3. poglavje

Simetrični kriptosistemi

- Bločne šifre, nekaj zgodovine, DES, AES
- Iterativne šifre, zmenjalno-permutacijske mreže
- Produktna šifra in Fiestelova šifra
- Opis šifer DES in AES
- Načini delovanja (ECB, CBC, CFB, OFB) in MAC
- Napadi in velika števila
- 3-DES, DESX in druge simetrične bločne šifre

Bločne šifre

Bločna šifra je simetrična šifra, ki razdeli čistopis na bloke fiksne dolžine (npr. 128 bitov), in šifrira vsak blok posamično (kontrast: *tekoča šifra* zašifrira čistopis po znakih – ponavadi celo po bitih).

Najmodernejše bločne šifre so **produktne šifre**, ki smo jih spoznali v prejšnjem poglavju: komponiranje več enostavnih operacij, katere (vsaka posebej) niso dovolj varne, z namenom, da povečamo varnost:
transpozicije, ekskluzivni ali (XOR), tabele, linearne transformacije, aritmetične operacije, modularno množenje, enostavne substitucije.

Primeri bločnih produktnih šifer: DES, AES, IDEA.

Nekatere želene lastnosti bločnih šifer

Varnost:

- **razpršitev**: vsak bit tajnopisa naj bo odvisen od vseh bitov čistopisa.
- **zmeda**: zveza med ključem ter biti tajnopisa naj bo zapletena,
- **velikost ključev**: mora biti majhna, toda dovolj velika da prepreči požrešno iskanje ključa.

Učinkovitost

- hitro šifriranje in odšifriranje,
- enostavnost (za lažjo implementacijo in analizo),
- primernost za hardware ali software.

Kratka zgodovina bločnih šifer DES in AES

Konec 1960-ih: IBM – Feistelova šifra in LUCIFER.

1972: NBS (sedaj NIST) izbira simetrično šifro za zaščito računalniških podatkov.

1974: IBM razvije DES, 1975: NSA ga "popravi".

1977: DES sprejet kot US Federal Information Processing Standard (FIPS 46).

1981: DES sprejet kot US bančni standard (ANSI X3.92).

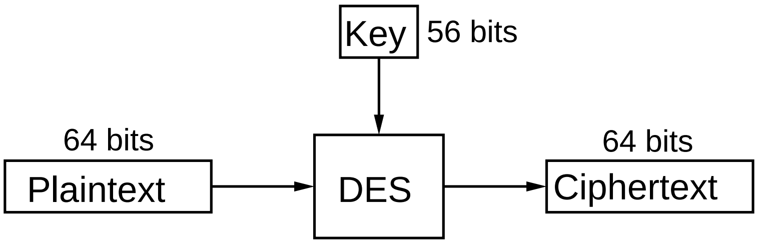
1997: AES (Advanced Encryption Standard) izbor

1999: izbranih 5 finalistov za AES

National Security Agency (NSA)

- www.nsa.gov
- ustanovljena leta 1952,
- neznana sredstva in število zaposlenih (čez 100.000?)
- Signals Intelligence (SIGINT):
  pridobiva tuje informacije.
- Information Systems Security (INFOSEC):
  ščiti vse občutljive (classified) informacije,
  ki jih hrani ali pošilja vlada ZDA,
- zelo vplivna pri določanju izvoznih regulacij ZDA za kriptografske produkte (še posebej šifriranje).

Data Encryption Standard (DES)



Ideja za DES je bila zasnovana pri IBM-u v 60-ih letih (uporabili so koncept Claude Shannona imenovan *Lucifer*).

NSA je zreducirala dolžino ključev s 128 bitov na 56.

V sredini 70-ih let je postal prvi komercijalni algoritem, ki je bil objavljen z vsemi podrobnostmi (FIPS 46-2).

Advanced Encryption Standard

AES je ime za nov FIPS-ov simetrični (bločni) kriptosistem, ki bo nadomestil DES.

Leta 2000 je zanj *National Institute of Standards and Technology (NIST)* izbral belgijsko bločno šifro **Rijndael**.

Dolžina *ključev* oziroma blokov je 128, 192 ali 256

Uporabljala pa ga tudi ameriška vlada, glej `http://csrc.nist.gov/encryption/aes/round2/r2report.pdf`.

Običajno uporabljamo **iterativne šifre**.

Tipični opis:

- krožna funkcija,
- razpored ključev,
- šifriranje skozi N_r podobnih krogov.

Naj bo K naključni binarni ključ določene dolžine.

K uporabimo za konstrukcijo podključev za vsak krog s pomočjo *javno* znanega algoritma.

Imenujemo jih **krožni ključi:** $K^1, \dots, K^{N_r}$.

Seznamu krožnih ključev $(K^1, \dots, K^{N_r})$ pa pravimo **razpored ključev**.

Krožna funkcija g ima dva argumenta:

(i) krožni ključ (K^r) in (ii) *tekoče stanje* (w^{r-1}).

Naslednje stanje je definirano z $w^r = g(w^{r-1}, K^r)$.

Začetno stanje, w_0, naj bo čistopis x.

Potem za tajnopis, y, vzamemo stanje po N_r krogih:

$$y = g(g(\dots g(g(x, K^1), K^2) \dots, K^{N_r - 1})K^{N_r}).$$

Da je odšifriranje možno, mora biti funkcija g injektivna za vsak fiksen ključ K_i, tj. $\exists\ g^{-1}$, da je:

$$g^{-1}(g(w, K), K) = w, \qquad \text{za vse } w \text{ in } K.$$

Odšifriranje opravljeno po naslednjem postopku:

$$x = g^{-1}(g^{-1}(\dots g^{-1}(g^{-1}(y, K^{N_r}), K^{N_r - 1}) \dots, K^2)K^1).$$

Zamenjalno-permutacijske mreže

(angl. *substitution-permutation network* – (**SPN**)).

Čistopis $\mathcal{P}$ in tajnopis $\mathcal{C}$ so binarni vektorji dolžine ℓm, $\ell, m \in \mathbb{N}$ (tj. ℓm je dolžina bloka).

SPN je zgrajen iz dveh komponent (zamenjave in permutacije):

$$\begin{aligned} \pi_S &: \{0,1\}^{\ell} \longrightarrow \{0,1\}^{\ell}, \\ \pi_P &: \{0,\dots,\ell m\} \longrightarrow \{0,\dots,\ell m\}. \end{aligned}$$

Permutacijo π_S imenujemo **S-škatla** in z njo zamenjamo ℓ bitov z drugimi ℓ biti.

Permutacija π_P pa permutira ℓm bitov.

Naj bo $x = (x_1, \ldots, x_{\ell m})$ binarno zaporedje, ki ga lahko smatramo za spoj m ℓ-bitnih podzaporedij označenih z $x_{(1)}, \ldots, x_{(m)}$.

SPN ima N_r krogov, v vsakem (razen zadnjem, ki je bistveno drugačen) opravimo m zamenjav z π_S in nato uporabimo še π_P. Pred vsako zamenjavo vključimo krožni ključ z XOR operacijo.

SPN šifra

$\ell, m, N_r \in \mathbb{N}$, π_S *in* π_P *permutaciji*, $\mathcal{P} = \mathcal{C} = \{0,1\}^{\ell m}$ *in* $\mathcal{K} \subseteq (\{0,1\}^{\ell m})^{N_r+1}$, *ki se sestoji iz vseh možnih razporedov ključev izpeljanih iz ključa* K *z uporabo algoritma za generiranje razporeda kjučev.*

Šifriramo z algoritmom SPN.

Alg. : *SPN*$(x, \pi_S, \pi_P, (K^1, \ldots, K^{N_r+1}))$

$w^0 := x$

for $r := 1$ **to** $N_r - 1$ **do** (*krožno mešanje ključev*)

$u^r := w^{r-1} \oplus K^r$

for $i := 1$ **to** m **do** $v^r_{(i)} := \pi_S(u^r_{(i)})$

$w^r := (v^r_{\pi_P(1)}, \ldots, v^r_{\pi_P(\ell m)})$

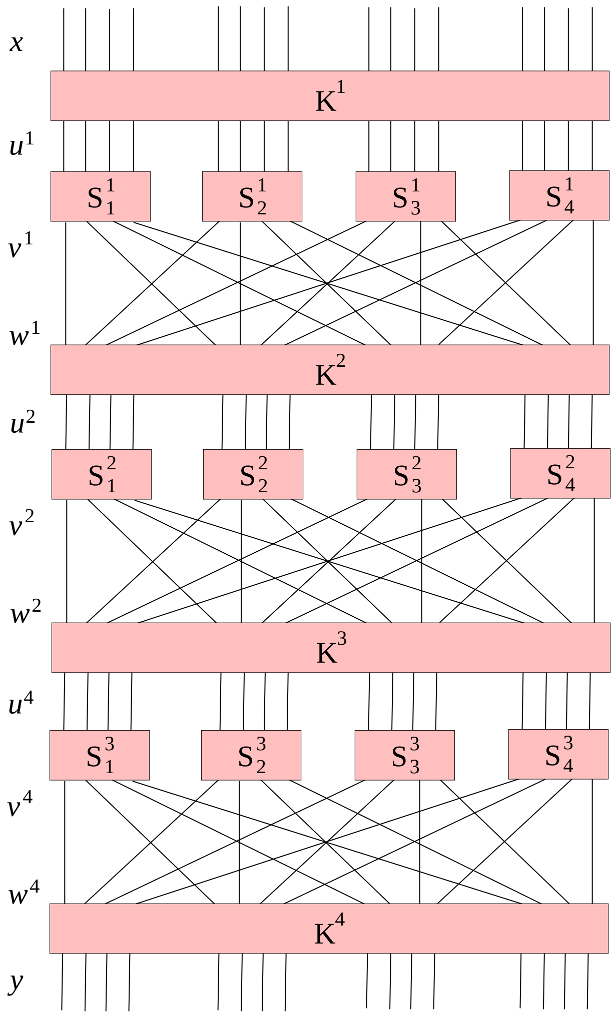
(*zadnji krog*)

$u^{N_r} := w^{N_r-1} \oplus K^{N_r}$

for $i := 1$ **to** m **do** $v^{N_r}_{(i)} := \pi_S(u^{N_r+1}_{(i)})$

$y := v^{N_r} \oplus K^{N_r+1}$

output (y)


x
K^1
u^1
S_1^1 S_2^1 S_3^1 S_4^1
v^1
w^1
K^2
u^2
S_1^2 S_2^2 S_3^2 S_4^2
v^2
w^2
K^3
u^4
S_1^3 S_2^3 S_3^3 S_4^3
v^4
w^4
K^4
y

Primer: naj bo $\ell = m = N_r = 4$, permutaciji π_S in π_P pa podani s tabelami:

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

ter

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

Naj bo ključ $K = (k_1, \dots, k_{32}) \in \{0,1\}^{32}$ definiran z

$$K = \texttt{0011 1010 1001 0100 1101 0110 0011 1111},$$

sedaj pa izberimo še razpored ključev tako, da je za $1 \le r \le 5$, krožni ključ K^r izbran kot 16 zaporednih bitov ključa K z začetkom pri k_{4r-3}:

$$\begin{aligned}
K^1 &= 0011\ 1010\ 1001\ 0100\\
K^2 &= 1010\ 1001\ 0100\ 1101\\
K^3 &= 1001\ 0100\ 1101\ 0110\\
K^4 &= 0100\ 1101\ 0110\ 0011\\
K^5 &= 1101\ 0110\ 0011\ 1111
\end{aligned}$$

Potem šifriranje čistopisa

$$x = 0010\ 0110\ 1011\ 0111$$

poteka v naslednjem vrstnem redu.

$$
\begin{aligned}
w^0 &= 0010\ 0110\ 1011\ 0111, & K^1 &= 0011\ 1010\ 1001\ 0100\\
u^1 &= 0001\ 1100\ 0010\ 0011, & v^1 &= 0100\ 0101\ 1101\ 0001\\
w^1 &= 0010\ 1110\ 0000\ 0111, & K^2 &= 1010\ 1001\ 0100\ 1101\\
u^2 &= 1000\ 0111\ 0100\ 1010, & v^2 &= 0011\ 1000\ 0010\ 0110\\
w^2 &= 0100\ 0001\ 1011\ 1000, & K^3 &= 1001\ 0100\ 1101\ 0110\\
u^3 &= 1101\ 0101\ 0110\ 1110, & v^3 &= 1001\ 1111\ 1011\ 0000\\
w^3 &= 1110\ 0100\ 0110\ 1110, & K^4 &= 0100\ 1101\ 0110\ 0011\\
u^4 &= 1010\ 1001\ 0000\ 1101, & v^4 &= 0110\ 1010\ 1110\ 1001\\
K^5 &= 1101\ 0110\ 0011\ 1111, & y &= 1011\ 1100\ 1101\ 0110
\end{aligned}
$$

Možno so številne varijacije SPN šifer.

Na primer, namesto ene S-škatle lahko uporabimo različne škatle. To lahko vidimo pri DES-u, ki uporabi 8 različnih škatel.

Zopet druga možnost je uporabiti obrnljive linearne transformacije, kot zamenjavo za permutacije ali pa samo dodatek. Tak primer je AES.

Feistelova šifra

Feistelova šifra: r krogov (rund)

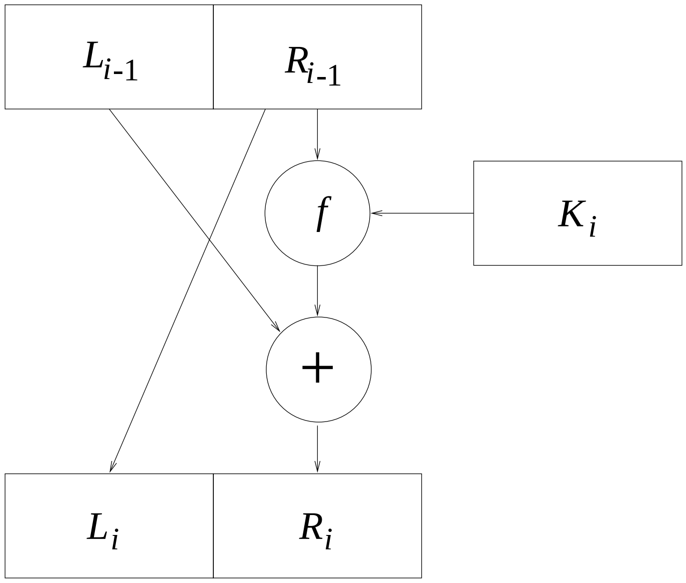
$$(L_{i-1}, R_{i-1}) \xrightarrow{K_i} (L_i, R_i).$$

kjer je $L_i = R_{i-1}$ in $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, in smo podključe K_i dobili iz osnovnega ključa K.

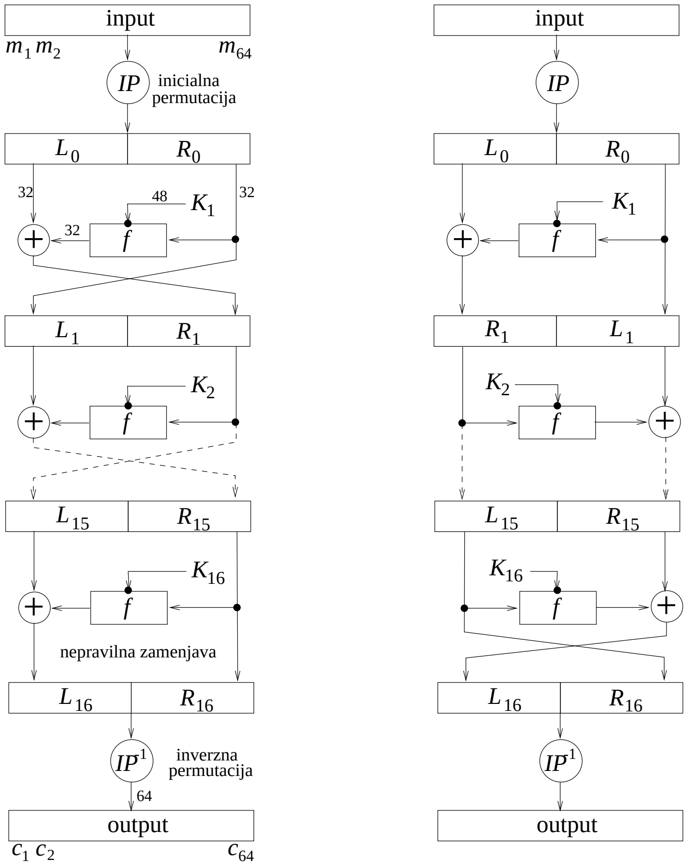
Končamo z (R_r, L_r) (in ne z (L_r, R_r)), zato je šifriranje enako odšifriranju, le da ključe uporabimo v obratnem vrstnem redu.

Funkcija f je lahko produktna šifra in ni nujno obrnljiva.

En krog



Opis šifre DES



DES-ove konstante

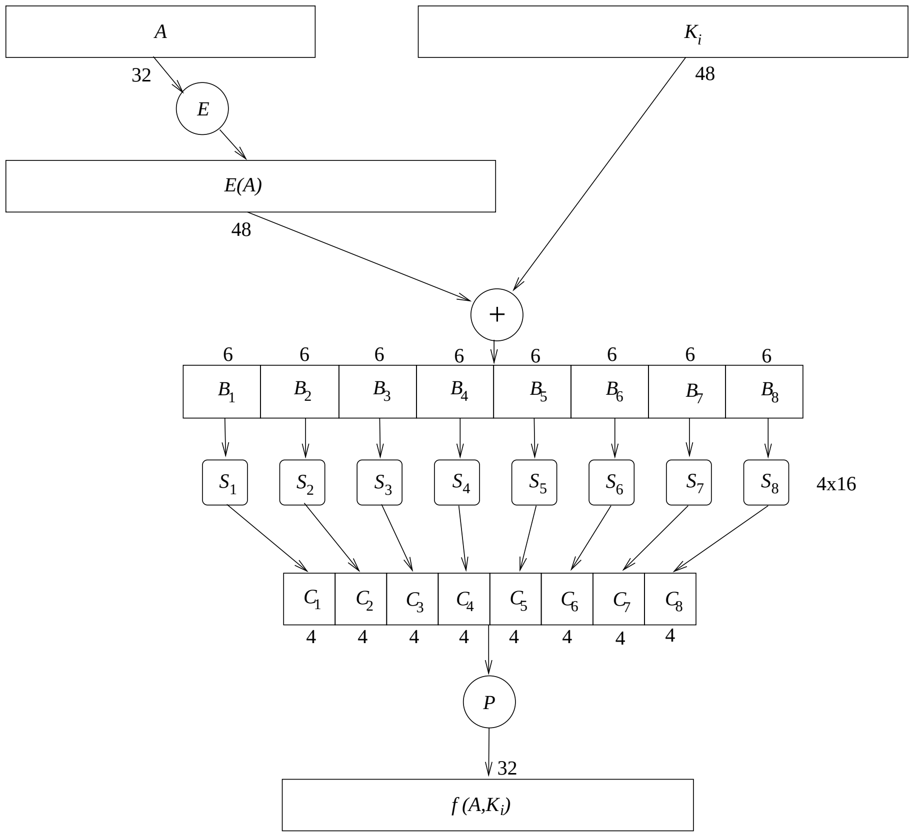
začetna in končna permutacija: IP, IP^{-1}

razširitev: E (nekatere bite ponovimo), permutacija P

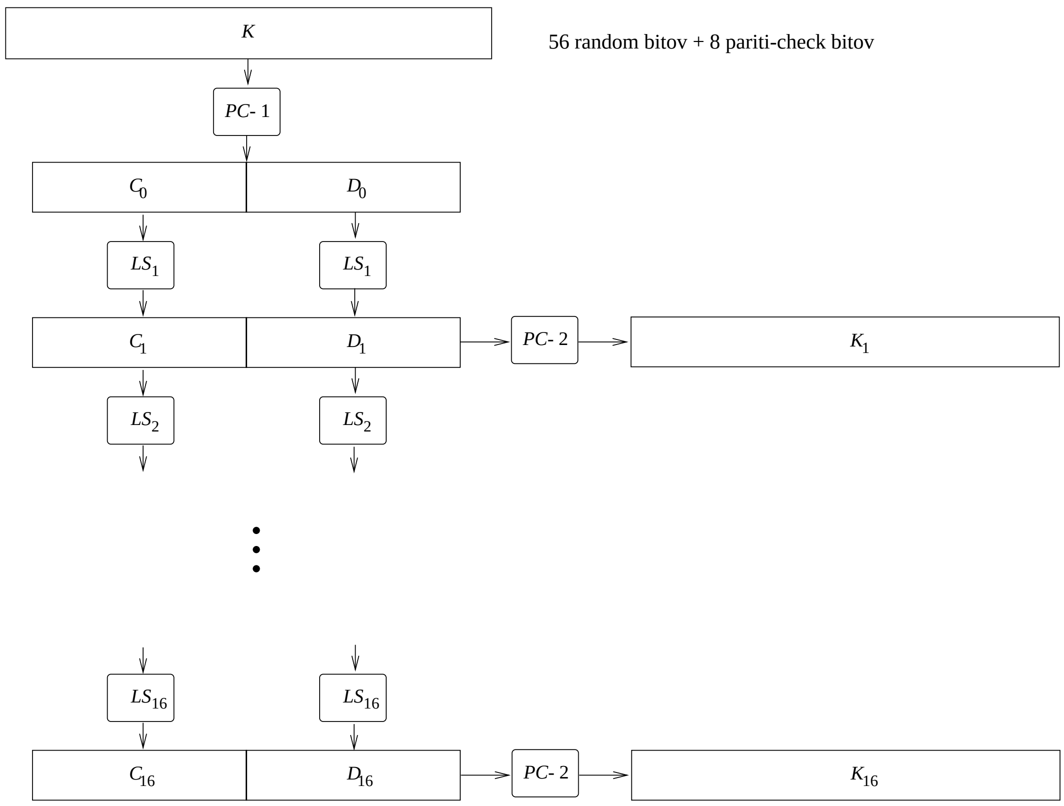
S-škatle: $S_1, S_2, \ldots, S_8$
(tabele: 4×16, z elementi $0 - 15$)

permutacije za gen. podključev: $PC-1$, $PC-2$

DES-ova funkcija



Računanje DES-ovih ključev



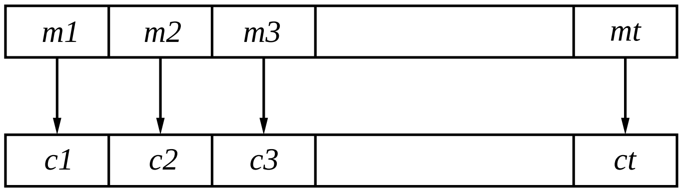
20 let je DES predstavljal delovnega konja kriptografije (bločnih šifer).

- do leta 1991 je NBS sprejel 45 hardwarskih implementacij za DES
- geslo (PIN) za bankomat (ATM)
- ZDA (Dept. of Energy, Justice Dept., Federal Reserve System)

Načini delovanja simetričnih šifer

– electronic codebook mode **(ECB)**
– cipher block chaining mode **(CBC)**
– cipher feedback mode **(CFB)**
– output feedback mode **(OFB)**

Pri **ECB** šifriramo zaporedoma blok po blok:
$c = c_1, c_2, \ldots, c_t$, kjer je $c_i = E_k(m_i)$.

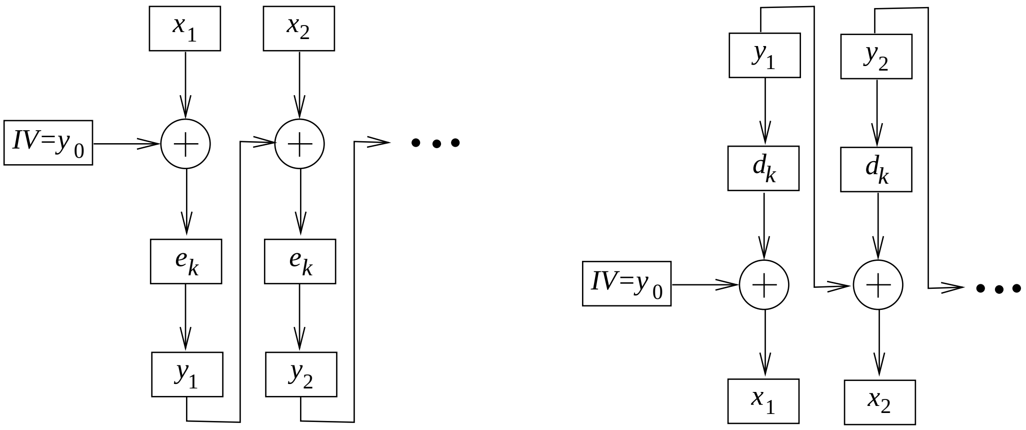


Odšifriranje: $m_i = D_k(c_i)$, $i = 1, 2, \ldots, t$.

Slabost: identični bloki čistopisa se (pri istem ključu) zašifrirajo v identične bloke tajnopisa.

Cipher Block Chaining mode – CBC

čistopis/tajnopis: 64 bitni bloki $x_1, x_2, \dots / y_1, y_2, \dots$
Šifriranje: $y_0 :=$ IV, $y_i := e_K(y_{i-1} \oplus x_i)$ za $i \geq 1$.



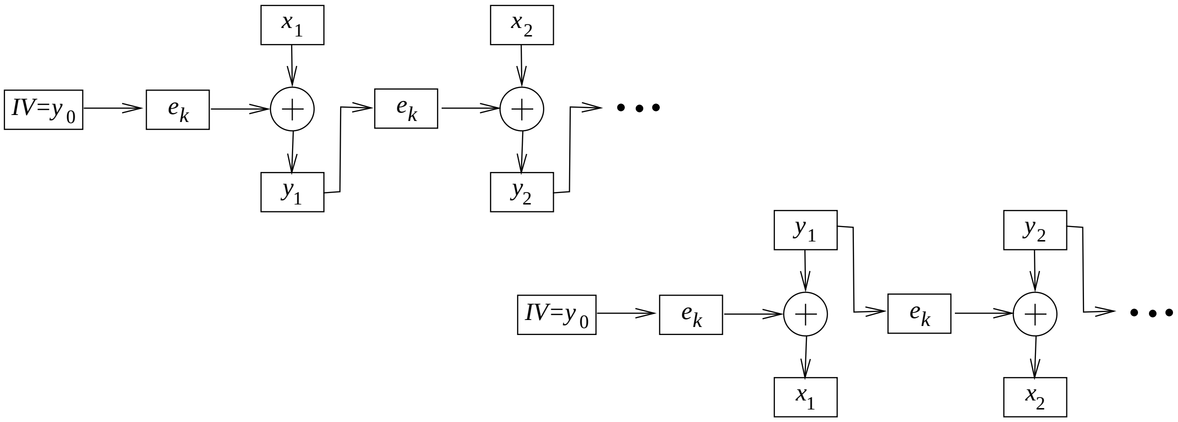
Odšifriranje: $y_0 :=$ IV, $x_i := y_{i-1} \oplus d_K(y_i)$ za $i \geq 1$.

Identičena čistopisa z različnimi IV dasta različen tajnopis. Eno-bitna napaka pri tajnopisu pokvari le odšifriranje dveh blokov.

Cipher Feedback mode – CFB

čistopis/tajnopis: 64 bitni bloki $x_1, x_2, \ldots / y_1, y_2, \ldots$

$y_0 := \mathrm{IV}$, šifriranje: $z_i := e_K(y_{i-1})$, $y_i := y_{i-1} \oplus x_i$, $i \geq 1$.



Odšifriranje ($y_0 := \mathrm{IV}$, $x_0 = d_K(\mathrm{IV})$):

$$z_i := d_K(x_{i-1}) \;\text{ in }\; x_i := y_i \oplus z_i \quad \text{za} \;\; i \geq 1.$$

CFB se uporablja za preverjanje celovitosti sporočila (angl. message authentication code - MAC).

Output Feedback mode – OFB

čistopis/tajnopis: 64 bitni bloki $x_1, x_2, \ldots / y_1, y_2, \ldots$

Inicializacija: $z_0 :=$ IV, šifriranje:

$$z_i := e_K(z_{i-1}) \text{ in } y_i := x_i \oplus z_i \quad \text{za } i \geq 1.$$

Odšifriranje:($z_0 :=$ IV)

$$z_i := e_K(z_{i-1}) \text{ in } x_i := y_i \oplus z_i \quad \text{za } i \geq 1.$$

OFB se uporablja za satelitske prenose.

Napadi na šifro DES

Požrešni napad: preverimo vseh 2^{56} ključev.

Leta 1993 Michael J. Wiener, Bell-Northern Research, Kanada, predstavi učinkovito iskanje DES ključa:

- **diferenčna kriptoanaliza** z 2^{47} izbranimi čistopisi (Biham in Shamir 1989)
  – je učinkovita tudi na nekaterih drugih bločnih šifrah,
- **linearna kriptoanaliza** z 2^{47} poznanimi čistopisi (Matsui 1993):

Slednja napada sta statistična, saj potrebujeta velike količine čistopisa in ustreznega tajnopisa, da določita ključ. Pred leti sta bila napada zanimiva le teoretično.

Wienerjev cilj je bil precizna ocena časa in denarja potrebnega za graditev čipov za iskanje DES ključa.

Požrešna metoda na prostor ključev: 2^{56} korakov je zlahka paralelizirana.

Dan je par čistopis-tajnopis (P, C) ter začetni ključ K. Registri za vsako iteracijo so ločeni, tako da je vse skupaj podobno tekočemu traku:

- hitrost 50 MHz
- cena \$10.50 na čip
- 50 milijonov ključev na sekundo
- skupaj: \$100 tisoč, 5760 čipov, rabi 35 ur

Pri linearni kriptoanalizi hranjenje parov zavzame 131,000 Gbytov. Implementirano leta 1993: 10 dni na 12 mašinah.

Po odkritju diferenčne kriptoanalize je Don Coppersmith priznal, da je IBM v resnici poznal ta napad (ne pa tudi linearno kriptoanalizo) že ko so razvijali DES:

"Po posvetovanju z NSA, smo se zavedali, da utegne objava kriterijev načrtovanja odkriti tehniko kriptoanalize. To je močno sredstvo, ki se ga da uporabiti proti mnogim tajnopisom.
To bi zmanjšalo prednost ZDA pred drugimi na področju kriptografije."

Novejši rezultati napadov

DES izivi pri RSA Security (3 poznani PT/CT pari):

T h e u n k n	o w n m e s s	a g e i s :	? ? ? ? ? ? ? ?

junij 1997: razbito z internetnim iskanjem (3m).

julij 1998: razbito v treh dneh z DeepCrack mašino (1800 čipov; \$250,000).

jan. 1999: razbita v 22 h, 15 min (DeepCrack + porazdeljena.mreža).

V teku (porazdeljena.mreža): RC5 – 64-bitni izziv:

- pričeli konec 1997; trenutna hitrost: 2^{36} ključev/sec (2^{25} secs/leto; pričakovani čas: ≤ 8 let).

Implementacijski napadi na DES

Napadi s pomočjo diferenčne analize porabe moči
(angl. differential power analysis (**DPA**) attacks):

- Kocher, Jaffe, Jun 1999,
- procesorjeva poraba moči je odvisna od instrukcij,
- merimo porabo moči inštrukcij, ki se izvedejo v 16-ih krogih DES-a
- ≈1000 tajnopisa zadoščajo za odkritje tajnega ključa.

Napadi s pomočjo diferenčne analize napak
(angl. differential fault analysis (**DFA**) attacks):

- Biham, Shamir 1997.
- napad: zberi naključne napake v 16-ih krogih DES-a.
- ≈200 napačnih odšifriranj zadošča za razkritje tajnega ključa.

Vse o napadih je veljalo za ECB način.

Isti čipe se da uporabiti tudi za druge načine, cena in čas pa se nekoliko povečata. Recimo po Wienerju za CBC način rabimo \$1 milijon in 4 ure.

Varnost DES-a lahko enostavno povečamo, če uporabimo **3-DES** (zakaj ne 2-DES?).

$$\boxed{\begin{array}{c}\mathrm{DES_E}(\mathrm{P},\mathrm{K_1})\longrightarrow \mathrm{DES_D}(\mathrm{DES_E}(\mathrm{P},\mathrm{K_1}),\mathrm{K_2})\\ \longrightarrow \mathrm{DES_E}(\mathrm{DES_D}(\mathrm{DES_E}(\mathrm{P},\mathrm{K_1}),\mathrm{K_2}),\mathrm{K_3})\end{array}}$$

Za $K_1 = K_2 = K_3$ dobimo običajni DES.

Običajno pa zamenjamo K_3 s K_1 in dobimo približno za faktor 10^{13} močnejši sistem.

Kako veliko je VELIKO?

sekund v enem letu	$\approx 3 \times 10^7$
(živimo “le” 2-3 milijarde sekund)	
starost našega sončnega sistema (v letih)	$\approx 6 \times 10^9$
urinih ciklov na leto (200 MHz)	$\approx 6.4 \times 10^{15}$
01-zaporedij dolžine 64	$\approx 2^{64} \approx 1.8 \times 10^{19}$
01-zaporedij dolžine 128	$\approx 2^{128} \approx 3.4 \times 10^{38}$
01-zaporedij dolžine 256	$\approx 2^{256} \approx 1.2 \times 10^{77}$
75 številčnih praštevil	$\approx 5.2 \times 10^{72}$
elektronov v vsem vesolju	$\approx 8.37 \times 10^{77}$

mega (M)	giga (G)	tera (T)	peta (P)	exa (E)
10^6	10^9	10^{12}	10^{15}	10^{18}

Računska moč

za naše potrebe bomo privzeli, da se smatra:

- 2^{40} operacij za *lahko*,
- 2^{56} operacij za *dosegljivo*,
- 2^{64} operacij za *komaj da dosegljivo*,
- 2^{80} operacij za *nedosegljivo*,
- 2^{128} operacij za *popolnoma nedosegljivo*.

3-DES je trikrat počasnejši od DES-a.

To je pogosto nesprejemljivo, zato je leta 1984 Ron Rivest predlagal **DESX**:

$$\boxed{\mathrm{DESX}_{\mathrm{k.k1.k2}}(\mathrm{x}) = \mathrm{k2} \oplus \mathrm{DES}_{\mathrm{k}}(\mathrm{k1} \oplus \mathrm{x}).}$$

DESX ključ $K = k.k1.k2$ ima

$$56 + 64 + 64 = 184 \text{ bitov.}$$

DESX trik onemogoči preizkušanje vseh mogočih ključev (glej P. Rogaway, 1996).
Sedaj rabimo več kot 2^{60} izbranega čistopisa.

Hitrost

Preneel, Rijmen, Bosselaers 1997.

Softwarski časi za implementacijo na 90MHz Pentiumu.

šifra	velikost ključa (biti)	hitrost
DES	56	10 Gbits/sec (ASIC chip)
DES	56	16.9 Mbits/sec
3DES	128	6.2 Mbits/sec
RC5-32/12	128	38.1 Mbits/sec
Arcfour	variable	110 Mbits/sec

Opis šifre AES

Dolžina blokov je 128 bitov, ključi imajo tri možne dolžine: 128 ($N_r = 10$), 192 ($N_r = 12$) in 256 ($N_r = 14$),

1. *Za dan čistopis x, inicializira* `State` *z x in opravi* ADDROUNDKEY, *ki z operacijo XOR prišteje* `RoundKey` *k* `State`.
2. *Za vsak od $N_r - 1$ krogov, opravi na* `State` *zaporedoma zamenjavo* SUBBYTES, *operaciji* SHIFTROWS *in* MIXCOLUMNS *ter izvede* ADDROUNDKEY.
3. *Naredi* SUBBYTES, SHIFTROWS *in* ADDROUNDKEY.
4. *Za tajnopis y definira* `State`.

Vse operacije v AES so opravljene s pomočjo besed in vse spremenljivke so sestavljene iz določenega števila besed.

Čistopis x je sestavljen iz 16-ih besed: $x_0, \ldots, x_{15}$.

`State` je sestavljen iz (4×4)-dim. matrike besed:

$$\begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{pmatrix}.$$

`State` dobi vrednosti iz x na naslednji način:

$$\begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{pmatrix} := \begin{pmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{pmatrix}.$$

Na vsako besedo bomo gledali kot na dve šestnajstiški števili.

Operacija SubBytes deluje kot zamenjava, permutacija π_S $\{0,1\}^8$, na vsaki besedi od `State` posebej, z uporabo S-škatel.

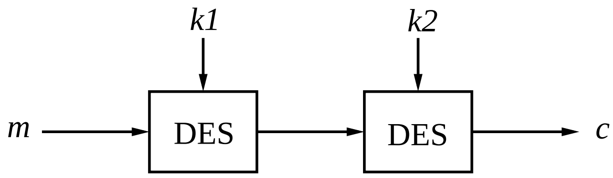
Druge simetrične šifre:

MARS, RC6, Serpent, Twofish
FEAL, IDEA, SAFER,
RC2, RC4, RC5,
LOKI, CAST, 3WAY,
SHARK, SKIPJACK,
GOST, TEA, ...

Dvojno šifriranje

2-DES: ključ $k = (k_1, k_2), \quad k_1, k_2 \in_R \{0, 1\}^{56}$.

Šifriranje: $c = \mathrm{DES}_{k_2}(\mathrm{DES}_{k_1}(m))$.



Odšifriranje: $m = \mathrm{DES}^{-1}_{k_1}(\mathrm{DES}^{-1}_{k_2}(c))$.

Dolžina ključa 2-DES-a je 112, torej za požrešno metodo potrebujemo 2^{112} korakov (nemogoče).

Opomba: dolžina blokov se ni spremenila.

Diferenčna kriptoanaliza

- požrešna metoda in metoda z urejeno tabelo
- diferenčna metoda (za 1, 3, 6 in 16 ciklov)

Bločni tajnopisi s simetričnim ključem
se ne uporabljajo samo za šifriranje, temveč tudi za konstrukcijo generatorjev psevdonaključnih praštevil, tokovnih tajnopisov, MAC in hash-funkcij.

1. **Požrešni napad**: preverimo vseh 2^{56} ključev (ne potrebujemo spomina).
2. Sestavimo **urejeno tabelo** $(e_K(x), K)$ za vseh 2^{56} ključev K in poiščemo v njej tak K, da je $y = e_K(x)$. Iskanje y-a je hitro, saj je tabela urejena.

   Ta metoda je praktična samo, če lahko večkrat uporabimo to tabelo.

Danes poznamo dva močna napada na DES: **diferenčno** kriptoanalizo in **linerno** kriptoanalizo.

Oba sta statistična, saj potrebujeta velike količine čistopisa in ustreznega tajnopisa, da določita ključ in zato nista praktična.

Zelo uspešna pa sta pri manjšem številu ciklov, npr. DES z 8imi cikli lahko razbijemo z diferenčno kriptoanalizo v nekaj minutah že na osebnem računalniku.

Diferenčno kriptoanalizo sta v letih 1990 in 1991 vpeljala Eli Biham in Adi Shamir (**izbran čistopis**).

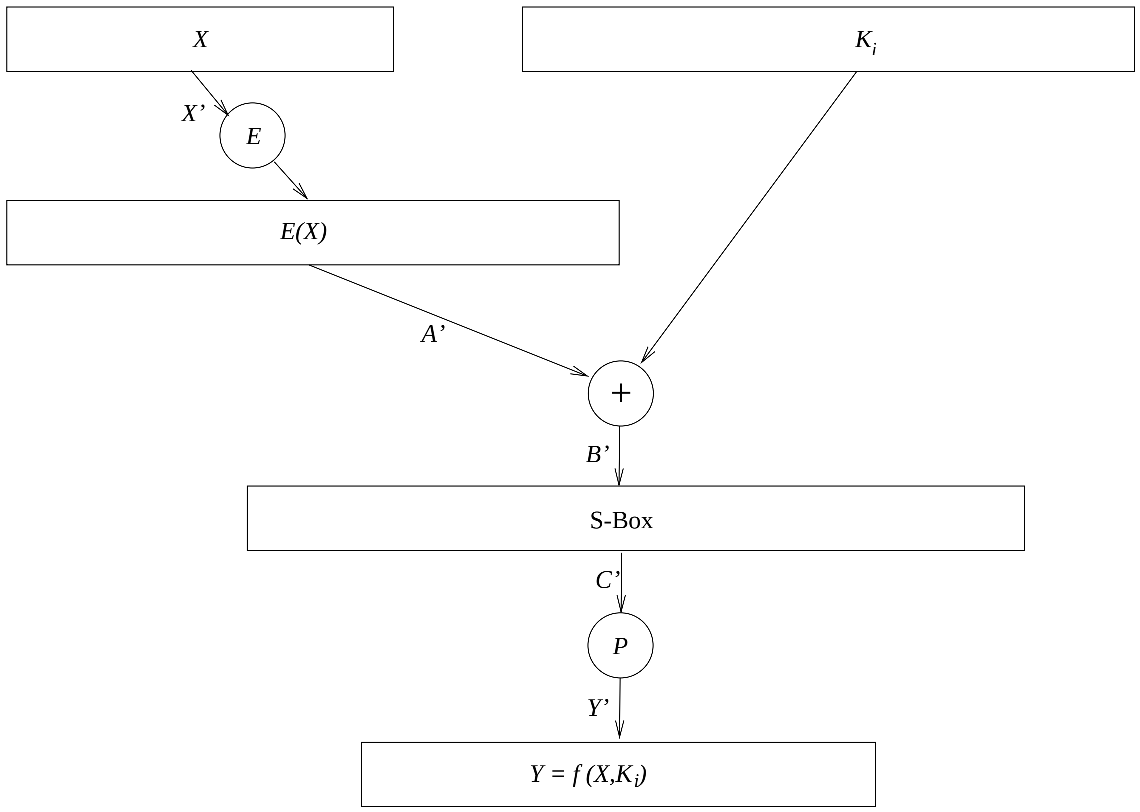
Oglejmo si pare tajnopisa za katere ima čistopis določene razlike. Diferenčna kriptoanaliza spremlja spreminjanje teh razlik, ko gre čistopis skozi nekaj ciklov DES-a in je šifriran z istim ključem.
Če poenostavimo, ta tehnika izbere pare čistopisa s fiksno razliko (čistopis je lahko izbran naključno).

Z uporabo razlik tajnopisa določimo verjetnosti različnih ključev. Analiza mnogih parov tajnopisa nam na koncu da najbolj verjeten ključ.

Naj bosta X in X^* par čistopisov z razliko X'. Tajnopisa Y in Y^* poznamo, zato poznamo tudi njuno razliko Y'. Naj bo $A^{(*)} := E(X^{(*)})$ in $P(C^{(*)}) = Y^{(*)}$.

Ker poznamo tudi razširitev E ter permutacijo P, poznamo A' in C' (glej sliko). $B^{(*)} = A^{(*)} \oplus K_i$ ne poznamo, vendar je njuna razlika B' enaka razliki A'.

Trik je v tem, da za dano razliko A' niso enako verjetne vse razlike C'. Kombinacija razlik A' in C' sugerira vrednosti bitov izrazov $A \oplus K_i$ in $A^* \oplus K_i$. Od tod pa s pomočjo A in A^* dobimo informacije o ključu K_i.


X
K_i
X'
E
E(X)
A'
+
B'
S-Box
C'
P
Y'
$Y = f(X, K_i)$

V primeru, ko imamo več kot en cikel, si pomagamo z določenimi razlikami, ki jih imenujemo **karakteristike**. Le-te imajo veliko verjetnost, da nam dajo določene razlike tajnopisa ter se razširijo, tako da definirajo pot skozi več ciklov.

Poglejmo si zadnji cikel DES-a
(začetno in končno permutacijo lahko ignoriramo).
Če poznamo K_{16} poznamo 48 bitov originalnega ključa. Preostalih 8 bitov dobimo s požrešno metodo.
Diferenčna kriptoanaliza nam da K_{16}.

Podrobnosti:

Škatla S_i oziroma funkcija $S_i\ :\ \{0,1\}^6 \longrightarrow \{0,1\}^4$ ima za elemente cela števila z intervala $[0, 15]$:

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
S_1:	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Naj bo $B_j = b_1 b_2 b_3 b_4 b_5 b_6$.
$S_i(B_j)$ določimo na naslednji način.

Bita $b_1 b_6$ določita vrstico v, biti $b_2 b_3 b_4 b_5$ pa stolpec s v tabeli S_i, katere (v, s)-ti element je $S_i(B_j) \in \{0,1\}^4$

Za razliko $B'_j \in (\mathbb{Z}_2)^6$ definiramo množico z 2^6 elementi: $\Delta(B'_j) := \{(B_j, B_j \oplus B'_j) \mid B_j \in (\mathbb{Z}_2)^6\}$

Primer: oglejmo si škatlo S_1 in naj bo $B'_j = 110100$ razlika (XOR) vhodov.

$\Delta(110100) =$ `{(000000,110100),(000001,110101),...,(111111,001011)}`

Za vsak urejen par izračunamo razliko izhoda iz S_1:

npr. $S_1(000000) = 1110$ in $S_1(110100) = 1001$
$\Longrightarrow$ razlika izhodov $C'_j = 0111$.

Tabela izhodnih razlik C_j' in možnih vhodov B_j za vhodno razliko $B_j' = 110100$:

```
0000  -
0001  8  000011, 001111, 011110, 011111, 101010, 101011, 110111, 111011
0010 16  000100, 000101, 001110, 010001, 010010, 010100, 100101, 011011,
         100000, 100101, 010110, 101110, 101111, 110000, 110001, 111010
0011  6  000001, 000010, 010101, 100001, 110101, 110110
0100  2  010011, 100111
0101  -
0110  -
0111 12  000000, 001000, 001101, 010111, 011000, 011000, 011101, 100011,
         101001, 101100, 110100, 111001, 111100
1000  6  001001, 001100, 011001, 101101, 111000, 111101
1001  -
1010  -
1011  -
1100  -
1101  8  000110, 010000, 010110, 011100, 100010, 100100, 101000, 110010
1110  -
1111  6  000111, 001010, 001011, 110011, 111110, 111111
```

Tabela izhodnih razlik in porazdelitev vhodov za vhodno razliko 110100 (števila morajo biti soda, zakaj?):

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	8	16	6	2	0	0	12	6	0	0	0	0	8	0	6

Pojavi se samo 8 od 16ih možnih izhodnih vrednosti.

Če pregledamo vse možnosti (za vsako škatlo S_i in vsako razliko), se izkaže, da je povprečno zastopanih samo 75-80% možnih razlik izhodov.

Ta neenakomerna porazdelitev je osnova za diferenčni napad.

Za vsako škatlo S_j (8 jih je) in za vsako vhodno razliko (2^6 jih je) sestavimo tako tabelo (skupaj 512 tabel).

Velja povdariti, da vhodna razlika ni odvisna od ključa K_i (saj smo že omenili, da je $A' = B'$), zato pa izhodna razlika C' *je* odvisna od ključa K_i.

Naj bo $A = A_1 \dots A_8$, $C = C_1 \dots C_8$ in $j \in \{1, \dots, 8\}$.

Potem poiščemo razliko $(C')_j$ v tabeli za S_j in $(A')_j$, ki nam določi vse možne vhode B_j iz katerih izračunamo vse $B_j \oplus A_j$, ki morajo vsebovati $(K_i)_j$.

Tako smo dobili nekaj kandidatov za $(K_i)_j$.

Primer: $A_1 = 000001$, $A_1^* = 110101$ in $C_1' = 1101$. Potem dobimo 13-to vrstico iz Tabele 1, ki vsebuje 8 elementov (torej smo zožili število možnosti iz $2^6 = 64$ na 8).

Z naslednjim parom čistopisa dobimo nove kandidate, $(K_i)_j$ pa leži v preseku novih in starih kandidatov...