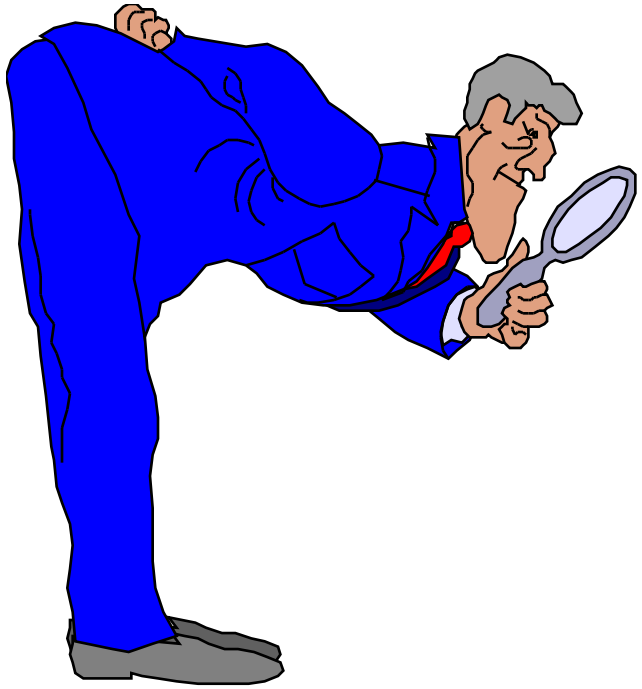
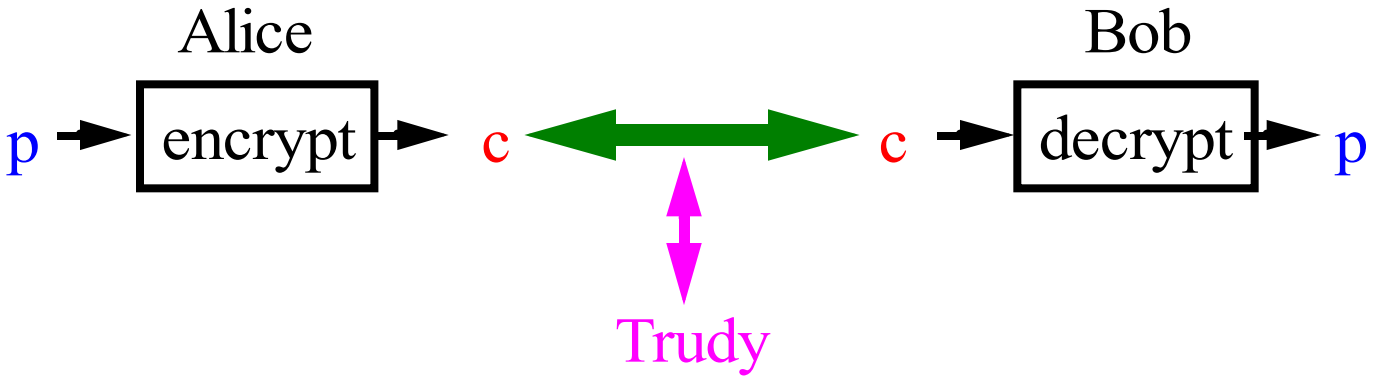
Úvod do kryptografie



- Secrecy
- Ciphers
- Secret Key Cryptography
- Key Exchange
- Public Key Cryptography
- Digital Signatures

Utajení

- Scénář: Alice chce poslat zprávu (přímý text, plaintext p) Bobovi. Komunikační kanál není chráněný a může být odposloucháván Trudy. Pokud mají Alice a Bob předem domluvený způsob šifrování zprávy (šifru, cipher), pak může být zpráva odeslána utajeně (šifrovaný text, ciphertext c)



Jakou použít šifru?
Jaká je složitost šifrování/dešifrování?
Jaká je velikost šifry vzhledem k otevřenému textu?
Pokud spolu Alice a Bob předtím nekomunikovali, jakým způsobem se dozvědí o šifře ?

Proč vlastně šifrujeme data?

- Potřeba utajovat určité informace je stará jak lidstvo
- o kryptologii hovoříme až v případě, kdy všichni používají stejný vyjadřovací prostředek (např. písmo)
- Od vzniku Internetu se mnohonásobně zvyšuje počet připojených počítačů
- V otevřených sítích jako je INTERNET je jednoduché jakoukoliv informaci ODPOSLECHNOUT a následně i ZNEUŽÍT!!
- v poslední době dochází k masovému využívání šifrování z různých důvodů
- Proto vznikla potřeba skrýt citlivé informace před nepovolanými osobami

Co to vlastně je kryptografie?

- Kryptosystém je systém umožňující šifrování a dešifrování zpráv.
- Šifrování, neboli kryptografie je transformace dat do nečitelné formy.

  Důvod - ochránit důvěrné a osobní informace znemožněním jejich čitelnosti těmi, komu nejsou určeny
- Dešifrování je opačný postup, tedy transformace šifrovaných dat do jejich původní (srozumitelné) podoby

Šifrování a dešifrování vyžaduje užití nějaké tajné informace, obvykle označované jako klíč

Šifrování – základní pojmy

- Kryptografie -věda o tvorbě šifer
- Kryptoanalýza - věda o prolamování šifer
- Kryptologie – věda o šifrování, zahrnuje kryptografii a kryptoanalýzu
- Otevřený text (plaintext) - originální tvar dat (to co má být zašifrováno)
- Šifrovaný text (ciphertext) – zašifrovaný tvar zprávy
- Šifrování (kryptování, enkryptování enciphering) – proces přeměny otevřeného textu na šifrovaný text
- Dešifrování (dekryptování, deciphering) – přeměna šifrovaného textu na otevřený text

Hodnocení kryptosystémů

- Různé úhly pohledu – rychlost výpočtu, bezpečnost, snadnost implementace

Základní zásady:

- šifrováním by neměl narůstat objem dat, pokud naroste, tak jen o konstantní velikost
- implementace by měla být únosně složitá
- rozumná implementace by měla být přiměřeně rychlá
- šifra by neměla obsahovat žádná omezení na data na která bude použita
- chyby při šifrování by se neměli nepřiměřeně šířit

Dělení šifer

Z hlediska zpracování zprávy:

- Blokové šifry – pracují s celými bloky dat (obvykle 8-128 bytů)
- Proudové šifry (streamové) - pracují s jednotlivými bitu zprávy zvlášť,
  - jsou považovány za méně bezpečné
  - jsou pomalejší než šifry blokové

Z hlediska šifrování :

- Symetrické šifry – odesilatel i příjemce sdílí jedno tajemství (klíč) nutné k šifrování a zašifrování zprávy
- Asymetrické šifry – odesilatel a příjemci šifrují a dešifrují zprávu různými klíči, nemusí spolu sdílet žádné tajemství.
  - Nevýhoda: je o několik řádů pomalejší než symetrická kryptografie

Historie šifrování

Steganografie – ukrývání zprávy jako takové (tajné inkousty, vyrývání zprávy do dřevěné tabulky zalité voskem, apod.)

Použití kódů – pro každou činnost se vytvoří kódové slovo

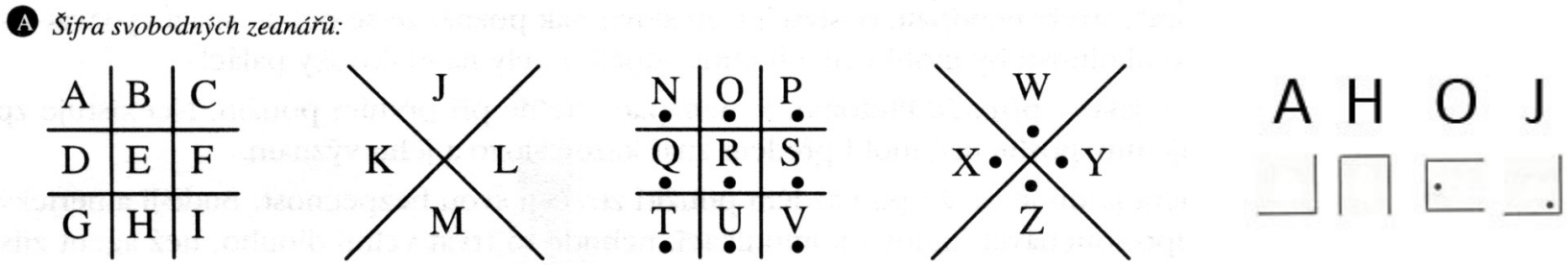
Substituční šifry

Obecně spočívá v nahrazení každého znaku zprávy jiným znakem podle nějakého pravidla. Nejstarší popis šifry Kámasútra (4.stol.). Nevýhoda - snadné prolomení šifry.

- **Posun písmen** (Caesarova šifra) – každé písmeno zprávy je posunuté o pevný počet pozic

  zaměň znak *a* za *d*
  zaměň znak *b* za *e*
  ...
  zaměň znak *z* za *c*

- **Tabulka záměn –** záměna znaku za jiný, bez jakékoliv souvislosti popř. na základě znalosti hesla



- *Příklad tabulky záměny, s použitím slova **VESLO** jako klíče:*

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

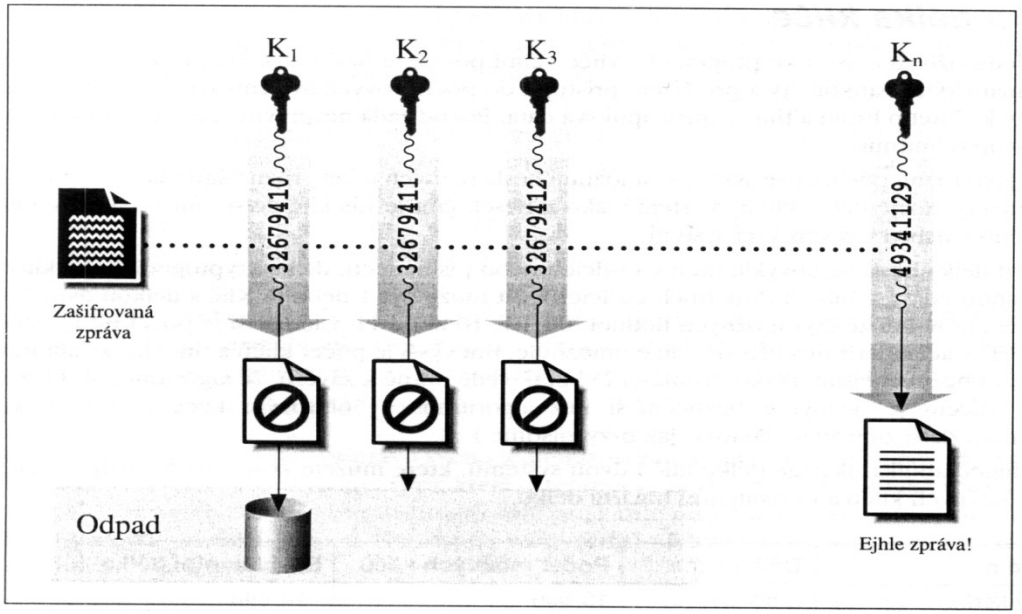
nebo

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

Narušení kódu

- Metoda hrubé síly - zkouší se všechny možné klíča a hledá se smysluplná zpráva.

  Nevýhoda: časově náročná metoda, substituční šifry jí dokáží čelit (volbou vhodně velkého klíče)



Narušení kódu

- Pokud Trudy použije statistiku jazyka ve kterém je šifrovaná zráva, může jednoduše prolomit monoalfabetickou substituční šifru.

  Např. v angličtině se nejčastěji vyskytují:

  - *znaky : e, t, o, a, n, i, ...*
  - bigrams: th, in, er, re, an, ...
  - trigrams: the, ing, and, ion, ...

- Využití frekvenční analýzy jazyka v kryptoanalýze poprvé zmiňuje v 9. století arabský filosof al-Kindi

Příklad (S. Singh, The Code Book, 1999, v češtině Kniha kódů a šifer, 2003)

- Šifra
- PCQ VMJYPD LBYK LYSO KBXBJXWXV BXV ZCJPO EYPD KBXBJYUXJ LBJOO KCPK. CP LBO LBCMKXPV XPV IYJKL PYDBL, QBOP KBO BXV OPVOV LBO LXRO CI SX'XJMI, KBO JCKO XPV EYKKOV LBO DJCMPV ZOICJO BYS, KXUYPD: 'DJOXL EYPD, ICJ X LBCMKXPV XPV CPO PYDBLK Y BXNO ZOOP JOACMPLYPD LC UCM LBO IXZROK CI FXKL XDOK XPV LBO RODOPVK CI XPAYOPL EYPDK. SXU Y SXEO KC ZCRV XK LC AJXNO X IXNCMJ CI UCMJ SXGOKLU?'

OFYRCDMO, LXROK IJCS LBO LBCMKXPV XPV CPO PYDBLK

Frekvenční analýza

- Identifikace často se vyskytujících písmen, bigramů, trigramů
- PCQ VMJYPD LBYK LYSO KBXBJXWXV BXV ZCJPO EYPD KBXBJYUXJ LBJOO KCPK. CP LBO LBCMKXPV XPV IYJKL PYDBL, QBOP KBO BXV OPVOV LBO LXRO CI SX'XJMI, KBO JCKO XPV EYKKOV LBO DJCMPV ZOICJO BYS, KXUYPD: 'DJOXL EYPD, X LBCMKXPV XPV CPO PYDBLK Y BXNO ZOOP JOACMPLYPD LC UCM LBO IXZROK CI FXKL XDOK XPV LBO RODOPVK CI XPAYOPL EYPDK. SXU Y SXEO KC ZCRV XK LC AJXNO X IXNCMJ CI UCMJ SXGOKLU?'

  OFYRCDMO, LXROK IJCS LBO LBCMKXPV XPV CPO PYDBLK
- První odhad: LBO je THE

Frekvenční analýza

- Předpokládejme, že pokud LBO reprezentuje THE můžeme nahradit L za T, B za H, and O za E a dostaneme
- PCQ VMJYPD THYK TYSE KHXHJXWXV HXV ZCJPE EYPD KHXHJYUXJ THJEE KCPK. CP THE THCMKXPV XPV IYJKT PYDHT, QHEP KHO HXV EPVEV THE LXRE CI SX'XJMI, KHE JCKE XPV EYKKOV THE DJCMPV ZEICJE HYS, KXUYPD: 'DJEXT EYPD, ICJ X LHCMKXPV XPV CPE PYDHLK Y HXNE ZEEP JEACMPTYPD TC UCM THE

  IXZREK CI FXKL XDEK XPV THE REDEPVK CI XPAYEPT EYPDK. SXU Y SXEE KC ZCRV XK TC AJXNE X IXNCMJ CI UCMJ SXGEKTU?'

  EFYRCDME, TXREK IJCS THE LHCMKXPV XPV CPE PYDBTK

Řešení

- **Kód**

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

- **Původní text:** Now during this time Shahrazad had borne King Shahriyar three sons. On the thousand and first night, when she had ended the tale of Ma'aruf, she rose and kissed the ground before him, saying: 'Great King, for a thousand and one nights I have been recounting to you the fables of past ages and the legends of ancient kings. May I make so bold as to crave a favour of your majesty?' Epilogue, Tales from the Thousand and One Nights

Aditivní šifry

- Vigenerova šifra - speciální případ polyalfabetické šifry. Základem šifrování je Vigenerův čtverec (otevřený text, následovaný 26 šifrovými abecedami, z nichž každá je oproti předchozí posunutá o jeden znak.

Šifrování se provádí tak, že každý znak šifrujeme podle jiné abecedy (jiného řádku). Jaký řádek čtverce použijeme je určeno klíčem (heslem)

	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
1	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
2	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
3	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
4	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
5	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
6	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
7	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
8	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
9	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
10	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	J
11	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
12	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
13	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
14	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
15	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
16	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
17	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
18	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
19	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
20	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
21	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
22	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
23	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
24	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	X
25	Z	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
26	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

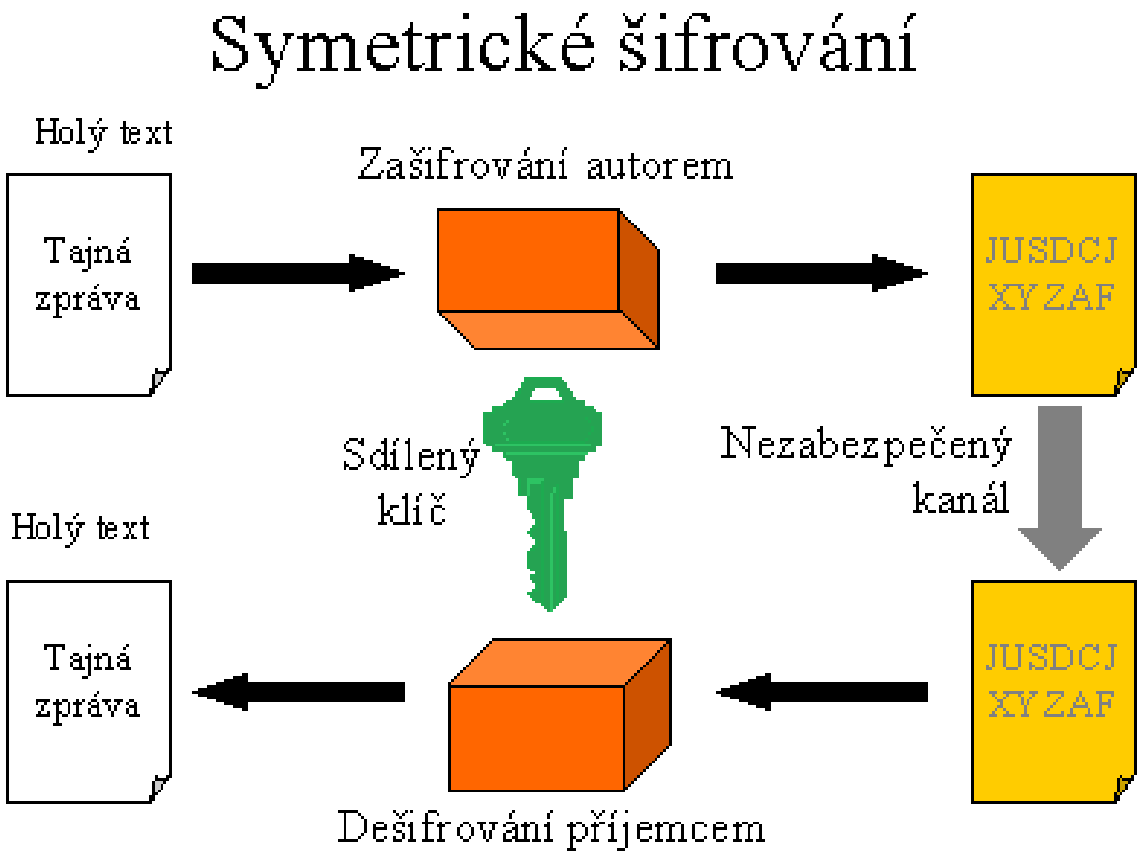
Příklad šifrování textu „Zlato je uloženo v jeskyni“ s klíčem (heslem) „POKLAD“

P	O	K	L	A	D	P	O	K	L	A	D	P	O	K	L	A	D	P	O	K	L
z	l	a	t	o	j	e	u	l	o	z	e	n	o	v	j	e	s	k	y	n	i
O	Z	K	E	O	M	T	I	V	Z	Z	H	C	C	F	U	E	V	Z	M	X	T

Symetrické šifrování

- Je nejpoužívanějším typem šifrovacího algoritmu
- Používá stejný šifrovací klíč k šifrování i dešifrování - což je jeho největší slabina
- Je velmi rychlý a používá se při velkém množství dat
- Klíč se musí dostat od odesilatele k adresátovi bezpečným kanálem (cestou), aby adresát mohl zprávu dešifrovat
- Pokud takový bezpečný kanál existuje, je často jednodušší zprávu nešifrovat a poslat ji rovnou tímto kanálem.

Symetrické šifrování

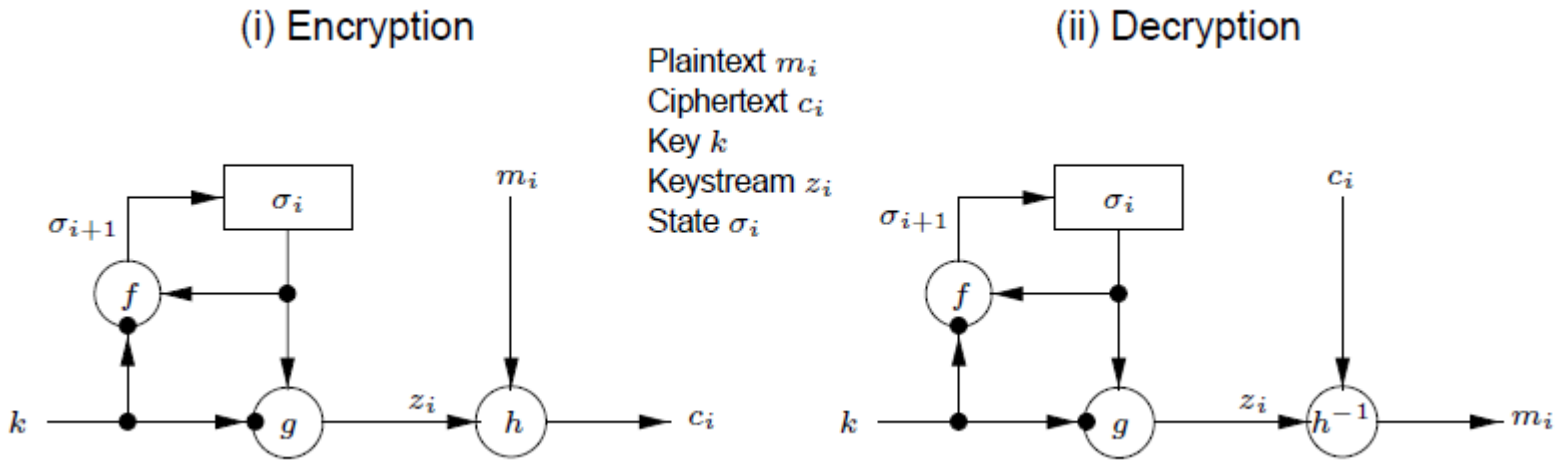


Proudové symetrické šifry

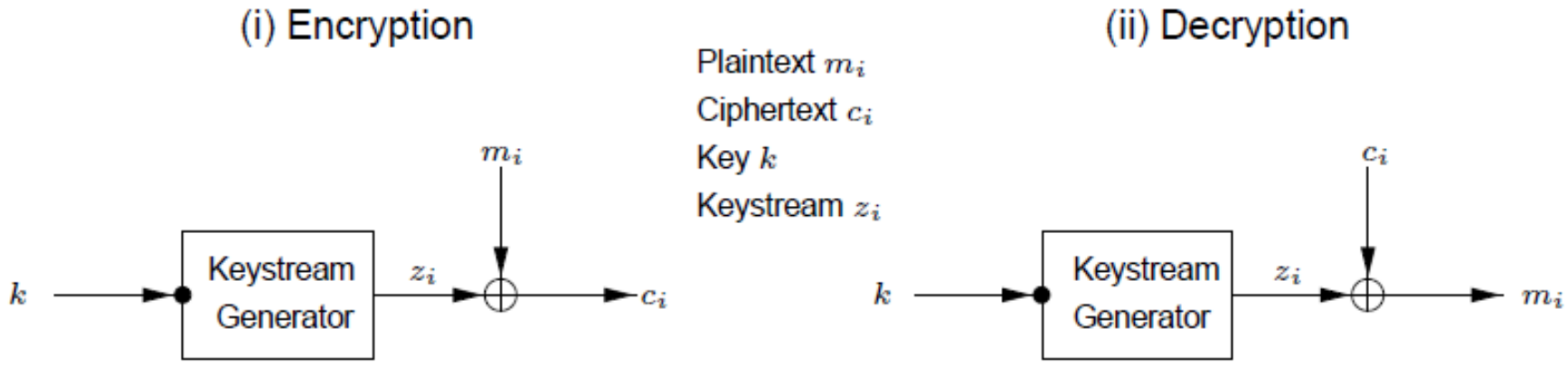
- n-bitový klíč K je použit pro generování proudu bitů delšího klíče (keystream), ten je použit k šifrování informace (provádí se operace XOR mezi bity keystreamu a vstupního textu.

Proudové šifry se dělí na:

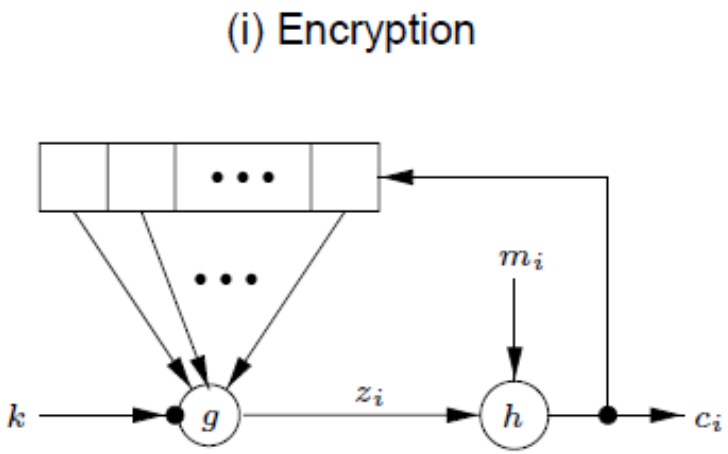
- synchronní - proudový klíč (keystream) je generován nezávisle na vstupním a šifrovaném textu.

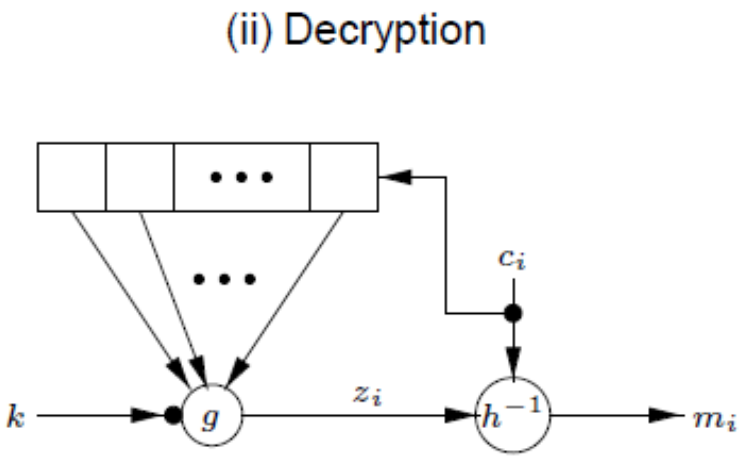


Binárně-aditivní proudová šifra – synchronní šifra, ve které jsou vstupní text ,šifrovaný text a keystream binární čísla a jako výstupní funkce h je použita operace XOR

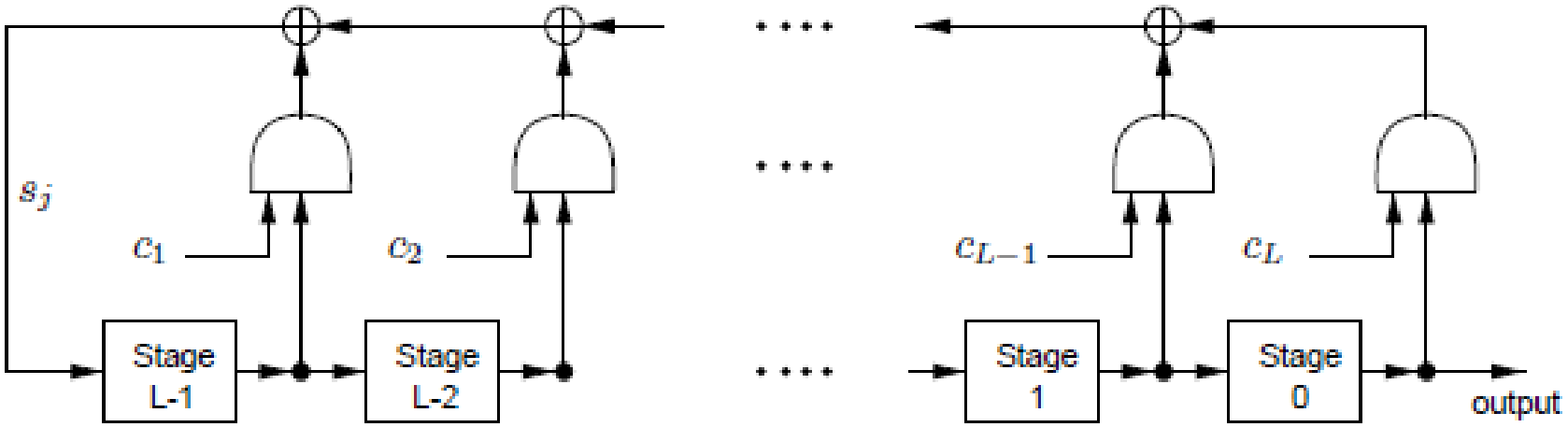


• šifry s vlastní synchronizací - proudový klíč (keystream) je funkcí klíče a pevného počtu bitů šifrovaného textu





Proudové šifry používají ke generování proudového klíče posuvné registry (LSFR – Linear Feedback Shift Register)



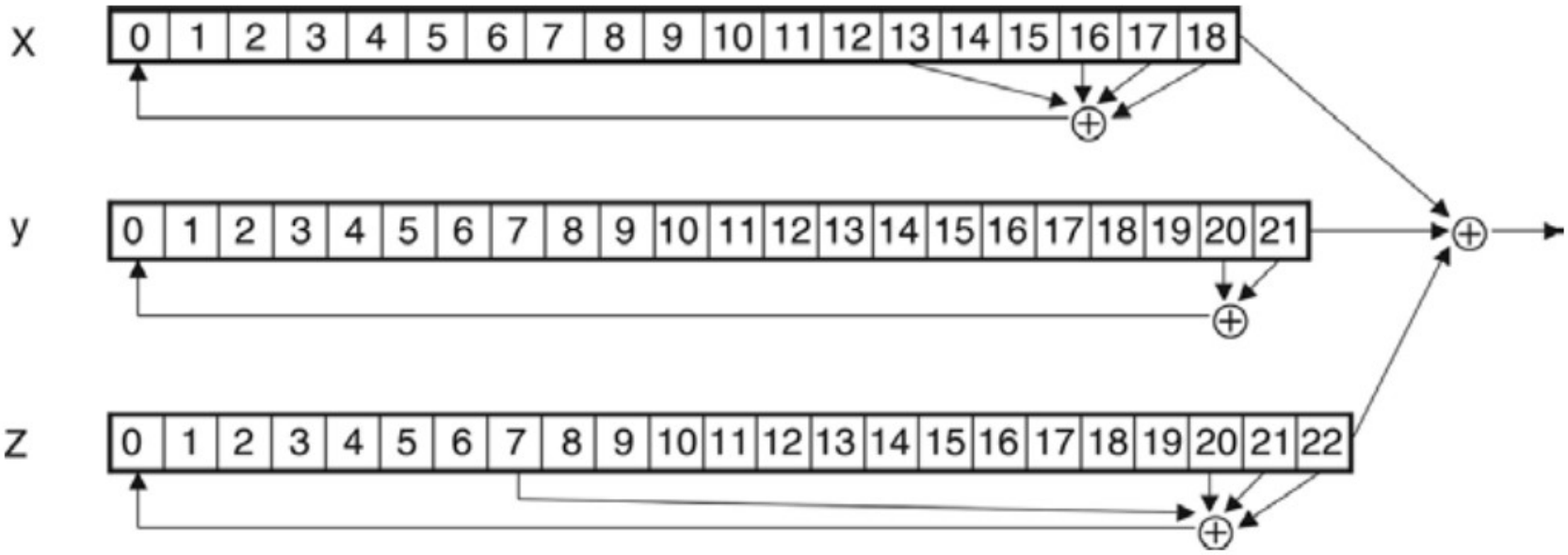
Šifra A5

A5 je proudová šifra vyvinutá k šifrování GSM hovoru mezi mobilní stanicí a základnovou stanicí BTS (Base Transceiver Station). Hovor v síti operátora, tj. od BTS přes BSC (Base Station Controller) až do ústředny MSC (Mobile Switching Center), není dále šifrován, takže ho lze odposlouchávat.
Existuje ve dvou variantách, které jsou na bázi proudových šifer: A5/1 a A5/2.

Šifra produkuje vždy 228 bitů proudu klíče, 114 bitů se používá pro šifrování komunikace od telefonu k základové stanici a 114 bitů pro šifrování komunikace v opačném směru.
Tajný klíč je uložen na SIM kartě telefonu.
Při každém spojení se sítí je z tajného klíče na SIM kartě a z náhodné výzvy o 128 bitech během autentizace vygenerován klíč K_c pro šifru A5.
Z tohoto klíče je pak vygenerováno 228 bitů proudu klíče.

Generátor proudového klíče tvoří 3 LSFR registry:

- X (19 bitů),
- Y (22 bitů),
- Z (23 bitů)



Operace v registru X

$$t = x_{13} \oplus x_{16} \oplus x_{17} \oplus x_{18}$$
$$x_i = x_{i-1} \quad \text{for } i = 18, 17, 16, \ldots, 1$$
$$x_0 = t$$

Operace v registru Y

$$t = y_{20} \oplus y_{21}$$
$$y_i = y_{i-1} \quad \text{for } i = 21, 20, 19, \ldots, 1$$
$$y_0 = t$$

Operace v registru Z

$$t = z_7 \oplus z_{20} \oplus z_{21} \oplus z_{22}$$
$$z_i = z_{i-1} \quad \text{for } i = 22, 21, 20, \ldots, 1$$
$$z_0 = t$$

Šifra A5/1 je implementována hardwarově - v každém hodinovém cyklu se určuje hodnota m jako

$$m = \text{maj}(x_8, y_{10}, z_{10})$$

kde funkce **maj(x,y,z)** vrací 0 pokud je většina bitů x,y,z nulová, jinak vrací 1. Registry X,Y, Z provádí posun, pokud jsou splněna následující pravidla :

$$\text{If } x_8 = m \text{ then } X \text{ steps}$$

$$\text{If } y_{10} = m \text{ then } Y \text{ steps}$$

$$\text{If } z_{10} = m \text{ then } Z \text{ steps}$$

Výsledný bit proudového klíče s je pak generován jako:

$$s = x_{18} \oplus y_{21} \oplus z_{22}$$

Mezi bitem proudového klíče a bitem otevřeného textu (při šifrování) popř. bitem šifrovaného textu se provádí operace XOR.

Šifra RC4

Používá se při zabezpečení bezdrátových WiFi sítí pracujících v bezlicenčních pásmech nazývané WEP (Wired Equivalent Privacy), které je součástí původního standardu IEEE 802.11 z roku 1999. Tato aplikace algoritmu RC4 však není ideálním příkladem použití.
WEP používá nevhodně aplikovanou proudovou šifru RC4 (Rivest Cipher verze 4). Podrobný popis algoritmu RC4 (vyvinutý Ronem Rivestem v roce 1987) byl známý pouze osobám, které podepsali důvěrný dodatek, neboť byl ve vlastnictví RSA Data Security, Inc. V září roku 1994 však anonymní odesílatel uveřejnil zdrojový kód algoritmu, a tak se rychle rozšířil po celém světě.
Algoritmus RC4 bylů v roce 2004 označen jako zastaralý a nedoporučovaný, přesto se stále používá.
Oproti A5, která je orientovaná na hw implementaci , RC4 je orientován na sw implementaci .
RC 4 generyje v každém kroku proudový klíč o délce 8 bitů (A5 generoval 1 bit).

Princip RC4:

základ algoritmu generování proudového klíče tvoří vyhledávací tabulka, která obsahuje permutace 256-bytových hodnot. Pokaždé, když je generován byte proudového klíče, je vyhledávací tabulka modifikována tak, aby obsahovala permutace množiny {0, 1, 2, …, 255}.

1.Inicializace tabulky klíčem key:

```
for i = 0 to 255
    S[i] = i
    K[i] = key[i  mod  N]
next i
j = 0
for i = 0 to 255
    j = (j + S[i] + K[i])  mod 256
    swap(S[i],S[j])
next i
i = j = 0
```

Klíč key může mít délku od 0 do 256 bytů

2. Generování bytu proudového klíče

$$i = (i + 1) \mod 256$$
$$j = (j + S[i]) \mod 256$$
$$\texttt{swap}(S[i], S[j])$$
$$t = (S[i] + S[j]) \mod 256$$
$$\texttt{keystreamByte} = S[t]$$

3. XOR mezi bytem proudového klíče a otevřeným textem (při šifrování), popř. šifrovaným textem (při dešifrování).

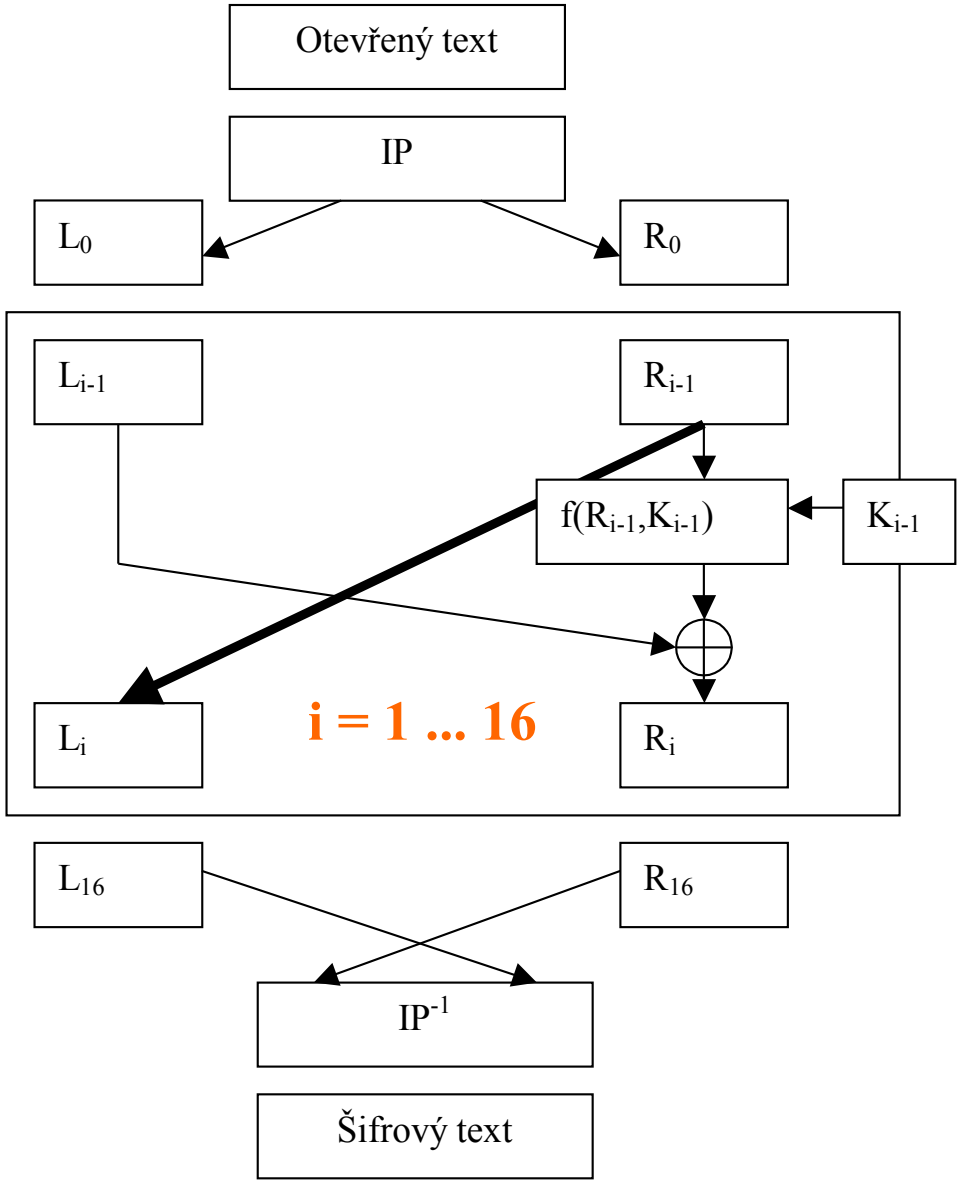
DES – Data encryption standard

- Jde o nejpoužívanější šifru na světě.
- Je výsledkem veřejné soutěže v roce 1977.
- Délka klíče je 56 bitů, což už v době vzniku bylo považováno za nepříliš bezpečné.
- Tuto délku klíče do původního návrhu IBM vnesla National Security Agency.
- Jde o iterovanou šifru, kdy je původní blok otevřené zprávy postupně šifrován pomocí šifrovacích zobrazení $E_{k(1)}$, $E_{k(2)}$,..., $E_{k(16)}$.
- Délka bloku je 64 bitů.
- Jednotlivá šifrování se nazývají *runda*.
- Původní klíč délky 56 bitů je *expandován* na 16 rundovních klíčů *k(1), k(2),…,k(16),* každý délky 48 bitů.

Základní schéma DES

IP je nějaká permutace na 64 bitech.

Blok o 64 bitech se rozdělí na levou a pravou polovinu délky 32 bitů.



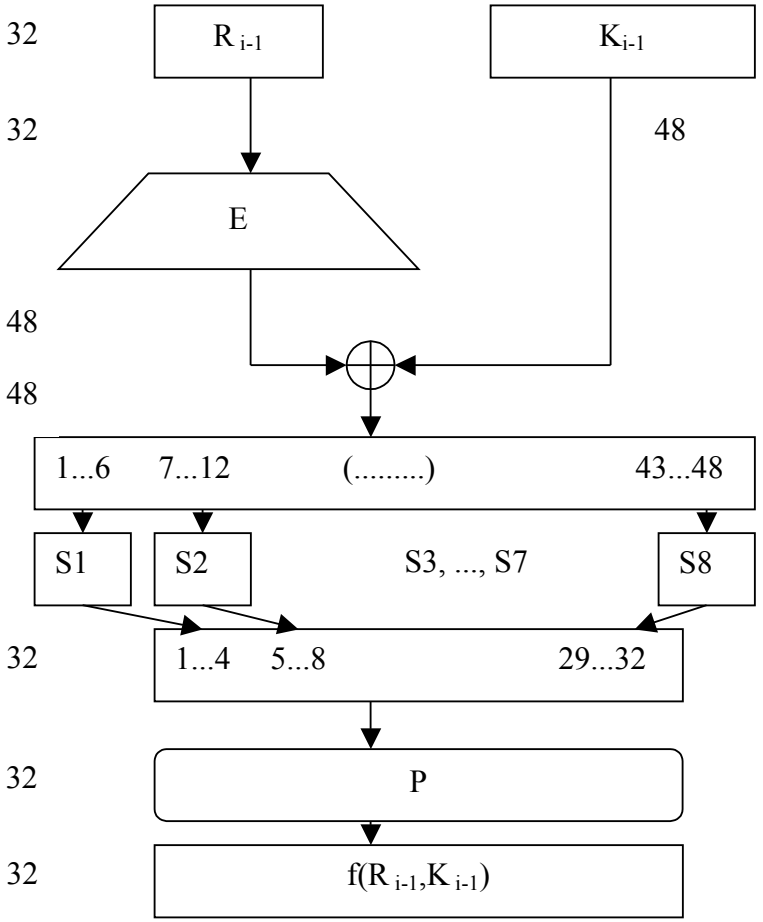
Rundovní funkce

Toto je schématické znázornění rundovní funkce.

E je expanzní funkce, která z posloupnosti 32 bitů udělá 48 bitů.

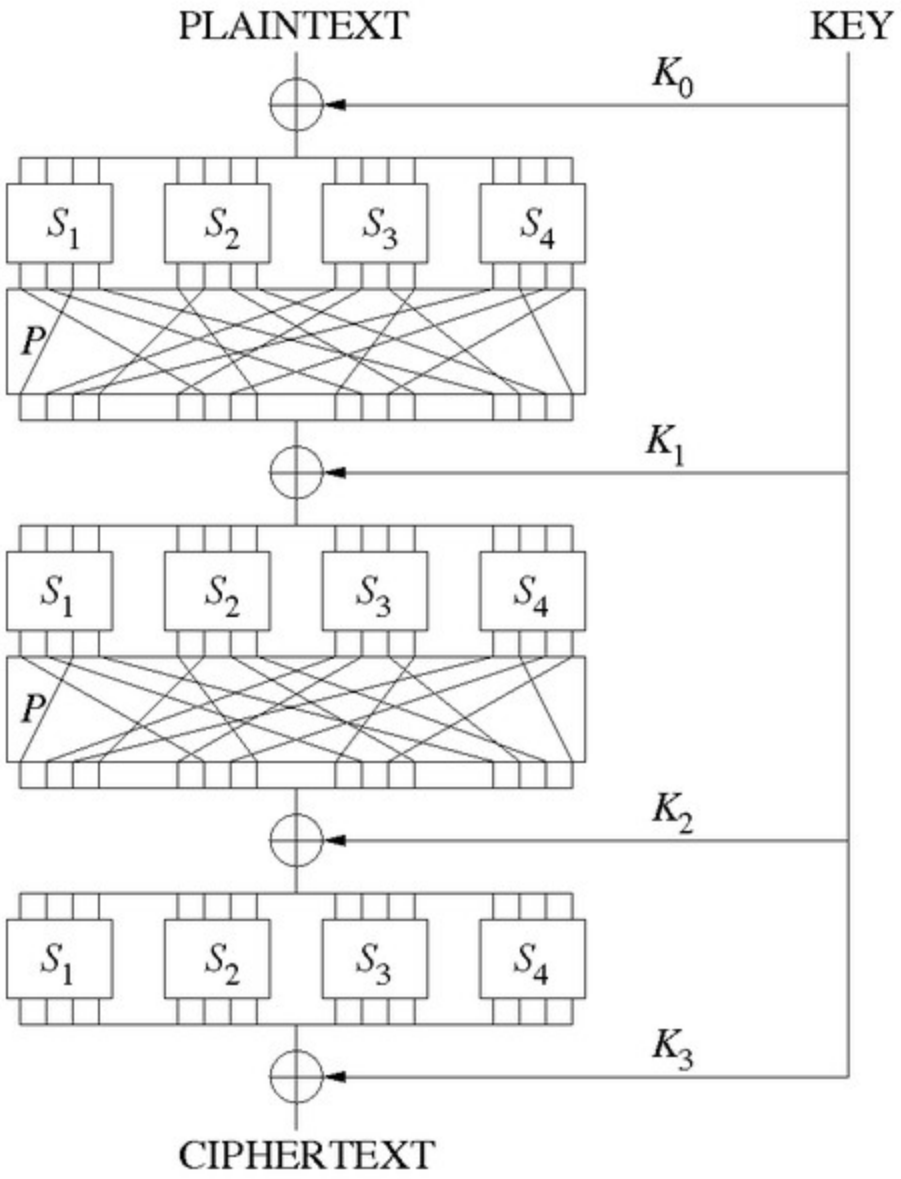
S – boxy nelineárně transformuji šestice bitů ve čtveřice bitů.

P je permutace na 32 bitech.



AES – Advanced encryption standard

- V roce 1997 byla vyhlášena celosvětová soutěž na návrh blokové šifry nové generace.
- Přihlásilo se 15 účastníků.
- Jako vítěz byla šifra navržená belgickými kryptology V. Rijmenem a J. Daemenem.
- Je založena na šifrovacím algoritmu Rijndael.
- Délka bloku je 128 bitů.
- AES podporuje tři délky klíčů – 128, 192 a 256 bitů.
- Počet rund se může měnit od 10 do 14 v závislosti na velikosti klíče.
- Algoritmus šifrování pracuje na principu tzv. substitučně permutační sítě SPN


PLAINTEXT
KEY
K_0
S_1
S_2
S_3
S_4
P
K_1
S_1
S_2
S_3
S_4
P
K_2
S_1
S_2
S_3
S_4
K_3
CIPHERTEXT

Popis algoritmu:

1.KeyExpansion – z šifrovacího klíče je odvozen rundovní klíč K_i

2.Inicializace - mezi zpracovávaným blokem a klíčem K_i je provedena operace XOR

3.Provedení rundy - každá runda se skládá ze čtyř operací

1. SubBytes - nelineární substituce, při které je každý byte nahrazen jiným z vyhledávací tabulky
2. ShiftRows - transpoziční krok – každý řádek stavu je cyklicky posunut
3. MixColumns – operace, která vezme bajty sloupce a lineární transformací je změní
4. AddRoundKey – mezi každým bytem stavu a subklíčem K_i se provádí bitový XOR

Operace 1 rundy pro klíč délky 128 bitů

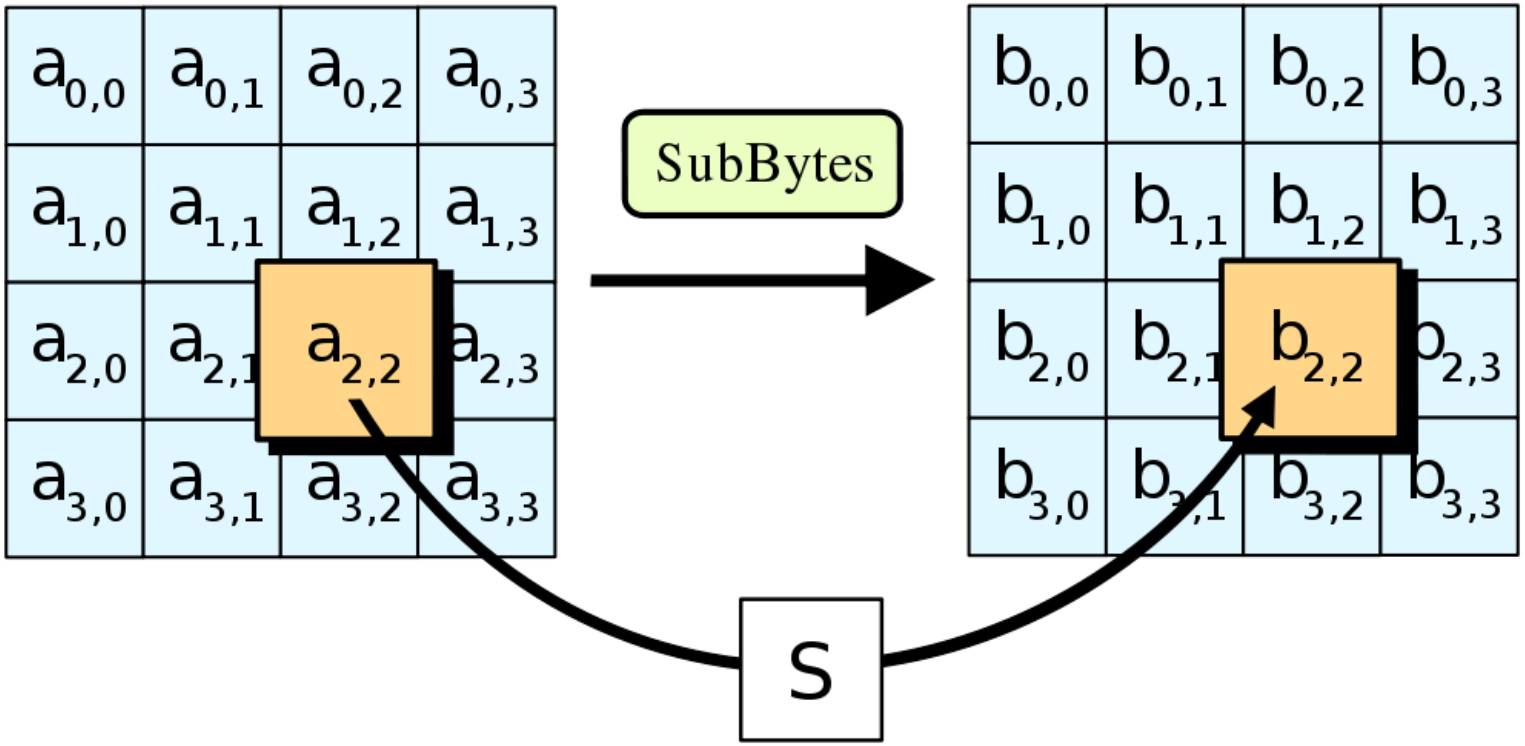
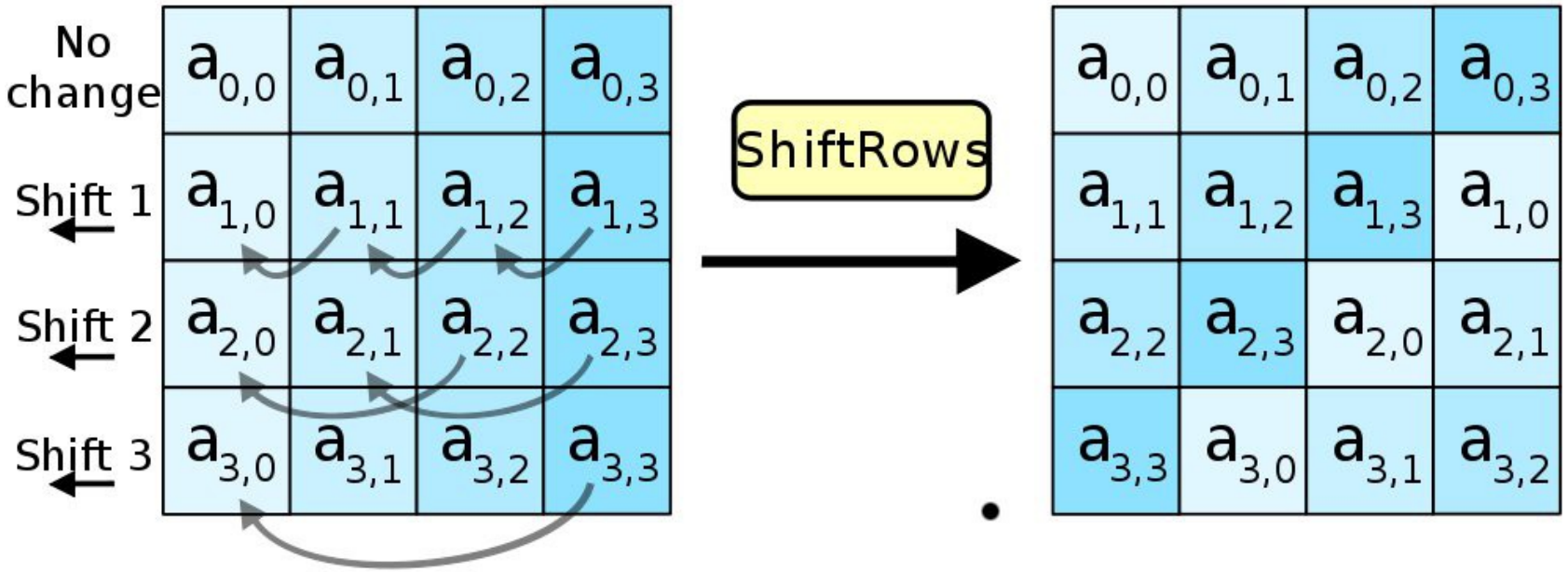
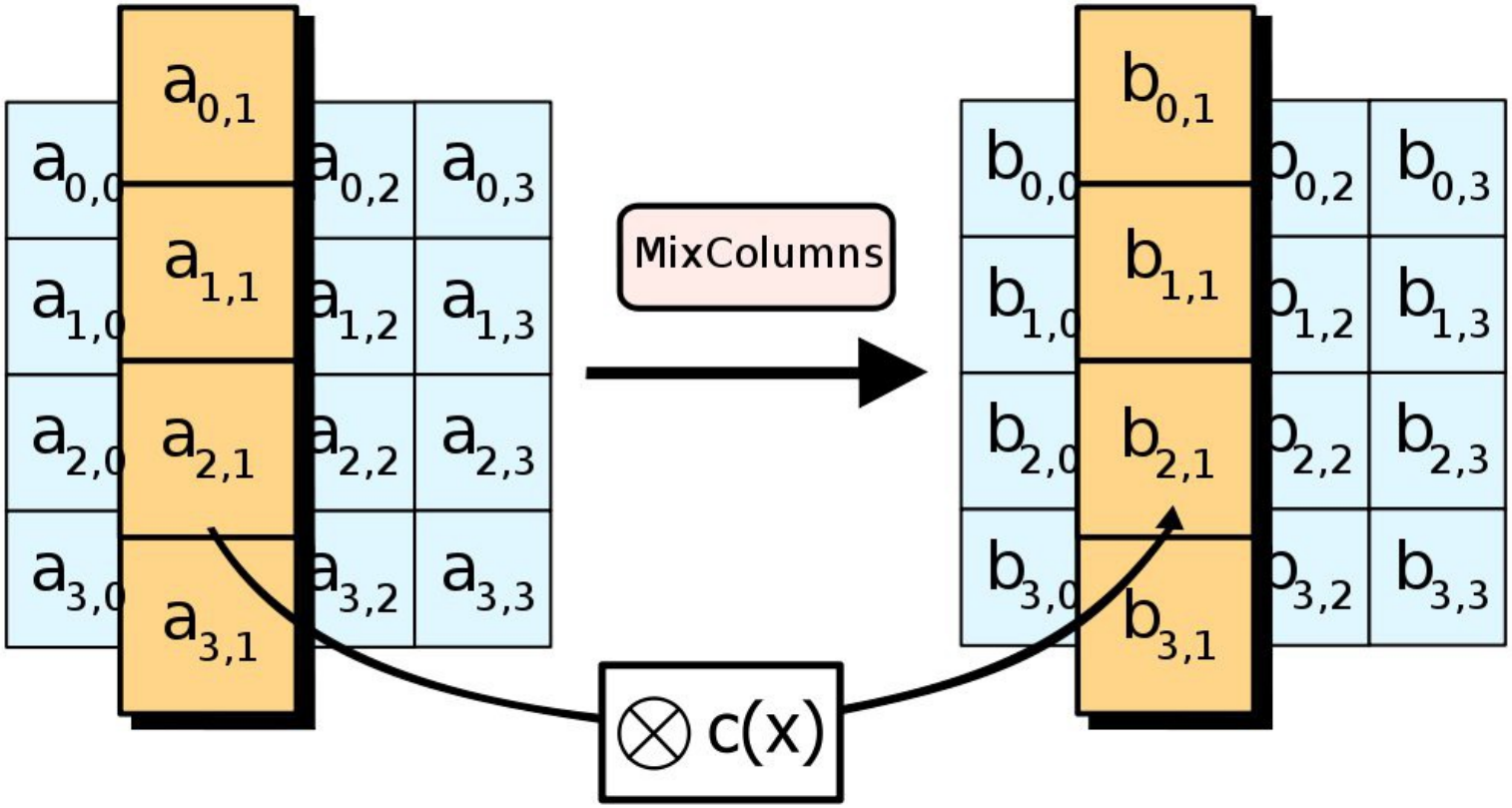
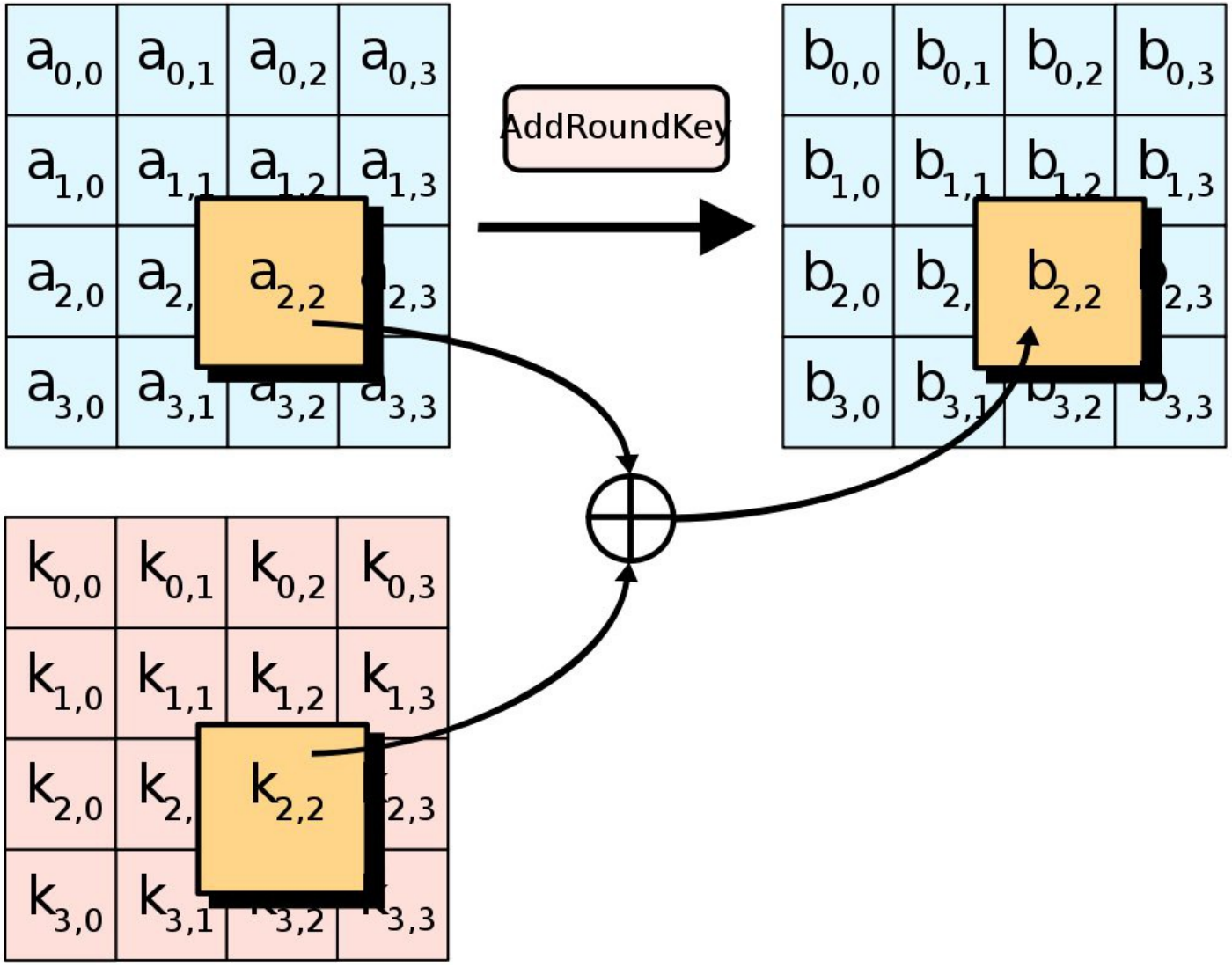


TABLE 3.4. AES ByteSub.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
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


No change
Shift 1
Shift 2
Shift 3
a0,0 a0,1 a0,2 a0,3
a1,0 a1,1 a1,2 a1,3
a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3
ShiftRows
a0,0 a0,1 a0,2 a0,3
a1,1 a1,2 a1,3 a1,0
a2,2 a2,3 a2,0 a2,1
a3,3 a3,0 a3,1 a3,2


a0,1
a0,0
a0,2
a0,3
a1,1
a1,0
a1,2
a1,3
a2,1
a2,0
a2,2
a2,3
a3,1
a3,0
a3,2
a3,3
MixColumns
b0,1
b0,0
b0,2
b0,3
b1,1
b1,0
b1,2
b1,3
b2,1
b2,0
b2,2
b2,3
b3,1
b3,0
b3,2
b3,3
c(x)


AddRoundKey

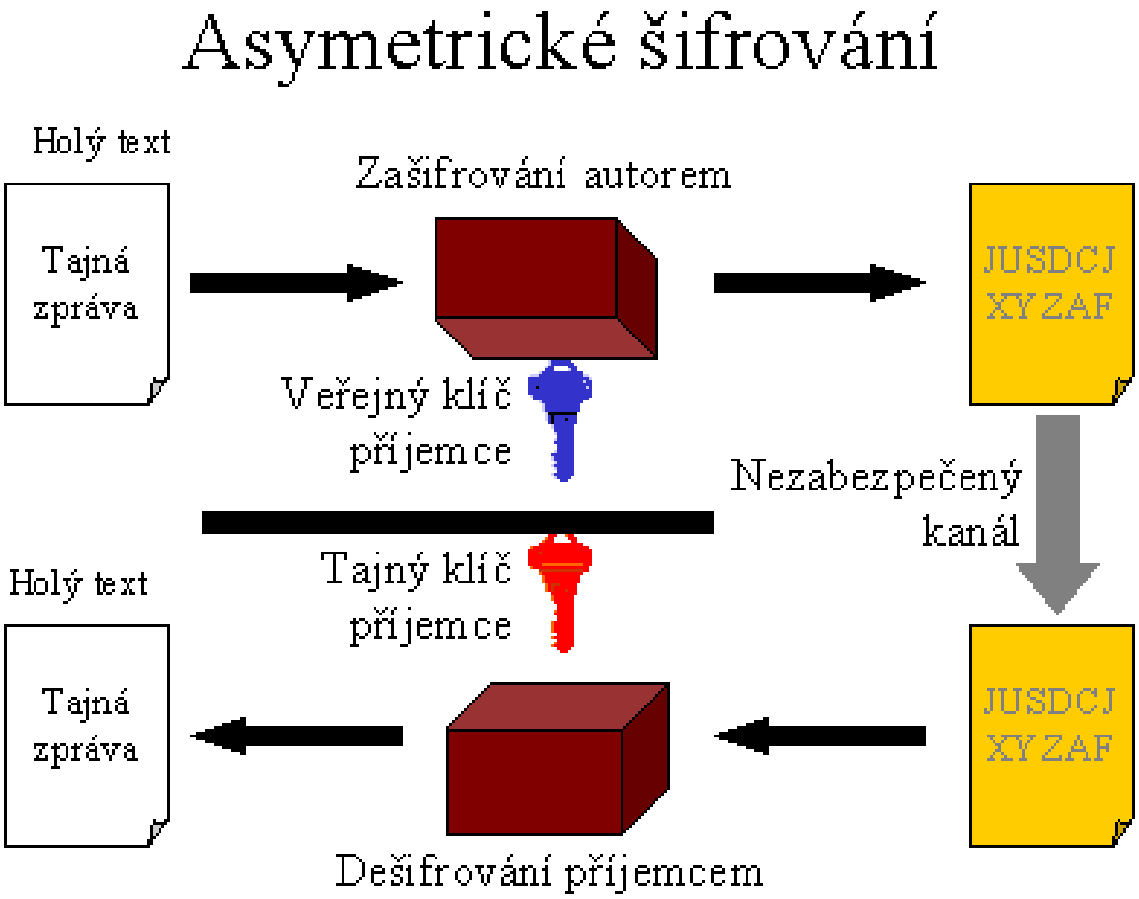
Další symetrické šifry:

- **Triple-DES** – šifrovací algoritmus DES se používá 3x se dvěma různými klíči
- **RC2, RC4** – Rivestovy kódy - klíč o délce 1 – 1024 bitů, RC2 je bloková šifra podobná DES, RC4 - proudová šifra
- **IDEA** (International Data Encryption Algorithm) – 128 bitový klíč

Asymetrické šifrování

- Používá jiný klíč k zašifrování a jiný klíč zpátky k dešifrování
- První z nich se nazývá veřejný, ostatní ho musejí znát. Druhý klíč se nazývá privátní
- Asymetrický šifrovací systém (systém s veřejným klíčem) je založen na principu jednocestné funkce, což jsou operace, které lze snadno provést pouze v jednom směru: ze vstupu lze snadno spočítat výstup, z výstupu však je velmi obtížné nalézt vstup.
- Nejběžnějším příkladem je například násobení: je velmi snadné vynásobit dvě i velmi velká čísla, avšak rozklad součinu na činitele (tzv. faktorizace) je velmi obtížný. (Na tomto problému je založen např. algoritmus RSA.)

Asymetrické šifrování



Asymetrické šifrování – algoritmy

- RSA (Rivest, Shamir, Aldeman) – algoritmus vhodný jak pro podepisování, tak pro šifrování

Princip:

Bezpečnost RSA je postavena na předpokladu, že rozložit velké číslo na součin prvočísel (faktorizace) je velmi obtížná úloha. Z čísla $n = pq$ je tedy v rozumném čase prakticky nemožné zjistit činitele p a q, neboť není znám žádný algoritmus faktorizace, který by pracoval v polynomiálním čase vůči velikosti binárního zápisu čísla n. Naproti tomu násobení dvou velkých čísel je elementární úloha.

Popis algoritmu:

Alice a Bob chtějí komunikovat prostřednictvím otevřeného (nezabezpečeného) kanálu a Bob by chtěl Alici poslat soukromou zprávu.

Tvorba klíčového páru:

Nejprve si bude Alice muset vyrobit pár veřejného a soukromého klíče:

1. Zvolí dvě různá velká náhodná prvočísla p a q.
2. Spočítá jejich součin $n = pq$.
3. Spočítá hodnotu Eulerovy funkce $\varphi(n) = (p - 1)(q - 1)$.
4. Zvolí celé číslo ***e*** menší než $\varphi(n)$, které je s $\varphi(n)$ nesoudělné.
5. Nalezne číslo ***d*** tak, aby platilo $de \equiv 1 \pmod{\varphi(n)}$.
6. Pokud ***e*** je prvočíslo tak $d = (1+r*\varphi(n))/e$, kde $r = [(e-1)\varphi(n)^{(e-2)}]$

Veřejným klíčem je dvojice (n, e), přičemž n se označuje jako *modul*, e jako *šifrovací* či *veřejný exponent*. **Soukromým klíčem** je dvojice (n, d), kde d se označuje jako ***dešifrovací* či *soukromý exponent***. (V praxi se klíče uchovávají v mírně upravené formě, která umožňuje rychlejší zpracování.)

- Veřejný klíč poté Alice uveřejní, respektive zcela pošle nešifrovaně Bobovi. Soukromý klíč naopak uchová v tajnosti.

Šifrování zprávy:

Bob nyní chce Alici zaslat zprávu *M*.

1. Tuto zprávu převede nějakým dohodnutým postupem na číslo *m* (*m* < *n*).
2. Šifrovým textem odpovídajícím této zprávě pak je číslo

   $c = m^e \bmod n$.

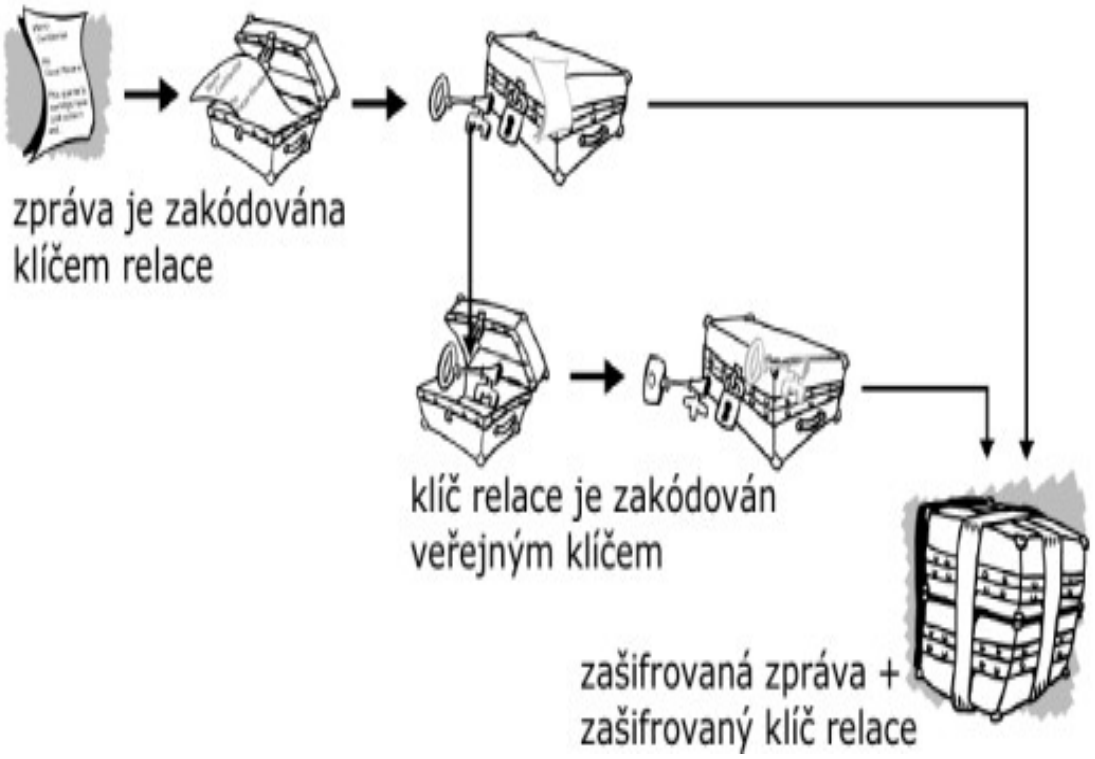
3. Tento šifrový text poté zašle nezabezpečeným kanálem Alici.

Dešifrování zprávy:

- Alice od Boba získá šifrový text *c*. Původní zprávu *m* získá následujícím výpočtem: $m = c^d \bmod n$.

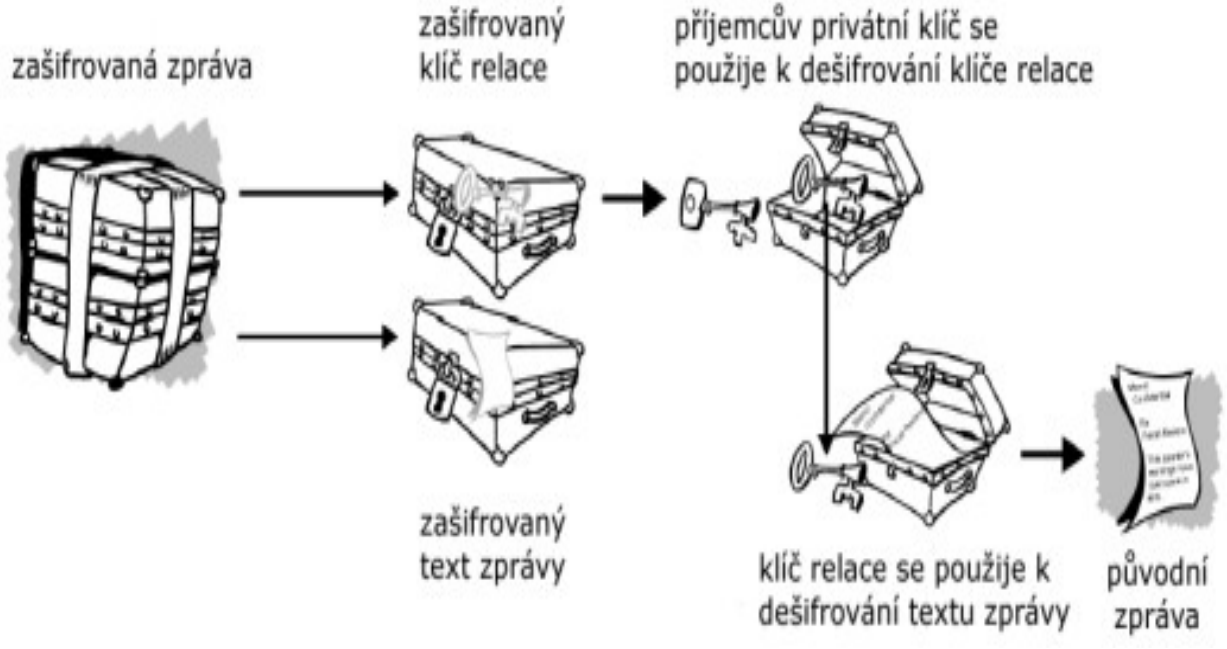
Hybridní šifrování I.

- Odesilatel zvolí klíč, kterým symetricky zašifruje zprávu. Tento klíč zašifruje veřejným klíčem adresáta a pošle ho spolu se zprávou adresátovi
- Adresát tedy dostane asymetricky zašifrovaný klíč a symetricky zašifrovanou zprávu.
- Klíč dešifruje svým privátním klíčem a použije ho k dešifrování textu
- Tím zaniká problém distribuce klíče při symetrickém šifrování a zároveň se celý proces zrychlí.



Hybridní šifrování II.

- Pokud chtějí dva počítače komunikovat přes otevřenou síť, kde je každý může „odposlechnout“, vytvoří *relaci*
- Na začátku vygeneruje jeden z nich klíč, zašifruje ho veřejným klíčem 2. počítače a pošle
- Druhý počítač si klíč dešifruje, oba dva mají stejný klíč, který kromě nich nikdo jiný nezná. Mohou tedy používat symetrické šifrování
- Každá další komunikace je symetricky zašifrovaná



Kryptografické hashovací funkce

- Matematická funkce
- Vstup: posloupnost libovolné konečné délky (text, hudba, obrázky, video,…)
- Výstup: posloupnost konstantní délky (typicky stovky bitů)
  nazývá se - haš, haše, hash, hashový kód, otisk zprávy

$$h=H(M)$$

Další obvyklé požadavky na HF zahrnují:

- **Nekorelovatelnost vstupních a výstupních bitů**
  - znemožnní statistickou kryptoanalýzu.

- **Odolnost vůči skoro-kolizím (*Near-collision resistance*)**
  - je obtížné nalézt *x* a *y* taková, že x≠y a zároveň *H*(*x*) a *H*(*y*) se liší jen v malém počtu bitů.

- **Lokální odolnost vůči získání předlohy.**
  - je obtížné najít i jen část vstupu *x* ze znalosti *H*(*x*)

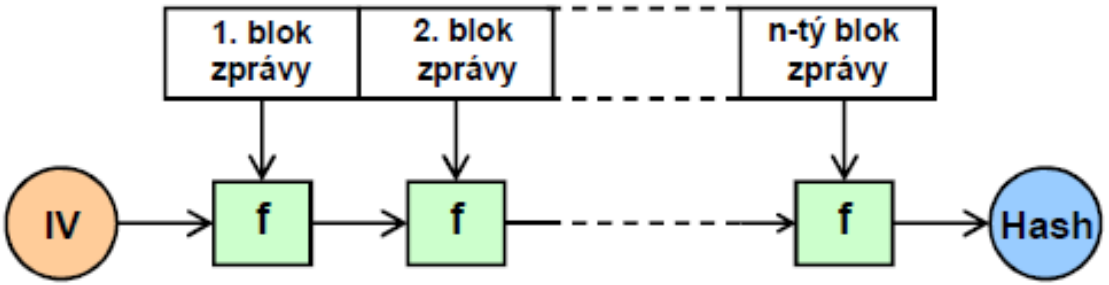
Použití hashovacích funkcí

- **Digitální / elektronický podpis** – zajištění integrity
- **PKI** - integrita X.509 certifikátů a seznamů CRL
- **Časové značky** – integrita časové značky
- **Kerberos** - generování klíčů
- **IEEE 802.1X EAP** : EAP-FAST, EAP-TLS, EAP-TTLS… používají TLS protokol, který používá hashovací funkce
- **APOP** - autentizace pomocí MD5
- **RADIUS** - integrita dat
- **IPsec (IKE, AH, ESP)** – integrita zpráv, generování pseudonáhodných posloupností.
- **SSL/TLS** - handshake protokol používá hashovací funkce kvůli tvorbě HMAC a při generování klíčů a IV
- **SSH** – HMAC a integrita přenášených dat
- **S/MIME** - použití hashovacích funkcí v digitálním podpise

Obecně se HF používají ke

- kontrola zachování integrity dat
- hashe pro digitální podpis
- kontrola uložených hesel
- porovnání obsahu dvou kopií dat
- generování pseudonáhodných posloupností (PRNG)

Merkle-Damgårdova struktura hashovací funkce



f – kompresní funkce

IV – inicializační vektor

Hash – výsledný otisk zprávy

- bloky *f* provádějí kompresní funkci
- toto schéma využívá většina moderních hashovacích funkcí – MD-5, SHA-1, RIPEMD-160
- vstup musí být doplněn na celistvý násobek délky bloků
- musí být jednodznačně určitelné kolik se doplnilo (jinak by jednoduše vznikala řada kolizí)
- Merkle-Damgårdovo zesílení (doplnění výplně posledního bloku o délku zprávy)

Merkleova meta-metoda pro tvorbu HF

Vstup: funkce *f* odolná vůči kolizím.

Výstup: iterovaná CRHF h odolná vůči kolizím

Vstup délky *x* se rozdělí na *n* bloků x_1 ,... ,x_n o délce *m* bitů. Poslední blok x_n se zleva doplní nulami na délku *m* bitů. Volitelně lze provést Merklovo-Dammgårdovao posílení.

Výpočet s-bitového výstupu zprávy *x:*

$h(x) = H_{t+1} = f(H_t \,||\, x_{t+1})$;

$H_0 = 0$

$H_i = f(H_{i-1} \,||\, x_i)$.

Používané hashovací funkce

MD2 – kompromitovaná, nepoužívá se
MD4 – kompromitovaná, nepoužívá se
MD5 –oblíbená, ale kompromitovaná funkce. Od srpna 2004 je veřejně znám postup nalezení kolizí a to i pro málo odlišná vstupní data.
RIPEMD-128 - kompromitovaná, nepoužívá se
RIPEMD-160 – může být kompromitována
WHIRLPOOL – považuje se za bezpečnou
TIGER – nebyla nalezena kolize
GRINDAHL – mladá (03/2007), jádro z AES

Používané hashovací funkce

SHA-0 – kompromitována, nepoužívá se

SHA-1 – oblíbená, ale již kompromitovaná funkce. V únoru 2005 byl zveřejněn objev algoritmu, který umožňuje nalézt kolizi podstatně rychleji než hrubou silou. Prakticky zatím neprovedeno.

SHA-2 – dosud považována za spolehlivou

- není to jedna hashovací funkce, ale více variant souhrnně označovaných jako SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)

MD-4

- Většina dnes používaných kryptografických hashovacích funkcí vychází z algoritmu MD-4 (Message Digest)
- MD4 byla navržena s ohledem na efektivní zpracování na existujících 32bitových procesorech.
- teoretická odolnost algoritmu MD-4 proti kolizím je 2^{64} pro 128-bitový výstup
- v praxi byly nalezeny kolize v prostoru hash kódů 2^{20} ke kompresní funkci
- není považována za bezpečnou
- 3 kola po 16 krocích ..celkem 48 rund

MD-5

Vstup: řetězec proměnné délky (neomezeně dlouhé)
Výstup: pevná délka128 bitů
Počet rund: 4
Počet kroků v rundě: 16

Vstup je zpravován po úsecích délky 512 bitů
Zpráva je zarovnána tak, aby byla dělitelná 512.

Zarovnání:
- zpráva se doplní zprava o jeden bit s hodnotou „1“
- zpráva se doplní zprava bity s hodnotou „0“ až do délky o 64 bitů menší než je číslo dělitelné 512
- posledních 64 bitů obsahuje číslo reprezentující původní délku zprávy mod 2^{64}

Změny vůči MD-4

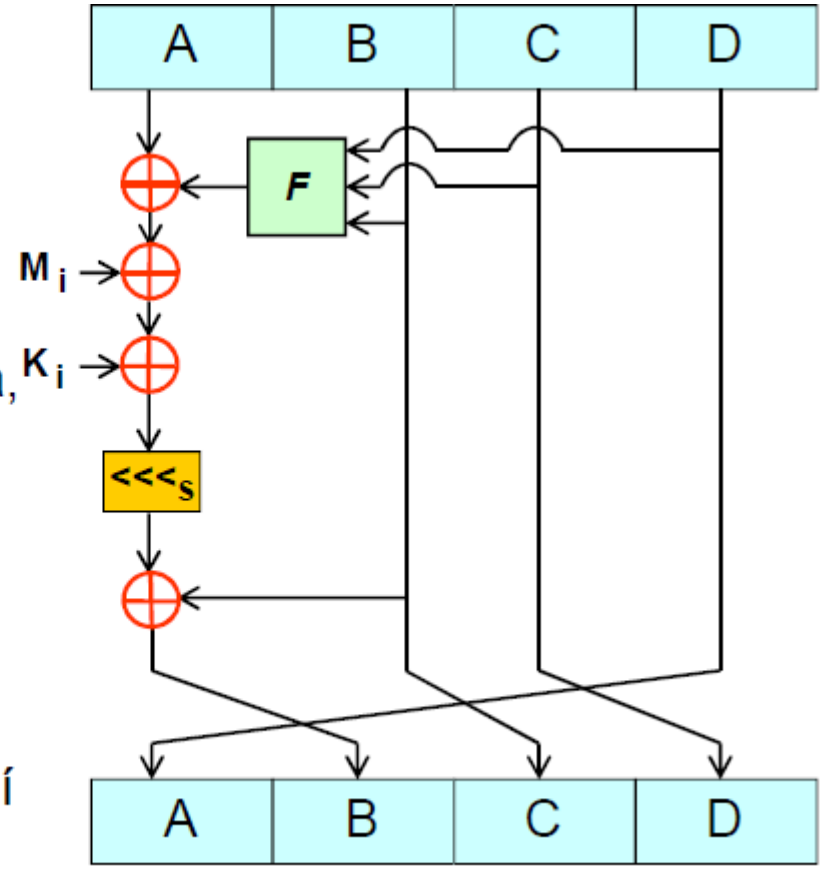
- konzervativní varianta MD4 (pomalejší)
- přidáno 4. kolo o 16 krocích
- celkem 64 rund
- změna logické funkce v druhé rundě
- jiné bitové posuny v jednotlivých krocích
- jiné aditivní konstanty (jedinečné v každé rundě)
- v každém kroku se připočítá výsledek z minulé rundy

(to urychluje tzv. lavinový efekt)

MD-5

M_i – vstupní zpráva 32bitů
K_i – konstanta 32bitů
– mění se pro každou operaci
F – nelineární funkce
$<<<_s$ – rotace o *s* bitů doleva, hodnota *s* se mění v každé operaci
⊕ – operace sčítání mod 2^{32}

A,B,C,D – pomocné registry
– na začátku konstanty
– po poslední rundě obsahují hash
- délka 128 bitů (4x32)



Jeden krok algoritmu MD-5

- vstupní blok M_i má délku 512 bitů
- před vstupem do vlastního algoritmu je rozdělen na 32 bitové části (16 různých bloků)
- každý blok vstupuje do kompresní funkce čtyřikrát, pokaždé s jinou nelineární funkcí *f*
- celkem tedy 4 kola *16 kroků = 64 rund
- konstanta K_i je spočítána jako celá část z 4294967296 násobku absolutní hodnoty funkce sin(i), kde i je úhel v radiánech

Operace v bloku „F“

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

$$H(X,Y,Z) = X \oplus Y \oplus Z$$

$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus$ *XOR*

$\vee$ *OR*

$\wedge$ *AND*

$\neg$ *NOT*

- v každém kole probíhá jiná nelineární operace v bloku F

Počáteční hodnoty IV

- registr A: 0x01234567
- registr B: 0x89abcdef
- registr C: 0xfedcba98
- registr D: 0x76543210

SHA-0 SHA-1

- SHA-0 představena NISTem v roce 1993 jako SHS (Secure Hash Standard)
- standard NIST PUB-180
- těsně před schválením v roce 1995 stažena (na pokyn NSA)
- mírně modifikována a schválena jako SHA-1 (PUB 180-1)
- byla přidána dodatečná rotace vlevo do každého z prováděných kol kompresní funkce
- v srpnu 1998 byl odhalen pravděpodobný důvod této změny – (Differential Collisions in SHA-0)

  www.springerlink.com/index/P795V6NJ1VJ525KP.pdf
- změna v SHA-1 ničí zarovnání bitů vstupu x po průchodu kompresní funkcí

- SHA – Secure Hash Algorithm
- odvozeno od MD4
- počet výstupních bitů rozšířen na 160
- kompresní funkce má o 1 kolo navíc
- každé kolo má 20 kroků místo původních 16
- celkem 80 rund (4x20)
- jiné hodnoty IV
- pět počátečních nenulových aditivních konstant
- konvence je big-endian (na rozdíl od algoritmů MD)

SHA-1

Vstup: řetězec proměnné délky (max. 2^{64} -1 bitů)
Výstup: pevná délka160 bitů
Počet rund: 4
Počet kroků v rundě: 20

Vstup je zpravován po úsecích délky 512 bitů
Zpráva je zarovnána tak, aby byla dělitelná 512.

Zarovnání: - zpráva se doplní zprava o jeden bit s hodnotou „1“
- zpráva se doplní zprava bity s hodnotou „0“ až do délky o 64 bitů menší než je číslo dělitelné 512
- posledních 64 bitů obsahuje číslo reprezentující původní délku zprávy

SHA-1 expanze bloku

- vstupní blok M_i má délu 512 bitů
- před vstupem do vlastního algoritmu je rozdělen a expandován:

  1) rozdělení na 32 bitové části (16 různých bloků) označených $W_0 \ldots W_{15}$

  2) expanzní funkce $E\{0,1\}^{512} \rightarrow E\{0,1\}^{2560}$ těchto 16 částí rozšíří na 80 podle schématu

  $$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1$$

  $t = 16..79$

$<<<_5$ – rotace o 5

$<<<_{30}$ – rotace o 30

F – nelineární funkce mění se v každé rundě

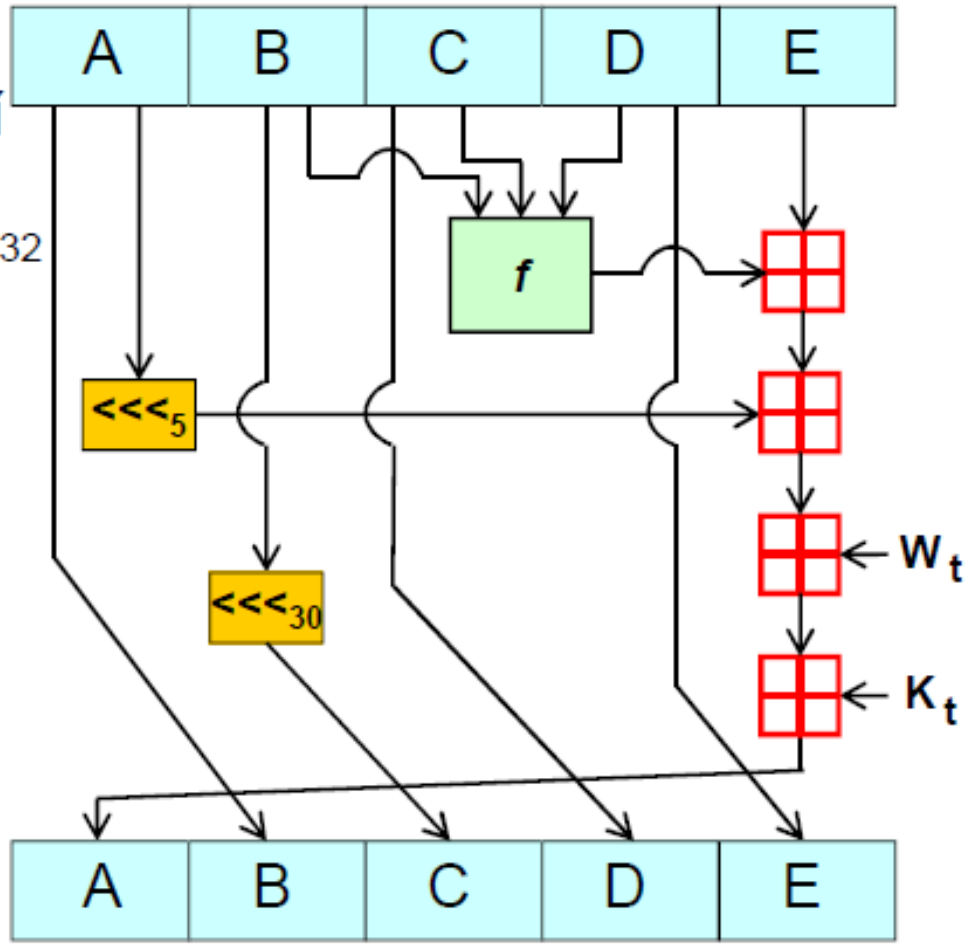
$\oplus$ – operace sčítání mod 2^{32}

W_t – vstupní zpráva, 32b

K_t – konstanta, 32b

A,B,C,D,E – vnitřní stavy - 32bitů (každý)

- celkem 160 bitů
- na konci procesu tam je hash
- na počátku konstanty



Funkce *F* v jednotlivých rundách algoritmu SHA-1

$$f(t,B,C,D) = (B \wedge C) \vee (\neg B \wedge D) \qquad (0 \le t \le 19)$$
$$f(t,B,C,D) = B \oplus C \oplus D \qquad (20 \le t \le 39)$$
$$f(t,B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) \qquad (40 \le t \le 59)$$
$$f(t,B,C,D) = B \oplus C \oplus D \qquad (60 \le t \le 79)$$

Konstanta K_t

K_t = 0x5A827999 (0 ≤ t ≤ 19)
K_t = 0x6ED9EBA1 (20 ≤ t ≤ 39)
K_t = 0x8F1BBCDC (40 ≤ t ≤ 59)
K_t = 0xCA62C1D6 (60 ≤ t ≤ 79)

Počáteční hodnoty registrů A,B,C,D,E

A = 0x67452301
B = 0xEFCDAB89
C = 0x98BADCFE
D = 0x10325476
E = 0xC3D2E1F0

- dosud se pokládala za bezpečnou
- není garantována bezpečnost po roce 2010 (tzn. vhodné pouze pro krátkodobou bezpečnost)
- NIST doporučuje ukončit používání SHA-1 , nejpozději do konce r. 2010
- Než budou představeny zcela nové HF, používat SHA-2
- 2005 - navržen hardwarový SHA-1 Cracker (podobně jako u DESu)
  - 303 PC
  - v každém PC 16 desek
  - na každé desce 32 jader
  - cena: 1.000.000 $
  - doba prolomení SHA-1 2 dny

http://csrc.nist.gov/CryptoToolkit/tkhash.htm

Elektronický podpis

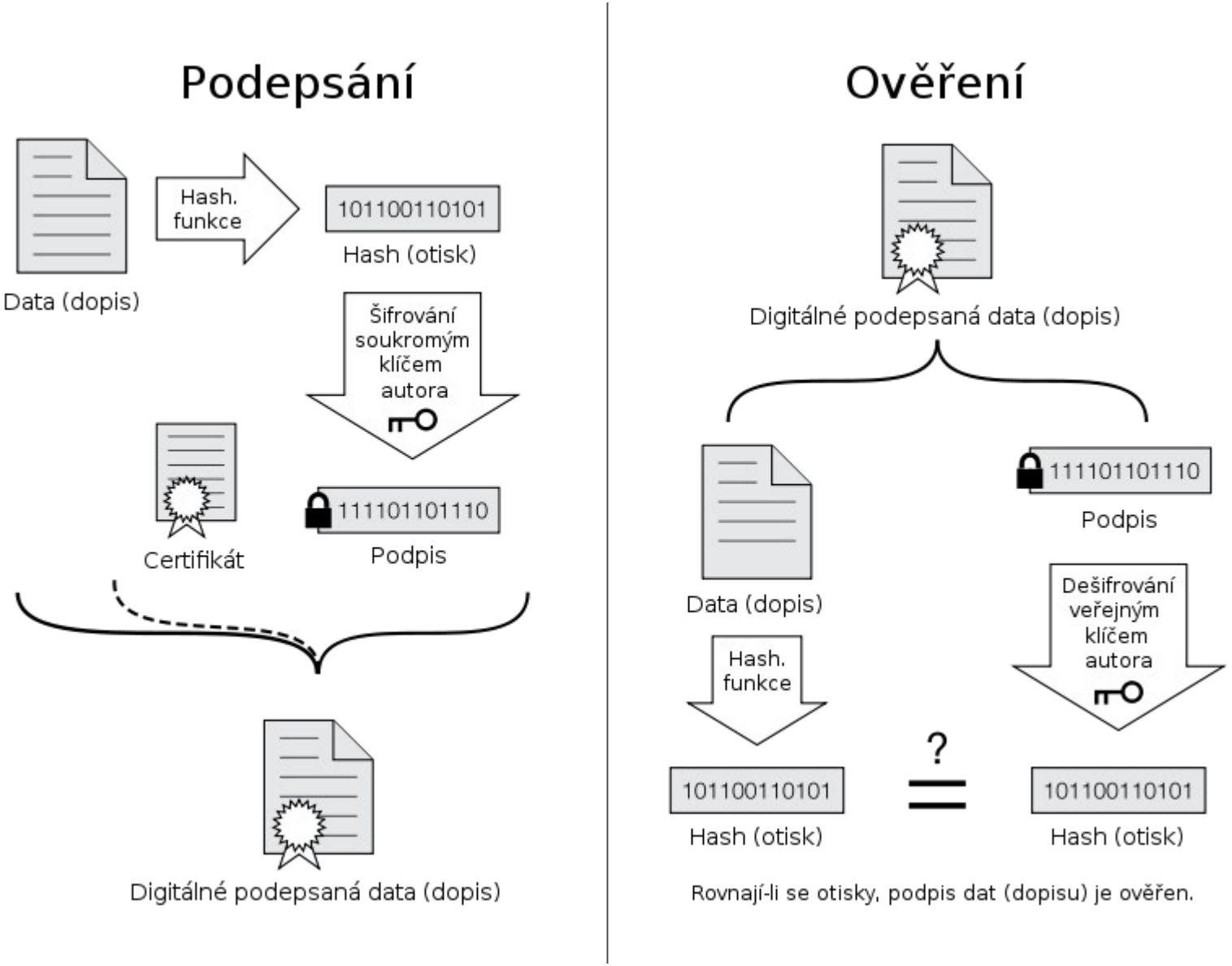
Elektronický podpis jsou elektronické identifikační údaje autora (odesílatele) elektronického dokumentu, připojené k tomuto dokumentu.

Zaručený elektronický podpis je elektronický podpis v takové formě, která zaručuje (zpravidla použitím kryptografických metod):

- autenticitu– lze ověřit původnost (identitu) subjektu, kterému patří elektronický podpis),
- integritu – lze prokázat, že po podepsání nedošlo k žádné změně, soubor není úmyslně či neúmyslně poškozen,
- nepopiratelnost – autor nemůže tvrdit, že podepsaný elektronický dokument nevytvořil (např. nemůže se zříct vytvoření a odeslaní výhružného dopisu),
- může obsahovat časové razítko, které prokazuje datum a čas podepsání dokumentu.

Elektronický popis – algoritmus

- Vybere se kryptografická hašovací funkce.
- Dále se rozhodne o parametrech L a N, které určují délku klíče. V původní verzi DSS (Digital Signature Standard) byla volba L omezena na násobky 64 v rozsahu 512 až 1024 včetně. Doporučují se dvojice L a N (1024,160), (2048,224), (2048,256) a (3072,256).
- Dále se vybere N-bitové prvočíslo q. Délka N musí být alespoň taková, jako délka výstupu použité hašovací funkce.
- Dále se vybere L-bitové prvočíslo p takové, že p-1 je násobek q.
- Nakonec se vybere g jako takové číslo, jehož multiplikativní řád modulo p je právě q. Toho lze dosáhnout dosazováním do vzorce $\boldsymbol{g=h^{(p-1)/q} \bmod p}$ pro náhodná h (kde $1< h < p-1$), dokud výsledek není různý od jedné. Většina náhodných voleb h uspěje, nejčastěji se používá $h=2$.
- Všechny výše zmíněné hodnoty mohou být sdílený více uživateli a nejsou tajné. Následuje vytvoření samotných klíčů.
- Nejdříve se náhodně vybere x v rozsahu $0<x<q$.
- pak se spočítá $y=g^x \bmod p$
- Veřejný klíč je pak dán jako čtveřice $\boldsymbol{(p,q,g,y)}$, soukromý klíč je dán jako x.


Podepsání
Hash. funkce
101100110101
Hash (otisk)
Data (dopis)
Šifrování soukromým klíčem autora
Certifikát
111101101110
Podpis
Digitálné podepsaná data (dopis)
Ověření
Digitálné podepsaná data (dopis)
111101101110
Podpis
Data (dopis)
Dešifrování veřejným klíčem autora
Hash. funkce
?
101100110101
Hash (otisk)
101100110101
Hash (otisk)
Rovnají-li se otisky, podpis dat (dopisu) je ověřen.

Podepisování

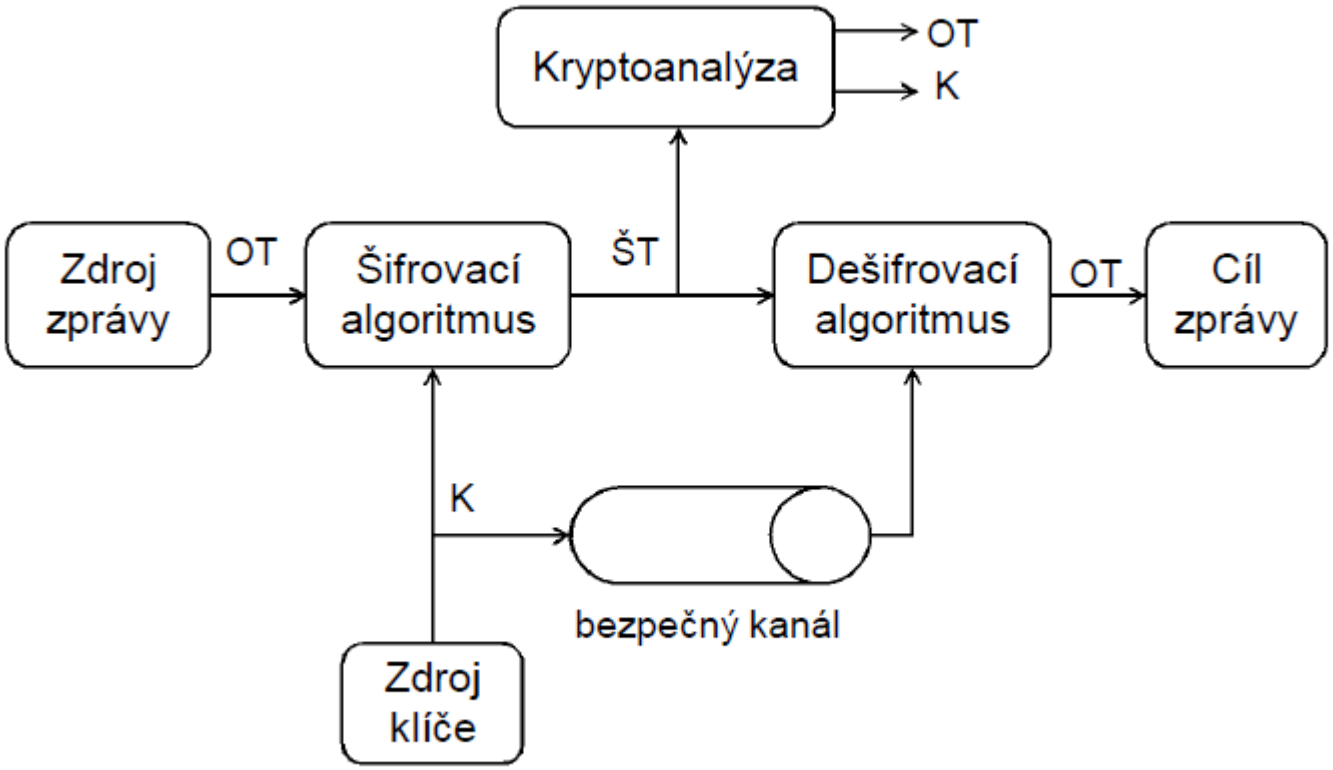
Při označení hašovací funkce písmenem H a zprávy písmenem z probíhá podepisování takto:

- pro danou zprávu se vybere náhodná hodnota k v rozsahu $0<k<q$
- spočítá se $r=(g^{k} \bmod p) \bmod q$
- spočítá se $s=(k^{-1}(H(z)+x\times r)) \bmod q$
- v nepříliš pravděpodobném případě, že je $r=0$ nebo $s=0$ se výpočet opakuje od začátku
- jinak je podpisem dvojice (r,s)

Ověřování podpisu

- pokud neplatí $0< r <q$ a $0< s <q$ pak je podpis automaticky zamítnut.
- jinak se spočítá $w = (s)^{-1} \bmod q$
- dále se spočítá $u1 = (H(z)*w) \bmod q$
- dále se spočítá $u2 = (r*w) \bmod q$
- nakonec se spočítá $v = ((g^{u1}*y^{u2}) \bmod p) \bmod q$
- Podpis platí, pokud platí $v = r$

Kvantová kryptografie


Kryptoanalýza
OT
K
Zdroj zprávy
OT
Šifrovací algoritmus
ŠT
Dešifrovací algoritmus
OT
Cíl zprávy
K
bezpečný kanál
Zdroj klíče

- Informace v tradičním systému (ne-kvantovém)
  - je možné ji neomezeně kopírovat
  - je možné vytvářet identické kopie zprávy
  - je možné ji měřit s libovolně malou chybou

- Informace v kvantovém systému
  - vychází z Heisenbergova principu neurčitosti
  - nelze ji libovolně kopírovat
  - těžk se uchovává, zpracovává
  - nelze vytvářet identické kopie
  - nelze kopírovat neznámý kvantový stav
  - pokus o zjištění přesné hodnoty způsobí změnu naměřené hodnoty

- prvotní myšlenka – Richard Feynman
- simulace kvantových systémů na klasickém počítači mají často exponenciálně rostoucí nároky na výpočetní čas v závislosti na délce vstupu
- nápad – nešlo by to využít obráceně -> urychlení některých algoritmů
- definice kvantového počítače - 1985 - David Deutsch
- Kvantový počítač využívá
  - principu superpozice
  - linearity kvantové mechaniky
  - jeho činnost je popsána unitárními operátory (z toho mj. plyne, že všechny s kvantovým počítačem jsou vratné)

Vernamova šifra

- one-time pad
- objevena v roce 1917 (Gilbert Vernam)
- jediný absolutně bezpečný kryptosystém
- nelze ho prolomit ani hrubou silou (brute-force attack)
- matematický důkaz provedl v roce 1949 C. E. Shannon
- problém s generováním a distribucí klíče

Použití:

- pro zabezpečení horké linky mezi Moskvou a Washingtonem
- projekt VENONA - http://en.wikipedia.org/wiki/Venona_project

Vernamova šifra

Požadavky nutné pro správnou funkci

- **Klíč je minimálně stejně dlouhý jako přenášená zpráva.**
  - jiné šifrovací systémy používají kratší klíče, což znamená, že počet možných klíčů je menší než počet možných zpráv
  - kratší klíč umožňuje útok hrubou silou

- **Klíč je dokonale náhodný.**
  - nelze použít klasické počítačové generátory pseudonáhodných posloupností
  - nejvhodnější je užití fyzikálních metod, například tepelného šumu nebo ještě lépe kvantových procesů (poločas rozpadu atd.)

Vernamova šifra

Šifrování: Znak otevřeného textu se přičítá na znak hesla pomocí operace XOR

Dešifrování: Znak šifrového textu se přičítá na znak hesla pomocí operace XOR

X	Y	X ⊗ Y	(X ⊗ Y) ⊗Y = X
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

Pravdivostní tabulka pro šifrování a dešifrování pomocí funkce XOR

Nepodmíněná bezpečnost

Systém, který nelze prolomit bez ohledu na dostupné množství výpočetního výkonu, protože ŠT neposkytuje dostatek informací nutných k jednoznačnému rozpoznání odpovídajícího OT

Systém s nepodmíněnou bezpečností bude funkční i ve věku kvantových počítačů.

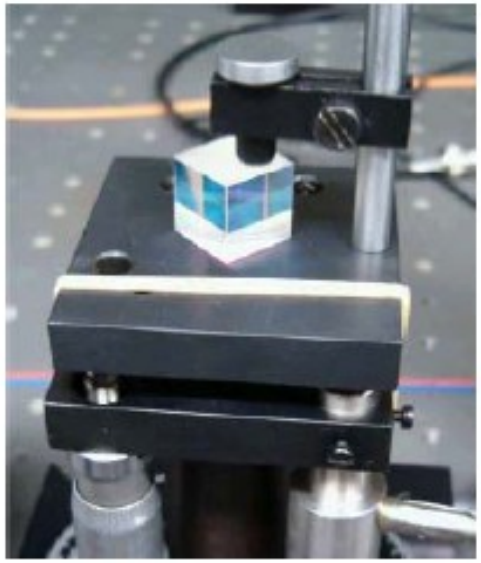
Vernamova šifra dokáže zajistit nepodmíněnou bezpečnost.

Kvantová kryptografie

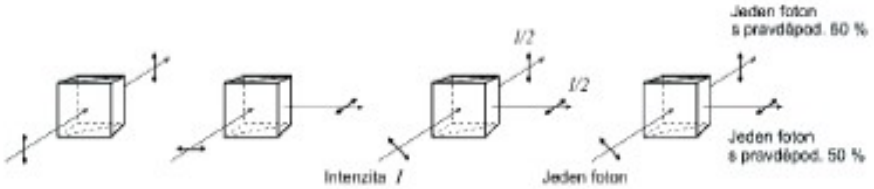
- existující protokoly využívající kvantových principů spoléhají na nemožnost tvorby identických kopií neznámého kvantového stavu.
- obvykle protokoly pro bezpečnou výměnu klíče
- schopnost detekce odposlechu
- zatím se nepoužívá pro uchovávání šifrovaných informací – obtížnost dlouhodobého zamezení interakce kvantového systému s okolním prostředím
- klasická kryptografie se spoléhá na výpočetní složitost

Polarizační kódování

- Praktická realizace – hranol z isladnského vápence
- rozdělí paprsek obsahující fotony s různou polarizací na dvě kolmé složky

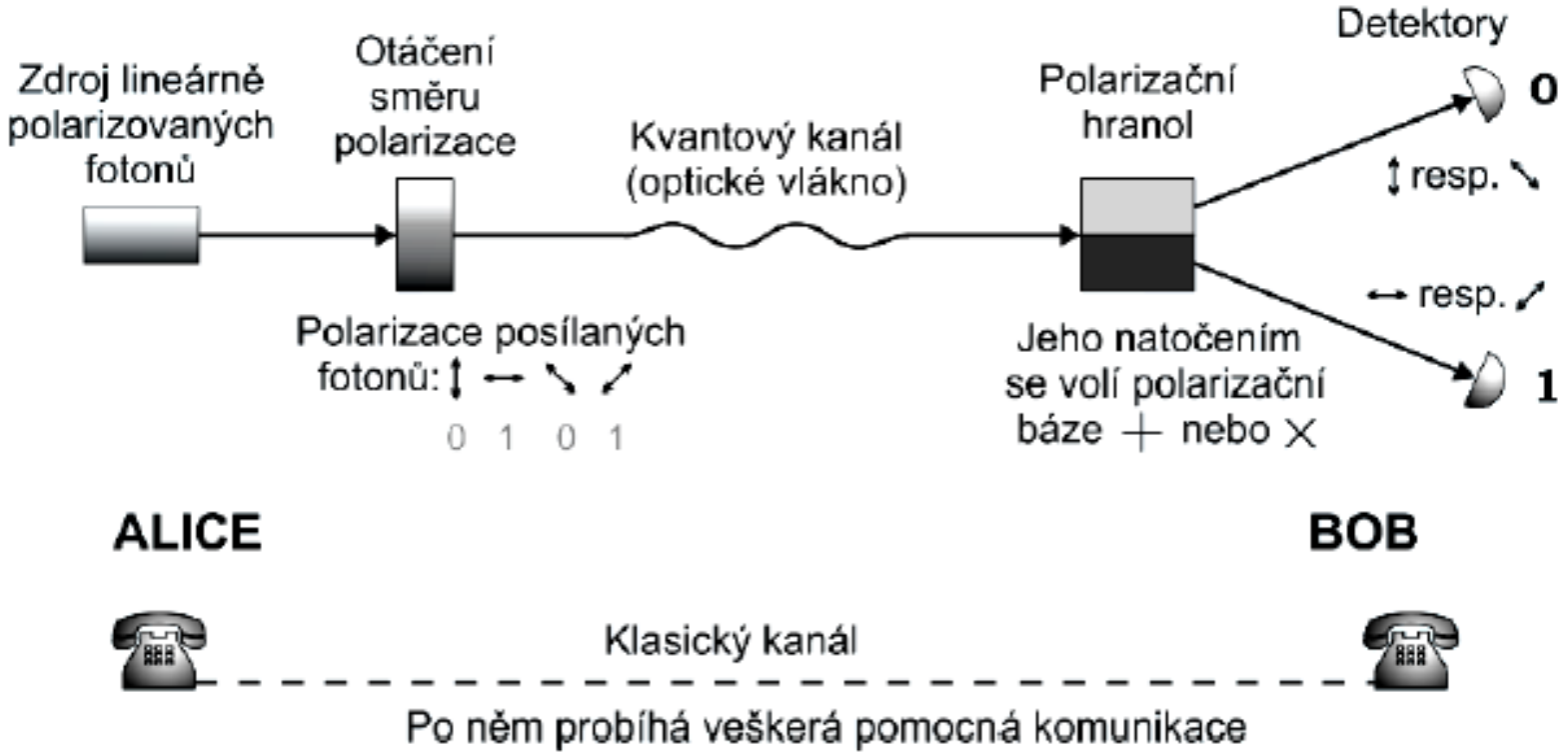


Obr. a foto – Laboratoř optiky, Univerzita Palackého, Olomouc



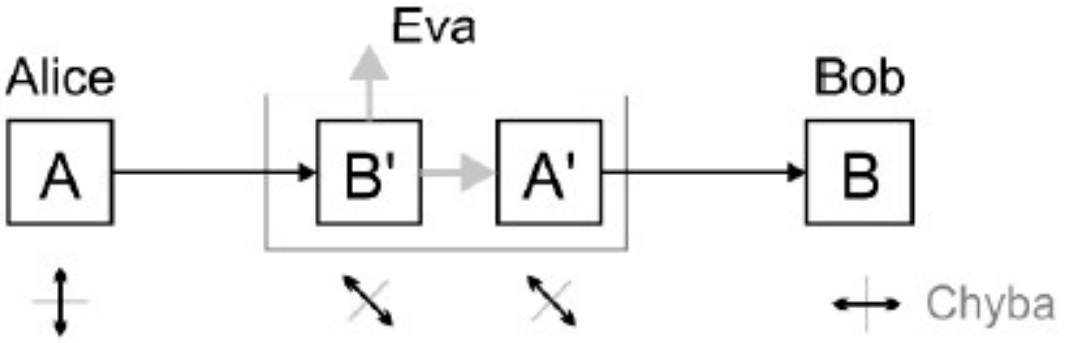
Foton se šikmou polarizací se buď odrazí nebo projde.
Po odrazu bude polarizován vodorovně, po průchodu svisle.

- Logické 0 a 1 jsou kódovány do dvou navzájem kolmých lineárních polarizací ze dvou polarizačních bází
- Báze jsou vůči sobě pootočeny o 45° (viz. Blochova koule)



Systém může fungovat i s jednou polarizací, ale případný útočník pak může s pravděpodobností ½ uhodnout polarizaci a odposlouchávat komunikaci.

Odposlech musí být aktivní – 1 foton nelze rozdělit na menší kvanta ani vytvořit jeho přesnou kopii. Útočník musí foton zachytit, změřit stejným zařízením jako má Bob a rychle ho znovu vyslat stejným zařízením jako má Alice.



Obrana proti odposlechu:

- polarizace pro každý foton se náhodně mění (na obou stranách - nezávisle na sobě)
- po přenosu si jiným kanálem sdělí jaké polarizace v daném kroku použili
- pokud oba použili stejnou bázi – bity si ponechají
- pokud použili různé báze – bity zahodí

- útočník tak má v každém kroku pravděpodobnost ½ , že zvolí bázi špatně
- při volbě špatné báze dojde s pravděpodobností ½ ke změně polarizace a tedy celkem ¼ přijatých bitů bude chybná

Alice a Bob srovnají část přenesených bitů -> zvýšená chybovost = odposlech

- srovnání 128 bitů – pravděpodobnost odhalení odposlechu

P = 1- (1-0,25)128 ~ 1-1,018.10-16 =0,999999999999998

Protokol BB84

- 1984
- Benett-Brassard
- slouží k bezpečné výměně klíče
- dohodnutý klíč je poté použit pro Vernamovu šifru
- založen na využití Heisenbergova principu neurčitosti ve spojení s polarizačním kódováním
- nejznámější kryptografický protokol využívající kvantové principy
- používán dodnes (s drobnými obměnami)
- využívá dvou kanálů
  - kvantový kanál slouží k přenosu šifrovacího klíče
  - jiný telekomunikační kanál slouží k přenosu šifrované zprávy
  - autentizace na druhém (ne-kvantovém) kanále se neřeší !

- kvantový kanál je realizován optickými vlákny
- vysoké nároky na kvalitu přenosové cesty

  - minimální útlum
  - malá disperze

Zpráva je přenášena pomocí fotonů s různou polarizací.

Rovina polarizace je pro každý bit volena absolutně náhodně a nezávisle.

V případě, že příjemce zvolí jinou bázi než odesílatel dostane náhodný výsledek s pravděpodobností ½

Čtyři možné roviny polarizace (viz diagram rovin).

Existují i protokoly se 6 nebo 2 polarizacemi.

K testování polarizace slouží měření ve dvou různých bázích.

Směr polarizace	Hodnota bitu	Báze
→	1	+
↑	0	+
↗	1	×
↖	0	×

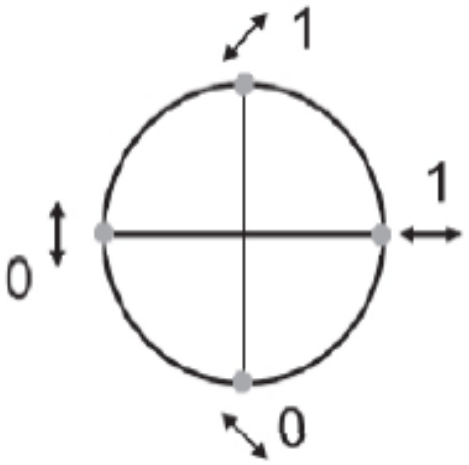


Diagram rovin

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0

Strana A generuje náhodné bity.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+

Strana A náhodně volí báze.

Strana A kóduje bity do polarizace fotonů a odesílá je příjemci B.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑
4.	×	×	+	×	+	+	+	+	+	×	×	×	×	+	+	+

Příjemce B náhodně volí báze.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑
4.	×	×	+	×	+	+	+	+	+	×	×	×	×	+	+	+
5.	↗	↖	→	↗	→	→	→	→	↑	↖	↗	↖	↖	↑	→	↑

Příjemce B měří přijaté fotony v předem zvolených bázích.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑
4.	×	×	+	×	+	+	+	+	+	×	×	×	×	+	+	+
5.	↗	↖	→	↗	→	→	→	→	↑	↖	↗	↖	↖	↑	→	↑
6.	1	0	1	1	1	1	1	1	0	0	1	0	0	0	1	0

Příjemce B dekóduje bity.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑
4.	×	×	+	×	+	+	+	+	+	×	×	×	×	+	+	+
5.	↗	↖	→	↗	→	→	→	→	↑	↖	↗	↖	↖	↑	→	↑
6.	1	0	1	1	1	1	1	1	0	0	1	0	0	0	1	0
7.			ok	ok				ok				ok				ok

Strany A a B se veřejně domluví, na kterých bázích se shodli.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑
4.	×	×	+	×	+	+	+	+	+	×	×	×	×	+	+	+
5.	↗	↖	→	↗	→	→	→	→	↑	↖	↗	↖	↖	↑	→	↑
6.	1	0	1	1	1	1	1	1	0	0	1	0	0	0	1	0
7.			ok	ok				ok				ok				ok
8.			1									0				

Strany A a B obětují libovolné (dohodnuté) množství bitů na detekci odposlechu.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑
4.	×	×	+	×	+	+	+	+	+	×	×	×	×	+	+	+
5.	↗	↖	→	↗	→	→	→	→	↑	↖	↗	↖	↖	↑	→	↑
6.	1	0	1	1	1	1	1	1	0	0	1	0	0	0	1	0
7.			ok	ok				ok				ok				ok
8.			1									0				
9.			ok									ok				

Pokud je většina kontrolovaných bitů shodná, znamená to, že nedošlo k odposlechu.

1.	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	0
2.	+	+	+	×	×	+	×	+	×	+	+	×	×	+	+	+
3.	→	↑	→	↗	↗	↑	↖	→	↖	↑	↑	↖	↗	→	↑	↑
4.	×	×	+	×	+	+	+	+	+	×	×	×	×	+	+	+
5.	↗	↖	→	↗	→	→	→	→	↑	↖	↗	↖	↖	↑	→	↑
6.	1	0	1	1	1	1	1	1	0	0	1	0	0	0	1	0
7.			ok	ok				ok				ok				ok
8.			1									0				
9.			ok									ok				
10.				1				1								0

Zbylé bity tvoří klíč, který bude použit pro vlastní šifrování (v tomto případě 110).

Dnes se navíc provádí tzv. *zesílení soukromí* (*privacy amplification*).
Z množství chyb detekovaných během přenosu se určí jak moc byl kanál odposloucháván resp. kolik informace z něj mohl útočník získat.

Strany A a B pak předem definovaným způsobem převedou dohodnutý klíč na jiný kratší např. pomocí hashovací funkce.

Cílem je minimalizovat útočníkovi informace o klíči.