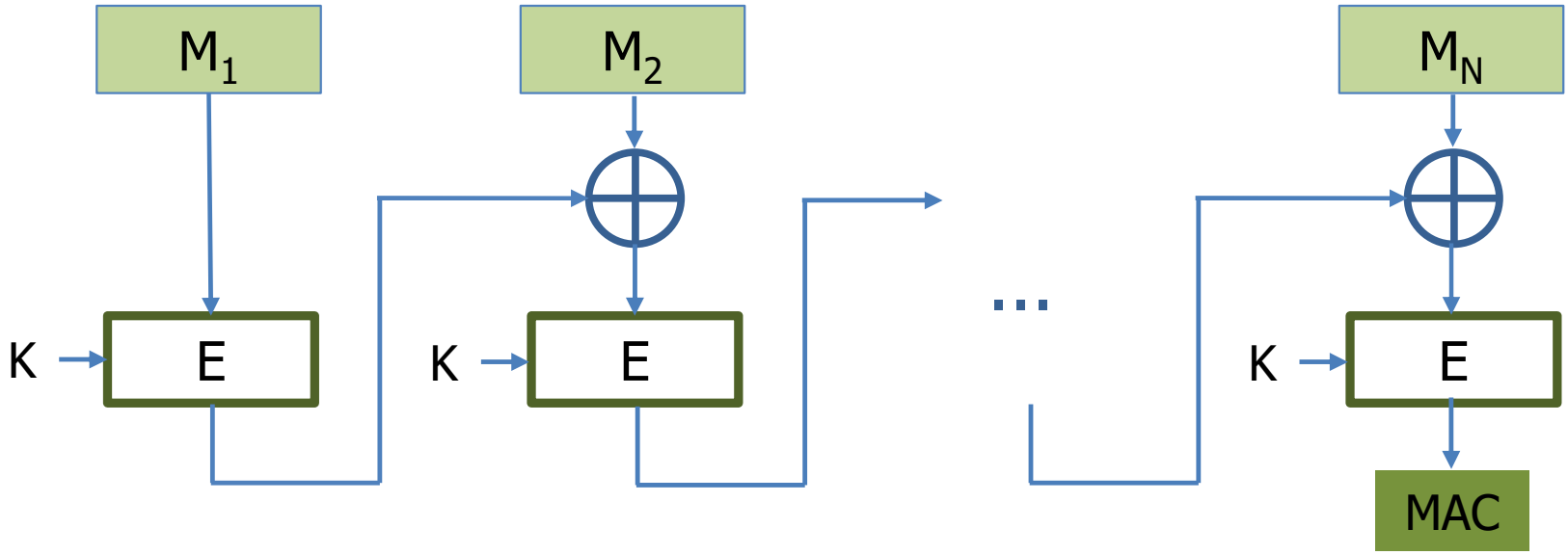
4. Autentifikacijsko kriptiranje

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.

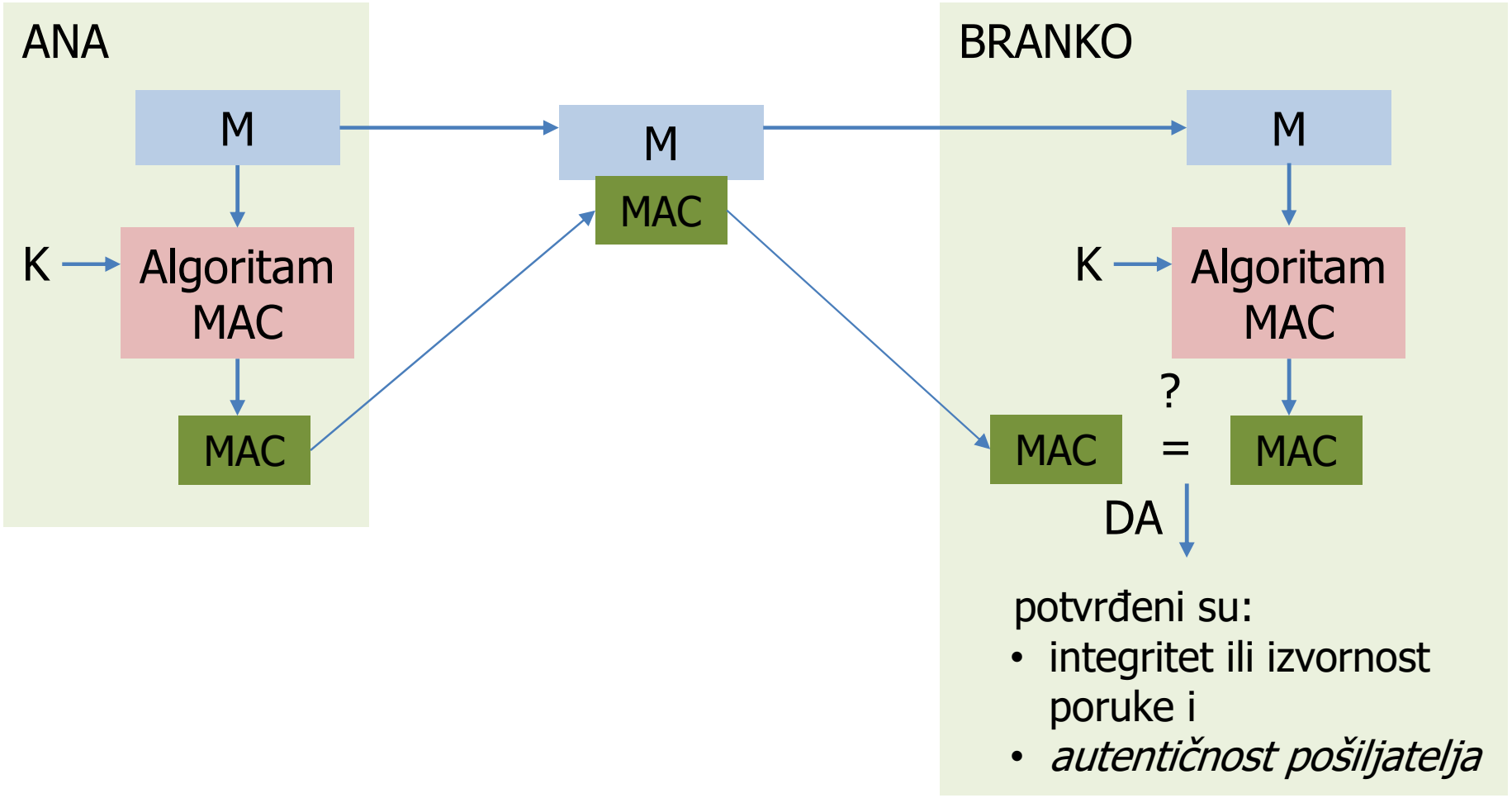
Algoritam MAC

- *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
- $HMAC(K,M) = H\{(K' \oplus opad) \,||\, H[(K' \oplus ipad) \,||\, M]\}$
  - K' = H(K) ako je K veći od veličine bloka, inače K' = K
  - konstanta *opad (outer padding*) = 0x5c5c5c...5c5c
  - konstanta *ipad* (*inner padding*) = 0x363636...3636
  - *opad* i *ipad* su veličine jednog bloka

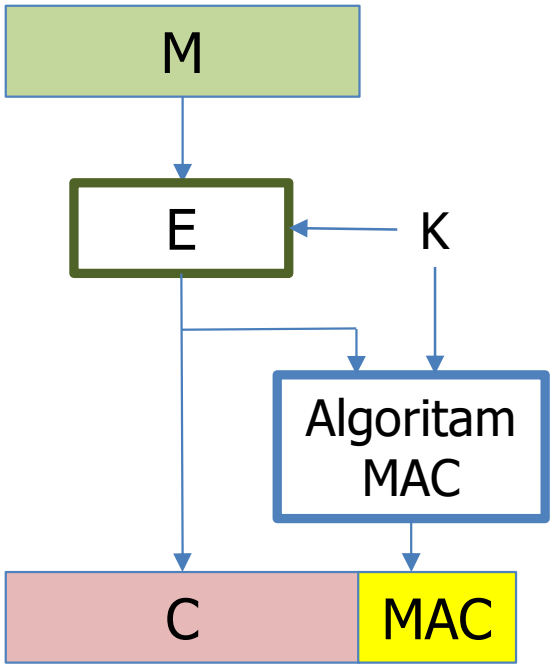
Kako osigurati i tajnost?

- 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)*

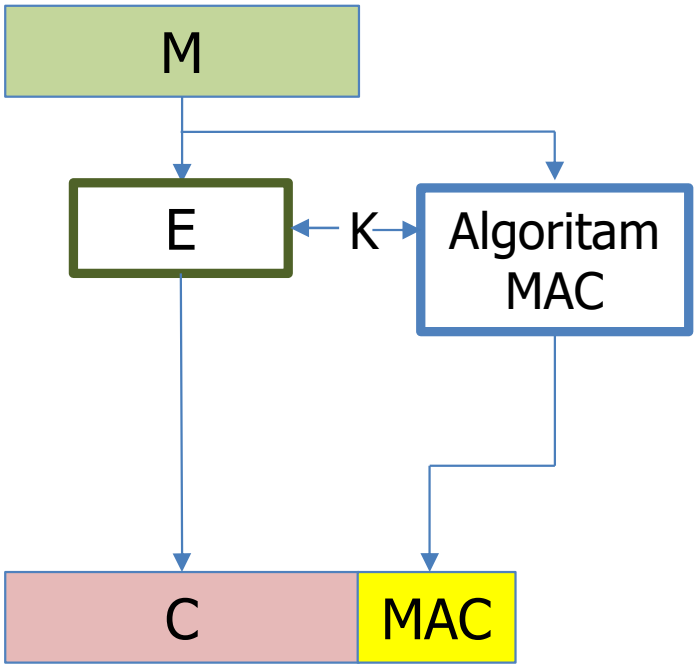
Kako uz integritet i autentičnost osigurati i tajnost?

Novi načini kriptiranja:

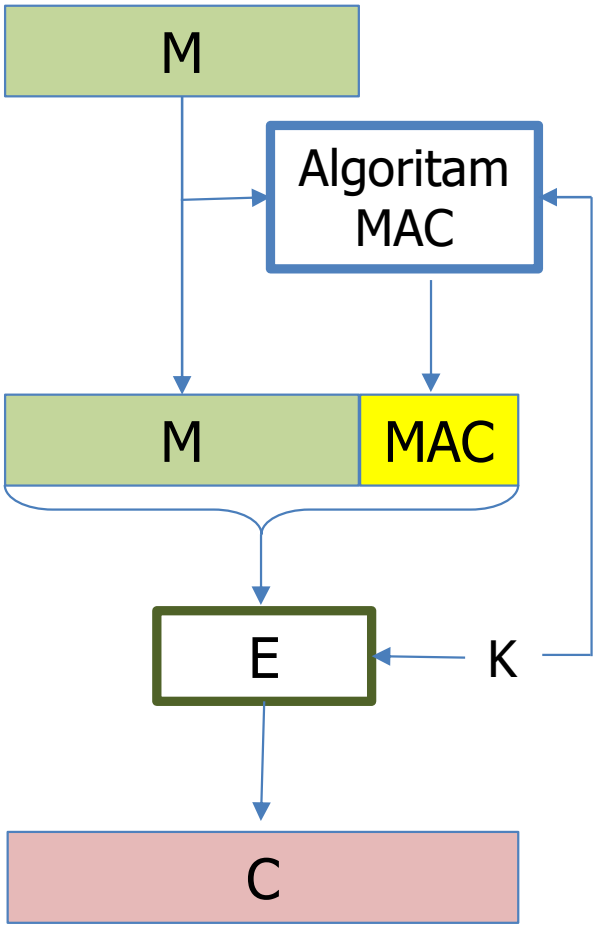
EtM



E&M



MtE

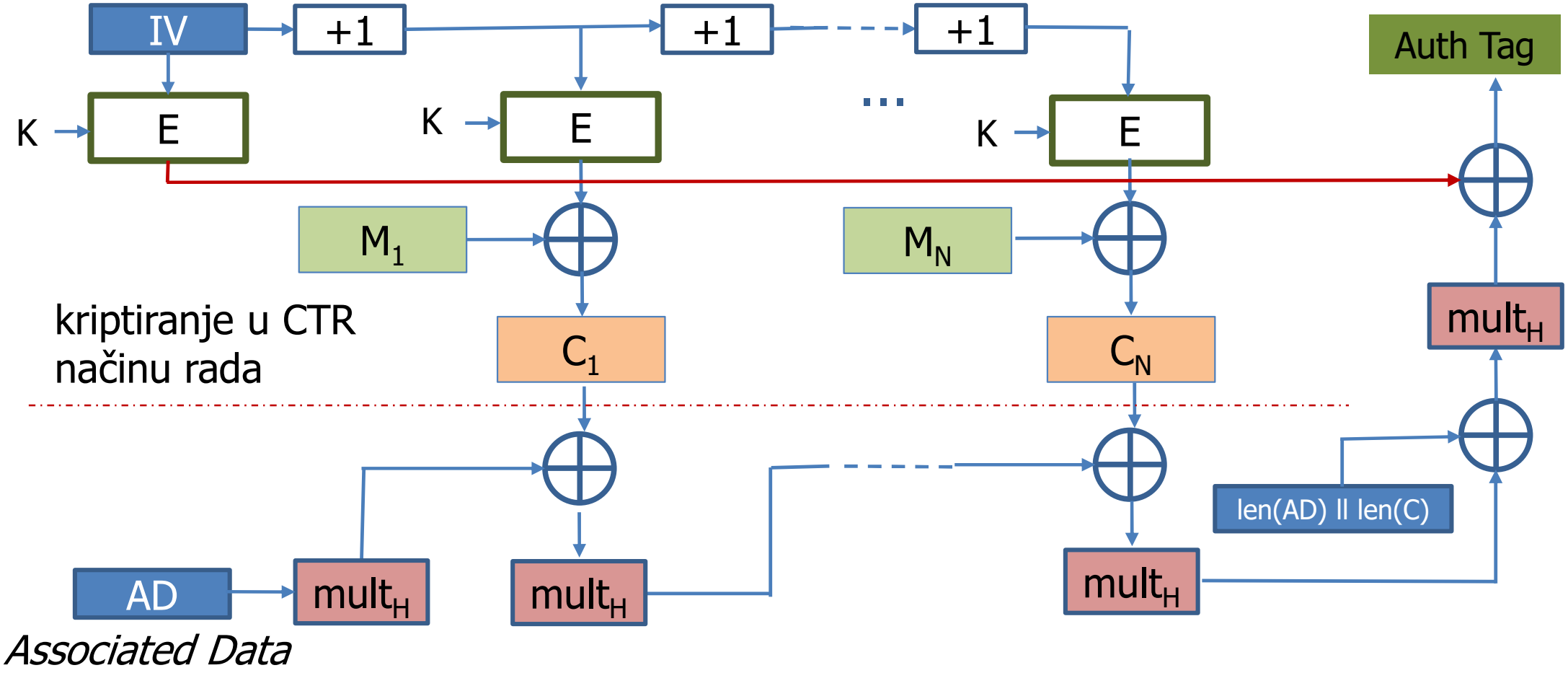


K nije isti ključ za simetričan algoritam i MAC, već se iz jednog ključa K generiraju dva

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 *K1* i *K2*
  - svakim se ključem osigurava sigurna komunikacija, ali samo u jednom smjeru:
    - Ana kriptira poruku ključem *K1* i šalje ju zajedno s dodatkom poruci MAC Branku, a Brankove poruke dekriptira i provjerava dodatak poruci MAC ključem *K2*
    - Branko kriptira poruku ključem *K2* i šalje ju zajedno s dodatkom poruci Ani, a Anine poruke dekriptira i provjerava dodatak poruci MAC ključem *K1*
    - samo Ana i Branko znaju ključeve i *K1* i *K2* 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 Prenel, ...), 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 (*lightweigt 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

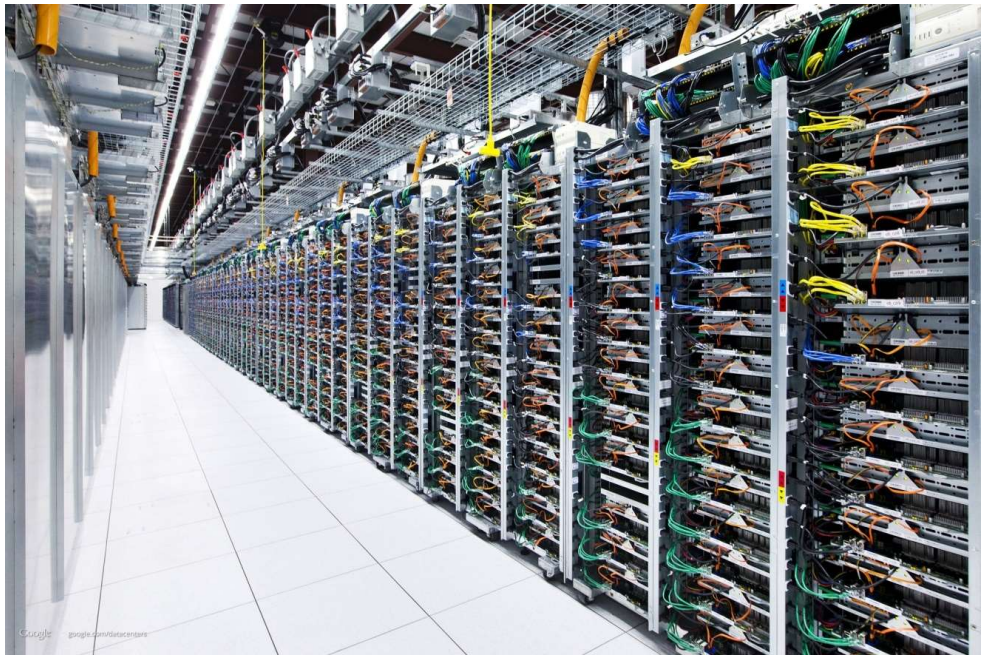
Algoritam kriptiranja bloka je *siguran*

- 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 kriptoanalize

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 *Ki*
  - ako je *C == E(M,Ki)* onda ispiši *Ki*
- 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 milijardu računala
  - 1 milijardu ključeva po sekundi po računalu
- gotovi smo za 10 tisuća milijardi godina

Primjer *uspješnog* napada na AES

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

[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:

  $C = DES(M, K)$ i $C' = DES(M', K')$

  (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 doznati tajni ključ

Napadi na DES

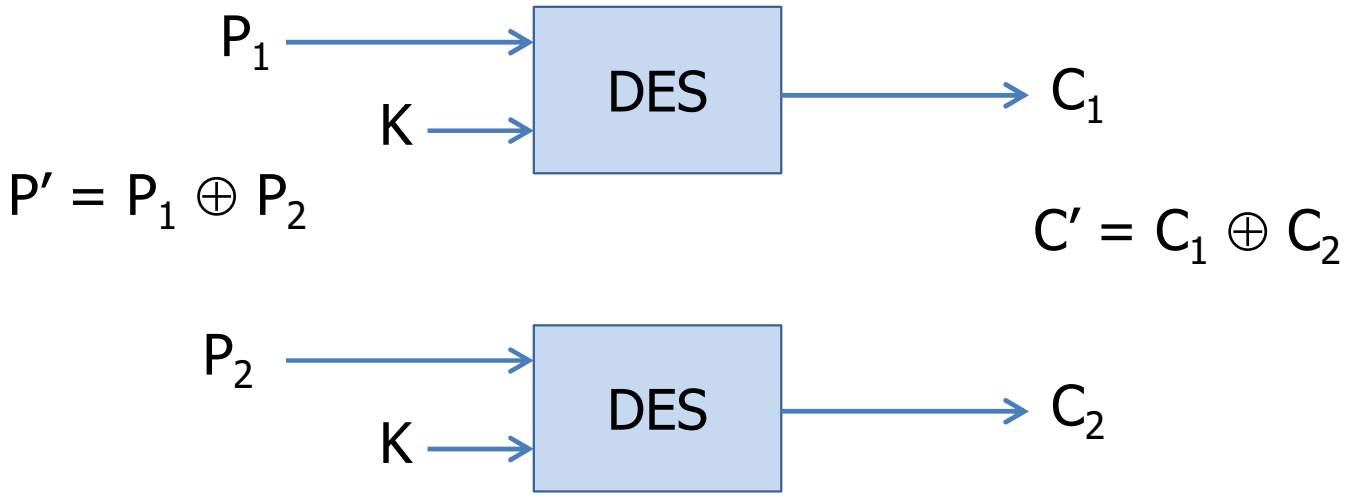
- bilo kakvim linearnim promjenama u postupku generiranja ključeva i u funkciji F, DES ne postaje otporniji na napade
- promjena u nelinearnom dijelu algoritma (S tablice) utječe na ranjivost algoritma
- DES bitno oslabljuje:
  - promjena redosljeda S tablica
  - slučajno odabrane S tablice
  - umjesto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

Kriptoanaliza

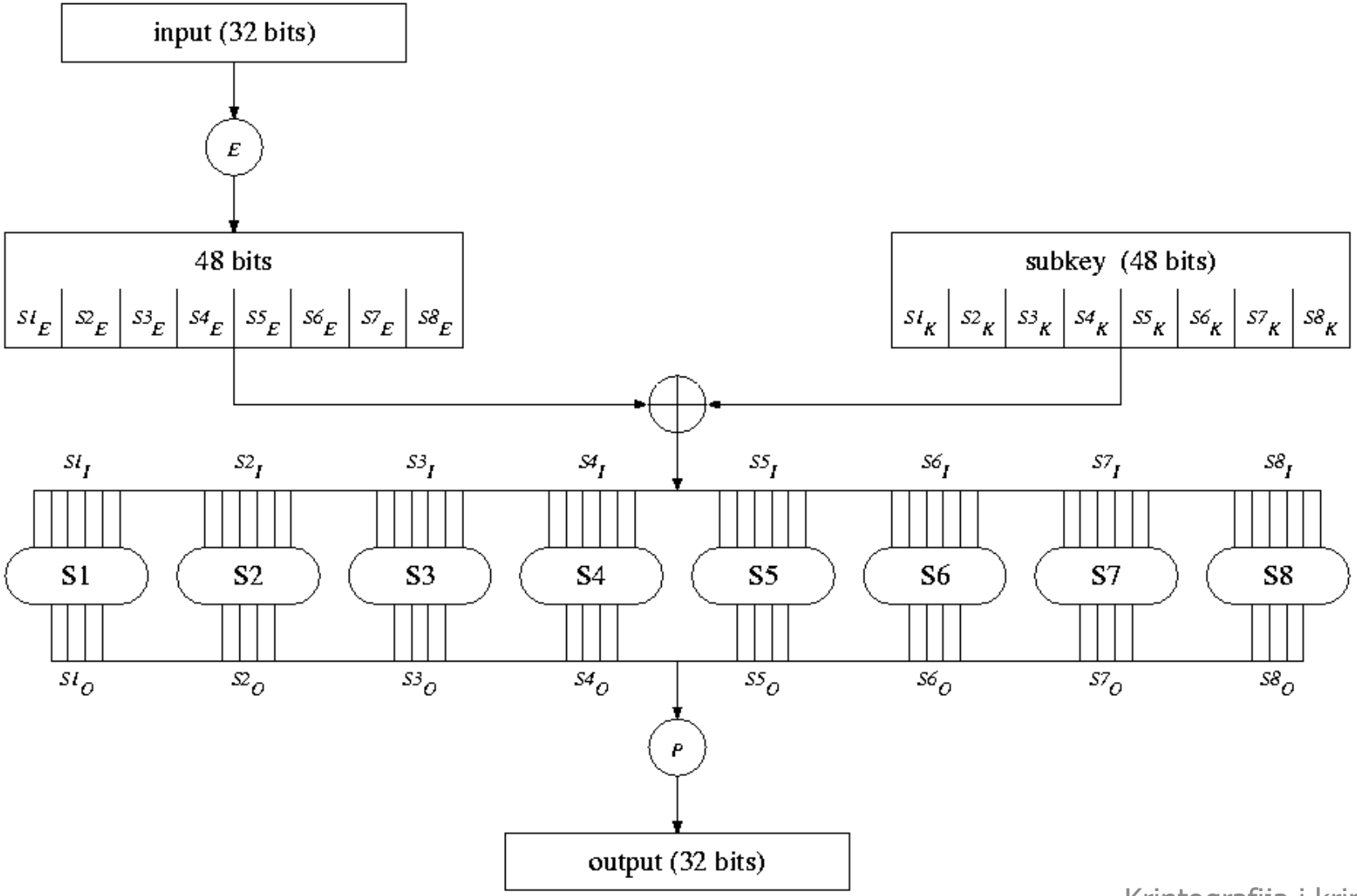
- diferencijalna kriptoanaliza
- linerna kriptoanaliza
- implementacijski napadi
  - napadi koji ne iskorištavaju slabosti algoritma (jer ih obično algoritam ni nema) već iskorištavaju sigurnosne propuste u programskim ili sklopovskim ostvarenjima
- uspješnost kriptoanalize ocjenjuje se uspoređivanjem 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 ograničen broj mogućih izlazhih razlika
  - primjerice ima i onih koje se sigurno neće pojaviti
- ulaz u neku od 8 S-tablica je veličine 6 bita, a izlaz 4 bita pa stoga postoji
  - $2^6 = 64$ mogućih 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 mogućnosti mogu se pobrojati i zapisati u tablicu

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

Input XOR	Output 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

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

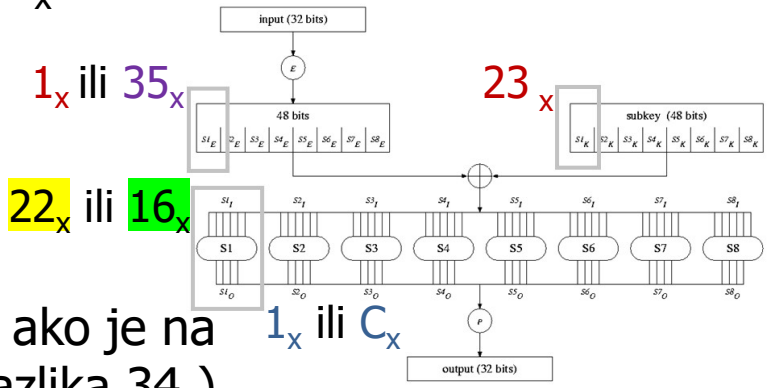
Na 6 od ukupno 64 načina se na izlazu dobije razlika 3x

Svi brojevi u tablici su parni jer je operacija XOR komutativna

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 ⊕ 000001 =
000001 ⊕ 000000 =
000010 ⊕ 000011 =
000011 ⊕ 000010 =
000100 ⊕ 000101 =
000101 ⊕ 000100 =
itd.
i sada treba samo izbrojati sve razlike na izlazu, npr.
S1(000000) = 1110
S1(000001) = 0000
a
1110 ⊕ 0000 = 1110
pa se u retku 1x i stupcu Ex=1110 dodaje jedna jedinica
ili
S1(000010) = 0100
S1(000011) = 1111
a
0100 ⊕ 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, $3E_x$, $3F_x$



1_x ili 35_x

23_x

22_x ili 16_x

1_x ili C_x

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

$06_x \oplus 32_x = 34_x$
$10_x \oplus 24_x = 34_x$
$16_x \oplus 22_x = 34_x$
$1C_x \oplus 28_x = 34_x$

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

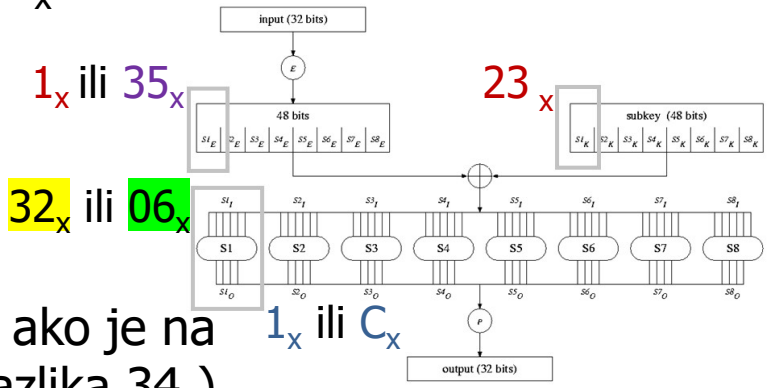
07_x, 33_x
11_x, 25_x
17_x, 23_x
$1D_x$, 29_x

primjer:
$1_x \oplus 23_x = 22_x$ u S1 izlaz je 1_x
$35_x \oplus 23_x = 16_x$ u S1, izlaz je C_x

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, $3E_x$, $3F_x$



1_x ili 35_x
23_x
32_x ili 06_x
1_x ili C_x

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

$06_x \oplus 32_x = 34_x$
$10_x \oplus 24_x = 34_x$
$16_x \oplus 22_x = 34_x$
$1C_x \oplus 28_x = 34_x$

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

07_x, 33_x
11_x, 25_x
17_x, 23_x
$1D_x$, 29_x

ili drugi primjer:
$1_x \oplus 07_x = 06_x$ u S1 izlaz je 1_x
$35_x \oplus 07_x = 32_x$ u S1, izlaz je C_x

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 linearnu aproksimaciju danog algoritma

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

- 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 lineanih 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

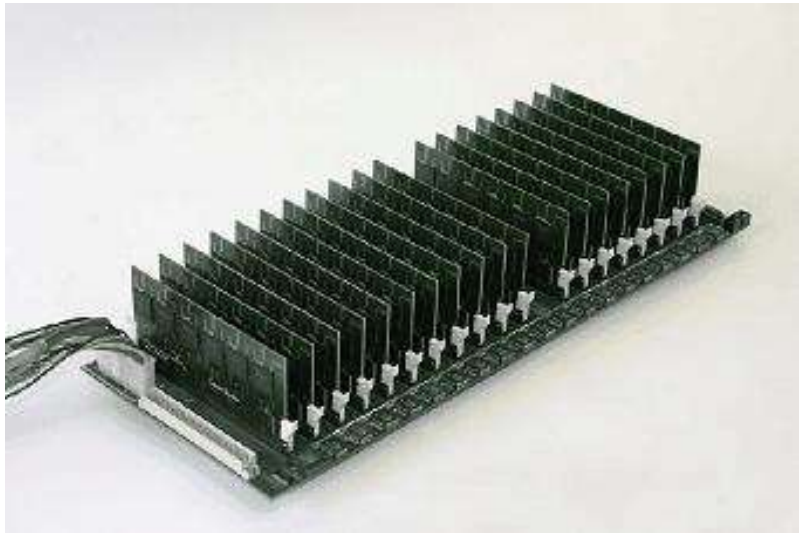
- DES Challenge II: 1998.
  - Deep Crack, 56 sati
  - trošak je bio $250.000, a nagrada $10.000

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, programibilan 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 sklopovskoj 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-chanels

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

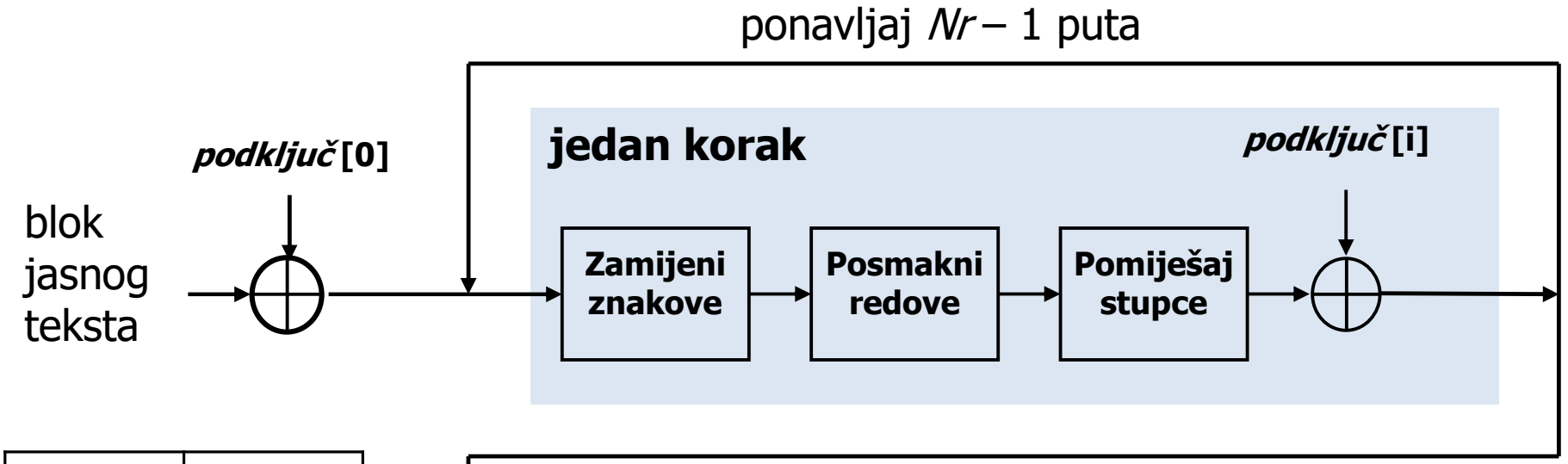
Analiza potrošnje električne energije

- engl. *Power Analysys*, 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

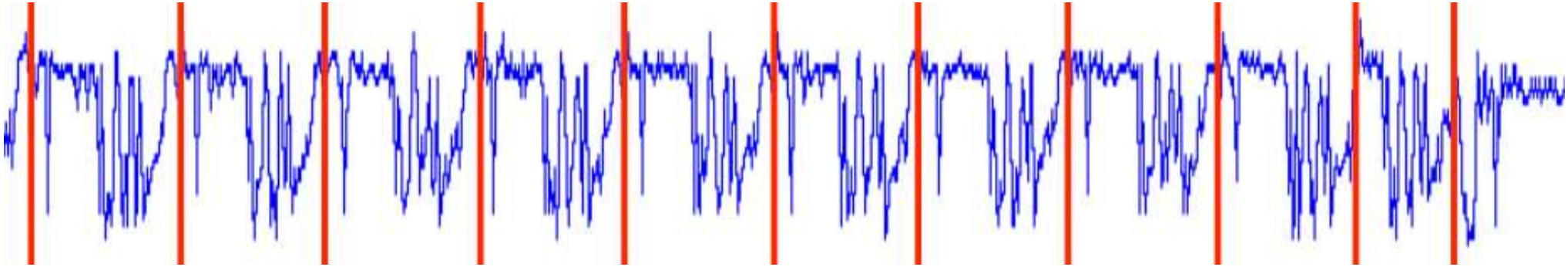
Jednostavna analiza potrošnje električne energije na primjeru kriptosustava AES



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

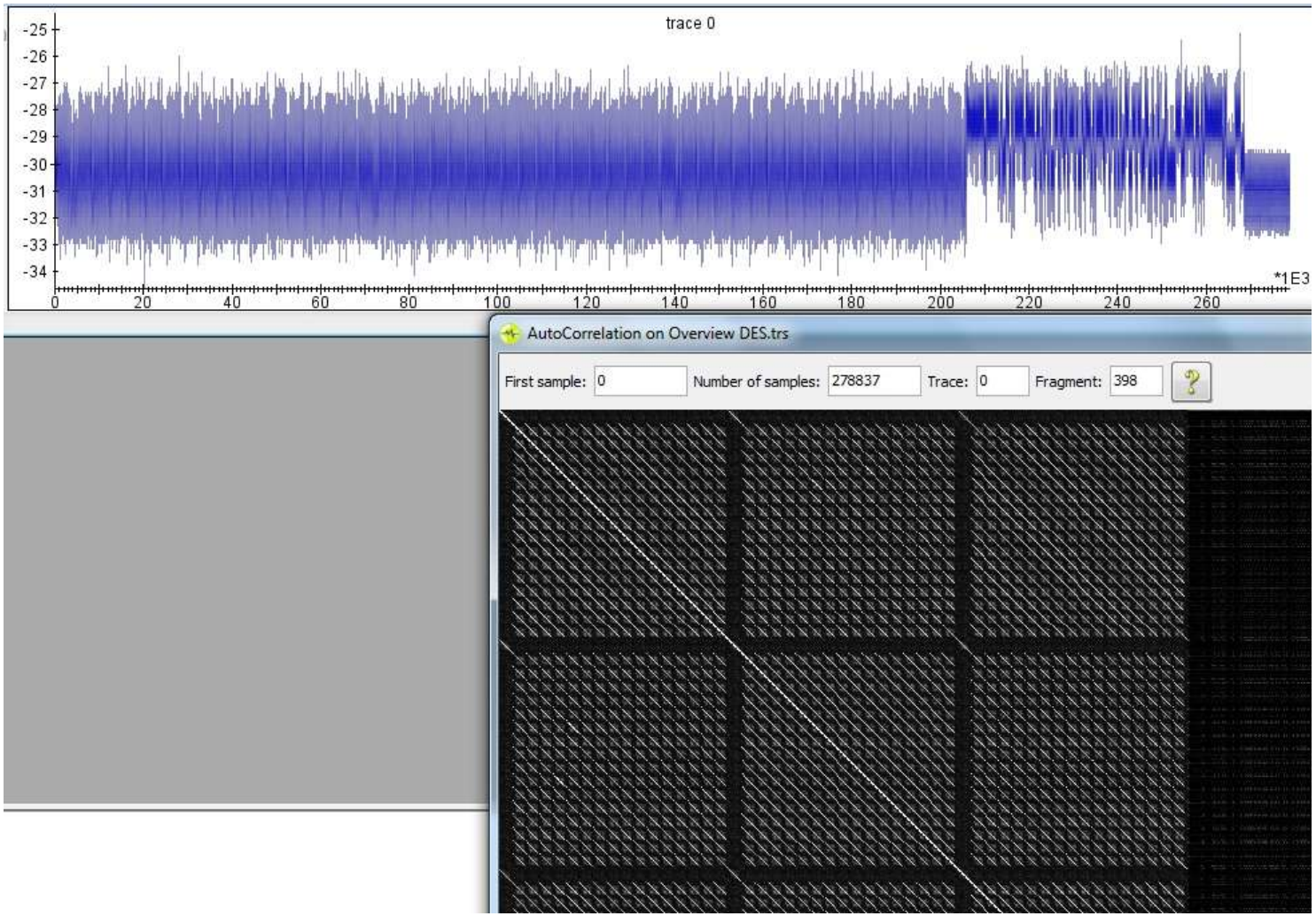
Jednostavna analiza potrošnje električne energije na primjeru kriptosustava AES

- 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: $C = RSA(M, S) = M^e \bmod n, \quad P = (e, n)$

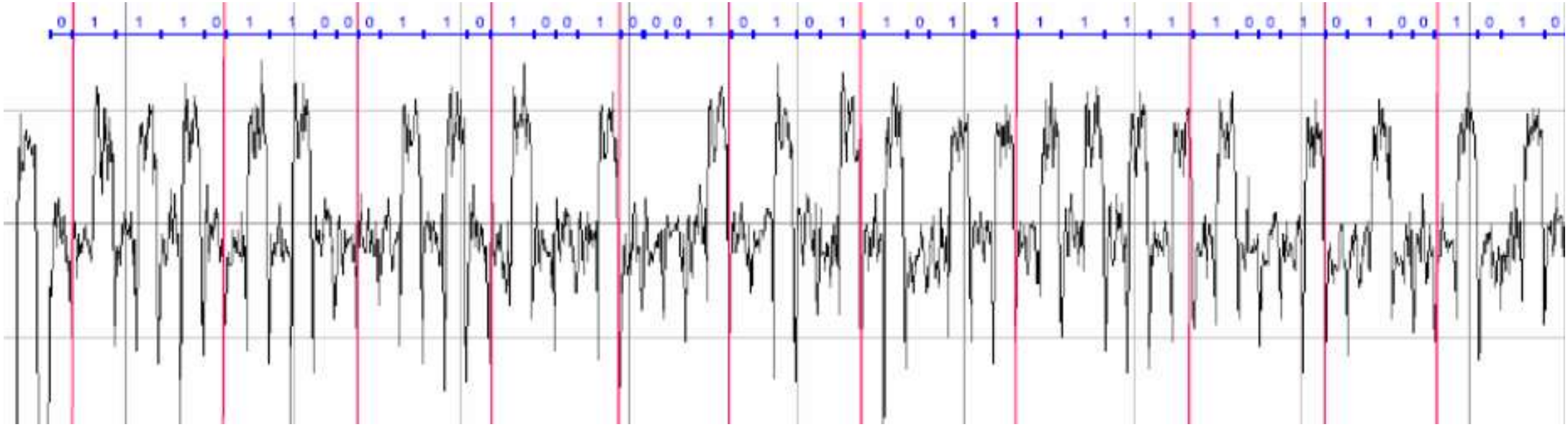
Dekriptiranje: $M = RSA^{-1}(C, P) = C^d \bmod n, \quad S = (d, n)$

pri čemu se koristi modularno potenciranje:
(želimo izračunati $d = b^a \bmod n$)

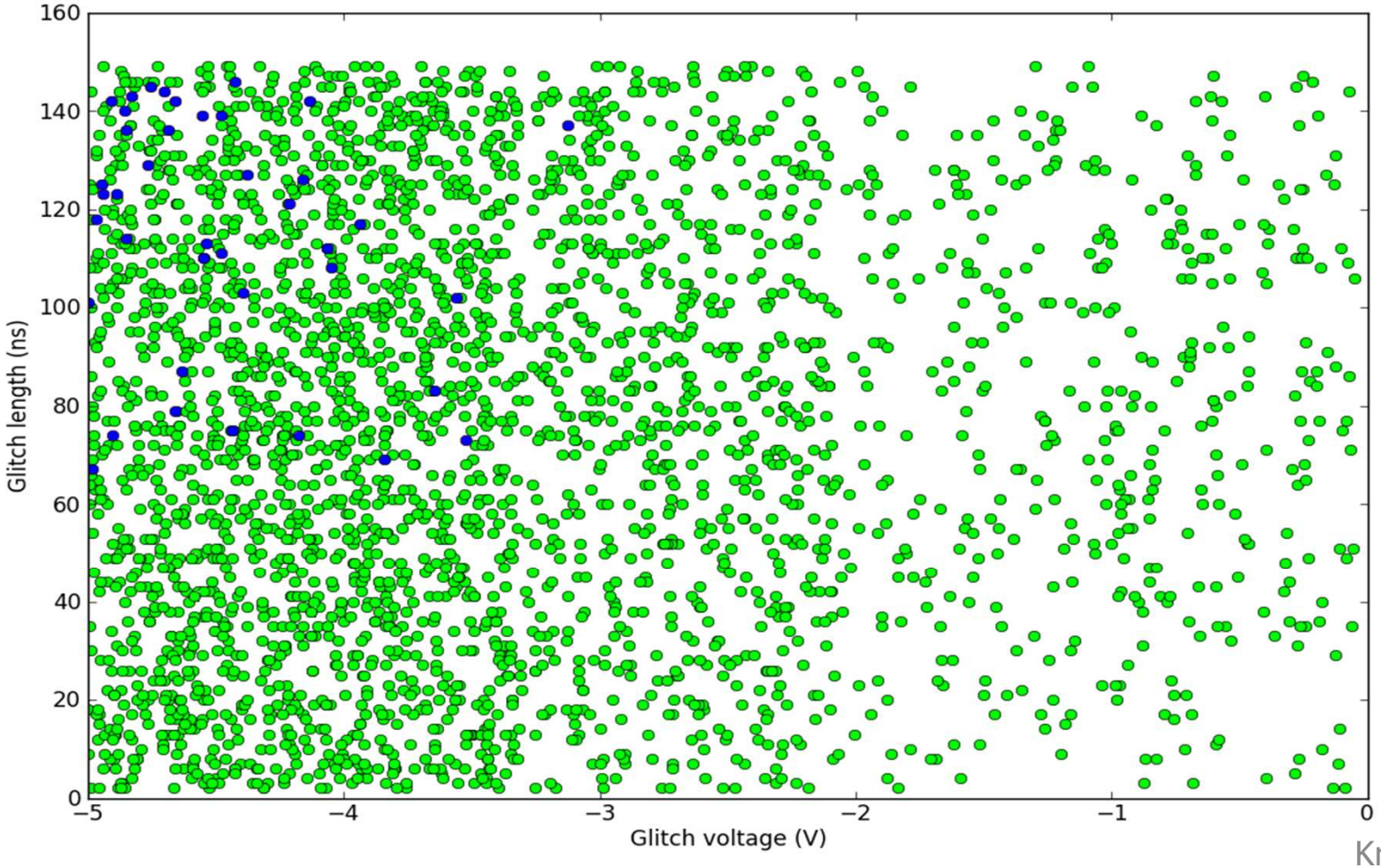
```
d = 1;
i = m;
dok je (i >= 0) {
    d = (d * d) mod n;
    ako je (a[i] == 1) {
        d = (d*b) mod n;
    }
    i --;
}
```

Jednostavna analiza potrošnje električne energije na primjeru kriptosustava RSA

- Koji ključ se koristi u ovom ostvarenju kriptosustava RSA?

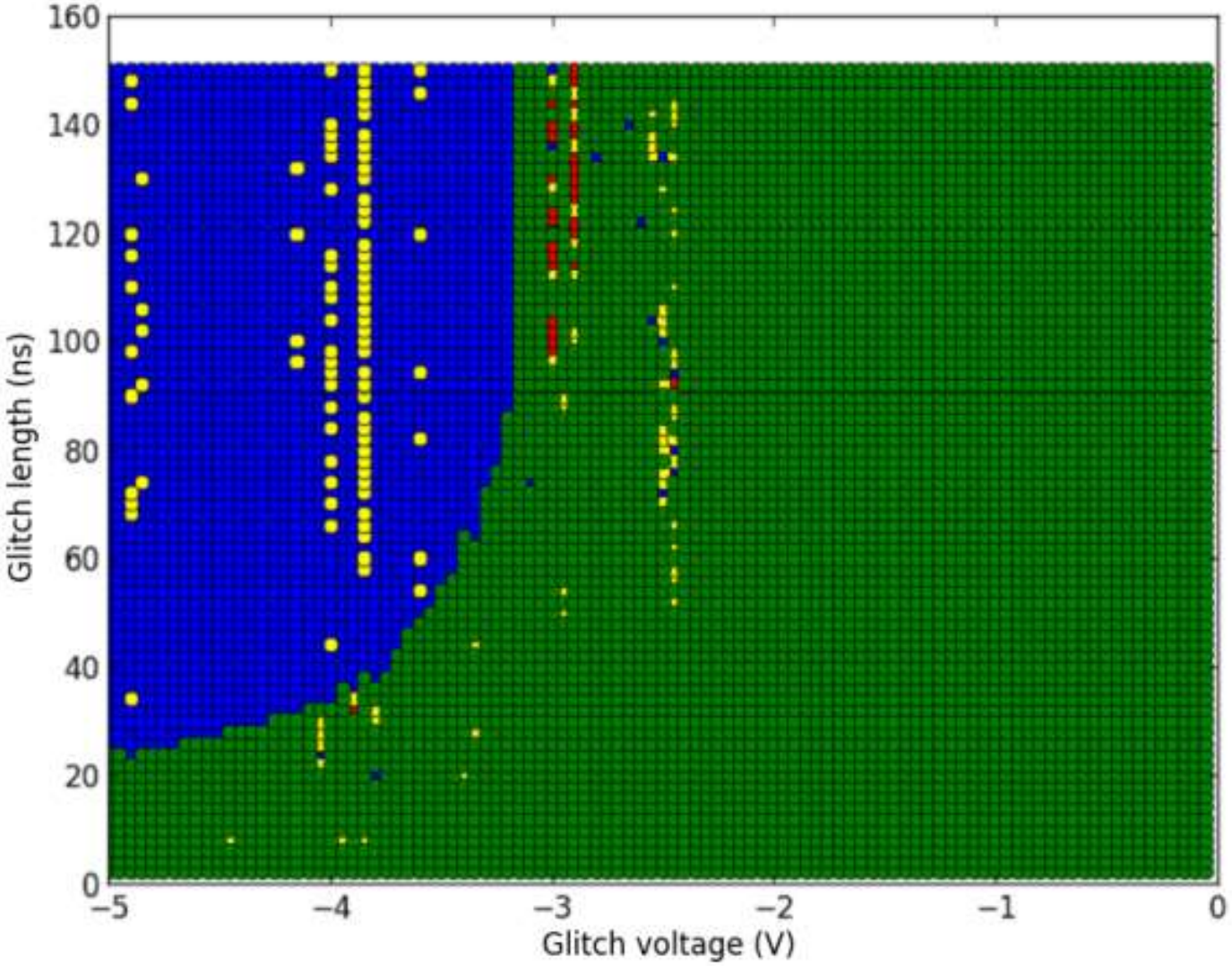


Napadi umetanjem grešaka (*fault analysis*) postupkom Monte Carlo



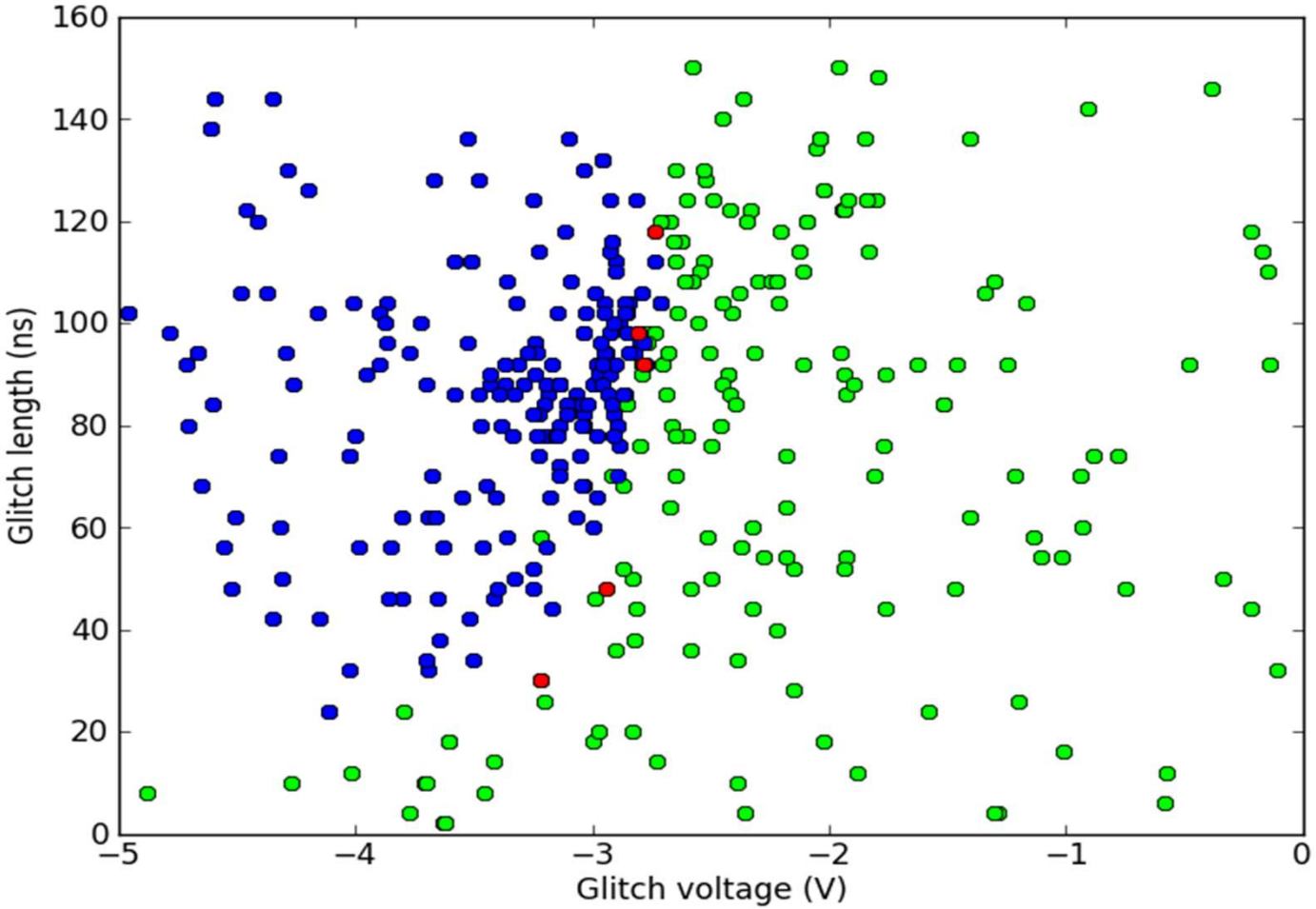
Odgovor na napad:
- Normalno (zeleno)
- Reset (plavo)
- Neodređeno (žuto)
- Uspješno (crveno)

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*)
  - zlćudni 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 zlćudni program na izolirano računalo (*eng. air-gapped computer*) za prikupljanje podataka i pripremu za prijenos, tj. elektromagnetsko prisluškivanje 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 prisluškuje 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. ☺