Historie kryptografie

1) Transpoziční algoritmy – Spartská skytála

Odesilatel a příjemce museli mít oba tzv. **skytálu**: byly to dva válce o přesně stejném průměru. Odesilatel navinul úzkou pergamenovou pásku spirálovitě okolo své skytály a napsal pak podle délky svou zprávu na pásku.

UNDTLATEDZEEIOVEMEJKSSMYNZ.EOI

IELAENLTCTENLOIEKRZOAMKKIUENN

Skytála odesílatele má průměr, který můžeme vyjádřit pomocí počtu písmen. Můžeme tedy jednoduše vyzkoušet různé rozsahy u. Zvolíme-li $u = 7$, dostaneme následující nesmysl:

U	E	V	S	O	N	L	O	E
N	D	E	M	I	L	O	A	V
D	Z	M	Y	I	T	I	M	N
T	E	E	N	E	C	E	K	
L	E	J	Z	L	T	K	K	
A	I	K	.	A	E	R	I	
T	O	S	E	E	N	Z	U	,

zvolíme-li ale uspořádání textu pro *u* = 9, dostaneme:

U	Z	J	E	N	O	M
N	E	K	O	L	I	K
D	E	S	I	T	E	K
T	I	S	I	C	K	I
L	O	M	E	T	R	U
A	V	Y	L	E	Z	E
T	E	N	A	N	O	V
E	M	Z	E	L	A	N
D	E	.				

Nejedná se o nic jiného, než o permutaci *míst* písmen.

2) Substituční algoritmy – Caesarova šifra

Šifru použitou Caesarem obdržíme tím způsobem, že místo abecedy zprávy budeme psát abecedu kryp- togramu, ale o 23 míst doprava, což znamená totéž, jako posunutí doleva o 3 místa:

Zpráva: **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**
Kryptogram: **ABCDEFGHIJKLMNOPQRSTUVWXYZ.**

Šifruje se tím způsobem, že nahradíme písmeno zprávy pod ním stojícím písmenem kryptogramu. Na- příklad ze slova **zprava** se stane zdánlivě nesmyslné slovo **CSUDYD**.

U obou typů šifer máme tzv. šifrovací klíč.

Statistická analýza:

Dle četností jednotlivých písmen je možné identifikovat klíč Caesarovi šifry.

Početnosti jsou uvedeny v následující tabulce:

Písmeno	Četnost v % němčina	Četnost v % angličtina
a	6.51	6.40
b	1.89	1.40
c	3.06	2.70
d	5.08	3.50
e	17.40	10.00
f	1.66	2.00
g	3.01	1.40
h	4.76	4.20
i	7.55	6.30
j	0.27	0.30
k	1.21	0.60
l	3.44	3.50
m	2.53	2.00

Písmeno	Četnost v % němčina	Četnost v % angličtina
n	9.78	5.60
o	2.51	5.60
p	0.79	1.70
q	0.02	0.40
r	7.00	4.90
s	7.27	5.60
t	6.15	7.10
u	4.35	3.10
v	0.67	1.00
w	1.89	1.80
x	0.03	0.03
y	0.04	1.80
z	1.13	0.02

Úloha: Vytvořte tabulku četností písmen českého jazyka a vytvořte dešifrátor Caesorovi šifry s neznámým klíčem. Prezentujte algoritmus.

3) Monoabecední šifrování.

Šifrování se nazývá **monoabecední**, jestliže každé písmeno abecedy zprávy je zašifrováno jako nějaké písmeno téže abecedy. Monoabecední šifrování si můžeme představit tím, že pod abecedu zprávy napíšeme abecedu kryptogramu. Např. následující metody šifrování jsou monoabecední.

Zpráva: **a b c d e f g h i j k l mn o p q r s t u v wx y z**
Kryptogram: **QWERTZUIOPASDFGHJKLYXCVBNM.**

4) Hillova šifra

Hillova šifra lineárně transformuje *d* znaků otevřeného textu na *d* znaků šifrového textu. Bude-li *d* = 2, pak

$$\mathbf{C} = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}, \mathbf{M} = \begin{pmatrix} m_0 \\ m_1 \end{pmatrix}, \mathbf{K} = \begin{pmatrix} k_0 & k_2 \\ k_1 & k_3 \end{pmatrix}.$$

Šifrování zahrnuje násobení regulární matice K blokem otevřeného textu M, t.j C = KM.

Dešifrování zahrnuje násobení matice K^{-1} blokem šifrového textu C, tj $M = K^{-1}C$.

Příklad. *d* = 2 a budeme pracovat nad abecedou s 27 znaky (26 písmen a mezera), tj. nad $\mathbf{Z}_{27}$ (to je potřeba z toho důvodu, abychom pracovali nad konečným tělesem). Zvolme

$$\mathbf{K} = \begin{pmatrix} 4 & 6 \\ 1 & 7 \end{pmatrix}, \text{ pak } \mathbf{K}^{-1} = \begin{pmatrix} 4 & 12 \\ 11 & 10 \end{pmatrix}.$$

Připomeňme, že $\boldsymbol{Z_p}$ je těleso a $\boldsymbol{Z_p^k}$ není těleso, ale existuje těleso o velikosti p^k.

Úloha: Vytvořte šifrátor a dešifrátor Hillovy šifry se známým klíčem v běžném programovacím prostředí (ne matematickém). Prezentujte algoritmus.

Příklad. V $\boldsymbol{Z_5}$ spočtěte inverzní matici k matici $((2,3),(4,2))^T$

5) Viegenerova šifra – polyabecední šifra

Viegenerův čtverec: Tento čtverec se skládá z 26 abeced, které jsou napsány pod sebou takovým způsobem, že první abeceda je obyčejná abeceda, druhá abeceda je o jedno písmeno posunutá, třetí o dvě atd. Jinak řečeno: Vigenerův čtverec sestává z 26 posouvacích šifer v přirozeném pořadí.
Klíčovým slovem může být libovolná posloupnost písmen; pro náš demonstranční případ vybereme slovo
VENUSE.

Klíčové slovo:	V	E	N	U	S	E	V	E	N	U	S	E
Zpráva:	**p**	**o**	**l**	**y**	**a**	**b**	**e**	**c**	**e**	**d**	**n**	**i.**

Při šifrování určí písmeno klíčového slova, které stojí nad určitým písmenem zprávy příslušnou abecedu tj. *řádku* ve Vigenerově čtverci a pomocí této abecedy bude písmeno zprávy šifrováno.
Celkem tedy máme

Klíčové slovo:	V	E	N	U	S	E	V	E	N	U	S	E
Zpráva:	**p**	**o**	**l**	**y**	**a**	**b**	**e**	**c**	**e**	**d**	**n**	**i**
Kryptogram:	**K**	**S**	**Y**	**S**	**S**	**F**	**Z**	**G**	**R**	**Y**	**F**	**M.**

Je jasné, že takováto šifrovací metoda staví Mr. X před podstatně větší problémy, než je tomu při monoabecedním šifrování. Četnost písmen je daleko rovnoměrnější, což lze poznat i na našem krátkém příkladu. Např. písmeno zprávy **e** bylo zašifrováno do **Z** a **R**, písmeno kryptogramu **S** vzniklo ze tří různých písmen zprávy (**o, y, a**).

6) Kryptoanalýza

Kerckhoffuv princip: Spolehlivost kryptosystému nesmí záviset na utajení algoritmu. Spolehlivost je založena pouze na utajení klíce.

Kasiského test: Pro určení délky klíčového slova ve Viegenerově polyabecední šifře. Test je založen na následující myšlence: vyskytují-li si ve zprávě dvě posloupnosti stejných

písmen (např. v němčině slovo **ein**), mohou obecně odpovídající posloupnosti v kryptogramu dopadnout různě. Jsou-li ale obě počáteční písmena posloupností zašifrována pomocí téhož písmene klíčového slova, jsou i obě písmena kryptogramu stejná. V tomto případě bude také druhé písmeno posloupnosti v zprávě zašifrováno pomocí téhož písmene klíčového slova; tedy obdržíme i v kryptogramu stejné písmeno. To tedy znamená: Budou-li obě počáteční písmena posloupností zprávy zašifrována pomocí téhož písmene klíčového slova, pak sestávají obě posloupnosti v kryptogramu ze stejných písmen.

Kdy může nastat případ, že dvě písmena jsou zašifrována pomocí téhož písmene klíčového slova? Právě tehdy, když se klíčové slovo mezi tato písmena *n*-krát vejde pro vhodné přirozené *n*.

Příklad:

Posloupnost	Odstup	Rozklad na součin prvočinitelů odstupu
JTD	50	$2 \cdot 5 \cdot 5$
VIQM	265	$5 \cdot 53$
TDMHZGNMWK	75	$2 \cdot 3 \cdot 3 \cdot 5$

MWK	75	$3 \cdot 5 \cdot 5$

Největší společný faktor je 5. Optimistický kryptoanalytik by mohl říci, že *že délka klíčového slova je 5* (ve skutečnosti funguje Kasiského test v praxi velmi dobře). Pokud je ale kryptoanalytik opatrný, mluví pouze o silné indicii pro délku klíčového slova 5.

Friedmanův test: Pro odhad délky klíčového slova.

Představme si nejprve libovolnou posloupnost písmen délky n. Buď n1 počet písmen a, n2 počet písmen b, . . . , n26 počet písmen z.

Zajímáme se o počet dvojic, kdy jsou obě písmena rovna **aa**. (Nepožadujeme, aby se uvažované dvojice skládaly ze za sebou následujících písmen.) Pro počet prvního **a** máme právě n_1 možností, pro výběr druhého **a** zbývá $n_1 - 1$ možností. Protože nezáleží na pořadí písmen, je počet hledaných dvojic roven $n_1 * (n_1 - 1)/2$

Je tedy počet dvojic, kdy jsou obě písmena stejná, roven

$$\frac{n_1 \cdot (n_1 - 1)}{2} + \frac{n_2 \cdot (n_2 - 1)}{2} + \cdots + \frac{n_{26} \cdot (n_{26} - 1)}{2} = \sum_{i=1}^{26} \frac{n_i \cdot (n_i - 1)}{2}.$$

Šance obdržení dvojice složené ze stejných písmen je určena následujícím výrazem:

$$\mathbf{I} = \frac{\sum_{i=1}^{26} n_i \cdot (n_i - 1)}{n \cdot (n - 1)}$$

a nazývá se Friedmanův **index koincidence**. Friedman sám značil toto číslo jako κ, proto se občas pro metodu, kterou v dalším předvedeme, používá název **kappa-test**.

V běžném textu se vyskytuje písmeno **a** s pravděpodobnpstí p_1, písmeno **b** s pravděpodobnpstí $p_2, \ldots,$ písmeno **z** s pravděpodobnpstí p_{26}.

Tedy pravděpodobnost toho, že obě písmena jsou si rovna je $p_1^2 + p_2^2 + \cdots + p_{26}^2$

Pro text v německém jazyce I = 0,0762

Pro náhodný text p_i=1/26 I = 0,0385

Monoabecední šifrování I= 0,0762

Předpokládejme, že klíčové slovo má délku *l* a skládá se z navzájem různých písmen.

Rozepišme náš kryptogram do *l* sloupců. Pak se v prvním sloupci nacházejí písmena číslo 1, *l*+1, 2*l*+1, . . . , tedy všechna ta písmena, která byla zašifrována pomocí prvního písmene klíčového slova. A tedy je v každém sloupci I= 0,0762.

Písmeno S_i klíč. slova	S_1	S_2	S_3	S_l
	1	2	3	l
	$l+1$	$l+2$	$l+3$	$2l$
	$2l+1$	$2l+2$	. . .	$3l$
	$3l+1$	. . .		
	. . .			

počet dvojic písmen, která se nacházejí *v tom samém sloupci je* roven (ne v konkrétním, ale ve stejném)

$$n \cdot (\frac{n}{l} - 1)/2 = \frac{n \cdot (n-l)}{2l}.$$

počet dvojic písmen, která se nacházejí *v různých sloupcí je* roven

$$n \cdot (n - \frac{n}{l})/2 = \frac{n^2 \cdot (l-1)}{2l}.$$

Na základě výše zmíněného pak máme, že očekávaný počet A dvojic stejných písmen je roven

$$A = \frac{n \cdot (n-l)}{2l} \cdot 0.0762 + \frac{n^2 \cdot (l-1)}{2l} \cdot 0.0385.$$

Pravděpodobnost, že získáme dvojici složenou ze stejných písmen, je rovna

$$\frac{A}{n \cdot (n-1)/2)} = \frac{(n-l)}{l \cdot (n-1)} \cdot 0.0762 + \frac{n \cdot (l-1)}{l \cdot (n-1)} \cdot 0.0385,$$

tj. po úpravě

$$\frac{A}{n \cdot (n-1)/2)} = \frac{1}{l \cdot (n-1)} \cdot [0.0377 \cdot n + l \cdot (0.0385 \cdot n - 0.0762)].$$

Zároveň víme, že index koincidence **I** je aproximací tohoto čísla; proto platí

$$\mathbf{I} = \frac{0.0377 \cdot n}{l \cdot (n-1)} + \frac{0.0385 \cdot n - 0.0762}{n-1}.$$

Vyjádříme-li si z výše uvedeného vztahu l, získáme důležitou Friedmanovu formuli pro délku klíčového slova:

$$\boxed{l \approx \frac{0.0377 \cdot n}{(n-1) \cdot \mathbf{I} - 0.0385 \cdot n + 0.0762}.}$$

Odtud pro konkrétní text vypočteme I a dostaneme odhad l.

Úloha: Vytvořte dešifrátor Viegenerovy šifry. Dešifrátor bude interaktivní, s použitím Fridmanova testu a Kaiského testu určí l a následně v každém sloupci dešifruje Caesorovu šifru.

Asymetrické šifrovací systémy

Budeme předpokládat, že každý účastník **T** má *dvojici* klíčů, a to

- **veřejný klíč E** = $\mathbf{E}_T$ k zašifrování;
- **soukromý (tajný) klíč D** = $\mathbf{D}_T$ k dešifrování;

které se vyznačují následující vlastností: *Ze znalosti klíče* $\mathbf{E}_T$ *nelze zjistit soukromý klíč* $\mathbf{D}_T$. Kryptosystém s touto vlastností se nazývá **asymetrický kryptosystém**.

Pokud navíc předpokládáme, že pro každou zprávu M platí

$$\mathbf{D}(\mathbf{E}(M)) = M,$$

mluvíme o **asymetrickém šifrovacím systému**. Asymetrický kryptosystém se nazývá **asymetrické podpisovací schéma**, pokud pro každou zprávu M lze pomocí veřejného klíče **E** prověřit, zda se k sobě M a $\mathbf{D}(M)$ hodí.

Všechny veřejné klíče jsou uloženy ve veřejně dostupném souboru (podobnému telefonnímu seznamu), zatímco soukromé klíče jsou tajné tj.

známé pouze jejich vlastníkům.

Šifrování a dešifrování pomocí asymetrického šifrovacího systému probíhá ve 3 krocích:

1. Chce-li **A** zaslat **B** zprávu M , pak
   najde veřejný klíč $\mathbf{E_B}$ pro **B**,
   - zašifruje zprávu M pomocí klíče $\mathbf{E_B}$ a
   - odešle $\mathbf{E_B}(M)$ k **B.**

2. **B** může kryptogram $\mathbf{E_B}(M)$ dešifrovat, protože zná jako jediný tajný klíč $\mathbf{D_B}$:

$$\mathbf{D_B}(\mathbf{E_B}(M)) = M.$$

3. Žádný jiný účastník nemůže ($\mathbf{E_B}(M)$ rozluštit, protože podle předpokladu ze znalosti ($\mathbf{E_B}$ a ($\mathbf{E_B}(M)$
   nelze získat znalost o $\mathbf{D_B}$.

Konkrétněji: Všichni uživatelé systému, kteří si přejí navzájem komunikovat, používají tentýž šifrovací algoritmus e a tentýž dešifrovací algoritmus d. Každý uživatel U_i má dvojici klíčů (K_i, L_i) tak, že pro každou možnou zprávu M platí identita

$$d(e(M, K_i), L_i) = M,$$

kde K_i je zveřejněn a uložen ve veřejném souboru; L_i zůstane utajem a mluvíme o něm jako o *soukromém klíči*; K_i se nazývá *veřejný klíč*. Pokud chce jiný uživatel U_j odeslat uživateli U_i zprávu M , postupuje následovně.

(a) Uživatel U_j najde veřejný klíč K_i uživatele U_i ve veřejném souboru.

(b) Uživatel U_j odešle kryptogram

$$C = e(M, K_i)$$

k uživateli U_i veřejným kanálem.

Bezpečnost systému závisí na funkcích e a d, které mají následující vlastnosti.

Vlastnost 1 Známe-li M a K, mělo by být snadné vypočíst C = e(M, K).

Vlastnost 2 Je-li dán pouze kryptogram C, není snadné výpočetně najít M .

Vlastnost 3 Je-li znám kryptogram C a tajný klíč Li, je snadné určit zprávu M.

Vlastnost 4 Mělo by být snadné generovat náhodné dvojice veřejný/soukromý klíč (Ki, Li).

RSA-algoritmus (1977)

Budeme potřebovat některé pojmy a tvrzení z teorie čísel.

Eulerova funkce $\phi(n)$ je počet všech přirozených čísel *k* takových, že 1 ≤ k ≤ n a NSD*(k,n)=1*, tedy *k* a *n* jsou nesoudělná čísla. Ihned z definice jsou patrné následující vlastnosti:

- $\phi(1) = 1$,
- $\phi(p) = p\text{-}1$ pro *p* prvočíslo,
- $\phi(p^m) = (p\text{-}1)p^{m\text{-}1}$ pro *p* prvočíslo a *m* kladný celý exponent.
- *Funkce φ je multiplikativní, tj. pro a, b* ∈ N*, NSD*(*a, b*) = 1*, je*

$$\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b).$$

Věta 1.15. *Je-li* $n = p_1^{\alpha_1} \cdot \ldots p_k^{\alpha_k}$ *kanonický rozklad čísla* n*, pak platí*

$$\phi(p_1^{\alpha_1} \cdot \ldots p_k^{\alpha_k}) = \phi(p_1^{\alpha_1}) \cdot \ldots \cdot \phi(p_k^{\alpha_k}),$$

přičemž

$$\phi(p_i^{\alpha_i}) = p_i^{\alpha_i} - p_i^{\alpha_i - 1}.$$

Věta 4.1 (Euler) *Nechť* $\mathrm{NSD}(c, m) = 1$. *Pak platí* $c^{\phi(m)} = 1 \ mod \ m$.

Věta 4.2 (Fermat) *Nechť* p *je prvočíslo* $\mathrm{NSD}(c, p) = 1$. *Pak platí* $c^{p-1} = 1 \ mod \ p$.

Při použití RSA-algoritmu každý účastník systému používá dvě (čtyři) 100-místná prvočísla. Kolik jich máme k dispozici? Použitím prvočíselné funkce π, která udává počet prvočísel menších než dopředu zvolené číslo n a odhaduje se pomocí odhadu $\pi(n)=n/ln(n)$.

Kolik prvočísel je tedy mezi 100 místnými čísly?

Jaká je pravděpodobnost, že si dva uživatelé vyberou stejný klíč? Tedy stejné prvočíslo?

Úloha: Pomocí Fermatovi věty vytvořte pravděpodobnostní test, že náhodné číslo p (100 místné) je prvočíslo. Uveďte algoritmus a naprogramujte. Test obsahuje k ověření, že p je prvočíslo.

Úloha: Vytvořte algoritmus na výpočet Eulerovy funkce. Naprogramujte.

Postup při šifrování RSA-algoritmu

1. Najděme dvě *velká* prvočísla p a q a položme $n = p \cdot q$.
2. Najděme *velké a náhodné* přirozené číslo d tak, že je nesoudělné s číslem $(p-1) \cdot (q-1)$.
3. Vypočtěme jediné přirozené číslo e ležící v oboru hodnot $1 \leq e \leq (p-1) \cdot (q-1)$ ze vztahu

$$e \cdot d = 1 \bmod (p-1) \cdot (q-1).$$

4. Zveřejněme veřejný klíč, který se skládá z dvojice přirozených čísel (e, n).
5. Reprezentujme zprávu M jako přirozené číslo z intervalu $\{1, \ldots, n\}$; rozdělme zprávu M do bloků, je-li příliš velká.
6. Zakódujme M do kryptogramu C dle předpisu

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

7. Dešifrujme pomocí soukromého klíče d a předpisu

$$D = C^d \bmod n.$$

RSA algoritmus se používá převážně pro elektronické podpisování.

Důkaz správnosti RSA algoritmu:

ed je kongruentní s 1 mod (*p-1)*(q-1)* a tedy je také kongruentní s 1 *mod (p-1)* to pak implikuje *ed=1+c(p-1)* pro nějaké c.

A jelikož $ed \equiv 1 \pmod{p-1}$ a $ed \equiv 1 \pmod{q-1}$, díky malé Fermatově větě platí, že

$$(m^e)^d \equiv m^{1+c(p-1)} \equiv m^1 (m^{p-1})^c \equiv m \cdot 1^c \equiv m \pmod{p}$$

a zároveň

$$(m^e)^d \equiv m \pmod{q}$$

Jelikož p a q jsou různá prvočísla, pomocí čínské věty o zbytcích je dáno

$$(m^e)^d \equiv m \pmod{pq}$$

Tudíž

$$c^d \equiv m \pmod{n}$$

Další algoritmy založené na kongruenci:

1. Ruksaková metoda – Merkl Hellman
2. Kvadratická kongruence - Rabinův systém s veřejným klíčem
3. Diskrétní logaritmus – Diffie Hellman

Rabinův systém s veřejným klíčem (1979)

Uveďme příklad systému s veřejným klíčem, o kterém lze ukázat, že jeho složitost je ekvivalentní s problémem faktorizace.

Každý uživatel systému vybere dvojici (p, q) velkých různých prvočísel, které uchová v tajnosti. Zároveň si vybere přirozené číslo $B < N = p \cdot q$.

Veřejný klíč bude dvojice (B, N), *soukromý klíč* bude faktorizace (p, q) čísla N.

Šifrovací funkce e zprávy M, kde M je reprezentovatelná jako přirozené číslo z definičním oboru $1, \ldots, N\text{-}1$
(v případě potřeby se zpráva rozparceluje na více bloků), je
$e(M) = M \cdot (M + B) \pmod N$.

Je-li C výsledný kryptogram, pak dešifrovací problém je nalézt M tak, že
$M^2 + B \cdot M = C \pmod N$.

Tvrzení 7.21 *Za předpokladu, že jak p tak q jsou kongruentní s 3 modulo 4, lze dešifrovací proceduru provést v polynomiálním čase.*

Příklad: p=7,q=11, n=77. B=2. Zašifruj M=20. Poté dešifruj zpět.

Úloha: Napište algoritmus pro kódování a dekódování Rabinova systému. Se známým veřejným i neveřejným klíčem. Naprogramujte.Prezentujte.

Diskrétní logaritmus

V tomto odstavci se budeme zabývat dalším příkladem toho, co lze v současnosti považovat za další jednosměrnou funkci. Definujme tedy *diskrétní logaritmus*. Uvažme n tak, že má primitivní kořen a, tj. platí $(a, n) = 1$ a pro všechna d, $1 \leq d \leq \varphi(n) - 1$ je $a^d \neq 1$. Pokud pro x, $1 \leq x \leq \varphi(n) - 1$, platí

$$y = a^x \bmod n, \tag{9.1}$$

říkáme, že x je *diskrétní logaritmus z y při základu a modulo n* a píšeme $x = \log_a y \bmod \varphi(n)$. Důležitost toho, aby a bylo primitivní kořen, spočívá v tom, že nám tím garantuje pro každé y, $1 \leq y \leq n-1$, $(y, n) = 1$ existenci jediného takového x tj. diskrétní logaritmus je korektně definovaná funkce.

Platí pak následující tvrzení.

Tvrzení 9.1 *Exponenciální funkce definovaná v 9.1 je jednosměrná, tj. provedení umocňování je snadné, ale logaritmování je obtížné.*

Bezpečná distribuce klíčů

Jeden za základních problémů klasické kryptografie je způsob bezpečné distribuce klíčů. Jaká je jistota, že když nemůžeme bezpečně přenášet zprávy, že klíče budou bezpečné?
Diffie a **Hellman** navrhli elegantní způsob vyřešení tohoto problému. Závisí na jednosměrné povaze problému diskrétní logaritmizace. Uvažme seznam uživatelů $(U_i : 1 \leq i \leq N)$, kteří spolu chtějí navzájem komunikovat. Buď p velké prvočíslo (o mnoho větší než N) a nechť a je primitivní kořen z p. Typický uživatel U_i si vygeneruje, nezávisle na ostatních uživatelích, pseudonáhodné číslo X_i v rozmezí od 1 do $p-1$ a ponechá ho v tajnosti. Zároveň prohlásí za svůj *veřejný klíč* přirozené číslo

$$Y_i = a^{X_i} \bmod p. \tag{9.2}$$

Přejí-li si uživatelé U_i a U_j komunikovat soukromě, použijí za svůj klíč číslo

$$K_{ij} = a^{X_i X_j} \bmod p. \tag{9.3}$$

Uživatel U_i si vypočte K_{ij} tím, že si najde ve veřejně přístupném souboru Y_j a použije vztah

$$K_{ij} = Y_j^{X_i} \bmod p. \tag{9.4}$$

Podobně to provede i uživatel U_j

$$K_{ij} = Y_i^{X_j} \bmod p. \tag{9.5}$$

Je-li p prvočíslo zapsané v dvojkové soustavě pomocí b bitů, je na umocnění potřeba nejvýše $2b$ násobení modulo p. Naopak však Mr. X potřebuje více než polynomiální počet operací, aby byl schopen napadnout systém. Poznamenejme, že doposud není známo, zda prolomení tohoto systému je ekvivalentní výpočtu diskrétního logaritmu.

Zde tedy oba účastníci na základě svého soukromého klíče X_i a veřejného klíče Y_j vygenerují společný unikátní klíč pro účastníka i a j. Tento klíč je ovšem neznámý těm, co znají pouze veřejné klíče Y_i a Y_j.

Tento klíč pak mohou tito dva účastníci používat v rychlé komunikaci pomocí symetrického šifrovacího algoritmu. Např. Viegenerovy šifry s dlouhým klíčem.

Šifrovací systém bez klíče

Uvažme následující šifrovací systém. Předpokládejme, že uživatel A chce poslat zprávu uživateli B a že tato zpráva je reprezentovatelná jako přirozené číslo v intervalu $\{0, 1, 2, \ldots, p-1\}$, kde p je velké prvočíslo. Uživatel A si vybere přirozené číslo a nesoudělné s $p-1$. Totéž provede uživatel B pro číslo b. Komunikace mezi A a B sestává ze tří kroků. Nejdřív A pošle B celé číslo

$$C = M^a \bmod p. \tag{9.6}$$

Potom B odešle A číslo

$$D = C^b \bmod p. \tag{9.7}$$

Pak A určí celé číslo a' tak, že $a \cdot a' = 1 \bmod p-1$ a zašle B číslo

$$E = D^{a'} \bmod p. \tag{9.8}$$

Následovně příjemce B dešifruje zprávu podle předpisu

$$F = E^{b'} \bmod p, \tag{9.9}$$

kde b' je celé číslo tak, že $b \cdot b' = 1 \bmod p - 1$ tj. existuje celé číslo t splňující $b \cdot b' = 1 + t \cdot (p-1)$. Ukažme, že $F = M$. Totiž

$$\begin{aligned} F &= E^{b'} = \left(D^{a'}\right)^{b'} = D^{a'b'} = C^{a'b'b} = \left(C^{a'}\right)^{b'b} = \left(C^{a'}\right)^{1+t(p-1)} \bmod p \\ &= C^{a'} \cdot \left(C^{a'}\right)^{t(p-1)} = C^{a'} = \left(M^{a}\right)^{a'} = M^{aa'} = M \bmod p. \end{aligned} \tag{9.10}$$

Vážnou nevýhodou tohoto systému je 3-násobná časová náročnost.

$c^{t(p-1)} = 1 \bmod p$ pro libovolné c<p, dle Fermatovy věty.

a a b by měli být zvoleny tak, aby a’ nebylo rovno b.