Financované Európskou úniou
NextGenerationEU

PLÁN OBNOVY

MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Základy kryptografie a ochrany dát

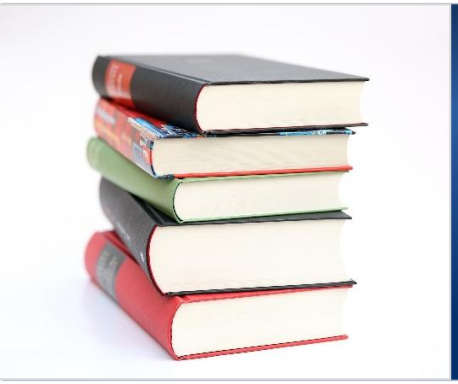
Kryptografia, ochrana dát a bezpečná komunikácia (Blok III)

Kurz: Špecialista kybernetickej bezpečnosti vo verejnej správe

Ing. Tomáš Majer, PhD.

KC KYB UNIZA, https://kc.uniza.sk

tomas.majer@fri.uniza.sk



Obsah

- Čo je kryptografia?
- História kryptografie.
- Klasické metódy šifrovania.
- Symetrické šifrovacie algoritmy.
- Asymetrické šifrovacie algoritmy.
- Kontrola integrity a autenticity.
- Digitálny podpis a infraštruktúra verejného kľúča (PKI).

Čo je kryptografia?

Kryptografia a kryptoanalýza

Kryptografia sa zaoberá štúdiom matematických metód na ochranu a utajenie informácie.

Niekedy sa používa termín **Kryptológia**, ktorá sa delí na:

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

Úlohy kryptografie

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

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Úlohy kryptografie

Ďalšie problémy, ktoré rieši kryptografia:

- Výmena kľúčov.
- Digitálne peniaze.
- Anonymné hlasovacie procedúry.
- ...

Kryptosystém

Priamy text – Plaintext $\rightarrow$ Zašifrovaný text – Ciphertext

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 $\mathcal{T} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, ktoré každej dvojici $K \in \mathcal{K}$, $M \in \mathcal{M}$ priradí zašifrovanú správu $C \in \mathcal{C}$ a také, že

  ak $\mathcal{T}(K, M) = C$ a $\mathcal{T}(K, M') = C$, potom $M = M'$.

  (Existuje teda inverzné zobrazenie $\mathcal{T}^{-1}(K, C) = M$.)

Značenie $\mathcal{T}(K, M) = E_K(M)$, $\mathcal{T}^{-1}(K, C) = D_K(C)$.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Typy kryptografických systémov

- **Symetrická kryptografia** – na šifrovanie i na dešifrovanie sa používa ten istý kľúč.
- **Nesymetrická kryptografia** – kryptografia s verejným kľúčom.
  Na šifrovanie sa používa tzv. verejný kľúč. Prijímateľ správu dešifruje svojím tajným kľúčom. Z verejného kľúča nie je možné odvodiť tajný kľúč.

Príklady symetrických algoritmov

- **Substitučná šifra** – nahrádza znak alebo reťazec znakov iným znakom resp. iným reťazcom.
- **Transpozičná šifra** – znaky ostávajú, mení poradie znakov
- **Monoalfabetická šifra** – šifruje sa znak po znaku, každý znak rovnakým zobrazením
- **Polyalfabetická šifra** – Šifrujú sa k-tice znakov, každý znak v k-tici iným kľúčom
- **Prúdová šifra** – šifruje sa znak po znaku, každý znak iným kľúčom, prúd kľúčov je rovnako dlhý ako šifrovaný text
- **Bloková šifra** – šifrujú sa celé bloky priameho textu

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Čo ostalo z Kerckhoffových zásad

1. Prezradenie šifrovacieho algoritmu nesmie ohroziť bezpečnosť systému
2. Bezpečnosť spočíva iba v utajení kľúča

Bruce Schneier: Existujú dva typy kryptografie:

1. Tá, ktorá zabráni vašej mladše setričke čítať vaše listy
2. Tá, ktorá zabráni ústrednej spravodajskej službe čítať vaše súbory

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

História kryptografie

Egypt, cca 1900 pred n.l.

Egyptský pisári použili neštandardné hieroglyfické symboly namiesto obvyklých hieroglyfov, čo sa považuje za prvé doložené použitie kryptografie.

Mezopotámia, cca 1500 pred n.l.

Tabuľka z Mezopotámie so zašifrovanou formulou na výrobu glazúrovanej keramiky.

600 – 500 pred n.l.

Hebrejci používali primitívnu šifru ATBAŠ založenú na nahradení prvého znaku abecedy posledným, druhého predposledným atď.

Grécko a Sparta, cca 500 pred n.l.

Gréci požívali „skytalé“ – odkaz napísaný na tenkom pásiku kože namotanom na palicu. Na prečítanie odkazu bola potrebná palica s rovnakým priemerom.

Starovekí Rím, Július Caesar, 100 – 44 pred n.l.

Július Caesar používal na šifrovanie správ abecedu posunutú o 3 znaky. Šifra bola na svoju dobu nerozlúštiteľná, kým ju neprezradili Caesarovi bývalí spojenci (Cicero), ktorí prešli k nepriateľom.

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

C	E	A	S	A	R
F	H	D	V	D	U

Indické a arabské korene kryptografie, 725 – 790 n.l.

Abu Abd al-Rahman al Khalil ibn Ahmad ibn Amr ibn Tammam al Farahidi al Zadi al Yahmadi napísal knihu o kryptografii inšpirovanú jeho lúštením šifier pre byzanského cisára. Jeho riešenie bolo založené na znalosti otvoreného textu, čo je dnes štandardná kryptoanalytická metóda.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Benátky, 1226 n.l.

Archívne nálezy pochádzajúce z tohto roku ukazujú, že v Benátkach bola používaná kryptografia, kde bodky a krúžky nahradzovali zvláštnym spôsobom slová a písmená.

Gabrieli di Lavinde, 1379 n.l.

- Vytvoril systém pozostávajúci s úplnej substitučnej abecedy rozšírenej o dvojpísmenové kódy pre cca dve desiatky najfrekventovanejších slov alebo mien.
- Navyše tiež tzv. klamače - nevýznamové skupiny písmen, ktoré mali sťažiť kryptoanalýzu zašifrovaných textov.
- Tento princíp sa používal takmer 450 rokov aj napriek tomu, že už boli k dispozícii aj silnejšie metódy.

Leon Battista Alberti, 1466 – 1467 n.l.

- Napísal prvú prácu napísanú v západnej Európe - 25 stranovú prácu venovanú kryptoanalýze.
- Dielo obsahuje výklad kryptoanalytických postupov na základe jazykových znalostí, roztriedenie systémov šifrovania na substitúciu a transpozíciu, objav polyalfabetickej substitúcie a šifrovanie kódov.

Johannes Trithemius, 1518 n.l.

- Bol to benediktínsky mních.
- Vydal prvú tlačená kniha s kryptologickou náplňou.
- Vymyslel šifru, pri ktorej sa každý znak prameho textu zašifroval podľa ďalšieho riadku Trithemiusovej tabuľky.

Trithemiusova tabuľka

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

$$T + M \rightarrow F$$

Giovani Batista Belas, 1553 n.l.

- Bol to taliansky šľachtic.
- Vydal brožúru "La cifra" popisujúcu kryptosystém, ktorého základom je tajný kód.
  - Tajným kódom je tu slovo prípadne veta, ktorá sa opakovane píše nad otvorený text.
  - Každé písmeno otvoreného textu je potom šifrované riadkom Trithemiusovej tabuľky určenej písmenom kódu nad ním.
  - (Šifra založená na tomto princípe sa neprávom pripisuje Vigenerovi).

Blaise de Vigenere, 1586 n.l.

- Vydal knihu „ Traicté des Chifres“. Navrhuje v nej kryptosystém, v ktorom aj samotná správa môže byť kľúčom.

S	V	J	E	D	N	O	M	J	E	J	P	O	U	Z	I	T	I	P	O	S	T	U	P	U	J
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
v	j	e	d	n	o	m	j	e	j	p	o	u	z	i	t	i	p	o	s	t	u	p	u		
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
N	E	N	H	Q	B	A	V	N	N	Y	D	I	T	H	B	B	X	D	G	L	N	J	J		

Príklad použitý z
Grošek, O, Porubský, Š: Šifrovanie. Algoritmy, metódu, prax. 1992 - Grada.
ISBN 80-85424-62-2

Mária Stuartova a Alžbeta Tudorova, 1586 n.l.

- Listy sprisahancov proti Alžbete prechádzajú rukami kryptoanalytika Thomasa Phelippesa.
- K listu zo 17. júla bolo dokonca pripísaná (správne zašifrovaná) požiadavka na upresnenie mien šiestich spojencov, ktorí mali zavraždiť Alžbetu.

Francois Viete, Francúzsko, 1540-1603

- Bol právnik a matematik, považovaný za zakladateľa algebry.
- Jeden z prvých matematikov pôsobiacich v šifrovacích službách.
- Poradca hugenotského kráľa Henryho IV. bojujúceho proti katolíkom a Španielom.
- Rozlúštil šifry spojencov i mnohé šifry Benátčanov.

Antoine Rossignol, 1599-1682

- Bol francúzsky matematik.
- Neskôr sa stal základom šifrovacej kancelárie kardinála Richelieu.
- V apríli 1628 opevnené juhofrancúzske mestečko Realmont držané hugenotmi bolo obliehané kráľovskými vojskami.
  - A. Rossignol rozlúštil zachytenú zašifrovanú hugenotskú správu žiadajúcu o pomoc.
  - Rozlúštenú správu vrátili do Realmontu, na čo sa jeho obyvatelia obratom vzdali.

sir Francis Bacon, 1623

- Vo svojom diele „ De Augmentis Scientarum“ popísal tzv. biliterálnu šifru, známu v dnešnej dobe ako 5-bitové binárne kódovanie.

A aaaaa	B aaaab	C aaaba	D aaabb	E aabaa	F aabab
G aabba	H aabbb	I abaaa	K abaab	L ababa	M ababb
N abbaa	O abbab	P abbba	Q abbbb	R baaaa	S baaab
T baaba	V baabb	W babaa	X babab	Y babba	Z babbb

Friedrich W. Kasiski, 1863

- Bol to dôstojník pruskej armády.
- Objavil a zverejnil metódu na riešenie polyalfabetickej šifry s opakujúcim sa heslom.

Auguste Kerckhofs, 1835-1903

V roku 1883 vydal knihu „ La Cryptografie militaire“.

Kerckhofs našiel metódu, ako rozlúštiť všeobecnú polyalfabetickú šifru s neperiodickým kľúčom, ak tento bol použitý niekoľkokrát.

Formuloval pravidlá, ktoré má spĺňať kryptosystém:

1. Systém má byť, keď nie teoreticky, tak aspoň prakticky nerozlúštiteľný.
2. Odhalenie systému nesmie spôsobiť ťažkosti korešpondentom.
3. Šifrovací kľúč má byť ľahko zapamätateľný a ľahko zmenený.
4. Zašifrovaný text sa má dať prenášať ďalekopisom.
5. Šifrovací aparát alebo dokument má byť prenosný a obsluhovateľný jednou osobou.
6. Šifrovací systém má byť ľahký, bez dlhého zoznamu pravidiel a bez prepiatych nárokov na duševnú činnosť.

Šifrovací stroj Enigma

- V r. 1919 si obyvateľ holandského Haagu Hugo Alexander Koch zapísal svoj patent šifrovacieho stroja založeného na rotoroch.
- V r. 1923 predal patent Arthurovi Scherbiusovi, nemeckému inžinierovi, ktorý ho vylepšil a nazval Enigma.
- Počas 2. svetovej vojny bol šifrovací stroj používaný v rôznych variáciách predovšetkým nemeckou armádou.

Začiatky asymetrickej kryptografie

- V r. 1976 Whitfeld Diffie a Martin Hellman publikujú *„New Directions in Cryptography“* zavádzajúcu pojem kryptosystému verejného kľúča (nazývaným aj asymetrická kryptografia).
- V r. 1977 oznámili Ronald L. Rivest, Adi Shamir a Leonard M. Adleman objav prvého konkrétneho kryptosystému s privátnym a verejným kľúčom **RSA**.

- V r. 1994 bolo faktorizované 129-ciferné číslo RSA-129. Podľa profesora Rivesta, jedného z tvorcov RSA, mala táto činnosť trvať 4.10^{16} rokov.

Blokové šifry DES, IDEA, AES

- V r. 1977 National Bureau of Standards (NBS) po konzultácii s National Security Agency (NSA) publikuje normu FIPS (Federal Information Processing Standard) pre šifrovací algoritmus Data Encryption Standard (DES) - blokovú šifru založenú na Feistelovej sieti s 56-bitovým tajným kľúčom.
- V r. 1990 Xuejia Lai a James Massey zo Švajčiarska vydali článok A Proposal for a New Block Encryption Standard, ktorý obsahoval návrh šifrovacieho algoritmu International Data Encryption Algorithm (IDEA) a mal nahradiť DES. Vzhľadom na ochranu patentom a nutnosti platiť poplatky sa ale neujal.
- V r. 1997 Bol rozlúštený 56-bitový kľúč k DES pomocou Internetu, podobne ako v roku 1994 u RSA.
- V r. 2000 šifrovací štandard DES bol, po takmer štvorročnej verejnej súťaži, nahradený belgickou šifrou Rijndael (známou pod názvom AES). Šifru Rijndael prihlásili do súťaže známi kryptológovia Joan Daemen a Vincent Rijmen.

Pretty Good Privacy

- V r. 1991 Phil Zimmermann zverejnil jeho prvú verziu Pretty Good Privacy (PGP).
- PGP je šifrovací program, ktorým sa dá zabezpečiť bezpečný prenos e-mailov, ale taktiež telefonovanie cez Internet.
- PGP primárne využíval algoritmy RSA a IDEA.
- Na budovanie dôvery vo verejný kľúč sa nepoužíva certifikačná autorita ale osobný kontakt účastníkov komunikácie.

Klasické metódy šifrovania

Frekvenčná analýza jazyka

— Šifrovanie —

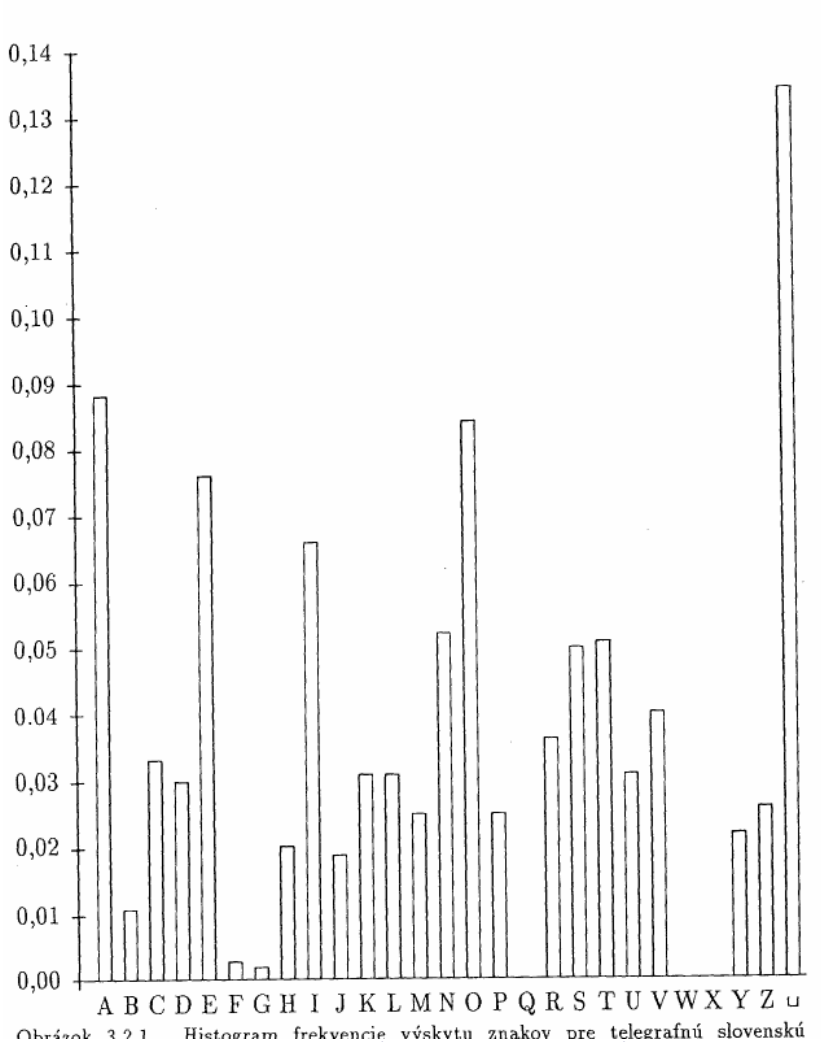
Písmeno	Pravdepodobnosť		Písmeno	Pravdepodobnosť	
	slovenčina	čeština		slovenčina	čeština
A	0,07340	0,054	Ň	0,00139	0,015
Á	0,01545	0,021	O	0,08308	0,068
Ä	0,00060	—	Ó	0,00075	0,000
B	0,01124	0,014	Ô	0,00128	—
C	0,02295	0,019	P	0,02538	0,027
Č	0,01077	0,008	Q	0,00000	0,000
D	0,02919	0,026	R	0,03783	0,029
Ď	0,00141	0,005	Ŕ	0,00006	—
E	0,06927	0,073	Ř	—	0,009
É	0,00669	0,010	S	0,04051	0,040
Ě	—	0,007	Š	0,00918	0,008
F	0,00266	0,002	T	0,04294	0,039
G	0,00222	0,002	Ť	0,00771	0,007
H	0,02050	0,020	U	0,02327	0,030
I	0,05594	0,034	Ú,Ů	0,00875	0,005
Í	0,00996	0,025	V	0,04057	0,039
J	0,01920	0,022	W	0,00011	0,000
K	0,03172	0,033	X	0,00047	0,001
L	0,02976	0,034	Y	0,01341	0,016
Ĺ	0,00006	—	Ý	0,00981	0,008
Ľ	0,00307	—	Z	0,01811	0,019
M	0,02539	0,029	Ž	0,00817	0,009
N	0,05185	0,040	␣	0,13489	0,163

Tabuľka 3.2.1. Relatívna frekvencia výskytu znakov pre zjednodušenú slovenskú a českú abecedu s medzerou

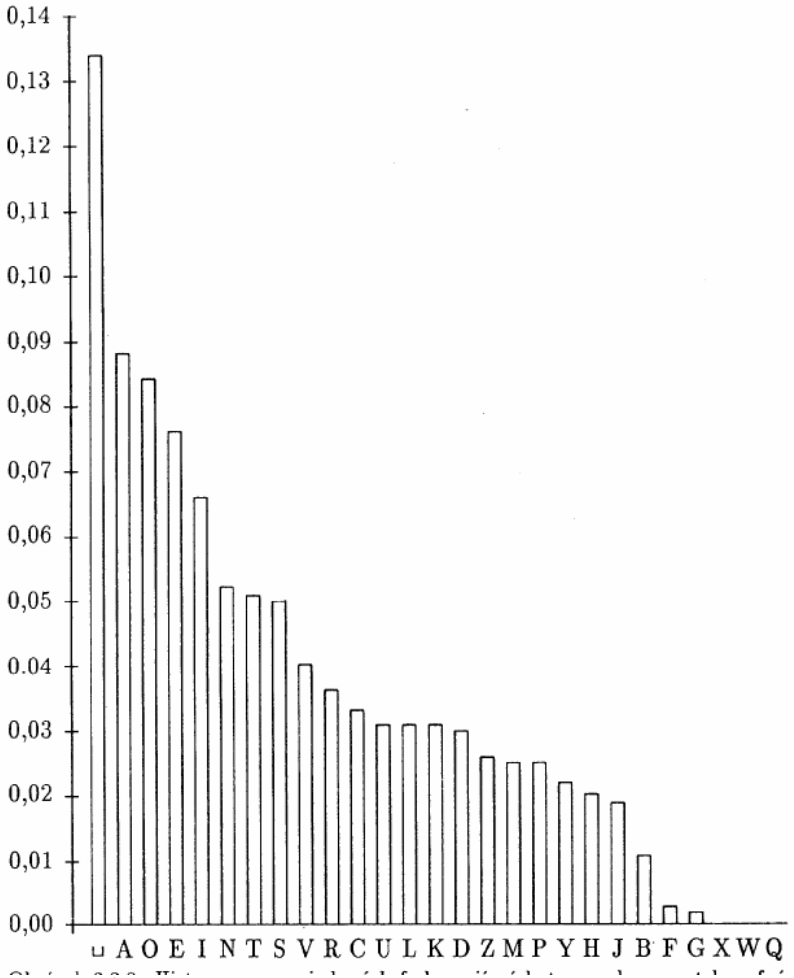
Frekvenčná analýza jazyka

Písmeno	Pravdepodobnosť		Písmeno	Pravdepodobnosť	
	slovenčina	čeština		slovenčina	čeština
A	0,08945	0,065	O	0,08511	0,067
B	0,01124	0,012	P	0,02538	0,016
C	0,03372	0,024	Q	0,00000	0,001
D	0,01124	0,031	R	0,03789	0,052
E	0,07596	0,107	S	0,04969	0,050
F	0,00266	0,023	T	0,03265	0,086
G	0,00222	0,013	U	0,03202	0,021
H	0,02050	0,043	V	0,04057	0,008
I	0,06590	0,056	W	0,00011	0,016
J	0,01920	0,001	X	0,00047	0,001
K	0,03172	0,003	Y	0,02322	0,016
L	0,03189	0,028	Z	0,02628	0,001
M	0,02539	0,020	␣	0,13489	0,182
N	0,05324	0,058			

Frekvenčná analýza jazyka



Obrázok 3.2.1 Histogram frekvencie výskytu znakov pre telegrafnú slovenskú



Frekvenčná analýza jazyka

Písmeno	Pravdepodobnosť		Písmeno	Pravdepodobnosť	
	slovenčina	angličtina		slovenčina	angličtina
A	0,11160	0,0856	N	0,05949	0,0707
B	0,01778	0,0139	O	0,09540	0,0797
C	0,02463	0,0279	P	0,03007	0,0199
D	0,03760	0,0378	Q	0,00000	0,0012
E	0,09316	0,1304	R	0,04706	0,0977
F	0,00165	0,0289	S	0,06121	0,0607
G	0,00175	0,0199	T	0,05722	0,1045
H	0,02482	0,0526	U	0,03308	0,0249
I	0,05745	0,0627	V	0,04604	0,0092
J	0,02158	0,0019	W	0,00001	0,0149
K	0,03961	0,0042	X	0,00028	0,0017
L	0,04375	0,0339	Y	0,02674	0,0199
M	0,03578	0,0249	Z	0,03064	0,0008

Tabuľka 3.2.3. Relatívna frekvencia výskytu znakov pre zjednodušenú slovenskú a anglickú abecedu bez medzerv

Frekvenčná analýza jazyka

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	0	50	245	238	0	3	16	77	4	222	221	439	160	298
B	56	0	5	6	62	0	0	0	50	13	3	38	5	20
C	99	1	0	0	170	0	0	527	428	0	159	28	1	134
D	160	12	21	2	237	0	0	4	160	0	25	22	18	174
E	16	95	139	408	0	12	14	128	1	317	102	194	132	400
F	9	0	0	0	26	0	0	0	77	0	0	3	0	1
G	26	0	0	0	19	0	0	0	20	0	0	1	2	4
H	81	0	6	0	27	0	0	0	19	2	3	69	3	33
I	408	16	345	38	472	8	2	41	20	19	95	153	101	191
J	63	4	3	7	260	0	0	4	46	0	2	4	18	11
K	181	0	4	13	204	0	0	0	4	0	0	73	5	52
L	340	11	1	4	268	0	1	1	314	0	31	0	7	87
M	174	3	1	0	220	1	0	0	198	0	3	17	0	43
N	613	0	30	7	598	6	6	0	577	0	26	0	1	29
O	2	192	265	329	3	36	32	91	2	116	143	242	338	110
P	68	0	5	0	90	0	0	0	39	0	3	72	0	18
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	441	4	11	15	413	1	14	7	356	0	5	0	15	50
S	286	0	21	0	154	4	0	15	283	0	240	101	33	41
T	391	0	6	5	251	0	1	2	374	0	60	21	12	125
U	11	18	147	99	0	0	6	27	1	118	38	51	25	25
V	380	11	16	11	351	0	0	10	144	0	15	41	1	103
W	4	0	0	0	1	0	0	0	0	0	0	0	0	0
X	0	0	0	0	3	0	0	0	14	0	0	0	0	0
Y	0	20	242	2	0	0	0	19	0	2	43	8	109	17
Z	284	16	0	75	149	0	0	17	173	7	31	20	67	148
␣	650	143	275	364	50	70	26	117	190	202	433	94	293	710

Tabuľka 3.2.4. Relatívna frekvencia výskytu dvojíc znakov pre telegrafnú slovenskú abecedu (časť 1)

	O	P	Q	R	S	T	U	V	W	X	Y	Z	␣
A	4	42	0	152	229	408	22	258	3	5	0	174	1473
B	147	0	0	29	18	1	44	0	0	0	92	2	5
C	111	0	0	4	15	16	36	0	0	0	13	0	46
D	288	28	0	52	47	1	79	28	0	0	85	60	120
E	38	41	0	174	178	200	12	92	0	13	0	80	1242
F	14	0	0	5	0	0	3	0	0	0	1	1	1
G	23	0	0	14	0	0	5	0	0	0	3	0	1
H	297	1	0	30	0	14	41	3	0	0	52	0	406
I	43	31	0	18	174	273	38	125	0	0	0	109	774
J	31	4	0	4	52	9	155	7	0	0	0	0	334
K	380	0	0	72	8	182	131	20	0	0	194	0	159
L	306	0	0	0	60	8	99	4	0	0	47	1	101
M	156	15	0	6	0	6	135	0	0	0	29	0	339
N	385	0	0	1	53	66	105	2	0	0	234	6	79
O	3	54	0	318	350	155	157	577	0	0	0	253	745
P	467	0	0	534	13	3	16	0	0	0	4	0	12
Q	0	0	0	0	0	0	0	0	0	0	0	0	0
R	391	6	0	0	34	16	86	24	0	5	66	11	38
S	151	153	0	7	10	804	110	57	0	0	27	0	138
T	528	0	0	230	16	2	122	96	2	0	88	1	353
U	0	60	0	43	134	106	0	36	0	0	0	66	686
V	277	7	0	17	93	2	24	0	0	0	291	63	294
W	0	0	0	0	0	0	0	0	0	0	0	0	1
X	1	3	0	0	0	0	0	2	0	0	0	0	2
Y	0	16	0	19	85	29	16	34	0	0	0	21	549
Z	115	19	0	32	17	17	28	63	0	0	5	0	110
␣	357	864	0	248	1049	368	234	723	1	2	0	545	0

Tabuľka 3.2.4. Relatívna frekvencia výskytu dvojíc znakov pre telegrafnú slovenskú abecedu (časť 2)

Frekvenčná analýza jazyka

␣PR	455	OVA	166	ICK	131	YCH	270	IST	113	VAT	85
␣NA	391	STA	166	A␣N	127	OST	236	ACI	111	TAT	84
CH␣	377	␣JE	166	JE␣	127	OVA	197	AST	107	ENE	83
␣A␣	362	HO␣	162	NOS	125	STI	181	NAS	107	EPR	82
␣PO	302	␣ST	162	ENI	124	PRE	180	EJS	105	NIC	82
OST	251	A␣P	160	O␣S	122	STA	173	NOV	105	EDN	79
EJ␣	248	PRI	157	A␣Z	118	TOR	159	ICH	104	CKE	78
YCH	233	E␣S	156	CIA	115	PRI	157	ALE	99	ENA	78
NE␣	231	TOR	155	OVE	115	ALI	156	EST	98	ITA	78
NA␣	215	TI␣	150	E␣V	114	ANI	148	SPO	98	NIA	78
IE␣	210	ALI	149	LA␣	114	NIE	141	NEJ	97	POD	78
␣SA	210	␣DO	147	␣VE	114	ENI	140	LAD	95	RAV	78
␣ZA	197	␣V␣	143	EHO	113	VED	140	NYC	94	RED	78
A␣S	194	OU␣	142	␣SP	113	KTO	138	CIT	92	AKO	77
SA␣	186	TO␣	141	STR	112	ICK	131	IAL	91	LOV	77
␣VY	186	NIE	140	E␣N	111	NOS	128	INA	91	SKO	77
PRE	180	␣RO	139	LI␣	110	PRA	127	APR	90	TIC	77
OM␣	178	VED	137	NY␣	109	OVE	126	OCI	90	AJU	76
STI	176	E␣P	134	E␣A	108	EHO	122	EDO	87	STO	75
IA␣	172	KTO	133	JU␣	108	STR	118	VAN	87	VOJ	75
␣␣NE	167	A␣␣V	132	␣␣KT	107	CIA	117	ANA	85	CHO	73

Frekvenčná analýza jazyka

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

Frekvenčná analýza jazyka

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

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

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.

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

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

Pre $q = 26$

$$\sum_{i=1}^{26} p(a_i)^2 - 0,03846$$

Index koincidencie

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 líši 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 $0,0553$.

Index koincidencie

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

Index koincidencie

Máme text (je jedno,či je priamy alebo zašifrovaný) obsahujúci n znakov. Z toho je n_1 znakov a_1, n_2 znakov a_2, atď. až n_q znakov a_q.

Počet neusporiadaných dvojíc, v ktorých sú oba znaky a_i je $\frac{n_i(n_i-1)}{2}$ počet všetkých neusporiadaných dvojíc znakov v danom texte je $\frac{n(n-1)}{2}$.

Pravdepodobnosť, že oba znaky budú a_i je teda

$$p(a_i)^2 \approx \frac{n_i(n_i-1)/2}{n(n-1)/2} = \frac{n_i(n_i-1)}{n(n-1)}$$

Pravdepodobnosť $\sum_{i=1}^{q} p(a_i)^2$ javu, že oba znaky budú rovnaké, odhadneme hodnotou

$$\kappa = \frac{\sum_{i=1}^{q} n_i(n_i-1)}{n(n-1)} \tag{18}$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Cézarovská šifra

Ceasar požíval na šifrovanie túto tabuľku – posun písmena o tri znaky

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

Zovšeobecnenie – posun o k znakov

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

Použijeme túto reprezentáciu (kódovanie) znakov abecedy $\{A, B, \ldots, Z\}$

$$A \equiv 0,\ B \equiv 1,\ C \equiv 2,\ D \equiv 3,\ \ldots,\ Y \equiv 24,\ Z \equiv 25$$

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

Potom možno abecedu považovať za okruh zvyškových tried $\mathbb{Z}_{26}$ s operáciami $\oplus$, $\otimes$ definovanými pre $a, b \in \mathbb{Z}_{26}$ takto:

$$a \oplus b = (a + b) \mod 26 \qquad a \otimes b = (a.b) \mod 26 \tag{1}$$

Cézarovská šifra

Pôvodná Ceasarova šifra:

$$\text{šifrovanie: } y = E(x) = x \oplus D \qquad \text{dešifrovanie: } x = D(y) = y \ominus D$$

Zovšeobecnená šifra – Ceasarovská šifra s kľúčom $k \in \mathbb{Z}_{26}$:

$$\text{šifrovanie: } y = E_k(x) = x \oplus k \qquad \text{dešifrovanie: } x = D_k(y) = y \ominus k$$

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

Cézarovská šifra

Útok hrubou silou – vyskúšať 25 kľúčov, pokiaľ nedostaneme zrozumiteľný dešifrovaný text.

Je to „ciphertext only attack“ a „brute force attack“.

Podstatná je tu skutočnosť, že vieme rozhodnúť, či dešifrovaná správa patrí do množiny $\mathcal{M}$ správ kryptosystému.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Afínna šifra

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. $k_1 \otimes k_1^{-1} = 1 \equiv \mathrm{B}$).

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

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.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Afínna šifra

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

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

$$k_1 \otimes 2 \oplus k_2 = 15 \tag{2}$$
$$k_1 \otimes 5 \oplus k_2 = 7 \tag{3}$$

Odčítaním rovnice (2) od (3)

$$k_1 \otimes 3 = -8 \mod 26 = 18 \qquad / * 9 \equiv 3^{-1} \tag{4}$$
$$k_1 = 18 * 9 \mod 26 = 162 \mod 26 = 6 \tag{5}$$

Dosadením za k_1 do (2)

$$(6 \otimes 2) \oplus k_2 = 15 \tag{6}$$
$$k_2 = 15 \ominus 12 = 3 \tag{7}$$

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 prameho textu inak.

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

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

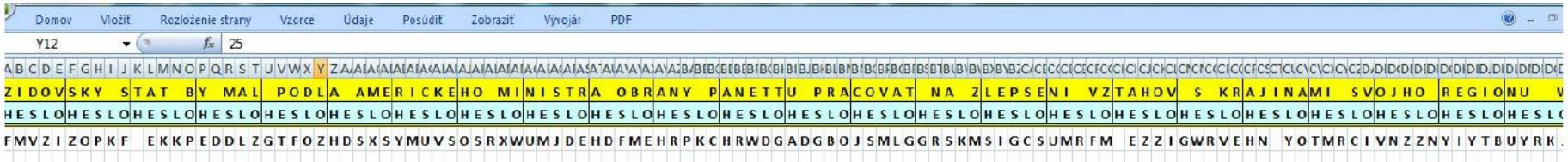
Vigenerovská šifra

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

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

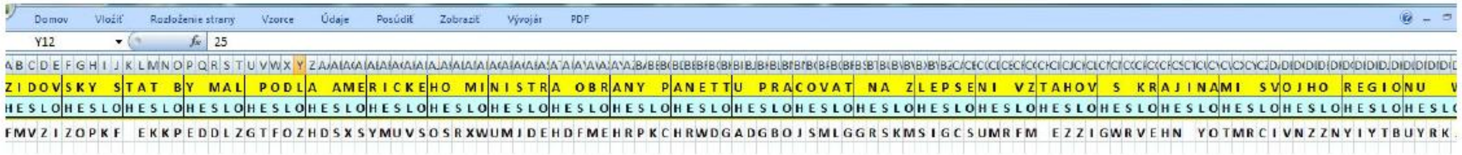
Ide o monoalfabetickú alebo polyalfabetickú šifru?

Ak ide o monoalfabetickú šifru, index koincidencie bude zašifrovaného textu bude blázky indexu jazyka (napr. 0,06). Ak bude bližšie hodnote $1/q$, jedná sa pravdepodobne o poyalfabetickú šifru.

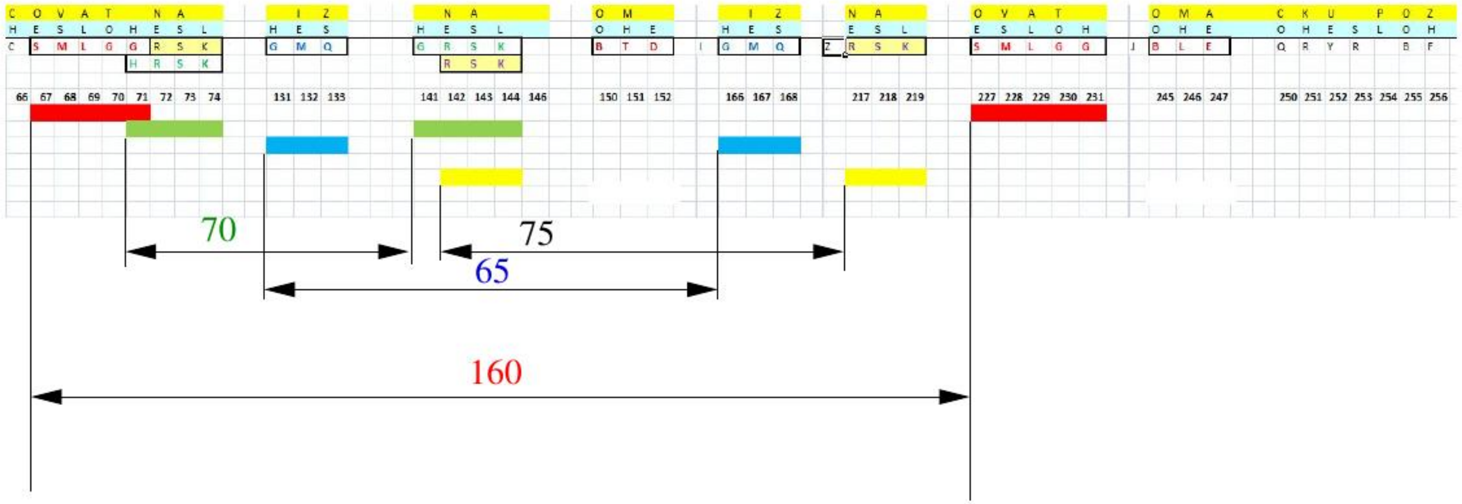


Priamy text mal $\kappa = 0,06216$.
Zašifrovaný text $\kappa = 0,04116$, pričom $1/27 = 0,03704$
Záver: s veľkou pravdepodobnosťou ide o polyalfabetickú šifru.

Kasiského test na zistenie dĺžky hesla



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



Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Kasiského test na zistenie dĺžky hesla

Prvvý výskyt	Druhý výskyt	Offset	Trojica
67	227	160	S M L
68	228	160	M L G
69	229	160	L G G
71	141	70	G R S
72	142	70	R S K
72	217	145	R S K
131	166	35	G M Q
142	217	75	R S K
192	244	52	W B L

Dĺžka kľúča je pravdepodobne najväčším spoločným deliteľom vzdialeností rovnakých výskytov

Zisťovanie dĺžky kľúča metódou koincidencie

Majme dva priame texty
$\mathbf{r} = r_1 r_2 \dots r_n$, $\mathbf{s} = s_1 s_2 \dots s_n$
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"
$\overline{\mathbf{r}} = T_1(r_1) T_2(r_2) \dots T_n(r_n)$,
$\overline{\mathbf{s}} = T_1(s_1) T_2(s_2) \dots T_n(s_n)$.
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.

Friedmanov test

Zoradíme zašifrovaný text $\mathbf{s} = s_1 s_2 \dots s_n$ do tabuľky

1	2	...	...	k
s_1	s_2	...	...	s_k
s_{k+1}	s_{k+2}	...	...	s_{2k}
s_{2k+1}	s_{2k+2}	...	...	s_{3k}
s_{3k+1}	s_{3k+2}	...	...	s_{3k}

Ak sa k rovná dĺžke kľúča jednotlivé stĺpce sú už zašifrované monoalfabeticky, a vtedy by indexy koincidencie počítané zvlášť pre každý stĺpec mali stúpnuť.
Nech $Z_1, Z_2, \dots, Z_t$ sú najčastejšie znaky v prvom stĺpci.
Medzi nimi je s veľkou pravdepodobnosťou zašifrované niektoré z najčastejších znakov - A, O, E, I.
Preto sa medzi znakmi typu $Z_i - A$, $Z_i - O$,$Z_i - E$, $Z_i - I$
nachádza 1. znak kľúča.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Hillovská šifra

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

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

$$\underbrace{x_{11}x_{12}\ldots x_{1n}}_{\mathbf{x}_1}\underbrace{x_{21}x_{22}\ldots x_{2n}}_{\mathbf{x}_2}\ldots\ldots\ldots\underbrace{x_{m1}x_{m2}\ldots x_{mn}}_{\mathbf{x}_m} \tag{1}$$

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

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

Hillovská šifra

Šifrovanie

$$\mathbf{y} = \mathbf{K}\mathbf{x} \quad \begin{pmatrix} y_1 \\ y_2 \\ .. \\ y_n \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \dots & \dots & \dots & \dots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{pmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{pmatrix} \qquad (3)$$

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

Hillovská šifra

Dešifrovanie

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

Dešifrovanie je korektné, lebo

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

Hillovská šifra

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 $\mathrm{mod}\,(-11305, 26) = 5$ je číslo, ku ktorému existuje v $\mathbb{Z}_{26}$ inverzný prvok – totiž 21.

Hillovská šifra

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

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

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

Hillovská šifra

$$\left(\begin{array}{cccc|cccc} 17 & 4 & 3 & 9 & 1 & 0 & 0 & 0 \\ 0 & 25 & 4 & 17 & 3 & 1 & 0 & 0 \\ 0 & 0 & 25 & 1 & 10 & 2 & 1 & 0 \\ 0 & 0 & 14 & 23 & 5 & 6 & 0 & 1 \end{array}\right)$$

$$\left(\begin{array}{cccc|cccc} 17 & 4 & 3 & 9 & 1 & 0 & 0 & 0 \\ 0 & 25 & 4 & 17 & 3 & 1 & 0 & 0 \\ 0 & 0 & 25 & 1 & 10 & 2 & 1 & 0 \\ 0 & 0 & 0 & 11 & 15 & 8 & 14 & 1 \end{array}\right)$$

Teraz máme maticu upravenú na hornú trujuhojníkovú. Ešte treba dosiahnúť nuly nad diagonálou.

Hillovská šifra

$$\left(\begin{array}{cccc|cccc} 17 & 4 & 3 & 0 & 10 & 10 & 24 & 11 \\ 0 & 25 & 4 & 0 & 20 & 17 & 2 & 15 \\ 0 & 0 & 25 & 0 & 11 & 6 & 21 & 7 \\ 0 & 0 & 0 & 11 & 15 & 8 & 14 & 1 \end{array}\right)$$

$$\left(\begin{array}{cccc|cccc} 17 & 4 & 0 & 0 & 17 & 2 & 9 & 6 \\ 0 & 25 & 0 & 0 & 12 & 15 & 8 & 17 \\ 0 & 0 & 25 & 0 & 11 & 6 & 21 & 7 \\ 0 & 0 & 0 & 11 & 15 & 8 & 14 & 1 \end{array}\right)$$

$$\left(\begin{array}{cccc|cccc} 17 & 0 & 0 & 0 & 13 & 10 & 15 & 22 \\ 0 & 25 & 0 & 0 & 12 & 15 & 8 & 17 \\ 0 & 0 & 25 & 0 & 11 & 6 & 21 & 7 \\ 0 & 0 & 0 & 11 & 15 & 8 & 14 & 1 \end{array}\right)$$

Hillovská šifra

Pretože $17^{-1} = 23(\mod 26)$, $25^{-1} = 25(\mod 26)$, $11^{-1} = 19(\mod 26)$, vynásobením prvého riadku matice číslom 23 druhého a tretieho číslom 25 a posledného číslom 19 (všetko modulo 26) dosiahneme:

$$\left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 13 & 22 & 7 & 12 \\ 0 & 1 & 0 & 0 & 14 & 11 & 18 & 9 \\ 0 & 0 & 1 & 0 & 15 & 20 & 5 & 19 \\ 0 & 0 & 0 & 1 & 25 & 22 & 6 & 19 \end{array}\right)$$

Máme teda:

$$\mathbf{K}^{-1} = \begin{pmatrix} 13 & 22 & 7 & 12 \\ 14 & 11 & 18 & 9 \\ 15 & 20 & 5 & 19 \\ 25 & 22 & 6 & 19 \end{pmatrix}$$

Hillovská šifra

$$\mathbf{K}.\mathbf{x} = \begin{pmatrix} 17 & 4 & 3 & 9 \\ 1 & 13 & 21 & 16 \\ 10 & 12 & 5 & 9 \\ 13 & 6 & 3 & 12 \end{pmatrix} \begin{pmatrix} A \equiv 0 \\ B \equiv 1 \\ C \equiv 2 \\ D \equiv 3 \end{pmatrix} = \begin{pmatrix} L \equiv 11 \\ Z \equiv 25 \\ X \equiv 23 \\ W \equiv 22 \end{pmatrix}$$

$$\mathbf{K}^{-1}.\mathbf{y} = \begin{pmatrix} 13 & 22 & 7 & 12 \\ 14 & 11 & 18 & 9 \\ 15 & 20 & 5 & 19 \\ 25 & 22 & 6 & 19 \end{pmatrix} \begin{pmatrix} L \equiv 11 \\ Z \equiv 25 \\ X \equiv 23 \\ W \equiv 22 \end{pmatrix} = \begin{pmatrix} A \equiv 0 \\ B \equiv 1 \\ C \equiv 2 \\ D \equiv 3 \end{pmatrix}$$

Ukážka toho, že zmena jedného znaku v bloku priameho textu má za následok (vo väčšine prípadov) zmenu všetkých znakov v zašifrovanom texte.

$$\mathbf{K}.\mathbf{x}' = \begin{pmatrix} 17 & 4 & 3 & 9 \\ 1 & 13 & 21 & 16 \\ 10 & 12 & 5 & 9 \\ 13 & 6 & 3 & 12 \end{pmatrix} \begin{pmatrix} P \equiv 15 \\ B \equiv 1 \\ C \equiv 2 \\ D \equiv 3 \end{pmatrix} = \begin{pmatrix} G \equiv 6 \\ O \equiv 14 \\ R \equiv 17 \\ J \equiv 9 \end{pmatrix}$$

Hillovská šifra

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

$$\mathbf{y}_1 = \mathbf{K}\mathbf{x}_1, \mathbf{y}_2 = \mathbf{K}\mathbf{x}_2, \ldots, \mathbf{y}_n = \mathbf{K}\mathbf{x}_n \tag{5}$$

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

$$\mathbf{Y} = \mathbf{K}.\mathbf{X} \tag{6}$$

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

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

Symetrické šifrovacie algoritmy
Prúdové šifry

Výpočtová a bezpodmienečná bezpečnosť

Definition (Výpočtová bezpečnosť kryptografického systému.)

Hovoríme, že kryptosystém je výpočtovo bezpečný, ak najlepší známy algoritmus na jeho prelomenie vyžaduje aspoň N krokov, kde N je špecifikované veľmi veľké číslo.
Iný prístup:
Hovoríme, že kryptosystém je výpočtovo bezpečný, ak problém jeho prelomenia je polynomiálne ekvivalentný s niektorou NP ťažkou úlohou.

Definition (Bezpodmienečná bezpečnosť kryptografického systému.)

Kryptosystém je bezpodmienečne bezpečný, ak ho nemožno prelomiť ani s nekonečným množstvom výpočtových prostriedkov.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Perfektná bezpečnosť

Definition

Hovoríme, že kryptosystém $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ má perfektnú bezpečnosť, keď pomienená pravdepodobnosť javu bola vyslaná priama správa $x \in \mathcal{P}$ za predpokladu javu bola prijatá zašifrovaná správa $y \in \mathcal{C}$, sa rovná pravdepodobnosti vyslania správy x, t.j.

$$P[M = x | C = y] = P[M = x].$$

Majme cézarovskú šifru $x,\ y,\ , k \in \mathbb{Z}_{26}$, $y = x \oplus_{26} k$, s rovnomerným rozdelením prvdepodobnosti kľúčov, t.j. $\forall k \in \mathbb{Z}_{26}\ P[K = k] = \frac{1}{26}$.

$$P[C = y] = \sum_{k \in \mathcal{K}} P[K = k].P[M = d_k(y)] =$$

$$\sum_{k \in \mathcal{K}} P[K = k].P[M = (y \ominus_{26} k)] =$$

$$\sum_{k \in \mathcal{K}} \frac{1}{26}.P[M = (y \ominus_{26} k)] = \frac{1}{26} . \underbrace{\sum_{k \in \mathcal{K}} P[M = (y \ominus_{26} k)]}_{=1} = \frac{1}{26}$$

$$P[C = y] = \frac{1}{26}$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Perfektná bezpečnosť

$$P[C = y|M = x] = P[K = y \ominus_{26} x] = \frac{1}{26}$$

$$P[M = x|C = y] = \underbrace{\frac{P[M = x].P[C = y|M = x]}{P[C = y]}} = \frac{P[M = x].\frac{1}{26}}{\frac{1}{26}} = P[M = x]$$

$$P(A|B) = \frac{P(A).P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

Theorem

Cézarovská šifra aplikovaná na jeden znak má perfektnú bezpečnosť, ak sa zakaždým použije iný kľúč s rovnomerným rozdelením pravdepodobnosti.

Theorem

Nech $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ je kryptosystém, kde $|\mathcal{P}| = |\mathcal{C}| = |\mathcal{K}|$. Potom tento systém má perfektnú bezpečnosť práve vtedy, keď každý kľúč sa používa s rovnakou pravdepodobnosťou $1/|\mathcal{K}|$ a pre každé $x \in \mathcal{P}$ a pre každé $y \in \mathcal{C}$ existuje práve jeden kľúč $k \in \mathcal{K}$ taký, že $y = e_k(x)$.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Informácia o znakoch správy v šifrovaných znakoch

Pokus $\mathbf{B} = \{b_1, b_2, \ldots, b_{26}\}$, $H(\mathbf{B}) = -\sum_{i=1}^{26} P(b_i).\log_2 P(b_i)$.
Pokus $\mathbf{B}$ predstavuje vyslanie jedného znaku priameho textu.
Pokus $\mathbf{A} = \{a_1, a_2, \ldots, a_{26}\}$, $P(a_i) = \frac{1}{26}$ $H(\mathbf{A}) = -\log_2(26)$.
Pokus $\mathbf{A}$ predstavuje prijatie jedného znaku zašifrovaného textu.

$$H(\mathbf{B}|a_i) = -\sum_{j=1}^{26} P(b_j|a_i)\log_2 P(b_j|a_i)$$

$$H(\mathbf{B}|\mathbf{A}) = \sum_{i=1}^{26} P(a_i)H(\mathbf{B}|a_i) = -\sum_{i=1}^{26} P(a_i)\sum_{j=1}^{26} P(b_j|a_i)\log_2 P(b_j|a_i)$$

Ale $P(b_j|a_i) = p(b_j)$

$$H(\mathbf{B}|\mathbf{A}) = -\sum_{i=1}^{26} P(a_i)\sum_{j=1}^{26} P(b_j)\log_2 P(b_j) = \sum_{i=1}^{26} P(a_i)H(\mathbf{B}) =$$
$$= H(\mathbf{B})\sum_{i=1}^{26} P(a_i) = H(\mathbf{B})$$

Stredná informácia o vyslanom znaku priameho textu (– o pokuse $\mathbf{B}$)
v prijatom znaku zašifrovaného textu (– v pokuse $\mathbf{A}$) je

$$I(\mathbf{A}, \mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A}) = H(\mathbf{B}) - H(\mathbf{B}) = 0$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Princíp prúdovej šifry

Princíp prúdovej šifry:
$x_1, x_2, \ldots, x_n, \ldots$ – prúd znakov prameho textu
$k_1, k_2, \ldots, k_n, \ldots$ – prúd kľúčov

Prúd zašifrovaných znakov bude:

$$y_1, y_2, \ldots, y_n, \cdots = E_{k_1}(x_1), E_{k_2}(x_2), \ldots, E_{k_n}(x_n), \ldots$$

Cézarovské a vigenèrovské šifry možno pozmeniť tak, že znaky abecedy si predstavíme zkódované nejakým binárnym kódom (napr. ASCII kódom) a namiesto operácie $\oplus$ vykonáme operáciu XOR po bitoch značenú symbolom $\oplus$.

XOR	0	1
0	0	1
1	1	0

Potom

$$E_k(x) = x \oplus k \quad \text{a} \quad D_k(y) = y \oplus k$$

Binárna operácia $\oplus$ je symetrická a asociatívna na $\mathbb{Z}_2$.
Platí $x \oplus x = 0$, $x \oplus 0 = x$ pre $x \in \{0, 1\}$.

Vernamova šifra (One Time Pad)

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

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

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:

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

Iný spôsob:
Predpoklad $P(k_i = 0) = 1/2 + \epsilon$, $P(k_i = 1) = 1/2 - \epsilon$.
Položme $z_i = k_{2i} \oplus k_{2i+1}$.

$$P(z_i = 0) = P(k_{2i} = 0).P(k_{2i+1} = 0) + P(k_{2i} = 1).P(k_{2i+1} = 1) =$$

$$\left(\frac{1}{2} + \epsilon\right)^2 + \left(\frac{1}{2} - \epsilon\right)^2 = \frac{1}{2} + 2\epsilon^2$$

Útok pri opakovanom použití kľúča

Predpokladajme, že dve postupnosti znakov priameho textu

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

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

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

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

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

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

Postupnosť $w_1, w_2, \ldots$ 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

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

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

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

Ak vytvorí postupnosť

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

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

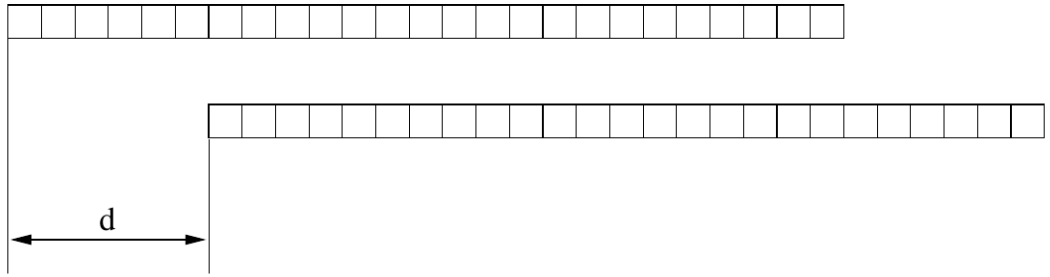
$$w_i = y_i \oplus z_{i+d} = (a_i \oplus k_{i+d}) \oplus (b_{i+d} \oplus k_{i+d}) = (a_i \oplus b_{i+d}) \oplus (k_{i+d} \oplus k_{i+d}) = a_i \oplus b_{i+d},$$

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

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Synchronizácia zašifrovaných textov

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



Generátory pseudonáhodnej postupnosti

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!!

Algoritmus RC4

Máme 256 S-boxov $S[0], S[1], \ldots, S[255]$, 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
```


Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Algoritmus RC4

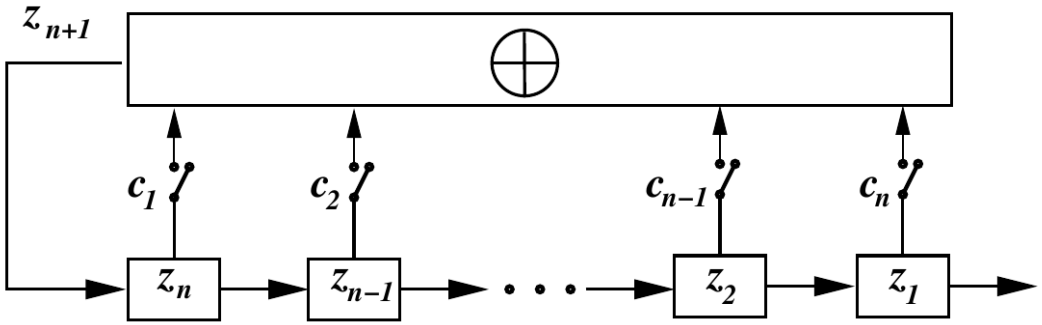
Kľúč môže byť až 256*8=2048 bitov. Týmito bitmi sa naplnia postupne 8-bitové čísla $K[0], K[1], \ldots, K[255]$.

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.

Posuvné registre s lineárnou spätnou väzbou (LFSR)



Postupnosť $c_1, c_2, \ldots, c_n$ – spätno-väzbová sekvencia – tap sequence

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

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

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

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

Spätnoväzobný polynóm

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

Primitívny polynóm stupňa n je taký polynóm ktorý je

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

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

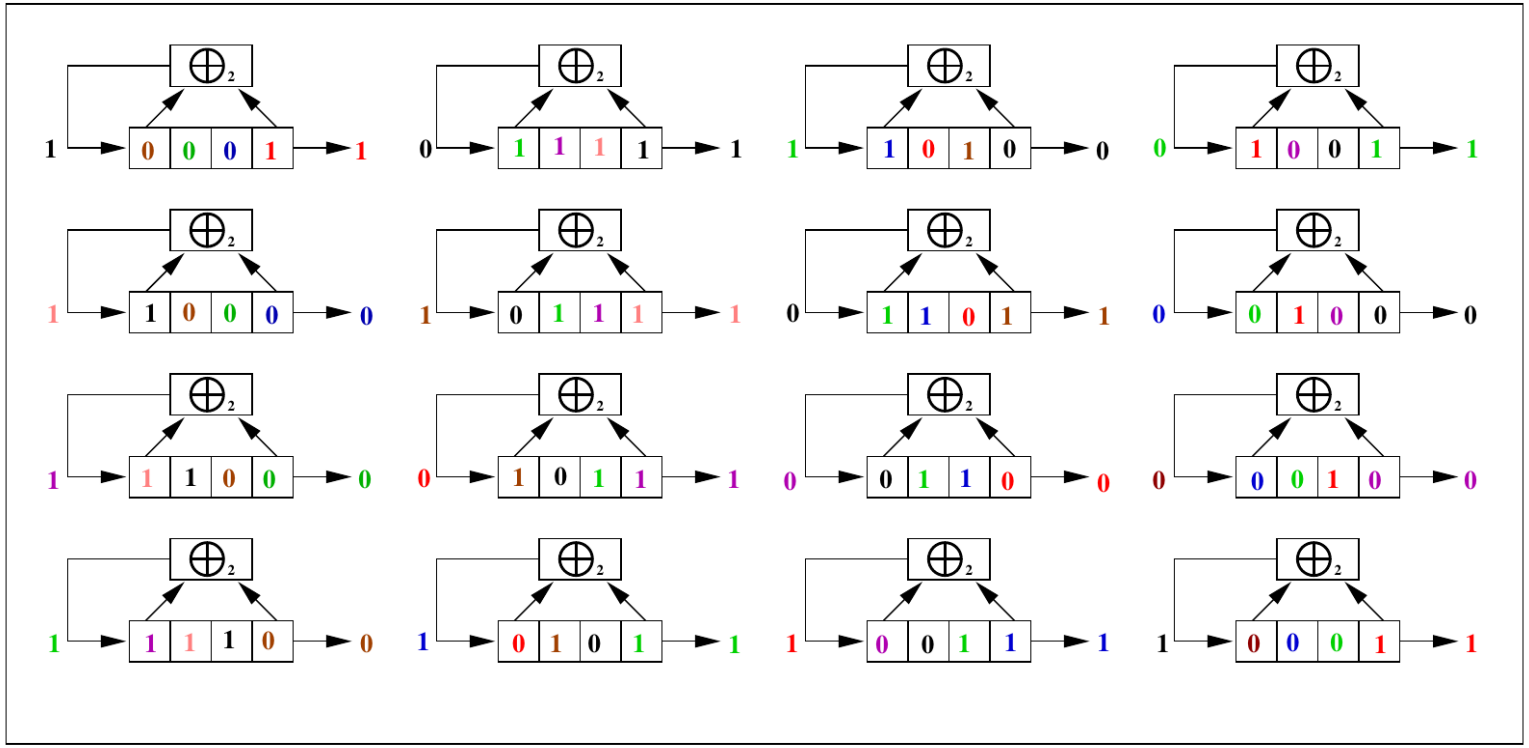
Nie je zaručená periodicita pre každý počiatočný stav singulárnych LFSR, preto sa v kryptografii nepoužívajú.

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

Hľadanie primitívnych polynómov je ťažké.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

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 $z_1, z_2, \ldots, z_n$
- Nastavenie spätnoväzbovej postupnosti n bitov $c_1, c_2, \ldots, c_n$

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

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

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 $z_1, z_2, \ldots, z_n$
- Nastavenie spätnoväzbovej postupnosti n bitov $c_1, c_2, \ldots, c_n$

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

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Útok na LFSR ak poznáme 2n bitov

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

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

$$\begin{aligned}
z_{n+1} &= c_1 z_n \oplus c_2 z_{n-1} \oplus \ldots \oplus c_{n-1} z_2 \oplus c_n z_1 \\
z_{n+2} &= c_1 z_{n+1} \oplus c_2 z_n \oplus \ldots \oplus c_{n-1} z_3 \oplus c_n z_2 \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
z_{2n} &= c_1 z_{2n-1} \oplus c_2 z_{2n-2} \oplus \ldots \oplus c_{n-1} z_{n+1} \oplus c_n z_n
\end{aligned}$$

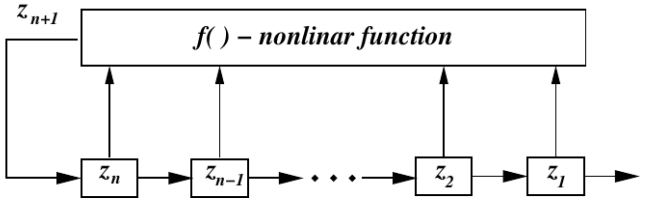
$$\begin{pmatrix} z_n & z_{n-1} & \ldots & z_1 \\ z_{n+1} & z_n & \ldots & z_2 \\ \ldots & \ldots & \ldots & \ldots \\ z_{2n-1} & z_{2n-2} & \ldots & z_n \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \ldots \\ c_n \end{pmatrix} = \begin{pmatrix} z_{n+1} \\ z_{n+2} \\ \ldots \\ z_{2n} \end{pmatrix}$$

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

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

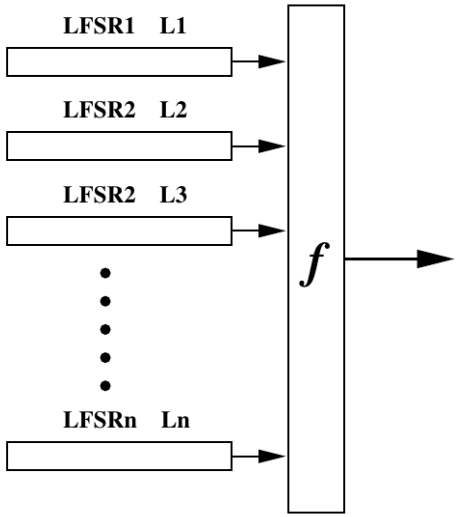
Pokusy o zlepšenie LFSR

Náhrada $\oplus$ nelineárnou funkciou:

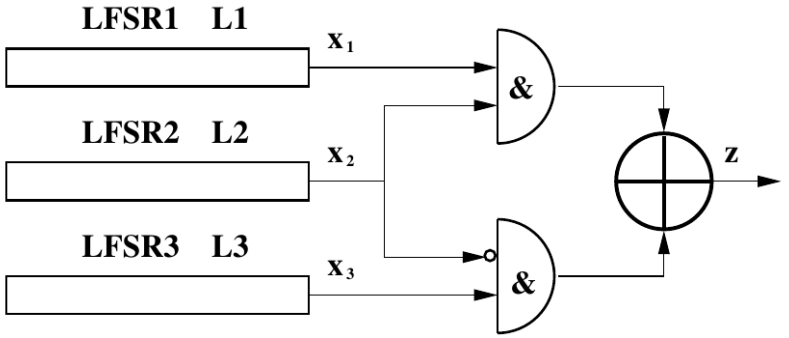


Nevýhoda:Ťažko sa teoreticky študujú, ťažko sa dokazujú vlastnosti ako napr. existencia krátkych cyklov.

Výstupy z viacerých LSFR použiť ako vstupy do nelineárnej funkcie.



Geffeho generátor



$$z = x_1.x_2 \oplus (1 \oplus x_2).x_3$$

$$P[z = x_1] = \underbrace{P[x_2 = 1]}_{=\frac{1}{2}} + \underbrace{P[x_2 = 0]}_{=\frac{1}{2}} . \underbrace{P[x_3 = x_1]}_{=\frac{1}{2}} = \frac{1}{2} + \frac{1}{2}.\frac{1}{2} = \frac{3}{4}$$

$$P[z = x_3] = \underbrace{P[x_2 = 0]}_{=\frac{1}{2}} + \underbrace{P[x_2 = 1]}_{=\frac{1}{2}} . \underbrace{P[x_3 = x_1]}_{=\frac{1}{2}} = \frac{1}{2} + \frac{1}{2}.\frac{1}{2} = \frac{3}{4}$$

Geffeho generátor

Kľúč Geffe-ho generátora – štartovacia náplň registrov LFSR1, LFSR2 a LFSR3 – t.j. $(2^{L_1}-1)(2^{L_2}-1)(2^{L_3}-1)$ možností.

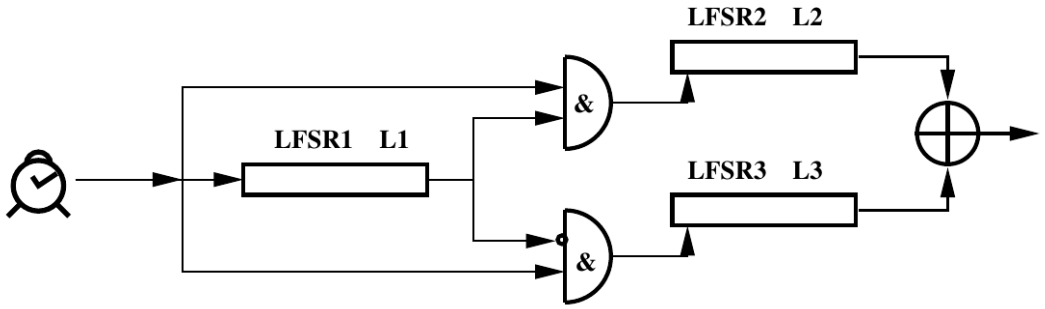
Korelačný útok:
Máme postupnosť $\mathbf{z} = z_1, z_2, \ldots, z_k, \ldots$ z výstupu generátora.

Krok 1.:
Ľubovoľne nastavíme LFSR2 a LFSR3 a postupne nastavujme LFSR1 a počítajme počet zhôd výstupu generátora s postupnosťou $\mathbf{z}$. Ak počet zhôd stôpne zhruba na $\frac{3}{4}$, bude LFSR1 nastavený tak ako na začiatku postupnosti $\mathbf{z}$.
Krok 2.:
Rovnakým spôsobom nastavíme počiatočný stav registra LFSR3.
Krok 3.:
Nakoniec dopočítame nastavenie registra LFSR2.

Namiesto $(2^{L_1}-1)(2^{L_2}-1)(2^{L_3}-1)$ možností počiatočného nastavenia registrov bude treba vyskúšať najviac $(2^{L_1}-1)+(2^{L_3}-1)$ možností.

Tento princíp je použiteľný pre akýkoľvek systém LFSR s akoukoľvek výstupnou funkciu, ak pre výstup x_i z i-teho LFSR platí $P[x_i = z] \neq \frac{1}{2}$.

Alternating step generátor



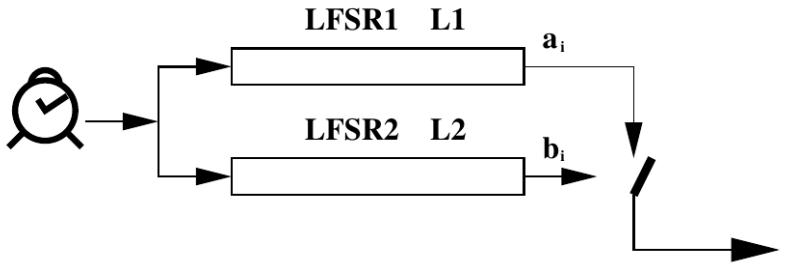
Podľa výstupu LFSR1 sa posúva práve jeden z generátorov LFSR2, LFSR3.
Ak je výstup z LFSR1 1, posunie sa generátor LFSR2, inak sa posunie LFSR3.
Ak sa LFSR1 modifikuje tak, aby po $(L_1 - 1)$ nulách vyslal ešte jednu nulu, cyklus tohoto generátora bude

$$2^{L_1}.(2^{L_2} - 1).(2^{L_3} - 1)$$

ak sú L_1, L_2, L_3 nesúdeliteľné.

PreL_1, L_2, L_3 nesúdeliteľné, $L_1 \approx L_2 \approx L_3 \approx 128$ je tento generátor bezpečný proti všetkým známym útokom.

Schrinking generátor



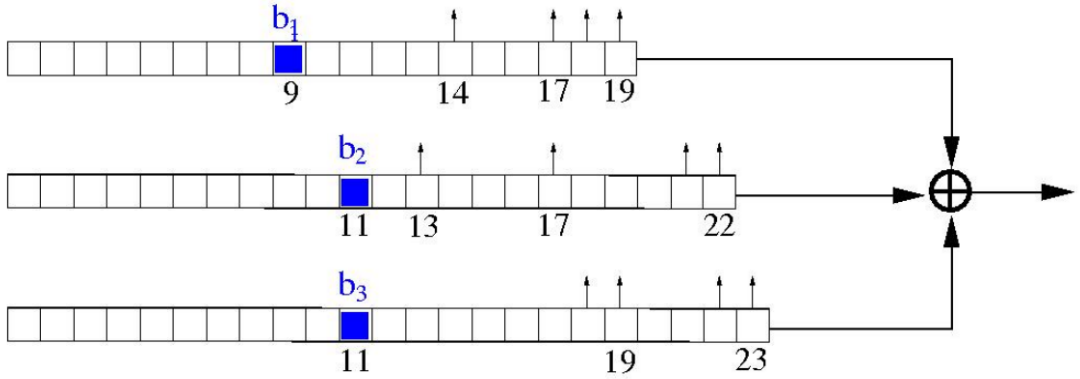
Ak $b_i = 1$, výstupom je bit a_i. Ak $b_i = 0$, zruš a_i.

Ak sú L_1, L_2 nesúdeliteľné, potom má generátor periódu

$$(2^{L_1} - 1).(2^{L_2} - 1)$$

Algoritmus GSM A5

LFSR1 – (19, 18, 17, 14, 0)
LFSR2 – (22, 21, 17, 13, 0)
LFSR3 – (23, 22, 19, 18, 0)



$$posun(i) = b_i \oplus \overline{T(b_1, b_2, b_3)}$$

$$\overline{T(b_1, b_2, b_3)} = \begin{cases} 0 & \text{ak } (b_1 + b_2 + b_3) \geq 2 \\ 1 & \text{ak } (b_1 + b_2 + b_3) \leq 1 \end{cases}$$

$$posun(i) = b_i \oplus \overline{T(b_1, b_2, b_3)}$$

Ďalšie používané generátory

Blum - Micalli generátor:
g, p dve veľké prvočísla

$$\begin{aligned} x_{i+1} &= g^{x_i} \mod p \\ b_i &= \begin{cases} 1 & \text{ak } x_i < \frac{p-1}{2} \\ 0 & \text{inak} \end{cases} \end{aligned}$$

RSA generátor:

p, q dve veľké tajné prvočísla
$N = p.q$
e nesúdeliteľné s $(p-1)(q-1)$

$$\begin{aligned} x_{i+1} &= x_i^e \mod N \\ b_i &= x_i \mod 2 \text{ (– najmenej významný bit } x_i) \end{aligned}$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Testovanie štatistických vlastností postupností

Majme postupnosť bitov

$$\mathbf{b} = b_1, b_2, \ldots, b_n$$

z nejakého generátora náhodných čísel.
Treba zistiť, či táto postupnosť je skutočne náhodná.

Nasledujúce testy umožnia vylúčiť také postupnosti, ktoré sa na šifrovanie nehodia.

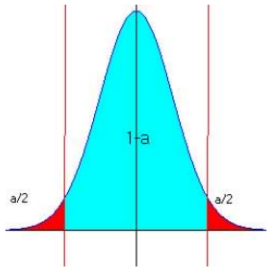
Princíp všetkých testov je nasledujúci:

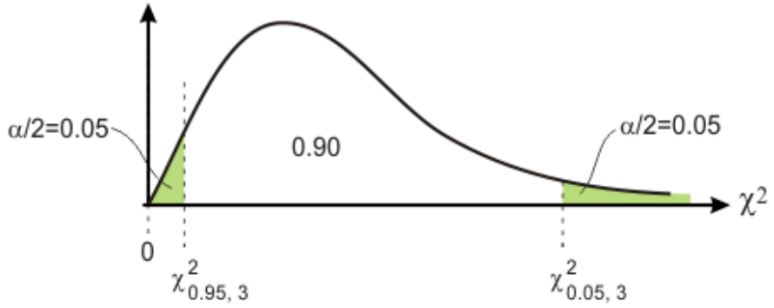
- Stanoví sa hypotéza H (napríklad "$P[b_i = 1] = P[b_i = 0] = \frac{1}{2}$" – t. j. prevdepodobnosť nuly a jedničky je rovnaká).
- Stanovíme tzv. stupeň významnost α ako pravdepodobnosť zamietnutia hypotézy H napriek tomu, že hypotéza H platí (to je tzv. chyba prvého druhu).
  Najčastejšie používané hodnoty sú $\alpha = 0.05$ a $\alpha = 0.01$.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY | KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

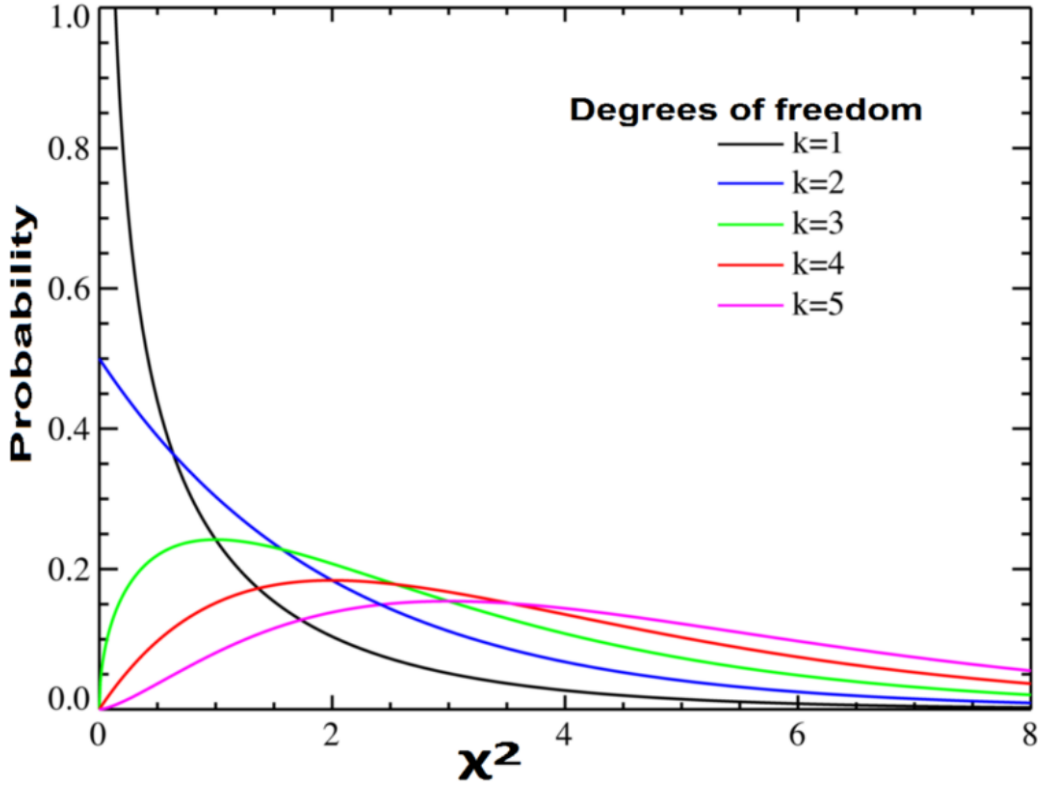
Testovanie štatistických vlastností postupností

- Určí sa náhodná veličina $X = f(b_1, b_2, \ldots, b_n)$ (nazývaná tiež štatistika), ktorá má za predpokladu platnosti hypotézy H známe rozdelenie f (najčastejšie normálne $f = N(0, 1)$ alebo $f = \chi^2(k)$ o k stupňoch voľnosti).
- Určí sa interval (a, b) – tzv. interval spoľahlivosti (confidence interval) taký, že $P[X \in (a, b)] = 1 - \alpha$.
  Oblasť na reálnej osi $(-\infty, a\rangle \cup \langle b, \infty)$ sa volá kritická oblasť.
- Ak X padne do kritickej oblasti, hypotézu H zamietame, pretože nastal neočakávaný jav.
- Ak X padne do intervalu (a, b), hypotézu H nezamietame.





Hustota rozdelenia chi-kvadrát



Frekvenčný test

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

n_0 – počet núl $\quad n_1$ – 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

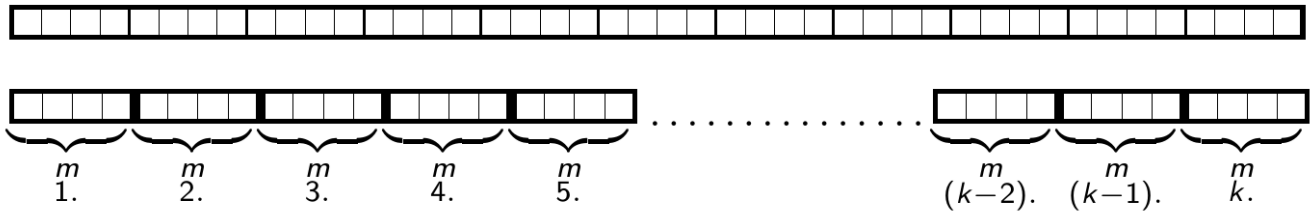
Dvojbitový sériový test
$n_{00}, n_{01}, n_{10}, n_{11}$ – počet výskytov dvojíc 00, 01, 10, 11 v postupnosti $\mathbf{b}$.

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 **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 $i = 0, 1, 2, \ldots, 2^m - 1$ 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: $\ldots 0 \underbrace{1\,1\,\ldots\,1}_{i} 0 \ldots$

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

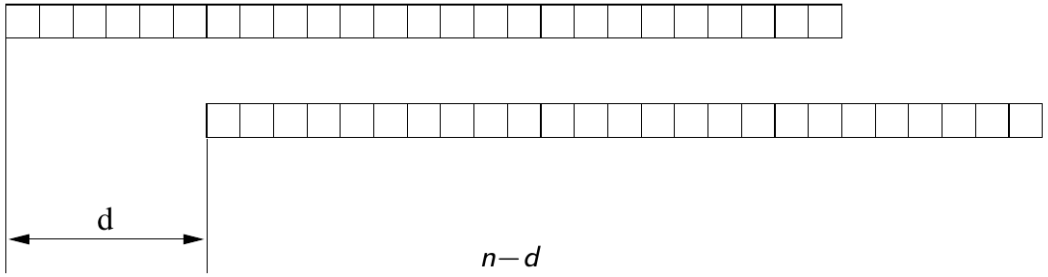
Očakávaný počet blokov dĺžky i v n-prvkovej postupnosti $\mathbf{b}$ je $e_i = \frac{n-i+3}{2^{i+2}}$.

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

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 **b** dlhý 20000 bitov.

1. Monobit test: $9654 < n_1 < 10346$
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.

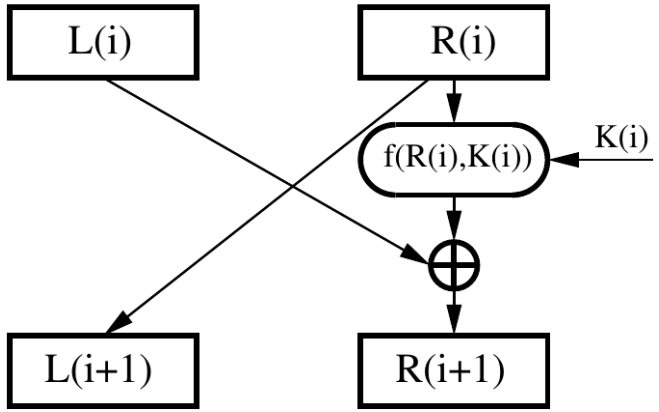
i	Dovolený rozsah B_i, G_i
1	2267 – 2733
2	1079 – 1421
3	502 – 748
4	223 – 402
5	90 – 223
6	90 – 223

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

Symetrické šifrovacie algoritmy
Blokové šifry

Kryptosystémy Feistelovho typu

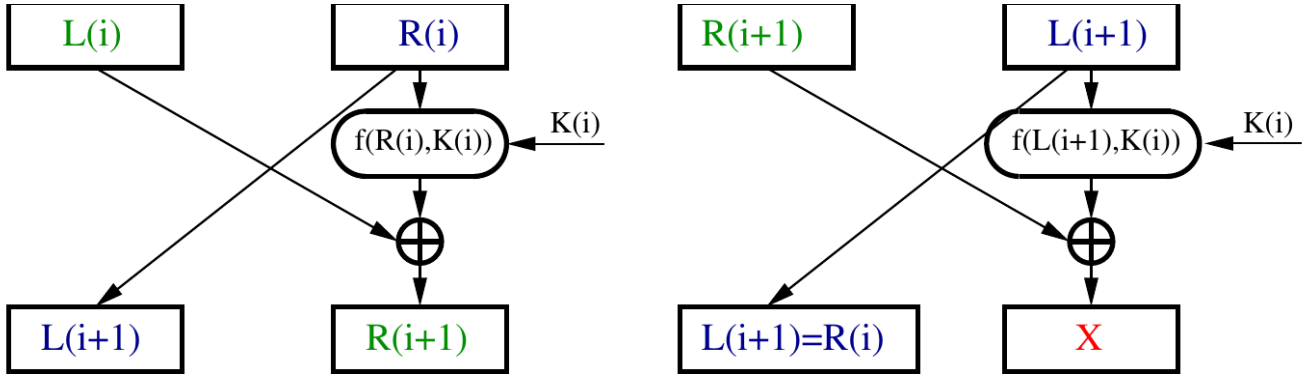
Sú to systémy s blokovou šifrou – šifrujú sa celé bloky priameho textu. Pre kryptosystémy Feistelovho typu musí mať blok párny počet bitov.
Blok sa rozdelí na dve rovnako dlhé časti – ľavú L_i a pravú R_i.



Šifrovanie prebieha po kolách
Jedno kolo urobí:

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

Kryptosystémy Feistelovho typu

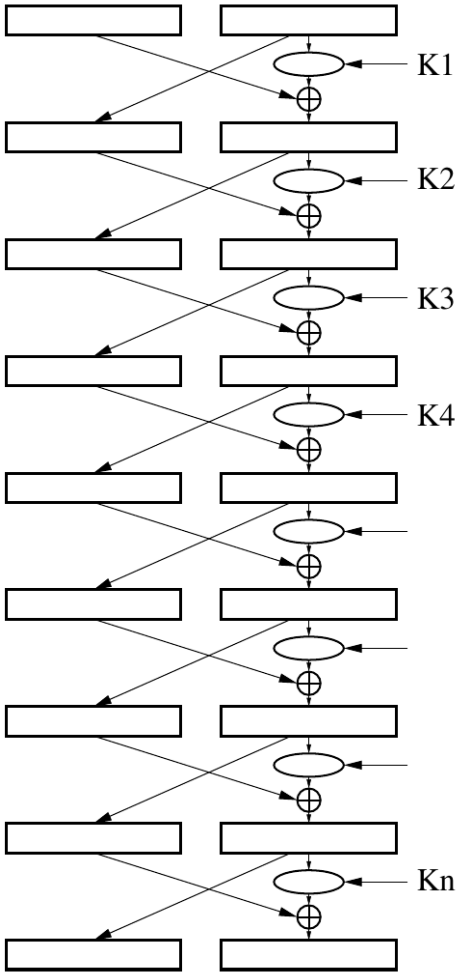


Počítajme X.

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

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.

Feistelova sieť



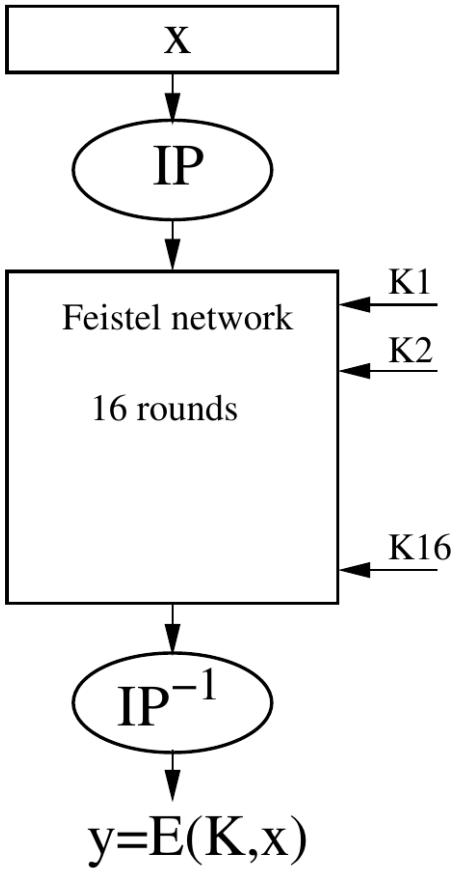
Feistelova sieť je iterované niekoľkonásobné opakovanie kolových algoritmov, každý s iným kolovým kľúčom.

Dešifrovanie sa urobí tou istou sieťou, ktorej sa na vstup vloží zašifrovaný text s poradím kolových kľúčov $K_n, K_{n-1,\ldots,K_1}$ a so zameneným poradím pravej a ľavej časti.

Dôležité: Práve popísaný inverzný mechanizmus nezáleží na tvare funkcie $f(R_i, K_i)$.

Na funkcii $f(R_i, K_i)$ však podstatne závisia kryptografické vlastnosti Feistelovej siete.

Data Encryption Standard (DES)



- 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 – vstupná a výstupná permutácia

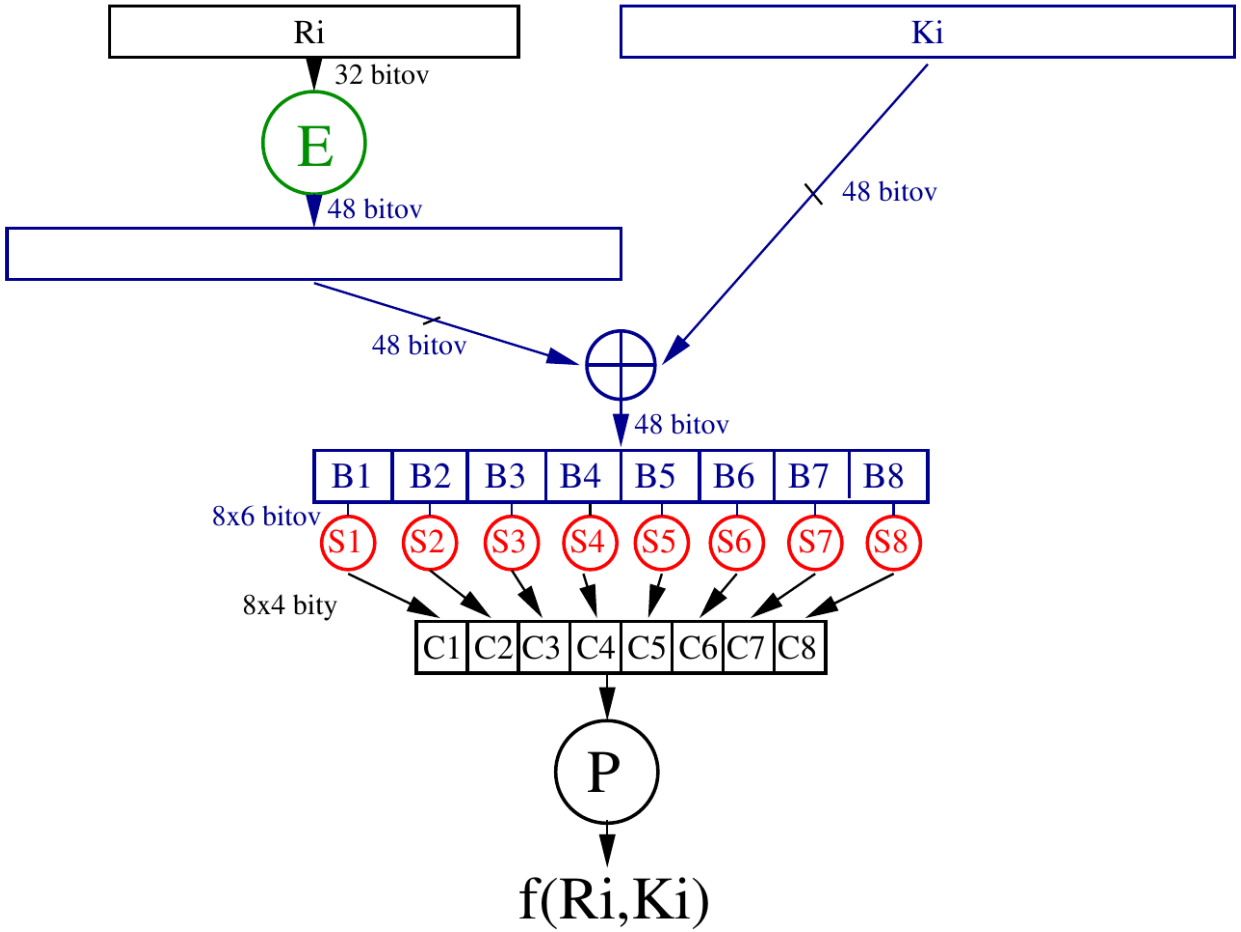
Table 12.1 Initial Permutation

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Table 12.8 Final Permutation

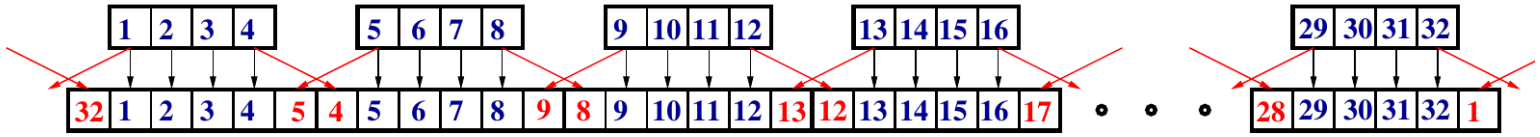
40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

DES – funkcia „f“

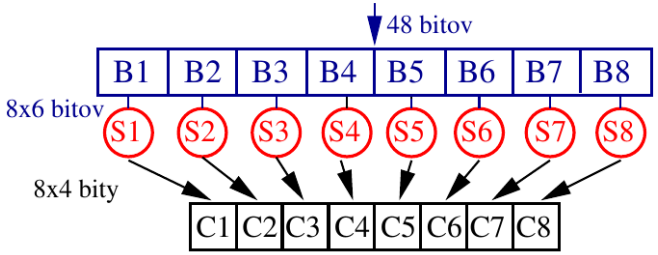


DES – expanzná operácia

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



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 $b_1 b_2 b_3 b_4 b_5 b_6$ a predstavuje adresu príslušného štvorbitového čísla C_i v S-boxe S_i.

DES – adresovanie v S-boxe

Adresa sa vypočíta takto:

Nech $B_1 = b_1 b_2 b_3 b_4 b_5 b_6$.

$b_1 b_6$ je číslo riadku, $b_2 b_3 b_4 b_5$ 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:
$B_1 = 101011$. $b_1 b_6 = (11)_2 = 3$, $b_2 b_3 b_4 b_5 = (0101)_2 = 5$.
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 – tabuľky S-boxov

S-box 2:

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

S-box 3:

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

S-box 4:

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

DES – tabuľky S-boxov

S-box 5:

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

S-box 6:

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

S-box 7:

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

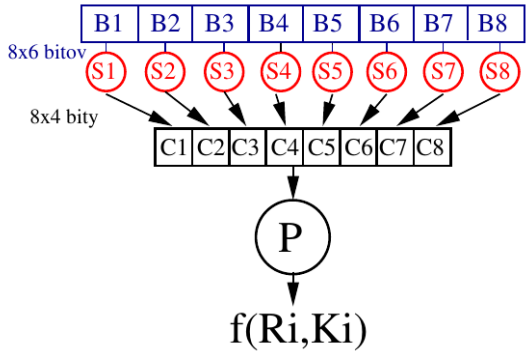
S-box 8:

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

DES – záverečná permutácia

Table 12.7 P-Box Permutation

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

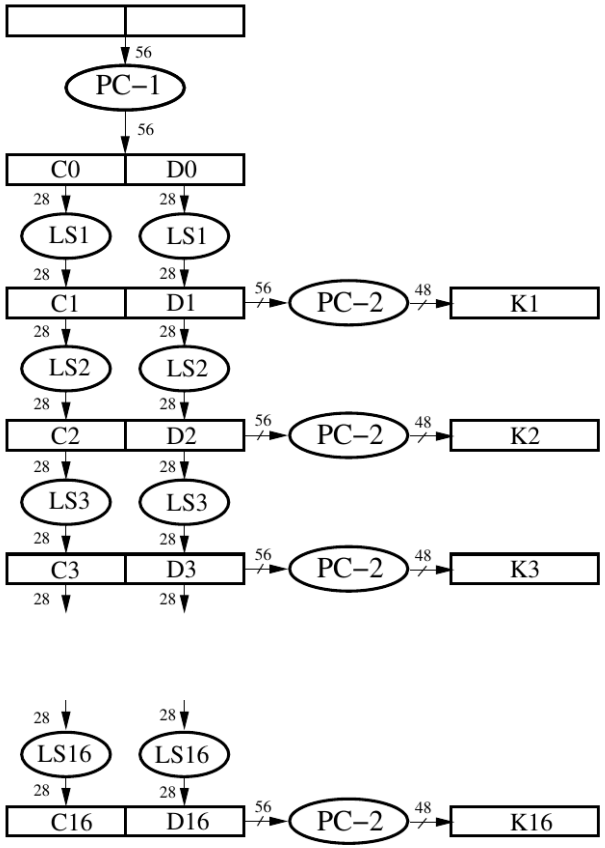


16	**7**	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	**8**	24	14
32	27	3	9
19	13	30	**6**
22	11	4	25

16	7	20	21
29	**12**	28	17
1	15	23	26
5	18	31	**10**
2	8	24	14
32	27	3	**9**
19	13	30	6
22	**11**	4	25

16	7	20	21
29	12	28	17
1	**15**	23	26
5	18	31	10
2	8	24	**14**
32	27	3	9
19	**13**	30	6
22	11	4	25

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 nepárnu 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 $LS_1, LS_2 \ldots, LS_{16}$. Pre $i = 1, 2, 9, 16$ je LS_i posun o jedno miesto, inak o 2 miesta.

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

DES – generovanie kolových kľúčov

Permutácia PC-1

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Zobrazenie PC-2

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

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.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

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.

Útoky proti DESu

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

$$\bigoplus_{i=1}^{64} a_i x_i \oplus \bigoplus_{i=1}^{64} b_i y_i = \bigoplus_{i=1}^{56} c_i k_i$$

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).

DES – pokusy o predĺženie kľúča

Namiesto jedného šifrovania kľúčom K_1 zašifrujeme dvakrát – najprv kľúčom K_1 a potom kľúčom K_2. Teda

šifrujeme: $y = E_{K_2}[E_{K_1}(x)]$ dešifrujeme: $x = D_{K_1}[D_{K_2}(y)]$

Ak by boli šifrovacie a dešifrovacie zobrazenia systému DES grupou, t.j. ak by pre K_1, K_2 existovalo K_3 také, že $E_{K_2}[E_{K_1}] = E_{K_3}$, dvojité šifrovanie by nemalo význam.

Príklady šifier, ktoré sú grupami:

- cézarovská šifra
- všeobecná monoalfabetická šifra
- permutačná šifra
- hillovská šifra

DES však nie je grupou.

DES – útok typu meet-in-the-middle

Predpokladajme že poznáme dvojicu x, y priameho textu a textu zašifrovaného dvojicou kľúčov K_1, K_2, t.j. $y = E_{K_2}[E_{K_1}(x)]$.
$D_{K_2}(y) = D_{K_2}\{E_{K_2}[E_{K_1}(x)]\} = E_{K_1}(x)$
Hľadáme takú dvojicu kľúčov K_1, K_2, pre ktoré je

$$D_{K_2}(y) = E_{K_1}(x).$$

Zostrojíme dve tabuľky – tabuľku 1. závislosti $E_{K_1}(x)$ na K_1 a tabuľku 2. závislosti $D_{K_2}(y)$ na K_2.

Ak nájdeme taký prvok v druhom stĺpci tabuľky 1., ktorý sa rovná niektorému prvku v druhom stĺpci tabuľky 2., našli sme v príslušných prvých stĺpcoch kandidátov na kľúče K_1, K_2.

K_1	$E_{K_1}(x)$
0	
1	
2	
L_1	z
$2^{56}-1$	

K_2	$D_{K_2}(y)$
0	
1	
2	
L_2	z
$2^{56}-1$	

DES – útok typu meet-in-the-middle

Postup možno zjednodušiť tak, že zostrojíme a zapamätáme len tabuľku 1. a postupne generujeme $D_{K_2}(y)$ pre $K_2 = 0, 1, \ldots$ a hľadáme jeho výskyt v druhom stĺpci tabuľky 1.

Pamäťové nároky: 2^n (2^{56}) riadkov tabuľky 1.

Výpočtové nároky:
2×2^n (2×2^{56}) kódovaní
$2^n . \log_2 2^n = n.2^n$ (56.2^{56}) krokov na usporiadanie tabuľky 1
a najviac $2^n . \log_2 2^n = n.2^n$ (56.2^{56}) krokov na vyhľadávanie v tabuľke 1.
Spolu: $2.2^n + n.2^n + n.2^n = (2 + 2n)2^n = (1 + n).2^{n+1}$ (57.2^{57}).

Sú známe aj efektívnejšie útoky.

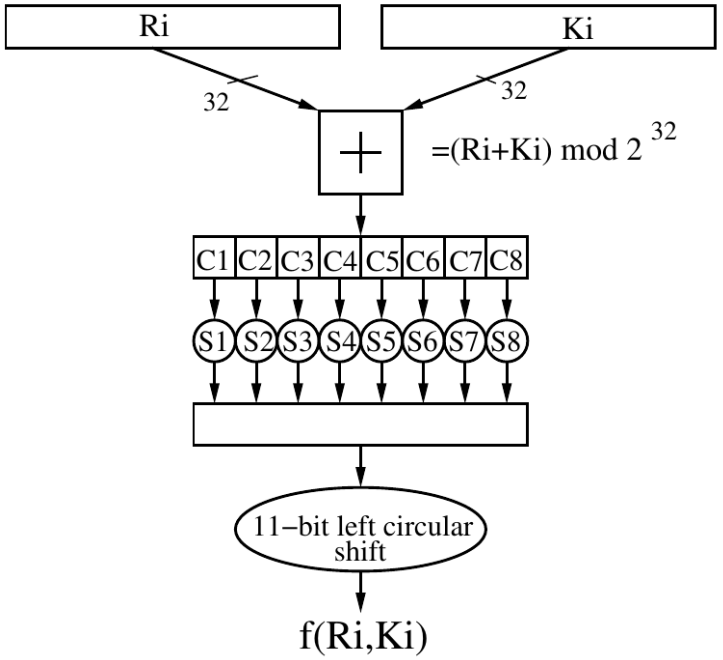
Útok hrubou silou na odhalenie kľúčov K_1, K_2 vyžaduje 2^{2n} (2^{112})kódovaní.

Dôsledok: Dvojité šifrovanie neprináša očakávané zosilnenie šifry.

DES – Tripple DES, 3DES

šifrujeme: $y = E_{K_3}\{D_{K_2}[E_{K_1}(x)]\}$ dešifrujeme: $y = D_{K_1}\{E_{K_2}[D_{K_3}(x)]\}$

Algoritmus 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 $0, 1, \ldots, 15$.

Algoritmus GOST

S-box 1:

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

S-box 2:

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

S-box 3:

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

S-box 4:

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

S-box 5:

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

S-box 6:

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

Algoritmus GOST

S-box 1:

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

S-box 2:

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

S-box 3:

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

S-box 4:

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

S-box 5:

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

S-box 6:

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

Algoritmus GOST

S-box 7:

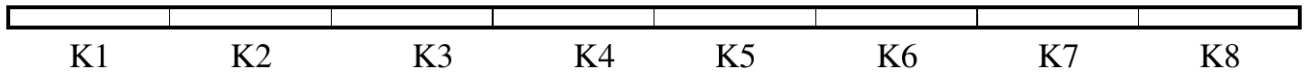
13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12

S-box 8:

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

Generovanie kolových kľúčov

Kľúč je 256 bitový. Možno ho rozdeliť na osem 32-bitových kľúčov $K_1, K_2, \ldots, K_8$.



Tieto sa potom použijú v poradí:
$K_1, K_2, \ldots, K_8, K_1, K_2, \ldots, K_8, K_1, K_2, \ldots, K_8, K_8, K_7, \ldots K_1$

Algoritmus IDEA

Xueija Lai and James Massey, 1992.

Bol patentovaný, US patent vypršal 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."

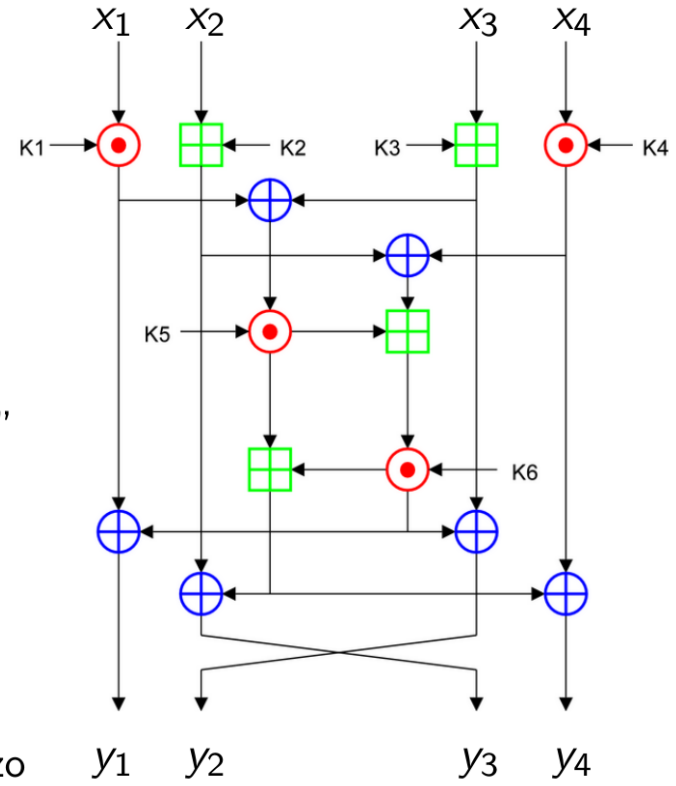
V kolách sa používajú tieto operácie:

$\oplus$ – XOR po bitoch

$\boxplus$ – sčítanie mod 2^{16}

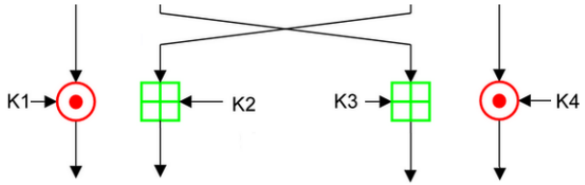
$\odot$ – násobenie mod $(2^{16}+1)$
pričom sa 16-bitové slovo pozostávajúce zo samých 0 považuje za reprezentáciu čísla 2^{16}.

Jedno kolo algoritmu IDEA



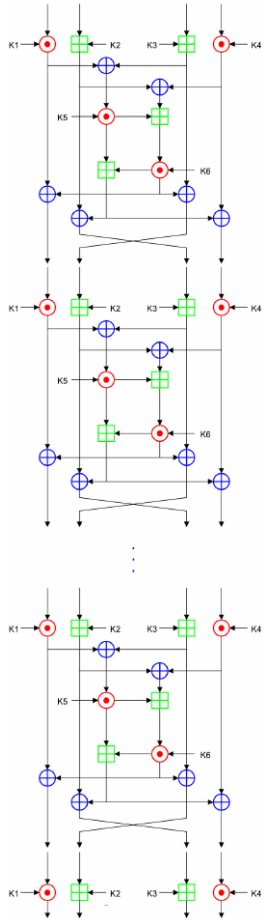
Algoritmus IDEA

Záverečné polovičné kolo



Generovanie kolových kľúčov

Každé kolo potrebuje 6 kľúčov a záverečné polovičné kolo 4 kľúče, t.j spolu $6 * 8 + 4 = 52$ 16-bitových kľúčov.
Najprv sa 128 bitový kľúč rozdelí na 8 16-bitových kľúčov.
Potom sa na kľúč aplikuje ľavý rotačný posun o 25 bitov a získa sa ďalších 8 kľúčov.
Kľúč sa znovu rotuje o 25 bitov a získa sa ďalších 8 kľúčov. Atď.



Algoritmus IDEA

Dešifrovanie

Ten istý algoritmus sa použije aj na dešifrovanie s tým, že ako kľúče sa použijú opačné resp. inverzné hodnoty kľúčov zo šifrovania vo vhodnom poradí.

Advanced Encryption Standard (AES)

- 1997 – inicializácia procesu výberu vhodného symetrického kryptografického algoritmu – NIST[1]
- Súťaže sa zúčastnilo 15 algoritmov
- 1998 publikovali Vincent Rijmen (1970) a Joan Daemen(ová) (1965) (Belgicko) algoritmus Rijndael
- Od roku 2002 bol Rijndael uznaný autoritami NIST, FIPS[2], NSA[3] za nový kryptografický štandard označovaný ako AES
- AES je jediný verejne dostupný šifrovací algoritmus schválený NSA pre najtajnejšie (top secret) informácie

Výhody:
– Výkonnosť v hardvérovej i softvérovej implementácii
– Nízke pamäťové nároky
– Možnosť ochrany pred útokmi parazitnými kanálmi

[1] NIST – National Institute of Standards and Technology
[2] FIPS – Federal Information Processing Standard
[3] NSA – National Security Agency

Algoritmus Rijndael (AES)

Galoisove pole $GF(2^8)$

Pole $GF(2^8)$ Prvky: polynómy typu

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$$

s koeficientami v $\mathbb{Z}_2$.
Takýto polynóm modeluje bajt $b_7b_6b_5b_4b_3b_2b_1b_0$. Tak napríklad $\{0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\}$ zodpovedá polynómu $x^6 + x^4 + x^2 + x + 1$.

Sčítanie v $GF(2^8)$ je sčítanie polynómov nad $\mathbb{Z}_2$.

$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x^6 + x^4 + x^2) = (x^7 + x + 1)$
$\{0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\} \oplus \{1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\}$= $\{1\ 0\ 0\ 0\ 0\ 1\ 1\}$
V hexadecimálnom zápise $(57)_H \oplus (D4)_H = (83)_H$.

Sčítaniu bajtov $\oplus$ zodpovedá počítačová bajtová operácia XOR po bitoch.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Algoritmus Rijndael (AES)

Násobenie v $GF(2^8)$ sa definuje ako

$$p(x) \otimes q(x) = p(x).q(x) \mod m(x),$$

kde $m(x)$ je ireducibilný polynóm stupňa 8 nad $GF(2^8)$.

AES používa tento ireducibliný polynóm $m(x) = x^8 + x^4 + x^3 + x + 1$.

$$\big(\underbrace{(x^6 + x^4 + x^2 + x + 1)}_{57_H=\{01010111\}} . \underbrace{(x^7 + x + 1)}_{83_H=\{10000011\}}\big) \mod \underbrace{(x^8 + x^4 + x^3 + x + 1)}_{=m(x)} =$$

$$(x^{13} + x^{11} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1) \mod m(x) =$$

$$= \underbrace{(x^7 + x^6 + 1)}_{C1_H=\{11000001\}}$$

V $GF(2^8)$ teda máme $\{01010111\} \otimes \{10000011\} = \{11000001\}$

Algoritmus Rijndael (AES)

Polynóm x zodpovedá bajtu $\{00000010\}$, t.j číslu $2 = (02)_H$.
Skúmajme, čomu sa rovná $\{00000010\} \otimes b$.

Nech
$b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$.
Potom
$x.b(x) = b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x$

Ak $b_7 = 0$, $x.b(x) \mod m(x) = x.b(x)$,
kde $m(x) = x^8 + x^4 + x^3 + x + 1$.
Túto operáciu vykoná ľavý posun bajtu b o 1 bit.

Ak $b_7 = 1$,
potom

$$x.b(x) \mod m(x) = x.b(x) \ominus m(x) = x.b(x) \oplus m(x).$$

Túto operáciu vykoná ľavý posun bajtu b a následne bitový XOR s bajtom $\{00011011\}$ (hexadecimálne $(1B)_H$).

Algoritmus Rijndael (AES)

Môžeme teda definovať funkciou
xtime(**b**)

```
1. if (b[7] == 1) t=00011011 else t=00000000;
2. for(i=7 to 1) b[i]=b[i-1];
3. b = b ⊕ t;
4. return b;
```

Potom násobenie $\mathbf{a} \otimes \mathbf{b} = \mathbf{c}$ realizujeme nasledovne:

```
1. c=00000000;
    p = a;
2. for(i=0 to 7);
      if(b[i] == 1) c = c ⊕ p;
      p=xtime(p);
3. return c;
```


Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Algoritmus Rijndael (AES)

$GF(2^8)$ s operáciami $\oplus$, $\otimes$ tvorí pole, v ktorom

- nulový prvok je polynóm $\quad 0 - 00000000$
- jednotkový prvok je prvok $1 - 00000001 \equiv 0x^7 + 0x^6 + \cdots + 0x + 1$
- ku každému prvku b existuje opačný prvok – je to samotné b,
- ku každému prvku $b \neq 0$ existuje inverzný prvok b^{-1}.

Inverzný prvok možno vypočítať rozšíreným Euklidovým algoritmom. Pre účely AES však stačí vypočítať tabuľku binárnej operácie $\otimes$ (má rozmer 256×256) a pre každé $b = 1, 2, \ldots, 255$ nájsť to c, pre ktoré je $b \otimes c = 1$, a položiť $b^{-1} = c$.

Ak vytvoríme tabuľku s 256 položkami typu

0	1	2^{-1}	3^{-1}	... ...	255^{-1}

inverzný prvok b^{-1} k prvku b získame ako položku tejto tabuľky na mieste (adrese) b.

Algoritmus Rijndael (AES)

- Bloková šifra
- Dĺžka bloku: 128 bitov
- Dĺžka kľúča: voliteľne 128, 192 alebo 256 bitov

128-bitový blok priameho textu berieme ako postupnosť 16 8-bitových bajtov:

$$a_{00}\,a_{10}\,a_{20}\,a_{30}\,a_{01}\,a_{11}\,a_{21}\,a_{31}\,a_{02}\,a_{12}\,a_{22}\,a_{32}\,a_{03}\,a_{13}\,a_{23}\,a_{33}$$

Tieto sa usporiadajú do tabuľky,
ktorý sa volý stav

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

Stav

k_{00}	k_{01}	k_{02}	k_{03}
k_{10}	k_{11}	k_{12}	k_{13}
k_{20}	k_{21}	k_{22}	k_{23}
k_{30}	k_{31}	k_{32}	k_{33}

Kolový kľúč

S týmto stavom sa iteračne vykonáva niekoľko kôl operácií, niektoré z nich závisia na kolovom kľúči, reprezentovanom ako matica bajtov

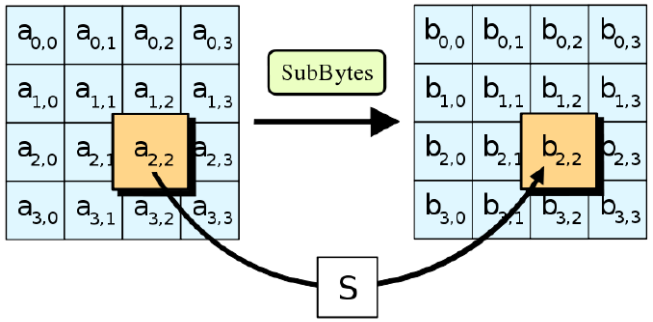
Rijndael – operácia SubByte

S každým bajtom a tabuľky Stav sa vykonajú dve operácie:

1. Najprv sa k hodnote a najde v poli $GF(2^8)$ inverzný prvok $x = a^{-1}$, ak $a \neq 0$.

   Ak $a = 0$, položíme $x = 0$.

2. Potom sa vypočíta byte $b = b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7$



$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

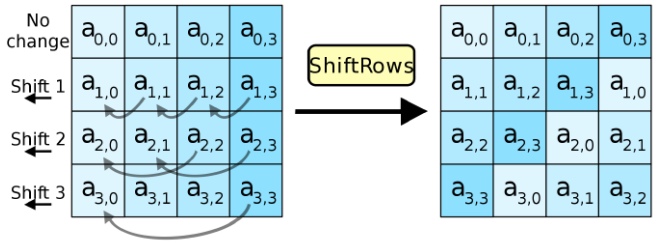
Rijndael – operácia SubByte

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Rijndael – operácia ShiftRows

Na riadky tabuľky Stav sa aplikujú nasledujúce ľavé rotačné posuny

1. 1. riadok ostáva bez zmeny
2. 2. riadok - posun o 1 bajt - t.j 8 bitov
3. 3. riadok - posun o 2 bajty - t.j 16 bitov
4. 4. riadok - posun o 3 bajty - t.j 24 bitov

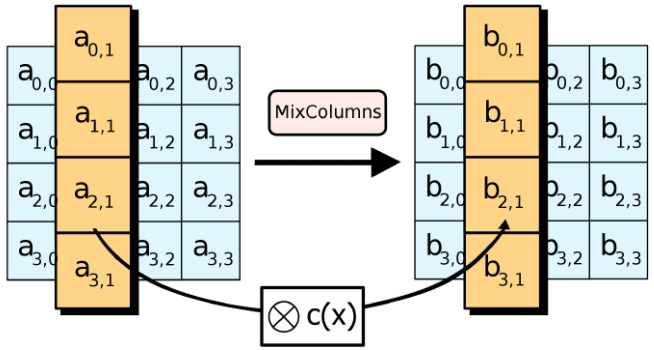


Rijndael – operácia MixColumns

Pri tejto operácii považujeme maticu Stav za maticu prvkov poľa $GF(2^8)$.
S každým jej stĺpcom $\mathbf{a}_i = \begin{bmatrix} a_{0i} & a_{1i} & a_{2i} & a_{3i} \end{bmatrix}^T$ vykonáme

$$\underbrace{\begin{bmatrix} b_{0i} \\ b_{1i} \\ b_{2i} \\ b_{3i} \end{bmatrix}}_{\mathbf{b}_i} = \underbrace{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}}_{\mathbf{M}} \underset{GF(2^8)}{\otimes} \underbrace{\begin{bmatrix} a_{0i} \\ a_{1i} \\ a_{2i} \\ a_{3i} \end{bmatrix}}_{\mathbf{a}_i} \quad \text{t. j.} \quad \mathbf{b}_i = \mathbf{M} \otimes \mathbf{a}_i$$

V maticovom tvare: $\mathbf{B} = \mathbf{M}.\mathbf{A}$



$$\mathbf{M}^{-1} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix}$$

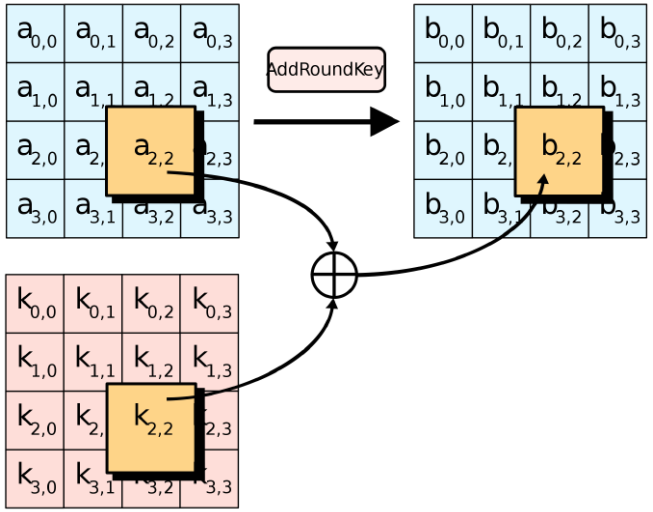
Rijndael – operácia AddRoundKey

V tomto kole sa pre každý prvok a_{ij} Stavu vykoná

$$b_{ij} = a_{ij} \oplus k_{ij},$$

kde k_{ij} je prvok matice príslušného kolového kľúča.
V maticovom tvare:

$$\mathbf{B} = \mathbf{A} \oplus \mathbf{K}.$$



Rijndael – šifrovací algoritmus

1. Inicializačné kolo
   - 1.1 AddRoundKey
2. for $Round = 1$ to $N_r - 1$
   - 2.1 SubBytes
   - 2.2 ShiftRows
   - 2.3 MixColumns
   - 2.4 AddRoundKey
3. Záverečné kolo (bez MixColumns)
   - 3.1 SubBytes
   - 3.2 ShiftRows
   - 3.3 AddRoundKey

Dĺžka kľúča	128	192	256
Počet kôl N_r	10	12	14

Rijndael – dešifrovací algoritmus

Malo by byť:

1. Inicializačné kolo
   - 1.1 AddRoundKey
   - 1.2 InvShiftRows
   - 1.3 InvSubBytes
2. for $Round = 1$ to $N_r - 1$
   - 2.1 AddRoundKey
   - 2.2 InvMixColumns
   - 2.3 InvShiftRows
   - 2.4 InvSubBytes
3. Záverečné kolo
   - 3.3 AddRoundKey

Je:

1. Inicializačné kolo
   - 1.1 AddRoundKey
2. for $Round = 1$ to $N_r - 1$
   - 2.1 InvSubBytes
   - 2.2 InvShiftRows
   - 2.3 InvMixColumns
   - 2.4 AddRoundKeyX
3. Záverečné kolo
   - 3.1 InvSubBytes
   - 3.2 InvShiftRows
   - 3.3 AddRoundKey

Poradie operácií InvShiftRows a InvSubBytes je zameniteľné.

$$AddRoundKey(InvMixcolumns(\mathbf{B})) = \mathbf{K} \oplus \mathbf{M}^{-1}.\mathbf{B}.$$
$$InvMixcolumns(AddRoundKey(\mathbf{B})) = \mathbf{M}^{-1}.(\mathbf{K} \oplus \mathbf{B}) = \mathbf{M}^{-1}\mathbf{K} \oplus \mathbf{M}^{-1}\mathbf{B}.$$

Rijndael – expanzia kolových kľúčov

Príklad pre 128 bitový kľúč

$\mathbf{W}_0$	$\mathbf{W}_1$	$\mathbf{W}_2$	$\mathbf{W}_3$	$\mathbf{W}_4$	$\mathbf{W}_5$	$\mathbf{W}_6$	$\mathbf{W}_7$	$\mathbf{W}_8$	$\mathbf{W}_9$	$\mathbf{W}_{10}$	$\mathbf{W}_{11}$
k_{00}	k_{01}	k_{02}	k_{03}								
k_{10}	k_{11}	k_{12}	k_{13}								
k_{20}	k_{21}	k_{22}	k_{23}								
k_{30}	k_{31}	k_{32}	k_{33}								
1. kolový kľúč				2. kolový kľúč				3. kolový kľúč			

$$\mathbf{W}_i = \begin{cases} \mathbf{W}_{i-4} \oplus W_{i-1} & \text{ak } i \text{ nie je delitené 4} \\ \mathbf{W}_{i-4} \oplus SubByte(RotByte(\mathbf{W}_{i-1})) \oplus Rcon(i/4) & \text{ak } i \text{ je delitené 4} \end{cases}$$

$$\begin{aligned} Rcon(i) &= [\{x^{i-1}\}\{00\}\{00\}\{00\}] \\ RotByte[w_1, w_2, w_3, w_4] &= [w_2, w_3, w_4, w_1] \end{aligned}$$

Rijndael – expanzia kolových kľúčov

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
  word temp
  i = 0
  while (i < Nk)
    w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
    i = i+1
  end while
  i = Nk
  while (i < Nb * (Nr+1)]
    temp = w[i-1]
    if (i mod Nk = 0)
        temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
    else if (Nk > 6 and i mod Nk = 4)
        temp = SubWord(temp)
    end if
    w[i] = w[i-Nk] xor temp
    i = i + 1
  end while
end
```

Nb – = 4 – počet stĺpcov matice Stav
Nk – = 4, 6 resp. 8 pre 128-, 192- resp. 256-bitový kľúč
(počet 32-bitových slov kľúča)
Nr – = 10, 12, resp. 16 pre 128-, 192- resp. 256-bitový kľúč – počet kôl

Operačné módy blokových šifier

Majme blokovú šifru so šifrovacím zobrazením $y = E_K(x)$ a dešifrovacím zobrazením $x = D_K(y)$.
Máme priamy text vyjadrený ako postupnosť blokov

$$x_1, x_2, \ldots, x_n$$

Je niekoľko spôsobov, ako vytvoriť zodpovedajúcu postupnosť blokov zašifrovaného textu

$$y_1, y_2, \ldots, y_n$$

s použitím zobrazenia $E_K(x)$ tak, aby sa pomocou dešifrovacieho zobrazenia dala zrekonštruovať pôvodná postupnosť

$$x_1, x_2, \ldots, x_n$$

Tieto spôsoby sa nazývajú operačné módy blokových šifier.

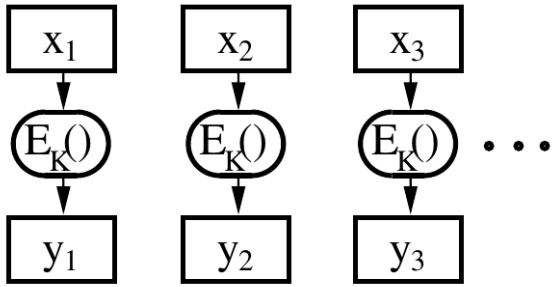
ECB – Electronic Code Book

Najjednoduchším módom je ECB mód, kedy sa šifruje priamy text blok po bloku predpisom

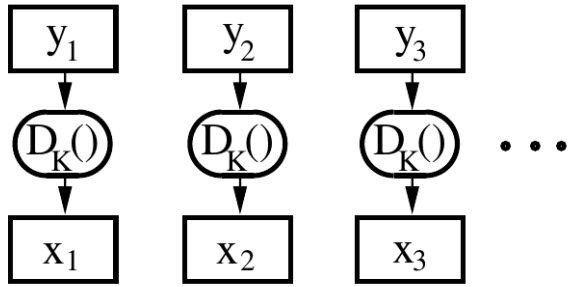
$$y_i = E_K(x_i)$$

a dešifruje predpisom

$$x_i = D_K(y_i)$$



Šifrovanie v ECB móde



Dešifrovanie v ECB móde

Nevýhoda: Rovnaký blok x_i sa zakaždým zašifruje na rovnaký blok y_i, čo môže uľahčiť niektoré útoky.

OFB – Output Feedback

Pri tomto móde sa najprv zvolí náhodný inicializačný blok IV zvaný tiež inicializačný vektor a položí sa $y_0 = IV$. Postupne sa vypočítajú $z_1 = E_K(y_0)$, a rekurentne $z_{i+1} = E_K(z_i)$.

Šifrujeme predpisom

$$y_i = z_i \oplus x_i$$

Zašifrovaná správa je postupnosť $y_0, y_1, y_2, \ldots, y_n$ (má o jeden blok viacej) a dešifrujeme predpisom

$$x_i = z_i \oplus y_i.$$

Tento mód pripomína prúdovú šifru s prúdom kľúčov $z_1, z_2, \ldots, z_n$, preto je nutné pre každú správu používať iný inicializačný vektor.

CBC – Cipher Block Chaining

Šifrujeme predpisom

$$y_i = E_K(x_i \oplus y_{i-1})$$

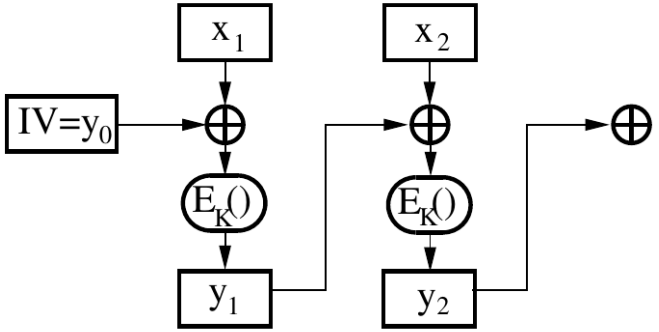
Zašifrovaná správa je postupnosť

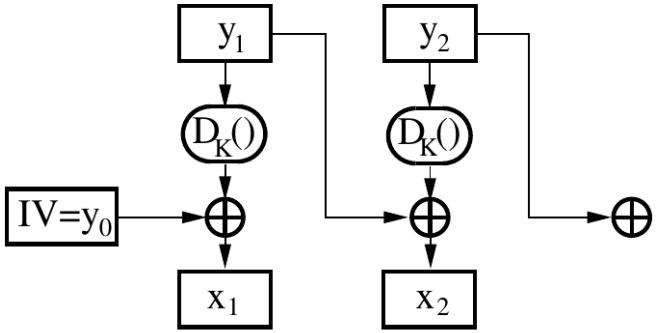
$$y_0, y_1, y_2, \ldots, y_n$$

(má o jeden blok viacej).

Dešifrujeme predpisom

$$x_i = y_{i-1} \oplus D_K(y_i).$$

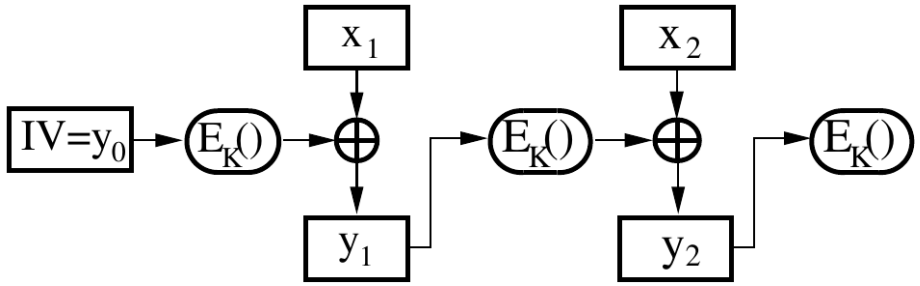




CFB – Cipher Feedback

Šifrujeme predpisom

$$y_i = E_K(y_{i-1}) \oplus x_i$$



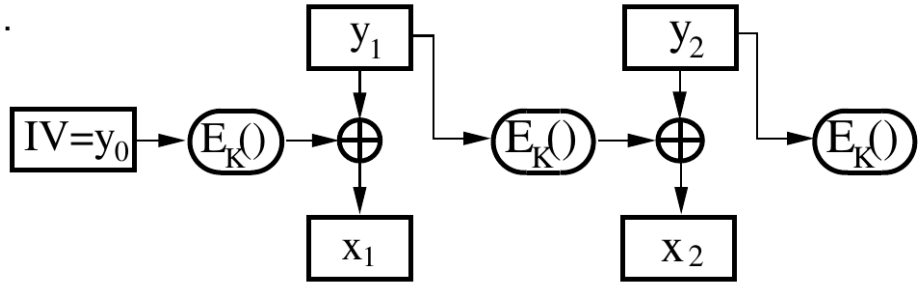
Zašifrovaná správa je postupnosť

$$y_0, y_1, y_2, \ldots, y_n$$

(má o jeden blok viacej).

Dešifrujeme predpisom

$$x_i = y_i \oplus E_K(y_{i-1}).$$



Asymetrické šifrovacie algoritmy

Jednocestné funkcie

Máme rovnicu v obore reálnych čísel s neznámou x

$$2^x = a.$$

Jej riešením je $x = \log_2(a)$.

Podobne riešenie rovnice $z^x = a$, kde $z > 0$ a x neznáma, je $x = \log_z(a)$.

Diskrétny logaritmus

Poznám a, prvočíslo p a číslo $1 < s < p$. Na aké x musím umocniť s, aby

$$s^x = a \mod p?$$

Iná formulácia. Aké je riešenie rovnice

$$s^x = a$$

v poli $\mathbb{Z}^p$?

Tento problém sa nazýva problém diskrétneho logaritmu.
Pre veľké p je hľadanie diskrétneho logaritmu ťažká úloha.

Diffie – Hellmanova výmena kľúčov

A a **B** sa dohodnú na veľkom prvočísle a čísle s, $1 < s < p$.

Čísla s, p môžu byť verejné, použiteľné opakovane aj pre viac používateľov.

A	**B**
• Zvolí $a < p$ tajné.	• Zvolí $b < p$ tajné.
• Vypočíta $\alpha = s^a \mod p$.	• Vypočíta $\beta = s^b \mod p$.
• Odošle α.	• Odošle β.
• Prijme β.	• Prijme α.
• Vypočíta kľúč $K_A = \beta^a \mod p$	• Vypočíta kľúč $K_B = \alpha^b \mod p$

Je $K_A = K_B$?

Platí:

$$K_A = \beta^\alpha = \left(s^b\right)^a = s^{ab} = (s^a)^b = \alpha^b = K_B \mod p$$

DH výmena kľúčov – intruder-in-the-middle útok

Nebezpečenstvo: Intruder in the middle attacack

$$\mathbf{A} \xrightarrow{\alpha=s^a} \mathbf{X} \xrightarrow{\alpha'=s^{a'}} \mathbf{B}$$

$$\mathbf{A} \xleftarrow{\beta'=s^{b'}} \mathbf{X} \xleftarrow{\beta=s^b} \mathbf{B}$$

$$\mathbf{A} \overset{K_1=s^{ab'}}{\longleftrightarrow} \mathbf{X} \overset{K_2=s^{a'b}}{\longleftrightarrow} \mathbf{B}$$

Úvod do teórie čísel

$\mathbb{N} = \{1, 2, 3, \dots\}$ – množina prirodzených čísel

$\mathbb{Z} = \{0, +1, -1, +2, -2, +3, -3, \dots, \}$ – množina celých čísel

Hovoríme, že celé číslo a delí celé číslo b a píšeme $a|b$, ak existuje celé číslo k také, že $b = k.a$.

Poznámka:
Platí $\forall a \in \mathbb{Z} \;\; 0 = 0.a$. Preto každé celé číslo a delí nulu, t. j. $a|0$.
0 nedelí žiadne nenulové číslo.

Relácia $.|.$ je tranzitívna – ak $a|b$ a $b|c$ potom $a|c$.
$b = k_1.a$, $c = k_2.b \Rightarrow$ potom $c = k_2.b = k_2(k_1.a)) = (k_1.k_2).a$

Nech $m \in \mathbb{Z}$. Triviálne delitele čísla m sú čísla 1, -1, m, $-m$.

Číslo $m \in \mathbb{Z}$ nazveme prvočíslom, ak má len triviálne delitele. Inak je m zložené číslo.

Úvod do teórie čísel

Každé prirodzené číslo $m > 1$ sa dá jednoznačne napísať v tvare

$$m = p_1^{\alpha_1}.p_2^{\alpha_2}.\ldots.p_k^{\alpha_k},$$

kde $p_1, p_2, \ldots, p_k$ sú navzájom rôzne prvočísla a $\alpha_1, \alpha_2, \ldots, \alpha_k$ sú prirodzené čísla.

Zisťovanie prvočíselnosti:
Eratostenovo sito

~~1~~ **2** 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~
~~1~~ **2** **3** ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~
~~1~~ **2** **3** ~~4~~ **5** ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~
~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~
~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ **11** ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~
~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ **11** ~~12~~ **13** ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~
~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ **11** ~~12~~ **13** ~~14~~ ~~15~~ ~~16~~ **17** ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~
~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ **11** ~~12~~ **13** ~~14~~ ~~15~~ ~~16~~ **17** ~~18~~ **19** ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY | KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Úvod do teórie čísel

Iný spôsob zisťovania prvočíslenosti čísla n je založený na skutočnosti, že ak $n = p.q$, kde $p > 1$, $q > 1$ a $p \leq q$, potom $p \leq \sqrt{n}$.

Na zistenie takého p možno použiť niektorý z postupov:

Postupne vydeľ číslo n číslami $2, 3, \ldots, [\sqrt{n}]$.

Alebo:
Vydeľ číslo n číslami $2, 3$ a potom postupne všetkými číslami tvaru $6k - 1,\ 6k + 1$ menšími než $\sqrt{n}$.

Alebo:
Vydeľ číslo n postupne všetkými prvočíslami menšími než $\sqrt{n}$.

Najväčší spoločný deliteľ

Hovoríme, že prirodzené číslo $d \in \mathbb{N}$ je najväčším spoločným deliteľom celých čísel $a \in \mathbb{Z}$, $b \in \mathbb{Z}$ a píšeme $d = NSD(a, b)$, ak platí

1. $d|a$ a tiež $d|b$.
2. Ak $d_1 \neq d$ a $d_1|a$, $d_1|b$, potom aj $d_1|d$.

Euklidov algoritmus pre výpočet $NSD(a, b)$ $\qquad r_0 = a,\ r_1 = b$

$$
\begin{aligned}
r_0 &= r_1.q_1 + r_2, && r_2 < r_1 \\
r_1 &= r_2.q_2 + r_3, && r_3 < r_2 \\
&\dots \\
r_{i-1} &= r_i.q_i + r_{i+1}, && r_{i+1} < r_i \\
&\dots \\
r_{m-1} &= r_m.q_m + 0
\end{aligned}
$$

$$r_m = NSD(r_0, r_1) = NSD(a, b)$$

Kongruencie

Hovoríme, že a je kongruenté s b modulo n a píšeme $a \equiv b \mod n$, ak $n|(a-b)$, t.j. ak rozdiel $(a-b)$ je deliteľný číslom n.

Platí: Relácia $\equiv$ je reláciou ekvivalencie na množine $\mathbb{Z}$ (resp $\mathbb{N}$) – relácia $\equiv$ je reflexívna, symetrická a tranzitívna.

1. $a \equiv a \mod n \; \forall a \in \mathbb{Z}$
2. Ak $a \equiv b \mod n$ potom aj $b \equiv a \mod n$
3. Ak $a \equiv b \mod n$, $b \equiv c \mod n$, potom $a \equiv c \mod n$

$a \equiv b \mod n$ platí práve vtedy, keď obe čísla a, b dávajú po delení číslom n ten istý zvyšok.

Ak $a \equiv b \mod n$, potom $a * c \equiv b * c \mod n$ pre ľubovoľné celé číslo c.

Ak $a \equiv b \mod n$, $c \equiv d \mod n$, potom $a + c \equiv b + d \mod n$.

Ak $a * c \equiv b * c \mod n$ a $NSD(c, n) = 1$, potom $a \equiv b \mod n$.

Rozšírený Euklidov algoritmus

$$
\begin{aligned}
r_0 &= r_1.q_1 + r_2 & t_2 &= (-q_1) \mod r_0 \\
r_1 &= r_2.q_2 + r_3 & t_3 &= (1 - q_2.t_2) \mod r_0 \\
r_2 &= r_3.q_3 + r_4 & t_4 &= (t_2 - q_3.t_3) \mod r_0 \\
&\dots \\
r_{i-1} &= r_i.q_i + r_{i+1} & t_{i+1} &= (t_{i-1} - q_i.t_i) \mod r_0 \\
r_i &= r_{i+1}.q_{i+1} + r_{i+2} \\
&\dots \\
r_{m-3} &= r_{m-2}.q_{m-2} + r_{m-1} & t_{m-1} &= (t_{m-3} - q_{m-2}.t_{m-2}) \mod r_0 \\
r_{m-2} &= r_{m-1}.q_{m-1} + r_m & t_m &= (t_{m-2} - q_{m-1}.t_{m-1}) \mod r_0 \\
r_{m-1} &= r_m.q_m + 0 & t_{m+1} &= (t_{m-1} - q_m.t_m) \mod r_0
\end{aligned}
$$

Rozšírený Euklidov algoritmus

Tvrdenie: $t_m r_1 \equiv r_m \mod r_0$ ($t_m b \equiv NSD(a,b) \mod a$).
Dokážeme indukciou pre $i = 2, 3, \ldots, m$ $t_i r_1 \equiv r_i \mod r_0$.

Pre $i = 2$:
Keďže $r_0 = r_1.q_1 + r_2$ je $r_2 = r_0 - r_1.q_1$.
Ďalej je $t_2 = (-q_1) \mod r_0$ čo je ekvivalentné s $q_1 + t_2 \equiv 0 \mod r_0$.

$$r_2 - t_2 r_1 \equiv r_0 - r_1 q_1 - t_2 r_1 \equiv r_0 - r_1 \underbrace{(q_1 + t_2)}_{\equiv 0 \mod r_0} \equiv 0 \mod r_0$$

Pre $i = 3$:

$$r_3 - t_3 r_1 \equiv r_1 - r_2 q_2 - t_3 r_1 \equiv r_1 - r_2 q_2 - (1 - q_2 t_2) r_1 \equiv q_2 t_2 r_1 - r_2 q_2 =$$
$$q_2 \underbrace{(t_2 r_1 - r_2)}_{\equiv 0 \mod r_0} \equiv 0 \mod r_0$$

Predpokladajme že: $t_i r_1 \equiv r_i \mod r_0$, $t_{i-1} r_1 \equiv r_{i-1} \mod r_0$.
Použijeme rekurzívne vzťahy $r_{i+1} = r_{i-1} - r_i q_i$, $t_{i+1} = t_{i-1} - q_i.t_i$

$$r_{i+1} - t_{i+1} r_1 \equiv r_{i-1} - r_i q_i - (t_{i-1} - q_i.t_i) r_1 \equiv r_{i-1} - r_i q_i - t_{i-1} r_1 + q_i.t_i r_1 \equiv$$
$$\underbrace{r_{i-1} - t_{i-1} r_1}_{\equiv 0 \mod r_0} + q_i \underbrace{(t_i r_1 - r_i)}_{\equiv 0 \mod r_0} \equiv 0 \mod r_0$$

Rozšírený Euklidov algoritmus

```
#include <stdio.h>
#include <string.h>
int main()
{int a,b,i,nsd,inv,q[100],r[100],t[100];
 printf("Zadaj a: \n");
 scanf("%d", &a);
 printf("Zadaj b:\n ");
 scanf("%d", &b);
 for(i=0;i<100;i++) r[i]=0,q[i]=0,t[i]=0;
 i=0; r[0]=a, r[1]=b, t[0]=0, t[1]=1;

 while (r[i+1]!=0)
    {q[i+1]=r[i]/r[i+1];
     r[i+2]=r[i]%r[i+1];
     t[i+2]=(t[i]-q[i+1]*t[i+1])%a;
     if(t[i+2]<0)t[i+2]=t[i+2]+a;
     i++;}

 nsd=r[i], inv=t[i];
 printf("nsd(%d,%d) = %d \n", a, b, nsd);
 printf("(%d)^ -1 mod %d = %d \n", b, a, inv);
 return 0;
}
```

Eulerova funkcia

Definícia. Nech $n \in \mathbb{N}$. Eulerova funkcia $\phi(n)$ je počet prirodzených čísel menších alebo rovných než n nesúdeliteľných s n.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$\phi(n)$	1	1	2	2	4	2	6	4	6	4	10	4	12	...

Ak p je prvočíslo, potom všetky čísla $1, 2, \ldots, p-1$ sú neúdeliteľné s p.

Ak p je prvočíslo, potom všetky súdeliteľné čisla s p^n menšie alebo rovné než p sú $1p, 2p, 3p, \ldots p^{n-1}.p$ – je ich presne p^{n-1}.

Tvrdenie. Nech $p \in \mathbb{N}$ je prvočíslo, $n \in \mathbb{N}$, $n \geq 1$. Potom platí:

$$\begin{aligned} \phi(p) &= p-1 \\ \phi(p^n) &= p^n - p^{n-1} = p^n\left(1-\frac{1}{p}\right) \end{aligned}$$

Vlastnosti Eulerovej funkcie

Tvrdenie. Nech $a,\ b \in \mathbb{N}$, a, b nesúdeliteľné. Potom

$$\phi(a.b) = \phi(a).\phi(b).$$

Dôsledok.

$$\phi(n) = \phi(\underbrace{p_1^{\alpha_1}.p_2^{\alpha_2}.....p_k^{\alpha_k}}_{=n}) = \phi(p_1^{\alpha_1}).\phi(p_2^{\alpha_2}).....\phi(p_k^{\alpha_k}) =$$

$$(p_1^{\alpha_1} - p_1^{\alpha_1-1}).(p_2^{\alpha_2} - p_2^{\alpha_2-1}).....(p_k^{\alpha_k} - p_k^{\alpha_k-1}) =$$

$$p_1^{\alpha_1}\left(1 - \frac{1}{p_1}\right).p_2^{\alpha_2}\left(1 - \frac{1}{p_2}\right).....p_k^{\alpha_k}\left(1 - \frac{1}{p_k}\right) =$$

$$\underbrace{p_1^{\alpha_1}.p_2^{\alpha_2}.....p_k^{\alpha_k}}_{=n}\left(1 - \frac{1}{p_1}\right).\left(1 - \frac{1}{p_2}\right).....\left(1 - \frac{1}{p_k}\right) =$$

$$n.\left(1 - \frac{1}{p_1}\right).\left(1 - \frac{1}{p_2}\right).....\left(1 - \frac{1}{p_k}\right)$$

Špeciálne: Pre $p,\ q \in \mathbb{N}$ obe prvočísla je $\phi(p.q) = (p-1).(q-1)$.

Binomická veta

Binomická veta:

$$(a+b)^p = a^p + \underbrace{\binom{p}{1}a^{p-1}b^1 + \binom{p}{2}a^{p-2}b^2 + \cdots + \binom{p}{i}a^{p-i}b^i + \cdots + \binom{p}{p-1}a^1 b^{p-1}}_{\text{ak je } p \text{ prvočíslo, tento súčet je deliteľný } p} + b^p$$

Ak je p prvočíslo, $1 \leq i < p$, potom

$$\binom{p}{i} = \frac{p(p-1)\ldots(p-i+1)}{1.2.\ldots.i} = p.\underbrace{\left[\frac{(p-1)\ldots(p-i+1)}{1.2.\ldots.i}\right]}_{\text{toto je celé číslo, lebo } p \text{ sa nemá s čím skrátiť}} = p.k$$

Dôsledok. Ak je p prvočíslo,

$$(a+b)^p \equiv a^p + b^p \mod p.$$

Malá Fermatova veta

Nech p je prvočíslo.

$$\begin{aligned}
2^p &= (1+1)^p \equiv 1^p + 1^p \equiv 2 \mod p \\
3^p &= (2+1)^p \equiv 2^p + 1^p \equiv 3 \mod p \\
4^p &= (3+1)^p \equiv 3^p + 1^p \equiv 4 \mod p \\
&\dots \\
c^p &= ((c-1)+1)^p \equiv (c-1)^p + 1^p \equiv (c-1)+1 \equiv c \mod p
\end{aligned}$$

Malá Fermatova veta. Nech p je prvočíslo, nech c je ľubovoľné prirodzené číslo. Potom

$$c^p \equiv c \mod p.$$

Ak navyše $c \in \{1, 2, \dots, p-1\}$, potom

$$c^{p-1} \equiv 1 \mod p.$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Eulerova veta

Nech $a_1, a_2, \ldots, a_k$ sú všetky navzájom rôzne nesúdeliteľné čísla s číslom m menšie než m, kde m je ľubovoľné číslo, $k = \phi(m)$.
Vezmime x nesúdeliteľné s m a skúmajme množinu čísel $\{a_1x, a_2x, \ldots, a_kx\}$.
Sú to zase všetko čísla nesúdeliteľné s m.

Pre každú dvojicu i, j, $i \neq j$ platí $a_ix \not\equiv a_jx \mod m$
– inak by muselo byt $a_i \equiv a_j \mod m$ (a keďže $1 \leq a_i$, $a_j \leq m-1$) aj $a_i = a_j$.

Pre každé a_ix existuje práve jedno $a_{\pi[x]}$ také, že $a_ix \equiv a_{\pi[x]} \mod m$. Preto je

$$x^{\phi(m)} \cdot \prod_{i=1}^{\phi(m)} a_i \equiv \prod_{i=1}^{\phi(m)} (a_ix) \equiv \prod_{i=1}^{\phi(m)} a_{\pi[i]} \equiv \prod_{i=1}^{\phi(m)} a_i \mod m$$

Keďže súčin $\prod_{i=1}^{\phi(m)} a_i$ je nesúdeliteľný s m, obe strany poslednej kongruencie možno týmto súčinom vydeliť, čim dostávame nasledujúcu vetu:

Eulerova veta. Pre ľubovoľné číslo x nesúdeliteľné s číslom m platí

$$x^{\phi(m)} \equiv 1 \mod m.$$

Okruhy a polia

$\mathbb{Z}_p = (\{0, 1, 2, \ldots, p-1\}, \oplus, \otimes)$, kde

$$\begin{aligned} a \oplus b &= a + b \mod p \\ a \otimes b &= a.b \mod p \end{aligned}$$

Štruktúra $\mathbb{Z}_p$ je pole práve vtedy, keď p je prvočíslo.

Platí tam:

1. $a \oplus b = b \oplus a$
2. $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
3. $a \oplus 0 = 0 \oplus a = a$
4. $\forall a \exists b \ (a \oplus b = b \oplus a = 0)$
5. $a \otimes b = b \otimes a$
6. $(a \otimes b) \otimes c = a \otimes (b \otimes c)$
7. $a \otimes 1 = 1 \otimes a = a$
8. $\forall (a \neq 0) \exists b \ (a \otimes b = b \otimes a = 1)$
9. $a \otimes 0 = 0 \otimes a = 0$
10. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Vlastnosti poľa Z_p

Nech p je prvočíslo.
Riešime rovnicu $a \otimes x = 1$, t.j. hľadáme také x, že $ax \equiv 1 \mod p$.
Vieme že platí

$$\begin{aligned} a^{\phi(p)} &\equiv 1 \mod p \\ a.a^{\phi(p)-1} &\equiv 1 \mod p \\ x &\equiv a^{\phi(p)-1} \mod p \\ a^{-1} &\equiv a^{\phi(p)-1} \mod p \end{aligned}$$

Keďže p je prvočíslo, $\phi(p) = p - 1$, v $\mathbb{Z}_p$ je $x = a^{-1} = a^{p-2}$.

Rovnica $a \otimes x = b$ má v $\mathbb{Z}_p$ riešenie $x = a^{-1} \otimes b = a^{p-2} \otimes b$.

Zisťovanie prvočíselnosti veľkých čísel

Nech M je veľké číslo. Ak je M prvočíslo, podľa Fermatovej vety platí pre každé prirodzené c, $c < M$

$$c^{M-1} \equiv 1 \mod M.$$

Ak sa teda nájde také prirodzené číslo $c < M$, že

$$c^{M-1} \not\equiv 1 \mod M,$$

potom je M zložené číslo.

Fermatov test prvočíselnosti.

1. Ak pre niektoré $c < M$ je $c^{M-1} \not\equiv 1 \mod M$, potom je c určite zložené číslo.
2. Ak pre dostatočne veľa čísel $c < M$ platí $c^{M-1} \equiv 1 \mod M$, potom c je pravdepodobne prvočíslo.

Phill Zimmermann v PGP použil túto procedúru na zisťovanie prvočíselnosti M:

- Vylúčil M ak neprešlo testom vydelením všetkými 16-bitovými prvočíslami
- Aplikoval Fermatov test pre štyri hodnoty c.

Zisťovanie prvočíselnosti veľkých čísel

Carmichaelove číslo – také zložené číslo M, že pre všetky $c < M$, c nesúdeliteľné s M platí $c^{M-1} \equiv 1 \mod M$.

$$
\begin{aligned}
561 &= 3 \cdot 11 \cdot 17 \\
1105 &= 5 \cdot 13 \cdot 17 \\
1729 &= 7 \cdot 13 \cdot 19 \\
2465 &= 5 \cdot 17 \cdot 29 \\
2821 &= 7 \cdot 13 \cdot 31 \\
6601 &= 7 \cdot 23 \cdot 41 \\
8911 &= 7 \cdot 19 \cdot 67
\end{aligned}
$$

Vlastnosti Carmichaelovho čísla M:

- M je zložené z aspoň troch prvočísel
- Žiadne prvočíslo sa v rozklade M neopakuje
- Carmichaelove čísla sú zriedkavé – medzi 1 a 10^{21} je ich najviac 20,138,200. Pravdepodobnosť, že číslo z intervalu $\langle 1, 10^{21} \rangle$ je Carmichaelovo je

$$\frac{2.10^7}{10^{21}} = 2.10^{-14}$$

$$9746347772161 = 7 \cdot 11 \cdot \dot{13} \cdot 17 \cdot 19 \cdot 31 \cdot 37 \cdot 41 \cdot 641$$

$C(N)$ počet Carmichaelových čísel medzi 1 a N

$$N^{0.332} < C(N) < N \cdot \exp\left(- \frac{\ln N \ln \ln \ln N)}{\ln \ln N} \right)$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Rabin – Millerov test prvočíselnosti

1. Vyjadri p v tvare $p = 1 + 2^s.r$, r nepárne
2. For $i = 1$ to t urob
   - 2.1 Vyber náhodné číslo a také, že $2 \leq a \leq p - 2$
   - 2.2 Polož $y = a^r \mod p$
   - 2.3 Ak $y \neq 1$ and $y \neq p - 1$ urob:

     j=1

     WHILE $(j \leq s - 1)$ and $(y \neq p - 1)$

     $$\begin{cases} y = y^2 \mod p \\ Ak\ y = 1,\ \text{RETURN ZLOŽENÉ} \\ j = j + 1 \end{cases}$$

     Ak $y \neq p - 1$ RETURN ZLOŽENÉ
3. RETURN PRVOČÍSLO S PRAVDEPODOBNOSŤOU $1 - \left(\frac{1}{4}\right)^t$

Kryptografické systémy s verejným kľúčom

Nevýhody symetrickej kryptografie:

- Každá dvojica účastníkov musí udržiavať svoj kľúč.
- Kľúčov je teda veľmi veľa a všetky sa musia udržať v tajnosti.

Princíp kryptografie s verejným kľúčom:

- Každý účastník A má jednu dvojicu kľúčov – Verejný kľúč $KV(A)$ a tajný kľúč $KT(A)$. Kľúč $KV(A)$ zverejní, kľúč $KT(A)$ utají.
- Účastník A šifruje správu x účastníkovi B tak, že nájde verejný kľúč $KV(B)$ a pošle správu $y = E_{KV(B)}(x)$.
- Účastník B dešifruje správu y predpisom $x = D_{KT(B)}(y)$.

RSA algoritmus

1. Účastník A zvolí dve veľké tajné prvočísla p, q.
2. $n = p \cdot q$
3. $\phi(n) = (p-1)(q-1)$
4. Ďalej zvojí dve čísla $1 < e < \phi(n)$, $1 < d < \phi(n)$ také, že

$$e \cdot d \equiv 1 \mod \phi(n)$$

5. Verejný kľúč účastníka A je dvojica (e, n),
   jeho tajný kľúč je dvojica (d, n)
6. Účastník B bude správu $x < n$ pre účastníka A šifrovať predpisom

$$y = x^e \mod n.$$

7. Účastník A dešifruje správu y predpisom

$$x = y^d \mod n.$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

RSA algoritmus – voľba prvočísel p a q

Problém voľby prvočísel p, q.

- Dostatočná veľkosť – aspoň 512 – 1024 bitov.
- Zisťovanie prvočíselnosti – použiť niektorý pravdepodobnostný test.
- Je ich dosť? Počet prvočísel menších než $n \approx \frac{n}{\ln n}$.

Niekedy sa požaduje, aby p, q boli silné prvočísla (strong prime).
Prvočíslo p je silné prvočíslo, ak

1. p je veľké
2. $p-1$ má veľký prvočíselný faktor, t.j. $p-1 = a_1 p_1$ pre niektoré veľké prvočíslo p_1
3. $p_1 - 1$ má veľký prvočíselný faktor, t.j. $p_1 - 1 = a_2 p_2$ pre niektoré veľké prvočíslo p_2
4. $p+1$ má veľký prvočíselný faktor, t.j. $p+1 = a_3 p_3$ pre niektoré veľké prvočíslo p_3

Voľba silných prvočísel bola motivovaná sťažením niektorých metód faktorizácie. Objavenie ďalších faktorizačných metód ukázalo, že pre tieto metódy silné prvočísla nepredstavujú problém.
Bruce Schneier ani Philip Zimmerman neodporúčajú silné prvočísla.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

RSA algoritmus – voľba exponentov e a d

Problém voľby čísel e, d.

- Veľmi často sa volí $e = 65537 = 2^{16} + 1$. e je prvočíslo.
- Číslo d také, že $e \cdot d \equiv 1 \mod \phi(n)$ sa nájde rozšíreným Euklidovým algoritmom.

Umocňovanie $x^d \mod n$ pre veľké d.

Bitová reprezentácia čísla d nech je $d[k-1] \ldots d[1]d[0]$.

```
temp=x;
y=1;
for(i=0; i<k; i++)
{if(d[i]==1) y=mod(y*temp,n);
 temp=mod(temp*temp,n);
}
return y;
```

RSA algoritmus – prečo to funguje

Nech $x < n$, $y = E(x) = x^e \mod n$.
Platí skutočne, že $D(y) = y^d \mod n = x$?

$$y^d \equiv (x^e)^d \equiv x^{ed} \equiv x^{k\phi(n)+1} \mod n$$

Čísla e, d boli vyberané tak aby $e.d \equiv 1 \mod \phi(n)$, t.j. aby $e.d = k.\phi(n) + 1$ pre nejaké prirodzené číslo k.

1. Ak x je nesúdeliteľné s n potom

$$\begin{aligned} x^{\phi(n)} &\equiv 1 \mod n \\ (x^{\phi(n)})^k &\equiv 1^k \mod n \\ x^{k.\phi(n)} &\equiv 1 \mod n \\ x \cdot x^{k.\phi(n)} &\equiv x \mod n \\ y^d \equiv x^{e.d} \equiv x^{k.\phi(n)+1} &\equiv x \mod n \end{aligned}$$

RSA algoritmus – prečo to funguje

2. Ak x a n sú súdeliteľné potom musí buď $p|x$ alebo $q|x$.

Nech $p|x$ potom $q \not| x$. (Inak by muselo byť $x = k.pq \geq n$.)

Eulerova veta ($x^{\phi(q)} \equiv 1 \mod q$) platí aj pre $x^{\phi(p)}$.

$$\begin{aligned} x^{\phi(n)} \equiv x^{\phi(p).\phi(q)} \equiv (x^{\phi(p)})^{\phi(q)} &\equiv 1 \mod q \\ (x^{\phi(p)})^{k.\phi(q)} &\equiv 1 \mod q \\ x^{k.\phi(p).\phi(q)} &\equiv 1 \mod q \\ x \cdot x^{k.\phi(n)} &\equiv x \mod q \end{aligned}$$

Máme teda

$$x^{k.\phi(n)+1} - x = L.q.$$

Keďže $p|x$, musí byť aj $p|L$, t.j. $L = M.p$. Preto je

$$x^{k.\phi(n)+1} - x = L.q = M.p.q = M.n$$

$$x^{k.\phi(n)+1} \equiv x \mod n.$$

Kontrola integrity a autenticity

Metódy zabezpečenia správy pri prenose

Na kontrolu toho, či správa nebola pri prenose zmenená, sa za správu pridáva časť zložená z kontrolných znakov, pomocou ktorej možno zistiť, či je správa nezmenená.

1. Kontrola paritou
2. Kontrola dekadických kódov modulo 10 resp. modulo 11
3. Lineárne (n, k)-kódy
4. Kontrolný súčet – napr. súčet všetkých čísel správy modulo 2^{64}
5. CRC – cyclic redundancy check
6. ...

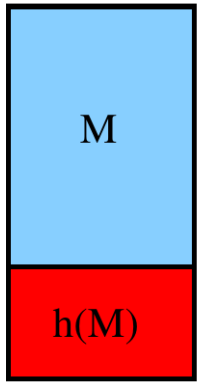
Tieto spôsoby sú účinné proti náhodným chybám, nie však proti zlomyselným útokom.
Útočník totiž ľahko dokáže zmeniť správu tak, tak aby kontrolné znaky zmenenej správy boli rovnaké ako kontrolné znaky pôvodnej správy.

Jednocestné hashovacie funkcie

V kryptografii sa na zaistenie správ proti zmenám pridáva ku každej správe M ďalšia (redundantná) časť, nazývaná **odtlačok správy**.

V anglickej literatúre MAC(M) – Message Autentification Code, MD(M) – Message Digest, Fingerprint.

Tieto sa vypočítavajú pomocou jednocestných hashovacích funkcií.



Požadované vlastnosti hashovacej funkcie $h(M)$

1. Pre každé M je ľahké vypočítať $h(M)$
2. Pre každé h je ťažké nájsť také M, že $h = h(M)$
3. Pre každé M je ťažké nájsť iné M' také, že $h(M) = h(M')$
4. Je ťažké nájsť dve rôzne náhodné správy $M \neq M'$ také, že $h(M) = h(M')$

Dvojica správ M, M' s vlastnosťou $h(M) = h(M')$ sa nazýva kolízia. Vlastnosť 4. sa nazýva odolnosť voči kolízii – collision resistence.

Narodeninový paradox

Birthday paradox - narodeninový paradox.

V skupine 23 ľudí sa s pravdepodobnosťou $> \frac{1}{2}$ nájde dvojica, ktorá má v ten istý deň narodeniny.

Majme n možných hodnôt $h(M)$ a náhodne generujme k správ $M_1, M_2, \ldots, M_k$.

Pri prvej správe M_1 nenastane kolízia s pravdepodobnosťou ..... $p = 1$

Pri pridaní druhej správy M_2
nenastane kolízia s pravdepodobnosťou ..... $p = (1 - \frac{1}{n})$

Za predpokladu, že medzi M_1 a M_2 nenastala kolízia,
po pridaní tretej správy M_3
nenastane kolízia s pravdepodobnosťou ..... $p = (1 - \frac{2}{n})$

Za predpokladu, že medzi $M_1, M_2, \ldots, M_{k-1}$ nenastala kolízia,
po pridaní k-tej správy M_k
nenastane kolízia s pravdepodobnosťou ..... $p = (1 - \frac{k-1}{n})$

Pravdepodobnosť kolízie

Pravdepodobnosť, že medzi k správami nenastala ani jedna kolízia je

$$\left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdot\ \ldots\ \cdot\left(1-\frac{k-1}{n}\right)=\prod_{i=1}^{k-1}\underbrace{\left(1-\frac{i}{n}\right)}_{\approx e^{-\frac{i}{n}}}.$$

$$e^{-x}=1-x+\underbrace{\frac{x^2}{2!}-\frac{x^3}{3!}+\frac{x^4}{4!}-\frac{x^5}{5!}+\ldots}_{\text{tento súčet je pre malé } x \text{ zanedbateľný}}$$

$$\prod_{i=1}^{k-1}e^{-\frac{i}{n}}=e^{\sum_{i=1}^{k-1}-\frac{i}{n}}=e^{-\frac{k(k-1)}{2n}}$$

Pravdepodobnosť, že nastala aspoň jedna kolízia je

$$1-e^{-\frac{k(k-1)}{2n}}.$$

Pravdepodobnosť kolízie

$$
\begin{aligned}
1 - e^{-\frac{k(k-1)}{2n}} &\geq \varepsilon \\
1 - \varepsilon &\geq e^{-\frac{k(k-1)}{2n}} \\
\ln(1-\varepsilon) &\geq -\frac{k(k-1)}{2n} \\
2n\ln(1-\varepsilon) &\geq -(k^2-k) \\
2n\ln(\frac{1}{1-\varepsilon}) &\leq (k^2-k)
\end{aligned}
$$

$$
k^2 - k - 2n\ln(\frac{1}{1-\varepsilon}) = 0
$$

$$
k_{1,2} = \frac{+1 \pm \sqrt{1 + 4.1.2n\ln(\frac{1}{1-\varepsilon})}}{2} =
$$

$$
= \frac{1}{2} \pm \sqrt{\frac{1}{4} + 2n\ln(\frac{1}{1-\varepsilon})} \approx \pm\sqrt{2n\ln(\frac{1}{1-\varepsilon})}
$$

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY
KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI ŽILINSKEJ UNIVERZITY V ŽILINE

Pravdepodobnosť kolízie

Ak teda $k \geq \sqrt{2n \ln(\frac{1}{1-\varepsilon})}$ pravdepodobnosť kolízie medzi k správami je väčšia než ε.

Pre narodeniny existuje $n = 365$ možností. Položme $\varepsilon = 0.5$ potom

$$k \geq \sqrt{2n \ln(\frac{1}{1-\varepsilon})} = \sqrt{730 \ln(\frac{1}{1-1/2})} == \sqrt{730 \ln(2))} = 22,4944$$

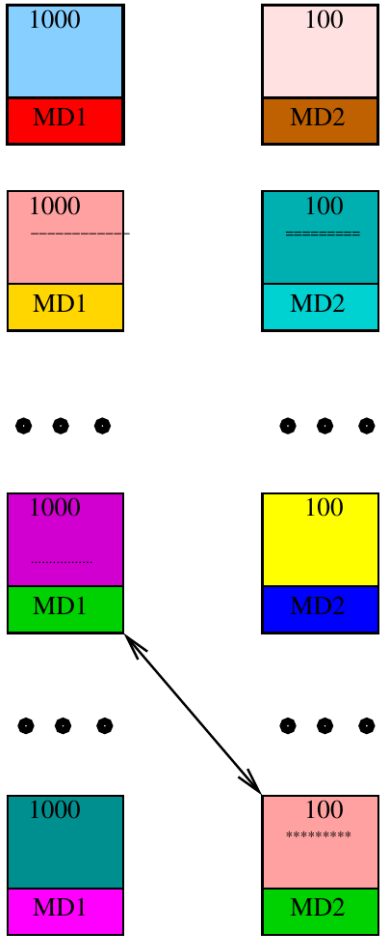
Všeobecne pre n a $\varepsilon = 0.5$

$$k \approx \sqrt{n.2. \ln(\frac{1}{2})} \approx 1,17.\sqrt{n}.$$

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.

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 dlžník potvrdil odtlačok h prislúchajúci 1000 eurovému variantu.

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

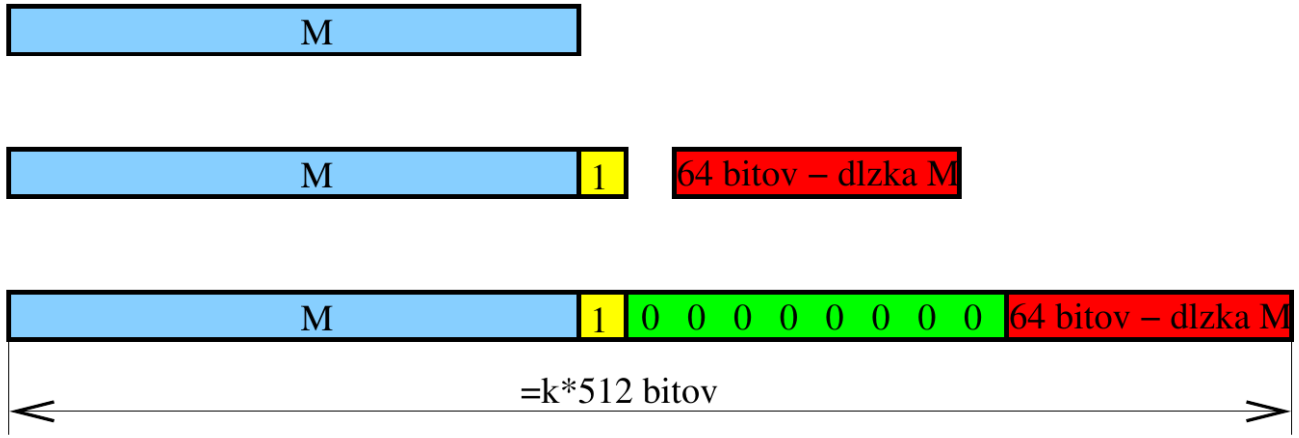
Všeobecný postup tvorby odtlačku správy

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

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 $h(M) = h_k$.

Algoritmus MD4

Správa sa pred výpočtom odtlačku musí upraviť tako.



1. Pridá sa jeden bit s hodnotou 1.
2. Vytvorí sa 64 bitové číslo obsahujúce dĺžku správy. Týmto číslom bude upravená správa končiť.
3. Medzi doplnenú jedničku a 64 bitov dĺžky sa vloží toľko núl, aby výsledná dĺžka správy bol násobkom 512.

Algoritmus MD4

- Dĺžka odtlačku algoritmu MD4 je 128 bitov, t.j. h_i má 128 bitov.
- S h_i sa pracuje ako so štvoricou (A, B, C, D) 32-bitových čísel
- Spracovávaná dĺžka bloku správy m_i je 512 bitov.
- S blokom textu sa pracuje ako so 16-ticou

$$X[0], X[1], X[2], \ldots, X[15]$$

  32-bitových čísel.
- Inicializačne sa nastaví hodnota $h_0 \equiv (A, B, C, D)$
- i-tý 512-bitový blok textu m_i sa vyjadrí v tvare šestnástich 32-bitových čísel $X[0], X[1], X[2], \ldots, X[15]$ a rekurentne sa vypočíta

$$h_i = f(m_i, h_{i-1})$$

- Ak m_k je posledný blok správy, potom h_k je odtlačok celej správy.

Algoritmus MD4

Význam použitých operácií:

$+$ – sčítanie modulo 2^{32}
$\wedge$ – logické and po bitoch
$\vee$ – logické or po bitoch
$\neg$ – logická negácia po bitoch

MD4 bude používať tieto funkcie:
$f(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$
$g(X, Y, Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$
$h(X, Y, Z) = X \oplus Y \oplus Z$

Nastavenie 128-bitového inicializačného vektora $IV = (A, B, C, D)$:
$A = 67452301$
$B = efcdab89$
$C = 98badcfe$
$D = 10325476$

Funkcia $h_i = f(m_i, h_{i-1})$

0 Vstup $(A, B, C, D) = h_{i-1}$,
$(X[0], X[1], \ldots, X[15]) = m_i$
1. Uloženie $h_{i-1} = (A, B, C, D)$
$AA = A$
$BB = B$
$CC = C$
$DD = D$
2. 1.kolo$(A, B, C, D, X[0-15])$
3. 2.kolo$(A, B, C, D, X[0-15])$
4. 3.kolo$(A, B, C, D, X[0-15])$
5. $A = A + AA$
$B = B + BB$
$C = C + CC$
$D = D + DD$
6 Return $h_i = (A, B, C, D)$

Algoritmus MD4 – 1. kolo

$$
\begin{aligned}
1.\ & A = (A + f(B,C,D) + X[0]) <<< 3 \\
2.\ & D = (D + f(A,B,C) + X[1]) <<< 7 \\
3.\ & C = (C + f(D,A,B) + X[2]) <<< 11 \\
4.\ & B = (B + f(C,D,A) + X[3]) <<< 19 \\
5.\ & A = (A + f(B,C,D) + X[4]) <<< 3 \\
6.\ & D = (D + f(A,B,C) + X[5]) <<< 7 \\
7.\ & C = (C + f(D,A,B) + X[6]) <<< 11 \\
8.\ & B = (B + f(C,D,A) + X[7]) <<< 19 \\
9.\ & A = (A + f(B,C,D) + X[8]) <<< 3 \\
10.\ & D = (D + f(A,B,C) + X[9]) <<< 7 \\
11.\ & C = (C + f(D,A,B) + X[10]) <<< 11 \\
12.\ & B = (B + f(C,D,A) + X[11]) <<< 19 \\
13.\ & A = (A + f(B,C,D) + X[12]) <<< 3 \\
14.\ & D = (D + f(A,B,C) + X[13]) <<< 7 \\
15.\ & C = (C + f(D,A,B) + X[14]) <<< 11 \\
16.\ & B = (B + f(C,D,A) + X[15]) <<< 19
\end{aligned}
$$

Algoritmus MD4 – 2. kolo

$$
\begin{array}{rlcl}
1. & A & = & (A + g(B,C,D) + X[0] + 5a827999) <<< 3 \\
2. & D & = & (D + g(A,B,C) + X[4] + 5a827999) <<< 5 \\
3. & C & = & (C + g(D,A,B) + X[8] + 5a827999) <<< 9 \\
4. & B & = & (B + g(C,D,A) + X[12] + 5a827999) <<< 13 \\
5. & A & = & (A + g(B,C,D) + X[1] + 5a827999) <<< 3 \\
6. & D & = & (D + g(A,B,C) + X[5] + 5a827999) <<< 5 \\
7. & C & = & (C + g(D,A,B) + X[9] + 5a827999) <<< 9 \\
8. & B & = & (B + g(C,D,A) + X[13] + 5a827999) <<< 13 \\
9. & A & = & (A + g(B,C,D) + X[2] + 5a827999) <<< 3 \\
10. & D & = & (D + g(A,B,C) + X[6] + 5a827999) <<< 5 \\
11. & C & = & (C + g(D,A,B) + X[10] + 5a827999) <<< 9 \\
12. & B & = & (B + g(C,D,A) + X[14] + 5a827999) <<< 13 \\
13. & A & = & (A + g(B,C,D) + X[3] + 5a827999) <<< 3 \\
14. & D & = & (D + g(A,B,C) + X[7] + 5a827999) <<< 5 \\
15. & C & = & (C + g(D,A,B) + X[11 + 5a827999]) <<< 9 \\
16. & B & = & (B + g(C,D,A) + X[15] + 5a827999) <<< 13
\end{array}
$$

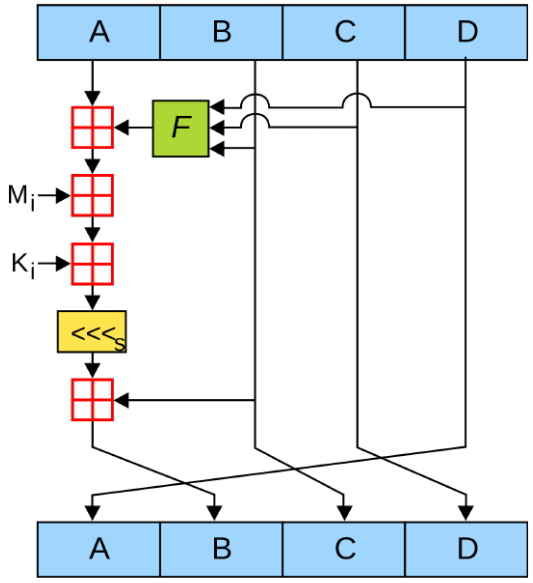
Algoritmus MD4 – 3. kolo

$$
\begin{aligned}
1.\ & A = (A + h(B,C,D) + X[0] + 6ed9eba1) <<< 3 \\
2.\ & D = (D + h(A,B,C) + X[8] + 6ed9eba1) <<< 9 \\
3.\ & C = (C + h(D,A,B) + X[4] + 6ed9eba1) <<< 11 \\
4.\ & B = (B + h(C,D,A) + X[12] + 6ed9eba1) <<< 15 \\
5.\ & A = (A + h(B,C,D) + X[2] + 6ed9eba1) <<< 3 \\
6.\ & D = (D + h(A,B,C) + X[10] + 6ed9eba1) <<< 9 \\
7.\ & C = (C + h(D,A,B) + X[6] + 6ed9eba1) <<< 11 \\
8.\ & B = (B + h(C,D,A) + X[14] + 6ed9eba1) <<< 15 \\
9.\ & A = (A + h(B,C,D) + X[1] + 6ed9eba1) <<< 3 \\
10.\ & D = (D + h(A,B,C) + X[9] + 6ed9eba1) <<< 9 \\
11.\ & C = (C + h(D,A,B) + X[5] + 6ed9eba1) <<< 11 \\
12.\ & B = (B + h(C,D,A) + X[13] + 6ed9eba1) <<< 15 \\
13.\ & A = (A + h(B,C,D) + X[3] + 6ed9eba1) <<< 3 \\
14.\ & D = (D + h(A,B,C) + X[11] + 6ed9eba1) <<< 9 \\
15.\ & C = (C + h(D,A,B) + X[7] + 6ed9eba1) <<< 11 \\
16.\ & B = (B + h(C,D,A) + X[15] + 6ed9eba1) <<< 15
\end{aligned}
$$

Algoritmus MD5

MD5 algoritmus

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



Algoritmus SHA1

SHA algoritmus

1. SHA produkuje 160-bitový hash
2. Spracováva 512 bitový blok textu $W[0], W[1], \ldots, W[15]$, 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

Algoritmus SHA1

```
Initialize hash value for this chunk:
a = h0; b = h1; c = h2; d = h3; e = h4;
Main loop:
for( i=0; i < 80; i++)
  {if (0 <= i) and (i <= 19) then   {f = (b and c) or ((not b) and d;)
                                     k = 0x5A827999;}
  else if( 20<=i) and (i <= 39)    {f = b xor c xor d;
                                     k = 0x6ED9EBA1;}
  else if(40 <= i) and (i <= 59) {f = (b and c) or (b and d)or(c and d);
                                     k = 0x8F1BBCDC;}
  else if(60<= i) and (i <= 79)   {f = b xor c xor d;
                                     k = 0xCA62C1D6;}
  temp = (a leftrotate 5) + f + e + k + w[i];
  e = d; d = c; c = b leftrotate 30; b = a; a = temp;
}

  Add this chunk's hash to result so far:
  h0 = h0 + a; h1 = h1 + b; h2 = h2 + c; h3 = h3 + d; h4 = h4 + e;
```

Prístupové heslá

Používateľ	Počítač
heslo	Skontroluje, či sa zhoduje s uloženým heslom
heslo	Skontroluje, či sa zhoduje s uloženým MD hesla

Slovníkový útok - Dictionary attack

- krstné mená
- zemepisná názvy
- astronomické názvy
- bájne postavy
- biblické postavy
- chemické prvky
- dni a mesiace
- mená hercov umelcov spevákov
- názvy kníh, diel udalostí

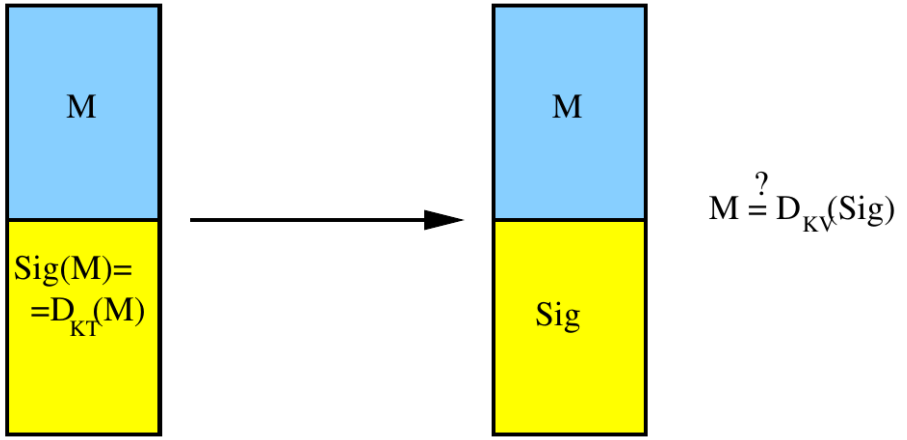
Používateľ	Salt	MD(Heslo, Salt)
peterp	$EA1DFC48_H$	128 bitov

Digitálny podpis

Princíp digitálneho podpisu

- 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 $M = M'$.

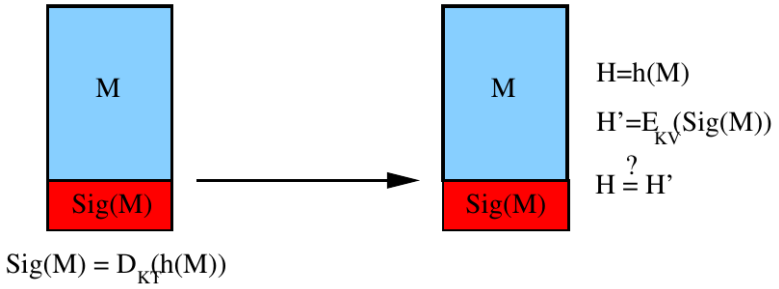
  Ak $M' \neq M$, potom buď správa bola zmenená, alebo podpis nie je pravý.

  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).

Princíp digitálneho podpisu

- 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).

Časové značky

- K správe M pridáme časť $X = MD(M), PUB$, kde $MD(M)$ je odtlačok správy M a PUB je verejne známa informácia, ktorá vznikla v deň podpisu správy M.
- Ako podpis pridáme časť $Y = sig(MD(M), PUB)$.
- V novinách publikujeme trojicu $MD(M)$, PUB, $sig(MD(M), PUB)$, takže môžeme dokázať, že sme podpísali správu s odtlačkov $MD(M)$, s verejne známou informáciou PUB.

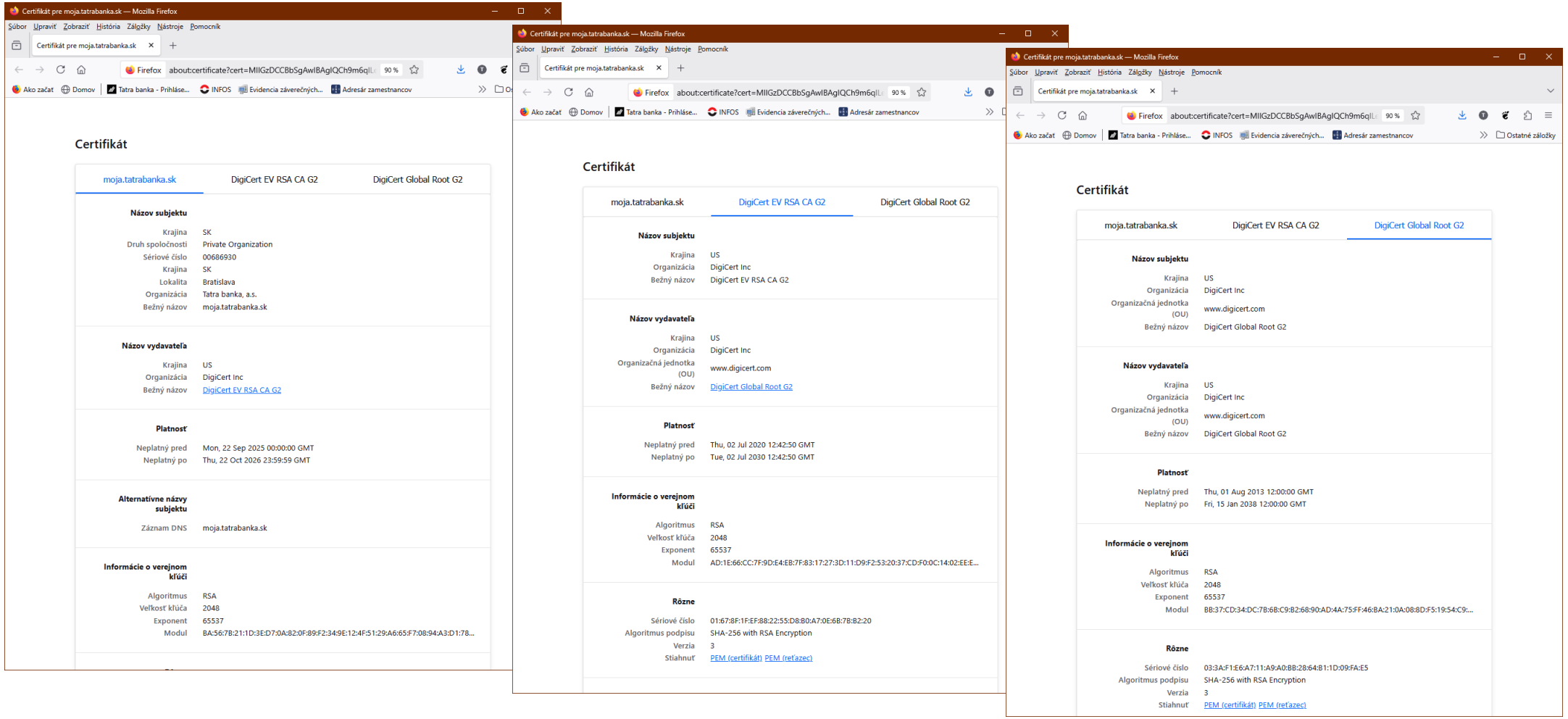
Problém distribúcie verejného kľúča

- Verejný kľúč sa distribuuje formou certifikátu, ktorý okrem informácii o verejnom kľúči a použitom algoritme obsahuje aj informácie o majiteľovi certifikátu (osoba alebo server), vymedzení časovej platnosti, vydavateľovi certifikátu a jeho digitálnom podpise.
- Certifikáty vydáva spravidla certifikačná autorita po dôkladnej kontrole identity (osobe po kontrole občianskeho preukazu, serveru po kontrole administrátorského prístupu – zápisu v DNS, publikovaní WWW stránky a pod.)
- Vo výnimočných prípadoch je potrebné ukončiť platnosť certifikátu okamžite (napríklad kvôli prezradeniu privátneho kľúča), preto certifikačná autorita udržiava a publikuje zoznam odvolaných certifikátov.
- Certifikačné autority vytvárajú hierarchiu, najvyššie je koreňová certifikačná autorita, pod ňou môžu byť lokálne certifikačné autority. Vzniká tzv. certifikačná cesta od koreňa k listu stromu.

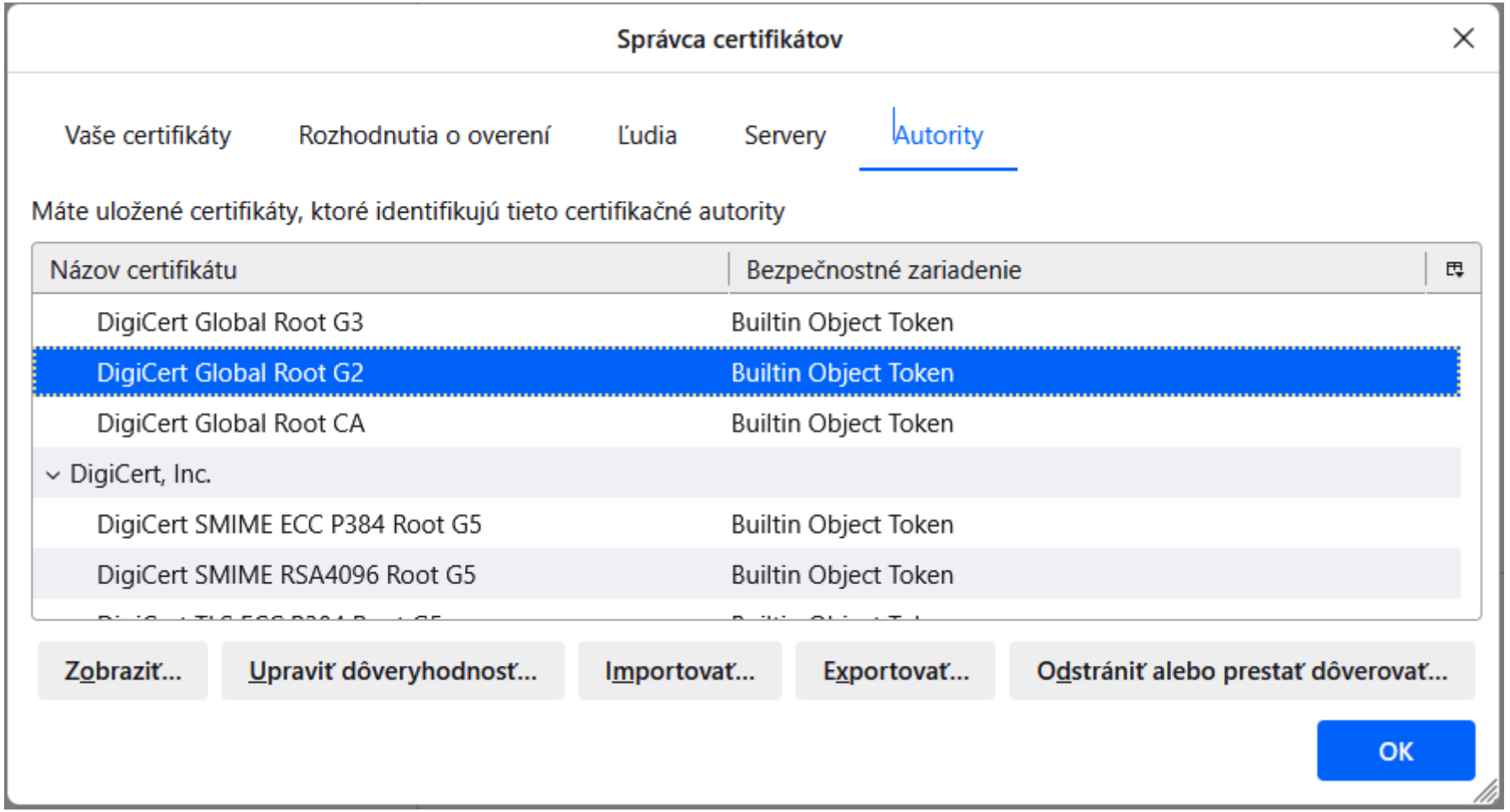
Overenie dôveryhodnosti certifikátu

- Overím digitálny podpis vydavateľa.
- Overím časovú platnosť certifikátu.
- Skontrolujem, či certifikát nebol odvolaný (online v zozname CRL).
- Skontrolujem certifikát vydavateľa (rekurzívne až po koreň).
- Certifikát koreňovej certifikačnej autority je tzv. self-signed certifikát a ten sa nedá poriadne skontrolovať. Preto sú tieto koreňové certifikáty inštalované spoločne s operačným systémom pripadne internetovým prehliadačom alebo manuálne. Existujú konfiguračné nástroje, ktoré umožňujú odstrániť (resp. pozastaviť platnosť) týmto certifikátom.

Overenie dôveryhodnosti certifikátu



Overenie dôveryhodnosti certifikátu



Financované Európskou úniou
NextGenerationEU

PLÁN OBNOVY

MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

KOMPETENČNÉ CENTRUM KYBERNETICKEJ BEZPEČNOSTI
ŽILINSKEJ UNIVERZITY V ŽILINE

Ďakujem za pozornosť

Základy kryptografie a ochrany dát

Kryptografia, ochrana dát a bezpečná komunikácia (Blok III)

Kurz: Špecialista kybernetickej bezpečnosti vo verejnej správe

Ing. Tomáš Majer, PhD.

KC KYB UNIZA, https://kc.uniza.sk

tomas.majer@fri.uniza.sk