CAESAROVA ŠIFRA

Caesarova šifra je jedna z nejstarších a nejjednodušších metod šifrování. Byla pojmenována po Juliu Caesarovi, který ji údajně používal k zabezpečení svých důležitých vojenských zpráv. Caesarova šifra funguje tak, že posune každé písmeno v otevřeném textu o pevně stanovený počet pozic v abecedě. Například, pokud je posun 3, pak 'A' se šifruje jako 'D', 'B' jako 'E', atd. Při posunu 3 se 'Z' stane 'C'. Tento posun se aplikuje na každé písmeno v zprávě, což vytváří šifrovaný text.

Caesarova šifra je typ substituční šifry, která je založena na posunu písmen v abecedě.
Matematické znázornění Caesarovy šifry může vypadat takto:

Když si zvolíme P jako množinu všech písmen v abecedě a p jako písmeno z této množiny, které chceme šifrovat.
Dále zvolíme k jako klíče, tedy počet pozic, o které chceme každé písmeno posunout. Pak šifrované písmeno c získáme jako:

$$c = (p + k) \bmod |P|$$

Kde $|P|$ je velikost abecedy (například pro anglickou abecedu je $|P| = 26$ a \mod značí modulo operaci, která zajišťuje, že pokud přesáhneme konec abecedy, začneme opět od začátku.

Pro dešifrování šifroavného písmena c použijeme posun:

$$p = (c - k) \bmod |P|$$

Toto je základní matematický popis operací potřebných pro šifrování a dešifrování textu pomocí Caesarovy šifry.

Příklad:
Pokud máme klíč k = 3 a chceme zašifrovat písmeno 'A' (které má v abecedě pozici 0), výsledek bude:

$$c = (0 + 3) \bmod 26 = 3$$

Takže 'A' se po šifrování stane 'D', což je čtvrté písmeno v abecedě (počítáno od 0).

Šifrování s posunem o 3: HELLO WORLD → KHOOR ZRUOG

H →	K
E →	H
L →	O
L →	O
O →	R
W →	Z
O →	R
R →	U
L →	O
D →	G

RSA ALGORITMUS

RSA algoritmus, pojmenovaný podle jeho tvůrců Ronalda Rivesta, Adiho Shamira a Leondarda Adlemana, je jedním z prvních veřejných klíčových kryptosystémů a je široce používán pro zabezpečený přenos dat. Byl poprvé publikován v roce 1977 a jeho bezpečnost spočívá v praktické nemožnosti faktorizovat velmi velká čísla na prvočísla. Klíčovým konceptem je, že i když je snadné vynásobit dvě velká prvočísla a získat jejich produkt, je extrémně obtížné provést opačnou operaci - rozložit dané velké číslo zpět na původní prvočísla.

RSA algoritmus pracuje s párem klíčů - veřejným a soukromým klíčem. Veřejný klíč je použit k šifrování zprávy a soukromý klíč k jejímu dešifrování.
Algoritmus využívá následující matematické principy:

1. Vyberou se dvě velká prvočísla p a q.
2. Vypočítá se n = pq, což je modul pro oba klíče a jeho velikost určuje sílu šifry.
3. Vypočet Eulerovi funkce $\phi(n)=(p-1)(q-1)$.
4. Výběr veřejného exponentu e tak, aby byl nesoudělný s $\phi(n)$ a $1 < e < \phi(n)$.
5. Vypočítá se soukromý exponent d tak, aby platilo de mod $\phi(n) = 1$.

Příklad:

Hodnota p	Hodnota q	Hodnota n	Hodnota ϕ(n)	Veřejný exponent e	Soukromý exponent d	Zpráva (ASCII kód)	Šifrovaný text	Dešifrovaný text
3	11	33	20	3	7	2	8	2

Pro šifrování zprávy M do šifrovaného textu C se použije:

$$C = M^e \bmod n$$

A pro dešifrování šifrovaného textu C zpět na původní zprávu M:

$$M = C^d \bmod n$$

VIGENÈROVA ŠIFRA

Vigenèrova šifra byla poprvé pospsána Giovanem Battistou Bellasem v 16. století a později znovuobjevena Blaisem de Vigenèrem v 19. století. Tato šifra byla považována za "neprůstřelou" až do 19. století, kdy byly objeveny metody jejího prolomení, jako je například metoda Kasikiho zkoumání. Vigenèrova šifra byla populární pro svou zdánlivou bezpečnost v dobách před počítačovou érou a byla používána pro vojenské i osobní tajné komunikace.

Vigenèrova šifra funguje tak, že přiřazuje každému písmenu v otevřeném textu písmeno z klíče.
Pokud je klíč kratší než text, opakuje se. Při šifrování se používá posun písmen podle pozice odpovídajícího písmene v klíči, podobně jako u Caesarovy šifry.
Matematicky lze šifrování a dešifrování popsat pomocí modulární aritmetiky, kde A je otevbřený text, K je klíč, C je šifrový text, a N je počet písmen v abecedě (typicky 26):

Šifrování: C = (A + K) mod N
Dešifrování: A = (C - K + N) mod N

Příklad:

TEXT	KLÍČ
HELLO WORLD	KEY
HELLOWORLD	KEYKEYKEYK

Výsledný šifrovaný text: **RIJVSUYVJN**

Jednotlivé znaky	Číselná hodnota znaků	Jednotlivé znaky klíče	Číselná hodnota znaků klíče	Šifrový text	Dešifrované číslo	Dešifrovaný text
H	7	K	10	R	7	H
E	4	E	4	I	4	E
L	11	Y	24	J	11	L
L	11	K	10	V	11	L
O	14	E	4	S	14	O
W	22	Y	24	U	22	W
O	14	K	10	Y	14	O
R	17	E	4	V	17	R
L	11	Y	24	J	11	L
D	3	K	10	N	3	D

Zadejte text a klíč k zašifrování:

TEXT	KLÍČ
auto	key
AUTO	KEYK

Výsledný šifrovaný text: **KYRY**

Jednotlivé znaky	Číselná hodnota znaků	Jednotlivé znaky klíče	Číselná hodnota znaků klíče	Šifrový text	Dešifrované číslo	Dešifrovaný text
A	0	K	10	K	0	A
U	20	E	4	Y	20	U
T	19	Y	24	R	19	T
O	14	K	10	Y	14	O

HILLOVA ŠIFRA

Hillova šifra byla vynalezena matematikem Lesterem S. Hillem v roce 1929. Jedná se o jednu z prvních šifer, která systematicky využívala matematiku, konkrétně lineární algebru. Hillova šifra je založena na použití matic pro šifrování bloků textu, což bylo v té době revoluční. Jedná se o polygrafickou šifru, což znamená, že šifruje vstupní text po blocích více znaků najedou, což komplikuje frekvenční analýzu, která je častou metodou pro prolomení šifer

Základem Hillovy šifry je šifrovací matice, která musí být čtvercová a musí mít inverzní matici v modulární aritmetice (obvykle se používá modulo 26 pro anglickou abecedu, kde A = 0, B =1, …, Z = 25).
Pokud je šifrovací matice n * n, pak šifrujeme bloky textu o délce n.

Předpokládejme, že máme šifrovací matici K a blok textu reprezentovaný jako vektro P, kde P má také délku n. Pak šifrovaný text C získáme jako:
C = K * P mod m

kde m je velikost abecedy (pro anglickou abecedu 26). Pro dešifrování musíme použít inverzní matici K^{-1} a aplikovat podobnou operaci na šifrovaný text

Pro příklad si můžeme vzít matici K = $\begin{pmatrix}3 & 3\\2 & 5\end{pmatrix}$ zašifrujeme text "HI"

Matice		Číselná hodnota písmen H a I	Součin matic	Hodnoty šifrovaného textu	Šifrovaný text
3	3	7	45	19	T
2	5	8	54	2	C

Zašifrovaný text HI tedy bude TC

Pro dešifraci je poté potřeba spočítat inverzní hodnotu determinantu matice K.
Determinant pro matici $\begin{pmatrix}3 & 3\\2 & 5\end{pmatrix}$ je vypočítán jako:

det(K) = (3*5)-(3*2) = 15 - 6 = 9

Nyní je potřeba nají inverzní hodnotu čísla 9 v modulu 26. To znamená, že hledáme číslo x, které splňuje rovnici:

$9x \equiv 1 \bmod 26$

Tuto hodnotu x lze najít buď zoušením, což je pro malá čísla docela proveditelné, nebo použitím rozšířeného Euklidova algoritmu pro inverzi v modulární artimetice.
Pro tento případ, pokud zkusíme několik hodnot, zjistíme že x = 3 je řešením, protože:

$9*3 = 27 \equiv 1 \bmod 26$

Takže inverzní hodnota determinantu 9 v modulu 26 je 3. Tuto hodntou pak použijeme při výpočtu inverzní matice K^{-1} tím, že každý prvek adjungované matice vynásobíme třemi a upravíme výsledek modulo 26

Dalším krokem je tedy výpočet adjungované matice. V případě 2x2 matice je to jednoduché: prohodí se hodnoty na hlavní diagonále a změní se znaménka na nediagonálních hodnotách. Matice poté bude vypadat takto: $\begin{pmatrix}5 & -3\\-2 & 3\end{pmatrix}$

Každá hodnota této matice se poté vynásobí hodnotou 3 a vezme se modulo 26, čili takto: $\begin{pmatrix}15 & -9\\-6 & 9\end{pmatrix}$

Poté se již pouze vynásobí tato matice s maticí hodnot šifrovaného textu, čili: $\begin{pmatrix}15 & -9\\-6 & 9\end{pmatrix} \times \begin{pmatrix}19\\2\end{pmatrix}$

Inverzní matice		Zašifrovaný text	Dešifrovaná matice	Dešifrovaná matice mod 26	Dešifrovaný text
15	-9	19	267	7	H
-6	9	2	-96	8	I

VERNAMOVA ŠIFRA

Vernamova šifra byla vynalezena v roce 1917 Gilbertem Vernamem, inženýrem v AT&T. Původně byla navržena jako mechanický systém pro zabezpečení telegrafních komunikací. Klíčovým postupem je, že klíč pro šifrování a dešifrování je stejně dlouhý jako samotná zpráva a musí být zcela náhodný a použit pouze jednou, což zajišťuje maximální bezpečnost. Pokud jsou tyto podmínky splněny, šifra je v principu nerozluštitelná, protože pro každou možnou dešifrovanou zprávu existuje stejně pravděpodobný klíč.

Vernamova šifra funguje na principu "xor" operace (exclusive OR), která je jednoduchá, ale velmi efektivní pro šifrování.
Pro každý bit/znak zprávy se provede operace XOR s odpovídajícím bitem/znakem klíče.
Operace XOR (exclusive OR) je binární operace, která porovnává dva bity a vrací 1, pokud jsou bity různé, a 0, pokud jsou stejné. Tato vlastnost je základem pro mnoho šifrovacích systémů, včetně Vernamovy šifry, protože umožňuje jednoduché šifrování a dešifrování zpráv.
XOR ($\oplus$) tedy funguje takto:
$0 \oplus 0 = 0$: Pokud jsou oba bity 0, výsledek je 0.
$0 \oplus 1 = 1$: Pokud je jeden bit 0 a druhý 1, výsledek je 1, protože se bity liší.
$1 \oplus 0 = 1$: Stejně tak, pokud je jeden bit 1 a druhý 0, výsledek je 1.
$1 \oplus 1 = 0$: Pokud jsou oba bity 1, výsledek je 0, protože bity nejsou různé.

00110101	Text
11100011	Klíč
11010110	Šifrovaný text

Matematicky:

Pokud zvolíme M jako zprávu, K jako klíč a C jako šifrovanou zprávu. Pro každý bit/znak platí:
$C_i = M_i \oplus K_i$
Pro dešifrování se použije stejná operace:
$M_i = C_i \oplus K_i$

Pro demonstraci použijeme jednoduchou zprávu a jednoduchý klíč. Předpokládejme, že naše zpráva je "AHOJ" a náš klíč je "KLMN".

Zpráva	Klíč	ASCII hodnoty znaků zprávy	ASCII hodnoty znaků klíče	Šifrování pomocí XOR	Šifrovaný text	Dešifrace	Dešifrovaný text
AHOJ	KLMN				KMCE		AHOJ
A	K	0	10	10	K	0	A
H	L	7	11	12	M	7	H
O	M	14	12	2	C	14	O
J	N	9	13	4	E	9	J

BLOWFISH

Blowfish je bloková šifra, která byla navržena s cílem být rychlá a efektivní jak v hardwaru, tak v softwaru. Je založena na Feistelově síti, což je struktura, která umožňuje šifrování a dešifrování použitím stejných kroků, ale s použitím klíčů v opačném pořádí.

Klíčové prkvy algrotimu:
Feistelova síť: Algoritmus dělí 64bitové bloky dat na dve 32bitové poloviny. Během každého z 16 cyklů se provádějí různé operace (např. substituce a permtuace) na jedné polovině, zatímco druhá polovina prochází nezměněna a následně se obě poloviny prohodí.
S-boxy a P-boxy: S-boxy jsou tabulky používané pro substituci, zatímco P-boxy se používají pro permutaci. Tyto boxy jsou inicializovýny pomocí klíčového plánu, který expanduje původní klíč.
Klíčový plán: Klíč je rozšířen a použit k inicializaci S-boxů a P-boxů. Tento proces je klíčový pro zabezpečení, protože i malá změna v klíči má velký vliv na generované S-boxy a P-boxy, a tím i na výsledek šifrování.

Proces šifrování:
1. Inicializace: Data jsou rozdělena na dvě poloviny.
2. 16 cyklů zpracování: Každý cyklus zahrnuje několik kroků, včetně rozdělení dat, aplikace S-boxů pro substituci, permutace pomocí P-boxů a kombinace s druhou polovinou dat.
3. Závěrečná výměna: Po 16 cyklech se poloviny dat vymění.
4. Výstup: Kombinace obou polovin dat tvoří 64bitový šifrovaný blok.