.2 \cdot 365^{1/2} = 23$ ljudi imaju isti dan rođendan je veća od 50%
- analogno:
vjerojatnost da dvije poruke iz skupa od $k=1.2 \cdot (2^n)^{1/2} = 1.2 \cdot 2^{n/2}$ poruka daju isti sažetak je veća od 50%, gdje je n duljina sažetka u bitovima

Primjer napada

M1.txt

```
00000000 d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
00000010 2f ca b5 87 12 46 7e ab 40 04 58 3e b8 fb 7f 89
00000020 55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 71 41 5a
00000030 08 51 25 e8 f7 cd c9 9f d9 1d bd f2 80 37 3c 5b
00000040 96 0b 1d d1 dc 41 7b 9c e4 d8 97 f4 5a 65 55 d5
00000050 35 73 9a c7 f0 eb fd 0c 30 29 f1 66 d1 09 b1 8f
00000060 75 27 7f 79 30 d5 5c eb 22 e8 ad ba 79 cc 15 5c
00000070 ed 74 cb dd 5f c5 d3 6d b1 9b 0a d8 35 cc a7 e3
```

M2.txt

```
00000000 d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
00000010 2f ca b5 07 12 46 7e ab 40 04 58 3e b8 fb 7f 89
00000020 55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 f1 41 5a
00000030 08 51 25 e8 f7 cd c9 9f d9 1d bd 72 80 37 3c 5b
00000040 96 0b 1d d1 dc 41 7b 9c e4 d8 97 f4 5a 65 55 d5
00000050 35 73 9a 47 f0 eb fd 0c 30 29 f1 66 d1 09 b1 8f
00000060 75 27 7f 79 30 d5 5c eb 22 e8 ad ba 79 4c 15 5c
00000070 ed 74 cb dd 5f c5 d3 6d b1 9b 0a 58 35 cc a7 e3
```

MD5 Sum (M1.txt) = **a4c0d35c95a63a805915367dcfe6b751**

MD5 Sum (M2.txt) = **a4c0d35c95a63a805915367dcfe6b751**

Primjer napada

M1.txt

00000000	d1	31	dd	02	c5	e6	ee	c4	69	3d	9a	06	98	af	f9	5c
00000010	2f	ca	b5	87	12	46	7e	ab	40	04	58	3e	b8	fb	7f	89
00000020	55	ad	34	06	09	f4	b3	02	83	e4	88	83	25	71	41	5a
00000030	08	51	25	e8	f7	cd	c9	9f	d9	1d	bd	f2	80	37	3c	5b
00000040	96	0b	1d	d1	dc	41	7b	9c	e4	d8	97	f4	5a	65	55	d5
00000050	35	73	9a	c7	f0	eb	fd	0c	30	29	f1	66	d1	09	b1	8f
00000060	75	27	7f	79	30	d5	5c	eb	22	e8	ad	ba	79	cc	15	5c
00000070	ed	74	cb	dd	5f	c5	d3	6d	b1	9b	0a	d8	35	cc	a7	e3

M2.txt

00000000	d1	31	dd	02	c5	e6	ee	c4	69	3d	9a	06	98	af	f9	5c
00000010	2f	ca	b5	07	12	46	7e	ab	40	04	58	3e	b8	fb	7f	89
00000020	55	ad	34	06	09	f4	b3	02	83	e4	88	83	25	f1	41	5a
00000030	08	51	25	e8	f7	cd	c9	9f	d9	1d	bd	72	80	37	3c	5b
00000040	96	0b	1d	d1	dc	41	7b	9c	e4	d8	97	f4	5a	65	55	d5
00000050	35	73	9a	47	f0	eb	fd	0c	30	29	f1	66	d1	09	b1	8f
00000060	75	27	7f	79	30	d5	5c	eb	22	e8	ad	ba	79	4c	15	5c
00000070	ed	74	cb	dd	5f	c5	d3	6d	b1	9b	0a	58	35	cc	a7	e3

MD5 Sum (M1.txt) = **a4c0d35c95a63a805915367dcfe6b751**

MD5 Sum (M2.txt) = **a4c0d35c95a63a805915367dcfe6b751**

Digitalni certifikat

FER-ov digitalni certifikat od 2008 do 2018

```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 75 (0x4b)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=US, ST=WA, L=Seattle, O=Thawte Consulting cc,
             OU=Certification Services Division,
             CN=Thawte Server CA/emailAddress=certs@thawte.com
Validity
    Not Before: May 13 23:33:08 2008 GMT
    Not After : Dec 31 23:59:59 2020 GMT
Subject: C=HR, ST=Grad Zagreb, L=Zagreb, O=FER, OU=CIP,
          CN=webmail.fer.hr/emailAddress=korisnik@webmail.fer.hr
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
        Modulus (2048 bit):
            00:cd:66:28:fb:b8:b3:b7:e0:72:77:48:2d:08:04:
            e1:6d:1c:c5:4f:57:73:0c:e6:db:3b:8e:cd:c6:25:
            61:7f:60:c9:da:a3:9f:1d:fa:d8:ef:00:7b:f9:54:
            65:ab:7e:9e:9b:6d:ff:d4:12:ad:f8:ac:87:6e:83:
            ec:65:5f:b4:2d:eb:b8:dc:1c:d7:32:b7:46:a5:e3:
            a1:6c:0b:4c:1b:0c:89:0a:fb:0e:3a:c0:0f:af:b2:
            62:1d:2f:60:e4:b1:27:b4:7c:59:00:2c:19:e9:f3:
            a3:88:fe:01:d6:56:be:26:c7:f8:42:b1:79:39:98:
            a1:b4:4a:84:dd:20:ca:e7:a9:db:6d:a6:73:88:e7:
            81:8b:3e:81:3d:00:e5:5d:7f:3d:9b:cd:ba:9b:28:
            88:88:7f:d7:69:2c:66:eb:8f:79:b8:ec:bc:bb:76:
            67:b1:00:2a:70:bd:f1:21:66:6f:ba:74:81:82:30:
            02:c0:a8:57:f8:9f:76:02:df:7f:49:44:4a:32:93:
            48:a4:25:73:47:10:21:20:fe:b6:d2:09:1a:60:4f:
            a5:d9:df:ea:55:49:43:c6:ce:96:0b:7d:a7:22:c1:
            3e:5b:28:2e:2c:04:7a:b2:93:89:db:d8:2b:59:86:
            a3:0a:c1:6f:f9:56:b2:a5:71:4c:4b:74:f3:b8:a1:
            b4:65
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Basic Constraints: critical
        CA:TRUE
Signature Algorithm: md5WithRSAEncryption
07:fa:4c:69:5c:fb:95:cc:46:ee:85:83:4d:21:30:8e:ca:d9:
a8:6f:49:1a:e6:da:51:e3:60:70:6c:84:61:11:a1:1a:c8:48:
e7:20:1b:8b:ca:a4:ab:8d:e9:51:d9:e2:4c:2c:59:a9:da:b9:
b2:75:1b:f6:42:f2:ef:c7:f2:18:f9:89:bc:a3:ff:8a:23:2e:
70:47
```

Primjer gdje bi se teoretski napad mogao primijeniti

Aktualan FER-ov digitalni certifikat

Pristup web stranicama FER-a:
*The connection to this site is
encrypted and authenticated
using TLS 1.2, ECDHE_RSA with
P-256, and AES_128_GCM.*

```
Certificate:
  Data: Version: 3 (0x2)
        Serial Number: 0a:2f:ab:75:d4:a1:ee:f5:ea:df:74:15:aa:fd:47:c4
  Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
  Validity Not Before: May 13 00:00:00 2018 GMT
                Not After : May 20 12:00:00 2020 GMT
  Subject: C=HR, L=Zagreb, O=Sveu\xC4\x8Dili\xC5\xAlte u Zagrebu, OU=CIP,
            CN=*.fer.unizg.hr
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
        Modulus:
          00:c6:bb:ca:00:b5:40:96:b3:6b:2e:94:7e:43:77:
          39:06:d2:4f:11:c0:c4:17:e5:eb:d6:10:a5:2c:fa:
          4c:f1:50:35:59:59:2b:fa:b5:22:26:3f:0a:ff:f2:
          f9:c4:d7:e2:67:5d:bf:b5:c1:cc:6b:77:31:e9:de:
          95:b0:76:53:47:f7:1f:fe:c4:5b:c1:a7:fd:c4:fc:
          61:d3:ea:b5:28:48:e5:d5:96:a0:11:ed:0b:00:a2:
          42:c9:fa:94:26:89:f5:37:db:0a:9a:f8:95:e8:a6:
          35:8a:68:33:90:c2:22:10:ad:65:3a:95:5f:64:1f:
          6f:43:88:b2:1c:f8:29:9e:51:6b:e4:2d:8c:3e:39:
          90:f7:31:8e:32:f8:0f:cf:3e:b4:7a:c6:f3:27:17:
          a3:4e:3c:7c:27:07:3d:68:fc:5e:9c:87:86:74:ea:
          22:32:d5:aa:93:e4:d4:78:23:d2:88:0f:e3:8f:05:
          8c:54:b8:95:29:eb:c2:0a:fc:26:20:ca:52:ff:ce:
          75:6b:29:82:d6:67:06:0b:49:53:37:0d:7e:cf:1c:
          7e:88:90:8d:7a:e7:99:fc:9f:d7:5c:e2:1f:73:19:
          cc:27:ba:31:6f:82:40:b0:cb:8a:d2:95:f4:6e:72:
          78:b6:02:f5:f4:0b:b6:60:32:fb:3f:34:66:f2:a4:
          12:c5
        Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Authority Key Identifier:
      keyid:67:FD:88:20:14:27:98:C7:09:D2:25:19:BB:E9:51:11:63:75:50:62
      URI:http://crl3.digicert.com/TERENASSLCA3.crl
  ...
  Signature Algorithm: sha256WithRSAEncryption
  a3:aa:9b:c3:04:c3:5c:64:32:9c:8f:08:31:89:15:8a:52:19:
  fb:02:e9:dd:ab:59:3e:9e:d8:b8:52:b2:8d:df:5a:29:dc:2b:
  c0:01:7d:96:87:5c:a7:01:7e:26:c9:3b:be:01:d3:9c:71:62:
  e3:e5:a2:ce:5d:ee:59:b5:ed:20:d8:80:27:ac:af:f5:6a:73:
  79:35:d2:c5
```

Napad tablicama s unaprijed izračunatim sažecima

- *engl. rainbow table*
- za najčešće korištene zaporce se unaprijed izračunaju sažeci
- zapisi u datoteci sa zaštićenim lozinkama se uspoređuju s unaprijed izračunatim sažecima
- 7 najčešće korištenih zaporki:

	2018.	2021.	2023.	2025.
1.	123456	123456	123456	123456
2.	password	123456789	password	111111
3.	123456789	12345	123456789	admin
4.	12345678	qwerty	12345	qwerty
5.	12345	password	12345678	password
6.	111111	12345678	qwerty	123456789
7.	1234567	111111	1234567	123123

Zaključak o funkcijama sažimanja

- kolizije su bezopasne sve dok izgledaju kao slučajan niz
- međutim, gubi se povjerenje u certifikate
 - protokoli koji koriste sažetak slučajnog simetričnog ključa nisu više sigurni
- problem nevidljivih podataka u Word dokumentu ili slučajnih nizova u slikama
- rješenje:
 - koristiti obične tekstualne datoteke ili potpisati sažetu datoteku (kao što PGP koristi *zip*)
 - koristiti novi algoritam sažimanja SHA-3

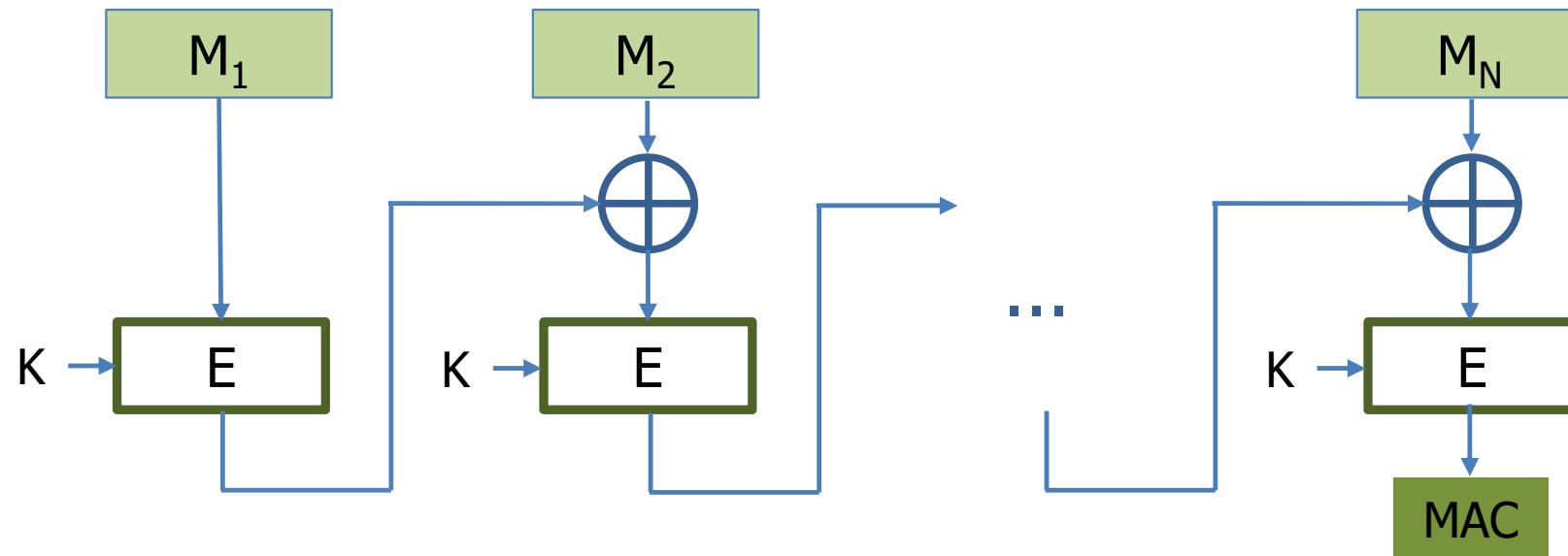
4. Autentifikacijsko kriptiranje

Autentifikacija poruka

- autentifikacijom korisnika bavimo se detaljno na predmetu Napredni operacijski sustavi
- digitalni potpis – time ćemo se pozabaviti kasnije na ovom predmetu
- postupak kriptiranja koji uključuje i autentifikaciju (*Authenticated Encryption, AE*) i osim tajnosti osigurava
 - integritet, odnosno izvornost (autentičnost) poruke
 - autentifikaciju pošiljatelja
- dodatak poruci MAC (*Message Authentication Code*) za razliku od digitalnog potpisa **ne koristi asimetričnu**, već samo simetričnu kriptografiju
 - ulaz u algoritam je uz poruku tajni ključ
 - Kako je osigurana autentičnost?
 - Poruku je poslao onaj tko ima tajni ključ.

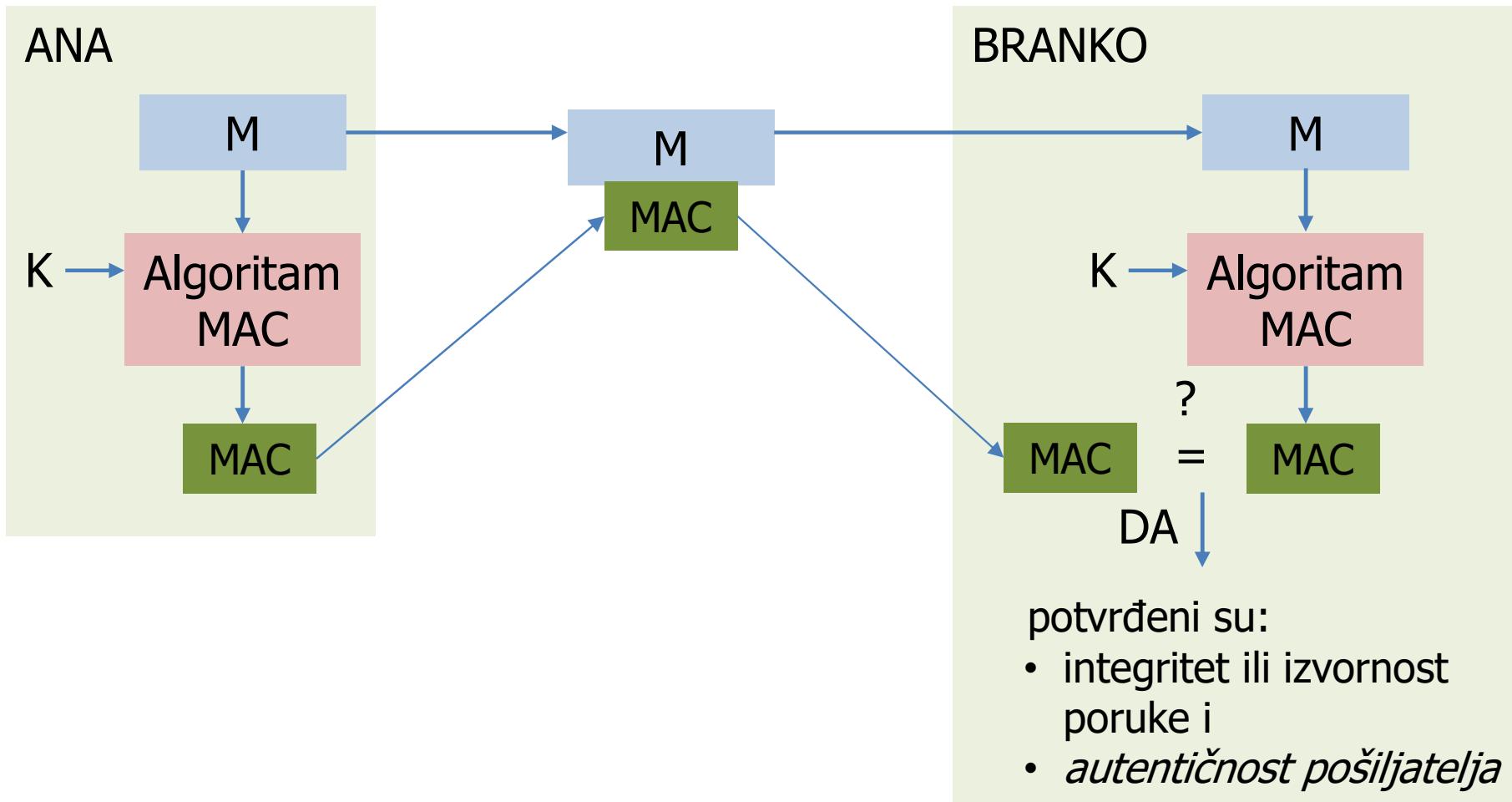
Algoritam MAC

- *Message Authentication Code*
- u CBC načinu rada naziva se CBC-MAC



- varijante:
 - One-key ili OMAC, PMAC, HMAC ...

Primjer kako se može koristiti dodatak poruci MAC



Algoritam HMAC

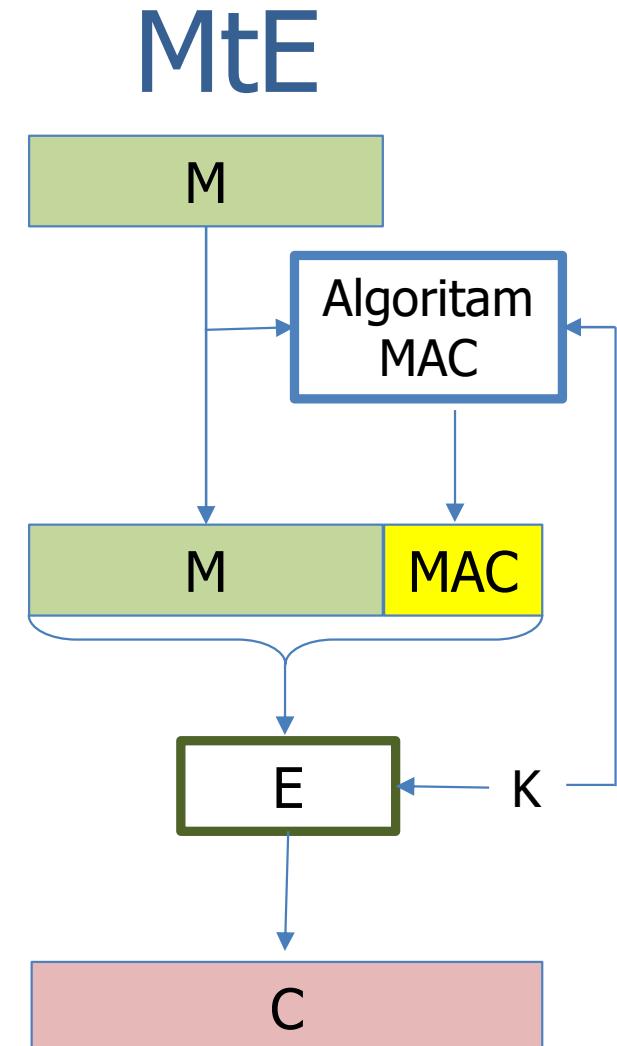
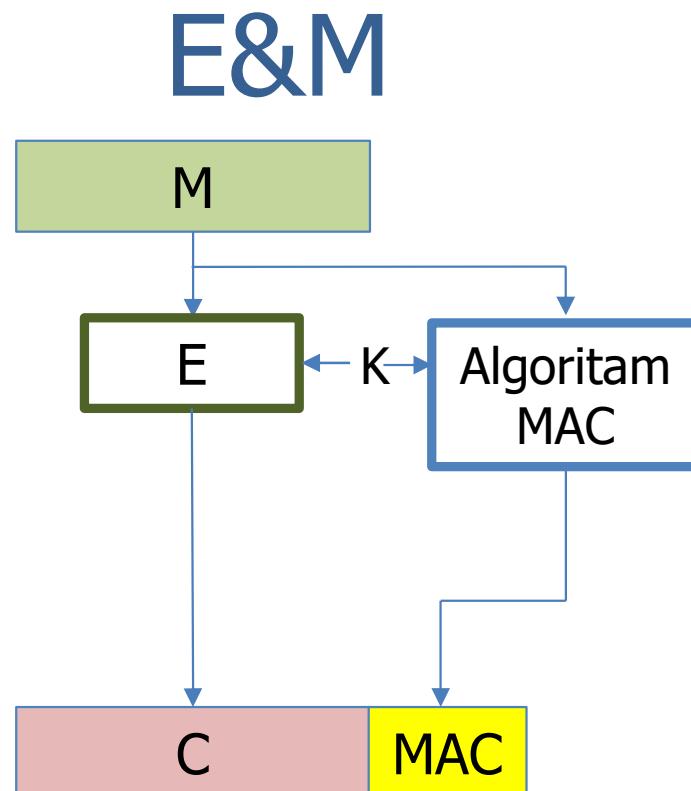
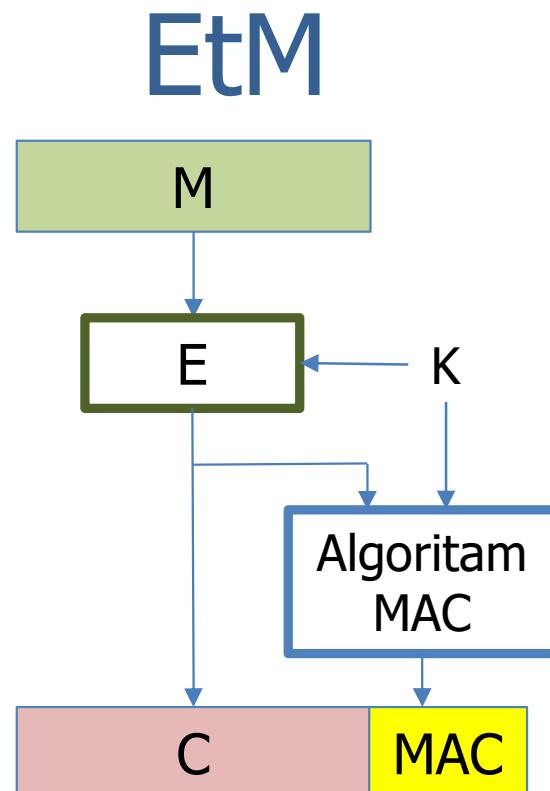
- umjesto blok simetričnog algoritma koristi funkciju za izračunavanje sažetka poruke
- *keyed-Hash Message Authentication Code*
 - HMAC_MD5
 - HMAC_SHA1
 - HMAC_SHA256
 - HMAC_SHA3
- $\text{HMAC}(K, M) = \text{H}\{(\text{K}' \oplus \text{opad}) \parallel \text{H}[(\text{K}' \oplus \text{ipad}) \parallel M]\}$
 - $\text{K}' = \text{H}(\text{K})$ ako je K veći od veličine bloka, inače $\text{K}' = \text{K}$
 - konstanta *opad (outer padding)* = 0x5c5c5c...5c5c
 - konstanta *ipad (inner padding)* = 0x363636...3636
 - *opad* i *ipad* su veličine jednog bloka

Kako osigurati i tajnost?

- ponovimo: dodatak poruci MAC osigurava
 - integritet, odnosno izvornost (autentičnost) poruke
 - autentifikaciju pošiljatelja
 - no, **nedostaje tajnost!**
- tajnost se osigurava u kombinaciji sa simetričnim kriptografskim algoritmima:
 - *Encrypt-then-MAC (EtM)*
 - *Encrypt-and-MAC (E&M)*
 - *MAC-then-Encrypt (MtE)*

Kako uz integritet i autentičnost osigurati i tajnost?

Novi načini kriptiranja:

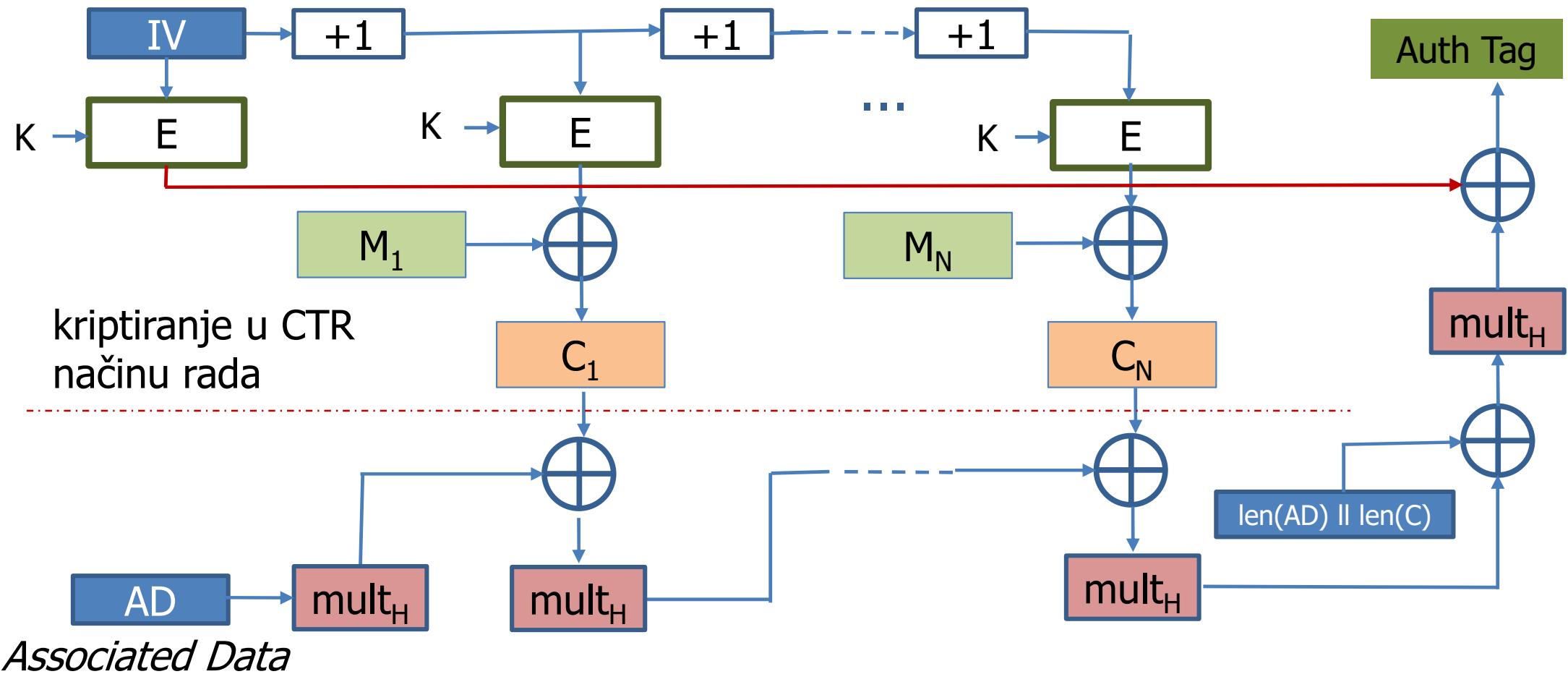


K nije isti ključ za simetričan algoritam i MAC, već se iz jednog ključa K generiraju dva

Način kriptiranja GCM - *Galois/Counter Mode*

- način autentifikacijskog kriptiranja koji je primjenjiv samo za simetrične blok algoritme s veličninom bloka 128 bita
- varijanta *Galois Message Authentication Code, GMAC* – samo za autentifikaciju
- ulaz:
 - jasni tekst
 - tajni ključ K
 - IV
 - povezani autentifikacijski podaci (*Associated Data, AD*)
 - duljina povezanih podataka i duljina kriptiranog teksta
- izlaz:
 - autentifikacijska značka (*Auth Tag*)

Način kriptiranja GCM - *Galois/Counter Mode*



Neporecivost

- ostvaruje se uz pomoć asimetrične kriptografije i to kriptiranjem privatnim ključem
 - time se ostvaruje i autentičnost
- samo autentičnost se može ostvariti i bez asimetrične kriptografije
 - MAC
 - autentifikacijsko kriptiranje
 - međutim, time NIJE ostvarena neporecivost

Kako osigurati autentifikaciju?

- asimetrična kriptografija zajedno s funkcijom sažimanja osigurava autentičnost (digitalni potpis) u smislu da autentificira pošiljatelja
- dodatak poruci MAC
 - osigurava integritet i izvornost, ali
 - jamči da je poruku poslao „onaj koji ima tajni ključ“
 - **ne autentificira točno pošiljatelja**
 - ako tajni ključ ima više od dva entiteta (više od dvije osobe)
 - ako tajni ključ imaju samo dvije osobe tada je pošiljatelj jedna od te dvije osobe i primatelj ga na taj način autentificira

Kako osigurati autentifikaciju bez asimetrične kriptografije?

- Kako se može osigurati autentičnost samo sa simetričnim algoritmom kriptiranja i funkcijom sažimanja ili MAC-om?
- ideja 1: uz pomoć **dijeljene tajne**, tj. tajnog ključa
 - iz tajnog ključa K kojeg su Ana i Branko razmijenili izračunaju se dva ključa K_1 i K_2
 - svakim se ključem osigurava sigurna komunikacija, ali samo u jednom smjeru:
 - Ana kriptira poruku ključem K_1 i šalje ju zajedno s dodatkom poruci MAC Branku, a Brankove poruke dekriptira i provjerava dodatak poruci MAC ključem K_2
 - Branko kriptira poruku ključem K_2 i šalje ju zajedno s dodatkom poruci Ani, a Anine poruke dekriptira i provjerava dodatak poruci MAC ključem K_1
 - samo Ana i Branko znaju ključeve i K_1 i K_2 i jedino su oni mogli kriptirati poruke tim ključevima

Kako osigurati autentifikaciju?

- ideja 2: dodavanjem autentifikacijskih podataka u jasni tekst
 - u jasni tekst (koji se kriptira) dodaju se autentifikacijski podaci, npr. „Ana šalje poruku Branku“
- ideja 3: objedinjavanjem u jedan algoritam kojemu je ulaz uz tajni ključ i poruku i dodatni povezani podaci (engl. *associated data, AD*)
 - povezani podaci nisu tajni, ali je osiguran njihov integritet
 - to je zapravo ostvarena ideja 2 u jednom algoritmu
 - takva se kriptografija naziva autentifikacijskom kriptografijom s povezanim podacima (engl. *Authenticated encryption with associated data, AEAD*)
 - NE osigurava neporecivost!
 - digitalni potpis osigurava neporecivost

Natječaj CAESAR

i zaključne napomene o autentifikacijskom kriptiranju

- nedostatak klasičnih autentifikacijskih kriptografskih shema poput *EtM*, *E&M* i *MtE* je upravo u primjeni više algoritama
- natječaj CAESAR (*Competition for Authenticated Encryption: Security, Applicability, and Robustness*) završio 20.3.2019. objavljeno **3 pobjednika** u 3 kategorije i **5 rezervna algoritma**
 - **Ascon**, **ACORN**, **AEGIS** (Bart Preneel, ...), OCB, **Deoxys**, COLM, AES-COPA, ELmD
 - 15 algoritama u trećem krugu natječaja, a ispali su:
 - AES-OTR, AEZ, CLOC and SILC, JAMBU, [Katje](#) (Daemen, ...), [Keyak](#) (Daemen, ...), MORUS, NORX, Tiaoxin
- 2018.-2023.g. NIST-ov natječaj za novi algoritam prilagođen okruženju s ograničenim računalnim resursima (*lightweight cryptography*)
 - u uvjetima natječaja je navedeno da algoritam treba osim simetričnog uključivati i autentifikacijsko kriptiranje (*Authenticated Encryption with Associated Data*, AEAD)
 - odabran je algoritam **ASCON**

Pobjednici na natječaju CAESAR

Pobjednici su birani u tri kategorije simetričnih blok algoritama:

1. Algoritmi koji su **najmanje zahtjevni** na računalne resurse (*Lightweight applications - resource constrained environments*)
 - **prvi izbor:** [Ascon \(web\)](#)
 - *drugi izbor:* [ACORN](#)
2. Algoritmi visokih performansi (*High-performance applications*) tj. **najbrži**:
 - **prvi izbor:** [AEGIS-128](#)
 - *drugi izbor:* [OCB](#)
3. Višerazinska sigurnost (*Defense in depth*), tj. **najsigurniji**:
 - **prvi izbor:** [Deoxys-II](#)
 - *drugi izbor:* [COLM](#) ili [AES-COPA](#) ili [ELmD](#)

5. **Napadi na kriptosustave**

Algoritam kriptiranja bloka je *siguran*

- ako je teško na temelju kriptiranog teksta pronaći
 - jasni tekst i/ili
 - ključ
- ... čak i ako napadač:
 - ima na raspolaganju mnogo parova (M, C) gdje je $C = E(M, K)$
 - može kriptirati i dekriptirati, tj. izračunati:
 - $C=E(M, K)$ za proizvoljni M
 - $M=D(C, K)$ za proizvoljni C

Osnovni algoritam kriptoanalyse

Napad grubom silom

- napadač pokušava dekriptirati kriptirani tekst sa svim mogućim ključevima
- neka je poznat M , $C=E(M,K)$
- algoritam radi sljedeće:
 - za svaki mogući ključ K_i
 - ako je $C == E(M,K_i)$ onda ispiši K_i
- takav se algoritam naziva algoritmom **grube sile**



Pretraživanje cijelog prostora rješenja napad *grubom silom*

- najjednostavnija i najsporija vrsta napada
- nije moguće spriječiti ovaj napad
- uspješnost svih napada na kriptosustave mjeri se usporedbom s pretraživanjem cijelog prostora
- Napad koji ima veću složenost od složenosti pretraživanja cijelog prostora smatra se neuspješnim!
- Pretpostavka: napadač ili već ima na raspolaganju čisti tekst ili pretpostavlja da čisti tekst ima neku standardnu strukturu koju je moguće prepoznati.
 - Inače, u slučaju dekriptiranja poruke bez prepoznatljive strukture, napadač nema nikakve šanse da pretraživanjem cijelog prostora sazna koji je pravi ključ.

Napad na kriptosustav AES grubom silom

- duljina ključa = 128 bita

- broj različitih ključeva =

340282366920938463463374607431768211456

- pretpostavke:

- 1 miliardu računala
 - 1 miliardu ključeva po sekundi po računalu

- gotovi smo za 10 tisuća milijardi godina



Primjer *uspješnog* napada na AES

- potpuni AES-128 sa složenošću $2^{126.1}$
- potpuni AES-192 sa složenošću $2^{189.7}$
- potpuni AES-256 sa složenošću $2^{254.4}$

[A. Bogdanov (KU Leuven), D. Khovratovich (MS Research Redmond), C. Rechberger (France Telecom), Biclique Cryptanalysis of the Full AES, ASIACRYPT, 2011.]

Pretraživanje pola prostora rješenja

- može se ostvariti kod mnogih kriptosustava za koje vrijedi simetrija:
$$C = DES(M, K) \quad \text{i} \quad C' = DES(M', K')$$

(X' oznaka za bitovni komplement vrijednosti X)
- fiksno se postavi jedan bit ključa u '0'
- za svaki K se uspoređuje dobiveni kriptirani tekst C'' sa C i C' i ako vrijedi jednakost, radi se o K odnosno K'
- ušteda je vrlo blizu 50%
- vrijedi za DES!
- **zaštita od napada pretraživanjem pola prostora:** koristiti kriptosustav za koji ne vrijedi navedeni tip simetrije ☺

Vrste napada na kriptosustave prema onome što je napadaču poznato

- napad s odabranim čistim tekstom (*chosen-plaintext attack*)
 - napadač posjeduje neograničene količine parova (M,C)
 - primjer s pametnim karticama
- napad s odabranim kriptiranim tekstom (*chosen-ciphertext attack*)
 - napadač posjeduje po svojoj volji odabrani C i pripadni M (također neograničene količine parova)
- napad s poznatim čistim tekstom (*known-plaintext attack*)
 - napadač posjeduje neke parove (M,C)
 - za napad mu treba određena količina parova
- napad s poznatim kriptiranim tekstom (*only-ciphertext attack*)
 - napadač posjeduje samo C a pokušava saznati K i M
 - napadaču je ovaj napad najteže uspješno provesti
- cilj je dozнати тајни клjuč

Napadi na DES

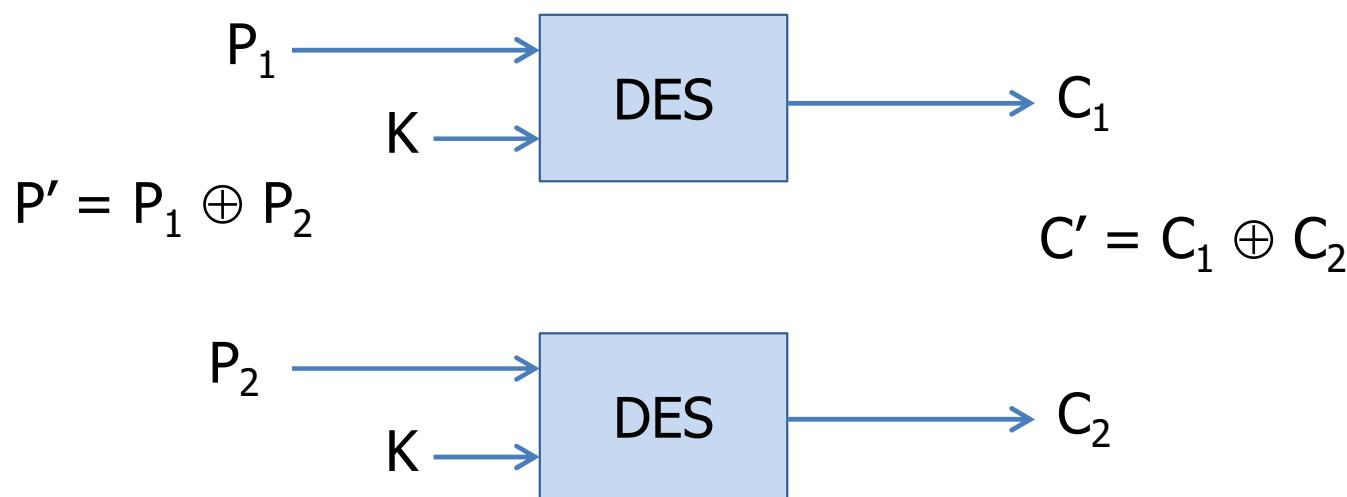
- bilo kakvim linearnim promjenama u postupku generiranja ključeva i u funkciji F, DES ne postaje otporniji na napade
- promjena u nelinearnom dijelu algoritma (S tablice) utječe na ranjivost algoritma
- DES bitno oslabljuje:
 - promjena redoslijeda S tablica
 - slučajno odabrane S tablice
 - umjesto XOR neka složenija funkcija
- pristup: analiza pojednostavljenog kriptosustava (s manje iteracija ili rundi, za primjerice DES sa samo tri runde)

Kriptoanaliza

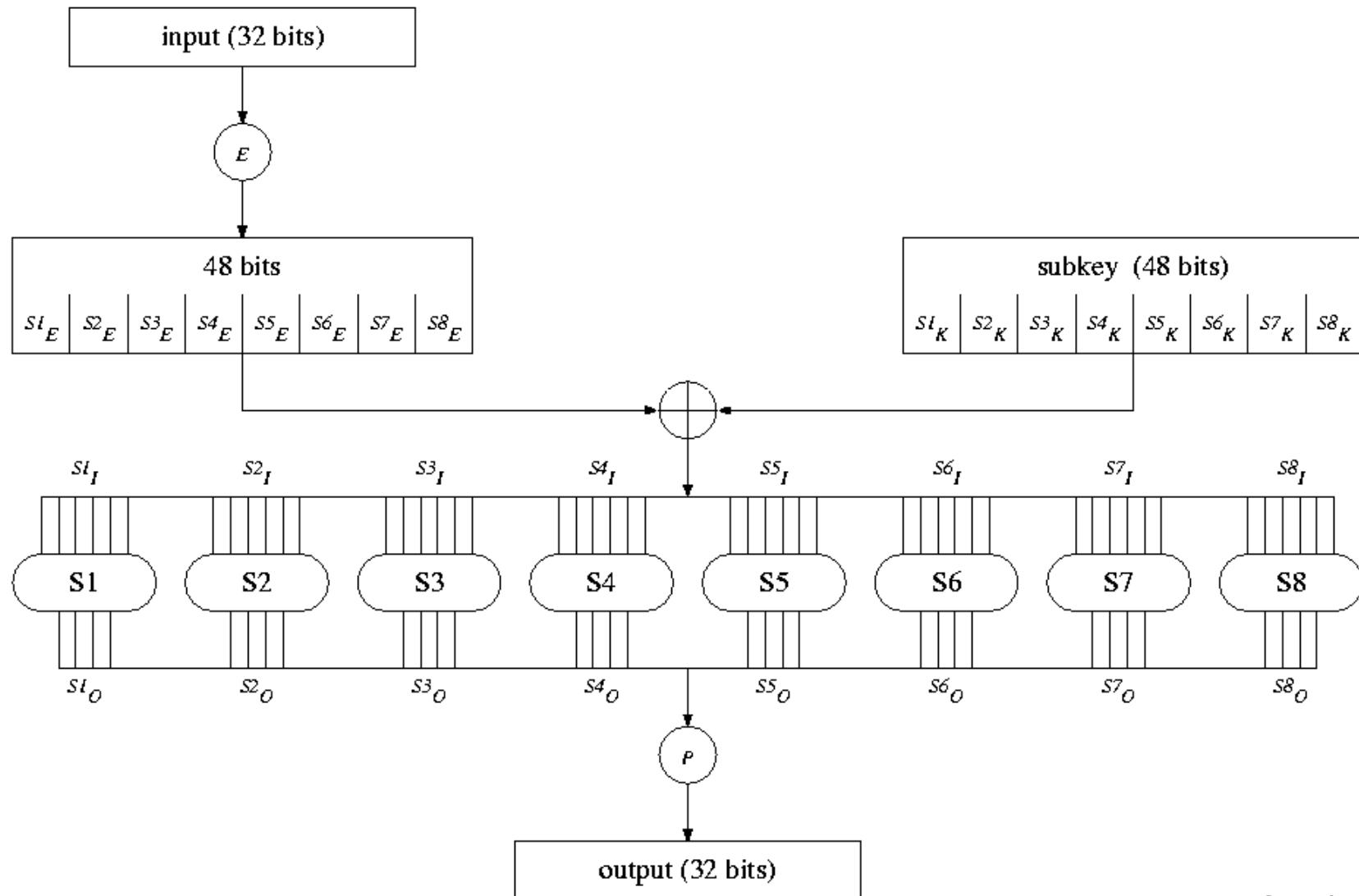
- diferencijalna kriptoanaliza
- linerna kriptoanaliza
- implementacijski napadi
 - napadi koji ne iskorištavaju slabosti algoritma (jer ih obično algoritam ni nema) već iskorištavaju sigurnosne propuste u programskim ili sklopovskim ostvarenjima
- uspješnost kriptoanalyse ocjenjuje se uspoređivanjem s napadom **grubom silom** odnosno ispitivanjem svih mogućih ključeva

Diferencijalna kriptoanaliza kriptosustava DES

- Eli Biham, Adi Shamir, knjiga pod naslovom "*Differential analysis of DES-like cryptosystems*", 1990.
- tehnika kojom se analizira učinak razlike između dva čista teksta na razliku između dva rezultirajuća kriptirana teksta
- razlike služe za određivanje vjerojatnosti mogućih ključeva



Ponavljanje: DES, funkcija F



S-tablice ili S-kutije (*engl. S-boxes*)

- nisu linearne
 - poznavanje razlike ulaznog para ne garantira poznavanje razlike izlaza iz S-tablica
- za bilo koju ulaznu razliku kod S-tablica postoji ogranicen broj mogucih izlaznih razlika
 - primjerice ima i onih koje se sigurno nece pojaviti
- ulaz u neku od 8 S-tablica je velicine 6 bita, a izlaz 4 bita pa stoga postoji
 - $2^6 = 64$ mogucih ulaznih razlika i
 - $2^4 = 16$ izlaznih razlika
- supstitucijska tablica S1:

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

- sve te mogucnosti mogu se pobrojati i zapisati u tablicu

Dio tablice koja prikazuje broj mogućih izlaznih razlika za pojedinu ulaznu razliku tablice S1

Ako su ulazi jednaki, onda i na izlazu nema razlike.

Na 6 od ukupno 64 načina se na izlazu dobije razlika 3x

Svi brojevi u tablici su parni jer je operacija XOR komutativna

Input XOR	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
0_x	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1_x	0	0	0	6	0	2	4	4	0	10	12	4	10	6	2	4
2_x	0	0	0	8	0	4	4	4	0	6	8	6	12	6	4	2
3_x	14	4	2	2	10	6	4	2	6	4	4	0	2	2	2	0
4_x	0	0	0	6	0	10	10	6	0	4	6	4	2	8	6	2
5_x	4	8	6	2	2	4	4	2	0	4	4	0	12	2	4	6
6_x	0	4	2	4	8	2	6	2	8	4	4	2	4	2	0	12
7_x	2	4	10	4	0	4	8	4	2	4	8	2	2	2	4	4
8_x	0	0	0	12	0	8	8	4	0	6	2	8	8	2	2	4
9_x	10	2	4	0	2	4	6	0	2	2	8	0	10	0	2	12
A_x	0	8	6	2	2	8	6	0	6	4	6	0	4	0	2	10
B_x	2	4	0	10	2	2	4	0	2	6	2	6	6	4	2	12
C_x	0	0	0	8	0	6	6	0	0	6	6	4	6	6	14	2
D_x	6	6	4	8	4	8	2	6	0	6	4	6	0	2	0	2
E_x	0	4	8	8	6	6	4	0	6	6	4	0	0	4	0	8
F_x	2	0	2	4	4	6	4	2	4	8	2	2	2	6	8	8
⋮																
30_x	0	4	6	0	12	6	2	2	8	2	4	4	6	2	2	4
31_x	4	8	2	10	2	2	2	2	6	0	0	2	2	4	10	8
32_x	4	2	6	4	4	2	2	4	6	6	4	8	2	2	8	0
33_x	4	4	6	2	10	8	4	2	4	0	2	2	4	6	2	4
34_x	0	8	16	6	2	0	0	12	6	0	0	0	0	8	0	6
35_x	2	2	4	0	8	0	0	0	14	4	6	8	0	2	14	0
36_x	2	6	2	2	8	0	2	2	4	2	6	8	6	4	10	0
37_x	2	2	12	4	2	4	4	10	4	4	2	6	0	2	2	4
38_x	0	6	2	2	2	0	2	2	4	6	4	4	4	6	10	10
39_x	6	2	2	4	12	6	4	8	4	0	2	4	2	4	4	0
$3A_x$	6	4	6	4	6	8	0	6	2	2	6	2	2	6	4	0
$3B_x$	2	6	4	0	0	2	4	6	4	6	8	6	4	4	6	2
$3C_x$	0	10	4	0	12	0	4	2	6	0	4	12	4	4	2	0
$3D_x$	0	8	6	2	2	6	0	8	4	4	0	4	0	12	4	4
$3E_x$	4	8	2	2	2	4	4	14	4	2	0	2	0	8	4	4
$3F_x$	4	8	4	2	4	0	2	4	4	2	4	8	8	6	2	2

U svakom retku suma brojeva je 64 jer se svaki 6-bitni ulaz može dobiti na 64 načina kao rezultat operacije XOR, npr.

$1x = 000001 =$
 $000000 \oplus 000001 =$
 $000001 \oplus 000000 =$
 $000010 \oplus 000011 =$
 $000011 \oplus 000010 =$
 $000100 \oplus 000101 =$
 $000101 \oplus 000100 =$
 itd.

i sada treba samo izbrojati sve razlike na izlazu, npr.

$S1(000000) = 1110$
 $S1(000001) = 0000$

a
 $1110 \oplus 0000 = 1110$
 pa se u retku $1x$ i stupcu $Ex=1110$
 dodaje jedna jedinica ili

$S1(000010) = 0100$
 $S1(000011) = 1111$

a
 $0100 \oplus 1111 = 1011$
 pa se u stupcu $Bx=1011$ dodaje jedna jedinica, i tako 64 puta

Izlazna razlika

Mogući ulazi za ulaznu razliku 34_x

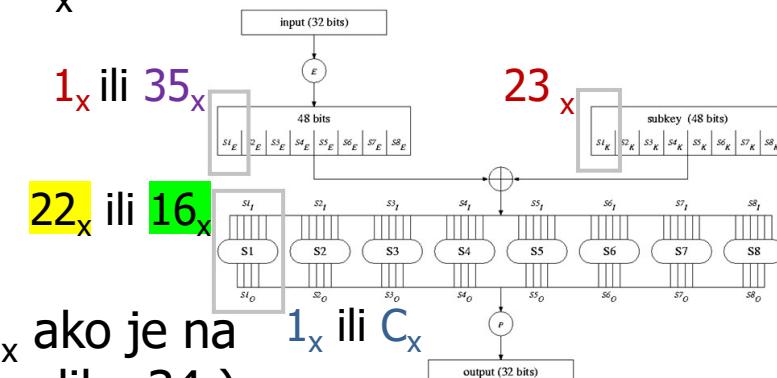
1_x	$03_x, 0F_x, 1E_x, 1F_x, 2A_x, 2B_x, 37_x, 3B_x$
2_x	$04_x, 05_x, 0E_x, 11_x, 12_x, 14_x, 1A_x, 1B_x, 20_x, 25_x, 26_x, 2E_x, 2F_x, 30_x, 31_x, 3A_x$
3_x	$01_x, 02_x, 15_x, 21_x, 35_x, 36_x$
4_x	$13_x, 27_x$
7_x	$00_x, 08_x, 0D_x, 17_x, 18_x, 1D_x, 23_x, 29_x, 2C_x, 34_x, 39_x, 3C_x$
8_x	$09_x, 0C_x, 19_x, 2D_x, 38_x, 3D_x$
D_x	$06_x, 10_x, 16_x, 1C_x, 22_x, 24_x, 28_x, 32_x$
F_x	$07_x, 0A_x, 0B_x, 33_x, 3E_x, 3F_x$

Ulaz u S-tablicu
(parovi za koje \oplus daje 34_x)

$$\begin{aligned} 06_x \oplus 32_x &= 34_x \\ 10_x \oplus 24_x &= 34_x \\ 16_x \oplus 22_x &= 34_x \\ 1C_x \oplus 28_x &= 34_x \end{aligned}$$

Mogući ključevi za izlaznu razliku D_x ako je na ulazu $S1E = 1_x$ i $S1E' = 35_x$ (ulazna razlika 34_x)

$$\begin{array}{ll} 07_x, & 33_x \\ 11_x, & 25_x \\ 17_x, & 23_x \\ 1D_x, & 29_x \end{array}$$



primjer:

$$\begin{aligned} 1_x \oplus 23_x &= 22_x \text{ u } S1 \text{ izlaz je } 1_x \\ 35_x \oplus 23_x &= 16_x \text{ u } S1, \text{ izlaz je } C_x \end{aligned}$$

izlazna razlika je $1_x \oplus C_x = D_x$

Izlazna razlika

Mogući ulazi za ulaznu razliku 34_x

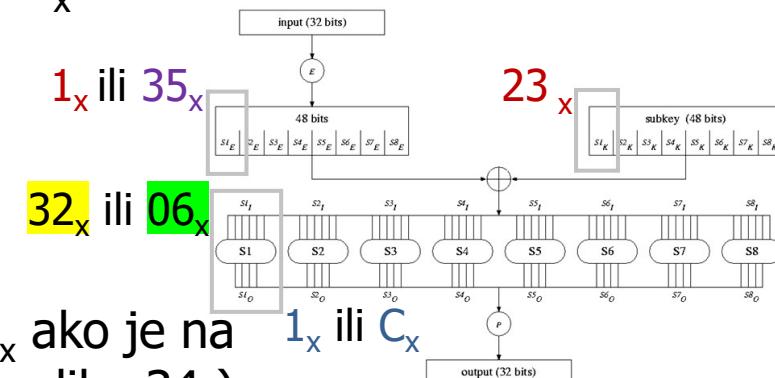
1_x	$03_x, 0F_x, 1E_x, 1F_x, 2A_x, 2B_x, 37_x, 3B_x$
2_x	$04_x, 05_x, 0E_x, 11_x, 12_x, 14_x, 1A_x, 1B_x, 20_x, 25_x, 26_x, 2E_x, 2F_x, 30_x, 31_x, 3A_x$
3_x	$01_x, 02_x, 15_x, 21_x, 35_x, 36_x$
4_x	$13_x, 27_x$
7_x	$00_x, 08_x, 0D_x, 17_x, 18_x, 1D_x, 23_x, 29_x, 2C_x, 34_x, 39_x, 3C_x$
8_x	$09_x, 0C_x, 19_x, 2D_x, 38_x, 3D_x$
D_x	$06_x, 10_x, 16_x, 1C_x, 22_x, 24_x, 28_x, 32_x$
F_x	$07_x, 0A_x, 0B_x, 33_x, 3E_x, 3F_x$

Ulaz u S-tablicu
(parovi za koje \oplus daje 34_x)

$$\begin{aligned} 06_x \oplus 32_x &= 34_x \\ 10_x \oplus 24_x &= 34_x \\ 16_x \oplus 22_x &= 34_x \\ 1C_x \oplus 28_x &= 34_x \end{aligned}$$

Mogući ključevi za izlaznu razliku D_x ako je na ulazu $S1E = 1_x$ i $S1E' = 35_x$ (ulazna razlika 34_x)

$$\begin{array}{ll} 07_x & 33_x \\ 11_x & 25_x \\ 17_x & 23_x \\ 1D_x & 29_x \end{array}$$



ili drugi primjer:

$$\begin{aligned} 1_x \oplus 07_x &= 06_x \text{ u } S1 \text{ izlaz je } 1_x \\ 35_x \oplus 07_x &= 32_x \text{ u } S1, \text{ izlaz je } C_x \end{aligned}$$

izlazna razlika je $1_x \oplus C_x = D_x$

Učinkovitost napada diferencijalnom kriptoanalizom

Broj rundi	4	6	8	9	10	11	12	13	14	15	16
Složenost	2^4	2^8	2^{16}	2^{26}	2^{35}	2^{36}	2^{43}	2^{44}	2^{51}	2^{52}	2^{58}

- Eli Biham, Adi Shamir, *Differential cryptoanalysis of the full 16-round DES*, 1991. - opisan je napad diferencijalnom analizom izvediv na potpuni DES koji je brži od pretraživanja pola prostora rješenja
- Joan Daemen, *Cipher and hash function design strategies based on linear and differential cryptoanalysis*, 1994. - opisana je metoda *Wide Trail Strategy* koja pruža zaštitu i od diferencijalne i od linearne analize

Linearna kriptoanaliza

- cilj je pronaći linearu aproksimaciju danog algoritma

$$P [i_1, i_2, \dots, i_a] \oplus C [j_1, j_2, \dots, j_b] = K [k_1, k_2, \dots, k_c]$$

- primjer: neka s vjerojatnošću p=100% vrijedi:

$$P [1, 4, 13] \oplus C [1, 2, 3, 4, 6, 9, 11] = K [5, 6, 8]$$

- paritet 5., 6. i 8. bita ključa jednoznačno je određen paritetom pojednih bitova čistog i kriptiranog teksta
 - duljina ključa efektivno smanjila za 1 bit
- aproksimacija nikada nema vjerojatnost ni blizu 100%, obično je ta vjerojatnost vrlo blizu 50%
 - taj nedostatak nadoknađuje se uzimanjem veće količine parova čisti/kriptirani tekst
- obično postoji više linearnih aproksimacija za neki algoritam

- DES Challenge I: 1997.

broj bitova ključa	vrijeme pronalaženja ključa
40	78 sekundi
48	5 sati
56	89 dana
64	41 godina
72	10.696 godina
80	2.738.199 godina
88	700.978.948 godina
96	179.450.610.898 godina
112	11.760.475.235.863.837 godina
128	770.734.505.057.572.442.069 godina

- DES Challenge II: 1998.

- Deep Crack, 56 sati
- trošak je bio \$250.000, a nagrada \$10.000



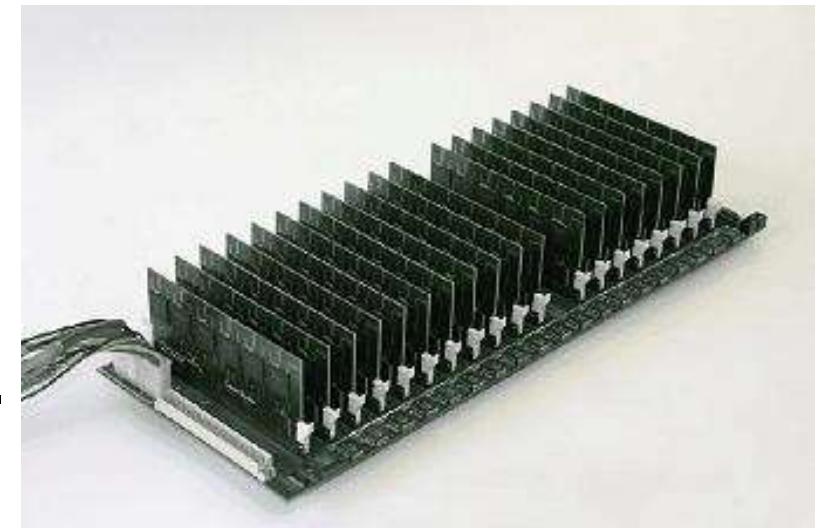
slika preuzeta s crack.sh

- DES Challenge III: 1999.

- distributed.net
- 22h i 15min nakon pretrage 22,2% prostora rješenja

COPACOBANA

- *A Cost-Optimized Parallel Code Breaker*
- razvila su ga sveučilište Ruhr iz Bochuma i Christian-Albrechts iz Kiela 2006. g.
- FPGA arhitektura, programabilan sustav
- može se iskoristiti i u druge svrhe
- 400 000 000 enkripcija u sekundi
- Sveučilišta u Bochumu i Kielu su 2006. g. izveli napad na DES
 - pretraga je trajala prosječno 7 dana
 - cijena ≈ 9 kEUR (2006.g.)



Implementacijski napadi

- napadi koji koriste propuste u programskoj i sklo povskoj implementaciji kriptografskih algoritma
 - napadi koji koriste sporedna svojstva kriptografskih uređaja (*engl. Side Channel Attacks, SCA*)
 - napadi koji analiziraju pogreške koje se javljaju u radu kriptografskih uređaja (*engl. fault analysis*)
 - napadi umetanjem grešaka (*engl. fault injection*)
 - mikrosondiranja

Napadi koji koriste sporedna fizikalna svojstva kriptografskih uređaja (engl. *side-channel attacks*)

- analiza potrošnje električne energije
- vremenski napadi
- zvuk i slika
- temperatura
- elektromagnetska zračenja

Side-channels

- *Something that enables you to know something about something without directly observing that something.*

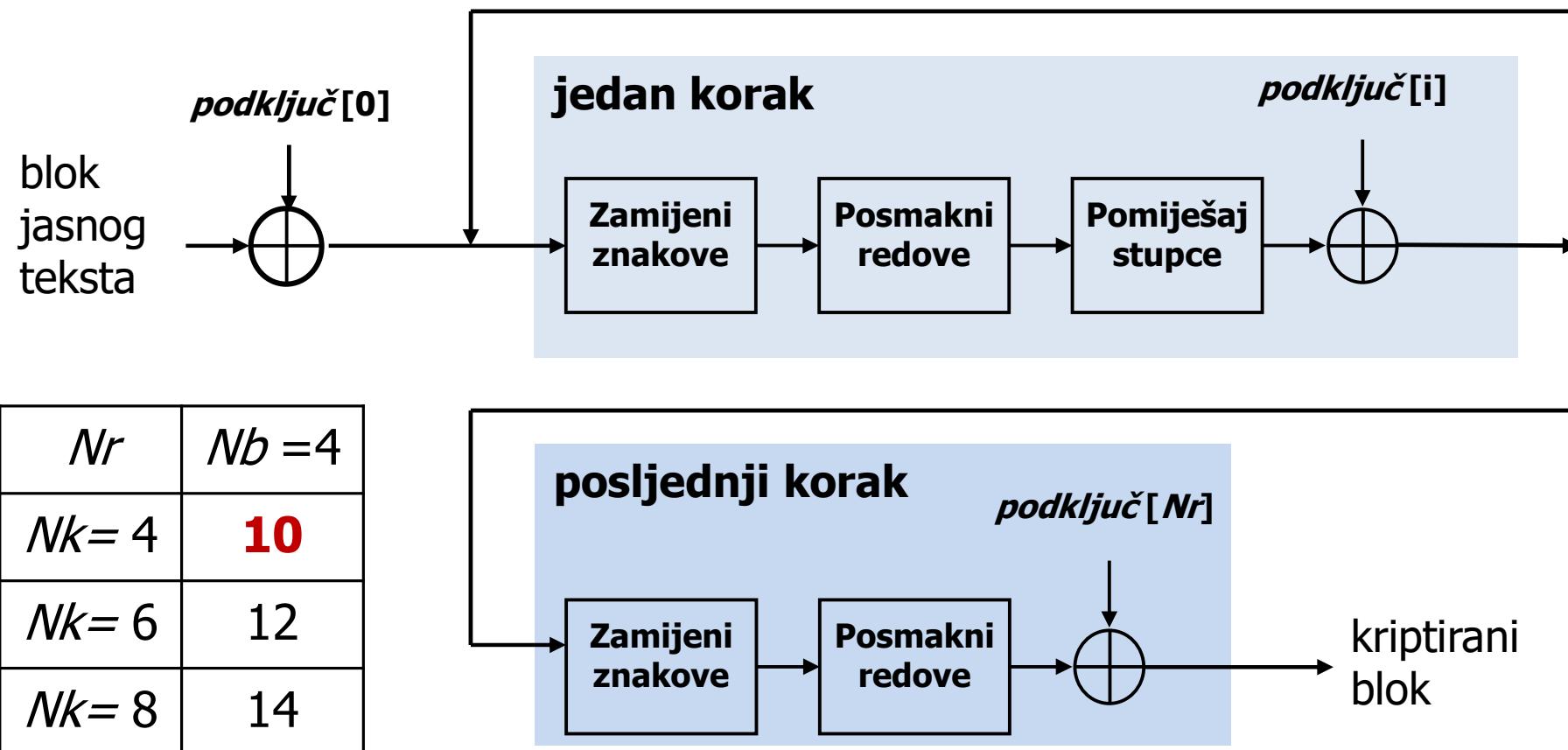


Analiza potrošnje električne energije

- engl. *Power Analysis*, PA
 - Jednostavna analiza potrošnje električne energije (*Simple Power Analysis - SPA*)
 - Diferencijalna analiza potrošnje električne energije (*Differential Power Analysis – DPA*)
- više o napadima temeljenima na analizi potrošnje električne energije na
http://www.zemris.fer.hr/predmeti/os2/kriptografska_radionica.html

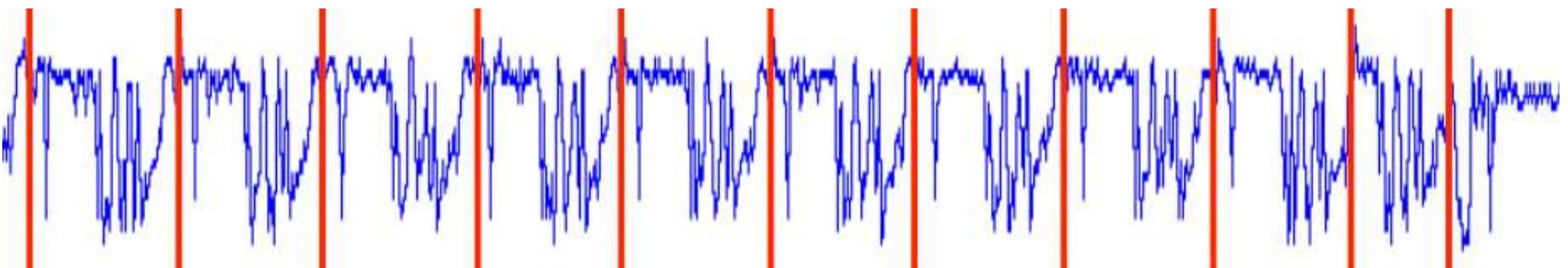
Jednostavna analiza potrošnje električne energije na primjeru kriptosustava AES

ponavljam $Nr - 1$ puta



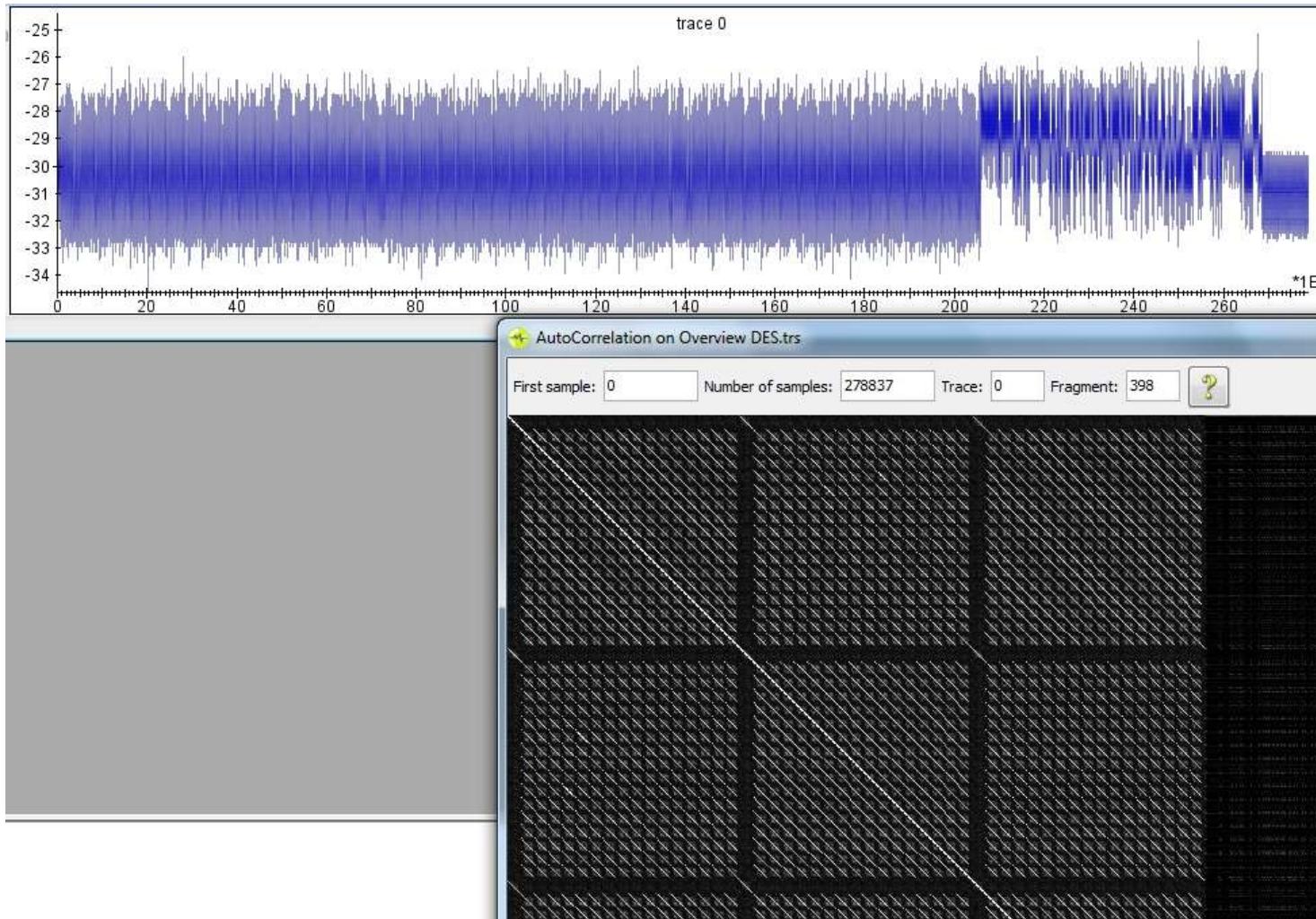
Jednostavna analiza potrošnje električne energije na primjeru kriptosustava AES

- Kolika je veličina ključa u ovom ostvarenju kriptosustava AES?



- 10 rundi = 128 bita veličina ključa

Jednostavna analiza potrošnje električne energije



- Koji je ovo algoritam?

Jednostavna analiza potrošnje električne energije na primjeru kriptosustava RSA

Kriptiranje: $C = RSA(M, S) = M^e \text{ mod } n, P = (e, n)$

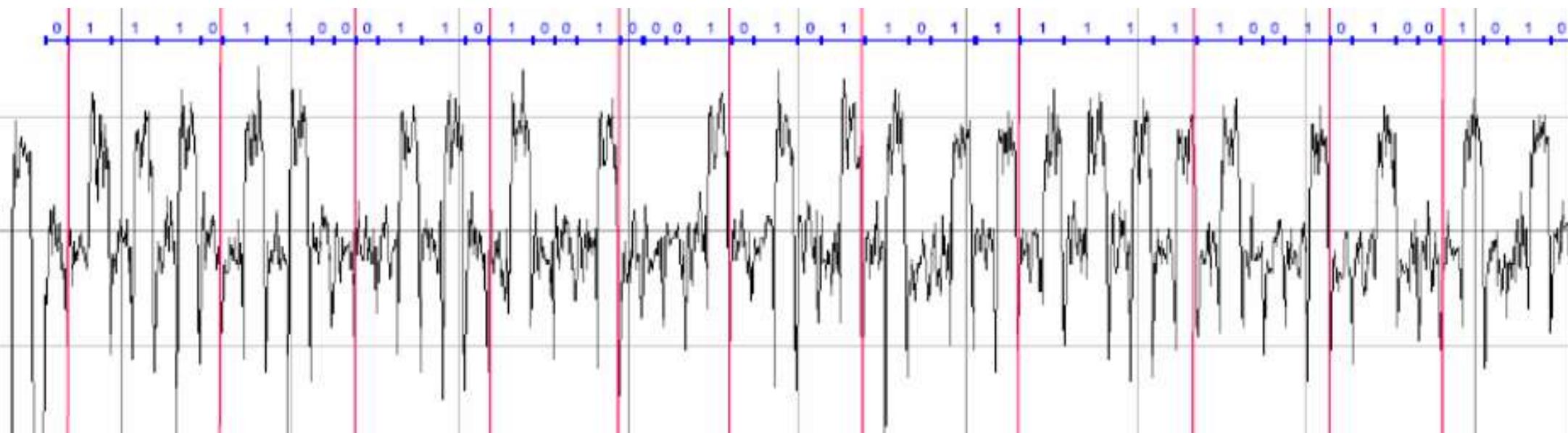
Dekriptiranje: $M = RSA^{-1}(C, P) = C^d \text{ mod } n, S = (d, n)$

pri čemu se koristi modularno potenciranje:
(želimo izračunati $d = b^a \text{ mod } n$)

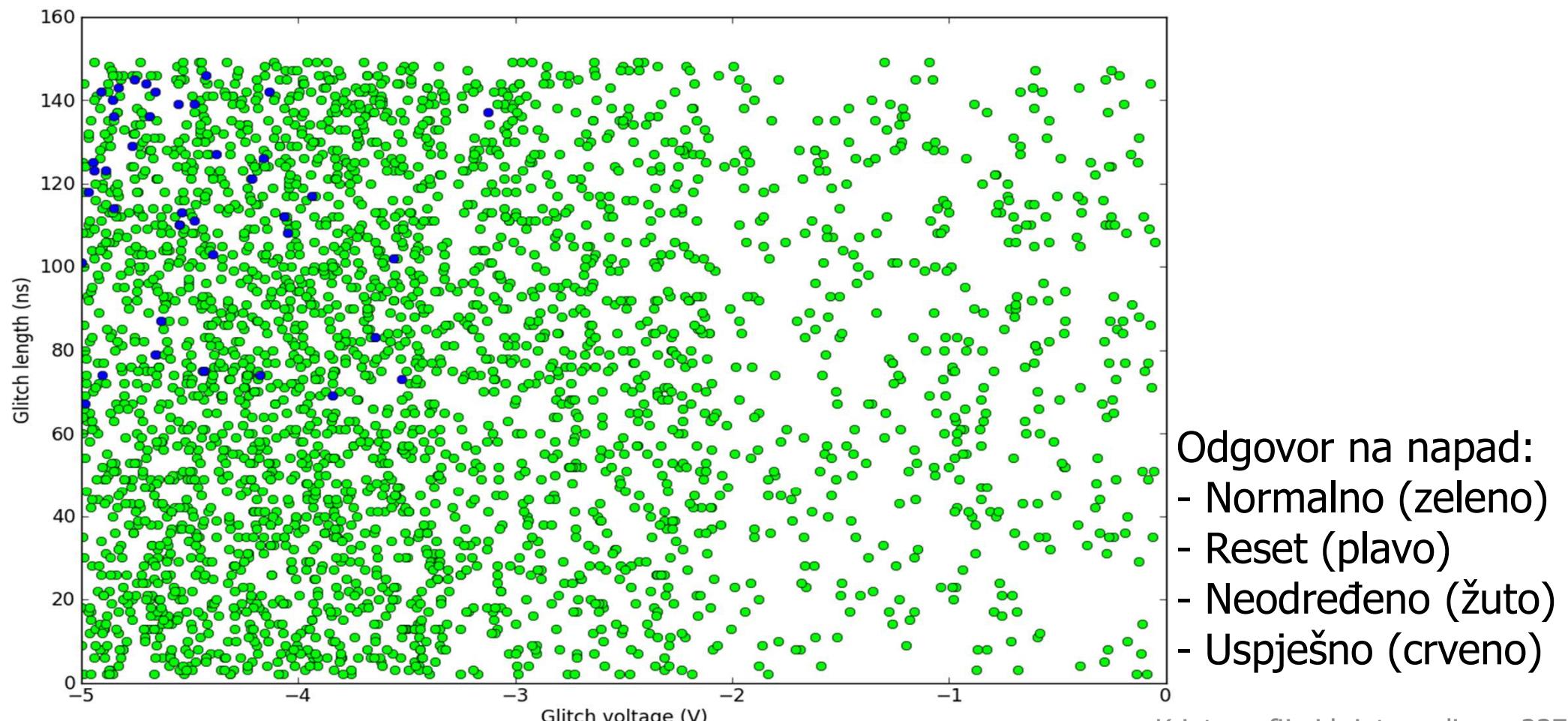
```
d = 1;
i = m;
dok je (i >= 0) {
    d = (d * d) mod n;
    ako je (a[i] == 1) {
        d = (d*b) mod n;
    }
    i--;
}
```

Jednostavna analiza potrošnje električne energije na primjeru kriptosustava RSA

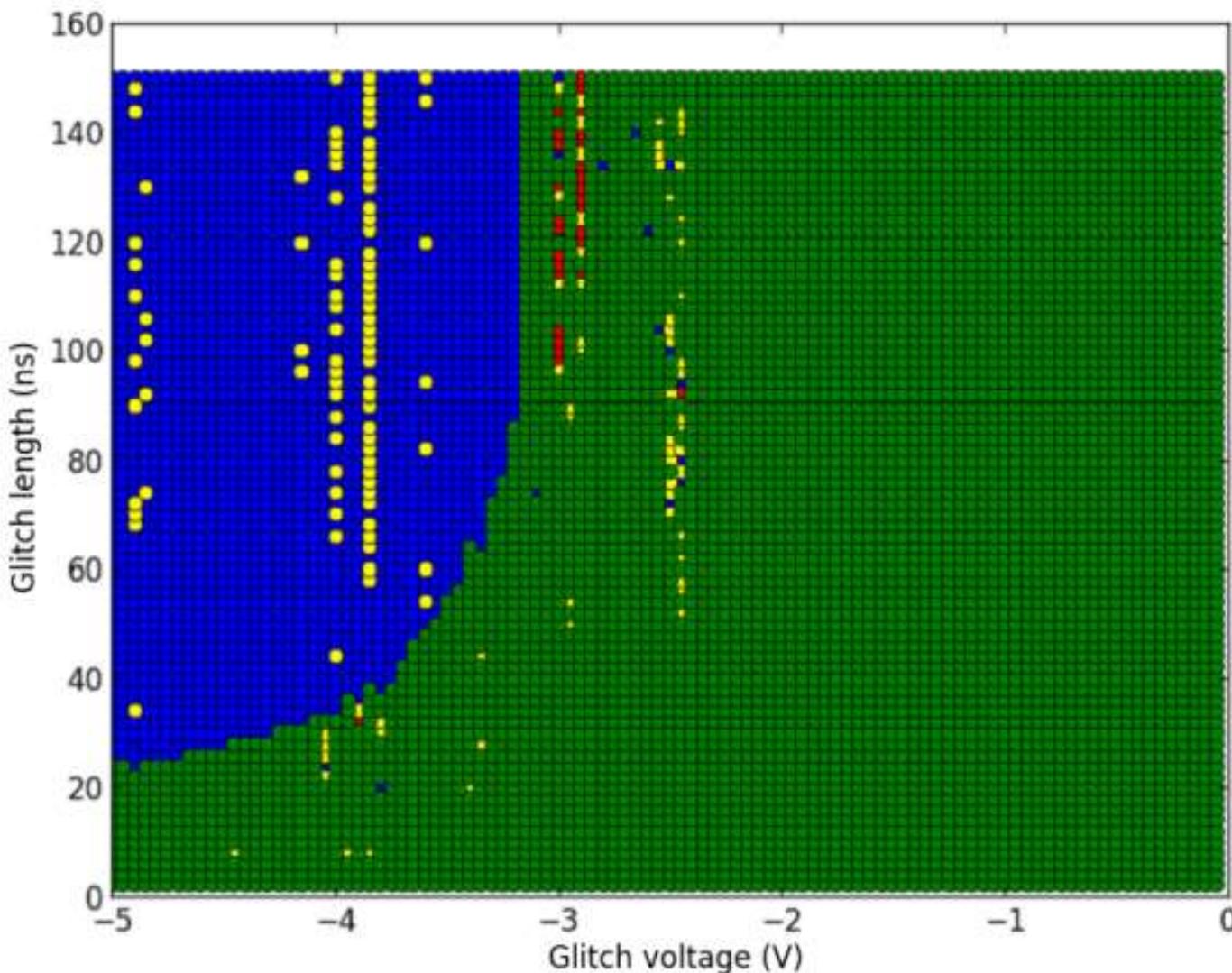
- Koji ključ se koristi u ovom ostvarenju kriptosustava RSA?



Napadi umetanjem grešaka (*fault analysis*) postupkom Monte Carlo



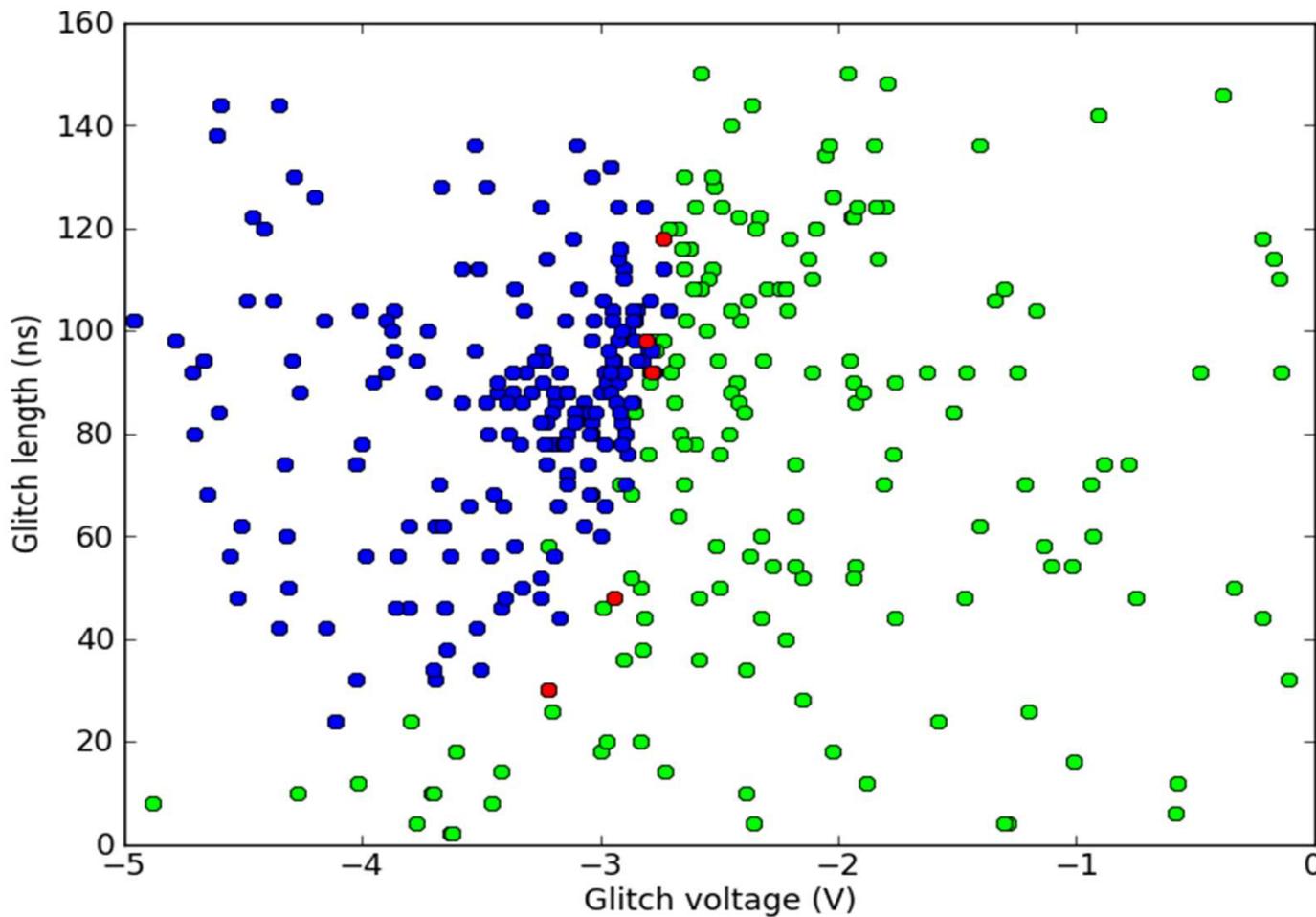
Napadi umetanjem grešaka iscrpnom pretragom



Odgovor na napad:

- Normalno (zeleno)
- Reset (plavo)
- Neodređeno (žuto)
- Uspješno (crveno)

Napadi umetanjem grešaka genetskim algoritmom



Odgovor na napad:

- Normalno (zeleno)
- Reset (plavo)
- Neodređeno (žuto)
- Uspješno (crveno)

Primjeri SCA napada

- 2025: Krađa piksela (*eng. pixnapping pixel-stealing attack*)
 - zločudni program koji u Android okruženju otkriva brojeve iz slike piksel po piksel, primjerice generirane prilikom dvorazinske autentifikacije
- 2024: "RAMBO" (Radiation of Air-gapped Memory Bus for Offense)
 - napadač instalira zločudni program na izolirano računalo (*eng. air-gapped computer*) za prikupljanje podataka i pripremu za prijenos, tj. elektromagnetsko prislушкиvanje tako da generira kontrolirano elektromagnetsko zračenje prilikom čitanja i pisanja na podatkovni dio sabirnice
 - jedinice i nule se kodiraju u radio signal kojeg napadač može pročitati s jednostavnim i jeftinim programskim radio (*eng. Software-Defined Radio, SDR*) uređajem s antenom koji prislушкиuje tako generirane signale na izoliranom računalu
 - brzine prijenosa su od 100 do 1000 bitova u sekundi
 - za prijenos primjerice biometrijske informacije veličine 10000 bitova potrebno je 10 do 100 s

Zaključak

- Koristiti provjerene programske i sklopovske implementacije algoritama koje su otporne i na implementacijske napade.
- Izbjegavati vlastita programska ostvarenja osim ako se one koriste za edukaciju. ☺