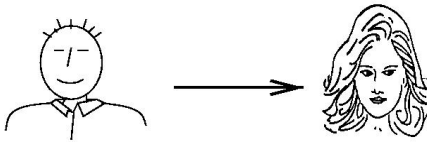
## Koncept kriptografije javnih ključev

Bojan pošlje Aniti pismo, pri tem pa si želi, da bi pismo lahko prebrala le ona (in prav nihče drug) **[zaščita]**.



Anita pa si poleg tega želi biti prepričana, da je pismo, ki ga je poslal Bojan prišlo prav od njega **[podpis]**.





*Predpostavimo*, da se Anita in Bojan prej dogovorita za **skupen ključ**, ki ga ne pozna nihče drug (simetričen kriptosistem).

Če Bojan z njim zašifrira pismo, je lahko prepričan, da ga lahko odklene le Anita.

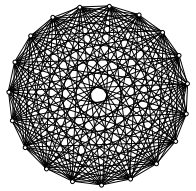
Hkrati pa je tudi Anita zadovoljna, saj je prepričana, da ji je pismo lahko poslal le Bojan.





Tak pristop je problematičen vsaj iz dveh razlogov:

1. Anita in Bojan se morata *prej* dogovoriti za skupen ključ,
2. upravljanje s ključi v omrežju z $n$ uporabniki je kradratne zahtevnosti $\binom{n}{2}$, vsak uporabnik pa mora hraniti $n-1$ ključev.

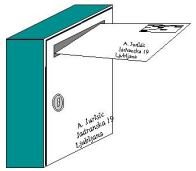






Leta 1976 sta Whit **Diffie** in Martin **Hellman** predstavila koncept kriptografije z javnimi ključi.

Tu ima za razliko od sim. sistema vsak uporabnik **dva** ključa, podatke *zaklepa*, drugi pa jih *odklepa*.



Pomembna lastnost tega sistema:
ključ, ki zaklepa, ne more odklepati
in obratno,
ključ, ki odklepa, ne more zaklepati.
To omogoči lastniku, da en ključ objavi, drugega pa hrani v tajnosti (npr. na pametni kartici). Zato imenujemo ta ključa zaporedoma **javni** in **zasebni**.





Ta pristop omogoča veliko presenetljivih načinov uporabe, npr. omogoča ljudem varno komuniciranje, ne da bi se predhodno srečali zaradi izmenjave/dogovora o tajnem ključu.

Vsak uporabnik najprej objavi svoj javni ključ, zasebnega pa zadrži zase.

Vsak lahko nato z javnim ključem zašifrira pismo, bral (odšifriral) pa ga bo lahko le lastnik ustreznega zasebnega ključa.

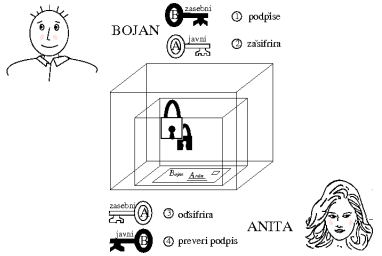




Bojan pošlje Aniti podpisano zasebno pismo:

(1) podpiše ga s svojim zasebnim ključem $Z_B$ in ga

(2) zašifrira z Anitinim javnim ključem $J_A$.




(3) Anita ga s svojim zasebnim ključem $Z_A$ odšifrira,

(4) z Bojanovim javnim ključem $J_B$ preveri podpis.

V razvoju javne kriptografije je bilo predlaganih
in razbitih veliko kriptosistemov.

Le nekaj se jih je obdržalo in jih lahko danes
smatramo za varne in učinkovite.

Glede na matematični problem na katerem temeljijo,
so razdeljene v tri skupine:

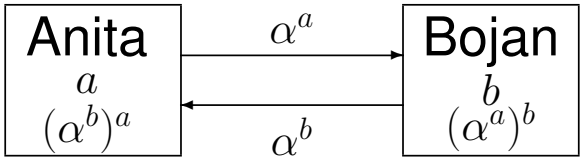
- **Sistemi faktorizacije celih števil**
  npr. RSA (Rivest-Shamir-Adleman).
- **Sistemi diskretnega logaritma**
  npr. DSA.
- **Kripto sistemi z eliptičnimi krivuljami**
  (Elliptic Curve Cryptosystems)





**Izmenjava ključev oziroma dogovor o ključu**

(Diffie-Hellman - DH)




Anita in Bojan si delita skupni element grupe: $\alpha^{ab}$.

Končne grupe so zanimive zato, ker računanje potenc lahko opravimo učinkovito, ne poznamo pa vedno učinkovitih algoritmov za logaritem (za razliko od $\mathbb{R}$).





**Kaj je kriptografija**

- cilji kriptografije
- širši pogled na kriptografijo
- gradniki kriptografije

Osnovna motivacija za naš študij je
uporaba kriptografije v realnem svetu.

Cilje kriptografije bomo dosegali z matematičnimi sredstvi.





**Cilji kriptografije**

1. **Zasebnost/zaupnost/tajnost:**
   varovanje informacij pred tistimi, ki jim vpogled ni dovoljen,
   dosežemo s šifriranjem.
2. **Celovitost podatkov:**
   zagotovilo, da informacija ni bila spremenjena
   z nedovoljenimi sredstvi (neavtoriziranimi sredstvi).
3. **Overjanje sporočila (ali izvora podatkov):**
   potrditev izvora informacij.
4. **Identifikacija:**
   potrditev identitete predmeta ali osebe.





5. **Preprečevanje tajenja:**
   preprečevanje, da bi nekdo zanikal
   dano obljubo ali storjeno dejanje.

6. **Drugi kriptografski protokoli:**

1. grb/cifra po telefonu
2. mentalni poker
3. shema elektronskih volitev
   (anonimno glasovanje brez goljufanja)
4. (anonimni) elektronski denar





**Cilji kriptografije**:

1. *zasebnost/zaupnost/tajnost*
2. *celovitost podatkov*
3. *overjanje sporočila (ali izvora podatkov)*
4. *identifikacija*
5. *preprečevanje nepriznavanja*
6. *drugi kriptografski protokoli*

**NAUK: Kriptografija je več kot samo širfiranje (enkripcija).**

**Širši pogled na kriptografijo**

**– varnost informacij**

Kriptografija je sredstvo, s katerim dosežemo varnost informacij, ki med drugim zajema:

**(a) Varnost računalniškega sistema**

tj. tehnična sredstva, ki omogočajo varnost računalniškega sistema, ki lahko pomeni samo en računalnik z več uporabniki, lokalno mrežo (LAN), Internet, mrežni strežnik, bankomat, itd.





Med drugim obsega:

- varnostne modele in pravila, ki določajo zahteve po varnosti, katerim mora sistem ustrezati
- varen operacijski sistem
- zaščito pred virusi
- zaščito pred kopiranjem
- kontrolne mehanizme (beleženje vseh aktivnosti, ki se dogajajo v sistemu lahko omogoči *odkrivanje* tistih kršitev varnostnih pravil, ki jih ni mogoče preprečiti)
- analiza tveganja in upravljanje v primeru nevarnosti





**(b) Varnost na mreži**

Zaščita prenašanja podatkov preko komercialnih mrež, tudi računalniških in telekomunikacijskih.

Med drugim obsega:

- protokole na internetu in njihovo varnost
- požarne zidove
- trgovanje na internetu
- varno elektronsko pošto





**Širši pogled na kriptografijo**
**– varnost informacij**

1. varnost računalniškega sistema
2. varnost na mreži

**NAUK: Kriptografija je samo majhen del varnosti informacij.**





**Gradniki kriptografije**

1. **matematika** *(predvsem teorija števil)*
2. **računalništvo** *(analiza algoritmov)*
3. **elektrotehnika** *(hardware)*
4. **poznavanje aplikacij** *(finance,...)*
5. **politika** *(restrikcije, key escrow, NSA,...)*
6. **pravo** *(patenti, podpisi, jamstvo,...)*
7. **družba** *(npr. enkripcija omogoča zasebnost, a otežuje pregon kriminalcev)*

**NAUK: Uporabna kriptografija je več kot samo zanimiva matematika.**





# 1. Klasična kriptografija

- zgodovina (hieroglifi, antika, II. svetovna vojna)
- zamenjalna šifra

**Klasične šifre in razbijanje**

- prikrita, zamenjalna (zamična, afina), bločna (Vigenerjeva, Hillova)
- Kerckhoffov princip in stopnje napadov
- napad na Vigenerja (Kasiski test, indeks naključja)
- napad na Hillovo šifro
- tokovne šifre

## Zgodovina

Kriptografija ima dolgo in zanimivo zgodovino:

– Hieroglifi, Špartanci, Cezar, ...

D. Kahn, **The Codebreakers** (The Story of Secret Writing),

hrvaški prevod: (K. and M. Miles), **Šifranti protiv špijuna**,
Centar za informacije i Publicitet, Zagreb 1979.
(429+288+451+325=1493 strani).





## Hieroglifi

Razvili so jih antični Egipčani.
Komunicirali so v jeziku
sestavljenemu iz sličič namesto besed.

Najbolj izobraženi ljudje so jih razumeli,
toda v religioznemu kontekstu
– **npr. napisi na grobovih** –
so njihovi duhovniki uporabljali tajne
kriptografske verzije znakov, da bi bila vsebina
več vredna (saj je šlo za božje besede) in bolj mistična.

Mnoge religije so uporabljale tajne znake,
ki so jih razumeli le določeni izbranci.





## Razbijalci šifer

obstajajo od kar poznamo šifriranje.

L. 1799 so v Egipčanski Rosetti
našli skoraj 2.000 let star kamen.
Na njemu so bili trije teksti:

- hieroglifi,
- pisava egiptčanov (demotic) in
- starogrščina.

Ko je bil končan prevod iz Grščine,
je bilo možno razvozlati tudi hieroglife,
iz katerih smo izvedeli o zgodovini antičnega Egipta.

DEL KAMNA IZ ROSETTE,
NA KATEREM JE BILA
NEZNANA PISAVA,
DOKLER JE ARHEOLOGI
NISO ODŠIFRIRALI.





## Še ena antična: o obriti glavi

Medtem, ko je bil genialni Histius na perzijskem sodišču, je hotel obvestiti Aristagorasa iz Grčije, da dvigne upor. Seveda je bilo pomembno, da nihče ne prestreže sporočilo.

Da bi zagotovil tajnost, je Histius obril sužnja, ki mu je nabolj zaupal, mu vtetoviral na glavo sporočilo [sužnju so rekli, da mu začenjajo zdraviti slepoto] in počakal, da mu zrastejo lasje.

Sužnju je bilo ukazano, da reče Aristagorasu:

"*Obrijte mojo glavo in poglejte nanjo.*"

Aristagoras je nato zares dvignil upor.





To je primer **prikrite šifre**,
sporočilo je prisotno, a na nek način prikrito.

Poznamo mnogo takšnih primerov.

Varnost takega sporočila je odvisna od trika
prikrivanja.

Tak trik je lahko odkriti, poleg tega pa ne omogoča hitrega šifriranja in odšifriranja.

To ne pride vpoštev za **resno uporabo**.





**Anglija: Sir John dobi sporočilo:** Worthie Sir John:- Hope, that is ye beste comfort of ye afflicted, cannot much, I fear me, help you now. That I would saye to you, is this only: if ever I may be able to requite that I do owe you, stand not upon asking me. 'Tis not much that I can do: but what I can do, bee ye verie sure I wille. I knowe that, if dethe comes, if ordinary men fear it, it frights not you, accounting it for a high honor, to have such a rewarde of your loyalty. Pray yet that you may be spared this soe bitter, cup. I fear not that you will grudge any sufferings; only if bie submission you can turn them away, 'tis the part of a wise man. Tell me, an if you can, to do for you anythinge that you wolde have done. The general goes back on Wednesday. Restinge your servant to command. - R.T.

Če vam uspe “med vrsticami” prebrati:

**PANEL AT EAST END**
**OF CHAPEL SLIDES**

verjetno ne boste občutili enakega olajšanja kot Sir John Trevanion, njemu pa je vsekakor uspelo pobegniti, sicer bi ga v gradu Colcester gotovo usmrtili prav tako, kot so Sir Charlesa Lucasa ter Sir Georga Lislea.





## Druga svetovna vojna

– Enigma (Nemčija),

– Tunny (Nemčija),

– Purple (Japonska),

– Hagelin (ZDA).





## Zamenjalna šifra

Tomaž Pisanski, Skrivnostno sporočilo
Presek V/1, 1977/78, str. 40-42.

```
YHW?HD+CVODHVTHVO-!JVG:CDCYJ(JV/-V?HV(
-T?HVW-4YC4(?-DJV/-(?S-VO3CWC%J(-V4-DC
V!CW-?CVNJDJVD-?+-VO3CWC%J(-VQW-DQ-VJ+
V?HVDWHN-V3C:CODCV!H+?-DJVD-?+CV3JO-YC
```

(črko Č smo zamenjali s C, črko Ć pa z D)





Imamo 26! = 403291461126656355840000000
možnosti z direktnim preizkušanjem,
zato v članku dobimo naslednje nasvete:

(0) Relativna frekvenca črk in presledkov v slovenščini: presledek 173,

```
E  A  I  O  N  R  S  L  J  T  V  D
89 84 74 73 57 44 43 39 37 37 33 30

K  M  P  U  Z  B  G  "C   H  "S  C  "Z  F
29 27 26 18 17 15 12 12  9  9  6  6  1
```





(1) Na začetku besed so najpogostejše črke
N, S, K, T, J, L.

(2) Najpogostejše končnice pa so
E, A, I, O, U, R, N.

(3) Ugotovi, kateri znaki zagotovo predstavljajo samoglasnike
in kateri soglasnike.

(4) V vsaki besedi je vsaj en samoglasnik
ali samoglasniški R.

(5) V vsaki besedi z dvema črkama je ena
črka samoglasnik, druga pa soglasnik.

(6) detektivska sreča





```
(0)  V  -  C  D  J  ?  H  W  O  (  +  3
    23 19 16 12 11 10  9  7  6  6  5  4

     Y  4  !  /  Q  :  %  T  N  S  G
     4  3  3  2  2  2  2  2  2  1  1
```

Zaključek `V --> ' '` (drugi znaki z visoko
frekvenco ne morejo biti).

Dve besedi se ponovita: `03CWC%J(-`,
opazimo pa tudi eno sklanjatev:
`D-?+-` ter `D-?+C`.

Torej nadaljujemo z naslednjim tekstom:

```
YHW?HD+C ODH TH O-!J G:CDCYJ(J /- ?H
(-T?H W-4YD4(?-DJ /-(?S- 03CWC%J(- 4-DC
!CW-?C NJDJ D-?+- 03CWC%J(- QW-DQ- J+
?H DWHN- 3C:C0DC !H+?-DJ D-?+C 3J0-YC
```





(3) Kanditati za samoglasnike e,a,i,o so znaki z visokimi frekvancami.
Vzamemo:

$$\{e,a,i,o\} = \{-,C,J,H\}$$

(saj D izključi -,H,J,C in ? izključi -,H,C,
znaki -,C,J,H pa se ne izključujejo)

Razporeditev teh znakov kot samoglasnikov izgleda prav verjetna.
To potrdi tudi gostota končnic, gostota parov je namreč:

```
AV CV HV JV VO ?H -D DC JM W- DJ UC CW -? VD
7  5  5  5  4  4  4  3  3  3  3  3  3  3  3
```





(5) Preučimo besede z dvema črkama:

```
   Samoglasnik na koncu
1) da ga na pa ta za (ha ja la)
2) "ce je le me ne se "se te ve "ze (he)
3) bi ji ki mi ni si ti vi
4) bo do (ho) jo ko no po so to
5) ju mu tu (bu)
6) r"z rt
```





```
   Samoglasnik na za"cetku
1) ar as             (ah aj au)
2) en ep              (ej eh)
3) in iz ig
4) on ob od os on     (oh oj)
5) uk up u"s ud um ur (uh ut)
```

in opazujemo besedi: /- ?H
ter besedi: J+ ?H.





J+ ima najmanj možnosti, + pa verjetno ni črka n, zato nam ostane samo še:

```
J+ ?H        DWHN-
/- ?H
iz te       (ne gre zaradi: D-?+C)
ob ta(e,o)  (ne gre zaradi: D-?+C)
od te       (ne gre zaradi: D-?+C)
```

tako da bo potrebno nekaj spremeniti in preizkusiti še naslednje:
on bo; on jo; in so; in se; in je; in ta; en je; od tu ...





(6) Če nam po dolgem premisleku ne uspe najti rdeče niti,
bo morda potrebno iskati napako s prijatelji

(tudi računalniški program z metodo lokalne optimizacije
ni zmogel problema zaradi premajhne dolžine tajnopisa,
vsekakor pa bi bilo problem mogoče rešiti z uporabo
elektronskega slovarja).

Tudi psihološki pristop pomaga, je svetoval Martin Juvan
in naloga je bila rešena (poskusite sami!).

Podobna naloga je v angleščini dosti lažja,
saj je v tem jeziku veliko členov THE, A in AN,
vendar pa zato običajno najprej izpustimo presledke
iz teksta, ki ga želimo spraviti v tajnopis.

V angleščini imajo seveda črke drugačno gostoto
kot v slovenščini.





Razdelimo jih v naslednjih pet skupin:

1. E, z verjetnostjo okoli 0.120,
2. T, A, O, I, N, S, H, R, vse z verjetnostjo med 0.06 in 0.09,
3. D, L, obe z verjetnostjo okoli 0.04,
4. C, U, M, W, F, G, Y, P, B, vse z verjetnostjo med 0.015 in 0.028,
5. V, K, J, X, Q, Z, vse z verjetnostjo manjšo od 0.01.





Najbolj pogosti pari so (v padajočem zaporedju): TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI in OF,

Najbolj pogoste trojice pa so (v padajočem zaporedju): THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR in DTH.





## Klasične šifre

### Transpozicijska šifra

V transpozicijski šifri ostanejo črke
originalnega sporočila nespremenjene, njihova mesta
pa so pomešana na kakšen sistematičen način

(primer: permutacija stolpcev).

Te šifre zlahka prepoznamo, če izračunamo
gostoto samoglasnikov (v angleščini je ta 40%,
in skoraj nikoli ne pade zunaj intervala 35%–45%).

Težko jih rešimo, vendar pa se potrpljenje na koncu
običajno izplača.





**Simetrična šifra** *je peterica* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *za katero velja:*

1. $\mathcal{P}$ *je končna množica možnih čistopisov*
2. $\mathcal{C}$ *je končna množica možnih tajnopisov*
3. $\mathcal{K}$ *je končna množica možnih ključev.*
4. *Za vsak ključ* $K \in \mathcal{K}$, *imamo šifrirni postopek*

   $e_K \in \mathcal{E}$ *in ustrezen odšifrirni postopek* $d_K \in \mathcal{D}$.

   $$e_K : \mathcal{P} \longrightarrow \mathcal{C} \quad \text{in} \quad d_K : \mathcal{C} \longrightarrow \mathcal{P}$$

   *sta taki funkciji, da je* $d_K(e_K(x)) = x$ *za vsak*

   $x \in \mathcal{P}$.





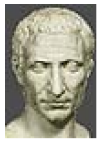
**Zamična šifra** (angl. shift cipher) je poseben primer zamenjalne šifre.

```
wewillmeetatmidnight

 22  4 22  8 11 11 12  4  4 19  0 19 12  8  3 13  8  6  7 19
  7 15  7 19 22 22 23 15 15  4 11  4 23 19 14 24 19 17 18  4

HPHTWWXPPELEXTOYTRSE
```

Cezarjeva šifra zašifrira njegovo ime v Ehbčt.



Cezar ukazal napad

↓

Ehbčt v zn̈čbco v r̈čscg

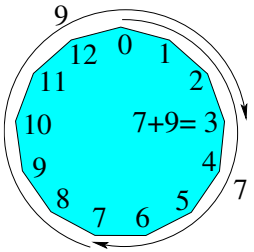
V kriptografiji si na splošno radi omislimo končne množice, kot pri številčnici na uri (npr. praštevilske obsege $\mathbb{Z}_p$).

**Kongruence:** naj bosta $a$ in $b$ celi števili in $m$ naravno število.

$$a \equiv b \pmod{m} \iff m | b - a.$$

**Primer:** za $p = 13$ velja
$7 +_{13} 9 = 7 + 9 \mod 13 = 3$ in
$5 *_{13} 4 = 5 * 4 \mod 13 = 7$

(saj ima pri deljenju s 13
vsota 16 ostanek 3,
produkt 20 pa ostanek 7),
možno pa je tudi deljenje.








Deljenje v primeru $p = 13$:

| $*_{13}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **1** | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 2 | 2 | 4 | 6 | 8 | 10 | 12 | **1** | 3 | 5 | 7 | 9 | 11 |
| 3 | 3 | 6 | 9 | 12 | 2 | 5 | 8 | 11 | **1** | 4 | 7 | 10 |
| 4 | 4 | 8 | 12 | 3 | 7 | 11 | 2 | 6 | 10 | **1** | 5 | 9 |
| 5 | 5 | 10 | 2 | 7 | 12 | 4 | 9 | **1** | 6 | 11 | 3 | 8 |
| 6 | 6 | 12 | 5 | 11 | 4 | 10 | 3 | 9 | 2 | 8 | **1** | 7 |
| 7 | 7 | **1** | 8 | 2 | 9 | 3 | 10 | 4 | 11 | 5 | 12 | 6 |
| 8 | 8 | 3 | 11 | 6 | **1** | 9 | 4 | 12 | 7 | 2 | 10 | 5 |
| 9 | 9 | 5 | **1** | 10 | 6 | 2 | 11 | 7 | 3 | 12 | 8 | 4 |
| 10 | 10 | 7 | 4 | **1** | 11 | 8 | 5 | 2 | 12 | 9 | 6 | 3 |
| 11 | 11 | 9 | 7 | 5 | 3 | **1** | 12 | 10 | 8 | 6 | 4 | 2 |
| 12 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | **1** |





**Afina šifra:**

$$e(x) = ax + b \pmod{26} \quad \text{za } a, b \in \mathbb{Z}_{26}$$

Za $a = 1$ dobimo zamično šifro.
Funkcija je injektivna, če in samo če je $D(a, 26) = 1$.

Imamo $|\mathcal{K}| = 12 \times 26 = 312$ možnih ključev.

Za zamično šifro in afino šifro pravimo, da sta **monoabecedni**, ker preslikamo vsako črko v natanko določeno črko.





**Vigenèrejeva šifra** (1586):

Naj bo $m \in \mathbb{N}$ in

$$\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m.$$

Za ključ $K = (k_1, k_2, \ldots, k_m)$ definiramo

$$\begin{aligned} e(x_1, \ldots, x_m) &= (x_1 + k_1, \ldots, x_m + k_m) \text{ in} \\ d(y_1, \ldots, y_m) &= (y_1 - k_1, \ldots, y_m - k_m), \end{aligned}$$

kjer sta operaciji "+" in "−" opravljeni po modulu 26.





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





Sporočilo

```
TO BE OR NOT TO BE THAT IS THE QUESTION
```

zašifriramo s ključem `RELATIONS`:

```
klju"c:       RELAT IONSR ELATI ONSRE LATIO NSREL
"cistopis:    TOBEO RNOTT OBETH ATIST HEQUE STION
tajnopis:     KSMEH ZBBLK SMEMP OGAJX SEJCS FLZSY
```

Npr. prvo črko tajnopisa dobimo tako, da pogledamo v tabelo na mesto (`R`, `T`).

Kako pa najdemo iz `T` in `K` nazaj `R`?

To ni monoabecedna šifra.

Pravimo ji **poliabecedna šifra**.

Vigenèrejeva šifra in $26^m$ možnih ključev.

Za $m = 5$ je število $1.1 \times 10^7$ že preveliko,
da bi "peš" iskali pravi ključ.





**Hillova šifra** (1929)

Naj bo $m$ neko naravno število in naj bo

$$\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^m.$$

Za $K$ vzemimo obrnljivo $m \times m$ matriko in definirajmo

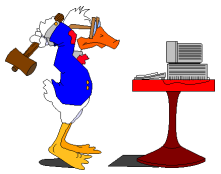
$$e_K(x) = xK \quad \text{in} \quad d_K(y) = yK^{-1},$$

pri čemer so vse operacije opravljene v $\mathbb{Z}_{26}$.





**Ponovimo:**



**Odšifriranje (razbijanje)**
**klasičnih šifer**

Kriptografske sisteme kontroliramo
s pomočjo ključev, ki določijo transformacijo podatkov.
Seveda imajo tudi ključi digitalno obliko
(binarno zaporedje: 01001101010101...).

Držali se bomo **Kerckhoffovega principa**,
ki pravi, da "nasprotnik"

*pozna kriptosistem oziroma algoritme,*
*ki jih uporabljamo, ne pa tudi ključe,*
*ki nam zagotavljajo varnost.*





Ločimo naslednje nivoje napadov na kriptosisteme:

1. **samo tajnopis**: *nasprotnik ima del tajnopisa,*
2. **poznani čistopis**: *nasprotnik ima del*
   *čistopisa ter ustrezen tajnopis,*
3. **izbrani čistopis**: *nasprotnik ima začasno na*
   *voljo šifrirno mašinerijo ter za izbrani $x \in \mathcal{P}$*
   *konstruira $e(x)$,*
4. **izbrani tajnopis**: *nasprotnik ima začasno na*
   *voljo odšifrirno mašinerijo ter za izbrani $y \in \mathcal{C}$*
   *konstruira $d(y)$.*





## Odšifriranje Vigenèrejeve šifre

**Test Friedericha Kasiskega (1863):**

(in Charles Babbage-a 1854)

poiščemo dele tajnopisa $\boldsymbol{y} = y_1 y_2 \dots y_n$, ki so identični
in zabeležimo razdalje $d_1, d_2, \dots$ med njihovimi začetki.
Predpostavimo, da iskani $m$ deli največji skupni delitelj teh števil.

Naj bo $d = n/m$.
Elemente tajnopisa $\boldsymbol{y}$ zapišemo po stolpcih v $(m \times d)$-razsežno matriko.
Vrstice označimo z $\boldsymbol{y_i}$, tj.

$$\boldsymbol{y_i} \;=\; y_i \, y_{m+i} \, y_{2m+i} \, \cdots$$





**Indeks naključja** (William Friedman, 1920):

Za zaporedje $\boldsymbol{x} = x_1 x_2 \dots x_d$ je **indeks naključja**
(angl. index of coincidence, oznaka $I_c(\boldsymbol{x})$)
**verjetnost**, da sta naključno izbrana elementa
zaporedja $\boldsymbol{x}$ enaka.

Če so $f_0, f_1, \dots, f_{25}$ frekvence črk $A, B, \dots, Z$ v zaporedju $\boldsymbol{x}$, je

$$I_c(\boldsymbol{x}) \;=\; \frac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{d}{2}} \;=\; \sum_{i=0}^{25} \frac{f_i(f_i-1)}{d(d-1)}.$$

Če so $p_i$ pričakovane verjetnosti angleških črk, potem je

$$I_c(\boldsymbol{x}) \approx \sum_{i=0}^{25} p_i^2 = 0.065.$$

Za povsem naključno zaporedje velja

$$I_c(\boldsymbol{x}) \approx 26\left(\frac{1}{26}\right)^2 = \frac{1}{26} = 0.038.$$

Ker sta števili .065 in .038 dovolj narazen, lahko s to metodo najdemo dolžino ključa

(ali pa potrdimo dolžino, ki smo jo uganili s testom Kasiskega).





Za podzaporedje $\boldsymbol{y_i}$ in $0 \le g \le 25$ naj bo

$$M_g(\boldsymbol{y_i}) = \sum_{i=0}^{25} p_i \frac{f_{i+g}}{d}.$$

Če je $g = k_i$, potem pričakujemo

$$M_g(\boldsymbol{y_i}) \approx \sum_{i=0}^{25} p_i^2 = 0.065$$

Za $g \neq k_i$ je običajno $M_g$ bistveno manjši od $0.065$.

Torej za vsak $1 \le i \le m$ in $0 \le g \le 25$ tabeliramo vrednosti $M_g$, nato pa v tabeli za vsak $1 \le i \le m$ poiščemo tiste vrednosti, ki so blizu $0.065$.

Ustrezni $g$-ji nam dajo iskane zamike $k_1, k_2, \ldots, k_m$.





## Odšifriranje Hillove šifre

Predpostavimo, da je nasprotnik določil $m$, ki ga uporabljamo, ter se dokopal do $m$ različnih parov $m$-teric (2. stopnja – poznan čistopis):

$$x_j = (x_{1,j}, x_{2,j}, ..., x_{m,j}),\ y_j = (y_{1,j}, y_{2,j}, ..., y_{m,j}),$$

tako da je $y_j = e_K(x_j)$ za $1 \le j \le m$.

Za matriki $X = (x_{i,j})$ in $Y = (y_{i,j})$ dobimo matrično enačbo $Y = XK$.

Če je matrika $X$ obrnljiva, je $K = YX^{-1}$.





Za Hillovo šifro lahko uporabimo tudi 1. stopnjo napada (samo tajnopis), glej nalogo 1.25.

Koliko ključev imamo na voljo v primeru Hillove šifre?
Glej nalogo 1.12.

Za afino-Hillovo šifro glej nalogo 1.24.





## Tokovne šifre

Naj bo $x_1x_2\ldots$ čistopis.

Doslej smo obravnavali kriptosisteme z enim samim ključem in tajnopis je imel naslednjo obliko.

$$\mathbf{y} = y_1y_2\cdots = e_K(x_1)e_K(x_2)\ldots$$

Taki šifri pravimo **bločna šifra** (angl. block cipher).





Posplošitev: iz enega ključa $K \in \mathcal{K}$ napravimo zaporedje (tok) ključev. Naj bo $f_i$ funkcija, ki generira $i$-ti ključ:

$$z_i = f_i(K, x_1, \ldots, x_{i-1}).$$

Z njim izračunamo:

$$y_i = e_{z_i}(x_i) \quad \text{in} \quad x_i = d_{z_i}(y_i).$$

Bločna šifra je poseben primer tokovne šifre (kjer je $z_i = K$ za vse $i \ge 1$).

**Sinhrona tokovna šifra** *je sedmerica*

$(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{L}, \mathcal{F}, \mathcal{E}, \mathcal{D})$ *za katero velja:*

1. $\mathcal{P}$ *je končna množica možnih* **čistopisov**,
2. $\mathcal{C}$ *je končna množica možnih* **tajnopisov**,
3. $\mathcal{K}$ *je končna množica možnih* **ključev**,
4. $\mathcal{L}$ *je končna množica tokovne* **abecede**,
5. $\mathcal{F} = (f_1, f_2, \dots)$ *je generator toka ključev:*

$$f_i : \mathcal{K} \times \mathcal{P}^{i-1} \longrightarrow \mathcal{L} \quad \text{za } i \geq 1$$

6. *Za vsak ključ* $z \in \mathcal{L}$ *imamo šifrirni* $(e_z \in \mathcal{E})$

   *in odšifrirni* $(d_z \in \mathcal{D})$ *postopek,*

   *tako da je* $d_z(e_z(x)) = x$ *za vsak* $x \in \mathcal{P}$.





Za šifriranje čistopisa $x_1 x_2 \dots$ zaporedno računamo

$$z_1, y_1, z_2, y_2, \dots,$$

za odšifriranje tajnopisa $y_1 y_2 \dots$ pa zaporedno računamo

$$z_1, x_1, z_2, x_2, \dots.$$

Tokovna šifra je **periodična** s periodo $d$ kadar, je $z_{i+d} = z_i$ za vsak $i \geq 1$

(poseben primer: Vigenèrejeva šifra).





Začnimo s ključi $(k_1, \dots, k_m)$ in naj bo $z_i = k_i$ za $i = 1, \dots, m$.
Definiramo linearno rekurzijo stopnje $m$:

$$z_{i+m} \;=\; z_i \;+\; \sum_{j=1}^{m-1} c_j \, z_{i+j} \mod 2,$$

kjer so $c_1, \dots, c_{m-1} \in \mathbb{Z}_2$ vnaprej določene konstante.

Za ustrezno izbiro konstant $c_1, \dots, c_{m-1} \in \mathbb{Z}_2$ in neničelen vektor $(k_1, \dots, k_m)$ lahko dobimo tokovno šifro s periodo $2^m - 1$.





Hitro lahko generiramo tok ključev z uporabo **LFSR**
(**Linear Feedback Shift Register**).

V zamičnem registru začnemo z vektorjem

$$(k_1, \dots, k_m).$$

Nato na vsakem koraku naredimo naslednje:

1. $k_1$ dodamo toku ključev (za XOR),
2. $k_2, \dots, k_m$ pomaknemo za eno v levo,
3. 'nov' ključ $k_m$ izračunamo z

$$\sum_{j=0}^{m-1} c_j \, k_{j+1} \quad (\text{to je "linear feedback"}).$$





**Primer:**

$c_0 = 1, c_1 = 1, c_2 = 0, c_3 = 0,$

torej je $k_{i+4} = k_i + k_{i+1}$.

Izberimo $k_0 = 1, k_1 = 0, k_2 = 1, k_3 = 0$.

Potem je $k_4 = 1, k_5 = 1, k_6 = 0, \dots$.





Naj bo $\boldsymbol{k} = (k_0, k_1, k_2, k_3)^t$ in

$$A := \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

Torej je $A(\boldsymbol{k}) = (k_1, k_2, k_3, k_4)^t$,

$A^2(\boldsymbol{k}) = A(k_1, k_2, k_3, k_4)^t = (k_2, k_3, k_4, k_5)^t$
...
$A^i(\boldsymbol{k}) = (k_i, k_{i+1}, k_{i+2}, k_{i+3})^t$.

Najdaljša možna perioda je 15.

Enkrat dobimo:

$$A^i(\boldsymbol{k}) = A^j(\boldsymbol{k})$$

in ker je $A$ obrnljiva

$$A^{i-j}(\boldsymbol{k}) = \boldsymbol{k}$$

Karakteristični polinom matrike $A$ je

$$f(x) = 1 + x + x^4.$$

Ker je $f(x)$ nerazcepen, je $f(x)$ tudi minimalni polinom matrike $A$.





Red matrike $A$ je najmanjše naravno število $s$, tako da je $A^s = I$.
Naj bo $e$ najmanjše naravno število, tako da $f(x) \mid (x^e - 1)$.

Potem je $e = s$.

$$\begin{aligned} 1 + x^{15} &= (x+1)(x^2+x+1)(x^4+x+1) \\ &\quad (x^4+x^3+1)(x^4+x^3+x^2+x+1). \end{aligned}$$

Splošno: če hočemo, da nam rekurzija stopnje $m$ da periodo $2^m - 1$, potem si izberemo nerazcepen $f$.

Analiza je neodvisna od začetnega neničelnega vektorja.

**Kriptoanaliza LFSR tokovne šifre:**
uporabimo lahko poznan čistopis, glej nalogo 1.27.





# 2. Uvod v računalniško varnost

- Varnostna politika
- Pomembnejše grožnje za varnost
- Pregled varnostnih ukrepov

V 90-ih se je pričela zbujati **informacijska doba**.

Podjetja, organizacije in celotna družba postajajo popolnoma odvisne od računalnikov in njihovega pravilnega delovanja.

Medtem, ko se učinkovitost pospešeno izboljšuje,
varnost ni bila še nikoli na slabšem.





**Naštejmo nekaj razlogov za tako stanje:**

- varnost ni bila pomembna na začetku, sedaj pa neprestano **zaostaja**,
- računalniki, sprva uporabljeni za računanje plač, nato za upravljanje z delnicami, EDI..., so danes nujno potrebno orodje pri strateškem odločanju,
- povezovanje v mreže je v velikem razmahu,
- *demokratizacija* informacijske tehnologije je premaknila računalnike iz računskih centrov na delovna mesta.





**Zakaj so računalniške goljufije privlačne?**

- malo izpostavljanja (dobro pripravljen napad lahko traja manj kot sekundo),
- nadzor - če obstaja - opravljajo računalniki (in ne ljudje),
- večina informacij je centraliziranih,
- kraja se lahko avtomatizira in ponavlja (torej ni nujno požrešna):
  $10.000 \times 100 = 1$ milion
  in jo bomo težje odkrili.





Čeprav običajno povezujemo varnostne težave z vdori hackerjev ali kriminalnih organizacij, večino težav povzroči **notranje osebje** (zaposleni, vzdrževalci...).

| **pogostost** | **razlogi** |
|---|---|
| 50–60% | napake (neizkušenost, neodgovornost, panika...) |
| 15–20% | notranje osebje (zaposleni, vzdrževalci...) |
| 10–15% | nesreče (poplave, strele, požar...) |
| 3–5% | zunanji dejavniki (hackerji, konkurenca, organiziran kriminal...) |

## Varnostna politika

Težko je doseči spodoben nivo varnosti brez ustrezne politike, ki naj obravnava vsaj naslednje:

- osnovni namen,
- pomen informacijske tehnologije za organizacijo,
- čas veljavnosti (naj bo kratkoročna, da jo dopoljnjujemo vsako (drugo) leto),
- vključitev v plan (sredstva in osebje),
- konkretni cilji, zahteve in odgovornost.

Izvajanje varnostne politike bo uspešna le, če jo podpre **ožje vodstvo**.





## Kvaliteta informacij

- **zaupnost** (pred konkurenco, vendar pa tudi zaščita zasebnosti zaposlenih),
- **celovitost** (zanesljivost informacij, ki morajo biti pravilne in avtentične),
- **dostopnost** (procesiranje in distribucija informacij).

Da določimo kvaliteto informacij, moramo zaslišati vse uporabnike računalniške infrastrukture.

**Analiza tveganja:** izgube ne vsebujejo samo čas in denar za odpravo učinkov groženj, temveč tudi finančne in pravne posledice.

$$\text{tveganje} = \text{verjetnost} \times \text{izgube}.$$





**100% varnega sistema ni, razen tistega,**
**ki ga izklopimo in skrijemo v bunker.**

Pogosto varnostni ukrepi niso "*prijazni uporabniku*" (angl. user-friendly) in če uporabniki niso povsem prepričani v njihovo nujnost, se jim bodo skušali ogniti.

**Ljudje** so pogosto najšibkejši člen v varnostni verigi.

To lahko izboljšamo le z neprestanim izobraževanjem, ki je usmerjeno v povečano budnost/pazljivost in zavest o odgovornosti.





En od ključnih principov varnosti je **odgovornost**
– vsakdo mora biti odgovoren za svoja dejanja.

**Vsakemu** procesu/ključu se dodeli lastnik/uporabnik, ki je odgovoren za ustrezna početja.

Kontrola dostopa, revizija in drugi servisi so odvisni od pravilne identifikacije uporabnikov ter overitve uporabnikove identitete.

Ko enkrat sistem overi uporabnikovo identiteto in pravico do vstopa do sistema, mora kontrolirati uporabnikov dostop do virov in sredstev.





## Pomembnejše grožnje za varnost

| **normalna/pričakovana uporaba** | **zlorabe** |
|---|---|
| dostop do rač. sistema | zunanji napadi |
| uporaba rač. sistema | zloraba harwara |
| navidezno pooblaščena uporaba | maškarada |
| neposredna uporaba | škodljivci in kasnejše napake |
| navidezno ustrezna uporaba | obhod namenjenih kontrol |
| aktivna uporaba | aktivna zloraba sredstev |
| navidezno normalna uporaba | pasivna zloraba sredstev |
| navidezno ustrezna uporaba | napake zaradi neaktivnosti |
| ustrezna uporaba | uporabljena za pasti |





Pri **zunanjih napadih** mislimo na grožnje, za katere ne potrebujemo fizičnega dostopa do računalnika ali mreže.

- **Vizualno vohunjenje:** s kamero (ali pa samo za hrbtom) lahko opazujemo vtipkavanje gesel.

  Vsebino z zaslona je možno prekopirati z razdalje (iz kombija pred stavbo) zaradi elektro-magnetne radijacije zaslona.
- **Social engineering:** uporabnika ali pa operaterja prepričamo v početje določenih neumnosti:

  e-pošta, nepreverjanje identitete (tel.).
- **Tiskane informacije** (hranjene/odvržene).

**Hardwarska zloraba:** običajno ne zbrišemo disket, diskov, trakov, ki so namenjeni za smeti.

- **Network sniffers** so pogosto dostopni na Internetu in običajno konfigurirani za poslušanje samo prvih 100 bytov.
- **Jamming** (*denial of service attack*), ki povzroči resne težave z uporabo mrež. Nezadovoljni zaposleni lahko poškodujejo hardware ali pa njegove namestitve (napajanje, hlajenje...).
- **Kraja**, posebno majhnih naprav, ki jih je lažje odnesti.





**Maškarata:** ko si enkrat napadalec pridobi gesla in druga sredstava za preverjanje avtentičnosti, lahko impersonira uporabnika:

- **kraja gesel** sploh ni tako nemogoča (okoli 25% jih lahko dobimo z ugibanjem, napadi s slovarji...),
- **Piggi-backing attacks:** fizičen dostop do komunikacijskih linij ali delovne postaje (*xlock*, *screen lock*, *keyboard lock*...) in izvedba določenih ukazov,





- **Playback attacks** (*network weaving*): kadar je težko ugotoviti izvor,
- **Spoofing attacks:** računalniki na lokalnih mrežah se predstavijo samo enkrat.

  Napadalec prepriča več strežnikov, da prihaja s pooblaščene naprave in strežnik izpolni njegove zahteve.





**Škodljivi programi** poskušajo pripraviti priložnosti za kasnejše napade:

| tip | pojasnilo |
|---|---|
| Trojan Horse | program, ki naredi nekaj nepričakovanega |
| Mule (Spoof) | program, ki oponoša nek drug program |
| Rabbit | program, ki prezasiti sistem |
| Worm | program, ki se množi po sistemu |
| Virus | del programa, ki se ob izvajanju pripne na druge programe |





**Napadi z obhodom** (angl. bypass) uporabijo obstoječe napake, da se izognejo avtentikaciji.

**Trapdoor** je vstopna pot, ki ni namenjena za običajno uporabo (v nekaterih primerih je bila implementirana za *debugging* in nato pozabljena). Npr.

- `sendmail`-program je vseboval `DEBUG` ukaz, ki ga je izkoristil Internet worm,
- *finger/talk daemon*





Številni programi ne preverjajo *input* in tako napadalec pošlje nepričakovane podatke (napačen format ali pa velikost) ali pa požene program na strežniku.

Nepričakovana količina podatkov povzroči *overflow* podatkov, ki se naloženi na skladu. Le-ta se popači (pomembne informacije kot npr. povratni naslovi za klice procedur se spremenijo).

Na ta način napadalec prepriča strežnikov program, da naredi kar mu je ljubo.

## Pregled varnostnih ukrepov

Težko je zajeti vse ukrepe, ki jih lahko uporabimo. Pogosto en ukrep ne bo dovolj za določeno grožnjo, nekateri ukrepi pa lahko vplivajo na več napadov.

| | **fizična zaščita** | **tehnični ukrepi** | **organizacijski ukrepi** |
|---|---|---|---|
| **preventiva** | straža pri vhodu ... | firewalls ... | izobraževanje ... |
| **odkrivanje** | detektorji gibanja ... | logs ... | call-back ... |
| **popravljanje** | UPS ... | anti-virus ... | backup ... |





**Tehnični ukrepi** so običajno softwarske rešitve (ali pa kombinacija hardwara in softwara).

- **kontrola dostopa**
  (zaščita notranjih sredstev
  in omejitev uporabnikovih pooblastil),
- **overjeni login-session**
  (zaščita dostopa do sistema,
  ki lahko vsebuje kriptografske protokole
  in call-back mehanizme, itd.),
- **dnevniki** (angl. logs)
  (omogoča izsledite varnostnih problemov),
- **anti-virusni programi** (skeniranje datotek za znane viruse, preverjanje integritete datotek),
- **cryptography** (zaščita zaupnosti, celovitosti in avtentičnosti podatkov; upravljanje s ključi...),
- **firewalls** (filtriranje paketov in obramba notranjega dela mreže pred vsiljivimi akcijami).





**Gesla** omogočajo kontrolo dostopa v večino računalnikov.

Dobro geslo si je enostavno zapomniti in težko uganiti (ta dva kriterija si seveda pogosto nasprotujeta).

Če si je geslo težko zapomniti, potem si jih ljudje običajno zapišejo ali pa jih pozabijo in si morajo pridobiti ponoven dostop na poseben in ranljiv način.

**Kako si zapomnimo varnejša gesla?**





# 3. 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





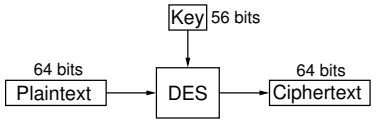
## National Security Agency (NSA)

- 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:** $K^1, \ldots, K^{N_r}$.

Seznamu krožnih ključev $(K^1, \ldots, K^{N_r})$ 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:

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

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 $\ell m$, $\ell, m \in \mathbb{N}$ (tj. $\ell m$ 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 $\pi_S$ imenujemo $S$-**škatla** in z njo zamenjamo $\ell$ bitov z drugimi $\ell$ biti.

Permutacija $\pi_P$ pa permutira $\ell m$ bitov.





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

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

**SPN šifra**

*$\ell, m, N_r \in \mathbb{N}$, $\pi_S$ in $\pi_P$ 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 ključev. Šifriramo z algoritmom SPN.*





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

$w^0 := x$

**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** $v^r_{(i)} := \pi_S(u^r_{(i)})$

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

*(zadnji krog)*

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

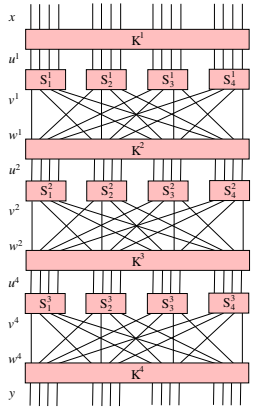
**for** $i := 1$ **to** $m$ **do** $v^{N_r}_{(i)} := \pi_S(u^{N_r+1}_{(i)})$

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

**output** $(y)$








Primer: naj bo $\ell = m = N_r = 4$, permutaciji $\pi_S$ in $\pi_P$ 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 $K = (k_1, \ldots, k_{32}) \in \{0,1\}^{32}$ definiran z

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

sedaj pa izberimo še razpored ključev tako, da je za $1 \leq r \leq 5$, 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 $K$.

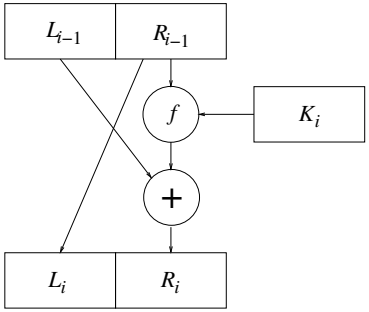
Končamo z $(R_r, L_r)$ (in ne z $(L_r, R_r)$), zato je šifriranje enako odšifriranju, le da ključe uporabimo v obratnem vrstnem redu.

Funkcija $f$ je lahko produktna šifra in ni nujno obrnljiva.





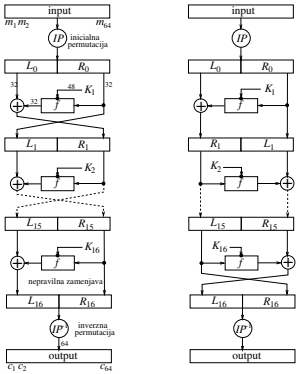
## En krog








## Opis šifre DES



## DES-ove konstante

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

razširitev: $E$ (nekatere bite ponovimo), permutacija $P$

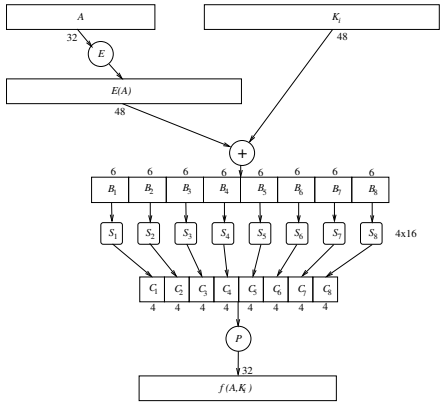
$S$-škatle: $S_1, S_2, \ldots, S_8$
(tabele: $4 \times 16$, z elementi $0 - 15$)

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





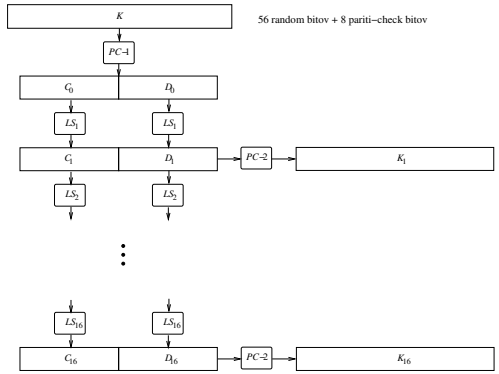
## DES-ova funkcija








## 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

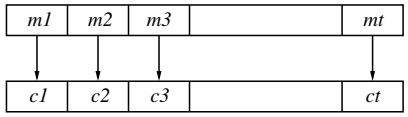
– electronic codebook mode **(ECB)**

– 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.

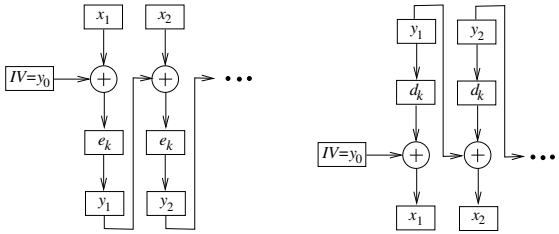




## Cipher Block Chaining mode – CBC

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

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



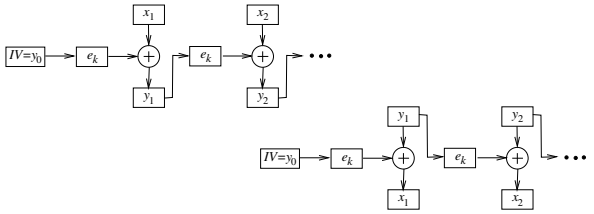

Odšifriranje: $y_0 := \mathrm{IV}$, $x_i := y_{i-1} \oplus d_K(y_i)$ za $i \geq 1$.

Identična 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 $x_1, x_2, \ldots / y_1, y_2, \ldots$

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




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

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

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: $z_0 := \text{IV}$, šifriranje:

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

Odšifriranje:($z_0 := \text{IV}$)

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

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: $\leq 8$ 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?).

$$\mathrm{DES_E}(\mathrm{P},\mathrm{K_1}) \longrightarrow \mathrm{DES_D}(\mathrm{DES_E}(\mathrm{P},\mathrm{K_1}),\mathrm{K_2})$$
$$\longrightarrow \mathrm{DES_E}(\mathrm{DES_D}(\mathrm{DES_E}(\mathrm{P},\mathrm{K_1}),\mathrm{K_2}),\mathrm{K_3})$$

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 | $\approx 6 \times 10^9$ |
| (v letih) | |
| 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**:

$$\mathrm{DESX_{k.k1.k2}}(\mathrm{x}) = \mathrm{k2} \oplus \mathrm{DES_k}(\mathrm{k1} \oplus \mathrm{x}).$$

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 $(4 \times 4)$-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 $\pi_S$ $\{0,1\}^8$, 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, ...

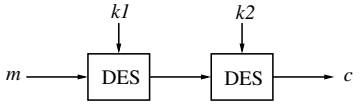




## Dvojno šifriranje

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

**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.





## Meet-in-the-middle napad na 2-DES

- Iz $c = E_{k_2}(E_{k_1}(m))$ sledi $E^{-1}_{k_2}(c) = E_{k_1}(m)$.
- Input: znani čp/tp pari $(m_1, c_1), (m_2, c_2), (m_3, c_3)$.
- Output: tajni ključ $(k_1, k_2)$.

Za vsak $h_2 \in \{0,1\}^{56}$, izračunaj $E^{-1}_{h_2}(c_1)$
in shrani $[E^{-1}_{h_2}(c_1), h_2]$ v tabelo indeksirano s prvo koordinato.

Za vsak $h_1 \in \{0,1\}^{56}$ naredi naslednje:

1. Izračunaj $E_{h_1}(m_1)$.
2. Išči $E_{h_1}(m_1)$ v tabeli.
3. Za vsako *trčenje* $[E^{-1}_{h_2}(c_1), h_2]$ v tabeli preveri,
   ali je $E_{h_2}(E_{h_1}(m_2)) = c_2$ in $E_{h_2}(E_{h_1}(m_3)) = c_3$.
   Če se to zgodi, potem izpiši $(h_1, h_2)$ in se vstavi.

**Analiza:**

- Število DES operacij je $\approx 2^{56} + 2^{56} = 2^{57}$.
- Pomnilnik: $2^{56}(64 + 56)$ bitov $\approx$ 983,040 TB.

**Zaključek:**

- 2-DES ima enako učinkovit ključ kot DES.
- 2-DES ni varnejši od DES-a.

**Time-memory tradeoff:**

- Čas: $2^{56+s}$ korakov; pomnilnik: $2^{56-s}$ enot, $1 \leq s \leq 55$. [DN]





## 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.

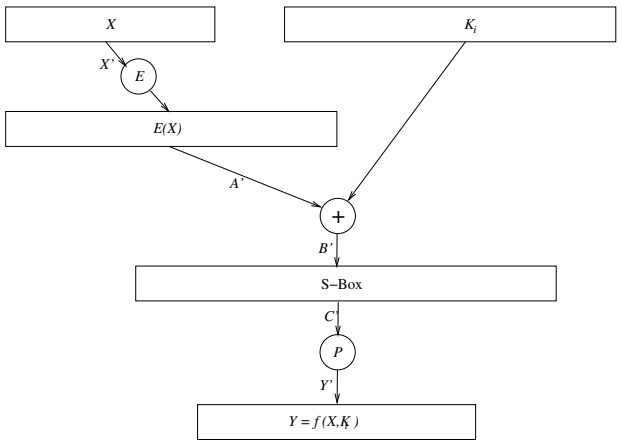




Naj bosta $X$ in $X^*$ par čistopisov z razliko $X'$.
Tajnopisa $Y$ in $Y^*$ poznamo, zato poznamo tudi njuno razliko $Y'$.
Naj bo $A^{(*)} := E(X^{(*)})$ in $P(C^{(*)}) = Y^{(*)}$.

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 $K_i$.








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 $B_j = b_1b_2b_3b_4b_5b_6$.
$S_i(B_j)$ določimo na naslednji način.

Bita $b_1b_6$ določita vrstico $v$, biti $b_2b_3b_4b_5$ 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: $\Delta(B'_j) := \{(B_j, B_j \oplus B'_j) \,|\, B_j \in (\mathbb{Z}_2)^6\}$

**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 $j \in \{1, \dots, 8\}$.

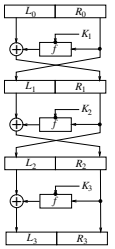
Potem poiščemo razliko $(C')_j$ v tabeli za $S_j$ in $(A')_j$, ki nam določi vse možne vhode $B_j$ iz katerih izračunamo vse $B_j \oplus A_j$, ki morajo vsebovati $(K_i)_j$.

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...

## Napad na DES s tremi cikli




Naj bo $L_0R_0$ in $L_0^*R_0^*$ par čistopisa
in $L_3R_3$ in $L_3^*R_3^*$ par tajnopisa za katere velja:

$$L_3 = L_2 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$$





Še $L_3^*$ izrazimo na podoben način in dobimo

$$L_3' = L_0' \oplus f(R_0, K_1) \oplus f(R_0^*, K_1) \oplus f(R_2, K_3) \oplus f(R_2^*, K_3)$$

Predpostavimo še, da je $R_0 = R_0^*$ oziroma $R_0' = 00 \dots 0$.

Od tod dobimo

$$L_3' = L_0' \oplus f(R_2, K_3) \oplus f(R_2^*, K_3),$$

$L_3'$ je razlika tajnopisov, $L_0'$ pa razlika čistopisov, torej poznamo

$$f(R_2, K_3) \oplus f(R_2^*, K_3) \ \ (= L_0' \oplus L_3').$$

Naj bo $f(R_2, K_3) = P(C)$ in $f(R_2^*, K_3) = P(C^*)$,
kjer sta $C$ in $C^*$ definirana enako kot prej
(izhoda iz $S$ škatel po tretjem ciklu).

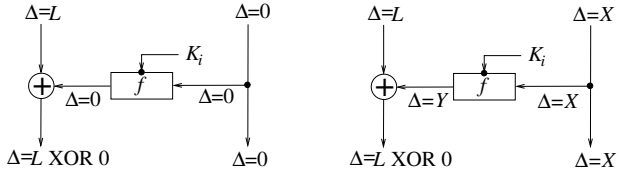
Potem je

$$C' = C \oplus C^* = P^{-1}(R_3' \oplus L_0').$$

Poznamo tudi $R_2 = R_3$ in $R_2^* = R_3^*$, saj sta $R_3$ in $R_3^*$ dela tajnopisa.

Torej smo prevedli kriptoanalizo DES-a s tremi cikli na diferenčno kriptoanalizo DES-a z enim ciklom.

## Napad na DES s 6-imi cikli




**(a)** Leva stran je karkoli, desna razlika pa je 0.
To je trivialna karakteristika in velja z verjetnostjo 1.

**(b)** Leva stran je karkoli, desna vhodna razlika pa je $0x60000000$
(vhoda se razlikujeta na 1. in 3. bitu).
Verjetnost, da bosta izhodni razliki $0x60000000$ in $0x00808200$
je enaka $14/64$.

**Karakteristika** za $n$-ciklov, $n \in \mathbb{N}$, je seznam

$$L'_0, R'_0, L'_1, R'_1, p_1, \ldots, L'_n, R'_n, p_n,$$

z naslednjimi lastnostmi:

- $L'_i = R'_{i-1}$ za $1 \leq i \leq n$.
- za $1 \leq i \leq n$ izberimo $(L_{i-1}, R_{i-1})$ in $(L^*_{i-1}, R^*_{i-1})$, tako da je $L_{i-1} \oplus L^*_{i-1} = L'_{i-1}$ in $R_{i-1} \oplus R^*_{i-1} = R'_{i-1}$. Izračunajmo $(L_i, R_i)$ in $(L^*_i, R^*_i)$ z enim ciklom DES-a. Potem je verjetnost, da je $L_i \oplus L^*_i = L'_i$ in $R_i \oplus R^*_i = R'_i$ natanko $p_i$.

**Verjetnost karakteristike** je $p = p_1 \times \cdots \times p_n$.





Začnimo s karakteristiko s tremi cikli:

$L'_0 = 0x40080000$, $R'_0 = 0x04000000$
$L'_1 = 0x40000000$, $R'_1 = 0x00000000$ $p = 1/4$
$L'_2 = 0x00000000$, $R'_2 = 0x04000000$ $p = 1$
$L'_3 = 0x40080000$, $R'_2 = 0x04000000$ $p = 1/4$
Potem velja

$$L'_6 = L'_3 \oplus f(R_3, K_4) \oplus f(R^*_3, K_4) \oplus f(R_5, K_6) \oplus f(R^*_5, K_6)$$

Iz karakteristike ocenimo $L'_3 = 0x04000000$ in
$R'_3 = 0x40080000$ z verjetnostjo 1/16.
Od tod dobimo razliko vhodov v $S$ škatle 4. cikla:
$001000000000000001010000 \ldots 0$.





Razlike vhodov v škatle $S_2$, $S_5$, $S_6$, $S_7$ in $S_8$ so 000000.
To nam omogoči, da z verjetnostjo 1/16 določimo v 6-tem ciklu
30 bitov originalnega ključa.

V tabelah ne smemo nikoli naleteti na prazno vrstico (**filtracija**).
Tako izključimo približno 2/3 napačnih parov,
med preostalimi pa je približno 1/6 pravilnih.

...





## Drugi primeri diferenčne kriptoanalize

Iste tehnike napadov na DES lahko uporabimo tudi
kadar imamo več kot 6 ciklov.

DES z $n$ cikli potrebuje $2^m$ izbranega čistopisa:

| n | m |
|---|---|
| 8 | 14 |
| 10 | 24 |
| 12 | 31 |
| 14 | 39 |
| 16 | 47 |

Na diferenčno kriptoanalizo so občutljivi tudi drugi algoritmi s substitucijami in permutacijami, kot na primer FEAL, REDOC-II in LOKI.





## Napad na DES s 16-imi cikli

Bihan in Shamir sta uporabila karakteristiko s
13-imi cikli in nekaj trikov v zadnjem ciklu.

Še več, z zvijačami sta dobila 56-bitni ključ, ki sta ga lahko testirala takoj
(in se s tem izognila potrebi po števcih).
S tem sta dobila linearno verjetnost za uspeh,
tj. če je na voljo 1000 krat manj parov,
imamo 1000 manj možnosti da najdemo pravi ključ.





Omenili smo že, da najboljši napad za DES s 16-imi cikli
potrebuje $2^{47}$ izbranih čistopisov.
Lahko pa ga spremenimo v napad z $2^{55}$ poznanega čistopisa,
njegova analiza pa potrebuje $2^{37}$ DES operacij.

Diferenčni napad je odvisen predvsem od strukture $S$ škatel.
Izkaže se, da so DES-ove škatle zoptimizirane proti takemu napadu.

Varnost DES-a lahko izboljšamo s tem, da povečamo število ciklov.
Vendar pa diferenčna kriptoanaliza DES-a s 17-imi ali 18-imi cikli
potrebuje toliko časa kot požrešna metoda (več ciklov nima smisla).

# 4. RSA sistem in faktorizacija

- Uvod
  - pomankljivosti simetrične kriptografije
  - kriptografija z javnimi ključi
- Teorija števil
- Opis in implementacija RSA
- Gostota praštevil
- Generiranje praštevil
- Gaussov izrek (o kvadratni recipročnosti)

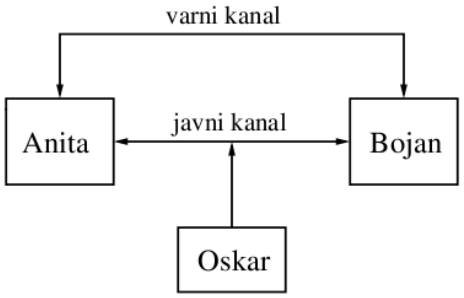




## Uvod

### Pomankljivosti simetrične kriptografije

Sodelujoči si delijo *tajno* informacijo.



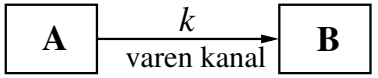





### Dogovor o ključu

Kako Anita in Bojan vzpostavita tajni ključ $k$?

**1. metoda:** delitev point-to-point




Varni kanal je lahko:
– kurir
– izmenjava na štiri oči (v temnem hodniku/ulici)

To ni praktično za večje aplikacije.



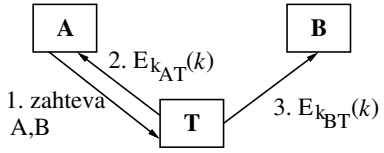


**2. metoda:** z neodvisnim centrom zaupanja $T$

– *Vsak* uporabnik $A$ deli tajni ključ $k_{AT}$ s
center zaupanja $T$ za simetrično šifrirno shemo $E$.

– Za vzpostavitev tega ključa mora $A$ obiskati
center zaupanja $T$ *samo enkrat*.

– $T$ nastopa kot **center za distribucijo ključev**:
(angl. key distribution centre - KDC):




1. $A$ pošlje $T$ zahtevek za ključ, ki si ga želi deliti z $B$.
2. $T$ izbere ključ $k$, ga zašifrira za $A$ s ključem $k_{AT}$.
3. $T$ zašifrira ključ $k$ za osebo $B$ s ključem $k_{BT}$.





### Problemi pri uporabi KDC

- centru zaupanja T moramo brezpogojno zaupati:
  - to ga naredi za očitno tarčo.
- Zahteva za stalno zvezo (on-line) s centrom T:
  - potencialno ozko grlo,
  - kritično za zanesljivost.

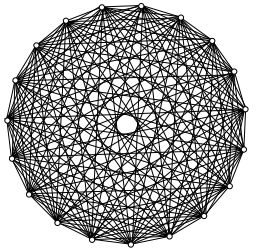


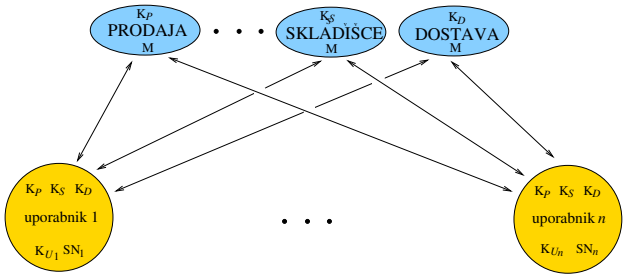


### Upravljanje ključev

- v mreži z $n$ uporabniki, mora vsak uporabnik deliti različen ključ z vsakim uporabnikom,
- zato mora hraniti vsak uporabnik $n-1$ različnih tajnih ključev,
- vseh tajnih ključev je $\binom{n}{2} \approx n^2/2$.

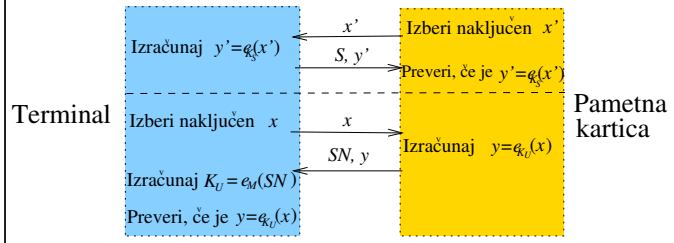
(Tudi preprečevanje tajenja je nepraktično.)









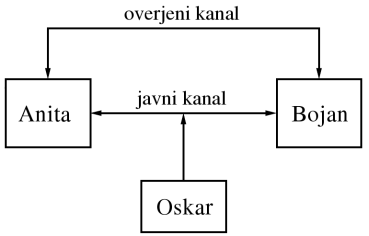








## Kriptografija z javnimi ključi

Udeleženci si predhodno delijo *overjeno/avtentično* informacijo.




L. **1976** sta jo predlagala Whitfield **Diffie** in Martin **Hellman**
(L. 1970 pa tudi James Ellis, ki je bil
član Communication Electronics Security Group pri
British Government Communications Headquarters).





## Generiranje para ključev

Vsaka oseba $A$ naredi naslednje:

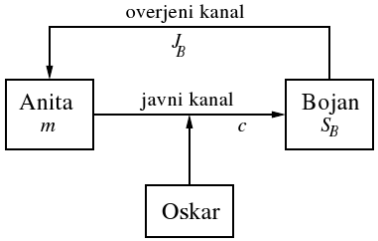
- generira par ključev $(J_A, S_A)$,
- $S_A$ je $A$-jev zasebni/tajni ključ,
- $J_A$ je $A$-jev javni ključ.

**Varnostna zahteva:** za napadalca mora biti nemogoče priti do kluča $S_A$ iz ključa $J_A$.





## Šifriranje z javnimi ključi




Da bi Bojanu poslala zaupno sporočil $m$, Anita:

- dobi overjenjo kopijo Bojanovega javnega kjuča $J_B$,
- izračuna $c = E(J_B, m)$, kjer je $E$ šifrirna funkcija,
- pošlje Bojanu tajnopis $c$.

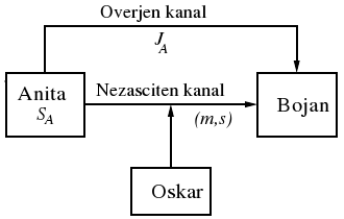
Za odšifriranje tajnopisa $c$ Bojan naredi naslednje

- Izračuna $m = D(S_B, c)$, kjer je $D$ odšifrirna funkcija.





## Digitalni podpisi




Za podpis sporočila $m$ Anita naredi naslednje:

- izračuna $s = \mathsf{Sign}(S_A, m)$,
- pošlje $m$ in $s$ Bojanu.

Bojan preveri Anitin podpis $s$ sporočila $m$:

- pridobi si overjeno kopijo javnega ključa $J_A$,
- sprejme podpis, če je $\mathsf{Verify}(J_A, m, s) = \mathsf{Accept}$.

**Prednosti kriptosistemov z javnimi ključi**

- Ni zahteve po varnem kanalu.
- Vsak uporabnik ima 1 par ključev.
- Poenostavljeno upravljanje s ključi.
- Omogoča preprečevanje tajenja.

**Pomanjkljivosti kriptosistemov z javnimi ključi**

- Sheme z javnimi ključi so počasnejše.
- Javni ključi so večji od simetričnih.





V praksi uporabljamo skupaj sheme s simetričnimi in javnimi ključi in jim rečemo **hibridne sheme**

**Primer:** Da bi Bojanu poslala podpisano tajno sporočilo $m$, Anita naredi naslednje:

- izračuna $s = \mathsf{Sign}(S_A, m)$,
- izbere tajni ključ $k$ simetrične šifrirne sheme (AES),
- pridobi overjeno kopijo Bojanovega javnega ključa $J_B$,
- pošlje $c_1 = E(J_B, k)$, $c_2 = \mathsf{AES}(k, (m, s))$.

Za odkritje sporočila $m$ in preverjanje avtentičnosti, Bojan:

- odšifrira $c_1$: $k = D(S_B, c_1)$,
- odšifrira $c_2$ z uporabo ključa $k$, da dobi $(m, s)$,
- pridobi overjeno kopijo javnega ključa $J_A$,
- preveri podpis $s$ sporočila $m$.





Že l. 1977 so Ronald L. **Rivest**, Adi **Shamir** in Leonard M. **Adleman** naredili prvo realizacijo takšnega kriptosistema (**RSA**)
(tajno pa že l. 1973 **C. Cocks** pri GCHQ).

Temu so sledili številni drugi nesimetrični kriptosistemi, med katerimi pa so danes najbolj pomembni naslednji:

- RSA (faktorizacija),
- Merkle-Hellman Knapsack (metoda nahrbtnika)
- Chor-Rivest
- McEliece (linearne kode),
- ElGamal (diskretni logaritem),
- eliptične krivulje.

Javni kriptosistemi **niso** nikoli brezpogojno varni, zato študiramo računsko/časovno zahtevne sisteme.