1.

Kryptografia

Kryptografia je štúdium matematických techník na ochra nu a utajenie informácie.
Niekedy sa používa ja termín Kryptológia, ktorá sa delí na

- Kryptografiu – vynachádzanie šifrovacích systémov a
- Kryptoanalýzu – študujúcu útoky voči šifrovacím systémom.

Úlohy kryptografie

- Utajenie informácie
- Zaistenie integrity údajov – zaistenie proti zmene správy
- Autentifikácia – zaistenie, že správa pochádza od určitého pôvodcu
- Identifikácia – zaistenie, že komunikujem stým s kým chcem
- Neodškriepiteľný digitálny podpis
- Steganografia – ukrytie správy v inom údajovom súbore

Kryptografické útoky.

Útok na kryptografický systém je postup, ktorý odhalí priamehy text (alebo aspoň jeho časť) alebo doknca zistí šifrovací kľúč.

Typy kryptografických útokov

- Brute force attack
- Ciphertext only attack
- Known plaintext attack
- Chosen plaintext attack
- Chosen ciphertext attack
- Dictionary attack
- Rubber hose attack

Kryptosystém je usporiadaná štvorica $(\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{T})$ kde

- $\mathcal{K}$ je množina kľúčov
- $\mathcal{M}$ je množina priamych textov
- $\mathcal{C}$ je množina zašifrovaných textov
- $\mathcal{T}$ je zobrazenie $\mathcal{T} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$, ktoré každej dvojici $K \in \mathcal{K}$, $M \in \mathcal{M}$ priradí zašifrovanú správu $C \in \mathcal{C}$ a také, že

V tomto systéme $\mathcal{K} = \{A, B, \ldots, Z\}$, kľúč $k = A \equiv 0$ je nepoužiteľný.
$\mathcal{M}$ je množina všetkých zrozumiteľných slovenských textov.

2. Afinna sifra:

Základem afinní šifry je následující transformace:

$$C_i = a \cdot T_i + b \pmod{m}$$

```
Ci - i-té písmeno šifrovaného textu
Ti - i-té písmeno otevřeného textu
a - parametr a, gcd(a,m) = 1
b - parametr b
m - modulo (jako modulo obvykle volíme prvočíslo, aby bylo předem jasné,
že gcd(a, m) = 1, a zároveň abychom útočníkovi nezjednodušovali práci
(pokud modulo není prvočíslo, tak je méně možností, jak se text dá
šifrovat - je tedy snazší šifru prolomit)).
```

Zašifrujte daný otevřený text pomocí afinní transformace.

```
Otevřený text: THEINITIAL
a = 5
b = 9
m = 26
```

```
T ⇒ 19 ⇒ 5*19 + 9 mod(26) = 0  ⇒ A
H ⇒ 7  ⇒ 5*7  + 9 mod(26) = 18 ⇒ S
E ⇒ 4  ⇒ 5*4  + 9 mod(26) = 3  ⇒ D
I ⇒ 8  ⇒ 5*8  + 9 mod(26) = 23 ⇒ X
N ⇒ 13 ⇒ 5*13 + 9 mod(26) = 22 ⇒ W
I ⇒ 8  ⇒ 5*8  + 9 mod(26) = 23 ⇒ X
T ⇒ 19 ⇒ 5*19 + 9 mod(26) = 0  ⇒ A
I ⇒ 8  ⇒ 5*8  + 9 mod(26) = 23 ⇒ X
A ⇒ 0  ⇒ 5*0  + 9 mod(26) = 9  ⇒ J
L ⇒ 11 ⇒ 5*11 + 9 mod(26) = 12 ⇒ M
```

Dešifrování

Při dešifrování musíme eliminovat transformaci vzniklou šifrováním. Provedeme proto takovouto transformaci:

$$T_i = (C_i - b) \cdot a^{-1} \pmod{m}$$

a^{-1} - multiplikativní inverze a v Z_m

Příklad

Dešifrujte šifrovaný text pomocí afinní transfromace.

```
Šifrovaný text: ASDXWXAXJM
a = 5
b = 9
m = 26
```

Rozšířeným Euklidovým algoritmem zjistíme:

$$a^{-1} = 21$$

```
A ⇒ 0  ⇒ (0  - 9) * 21 mod(26) = 19 ⇒ T
S ⇒ 18 ⇒ (18 - 9) * 21 mod(26) = 7  ⇒ H
D ⇒ 3  ⇒ (3  - 9) * 21 mod(26) = 4  ⇒ E
X ⇒ 23 ⇒ (23 - 9) * 21 mod(26) = 8  ⇒ I
W ⇒ 22 ⇒ (22 - 9) * 21 mod(26) = 13 ⇒ N
X ⇒ 23 ⇒ (23 - 9) * 21 mod(26) = 8  ⇒ I
A ⇒ 0  ⇒ (0  - 9) * 21 mod(26) = 19 ⇒ T
X ⇒ 23 ⇒ (23 - 9) * 21 mod(26) = 8  ⇒ I
J ⇒ 9  ⇒ (23 - 9) * 21 mod(26) = 0  ⇒ A
M ⇒ 12 ⇒ (12 - 9) * 21 mod(26) = 11 ⇒ L
```

Útoky na afinní šifru:

Protože je zde již daleko více kombinací, tak by byl útok hrubou silou poměrně neefektivní (nic méně pořád proveditelný). Skutečnou slabinou afinní šifry je frekvenční analýza.

Kľúč – dvojica prvkov k_1, k_2 okruhu $\mathbb{Z}_{26}$ taká,
že existuje prvok $k_1^{-1} \in \mathbb{Z}$ inverzný ku k_1 (t.j. $k_1 \otimes k_1^{-1} = 1 \equiv B$).

$$\begin{aligned} \text{šifrovanie:} \quad y &= E_{k_1,k_2}(x) = (x \otimes k_1) \oplus k_2 \\ \text{dešifrovanie:} \quad x &= D_{k_1,k_2}(y) = (y \ominus k_2) \otimes k_1^{-1} \end{aligned}$$

Množina kľúčov $\mathcal{M}$ – množina všetkých usporiadaných dvojíc (k_1, k_2) taká, že existuje $k_1^{-1} \in \mathbb{Z}$.

$k_1 = 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25$ – 12 možností

$k_2 = 0, 1, 2, \ldots, 24, 25$ –26 možností

Slabý kľúč $(k_1, k_2) = (1, 0)$.

Množina použiteľných kľúčov obsahuje 12.26-1 = 311 prvkov.

Kryptoanalýza afinnej šifry

„Ciphertext only attack" hrubou silou vyžaduje vyskúšať 311 kľúčov.
Known plaintext attack:
Uhádneme že $E_{k_1,k_2}(C) = P$, $E_{k_1,k_2}(F) = H$,
t.j. $E_{k_1,k_2}(2) = 15$, $E_{k_1,k_2}(5) = 7$

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	15	18	19	20	21	22	23	24	25

$$\begin{aligned} k_1 \otimes 2 \oplus k_2 &= 15 && (2) \\ k_1 \otimes 5 \oplus k_2 &= 7 && (3) \end{aligned}$$

Odčítaním rovnice (2) od (3)

$$\begin{aligned} k_1 \otimes 3 &= -8 \mod 26 = 18 && / * 9 \equiv 3^{-1} \quad (4) \\ k_1 &= 18 * 9 \mod 26 = 162 \mod 26 = 6 && (5) \end{aligned}$$

Dosadením za k_1 do (2)

$$\begin{aligned} (6 \otimes 2) \oplus k_2 &= 15 && (6) \\ k_2 &= 15 \ominus 12 = 3 && (7) \end{aligned}$$

3. Vseobecna monoalfabeticka sifra

Všeobecná monoalfabetická šifra

π – ľubovoľná permutácia abecedy $\mathbb{Z}_{26}$

π^{-1} – inverzná permutácia k permutácii π

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

Šifrujeme znak po znaku predpisom $y = E_\pi(x) = \pi(x)$

Dešifrujeme znak po znaku predpisom $x = D_\pi(y) = \pi^{-1}(y)$

Priestor kľúčov $\mathcal{K}$ je obrovský $|\mathcal{K}| = 26! \approx 10^{27}$

Pokus o matematickú formuláciu problému kryptoanalýzy

- p_{ij} pravdepodobnosť výskytu dvojice znakov $a_i a_j$ v jazyku.
- r_{pq} relatívna početnosť znakov $a_p a_q$ v zašifrovanom texte
- $x_{ip} = \begin{cases} 1 & \text{ak } a_i \text{ bolo zašifrované na } a_p \\ 0 & \text{inak} \end{cases}$

Minimalizovať
$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{n} \sum_{=1}^{n} x_{ip} x_{jq} (p_{ij} - r_{pq})^2$$

za podmienok
$$\begin{aligned} \sum_{i=1}^{n} x_{ip} &= 1 \quad \text{pre } p = 1, 2, \ldots, n \\ \sum_{p=1}^{n} x_{ip} &= 1 \quad \text{pre } i = 1, 2, \ldots, n \\ x_{ip} &\in \{0, 1\} \end{aligned}$$

Najčastejšie znaky slovenskej abecedy sú medzera a

A, O, E, I, N, T, S

Postup pri kryptoanalýze (Grošek, Porubský):

- Ak bola použitá taká permutácia, ktorý zachováva medzeru, treba analyzovať najskôr kratšie slová, ktoré poskytujú menší priestor pre kombinácie
- Hľadať charakteristické kombinácie znakov (trojice, štvorice). Tie sa najčastejšie vyskytujú na začiatkoch a na koncoch slov.
- Odhadnúť na základe „postranných informácií", ktoré slová by sa mohli v texte vyskytnúť
- Odhadnúť, ktoré znaky sú samohlásky a ktoré spoluhlásky

Vytipovanie samohlások takto:

- samohlásky sú často obkolesené spoluhláskami
- spoluhlásky sú často obkolesené samohláskami
- písmená s malým počtom rôznych susedov sú často spoluhlásky a títo susedia sú často samohlásky
- ak sa dvojica XY vyskytuje často aj v opačnom poradí YX jedno z nich je samohláska
- skoro v každom normálnom slove je samohláska

4. Vigenerovska

Polyalfabetické šifry.

Nevýhoda monoalfabetických šifier – relatívna početnosť zašifrovaného písmena v zašifrovanom texte závisí na pravdepodobnosti výskytu tohoto písmena v použitom jazyku.

Nová myšlienka – síce šifrovať znak po znaku, ale každý znak priameho textu inak.

Teda zašifrovaný text $y_1 y_1 \ldots y_n$ dostaneme z priameho textu $x_1 x_1 \ldots x_n$ takto:

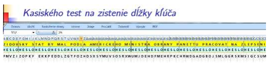
$$\begin{aligned} y_1 &= E_{K_1}(x_1) \\ y_2 &= E_{K_2}(x_2) \\ &\ldots \\ y_n &= E_{K_n}(x_n) \end{aligned}$$

Vigenèrovské šifry

Najjednoduchší spôsob je nasledovný: zvolí sa kľúč – napr. „HESLO" a potom zašifrovaný text $y_1 y_2 \ldots y_n$ dostaneme z priameho textu $x_1 x_2 \ldots x_n$ takto:

$$\begin{aligned} y_1 &= x_1 \oplus H \\ y_2 &= x_2 \oplus E \\ y_3 &= x_3 \oplus S \\ y_4 &= x_4 \oplus L \\ y_5 &= x_5 \oplus O \\ y_6 &= x_6 \oplus H \\ y_7 &= x_7 \oplus E \\ y_8 &= x_8 \oplus S \\ y_9 &= x_9 \oplus L \end{aligned}$$

Kasiského test na zistenie dĺžky kľúča



Hľadajú sa opakované výskyty toho istého reťazca v zašifrovanom texte.
Je nádej, že vzdialenosť výskytov je násobkom dĺžky kľúča.

Dĺžka kľúča je pravdepodobne najväčším spoločným deliteľom vzdialeností rovnakých výskytov

5. Index koincidencie

Index koincidencie

Ak by všetky znaky abecedy $A = \{a_1, a_2, \ldots, a_q\}$ s q znakmi mali rovnakú pravdepodobnosť, potom $p(a_i) = \frac{1}{q}$.
Hľadáme spôsob, ako kvantifikovať mieru nerovnomernosti pravdepodobností.

$$\sum_{i=1}^{q} (p(a_i) - \frac{1}{q})^2$$

Index koincidencie (2)

Definícia

Číslo $\sum_{i=1}^{q} p(a_i)^2$ *sa nazýva* **index koincidencie**.

Čím je index koincidenci väčší než $\frac{1}{q}$, tým viac sa rozdelenie pravdepodobnosti viac lľši od rovnomerného rozdelenia.

Pre slovenskú telegrafnú abecedu bez medzery je index koincidencie asi 0,06027, pričom $\frac{1}{q} = 0,03846$.
Pre slovenskú abecedu s diakritikou, číslami a interpunkčnými znakmi v kódovaní používanom v počítačoch sme odhadli index koincidencie na 0,0553.

Ďalší význam indexu koincidencie:

Pravdepodobnosť, že dva náhodne vybrané znaky z jazyka (resp. zo zdroja informácie) sa budú oba rovnať a_i je $p(a_i)$.
Jav, že dva náhodne vybrané znaky budú rovnaké je zjednotením nasledújúcicj disjujnktných javov

- že oba znaky sa budú rovnať a_1 – pravdepodobnosť $p(a_1)^2$
- že oba znaky sa budú rovnať a_2 – pravdepodobnosť $p(a_2)^2$
- ....
- že oba znaky sa budú rovnať a_q – pravdepodobnosť $p(a_q)^2$

Pravdepodobnosť javu, že dva náhodne vybrané znaky budú rovnaké, je súčet pravdepodobností týchto javov, a teda $\sum_{i=1}^{q} p(a_i)^2$

Zisťovanie dĺžky kľúča metódou koincidencie

Majme dva priame texty
$\mathbf{r} = r_1 r_2 \ldots r_n$, $\mathbf{s} = s_1 s_2 \ldots s_n$
Pravdepodpobnosť, že $r_i = s_i$ je index koincidencie slov. jazyka κ.
Nech teto texty sú zašifrované znak po znaku rovnakým kľúčom.
Príslušné zašifrované texty sú"
$\bar{\mathbf{r}} = T_1(r_1) T_2(r_2) \ldots T_n(r_n)$,
$\bar{\mathbf{s}} = T_1(s_1) T_2(s_2) \ldots T_n(s_n)$.
Pravdepodobnosť javu , že $T_i(r_i) = T_i(s_i)$, je rovnaká ako pravdepodpobnosť javu, že $r_i = s_i$, lebo $T_i(r_i) = T_i(s_i)$ práve vtedy, keď $r_i = s_i$. Teda

$$P(T_i(r_i) = T_i(s_i)) = P(r_i = s_i) = \kappa$$

Ak sledujeme počet zhôd na rovnakých miestach zašifrovaného a posunutého zašifrovaného textu, počet zhôd by mal nápadne stúpnuť, ak je posun o násobok dĺžky kľúča.

6.Hillovska sifra

Majme priamy text v q-znakovej abecede $A = \{a_0, a_1 \ldots, a_{q-1}\}$.
Prvky abecedy A stotožníme s prvkami okruhu $\mathbb{Z}_q$.
Na abecede A tak máme operácie $\oplus$ a $\otimes$.
Ak je q prvočíslo, je $\mathbb{Z}_q$ poľom a ku každému $a \in A$ $a \neq 0$ existuje $a^{-1} \in A$ také, že $a \otimes a^{-1} = 1$.
Ak q nie je prvočíslo, potom inverzné prvky existujú len k tým znakom, ktoré nie sú súdeliteľné s q.
Preferujeme teda q prvočíslo.
Existujú konečné telesá s $q = p^n$ prvkami, kde p je prvočíslo.
Sú to tzv. Galoisove polia, značia sa $GF(p^n)$.
Na abecedách, ktoré nemajú prvočíselný počet prvkov alebo počet prvkov rovnajúci sa prirodzenej mocnine prvočísla, nemožno zaviesť operácie $\oplus$ a $\otimes$ tak, aby štruktúra $(A, \oplus, \otimes)$ bola poľom.

Hillovská šifra je polyalfabetická šifra šifrujúca naraz celý blok priameho textu dĺžky n .

$$\underbrace{x_{11} x_{12} \cdots x_{1n}}_{\mathbf{x}_1} \underbrace{x_{21} x_{22} \cdots x_{2n}}_{\mathbf{x}_2} \cdots\cdots\cdots \underbrace{x_{m1} x_{m2} \cdots x_{mn}}_{\mathbf{x}_m} \qquad (1)$$

Kľúčom je štvorcová matica typu $n \times n$ taká že k nej existuje inverzná matica $\mathbf{K}^{-1}$.

$$\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} & \ldots & k_{1n} \\ k_{21} & k_{22} & \ldots & k_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ k_{n1} & k_{n2} & \ldots & k_{nn} \end{pmatrix} \qquad (2)$$

Šifrovanie

$$\mathbf{y} = \mathbf{K}\mathbf{x} \quad \begin{pmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & \ldots & k_{1n} \\ k_{21} & k_{22} & \ldots & k_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ k_{n1} & k_{n2} & \ldots & k_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{pmatrix} \qquad (3)$$

$$\begin{aligned} y_1 &= k_{11} x_1 + k_{12} x_2 + \cdots + k_{1n} x_n \\ y_2 &= k_{21} x_1 + k_{22} x_2 + \cdots + k_{2n} x_n \\ \ldots \\ y_n &= k_{n1} x_1 + k_{n2} x_2 + \cdots + k_{nn} x_n \end{aligned}$$

Dešifrovanie

$$\mathbf{x} = \mathbf{K}^{-1} \mathbf{y}$$

Dešifrovanie je korektné, lebo

$$\mathbf{K}^{-1} \mathbf{y} = \mathbf{K}^{-1}.(\mathbf{K}.\mathbf{x}) = (\mathbf{K}^{-1}.\mathbf{K}).\mathbf{x} = \mathbf{I}.\mathbf{x} = \mathbf{x} \qquad (4)$$

Príklad: Abeceda
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$\} \equiv \mathbb{Z}_{26}$.
VSTUPNA MATICA

$$\mathbf{K} = \begin{pmatrix} 17 & 4 & 3 & 9 \\ 1 & 13 & 21 & 16 \\ 10 & 12 & 5 & 9 \\ 13 & 6 & 3 & 12 \end{pmatrix}$$

Či je regulárna, zistíme tak, že v niektorom tabuľkovom procesore vypočítame jej determinant. Tu je $\det \mathbf{K} = -11305$ a $\mod(-11305, 26) = 5$ je číslo, ku ktorému existuje v $\mathbb{Z}_{26}$ inverzný prvok – totiž 21.

Výpočet inverznej matice. Ekvivalentnými úpravami matíc upravime maticu $(\mathbf{K}|\mathbf{I})$, kde $\mathbf{I}$ je jednotková štvorcová matica, na tvar $(\mathbf{I}|\mathbf{K}^{-1})$.

$$\left(\begin{array}{cccc|cccc} 17 & 4 & 3 & 9 & 1 & 0 & 0 & 0 \\ 1 & 13 & 21 & 16 & 0 & 1 & 0 & 0 \\ 10 & 12 & 5 & 9 & 0 & 0 & 1 & 0 \\ 13 & 6 & 3 & 12 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$\left(\begin{array}{cccc|cccc} 17 & 4 & 3 & 9 & 1 & 0 & 0 & 0 \\ 0 & 25 & 4 & 17 & 3 & 1 & 0 & 0 \\ 0 & 2 & 17 & 19 & 4 & 0 & 1 & 0 \\ 0 & 6 & 16 & 25 & 13 & 0 & 0 & 1 \end{array}\right)$$

Known plaintext attack proti hillovskej šifre. Predpokladajme, že poznáme n dvojíc priameho textu a príslušného ciphertextu.

$$\mathbf{y}_1 = \mathbf{K}\mathbf{x}_1, \mathbf{y}_2 = \mathbf{K}\mathbf{x}_2, \ldots, \mathbf{y}_n = \mathbf{K}\mathbf{x}_n \qquad (5)$$

Zostrojme štvorcové matice typu $n \times n$ $\mathbf{X}$, $\mathbf{Y}$, ktorých stĺpce budú tvorené stĺpcovými vektormi $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, resp. $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$, t.j.:

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \qquad \mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n).$$

Potom vzťahy (5) možno zapísať v maticovom tvare

$$\mathbf{Y} = \mathbf{K}.\mathbf{X} \qquad (6)$$

Vynásobením rovnice (6) maticou $\mathbf{X}^{-1}$ sprava (za predpokladu, že $\mathbf{X}^{-1}$ existuje) dostávame:

$$\mathbf{Y}.\mathbf{X}^{-1} = (\mathbf{K}.\mathbf{X}).\mathbf{X}^{-1} = \mathbf{K}.(\mathbf{X}.\mathbf{X}^{-1}) = \mathbf{K}.\mathbf{I} = \mathbf{K}$$

7. Transpozicna – permutacna sifra

Principem transpoziční šifry je změna pořadí jednotlivých znaků textu (permutace) na základě předem domluveného systému. Výhodou tohoto postupu je jeho jednoduchost – může ho použít klidně dítě, protože není obvykle třeba jakákoliv znalost matematiky. Nevýhodou je jeho více či méně snadná analýza (dle pravidla transformace), další významnou nevýhodou je snadné odhalení jazyka otevřeného textu pomocí frekvenční analýzy (znaky zůstavají totožné, mění se jen jejich pořadí).

Použitím transpoziční šifry dochází k difůzi – rozprostření redundance jazyka napříč zprávou.

Příklad

OT: SKAKALPESPRESOVESPRESZELENOULOUKU
Pravidlo: Zapišme text do tabulky s pěti sloupci (padding: „X"), tabulku překlopme podle diagonály a čtěme po řádcích.

ŠT: SLRESNUKPESZOKAESPEUUSKORLLXAPVEEOX

```
SKAKA
LPESP              SLRESNU
RESOV              KPESZOK
ESPRE      →       AESPEUU
SZELE              KSORLLX
NOULO              APVEEOX
UKUXX
```

Transpoziční šifra

Dešifrování

Protože měl obdélník 5 sloupců, tak má po transformaci 5 řádek. Zapišme tedy došlý ŠT do obdélníku s pěti řádky – máme 35 znaků, tzn. 7 sloupců. Tímto jsme dosáhli toho, že máme před sebou obdélník, který měl šifrující na pravé straně od šipky (vizte obrázek). Buď ho ještě otočíme nebo rovnou čteme po sloupcích.

Útok pomocí vycpávky

Nejefektivnější je zaútočit na tento typ kryptosystému pomocí paddingu (vycpávky). Protože víme, že byly vloženy znaky X na konec textu (aby se text zarovnal na plný obdélník) a z ŠT víme, že po transformaci je padding od sebe vzdálen 7 pozic a má celkem 35 znaků, tak je zřejmé, že měl původní obdélník původně 7 řádek a 5 sloupců – můžeme dešifrovat.

Útok faktorizací

Protože se každé číslo dá rozložit pouze na omezený počet dělitelů a známe počet znaků ŠT, tak také můžeme snadno prozkoušet všechny kombinace počtů řádků a sloupců.

8. One time Pad + utok

Vernamova šifra nebo také **jednorázová tabulková šifra** (anglicky *one-time pad*) je jednoduchý šifrovací postup patentovaný v roce 1917 Gilbertem Vernamem. Spočívá v posunu každého znaku zprávy o náhodně zvolený počet míst v abecedě. To se prakticky rovná náhradě zcela náhodným písmenem a na tomto faktu je založen důkaz, že Vernamova šifra **je v principu nerozluštitelná**.

Postup šifrování [editovat | editovat zdroj]

Vezmeme jednotlivá písmena tajné zprávy a každé z nich posuneme o několik pozic v abecedě. Například první písmeno je posunuto o 5 pozic, druhé o 1, třetí o 14, čtvrté o 24, další o 9, 0, 3, 9, 19. Když při posouvání překročíme konec abecedy, pokračujeme od jejího začátku. Ze slova ALDEBARAN tak dostaneme šifrový text FMRCKAUJG. Posloupnost 5, 1, 14, 24, 9, 0, 3, 9, 19 je klíčem k rozluštění zprávy. Kdo ji zná, dokáže snadno posunout písmena opačným směrem a získat původní text. Bez znalosti klíče je luštění odposlechnuté zprávy nemožné.

Text zašifrován

Podmínky spolehlivosti [editovat | editovat zdroj]

Porušení kterékoli z následujících podmínek má za následek oslabení šifry, takže by již nebyla nerozluštitelná.

1. **Klíč je tak dlouhý jako přenášená zpráva.** Jiné šifrovací systémy používají kratší klíče, což znamená, že počet možných klíčů je menší než počet možných zpráv. Kratší klíč v principu umožňuje útok hrubou silou.
2. **Klíč je dokonale náhodný.** Nepřipadají v úvahu počítačové generátory pseudonáhodných čísel, neboť jejich činnost lze předvídat. Nejvhodnější je užití fyzikálních metod - hardwarový generátor náhodných čísel.
3. **Klíč nelze použít opakovaně.** Tato podmínka je vlastně důsledkem předchozí, protože opakovaný klíč není náhodný. Dostane-li útočník do ruky dvě zprávy zašifrované týmž klíčem, má často velmi snadnou cestu k rozluštění.

Vernamova šifra je symetrická proudová šifra spočívající v binární operací *XOR* nad otevřeným textem a předem smluveným náhodným klíčem (šumem).

Klíč se za žádných okolností nesmí recykovat - na každou komunikaci se použije vždy nový - protože jinak by útočník mohl *XOR*ovat obě **zašifrované** zprávy, čímž by získal *XOR* obou **nezašifrovaných** zpráv, z čehož lze statistickými metodami získat otevřený text obou zpráv. Pokud se dodrží tato zásada a klíč je vygenerován skutečně náhodným způsobem (nikoliv pseudonáhodným), pak šifru nelze prolomit.

One time pad - Vernamova šifra

Abeceda $A = \mathbb{Z}_2$.

Množina kľúčov i zašifrovaných textov je $\mathbb{Z}_2$.

$x_1, x_2, \ldots, x_n, \ldots$ – prúd znakov priameho textu

$k_1, k_2, \ldots, k_n, \ldots$ – prúd kľúčov, $P(k_i = 0) = P(k_i = 1) = \frac{1}{2}$

$y_1, y_2, \ldots, y_n, \ldots$ – prúd znakov zašifrovaného textu

$$\begin{array}{cccccc} x_1, & x_2, & x_3, & \ldots, & x_i, & \ldots \\ k_1, & k_2, & k_3, & \ldots, & k_i, & \ldots \\ y_1 = x_1 \otimes k_1, & y_2 = x_2 \otimes k_2, & y_3 = x_3 \otimes k_3, & \ldots, & y_i = x_i \otimes k_i, & \ldots \end{array}$$

Ak sú kľúče $k_1, k_2, \ldots, k_n, \ldots$ vyberané náhodne s rovnomerným rozdelením pravdepodobnosti, niet šance na prelomenie Vernamovej šifry.
Nevýhody:

- Kľúč musí byť aspoň tak dlhý, ako je správa
- Kľúč sa nesmie použiť viac, ako raz

Získavanie náhodných postupností

- z výstupu Gieiger-Mullerovho počítača
- meranie nepravidelností silne zamestnaného servera
- meranie teplotných fluktuácií

Výsledky takýchto meraní budú síce náhodné, ale pravdepodobnosti núl a jedničiek nemusia byť rovnaké.
Jeden spôsob vyrovnávania prevdepodobností je tento:

$$\underbrace{00}_{-} | \underbrace{00}_{-} | \underbrace{10}_{1} | \underbrace{11}_{-} | \underbrace{01}_{0} | \underbrace{01}_{0} | \underbrace{00}_{-} | \underbrace{11}_{-} | \underbrace{10}_{1} | \underbrace{00}_{-} | \underbrace{10}_{1} |$$

Generátor RC4

Máme 256 S-boxov $S[0], S[1], \ldots, S[255]$, ktoré obsahujú niektorú permutáciu čísel 0 až 255.

Kľúč môže byť až 256*8=2048 bitov. Týmito bitmi sa naplnia postupne 8-bitové čísla $K[0], K[1], \ldots, K[255]$.

Útok pri viacnásobnom použití toho istého prúdu kľúčov

Predpokladajme, že dve postupnosti znakov priameho textu

$$a_1, a_2, \dots, a_n, \dots, \quad b_1, b_2, \dots, b_n, \dots$$

boli zašifrované tý istým prúdom kľúčov $k_1, k_2, \dots, k_n, \dots$.

Kryptoanalytik dostane dva zašifrované texty $y_1, y_2, \dots, y_n, \dots$, $z_1, z_2, \dots, z_n, \dots$ také, že

$$y_i = a_i \otimes k_i, \quad z_i = b_i \otimes k_i.$$

Kryptoanalytik si vypočíta postupnosť $w_1, w_2, \dots, w_n, \dots$, kde $w_i = y_i \otimes z_i$.
Platí:

$$w_i = y_i \otimes z_i = (a_i \otimes k_i) \otimes (b_i \otimes k_i) = (a_i \otimes b_i) \otimes (k_i \otimes k_i) = (a_i \otimes b_i) \otimes 0 = (a_i \otimes b_i)$$

Postupnosť $w_1, w_2, \dots$ je postupnosť znakov jedného priameho textu zašifrovaná postupnosťou iného priameho textu a takáto postupnosť nesie dostatok informácie na odhalenie podstanej časti oboch priamych textov a v konečnom dôsledku aj postunosti bitov kľúča.

Synchronizácia zašifrovaných textov

Kryptoanalytik zachytí dve postupnosti zašifrovaných textov

$$y_1, y_2, \dots, y_n, \dots, \quad z_1, z_2, \dots, z_n, \dots,$$

ktoré boli zašifrované tým istým prúdom kľúčov, avšak sú navzájom posunúté o d pozícií, t.j.

$$y_i = a_i \otimes k_{i+d}, \quad z_i = b_i \otimes k_i.$$

Ak vytvorí postupnosť

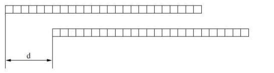
$$w_i = y_i \otimes z_i = (a_i \otimes k_{i+d}) \otimes (b_i \otimes k_i) = (a_i \otimes b_i) \otimes (k_{i+d} \otimes k_i),$$

táto sa bude javiť ako postupnosť náhodných bitov.
Ak však posunie zašifrovaný text $z_1, z_2, \dots, z_n, \dots$, oproti textu $y_1, y_2, \dots, y_n, \dots$ od d pozícií dozadu, a vytvorí postupnosť

$$w_i = y_i \otimes z_{i+d} = (a_i \otimes k_{i+d}) \otimes (b_{i+d} \otimes k_{i+d}) = (a_i \otimes b_{i+d}) \otimes (k_{i+d} \otimes k_{i+d}) = a_i \otimes b_{i+d},$$

počet núl v tejto postupnosti nápadne stúpne, lebo pravdepodobnosť nuly je pravdepodobnosťou, že $a_i = b_{i+d}$, čo sa rovná príslušnému indexu koincidencie.

Posúvame proti sebe oba zašifrované texty. Pri zasynchronizovaní – nájdení správnej vzdialenosti d počet zhôd nápadne stúpne.



9.Pseudogeneratory (linearny kongruencny, kvadraticky, kubicky, RC4)

Použitie generátorov náhodných čísel

Lineárny kongruenčný generátor

$$X_n = (aX_{n-1} + b) \mod m$$

Perióda max $m - 1$.

Kvadratický kongruenčný generátor

$$X_n = (aX_{n-1}^2 + bX_{n-1} + c) \mod m$$

Kubický kongruenčný generátor

$$X_n = (aX_{n-1}^3 + bX_{n-1}^2 + cX_{n-1} + d) \mod m$$

Joan Boyar dokázala, že lineárny a nesjkôr aj ostatné kongruenčné generátory sú kryptograficky slabé. Nesmú sa používať v silnej kryptografii!!

Máme 256 S-boxov $S[0], S[1], \dots, S[255]$, ktoré obsahujú niektorú permutáciu čísel 0 až 255.

```
rand()
i=i+1 mod 256
j=j+S[i] mod 256
swap(S[i],S[j])
t=(S[i]+S[j]) mod 256
k=S[t]
return k
```

Inicializačná procedúra pre RC4

Kľúč múže byť až 256*8=2048 bitov. Týmito bitmi sa naplnia postupne 8-bitové čísla $K[0], K[1], \dots, K[255]$.

Inicializačná procedúra je takáto:

```
for i=0 to 255
{
  S[i]=i
}
j=0
for i=0 to 255
{
  j=(j+S[i]+K[i]) mod 256
  swap(S[i],S[j])
}
i=0
j=0
```

Podobné sú generátory pseudonáhodných čísel označované ako VMPC.
Je tu to isté nebezpečenstvo pri viacnásobnom používaní rovnakého kľúča ako pri Vernamovej šifre.

10.Testy generatorov (frekvencny, TwoBits, Runs, Poker, FIPS 140-1, Autokorelacny)

Frekvenčný test

Máme postupnosť bitov $\mathbf{b} = b_1, b_2, \dots, b_n$.

$$n_0 \text{ – počet núl} \quad n_1 \text{ – počet jednotiek} \quad n = n_o + n_1$$

Za predpokladu, že $\mathbf{b}$ je náhodná postupnosť s rovnakou pravdepodobnosťou núl a jednotiek má štatistika

$$X_1 = \frac{(n_0 - n_1)^2}{n}$$

$\chi^2(1)$ rozdelenie s jedným stupňom voľnosti pre $n \geq 10$ a testovaná hypotéza H je že $X_1 = 0$.

Dvojbitový sériový test

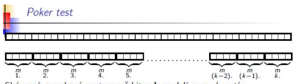
Dvojbitový sériový test
$n_{00}, n_{01}, n_{10}, n_{11}$ – počet výskytov dvojíc 00, 01, 10, 11 v postupnosti $\mathbf{b}$.

Platí $n_{00} + n_{01} + n_{10} + n_{11} = n - 1$.

$$X_2 = \frac{4}{n-1}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1$$

Pre $n \geq 21$ má štatistika X_2 rozdelenie $\chi^2(2)$ s dvoma stupňami voľnosti.
Testujeme platnosť hypotézy $X_2 = 0$.

Poker test



Skúmanú n-prvkovú postupnosť bitov $\mathbf{b}$ rozdelíme na k m-tíc.

Zrejme je $k.m \leq n$.

Číslo m musí byť zvolené tak, aby $k \geq 5.2^m$.

Každá m-tica bitov predstavuje číslo v rozmedzí 0 až $2^m - 1$.
Pre $i = 0, 1, 2, \dots, 2^m - 1$ označme n_i počet m-tíc takých, že predstavujú binárny rozvoj čísla i.

$$X_3 = \frac{2^m}{k} \cdot \left(\sum_{i=0}^{2^m-1} n_i^2\right) - k$$

Štatistika X_3 má rozdelenie $\chi^2(2^m - 1)$ a testujeme hypotézu $X_3 = 0$.

Runs test

Blok dĺžky n je postupnosť n jednotiek v postupnosti $\mathbf{b}$ z oboch strán ohraničená nulou alebo začiatkom alebo koncom postupnosti b.

Medzera (Gap) dĺžky n je postupnosť n núl v postupnosti $\mathbf{b}$ z oboch strán ohraničená jednotkou alebo začiatkom alebo koncom postupnosti b.
Pravdepodobnosť výskytu bloku dĺžky i: $\dots 0 \underbrace{1\,1 \dots 1}_{i} 0 \dots$

v nekonečne dlhej náhodnej postupnosti bitov je $\frac{1}{2^{i+2}}$.

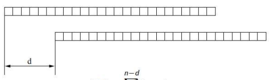
Očakávaný počet blokov dĺžky i v n-prvkovej postupnosti $\mathbf{b}$ je $e_i = \frac{n-i+3}{2^{i+2}}$.

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i}$$

kde k je najväčši také, že $e_i \geq 5$ a B_i, G_i je skutočný počet blokov, resp. medzier dĺžky i v postupnosti $\mathbf{b}$.
Štatistika X_4 má rozdelenie $\chi^2(2k - 2)$, testovaná hypotéza je $X_4 = 0$.

Autokorelačný test

d – pevné číslo $1 \leq d \leq [\frac{n}{2}]$



$$A(d) = \sum_{i=1}^{n-d} b_i \oplus b_{i+d}$$

$$X_5 = 2.\frac{A(d) - \frac{n-d}{2}}{\sqrt{n-d}}$$

Štatistika X_5 má normálne rozdelenie $N(0, 1)$.

Testujeme hypotézu $X_5 = 0$.

FIPS 140–1 štatistický test

Test je určený pre reťaze $\mathbf{b}$ dlhý 20000 bitov.

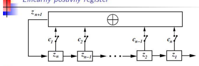
1. Monobit test: $1 < n_1 < 10346$
2. Poker test pre $m = 4$: $1.03 < X_3 < 57.4$
3. Runs test.
   Pre $i = 1, 2, 3, 4, 5$ B_i resp. G_i – počet blokov resp. medzier dĺžky i.
   Pre $i = 6$ B_6 resp. G_6 počet blokov resp. medzier dĺžky 6 a viac.

i	Dovolený rozsah B_i, G_i
1	2267 – 2733
2	1079 – 1421
3	502 – 748
4	223 – 402
5	90 – 223
6	90 – 223

4. Long run test. Nesmie existovať blok alebo medzera dĺžky 34 alebo viac.

11.LFSR

Lineárny posuvný register



Postupnosť $c_1, c_2, \dots, c_n$ – zpätno-väzbová sekvencia – tap sequence

$$z_{n+1} = c_1 z_n \oplus c_2 z_{n-1} \oplus \dots \oplus c_{n-1} z_2 \oplus c_n z_1 \qquad (1)$$

Maximálna perióda LFSR dĺžky n je $2^n - 1$.

Zpätno-väzbový polynóm – connection polynomial – je polynóm nad $\mathbb{Z}_2$:

$$1 + c_1 x + c_2 x^2 + c_3 x^3 + \dots + c_n x^n$$

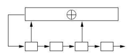
Primitívny polynóm stupňa n je taký polynóm ktorý je

- ireducibilný
- je deliteľom polynómu $x^{2^n-1} + 1$
- nie je deliteľom žiadneho polnómu tvaru $x^d + 1$, kde d delí $2^n - 1$

Zpätnoväzobný polynóm

Platí: Lineárny posuvný register dĺžky n má maximálnu periódu $2^n - 1$ práve vtedy, keď jeho zpätnoväzobný polynóm je primitívny.

Singulárny LFSR je taký LFSR, ktorého dĺžka je väčšia než stupeň väzobného polynómu.

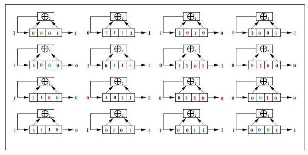


Nie sú zaručená periodicita pre každý počiatočný stav singulárnych LFSR, preto sa v kryptografii nepoužívajú.

Zistiť, či je daný polynóm primitívny je algoritmicky riešiteľný problém.

Hľadanie primitívnych polynómov je ťažké.

Príklad práce LFSR



LFSR v tabuľkovom procesore

	A	B	C	D	E
1	0	0	0	0	0
2	=MOD(A1+D1+ E1;2)	=A1	=B1	=C1	=D1

Druhý riadok tabuľky sa rozkopíruje do ďalších riadkov stĺpcov A až E.

Výstupné bity z LFSR sa použijú ako prúd pseudonáhodných binárnych čísel.

Kľúč:

- Počiatočné nastavenie registra – n bitov $z_1, z_2, \dots, z_n$
- Nastavenie zpätnoväzbovej postupnosti n bitov $c_1, c_2, \dots, c_n$

Ak poznáme zpätnoväzobnú postupnosť a ak a odchytíme porade n bitov z LFSR, ďalšie bity ľahko vypočítame podľa rovnice (1).

Útok na LFSR ak poznáme 2n bitov

Ak poznáme len dĺžky LFSR postupujeme nasledovne:
Predpokladajme, že poznáme n – dĺžku LFSR a $2n$ výstupných bitov:

$$z_{2n}, z_{2n-1}, \dots, z_2, z_1$$

$$\begin{array}{l}
z_{n+1} = c_1 z_n \oplus \dots \oplus c_{n-1} z_2 \oplus c_n z_1 \\
z_{n+2} = c_1 z_{n+1} \oplus \dots \oplus c_{n-1} z_3 \oplus c_n z_2 \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
z_{2n} = c_1 z_{2n-1} \oplus \dots \oplus c_{n-1} x_n z n - 1 \oplus c_n z_n
\end{array}$$

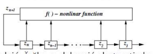
$$\begin{pmatrix} z_n & z_{n-1} & \dots & z_1 \\ z_{n-1} & z_{n-2} & \dots & z_2 \\ \dots & \dots & \dots & \dots \\ z_{2n-1} & z_{2n-2} & \dots & z_n \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ .. \\ c_n \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ .. \\ z_n \end{pmatrix}$$

$$\mathbf{Zc} = \mathbf{z} \qquad \mathbf{c} = \mathbf{Z}^{-1}\mathbf{z}$$

Dôsledok: Kryptografia pomocou LFSR je veľmi slabá a nesmie sa používať.

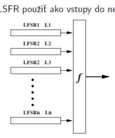
Pokusy o zlepšenie bezpečnosti LFSR

Náhrada $\oplus$ nelineárnou funkciou:



Nevýhoda:Ťažko sa teoreticky študujú, ťažko sa dokazujú vlastnosti ako napr. existencia krátkych cyklov.

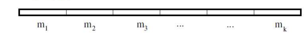
Výstupy z viacerých LSFR použiť ako vstupy do nelineárnej funkcie.



Ak by mal odtlačok správy 64 bitov, t.j. $n = 2^{64}$, stačí vytvoriť $1,17 * 2^{32} \approx 5 * 10^9$ náhodných správ, aby sme s pravdepodobnosťou 1/2 našli kolíziu.
Preto sa používajú odtlačky dlhé 128, 160, 256 bitov.

Všeobecný postup tvorby odtlačku správy

1. Správa M, pre ktorú sa robí odtlačok, sa rozdelí na na rovnako dlhé bloky $m_1, m_2, \dots$.

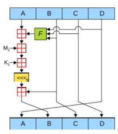


2. Hashovací algoritmus má pevne stanovený inicializačný vektor IV. Položíme $h_0 = IV$.
3. Rekurzívne počítame $h_i = f(m_i, h_{i-1})$.
4. Výsledný odtlačok celej správy $h(M) = h_k$.

dlzka odtlacku 3 kola

MD5

1. Je zosilnením algoritmu MD4.
2. Dáva 128-bitový hash.
3. Pracuje s 512-bitovým blokom textu
4. Namiesto troch kôl má 4 kolá
5. Má pozmenené funkcie takto
   $F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$
   $G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$
   $H(X, Y, Z) = X \oplus Y \oplus Z$
   $I(X, Y, Z) = Y \oplus (X \vee \neg Z)$
6. Beží asi o 30% pomalšie než MD4



SHA

SHA algoritmus

1. SHA produkuje 160-bitový hash
2. Spracováva 512 bitový blok textu $W[0], W[1], \dots, W[15]$, ktorý expanduje do 80 takto: pre $16 \geq j \leq 79$

$$W[j] = W[j-3] \oplus W[j-8] \oplus W[j-14] \oplus W[j-16]$$

3. Má 4 kolá po 20 krokov

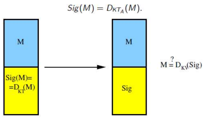
12.Hash funkcie vseobecne, vlastnosti,(vseobecne, MD5, SHA) dorobit

13.Digitalny podpis - (plus Birdthday attack, zmenky)

Digitálny podpis

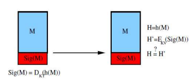
- Účastník A s dvojicou kľúčov KV_A, KT_A podpíše správu M tak, že k nej pripojí výsledok dešifrovania správy M kľúčom KT_A. Teda

$$Sig(M) = D_{KT_A}(M).$$

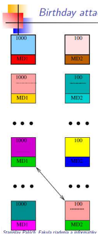


- Účastník B overí pravosť podpisu tak, že vypočíta $M' = E_{KV_A(Sig(M))}$ a skontroluje, či $M = M'$.
  Ak $M' \neq M$, potom buď správa bola zmenená, alebo podpis nie je pravý.
  Ak $M' = M$, potom je podpis pravý a správa nezmenená.
  Jediný človek – účastník A – mohol ku správe M vytvoriť $Sig(M) = D_{KT_A}(M)$, pretože on jediný má kľúč KT_A).

Digitálny podpis



Birthday attack - Narodeninový útok

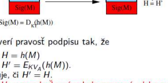


1. Útočník vytvorí dve zmenky – jednu na 100 euro, druhú na 1000 euro
2. Z obidvoch zmeniek bezvýznami zmenami vytvára ich ďalšie varianty dokedy nenájde kolíziu – dvojicu 100 eurového a 1000 eurového variantu s rovnakým odtlačkom h. Ak má odtlačok n možných hodnôt, stačí mu vytvoriť $1.17\sqrt{n}$ dvojíc variantov zmeniek, aby s pravdepodobnosťou $> \frac{1}{2}$ našiel kolíziu.
3. Dĺžníkovi dá potvrdiť 100 eurový variant s odtlačkom h.
4. Po čase vymáha 1000 euro na základe toho, že mu dĺžník potvrdil odtlačok h prislúchajúci 1000 eurovému variantu.

Poučenie: Pred podpisom digitálneho dokumentu vždy v ňom urobiť malú zmenu.

Digitálny podpis

- Účastník A s dvojicou kľúčov KV_A, KT_A podpíše správu M tak, že
  1. Vypočíta $h(M)$ odtlačok správy M.
  2. Odtlačok správy $h(M)$ zašifruje svojim tajným kľúčom:
     $$Sig(M) = D_{KT_A}(h(M)).$$
  3. $Sig(M)$ pripojí k správe M ako svoj digitálny podpis



- Účastník B overí pravosť podpisu tak, že
  1. Vypočíta $H = h(M)$
  2. Vypočíta $H' = E_{KV_A}(h(M))$.
  3. Skontroluje, či $H' = H$.
     Ak $H' \neq H$, potom buď správa bola zmenená, alebo podpis nie je pravý.
     Ak $H' = H$, potom je podpis pravý a správa nezmenená.
     Jediný človek – účastník A – mohol ku správe M vytvoriť $Sig(M) = D_{KT_A}(h(M))$, pretože on jediný má kľúč KT_A).

14.AES

dorobit

15.Asymetricka kryptografia (vseobecne principy, RSA)

dorobit

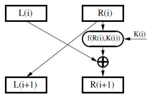
16.Symetricku kryptografiu - Feistelove kola (hlavne DES - komplet, GOST, IDEA- posledne dve hlavne dlzka bloku a dlzka kluca plus omacka)

Všeobecný princíp symetrickej kryptografie

1. A a B sa dohodnú na kryptosystéme
2. A a B sa dohodnú na kľúči
3. A (resp. B) šifruje priamy text x ako $y = E_K(x)$
4. B (resp. A) dešifruje zašifrovaný text y ako $x = D_K(y)$

Kryptosystémy Feistelovho typu

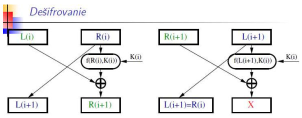
Sú to systémy s blokovou šifrou – šifrujú sa celé bloky priameho textu. Pre kryptosystémy Feistelovho typu musí mať blok párny počet bitov.
Blok sa rozdelí na dve rovnako dlhé časti – ľavú L_i a pravú R_i.



Šifrovanie prebieha po kolách
Jedno kolo urobí:

$$\begin{aligned} R_{i+1} &= L_i \oplus f(R_i, K_i) \\ L_{i+1} &= R_i \end{aligned}$$

Dešifrovanie

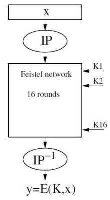


Počítajme X.

$$X = \underbrace{R_{i+1}}_{=L_i \oplus f(R_i, K_i)} \oplus f(\underbrace{L_{i+1}}_{=R_i}, K_i) = L_i \oplus \underbrace{f(R_i, K_i) \oplus f(R_i, K_i)}_{=0} = L_i$$

Dôsledok: Ak kolovému algoritmu vložíme kolový kľúč K_i, na miesto pravej časti L_{i+1} a na miesto ľavej časti R_{i+1}, dostaneme na jeho výstupe na pravej a ľavej časti porade pôvodné L_i a R_i. Ten istý kolový algoritmus (s prehodenou ľavou a pravou stranou a tým istým kolovým kľúčom) teda môžeme použiť ako inverznú funkciu.

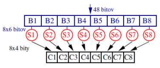
DES – Data Encryption Standard



- Vyvinutý v IBM, publikovaný 1975
- Bloková šifra – 64-bitový blok
- 56-bitový kľúč
- Feistelova sieť so 16 kolami so vstupnou a výstupnou permutáciou
- IP – vstupná (inicializačná) permutácia
- IP^{-1} – výstupná permutácia

Vstupná a výstupná permutácia nemajú žiaden vplyv na bezpečnosť kryptosystému.

DES – Použitie S-boxov



- S-box je tabuľka so štyrmi riadkami a šestnástimi stĺpcami.
- Riadky sú číslované od 0 do 3, stĺpce sú číslované od 0 do 15.
- DES používa 8 S-boxov, bloku B_i je priradený S-box S_i.
- Každé B_i je 6-bitové číslo $b_1b_2b_3b_4b_5b_6$ a predstavuje adresu príslušného štvorbitového čísla C_i v S-boxe S_i.

DES – Adresovanie v S-boxe

Adresa sa vypočíta takto:

Nech $B_1 = b_1b_2b_3b_4b_5b_6$.

b_1b_6 je číslo riadku, $b_2b_3b_4b_5$ je číslo stĺpca v príslušnom S-boxe.
(Riadky i stĺpce sú číslované od 0 po 3 resp. od 0 po 15.)

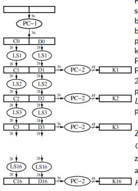
S-box 1:

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

Príklad:
$B_1 = 101011$. $b_1b_6 = (11)_2 = 3$, $b_2b_3b_4b_5 = (0101)_2 = 5$.
V S-boxe S_1 je v riadku 3 a stĺpci 5 číslo 9 (pozor, riadky a stĺpce sa číslujú od 0), ktorého binárny rozvoj je 1001. Je teda

$$S_1(B_1) = S_1(101011) = 1001 = C_1.$$

DES – Generovanie kolových kľúčov



Kľúč pre systém DES je 56-bitový, ale ukladá sa ako 64 bitov s tým, že v každom bajte je 7 bitov kľúča a jeden kontrolný bit doplňujúci bajt na napárnu paritu. Po odstránení paritných bitov sa získa 56 bitov kľúča, ktorých poradie sa zmení podľa permutácie PC-1.
Potom sa 56 bitov kľúča rozdelí na dve 28-bitové časti C_0, D_0, na každú z nich sa postupne aplikuje ľavý rotačný posun $LS_1, LS_2 \dots, LS_{16}$. Pre $i = 1, 2, 9, 16$ je LS_i posun o jedno miesto, inak o 2 miesta.

Získa sa tak postupnosť
$C_1D_1, C_2D_2, \dots, C_{16}D_{16}$ 56 bitových reťazcov, z ktorých operácia PC-2 výberom 48 bitov a ich permutáciou vytvorí postupne kľúče $K_1, K_2, \dots, K_{16}$.

DES – Pravidlá tvorby S- boxov

Jediná nelinearita šifrovacieho algoritmu DES je v S-boxoch. Na nich závisí odolnosť DESu.

1. Každý riadok je permutáciou čísel 0 – 15.
2. Žiaden S-box nie je lineárnou alebo afinnou funkciou vstupov
3. Zmena jedného vstupného bitu S-boxu spôsobí zmenu aspoň dvoch bitov výstupu
4. Pre každý S-box a pre každé šesťbitové x
$S(x)$ a $S(x \oplus 001100)$ sa líšia aspoň v dvoch bitoch
5. Pre každý S-box a pre každé šesťbitové x a pre ľubovoľné bity $r, s \in \{0, 1\}$ $S(x) \neq S(x \oplus 11rs00)$.
6. Ak fixujeme hodnotu jedného vstupného bitu, potom počet vstupných hodnôt, pre ktoré je ľubovoľný určený bit rovný 0 (alebo 1), je medzi 13 a 19.

Útoky proti DESu

Útok hrubou silou.
Počet kľúčov 2^{56} sa ukazuje v dnešnej dobe malý. Podarilo sa prelomiť DES distribuovaným výpočtom na Internete.

Diferenciálna kryptoanalýza.
Je to útok typu "chosen plaintext attack". Šifrovaciemu algoritmu s neznámym kľúčom sa dávajú šifrovať dvojice priamych textov P_1, P_2 s určitou diferenciou $P_1 \oplus P_2$ a na základe diferencie príslušných zašifrovaných textov sa usudzuje na niektoré vlastnosti kľúča.

Lineárna kryptoanalýza

Lineárna kryptoanalýza.
Ak pre priamy text $x_1x_2 \dots x_{64}$, kľúč $k_1k_2 \dots k_{56}$ a pre príslušný zašifrovaný text $y_1y_2 \dots y_{64}$ platí

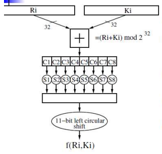
$$\bigoplus_{i=1}^{64} a_ix_i \oplus \bigoplus_{i=1}^{64} b_iy_i = \bigoplus_{i=1}^{56} c_ik_i$$

s pravdepodobnosťou rôznou od $\frac{1}{2}$, dá sa to využiť pri kryptoanalýze.
Pre DES platí

$$x_{17} \oplus y_3 \oplus y_8 \oplus y_{14} \oplus y_{25} = K_{i,26}$$

s pravdepodobnosťou $\frac{1}{2} - \frac{5}{16} = \frac{3}{16}$.
Na základe tohoto faktu bol navrhnutý chosen plaintext attack analyzujúci priemerne 2^{43} známych priamych textov, ktorý odhalil kľúč za 50 dní práce 12 počítačov HP9735 (v roku 1994).

GOST



Sovietsky kryptovací systém používaný v časoch studenej vojny.

Bloková šifra.

64 bitový blok, 256 bitový kľúč.

Feistelova sieť s 32 kolami.

S-boxy sú jednoriadkové tabuľky obsahujúce permutácie čísel $0, 1, \dots, 15$.

IDEA – Špecifikácia

IDEA – International Data Encryption Algorithm (Xueija Lai and James Massey) - 1992.
Je patentovaný, US patent vyprší 7.1.2012.

Bloková šifra – blok 64 bitov
Kľúč 128 bitov.

64- bitový blok sa rozdelí na 4 16-bitové časti x_1, x_2, x_3, x_4, s ktorými s urobí 8 kôl algoritmu plus záverečné "polovičné kolo."

V kolách sa používajú tieto operácie:

$\oplus$ – XOR po bitoch

$\boxplus$ – sčítanie mod 2^{16}

$\odot$ – násobenie mod $(2^{16} + 1)$ pričom sa 16-bitové slovo pozostávajúce zo samých 0 považuje za reprezentáciu čísla 2^{16}.

Jedno kolo algoritmu IDEA