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: $x_0, \dots, x_{15}$.

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

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

`State` dobi vrednosti iz x na naslednji način:

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

Na vsako besedo bomo gledali kot na dve šestnajstiški števili.

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

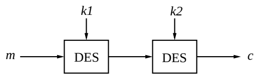
Druge simetrične šifre:

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

Dvojno šifriranje

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

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



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

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

Opomba: dolžina blokov se ni spremenila.

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, $1 \le s \le 55$. [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**).

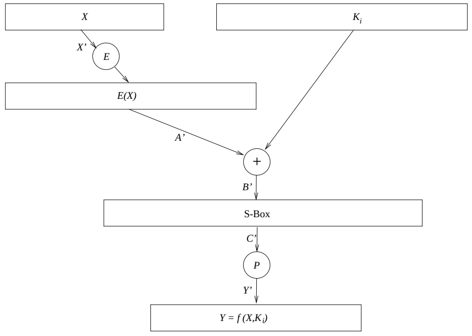
Oglejmo si pare tajnopisa za katere ima čistopis določene razlike. Diferenčna kriptoanaliza spremlja spreminjanje teh razlik, ko gre čistopis skozi nekaj ciklov DES-a in je šifriran z istim ključem.
Če poenostavimo, ta tehnika izbere pare čistopisa s fiksno razliko (čistopis je lahko izbran naključno).

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: $\Delta(B'_j) := \{(B_j, B_j \oplus B'_j) \mid B_j \in (\mathbb{Z}_2)^6\}$

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

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

Za vsak urejen par izračunamo razliko izhoda iz S_1:

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

Tabela izhodnih razlik C'_j in možnih vhodov B_j za vhodno razliko $B'_j = 110100$:

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

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

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

Pojavi se samo 8 od 16ih možnih izhodnih vrednosti.

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

Ta neenakomerna porazdelitev je osnova za diferenčni napad.

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

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

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

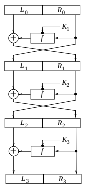
Potem poiščemo razliko $(C')_j$ v tabeli za S_j in $(A')_j$, ki nam določi vse možne vhode B_j iz katerih izračunamo vse $B_j \oplus A_j$, ki morajo vsebovati $(K_i)_j$.

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

Primer: $A_1 = 000001$, $A_1^* = 110101$ in $C'_1 = 1101$. Potem dobimo 13-to vrstico iz Tabele 1, ki vsebuje 8 elementov (torej smo zožili število možnosti iz $2^6 = 64$ na 8).

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

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

$$f(R_2, K_3) \oplus f(R_2^*, K_3) \quad (= L'_0 \oplus L'_3).$$

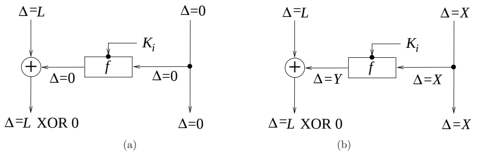
Naj bo $f(R_2, K_3) = P(C)$ in $f(R_2^*, K_3) = P(C^*)$, kjer sta C in C^* definirana enako kot prej (izhoda iz S škatel po tretjem ciklu). Potem je

$$C' = C \oplus C^* = P^{-1}(R'_3 \oplus L'_0).$$

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

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

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: $001000000000000001010000 \dots 0$.

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. poglavje

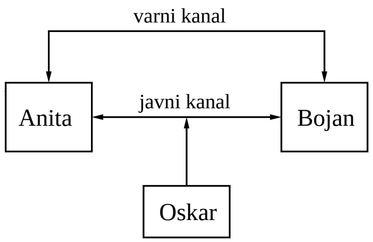
RSA sistem in faktorizacija

- 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

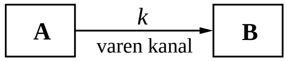
Sodelujoči si delijo *tajno* informacijo.



Dogovor o ključu

Kako Anita in Bojan vzpostavita tajni ključ k?

1. metoda: delitev point-to-point

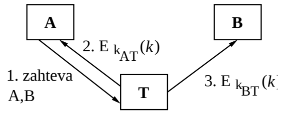


Varni kanal je lahko:
– kurir
– izmenjava na štiri oči (v temnem hodniku/ulici)

To ni praktično za večje aplikacije.

2. metoda: z neodvisnim centrom zaupanja T

– *Vsak* uporabnik A deli tajni ključ k_{AT} s centrom 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}.

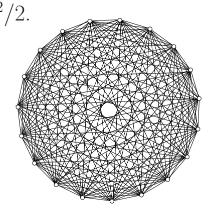
Problemi pri uporabi KDC

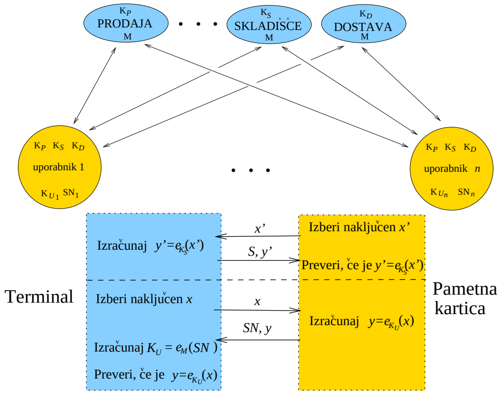
- centru zaupanja T moramo brezpogojno zaupati:
  – to ga naredi za očitno tarčo.
- Zahteva za stalno zvezo (on-line) s centrom T:
  – potencialno ozko grlo,
  – kritično za zanesljivost.

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 $\binom{n}{2} \approx n^2/2$.

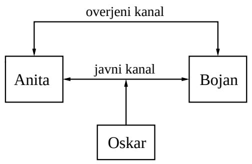
(Tudi preprečevanje tajenja je nepraktično.)





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).

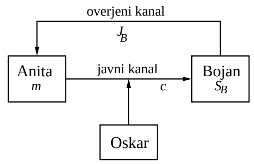
Generiranje para ključev

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



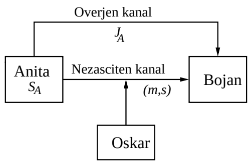
Da bi Bojanu poslala zaupno sporočil m, Anita:

- 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.

Teorija števil

Evklidov algoritem in reševanje Diofantske enačbe

$$ax + by = d, \qquad \text{kjer} \quad D(a, b) \mid d.$$

Evklidov algoritem je zasnovan na preprostem dejstvu, da iz $k \mid a$ in $k \mid b$ sledi $k \mid a - b$.

Če je $D(a, b) = 1$ in poznamo eno rešitev (x_0, y_0), tj.

$$ax_0 + by_0 = d,$$

potem ima poljubna rešitev (x, y) naslednjo obliko:

$$x = x_0 - kb, \quad y = y_0 + ka, \qquad \text{za } k \in \mathbb{Z}.$$

Zgodovina Evklidovega algoritma

Evklidov algoritem poišče največji skupni delitelj dveh naravnih števil in je zasnovan na dejstvu, da če število d deli števili a in b, potem deli tudi njuno razliko $a - b$.

V literaturi naletimo nanj prvič 300 p.n.š. v 7. knjigi Evklidovih **Elementov**.

Nakateri strokovnjaki so mnenja, da je njegov avtor Eudoxus (c. 375 p.n.š.). Gre za *najstarejši* netrivialen algoritem, ki je preživel do današnjih dni (glej Knuth).

Eno rešitev lahko poiščemo z **razširjenim Evklidovim algoritmom**.

Privzemimo, da je $a > b$ in zapišimo zgornjo enačbo malo bolj splošno (z zaporedji):

$$ap_i + bq_i = r_i.$$

Poiščimo dve trivialni rešitvi:

$$p_1 = 1, \quad q_1 = 0, \quad r_1 = a$$

in

$$p_2 = 0, \quad q_2 = 1, \quad r_2 = b.$$

Zaradi rekurzije

$$r_{i+1} = r_i - s_i r_{i-1}$$

(kjer je s_i izbran tako, da je $r_{i+1} < r_i$) si lahko izberemo še

$$p_{i+1} = p_i - s_i p_{i-1} \quad \text{in} \quad q_{i+1} = q_i - s_i q_{i-1}.$$

Ko računamo a^{-1} (po modulu praštevila p), računamo samo r_i ter p_i (ne pa tudi q_i).

Zgled za razširjeni algoritem:

$4864 = 1{\cdot}3458{+}1406$	$p_2 := p_1 - 1 \cdot p_0 = 1$
$3458 = 2{\cdot}1406{+}646$	$p_3 := p_2 - 2 \cdot p_1 = -2$
$1406 = 2{\cdot}646{+}114$	$p_4 := p_3 - 2 \cdot p_2 = 5$
$646 = 5{\cdot}114{+}76$	$p_5 := p_4 - 5 \cdot p_3 = -27$
$114 = 1{\cdot}76{+}38$	$p_6 := p_5 - 1 \cdot p_4 = 32$
$76 = 2{\cdot}38{+}0$	$p_7 := p_6 - 2 \cdot p_5 = -91$

$4864 = 1{\cdot}3458{+}1406$	$p_2 = 1 \quad q_2 = -1$
$3458 = 2{\cdot}1406{+}646$	$p_3 = -2 \quad q_3 = 3$
$1406 = 2{\cdot}646{+}114$	$p_4 = 5 \quad q_4 = -7$
$646 = 5{\cdot}114{+}76$	$p_5 = -27 \quad q_5 = 38$
$114 = 1{\cdot}76{+}38$	$p_6 = 32 \quad q_6 = -45$
$76 = 2{\cdot}38{+}0$	$p_7 = -91 \quad q_7 = 128$

$$4864 \cdot (-91) + 3458 \cdot (128) = 38$$

Čeprav uporabljamo ta algoritem že stoletja, pa je presenetljivo, da ni vedno najboljša metoda za iskanje največjega skupnega delitelja.

R. Silver in **J. Terzian** sta leta **1962** (v lit. J. Stein, *J. Comp. Phys.* **1** (1967), 397-405) predlagala **binarni algoritem**:

B1. *Poišči tak največji $k \in \mathbb{Z}$, da bosta števili a in b deljivi z 2^k; $a \leftarrow a/2^k$ in $b \leftarrow b/2^k$, $K \leftarrow 2^k$.*

B2. *Dokler $2|a$ ponavljaj $a \leftarrow a/2$ in dokler $2|b$ ponavljaj $b \leftarrow a/2$.*

B3. *Če je $a = b$, je $D(a, b) = a * K$, sicer pa v primeru $a > b$, priredi $a \leftarrow a - b$, sicer $b \leftarrow b - a$ in se vrni na korak B2.*

Lehmerjev algoritem deli z majhnimi namesto velikimi števili (izboljšave J. Sorenson, Jaebelan,...).

Dobro vprašanje je kako prenesti te ideje v GF(2^n).

R. Schroeppel je že naredil prvi korak s svojim algoritmom **almost inverse**.

Kitajski izrek o ostankih. *Če so števila $m_1, m_2, ..., m_r$ paroma tuja, tj. $D(m_i, m_j) = 1$ za $i \neq j$, in $a_1, a_2, ..., a_r \in \mathbb{Z}$, potem ima sistem kongruenc*

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

enolično rešitev po modulu $M = m_1 \cdot m_2 \cdots m_r$,

$$x = \sum_{i=1}^{r} a_i \cdot M_i \cdot y_i \pmod{M},$$

kjer je $M_i = M/m_i$, $y_i = M_i^{-1} \bmod m_i$, $i = 1, ..., r$.

(angl. Chinese Reminder Theorem oziroma CRT)

Red elementa g v končni multiplikativni grupi je najmanjše celo število m tako, da $g^m = 1$.

Lagrangev izrek: *Naj bo G multiplikativna grupa reda n in $g \in G$, potem red g deli n.*

Naj bo p praštevilo. Generatorju multiplikativne grupe $\mathbb{Z}_p^*$ pravimo **primitiven element**.

DN: Koliko primitivnih elementov ima $\mathbb{Z}_p$?
Naj bo α primitiven element, potem za $\forall \beta \in \mathbb{Z}_p^*$ obstaja tak $i \in \{0, 1, \ldots, p-2\}$, da je $\beta = \alpha^i$.
Pokaži, da je red elementa β enak $\dfrac{p-1}{D(p-1, i)}$.

Eulerjevo funkcijo φ definiramo s

$$\varphi(n) = |\{x \in \mathbb{N} \,|\, x < n \text{ in } D(x, n) = 1\}|.$$

Potem za praštevilo p, naravno število n in poljubni tuji si števili a in b velja

$$\varphi(p^n) = p^n - p^{n-1} \quad \text{in} \quad \varphi(ab) = \varphi(a)\varphi(b).$$

Če poznamo faktorizacijo števila n, poznamo tudi $\varphi(n)$.

Fermatov izrek
Za praštevilo p in $b \in \mathbb{Z}_p$ velja $\quad b^p \equiv b \pmod{p}$.

Eulerjev izrek
Če je $a \in \mathbb{Z}_n^*$ oziroma $D(n, a) = 1$, potem velja

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$