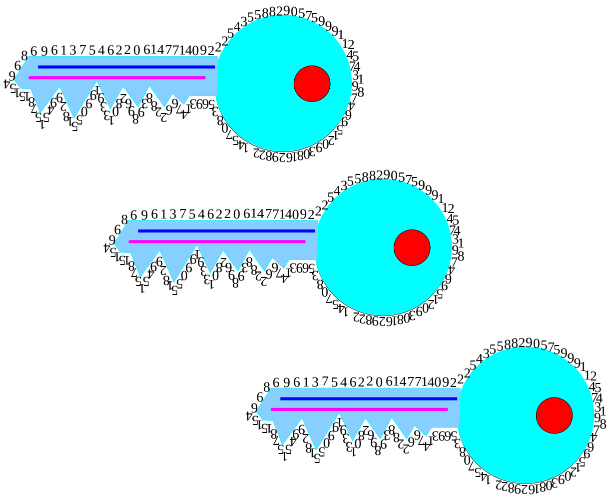
PAMETNE KARTICE V ZASEBNEM ŽIVLJENJU JAVNIH KLJUČEV

Aleksandar Jurišić
Center za kriptografijo in računalniško varnost
Politehnika Nova Gorica

http://valjhun.fmf.uni-lj.si/~ajurisic



Odkar so ljudje pričeli komunicirati,
pa naj si bo to preko govora, pisave, radija,
telefona, televizije ali računalnikov,
so želeli tudi **skrivati** vsebino svojih sporočil.

Ta nuja, oziroma že kar obsedenost
po **tajnosti**, je imela dramatičen vpliv
na vojne, monarhije in seveda tudi
na individualna življenja.

Vladarji in generali so odvisni od uspešne in učinkovite komunikacije že tisočletja, hkrati pa se zavedajo posledic, v primeru, če njihova sporočila pridejo v napačne roke, izdajo dragocene skrivnosti rivalom ali odkrijejo vitalne informacije nasprotnikom.

Danes vse to velja tudi za moderna vodstva uspešnih podjetij in tako postaja

“informacijska/računalniška varnost”

eno izmed najbolj pomembnih gesel *informacijske dobe*.

Vlade, industrija ter posamezniki,
vsi hranijo informacije v **digitalni obliki**.

Ta medij nam omogoča številne prednosti
pred fizičnimi oblikami:

- je zelo kompakten,

- prenos je takorekoč trenuten,

hkrati pa je omogočen tudi

- organiziran dostop do raznovrstnih podatkovnih baz.

Z razvojem

- telekomunikacij,
- računalniških omrežij in
- obdelovanja informacij

pa je precej lažje prestreči in spremeniti *elektronsko informacijo* kot pa njenega *papirnega predhodnika*.

Zato so se povečale zahteve po **varnosti**.

Informacijska in računalniška varnost

opisuje vse preventivne postopke in sredstva
s katerimi preprečimo nepooblaščeno uporabo
elektronskih podatkov ali sistemov,
ne glede na to ali gre pri ustreznih podatkih kot sta

digitalni denar (nosilec vrednosti) in
digitalni podpis (za prepoznavanje)

za

- razkritje,
- spreminjanje,
- zamenjavo,
- uničenje,
- preverjanje verodostojnosti.

Predlagani so bili številni ukrepi, a niti eden med njimi ne zagotavlja *popolne varnosti*.

Med preventivnimi ukrepi, ki so na voljo danes, nudi

kriptografija

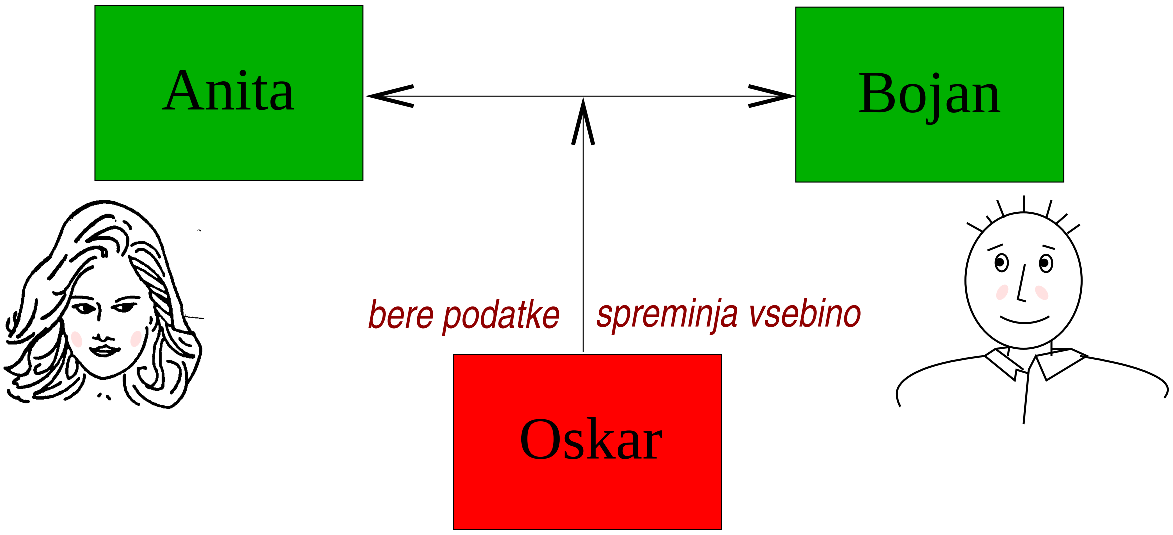
(če je seveda pravilno implementirana ter uporabljana)

največjo stopnjo varnosti

glede na svojo prilagodljivost digitalnim medijem.

Kaj je kriptografija?

Kriptografija je veda o komunikaciji v prisotnosti aktivnega napadalca.



Primer:

pošiljanje papirnih dokumentov po pošti

Kakšna zagotovila varnosti so na voljo? In kako?

- **Fizična varnost:** zapečatene kuverte.
- **Zakonska infrastruktura:**
  ročni podpis je zakonsko sprejeto sredstvo,
  zakoni proti odpiranju/oviranju pošte, itd.
- **Poštna infrastruktura:**
  varni in sprejeti mehanizmi za
  dostavljanje pošte širom po svetu.

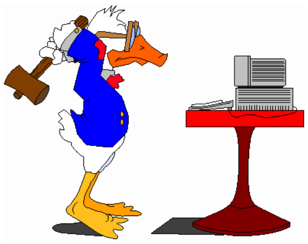
Primer: elektronski podatki

- **ZA**: hranjenje je enostavno in poceni, hiter in enostaven transport.
- **PROTI**: enostavno kopiranje; transportni mediji niso varni (npr. pogovor po mobilnem telefonu, internetna seja, ftp seja, komunikacija s pomočjo elektronske pošte).
- **Vprašanje**: Kako lahko omogočimo/ponudimo enake možnosti za papirni kakor tudi elektronski svet?

Osnovni cilji kriptografije

1. **Zaupnost**: ohranjevanje tajnosti pred vsemi razen pooblaščenimi.
2. **Celovitost**: zagotovilo, da informacija ni bila spremenjena (z nepooblaščenimi sredstvi).
3. **Overjanje podatkov**: potrditev izvora informacij.
4. **Pristnost**: potrditev identitete.
5. **Preprečevanje tajenja**: preprečiti neizpolnitev sprejetih obvez ali akcij.

Dešifriranje (razbijanje) klasičnih tajnopisov



Kriptografske sisteme kontroliramo s pomočjo ključev, ki določijo transformacijo podatkov.
Seveda imajo tudi ključi elektronsko 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.

Vohunova dilema

Bilo je temno kot v rogu, ko se je vohun vračal v grad po opravljeni diverziji v sovražnem taboru.

Ko se je približal vratom, je zaslišal šepetajoč glas:

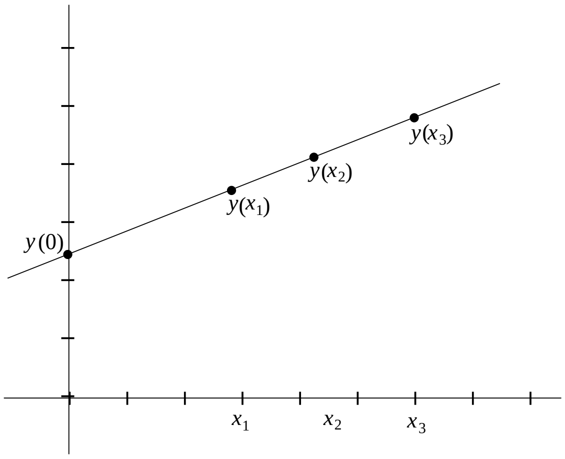
Kako vohun prepriča "stražarja", da pozna geslo, ne da bi ga izdal morebitnemu vsiljivcu/prisluškovalcu?

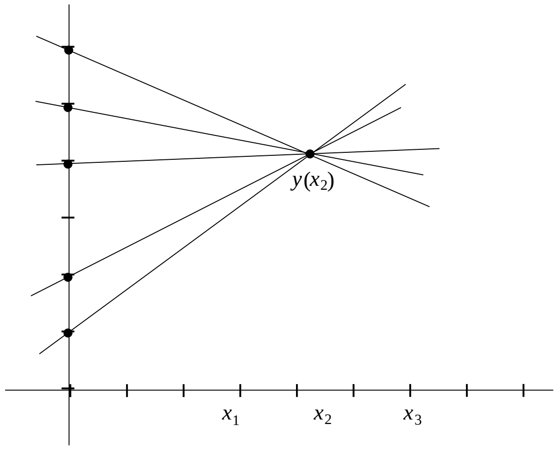
Deljenje skrivnosti

Problem: *V banki morajo trije direktorji odpreti trezor vsak dan, vendar pa ne želijo zaupati kombinacijo nobenemu posamezniku. Zato bi radi imeli sistem, po katerem lahko odpreta trezor poljubna dva med njimi.*

Ta problem lahko rešimo z $(2, 3)$-stopenjsko shemo.

Stopenjske sheme za deljenje skrivnosti sta leta 1979 neodvisno odkrila **Blakey in Shamir**.





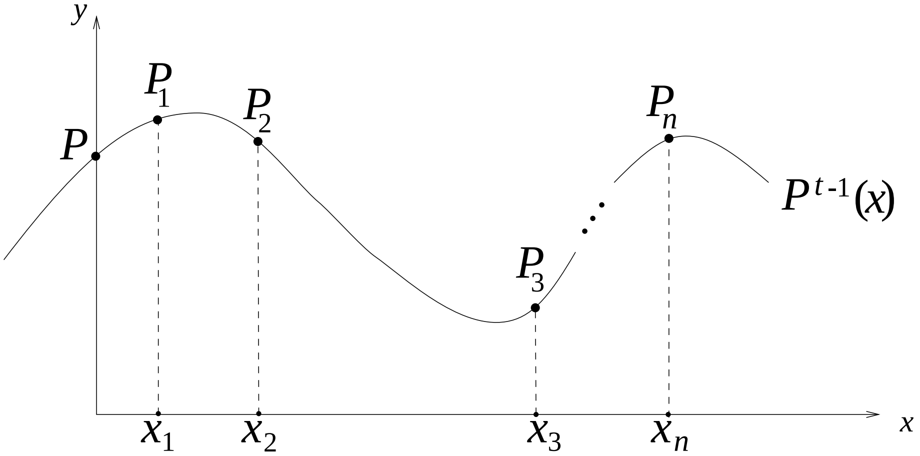
Vsak dobi le y-koordinato svoje točke.
Program v trezorju ima še ustrezne od 0 različne x- koordinate, zato lahko izračuna ključ $y(0)$.
Vsaki točki natanko določata premico in s tem ključ.

Če imamo eno samo točko, ne moremo ugotoviti, kateri ključ je pravi, saj so vsi videti enako dobri.

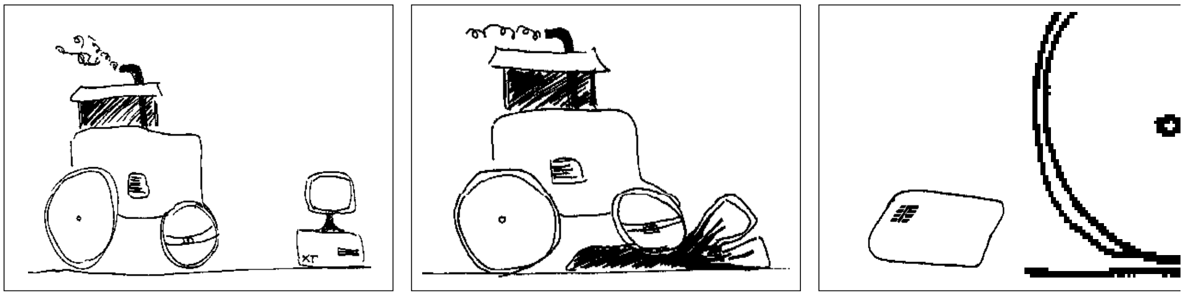
V splošnem je (t, n)**-stopenjska shema** za deljenje skrivnosti K med n oseb (množica $\mathcal{P}$), $2 \leq t \leq n$, metoda, za katero velja

- poljubnih t oseb lahko izračuna vrednost K,
- nobena skupina s $t-1$ osebami (ali manj) ne more izračunati prav nobene informacije o vrednosti K.

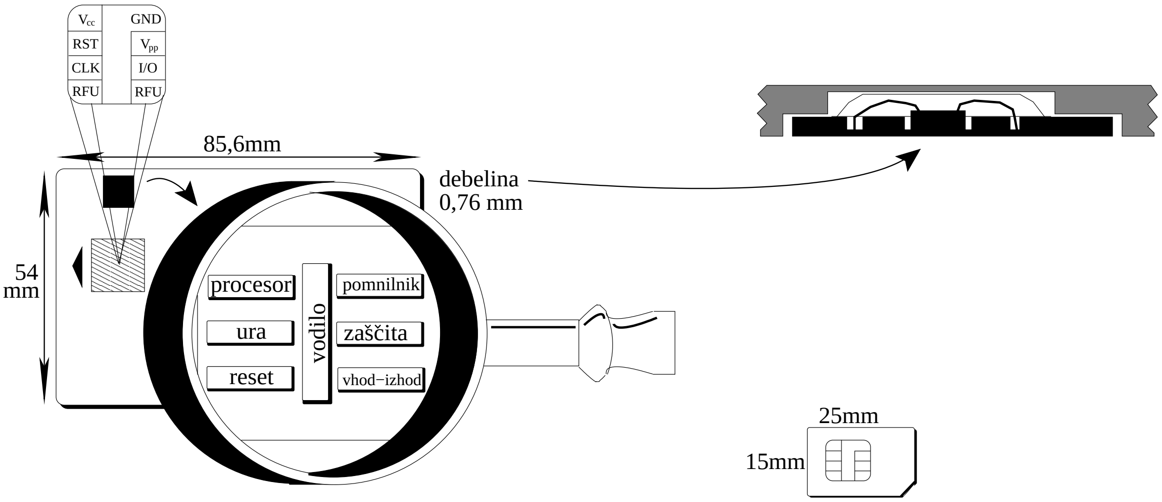
Varnost te sheme mora biti *brezpogojna*, tj. neodvisna od kakšnega računsko zahtevnega problema, kot je na primer faktorizacija v primeru RSA.



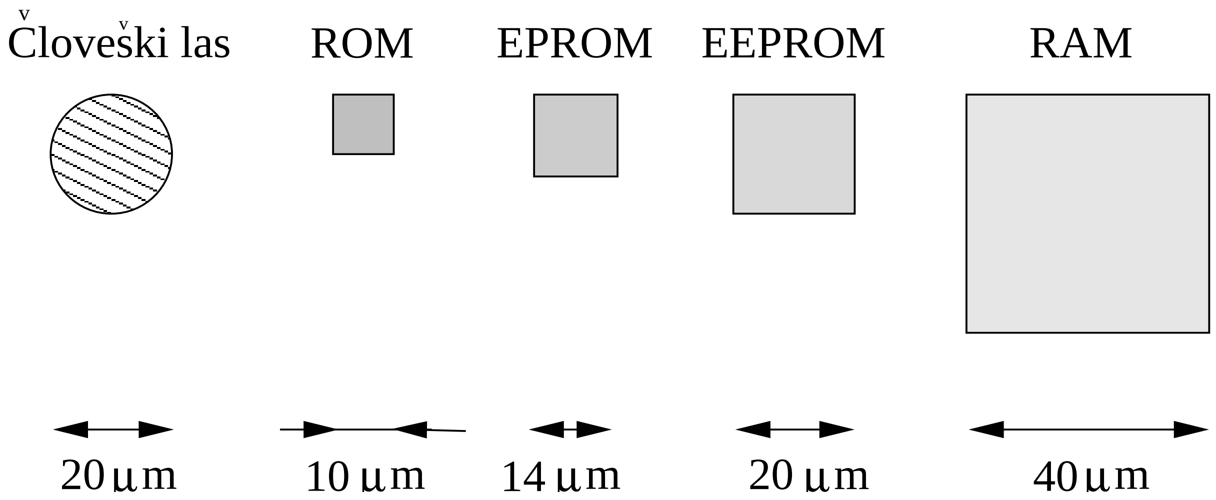
Pametne kartice



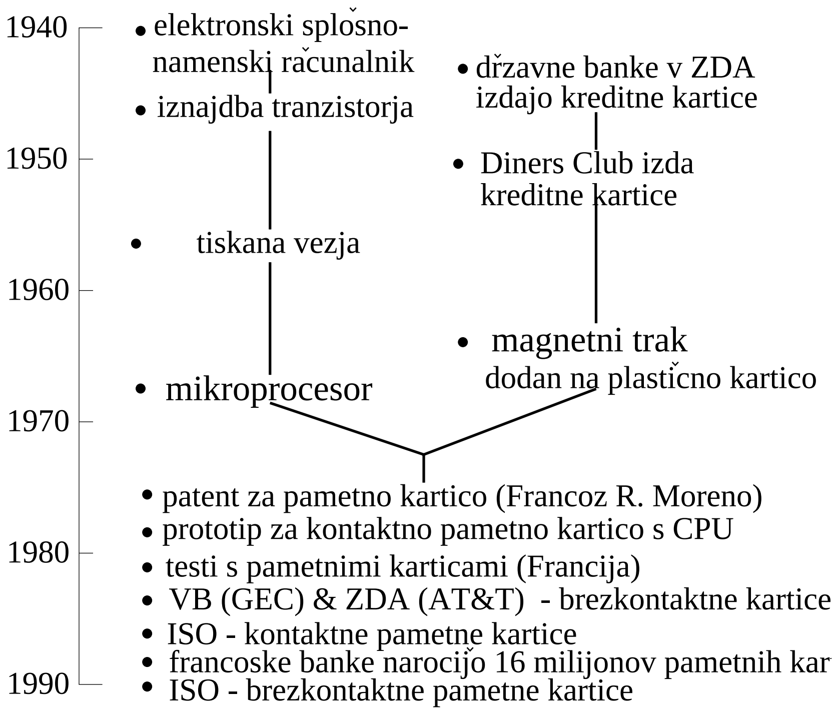
Po računski moči so pametne kartice primerljive z originalnim IBM-XT računalnikom,
kartice s **kripto koprocesorjem** pa v nekaterih opravilih prekašajo celo 50 Mhz 486 računalnik.



Velikost pametne kartice ustreza ISO 7810 standardu, sestavljajo pa jo mikroprocesor, pomnilnik (ROM, RAM, EEPROM), vhodno/izhodna enota (I/O).

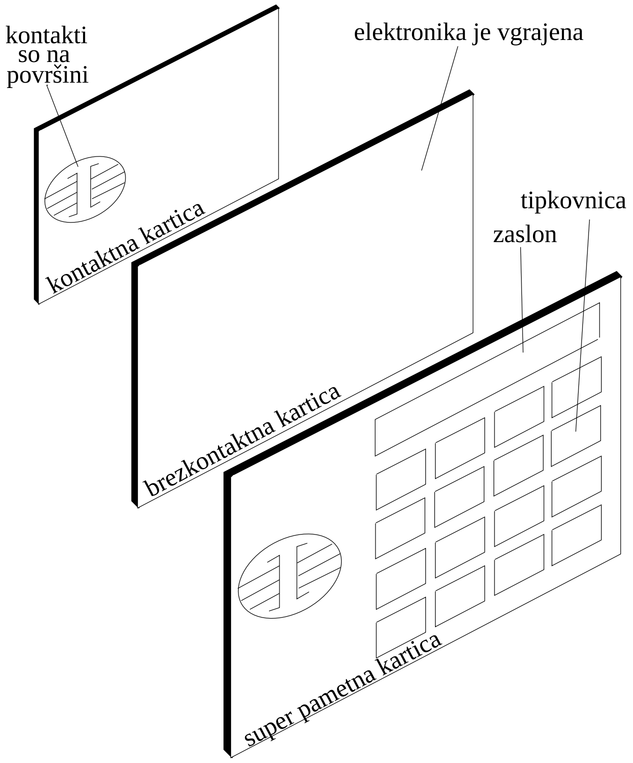


Velikost posameznih pomnilniških celic, ki lahko hranijo 1 bit informacije. Velikosti so približne in se nanašajo na 0.8μm tehnologijo.



V naslednjih petih letih pričakujemo hiter razvoj, vse dokler ne bodo pametne kartice zares postale računalnik v našem žepu.

Vrste pametnih kartic



Zakaj pametna kartica

Gotovo je najbolj pomembna razlika med pametno kartico in magnetno kartico

varnost.

Pametna kartica ima svoj **procesor**, ki kontrolira vse interakcije med od zunaj nedostopnim spominom in različnimi zunanjimi enotami.

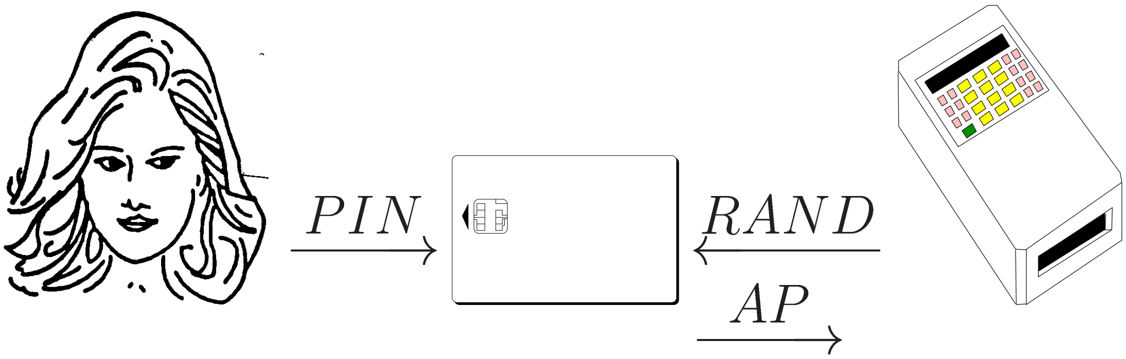
Dodaten, pomemben, del pametne kartice je **non-volatile spomin (ROM)**,
t.j. spomin, ki se ga ne da spremeniti in
ostane prisoten tudi po prekinitvi napajanja.

Zagotovitev varnosti

Identifikacija se opravi v dveh delih:

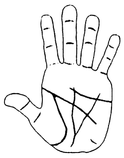
(a) kartica mora biti zares prepričana, da jo uporablja njen lastnik (lokalno overjanje),

(b) kartica komunicira (varno) z računalnikom (dinamično overjanje).



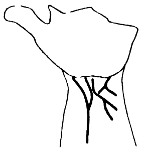
Biometrični testi

prstni odtisi

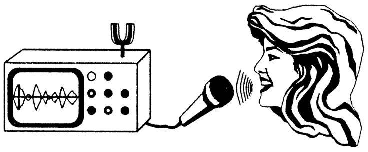


geometrija roke

odtis noge (otroci)

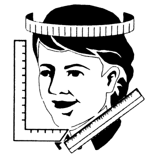


vzorec ven

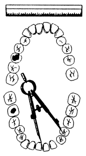


prepoznavanje glasu

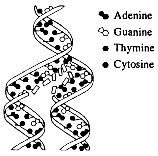
vzorec zenice



prepoznavanje obraza



zapis zob



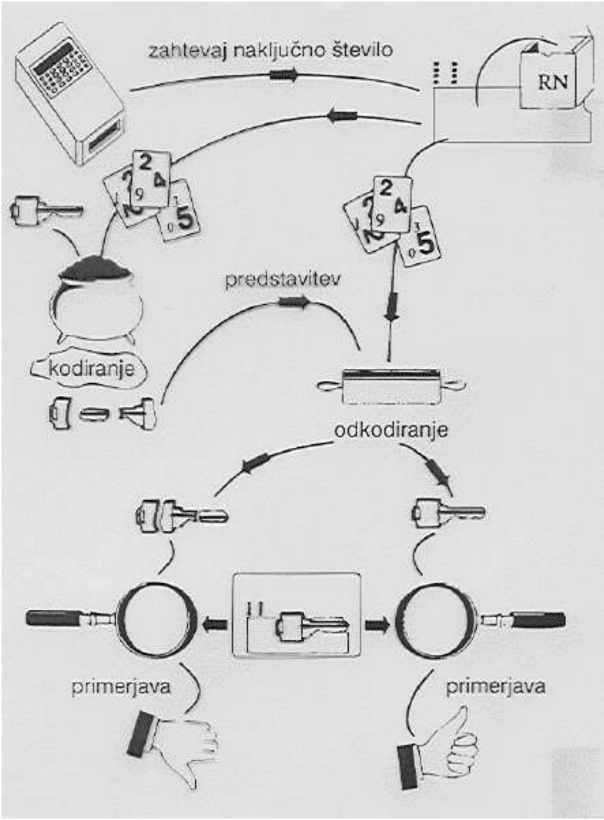
RNK (DNA)

podpis

Pametna kartica zgenerira naključno število, ter ga pošlje čitalniku.

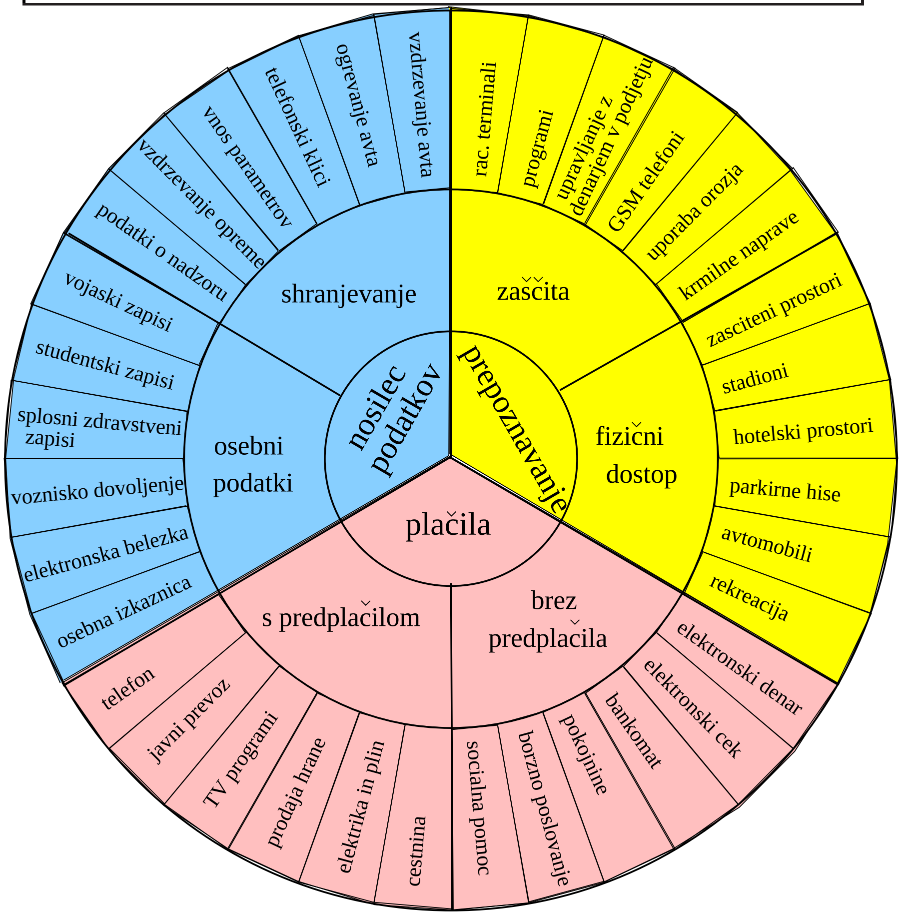
Ta ga zašifrira z zasebnim ključem in rezultat pošlje pametni kartici.

Če pametna kartica uspešno dešifrira naključno število z javnim ključem, potem je prepričana o pristnosti čitalnika.

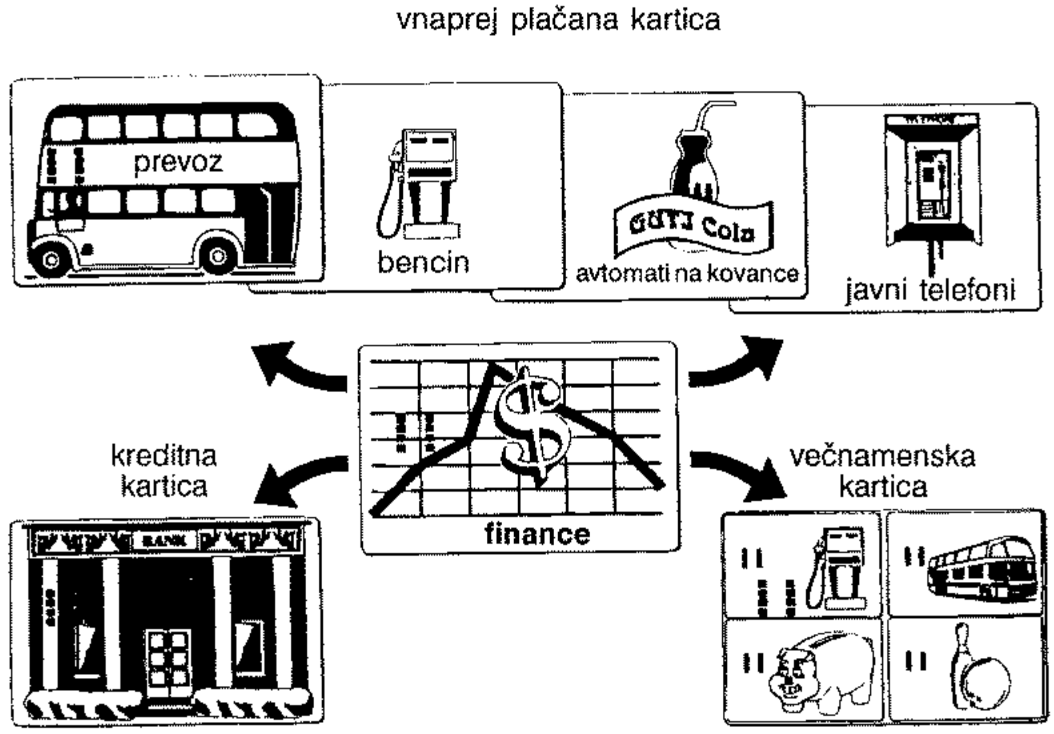


Enak proces poteka v nasprotni smeri.

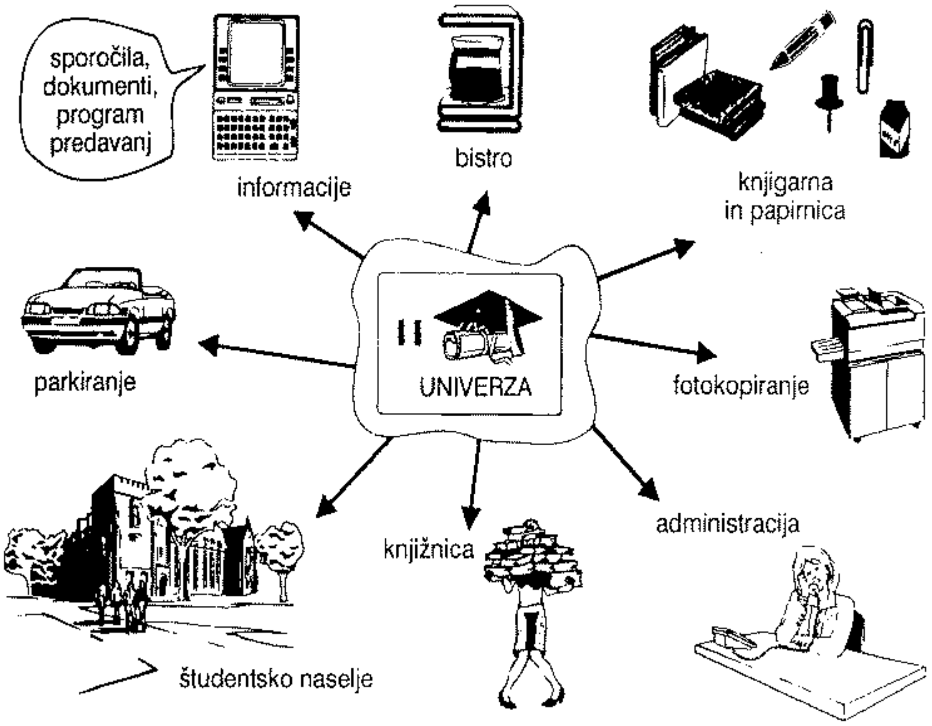
Uporaba pametnih kartic



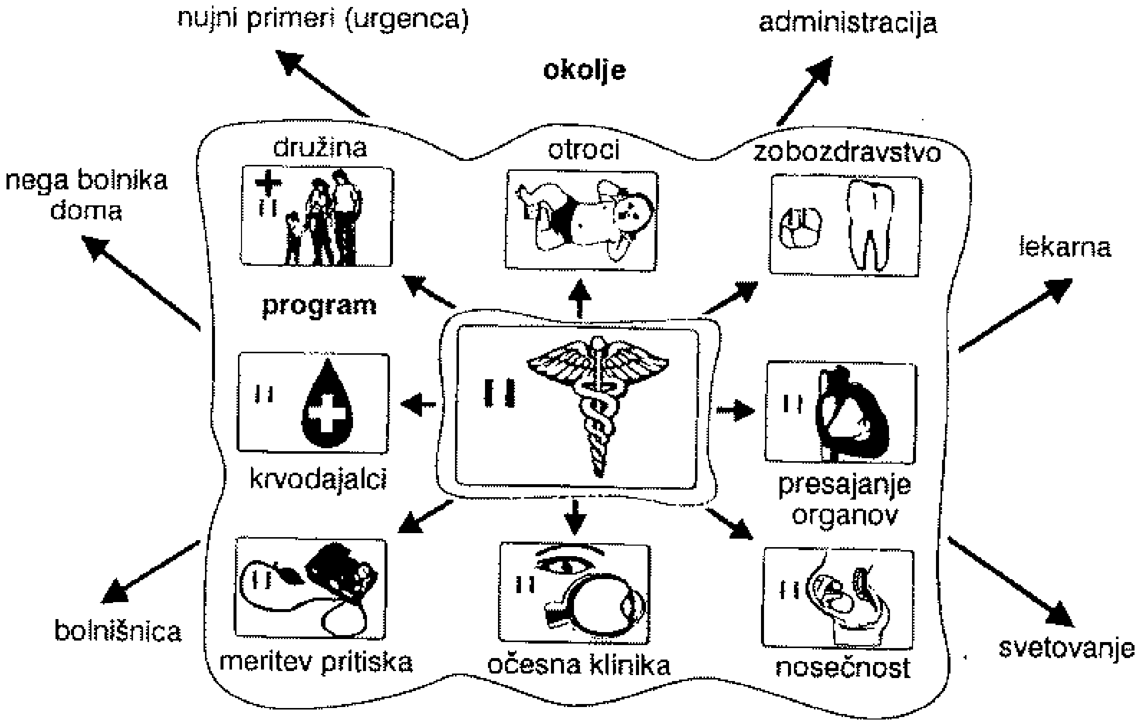
Plačilne, kreditne in večnamenske kartice, ki se uporabljajo na področju *financ*.



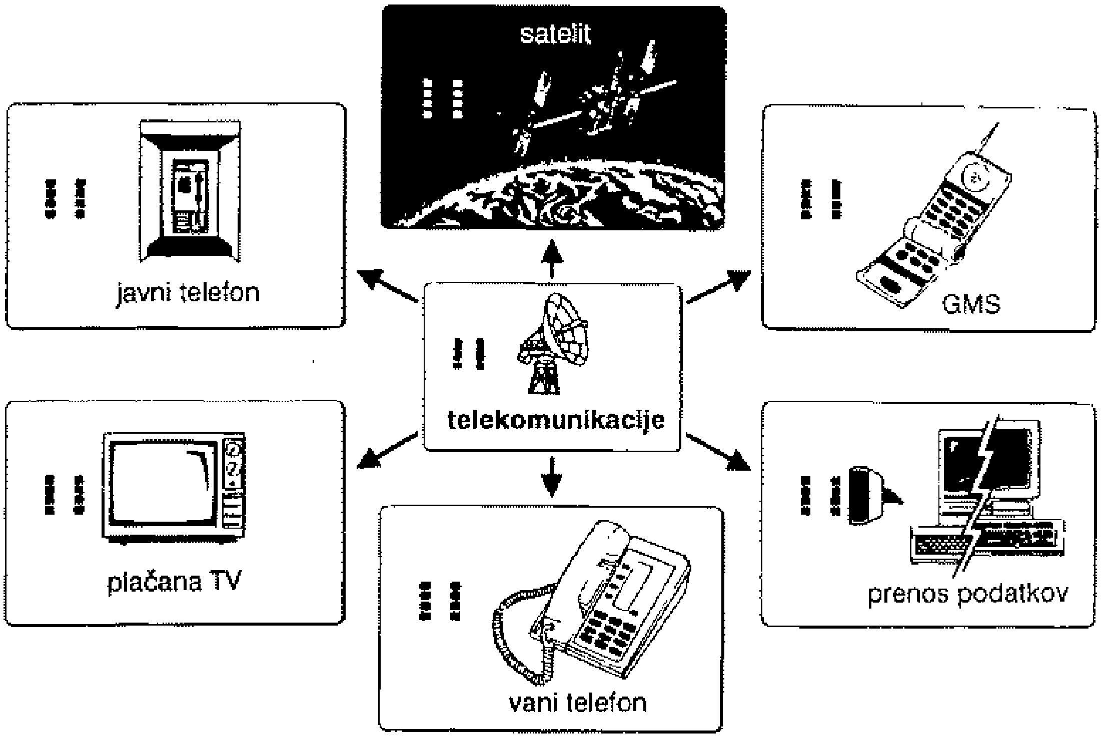
Uporaba pametnih kartic na *univerzi/fakulteti*, ki je ponekod mesto v malem.



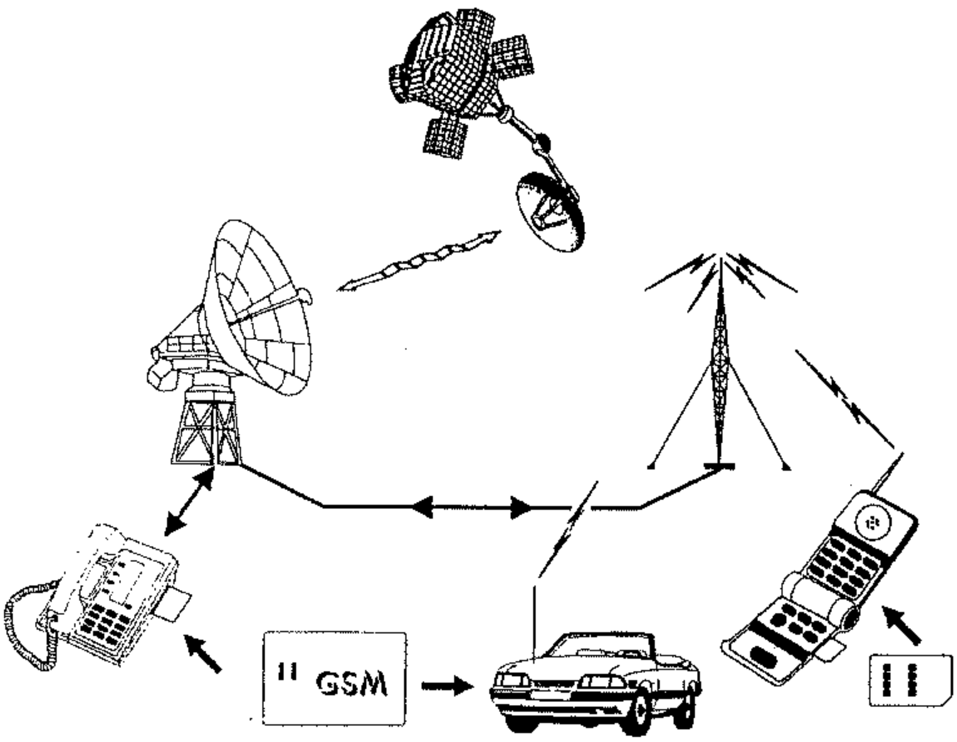
Področja v *zdravstvu*, kjer se uporabljajo pametne kartice.



Uporaba pametne kartice v *telekomunikacijah* in uporabniški elektrotehniki.

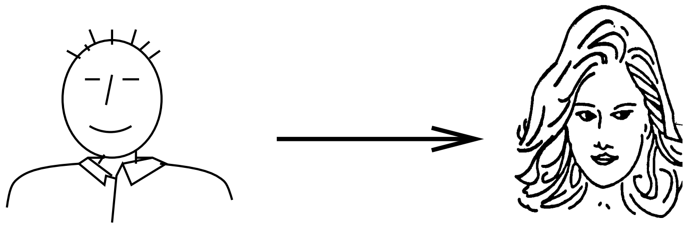


GSM (globalni sistem za prenosno komuniciranje)



Koncept javne kriptografije

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]**.

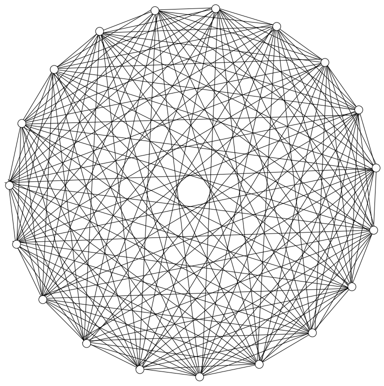
Predpostavimo, da se Anita in Bojan prej dogovorita za **skupen ključ**, ki ga ne pozna nihče drug (simetričen kriptosistem).

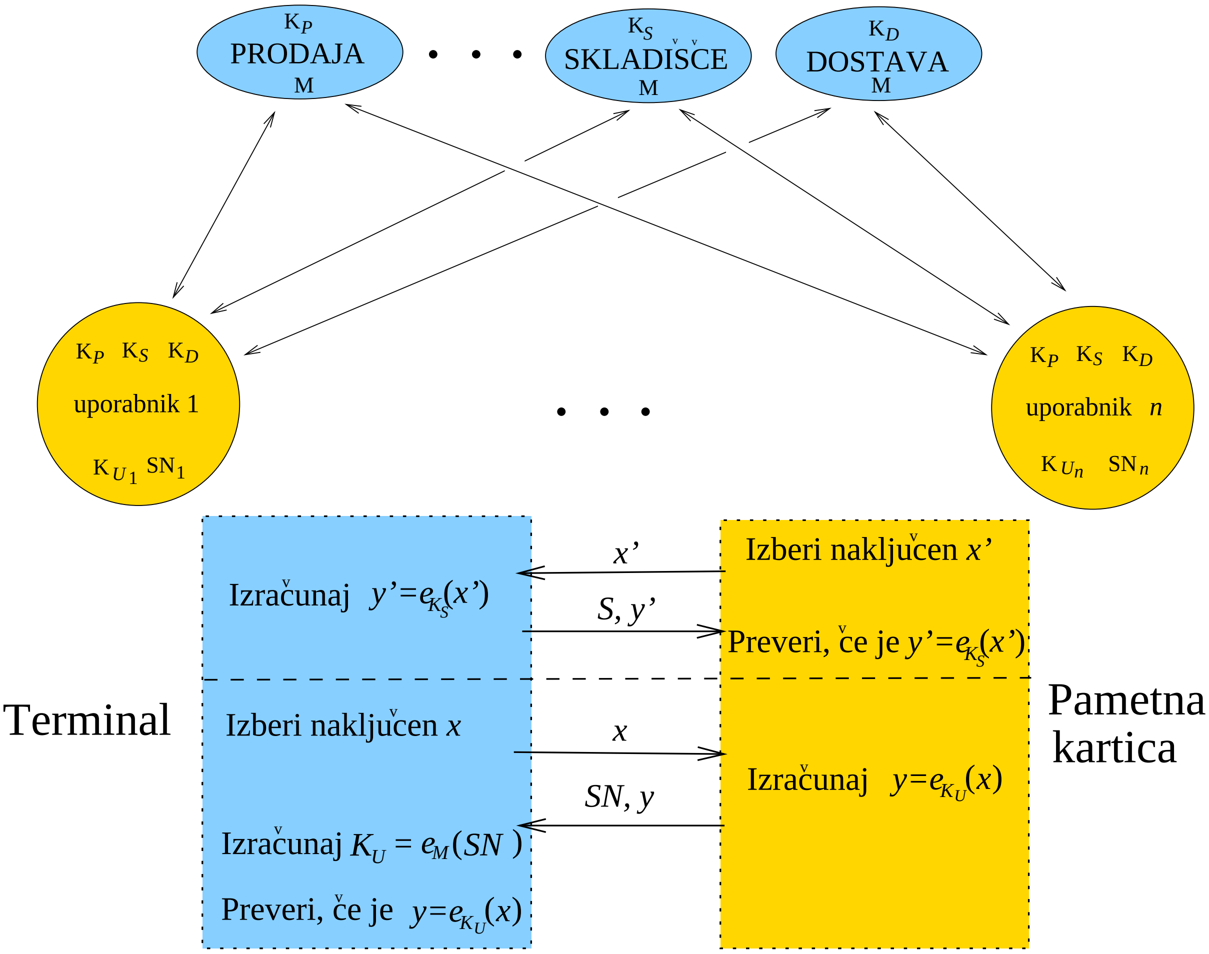
Če Bojan z njim zašifrira pismo, je lahko prepričan, da ga lahko odklene le Anita.

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.



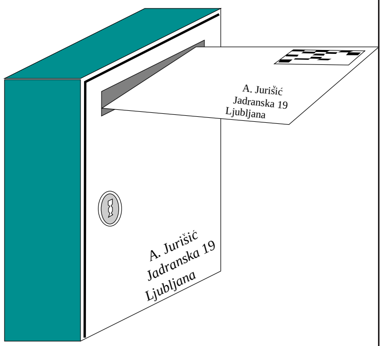


K_P
PRODAJA
M
K_S
SKLADIŠČE
M
K_D
DOSTAVA
M
K_P K_S K_D
uporabnik 1
K_{U_1} SN_1
K_P K_S K_D
uporabnik n
K_{U_n} SN_n
Izračunaj $y'=e_{K_S}(x')$
Izberi naključen x'
x'
S, y'
Preveri, če je $y'=e_{K_S}(x')$
Terminal
Izberi naključen x
x
Izračunaj $y=e_{K_U}(x)$
SN, y
Izračunaj $K_U = e_M(SN)$
Preveri, če je $y=e_{K_U}(x)$
Pametna kartica

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.



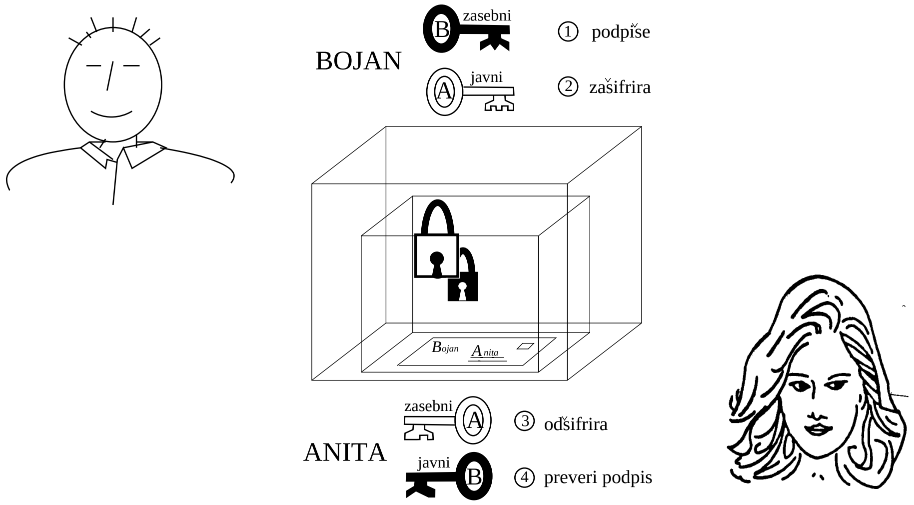
To omogoči lastniku, da en ključ objavi, drugega pa hrani v tajnosti (npr. na pametni kartici). Zato imenujemo ta ključa zaporedoma **javni** in **zasebni**.

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.

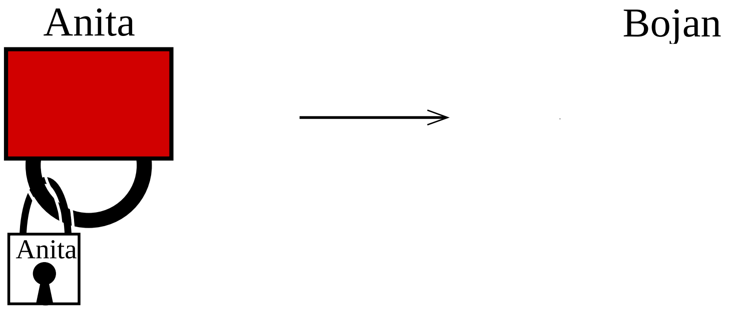
Bojan pošlje Aniti podpisano zasebno pismo:

(1) podpiše ga s svojim zasebnim ključem Z_B in ga
(2) zašifrira z Anitinim javnim ključem J_A.

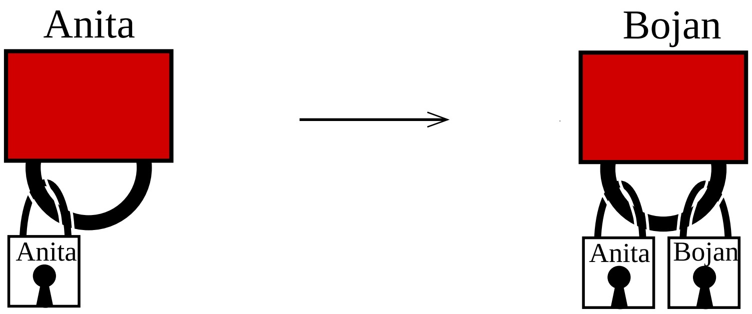


(3) Anita ga s svojim zasebnim ključem Z_A odšifrira,
(4) z Bojanovim javnim ključem J_B preveri podpis.

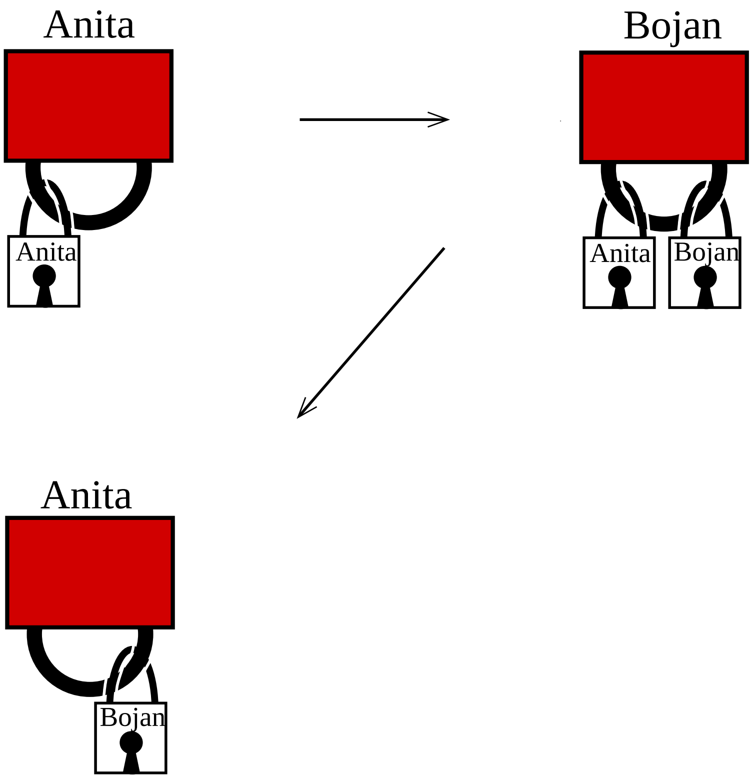
Protokol Massey-Omura



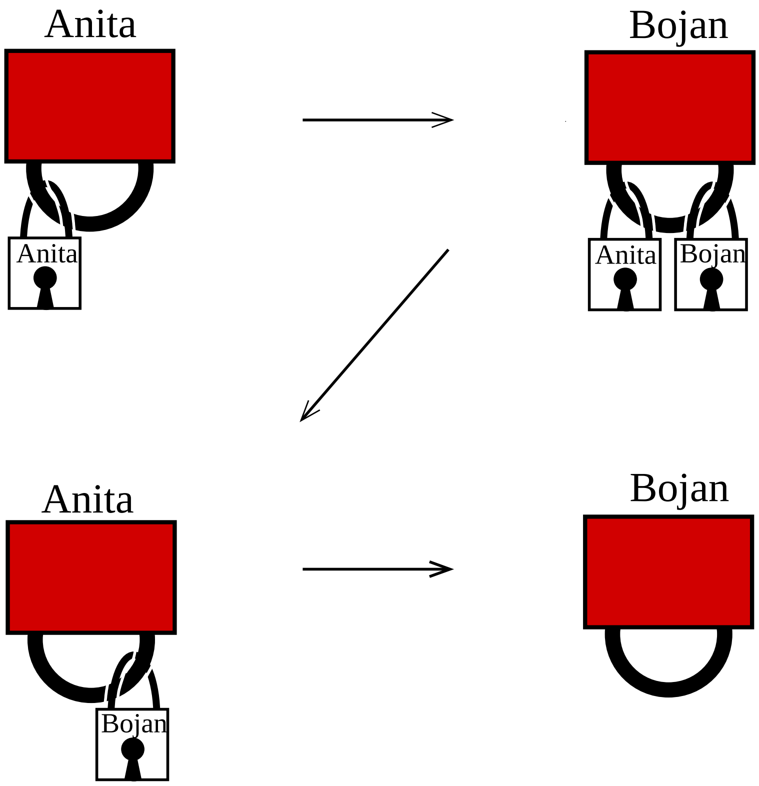
Protokol Massey-Omura (1. korak)



Protokol Massey-Omura (2. korak)



Protokol Massey-Omura (3. korak)



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)

Problem diskretnega logaritma

v končni grupi G:

za dana $\alpha, \beta \in G$*, kjer je red elementa* α *enak* n*, najdi* $x \in \{0, \ldots, n-1\}$*, tako da je* $\alpha^x = \beta$*.*

Število x se imenuje **diskretni logaritem** osnove α elementa β.

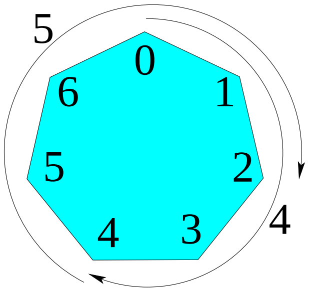
Oznaka **$\log_\alpha \beta$** ali pa še bolje **$\mathrm{Ind}_\alpha \beta$**.

Osnovna razloga za uporabo različnih grup:

- operacije v nekaterih grupah so izvedene enostavneje v programih (software) in programski opremi (hardware) kot v drugih grupah,
- problem diskretnega logaritma je v določeni grupi lahko zahtevnejši kot v drugi.

ElGamalovi protokoli

Delimo jih v tri razrede:



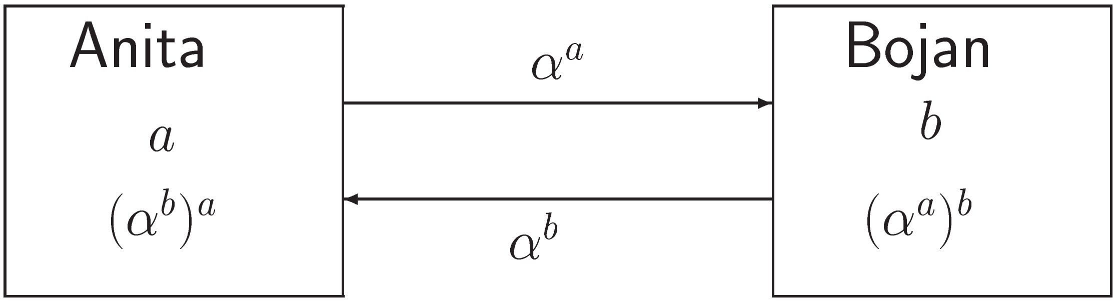
1. protokoli za izmenjavo ključev,
2. sistemi z javnimi ključi,
3. digitalni podpisi.

V kriptografiji si omislimo končne množice, kot pri številčnici na uri (npr. praštevilski obsegi $\mathbb{Z}_p$).

Primer: za $p = 7$ velja $4+5=2$ in $5 \times 4 = 6$ (saj ima vsota 9 ostanek 2 pri deljenju s 7, produkt 20 pa ostanek 6), možno pa je tudi deljenje.

1. Izmenjava ključev

(Diffie-Hellman)



Anita in Bojan si delita skupni element grupe: α^{ab}.

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

2. Sistem javnih ključev

(dva ključa, asimetrični sistem)

$m \in \mathcal{M} = G$

privatni a

α^a

Anita

$$\xrightarrow{(\alpha^a,\, m\alpha^{ab})}$$

privatni b

javni α^b

Bojan

3. Digitalni podpis (DSA)

Genereranje ključa osebe A:

1. *Izberi praštevilo* q, $2^{159} < q < 2^{160}$.
2. *Izberi praštevilo* p, *tako da je* $2^{1023} < p < 2^{1024}$ *in* $q|p-1$.
3. *Izberi* $t \in \mathbb{Z}_p^*$ *in izračunaj* $\alpha = t^{(p-1)/q} \mod p$ ($\alpha \neq 1$ *je gen. cikične grupe reda* q v $\mathbb{Z}_p^*$).
4. *Izberi naključno število* a *manjše od* q (**zasebni ključ**).
5. *Izračunaj* $y = \alpha^a \mod p$.

Javni ključ osebe A je (p, q, α, y).

Podpis sporočila m osebe A:

1. *Izberi naključno naravno število* $k < q$.
2. *Izračunaj* $r = (\alpha^k \bmod p) \bmod q$.
3. *Izračunaj* $k^{-1} \bmod q$.
4. $s = k^{-1}(h(m) + ar) \bmod q$,
   kjer je h *zgoščevalna funkcija SHA-1 (Secure Hash Algorithm) in* $s \neq 0$.
5. **Podpis** *je par* (s, r).

Obstaja veliko variant zgornjega podpisa.

Prevarjanje podpisa (r, s) sporočila m osebe A:

1. *Preskrbi si avtentično kopijo javnega ključa* (p, q, α, y) *osebe* A.
2. *Izračunaj* $w = s^{-1} \bmod q$ *in* $h(m)$.
3. *Izračunaj* $u_1 = h(m)w \bmod q$
   in $u_2 = rw \bmod q$.
4. *Izračunaj* $v = (\alpha^{u_1} y^{u_2} \bmod p) \bmod q$.
5. *Sprejmi podpis če in samo če je* $v = r$.

Preiskus: $(s = k^{-1}(h(m) + xr) \bmod q,\ w = s^{-1} \bmod q,$

$y = g^x \bmod p, \quad u_1 = h(m)w \bmod q, \quad u_2 = rw \bmod q\ )$:

$$
\begin{aligned}
v &= (g^{u_1} y^{u_2} \bmod p) \bmod q \\
&= (g^{h(m)w} g^{xrw} \bmod p) \bmod q \\
&= (g^{(h(m)+xr)w} \bmod p) \bmod q \\
&= (g^{k} \bmod p) \bmod q \\
&= r
\end{aligned}
$$

Odlična uvodna knjiga o moderni kriptografiji je:
Albrecht Beutelspacher, **Cryptology**, MAA, 1994.
(pri nas v CTK 206835 INF-A)

DSA (brez uporabe zgoščevalnih funkcij) podvoji dolžino podpisanega sporočila.

Resnejši problem nastane, ker je mogoče preurejati dele podpisanega sporočila ali pa nekatere celo izpustiti/dodati.

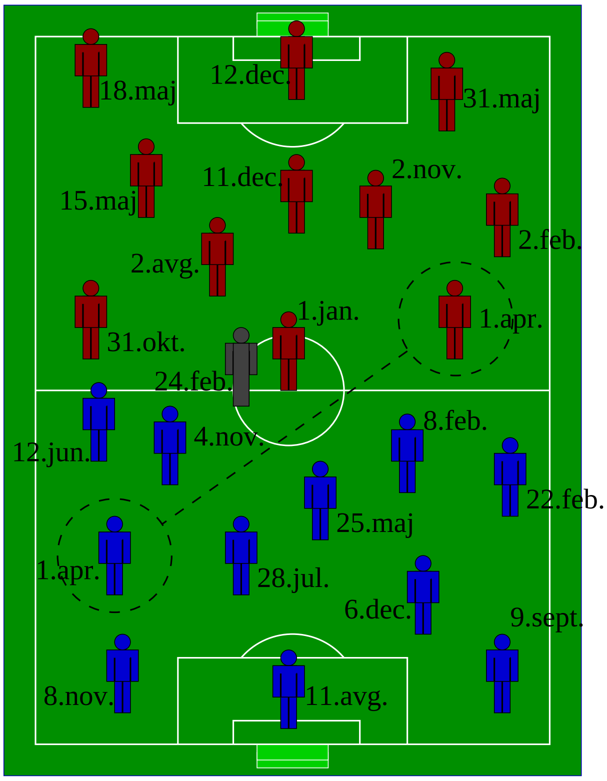
Celovitost podatkov ne more biti zagotovljena izključno s podpisovanjem majhnih delov dokumenta, zato vpeljemo **zgoščevalne funkcije** (angl. Hash Functions), ki poljubno dolgemu sporočilu priredijo kratko zaporedje bitov, ki jih potem podpišemo.

Kakšno naključje!!! Mar res?

Na nogometni tekmi sta na igrišču dve enajsterici in sodnik, skupaj **23 oseb**.

Kakšna je verjetnost, da imata **dve osebi** isti rojstni dan?

Ali je ta verjetnost lahko večja od 0.5?



Napad s paradoksom rojstnih dnevov!

Grupa na eliptični krivulji

Za kriptografijo sta jo leta 1985 prva predlagala Neal Koblitz in Victor Miller.

Eliptična krivulja E nad obsegom $\mathbb{Z}_p$ je definirana z Weierstrassovo enačbo:

$$\boldsymbol{y^2 = x^3 + ax + b} \tag{1}$$

kjer sta $a, b \in \mathbb{Z}_p$ in $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ ($GF(2^m)$: $y^2 + xy = x^3 + ax^2 + b$).

$$E(\mathbb{Z}_p) := \{(x, y) \,|\, x, y \in \mathbb{Z}_p, \text{ ki ustrezajo (1)}\} \cup \mathcal{O}.$$

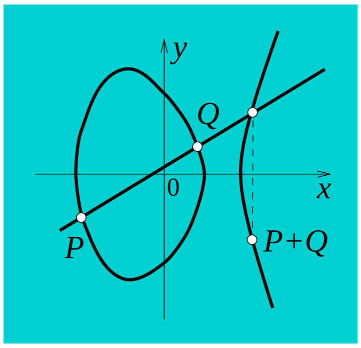
Pravilo za seštevanje

1. $P=(x_1,y_1)$, $Q=(x_2,y_2)\in E(\mathbb{Z}_p)$,
   kjer $P\neq -Q := (x_2,-y_2)$.

   Potem je $P+Q=(x_3,y_3)$, kjer je

   $x_3=\lambda^2-x_1-x_2, \quad y_3=\lambda(x_1-x_3)-y_1$, in

$$\lambda = \begin{cases} \dfrac{y_2-y_1}{x_2-x_1} & ; \text{za } P\neq Q \\[2ex] \dfrac{3x_1^2+a}{2y_1} & ; \text{za } P=Q. \end{cases}$$



2. $P+\mathcal{O}=\mathcal{O}+P=P \quad$ in $\quad P+(-P)=\mathcal{O}$
   za vsak $P\in E(\mathbb{Z}_p)$.

Izrek (Hasse). $p+1-2\sqrt{p} \leq |E| \leq p+1+2\sqrt{p}.$

Schoofov algoritem (1985) izračuna $|E|$ v $O((\log p)^8)$ bitnih operacijah.

Grupa E je izomorfna $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$, kjer je $n_2|n_1$ in $n_2|(p-1)$, tako da lahko najdemo ciklično podgrupo $\mathbb{Z}_{n_1}$, ki jo uporabimo za ElGamalov kriptosistem.

Podeksponentno metodo **index calculus** zaenkrat ne znamo uporabiti pri DLP na eliptični grupi (razen če ni eliptična krivulja supersingularna).

Zato si lahko izberemo eliptično krivuljo s ciklično podgrupo velikosti (samo) okoli 2^{160}.

Uganjevanje fraz, ki jih uporabljamo za gesla

primer	število znakov	zahtevnost	dolžina gesla	čas za razbijanje
mucka	5	25 (majhne črke)	12 bitov	40 minut
br1a9Az	7	62 (črke in številke)	24 bitov	22 let
TH,X1lb<V+	10	95 (znaki na tipkov.)	40 bitov	nedosegljivo

Če uporabimo angleško ali slovensko besedo, dobimo zaporedje s približno 1.3 biti entropije na en znak (t.j. prostor za besedo proti popolnoma naključnim znakom).

Napadi z grobo silo (angl. Brute Force Attack)

posameznik ima 1 PC in programsko opremo
($2^{17} - 2^{24}$ ključev/sek.),

majhna skupina, 16 PC ($2^{21} - 2^{28}$ ključev/sek.),

akademska omrežja, 256 PC ($2^{25} - 2^{32}$ ključev/sek.),

veliko podjetje z \$1,000,000 za strojno opremo
(2^{43} ključev/sek.),

vojaška obveščevalna organizacija z \$1,000,000,000 za strojno opremo in napredno tehnologijo
(2^{55} ključev/sek.).

Napadi z grobo silo

dolžina ključa (v bitih)	**posamični napadalec**	**majhne skupine**	**razisko-valna omrežja**	**velika podjetja**	**vojaške obveščevalne službe**
40	tedni	dnevi	ure	milisekunde	mikrosekunde
56	stoletja	desetletja	leta	ure	sekunde
64	tisočletja	stoletja	destletja	dnevi	minute
80	∞	∞	∞	stoletja	stoletja
128	∞	∞	∞	∞	tisočletja

Povprečen čas za napad z grobo silo

dolžina ključev (v bitih)	**število možnih ključev**	**potreben čas za eno šifriranje/μsek.**	**potreben čas za 10^6 šifriranj/μsek.**
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{sec} \approx 36$ min	≈ 2 milisek.
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{sec} \approx 1142$ let	≈ 10 ur
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{sec} \approx 5 \times 10^{24}$	$\approx 5 \times 10^{18}$ let

Dolžina ključev

simetrične šifre (AES)	**asimetrične** (RSA, DSA, DH)	**eliptične krivulje**
40 bitov	274 bitov	80 bitov
56 bitov	384 bitov	106 bitov
64 bitov	512 bitov	132 bitov
80 bitov	1024 bitov	160 bitov
96 bitov	1536 bitov	185 bitov
112 bitov	2048 bitov	237 bitov
120 bitov	2560 bitov	256 bitov
128 bitov	3072 bitov	270 bitov

Digitalni podpisi v $\mathbb{Z}_p$ in na EC

grupa	$\mathbb{Z}_p^*$	$E(\mathbb{Z}_p)$
elementi	množica celih števil $\{1, 2, \ldots, p-1\}$	točke (x, y), ki zadoščajo enačbi eliptične krivulje E in še točka v neskončnosti
operacija	množenje po modulu p	seštevanje točk na eliptični krivulji
oznake	elementi: g, h množenje: $g \times h$ multiplikativni inverz: h^{-1} deljenje: g/h potenciranje: g^a	elementi: P, Q seštevanje: $P + Q$ nasprotna točka: $-Q$ odštevanje: $P - Q$ skalarno množenje točke: aP
problem diskretnega logaritma	Za dana $g, h \in \mathbb{Z}_p^*$ poišči tako celo število a da je $h = g^a \bmod p$.	Za dani točki $P, Q \in E(\mathbb{Z}_p)$ poišči tako celo število a da je $Q = aP$.

Grupe

Digital Signature Algorithm (DSA) eliptični analog ECDSA

DSA	ECDSA
1. Izberi praštevili p in q velikosti $2^{1023} < p < 2^{1024}$, $2^{159} < q < 2^{160}$, tako da $q \mid p-1$.	1. Izberi tako eliptično krivuljo $E\colon\ y^2 = x^3 + ax + b$ nad $\mathbb{Z}_q$, da je število $\lvert E(\mathbb{Z}_p)\rvert$ deljivo s praštevilom $n \approx 160$-bitov.
2. $t \in \mathbb{Z}_p^*$, izračunaj $g = t^{(p-1)/q} \bmod p$, potem je $g \neq 1$ in ima red q v $\mathbb{Z}_p^*$.	2. Izberi točko P na $E(\mathbb{Z}_q)$ katere red je praštevilo n.
3. Uporabi multiplikativno grupo $\{g^0, g^1, \ldots, g^{q-1}\}$	3. Uporabi aditivno grupo $\{\mathcal{O}, P, 2P, \ldots, (n-1)P\}$

Generiranje ključa pri DSA in ECDSA

DSA	ECDSA
1. Izberi naključno celo število $x \in [2, q-2]$, tj. **zasebni ključ**	1. Izberi naključno celo število $d \in [2, n-2]$, tj. **zasebni ključ**
2. Izračunaj $y = g^x \bmod p$, **javni ključ** je (p, q, g, y).	2. Izračunaj $Q = dP$, **javni ključ** je (E, n, q, Q).

DSA	ECDSA
q	n
g	P
x	d
y	Q

Podpisovanje sporočila m

DSA	ECDSA
1. Izberi naključno celo število $k \in [2, q-2]$, tj. **zasebni ključ**.	1. Izberi naključno celo število $k \in [2, n-2]$, tj. **zasebni ključ**.
2. Izračunaj $g^k \bmod p$, $r = (g^k \bmod p) \bmod q$, $0 \neq s = k^{-1}(h(m) + xr) \bmod q$.	2. Izračunaj $kP = (x_1, y_1)$, $r = x_1 \bmod n$, $0 \neq s = k^{-1}(h(m) + dr) \bmod n$.
Podpis je par (r, s).	

Preverjanje podpisa (r, s) sporočila m osebe A

DSA	ECDSA
1. Preskrbi si avtentično kopijo javnega ključa osebe A: (p, q, g, y)	(E, n, q, Q)
2. Izračunaj $s^{-1} \bmod p$ in $h(m)$, $u_1 = h(m)s^{-1} \bmod q$, $u_2 = rs^{-1} \bmod q$, $v = (g^{u_1}y^{u_2} \bmod p) \bmod q$.	2. Izračunaj $s^{-1} \bmod n$ in $h(m)$ $u_1 = h(m)s^{-1} \bmod n$, $u_2 = rs^{-1} \bmod n$, $u_1P + u_2Q = (x_0, y_0)$ in $v = x_0 \bmod n$.
Sprejmi podpis samo in samo če je $v = r$.	

Množenje točke na eliptični krivulji s skalarjem (kP) je analogno potenciranju v $\mathbb{Z}_p^*$.

Obstajajo številne tehnike za pospešitev modularnega potenciranja (uporaba spomina in vnaprejšnjega računanja). Takšne metode lahko večinoma uporabimo tudi pri eliptičnem skalarnem množenju.

Poznamo pa tudi učinkovite izboljšave eliptičnega skalarnega množenja, ki nimajo analoga pri modularnem potenciranju.

1. Eliptično krivuljo in končni obseg si lahko izberemo tako, da je eliptično skalarno množenje čim hitrejše. Npr. izberemo tak končni obseg $\mathbb{Z}_p$, da je p Mersenovo praštevilo in s tem pocenimo redukcijo.

2. Odštevanje na eliptični krivulji je enako učinkovito kot seštevanje. To ne velja za modularno deljenje, ki je dražje od množenja. *Predznačen binarni razvoj*.

3. *Kompleksno množenje*
(anomalne binarne krivulje (ABC)).

Implementacija

Potenciranje opravimo z algoritmom kvadriraj in množi:

$$\alpha^{21} = (\alpha)(\alpha^4)(\alpha^{16})$$

Najprej izračunamo faktorje α, α^2, α^4, α^8, α^{16}, in jih nato zmnožimo.

Namesto 20 množenj smo jih potrebovali le 6.

Ali je lahko kvadriranje hitrejše od (splošnega) množenja?

NE!

$$ab = \frac{(a+b)^2 - a^2 - b^2}{2}.$$

Če je kvadriranje ‘lahko’, potem tudi splošno množenje ni dosti težje od seštevanja.

DA!

V normalni bazi končnega obsega GF(2^n)
je kvadriranje ciklični zamik,
množenje pa ostane težko v splošnem.

Znano je, da ima vsak končni obseg GF(p^n),
p praštevilo, $n \in \mathbb{N}$, bazo nad podobsegom GF(p),
ki je naslednje oblike:

$$N = \{\alpha, \alpha^p, \cdots, \alpha^{p^{n-1}}\}$$

in ji pravimo **normalna** baza.

Naj bo $\alpha_i = \alpha^{p^i}$. Potem velja

$$(\alpha_i)^p = \alpha_{i+1}, \qquad (i \pmod n)).$$

Če $A = (a_0, a_1, \ldots, a_{n-1})$ v normalni bazi N, potem $A^p = (a_{n-1}, a_0, a_1, \ldots, a_{n-2})$, saj je

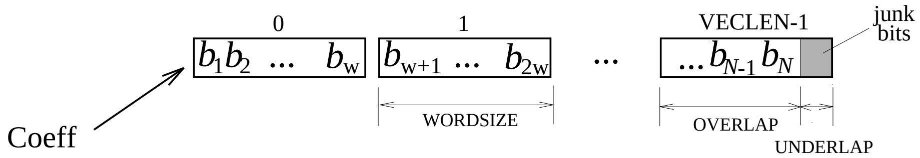
$$(\sum a_i \alpha_i)^p = \sum a_i \alpha_{i+1}, \qquad (\alpha_n = \alpha_0).$$

Če je $p = 2$, je kvadriranje cikličen zamik.

V praksi so takšne baze zelo praktične za hardwarsko implementacijo množenja v obsegu $\mathrm{GF}(p^n)$, še posebej, kadar je $p = 2$.

Element končnega obsega v predstavimo kot vektor. V hardwaru ponavadi delamo v GF(2), torej je v 01-vektor, ki ga hranimo v registru dolžine n, in je vsota vektorjev enaka XOR po koordinatah.

V softwaru pa hranimo binarni vektor v v besedah (npr. long integers)



V splošnem obstaja veliko število različnih baz za GF(q^m) nad GF(q).

Polinomska baza (PB) končnega obsega $\mathrm{GF}(q^n)$ nad $\mathrm{GF}(q)$ je oblike

$$\mathrm{GF}(q)[x]/(f), \qquad \text{kjer je } f \in \mathrm{GF}(q)[x]$$

nerazcepen polinom stopnje n,
in je vsak element v $\mathrm{GF}(q^n)$ predstavljen s polinomom v $\mathrm{GF}(q)[x]$ stopnje manjše od n.

Učinkovitost računanja množenja in kvadriranja v tej bazi lahko bistveno pospešimo, če si izberemo za nerazcepen polinom f trinom.

R. Schroeppel, C. Paar, E. de Win, M. Rossing, ...
Siemens, Certicom, Motorola, ...

DLP: Semaev, Menezes et al. [MOV], Frey, Smart, ...

Evklidov in Berlekampov algoritem

Naj za $a, b \in \mathbb{N}$ velja $0 \neq b \leq a$.
Za računanje največjega skupnega delitelja $D(a, b)$ poiščemo natanko določeni števili s in r za kateri velja

$$a = bs + r, \qquad 0 \leq r < b.$$

Če je ostanek r enak 0, potem je $D(a, b) = b$, sicer pa delimo zadnji delitelj z zadnjim ostankom in to ponavljamo.

Zadnji od nič različen ostanek je največji skupni delitelj števil a in b.

Ta proces lahko predstavimo z naslednjimi enačbami

$$\boxed{\begin{array}{rcl} a &=& s_1 b + r_1, \quad 0 < r_1 < b \\ b &=& s_2 r_1 + r_2, \quad 0 < r_2 < r_1 \\ r_1 &=& s_3 r_2 + r_3, \quad 0 < r_3 < r_2 \\ &\vdots& \\ r_{k-3} &=& s_{k-1} r_{k-2} + r_{k-1}, \quad 0 < r_{k-1} < r_{k-2} \\ r_{k-2} &=& s_k r_{k-1} + r_k, \quad 0 < r_k < r_{k-1} \\ r_{k-1} &=& s_{k+1} r_k + 0. \end{array}}$$

Potem je $D(a,b) = r_k$. Definirajmo še $r_{-1} := a$ in $r_0 := b$. Zaporedje $\{r_i\}$ imenujemo **Evklidovo zaporedje** števil a in b.

Oglejmo si postopek za iskanje posebne rešitve linearne diofantske enačbe $ax + by = 1$, ki poteka vzporedno z Evklidovim algoritmom.
Poiščimo zaporedji števil $\{p_i\}$ in $\{q_i\}$, za katera velja:

$$a\,p_i + b\,q_i \;=\; r_i,$$

kjer je $\{r_i\}$ Evklidovo zaporedje števil a in b, tj.

$$r_{-1} = a, \quad r_0 = b, \qquad \text{in} \qquad r_{i+1} = r_{i-1} - s_i\,r_i.$$

Zaradi začetnih pogojev si lahko izberemo kar

$$\boxed{p_{-1} = 1,\;\; q_{-1} = 0 \qquad \text{in} \qquad p_0 = 0,\;\; q_0 = 1.}$$

Zaradi rekurzivne zveze z množenjem in odštevanjem naslednjih enačb

$$\begin{array}{lclcl} a\,p_{i-1} & + & b\,q_{i-1} & = & r_{i-1}\ , \\ a\,p_i & + & b\,q_i & = & r_i\ , \\ a\,p_{i+1} & + & b\,q_{i+1} & = & r_{i+1}\ , \end{array}$$

dobimo

$$a\,(p_{i-1} - s_i\,p_i) + b\,(q_{i-1} - s_i\,q_i) \;=\; a\,p_{i+1} + b\,q_{i+1},$$

tako da si lahko izberemo še

$$\boxed{p_{i+1} = p_{i-1} - s_i\,p_i \quad \text{in} \quad q_{i+1} = q_{i-1} - s_i\,q_i, \qquad i \geq 0.}$$

in dobimo **razširjen Evklidov algoritem**.

Računanje multiplikativnega inverza z razširjenim Evklidovim algoritmom

Primer: $a(x) = x^4 + x + 1$ in $f(x) = x^5 + x^2 + 1$:

$$\begin{array}{rcl@{\qquad}rcl}
r_{-2} = x^5+x^2+1 & = & x(x^4+x+1)+x+1 & x \cdot 1+0 & = & x = p_0(x) \\
r_{-1} = x^4+x+1 & = & x^3(x+1)+x^3+x+1 & x^3 x+1 & = & x^4+1 \\
x^3+x+1 & = & x^2(x+1)+x^2+x+1 & x^2 x+x^4+1 & = & x^4+x^3+1 \\
x^2+x+1 & = & x(x+1)+1 & x \cdot x+x^4+x^3+1 & = & x^4+x^3+1 = p_1(x) \\
r_0 = x+1 & = & x \cdot 1+1 & x(x^4+x^3+x^2+1)+x & = & x^5+x^4+x^3 \\
r_1 = 1 & = & 1 \cdot 1+0 & 1(x^4+x^3+x^2+1)+x^5+x^2+x^3 & = & x^5+x^2+1 = p_2(x)
\end{array}$$

Ne računamo kvocijentov, pač pa množimo z x (tj. zamik) dokler polinoma nista enake stopnje.

Nato pa polinom, ki je imel večjo stopnjo, zamenjamo z razliko dobljenih polinomov.

Invertiranje z $a(x)^{-1} = a(x)^{|G|-1}$ je predrago.

ZAČETEK:	$\begin{pmatrix}1\,0\,0\,1\,0\,1,\ 1\\ 0\,1\,0\,0\,1\,1,\ 0\end{pmatrix}$	$= \begin{pmatrix}r_{-2},\ p_{-1}\\ r_{-1},\ p_{-2}\end{pmatrix} = \begin{pmatrix}f(x),1\\ a(x),0\end{pmatrix}$	PRVI
zamik :	$\begin{pmatrix}1\,0\,0\,1\,0\,1,\ 1\\ 1\,0\,0\,1\,1,\ 0\,0\end{pmatrix}$	$= \begin{pmatrix}r_{-2},\ xp_{-1}\\ xr_{-1},\ p_{-2}\end{pmatrix}$	
seštevanje:	$\begin{pmatrix}0\,0\,0\,0\,1\,1,\ 1\\ 1\,0\,0\,1\,1,\ 0\,1\end{pmatrix}$	$= \begin{pmatrix}r_{-2}+s_0r_{-1},\ xp_{-1}\\ xr_{-1},\ p_{-2}+s_0p_{-1}\end{pmatrix}$	
zamik :	$\begin{pmatrix}0\,0\,0\,1\,1,\ 1\,0\\ 1\,0\,0\,1\,1,\ 0\,1\end{pmatrix}$	$= \begin{pmatrix}r_0,\ p_{-1}\\ r_{-1},\ p_0\end{pmatrix}$	DRUGI
zamik $3\times$:	$\begin{pmatrix}1\,1,\ 1\,0\,0\,0\,0\\ 1\,0\,0\,1\,1,\ 0\,1\end{pmatrix}$	$= \begin{pmatrix}x^3r_0,\ p_{-1}\\ r_{-1},\ x^3p_0\end{pmatrix}$	
seštevanje:	$\begin{pmatrix}1\,1,\ 1\,0\,0\,0\,1\\ 0\,1\,0\,1\,1,\ 0\,1\end{pmatrix}$	$= \begin{pmatrix}x^3r_0,\ p_{-1}+x^3p_0\\ r_{-1}+x^3r_0,\ x^3p_0\end{pmatrix}$	
zamik :	$\begin{pmatrix}1\,1,\ 1\,0\,0\,0\,0\\ 1\,0\,1\,1,\ 0\,1\,0\end{pmatrix}$	$= \begin{pmatrix}x^2r_0,\ p_{-1}+x^3p_0\\ r_{-1}+x^3r_0,\ x^2p_0\end{pmatrix}$	
seštevanje:	$\begin{pmatrix}1\,1,\ 1\,0\,0\,1\,1\\ 0\,1\,1\,1,\ 0\,1\,0\end{pmatrix}$	$= \begin{pmatrix}x^2r_0,\ p_{-1}+(x^3+x^2)p_0\\ r_{-1}+(x^3+x^2)r_0,\ x^2p_0\end{pmatrix}$	
zamik :	$\begin{pmatrix}1\,1,\ 1\,0\,0\,1\,1\\ 1\,1\,1,\ 0\,1\,0\,0\end{pmatrix}$	$= \begin{pmatrix}xr_0,\ p_{-1}+(x^3+x^2)p_0\\ r_{-1}+(x^3+x^2)r_0,\ xp_0\end{pmatrix}$	
seštevanje:	$\begin{pmatrix}1\,1,\ 1\,0\,1\,1\,1\\ 0\,0\,1,\ 0\,1\,0\,0\end{pmatrix}$	$= \begin{pmatrix}xr_0,\ p_{-1}+(x^3+x^2+x)p_0\\ r_{-1}+(x^3+x^2+x)r_0,\ xp_0\end{pmatrix}$	
zamik :	$\begin{pmatrix}1\,1,\ 1\,0\,1\,1\,1\\ 0\,1,\ 0\,1\,0\,0\,0\end{pmatrix}$	$= \begin{pmatrix}r_0,\ p_1\\ r_1,\ p_0\end{pmatrix}$	TRETJI
zamik :	$\begin{pmatrix}1\,1,\ 1\,0\,1\,1\,1\\ 1,\ 0\,1\,0\,0\,0\,0\end{pmatrix}$	$= \begin{pmatrix}r_0,\ xp_1\\ xr_1,\ p_0\end{pmatrix}$	KONEC

SigGen in primerjava EC z RSA

Razvita je bila v **Certicom Corp., Kanada**, v sodelovanju s Schlumberger Smart Cards and Systems.

Uporablja Motorolin čip 68SC28:

- ROM 12.790 zlogov,
- EEPROM 8.112 zlogov,
- RAM 240 zlogov.

Vsebuje tehnologijo MULTIFLEX^{TM} ter tehnologijo eliptičnih krivulj $(CE)^2$, ki jo razvija podjetje Certicom Corp.

SigGen kartica je zelo prikladna za končnega uporabnika ter za proces prepoznavanja:

- je poceni,
- podpis je opravljen v pol sekunde,
- rabi samo 90 zlogov RAM-a,
- program ne zasede niti 4 KB.

Je edina pametna kartica, ki opravi digitalni podpis kar z obstoječim procesorjem.

Eliptični kripto-sistemi nudijo največjo moč glede na število bitov ključa med današnjimi javnimi kripto-sistemi.

Manjši ključi omogočajo

- manjše sistemske parametre,
- manjša potrdila z javnimi ključi,
- hitrejšo implementacijo,
- manjše zahteve po energiji,
- manjše procesorje,

itd.


Ali se morda približuje dan, ko bomo pod stalnim nadzorom od rojstva dalje?
Gen-0000E
412541954
251155-RA