Kryptológia – úvod

Michal Rjaško

Katedra informatiky
FMFI UK, Bratislava
rjasko@dcs.fmph.uniba.sk

Úvod do informačnej bezpečnosti (LS 2013/2014)

Obsah

Informačná bezpečnosť a kryptológia

- Ukradnutý / zabudnutý notebook s dôvernými údajmi
  - Disk bol hardvérovo šifrovaný
  - Cold Boot útok
  - Evil maid útok
- WPS (WiFi Protected Setup)
- Náhodnosť kryptografických kľúčov (napr. Debian)
- Šifrované USB kľúče (certifikované FIPS 140-2)
- Timing útoky
- Kolízie v hašovacích tabuľkách
- ...

Úvod – kryptológia

- kryptológia – dôležitá súčasť informačnej bezpečnosti
- informačná bezpečnosť $\setminus$ kryptológia $=$???
- kryptológia $=$ kryptografia $+$ kryptoanalýza
- kryptografia slúži na zabezpečenie:
  - dôvernosti – šifrovanie
  - integrity a autentickosti – autentizačné kódy, podpisy
- ďalšie objekty záujmu:
  - zdieľanie tajomstva,
  - (overiteľné) výpočty nad súkromnými dátami
  - e-cash
  - e-voľby
  - ...

Úvod – kryptológia

- kryptológia – dôležitá súčasť informačnej bezpečnosti
- informačná bezpečnosť $\setminus$ kryptológia $=$???
- kryptológia $=$ kryptografia $+$ kryptoanalýza
- kryptografia slúži na zabezpečenie:
  - dôvernosti – šifrovanie
  - integrity a autentickosti – autentizačné kódy, podpisy
- ďalšie objekty záujmu:
  - zdieľanie tajomstva,
  - (overiteľné) výpočty nad súkromnými dátami
  - e-cash
  - e-voľby
  - ...

Kryptológia

- kryptológia
  - matematika
  - detaily sú podstatné
  - implementácia a použitie
- kvalitná kryptografia je nutná, ale nie postačujúca
- poskytuje (falošný ?) pocit bezpečia
  - „šifrujeme“ – ako? mód? správa kľúčov? kontext? . . .
  - „podpisujeme“ – ako? implementácia? správa kľúčov? . . .

Kryptológia

- kryptológia
  - matematika
  - detaily sú podstatné
  - implementácia a použitie
- kvalitná kryptografia je nutná, ale nie postačujúca
- poskytuje (falošný ?) pocit bezpečia
  - „šifrujeme“ – ako? mód? správa kľúčov? kontext? . . .
  - „podpisujeme“ – ako? implementácia? správa kľúčov? . . .

Kryptológia a falošný pocit bezpečia

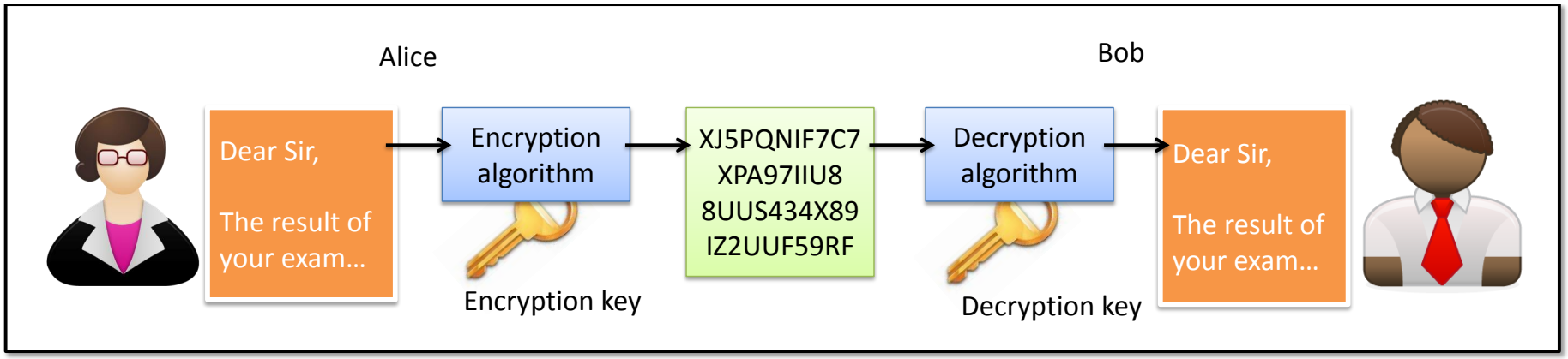
Q: Is your site secure?

A: Yes, our site is completely secure and features 128-bit SSL (Secure Socket Layer) encryption. This ensures that your payment information is secure and safe.

Symetrické šifrovanie – úvod 1

- klasický cieľ kryptografie – dôverný prenos dát
- komunikujúce subjekty zdieľajú tajný kľúč
- kľúč je rovnaký pre odosielateľa aj príjemcu $\Rightarrow$ symetrické šifrovanie
- otvorený text = pôvodná správa, text, dokument, dáta
- šifrový text = zašifrovaný text, výstupné dáta šifrovacieho algoritmu

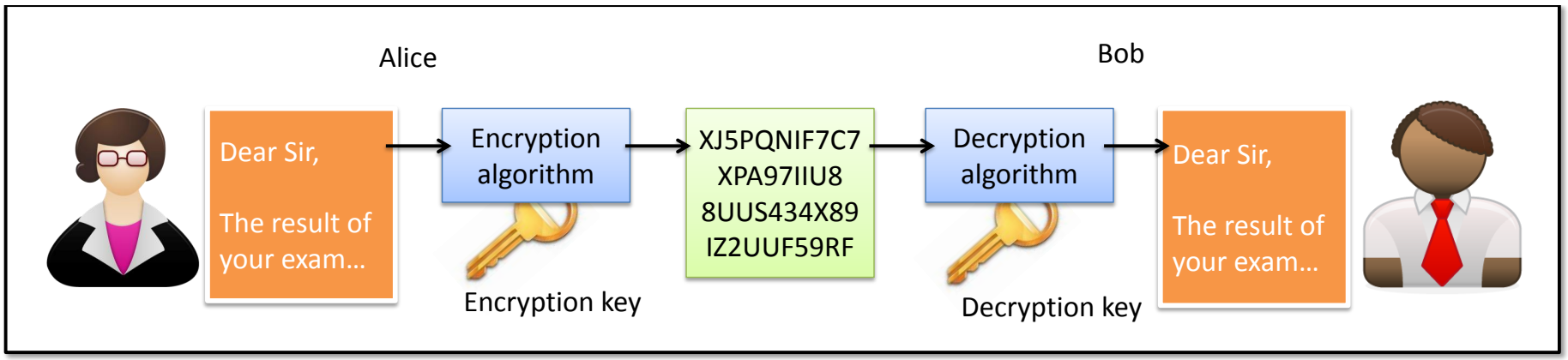
Symetrické šifrovanie – úvod 2



- šifrovanie: $E : P \times K \to C$
- dešifrovanie: $D : C \times K \to P$

- čo očakávame od E, D?

Symetrické šifrovanie – úvod 2



- šifrovanie: $E : P \times K \to C$
- dešifrovanie: $D : C \times K \to P$

- čo očakávame od E, D?

Symetrické šifrovanie – úvod 3

Očakávame:

- korektnosť:

$$\forall k \in K \; \forall p \in P : \quad D_k(E_k(p)) = p$$

- bezpečnosť: ako definovať?
- Kerckhoffov princíp:

  bezpečnosť šifrovania nezávisí na utajení algoritmu, ale výlučne na utajení kľúča

  (vs. security by obscurity)

Ako definovať bezpečnosť šifrovacej schémy?

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*
  - možno nám stačí len časť OT

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*
  - možno nám stačí len časť OT
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie určiť ani jeden znak otvoreného textu k danému šifrovému textu*

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*
  - možno nám stačí len časť OT
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie určiť ani jeden znak otvoreného textu k danému šifrovému textu*
  - možno nám stačí určiť, či daná hodnota v OT nie je väčšia ako 10 000.

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*
  - možno nám stačí len časť OT
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie určiť ani jeden znak otvoreného textu k danému šifrovému textu*
  - možno nám stačí určiť, či daná hodnota v OT nie je väčšia ako 10 000.
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať akúkoľvek zmysluplnú informáciu o otvorenom texte pre daný šifrový text*

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*
  - možno nám stačí len časť OT
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie určiť ani jeden znak otvoreného textu k danému šifrovému textu*
  - možno nám stačí určiť, či daná hodnota v OT nie je väčšia ako 10 000.
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať akúkoľvek zmysluplnú informáciu o otvorenom texte pre daný šifrový text*
  - čo znamená "zmysluplná informácia"?

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*
  - možno nám stačí len časť OT
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie určiť ani jeden znak otvoreného textu k danému šifrovému textu*
  - možno nám stačí určiť, či daná hodnota v OT nie je väčšia ako 10 000.
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať akúkoľvek zmysluplnú informáciu o otvorenom texte pre daný šifrový text*
  - čo znamená "zmysluplná informácia"?
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie vypočítať akúkoľvek funkciu nad otvoreným textom pre daný šifrový text*

Ako definovať bezpečnosť šifrovacej schémy?

- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať kľúč*
  - možno nám stačí poznať otvorený text
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať otvorený text k danému šifrovému textu*
  - možno nám stačí len časť OT
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie určiť ani jeden znak otvoreného textu k danému šifrovému textu*
  - možno nám stačí určiť, či daná hodnota v OT nie je väčšia ako 10 000.
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie získať akúkoľvek zmysluplnú informáciu o otvorenom texte pre daný šifrový text*
  - čo znamená "zmysluplná informácia"?
- *Šifrovacia schéma je bezpečná, ak žiadny útočník nevie vypočítať akúkoľvek funkciu nad otvoreným textom pre daný šifrový text*
  - blízke skutočnej definícii

Vernamova šifra (one-time pad)

- správa $m = m_1, m_2, \ldots, m_t \in \{0,1\}^t$
- kľúč $k = k_1, k_2, \ldots, k_t \in \{0,1\}^t$
- šifrovanie: $c = m \oplus k \quad (c_i = m_i \oplus k_i)$
- dešifrovanie: $c \oplus k = (m \oplus k) \oplus k = m$

Vernamova šifra (one-time pad)

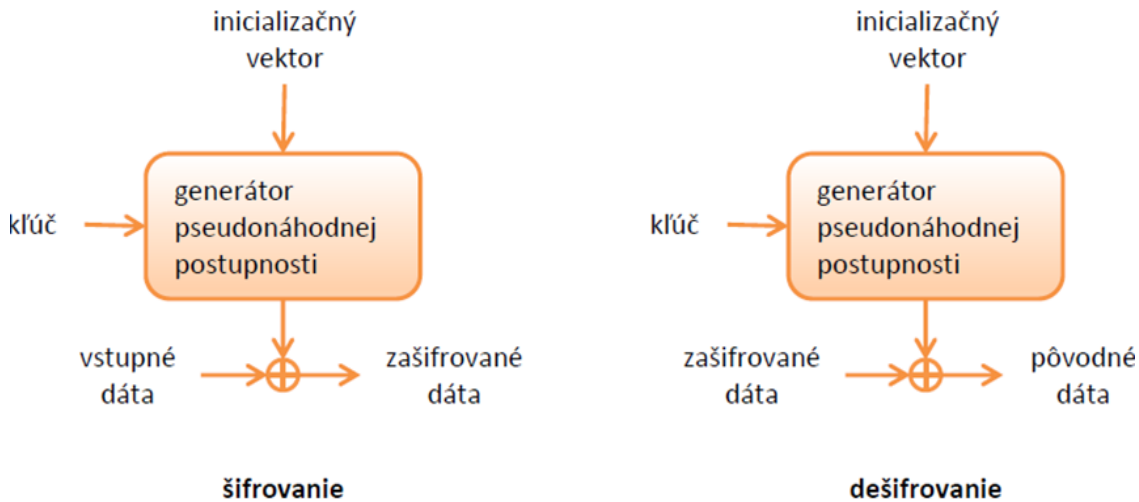
- výhody:
  - jednoduché (rýchle) šifrovanie a dešifrovanie
    - operácia XOR je veľmi rýchla
  - absolútne bezpečná šifra
    - Pre daný šifrový text, každý otvorený text je rovnako pravdepodobný (bez ohľadu na výpočtovú silu útočníka)
    - . . . iba v prípade, že kľúč je úplne náhodný
    - . . . iba v prípade, že kľúč je rovnako dlhý ako otvorený text
- nevýhody:
  - kľúč rovnako dlhý ako otvorený text
  - neposkytuje integritu
    - útočník nemôže získať otvorený text, ale môže ho cestou pozmeniť
  - „jednorazový“ kľúč
    - útočník by inak mohol získať XOR otvorených textov

Vernamova šifra (one-time pad)

- výhody:
  - jednoduché (rýchle) šifrovanie a dešifrovanie
    - operácia XOR je veľmi rýchla
  - absolútne bezpečná šifra
    - Pre daný šifrový text, každý otvorený text je rovnako pravdepodobný (bez ohľadu na výpočtovú silu útočníka)
    - . . . iba v prípade, že kľúč je úplne náhodný
    - . . . iba v prípade, že kľúč je rovnako dlhý ako otvorený text
- nevýhody:
  - kľúč rovnako dlhý ako otvorený text
  - neposkytuje integritu
    - útočník nemôže získať otvorený text, ale môže ho cestou pozmeniť
  - „jednorazový“ kľúč
    - útočník by inak mohol získať XOR otvorených textov

Prúdové šifry

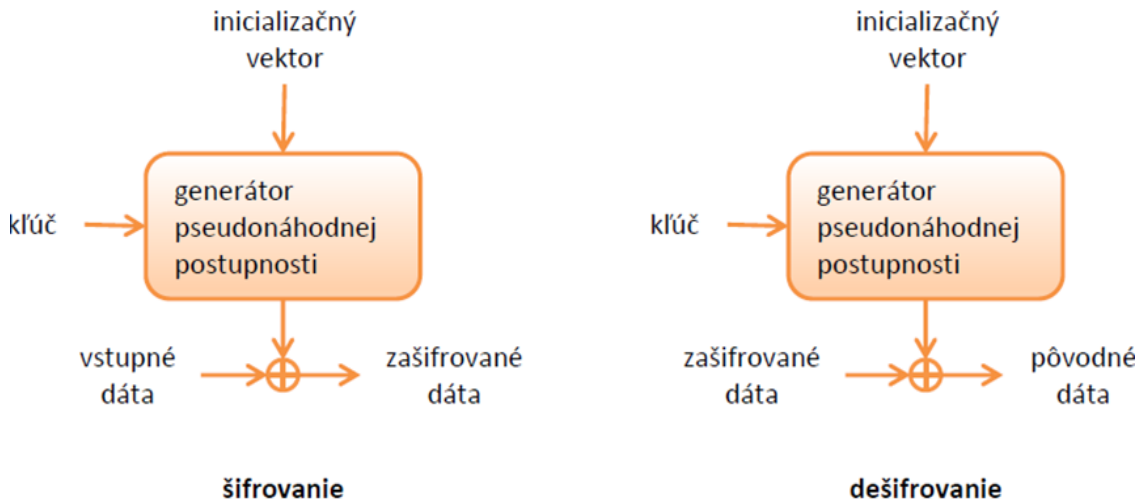
- krátky kľúč použitý na (deterministické) generovanie bežiaceho kľúča
- zväčša používané – (aditívne) synchrónne prúdové šifry
- najznámejšie prúdové šifry: RC4 (softvér, WiFi), A5 (GSM), E0 (Bluetooth)



- zvyčajné výhody (oproti blokovým šifrám):
  - vhodné pre prúd OT, jednoduchší algoritmus, rýchlejšie šifrovanie/dešifrovanie
- zvyčajné nevýhody:
  - vyžadujú synchronizáciu, bez akejkoľvek integrity, zložitý „seek“

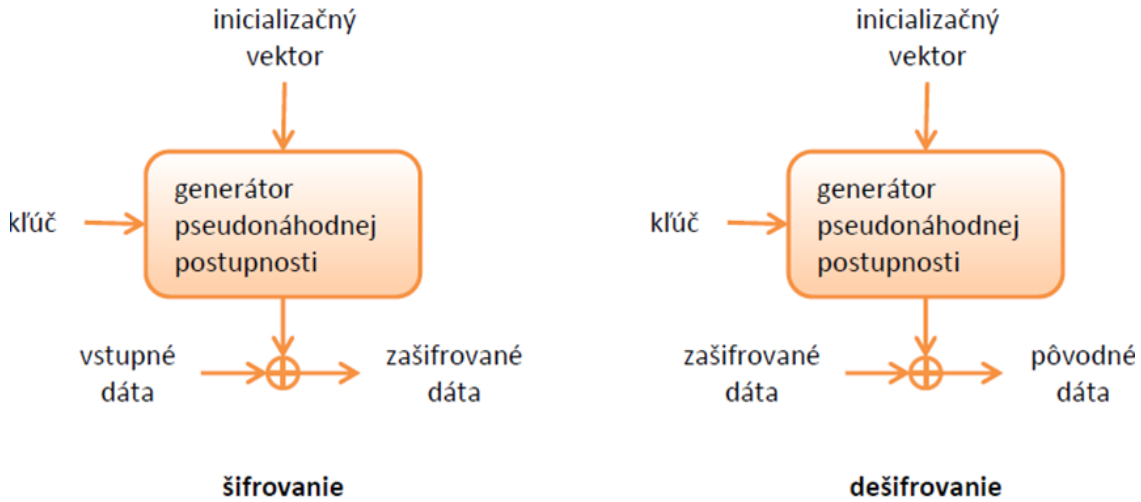
Prúdové šifry

- krátky kľúč použitý na (deterministické) generovanie bežiaceho kľúča
- zväčša používané – (aditívne) synchrónne prúdové šifry
- najznámejšie prúdové šifry: RC4 (softvér, WiFi), A5 (GSM), E0 (Bluetooth)



- zvyčajné výhody (oproti blokovým šifrám):
  - vhodné pre prúd OT, jednoduchší algoritmus, rýchlejšie šifrovanie/dešifrovanie
- zvyčajné nevýhody:
  - vyžadujú synchronizáciu, bez akejkoľvek integrity, zložitý „seek“

Prúdové šifry



- je dôležité aby sa nepoužil ten istý kľúč a IV 2x.
- napr. v prípade zabezpečenia WEP vo WiFi sieťach má inicializačný vektor dĺžku 24 bitov
  - po prenesení 2^{24} správ sa IV zopakuje – útočník môže získať XOR otvorených textov.

Blokové šifry

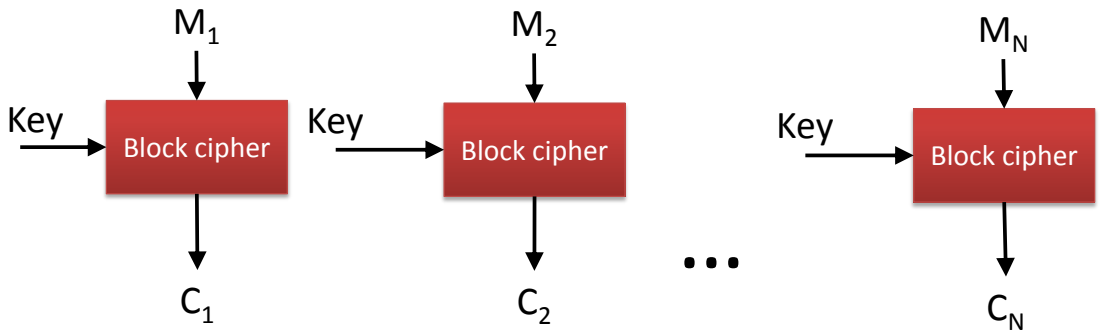
- šifrovanie/dešifrovanie blokov dát: $E_k, D_k : \{0,1\}^n \to \{0,1\}^n$
- E_k a D_k sú inverzné bijekcie
- mód – spôsob šifrovanie dlhých OT, napr.
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)
  - CTR (Counter)
- najpoužívanejšie blokové šifry: AES, (3)DES
- zvyčajné výhody (oproti prúdovým šifrám):
  - štandardy
  - flexibilita - vieme prispôsobiť mód blokovej šifry podľa potreby
    - šifrovanie disku, paralelné šifrovanie, iba sekvenčné, ...
- zvyčajné nevýhody:
  - vo všeobecnosti sú pomalšie ako prúdové šifry (ale nie výrazne)

Blokové šifry

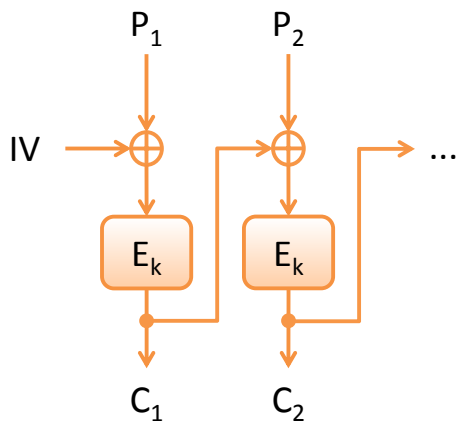
- šifrovanie/dešifrovanie blokov dát: $E_k, D_k : \{0,1\}^n \to \{0,1\}^n$
- E_k a D_k sú inverzné bijekcie
- mód – spôsob šifrovanie dlhých OT, napr.
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)
  - CTR (Counter)
- najpoužívanejšie blokové šifry: AES, (3)DES
- zvyčajné výhody (oproti prúdovým šifrám):
  - štandardy
  - flexibilita - vieme prispôsobiť mód blokovej šifry podľa potreby
    - šifrovanie disku, paralelné šifrovanie, iba sekvenčné, ...
- zvyčajné nevýhody:
  - vo všeobecnosti sú pomalšie ako prúdové šifry (ale nie výrazne)

Módy blokových šifier

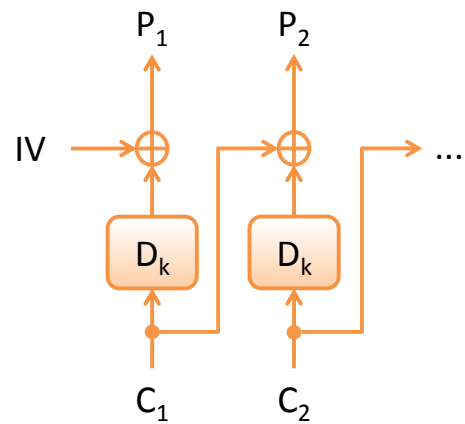
- ECB (Electronic Code Book)



- CBC (Cipher Block Chaining)



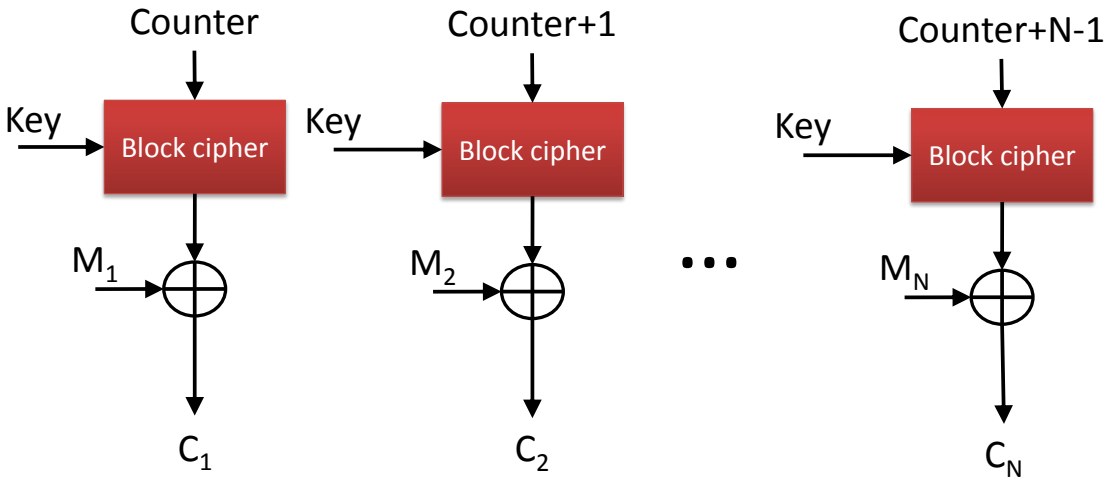
šifrovanie v CBC móde



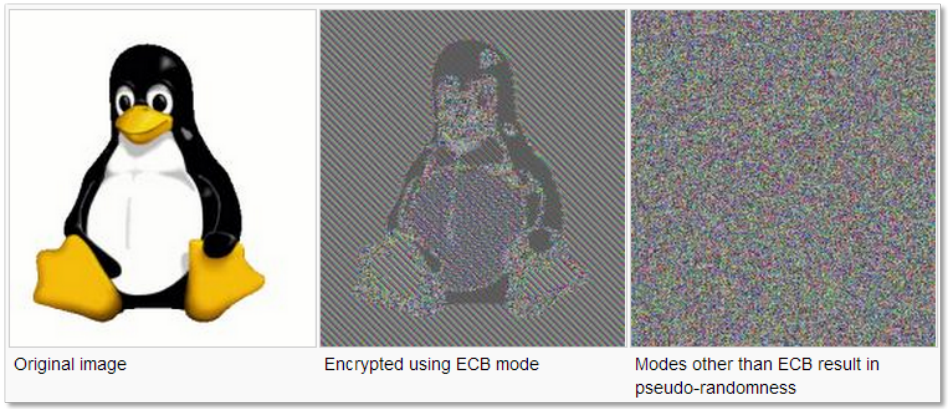
dešifrovanie v CBC móde

Módy blokových šifier

- CTR (Counter)

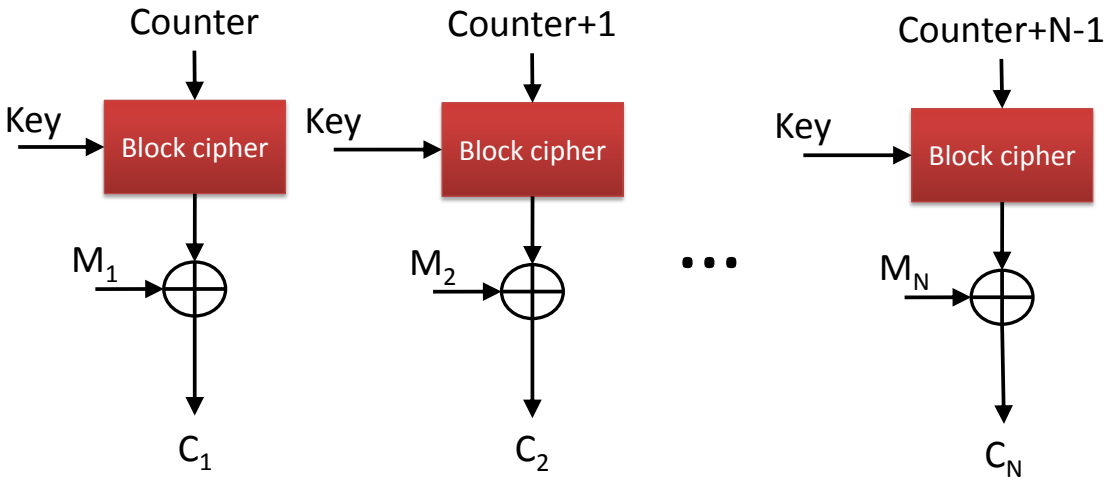


- ECB vs. CBC

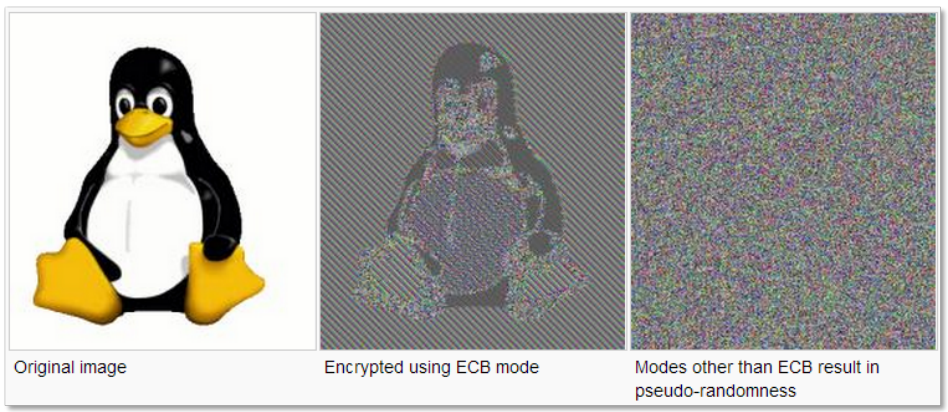


Módy blokových šifier

- CTR (Counter)



- ECB vs. CBC



Typy útokov na šifrovacie algoritmy

- základné útoky:
  - (COA) len so znalosťou šifrového textu
  - (KPA) so znalosťou otvoreného textu
  - (CPA) s možnosťou voľby otvoreného textu
  - (CCA) s možnosťou voľby šifrového textu
- ciele útokov:
  - získať kľúč
  - dešifrovať neznámy ŠT
  - zašifrovať nový OT
  - **rozlíšiť, ktorému z dvoch (útočníkom vybraných) OT zodpovedá daný ŠT**
  - ...

Typy útokov na šifrovacie algoritmy

- základné útoky:
  - (COA) len so znalosťou šifrového textu
  - (KPA) so znalosťou otvoreného textu
  - (CPA) s možnosťou voľby otvoreného textu
  - (CCA) s možnosťou voľby šifrového textu
- ciele útokov:
  - získať kľúč
  - dešifrovať neznámy ŠT
  - zašifrovať nový OT
  - **rozlíšiť, ktorému z dvoch (útočníkom vybraných) OT zodpovedá daný ŠT**
  - ...

Štandardy (blokové šifry)

- AES (Advanced Encryption Standard)
  - algoritmus Rijndael
  - verejný výber štandardu (NIST, 1997-2001)
  - štandard: FIPS PUB 197, 2001
  - bloková šifra, 128 bitov dlhý blok
  - variabilná dĺžka kľúča: 128, 192, 256 bitov
- DES/3DES
  - predchádzajúci štandard, 70-te roky 20. storočia
  - bloková šifra, 64 bitov dlhý blok (málo!)
  - DES: dĺžka kľúča 56 bitov (málo!)
  - 3DES: dĺžka kľúča 168 (resp. 112) bitov
    - Meet in the middle útok

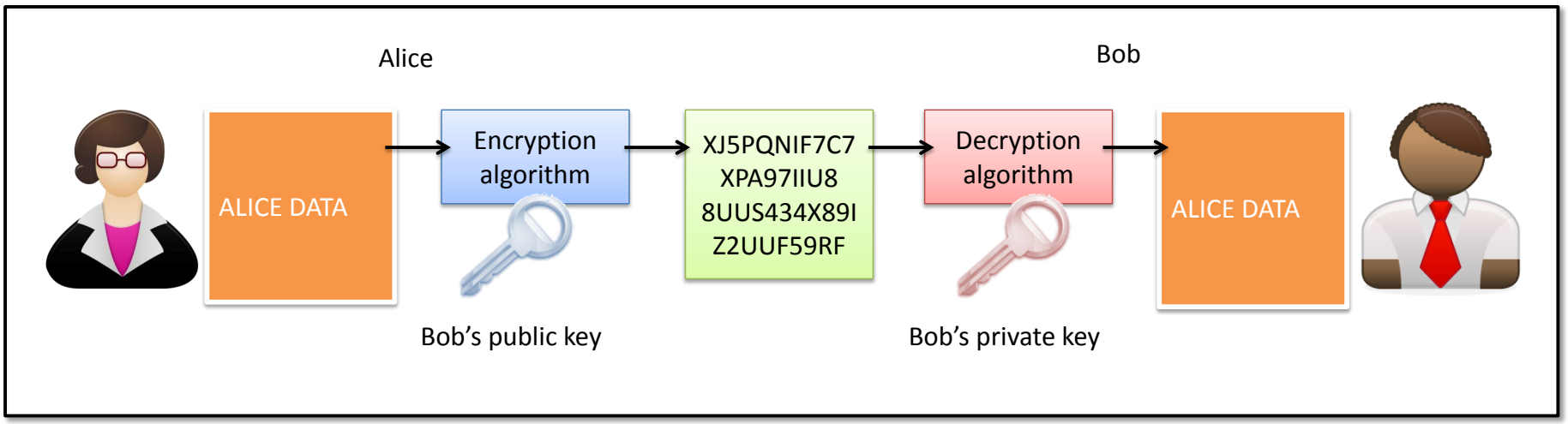
Výkon – softvérové implementácie

- knižnica Crypto++ v.5.6.0, (Windows Vista)
- hardvér: Intel Core 2, 1.83 GHz

alg.	**MB/s**
AES-128	109
AES-192	92
AES-256	82
3DES	13

- špecializovaný HW: AES-128 $\sim$ 21 Gbit/s

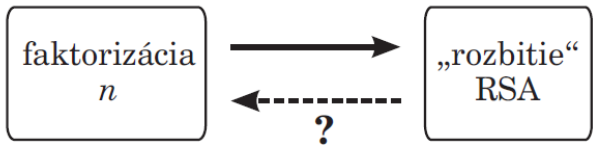
Asymetrické šifrovanie



- dvojica rôznych kľúčov:
  - verejný – šifrovanie $\Rightarrow$ ktokoľvek vie šifrovať
  - súkromný – dešifrovanie $\Rightarrow$ len vlastník vie dešifrovať
- jednoduchšia správa kľúčov
- bezpečnosť:
  - CPA útok je vždy možný
  - verejný kľúč $\not\rightarrow$ algoritmus na dešifrovanie
- najznámejšie systémy: RSA, ElGamal

RSA

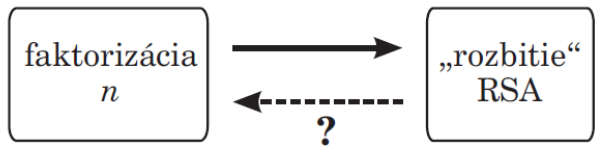
- (1978) Rivest, Shamir, Adleman
- bezpečnosť súvisí s problémom faktorizácie veľkých čísel



- konštrukcia:
  - $n = p \cdot q$, p, q sú veľké prvočísla
  - $e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$
  - verejný kľúč: (e, n) (najčastejšia voľba $e = 65537$).
  - súkromný kľúč: d
- šifrovanie $(E : Z_n \to Z_n)$: $E(m) = m^e \bmod n$
- dešifrovanie $(D : Z_n \to Z_n)$: $D(c) = c^d \bmod n$
- RSA je bijekcia
- RSA je deterministické (fuj!) $\Rightarrow$ v praxi sa znáhodňuje

RSA

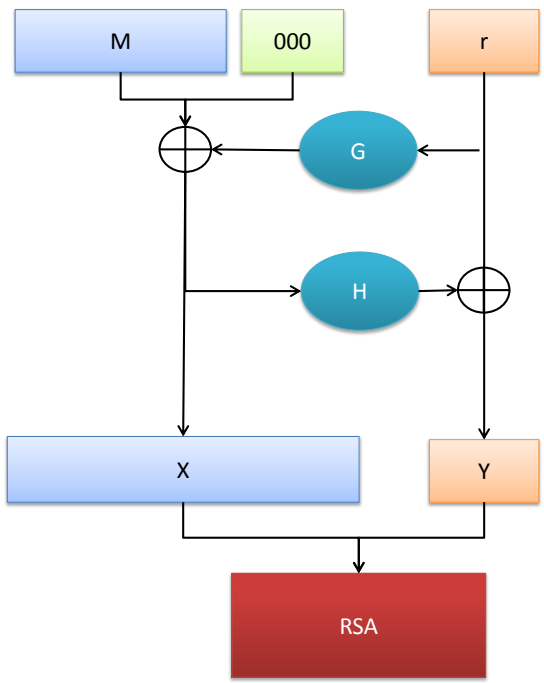
- (1978) Rivest, Shamir, Adleman
- bezpečnosť súvisí s problémom faktorizácie veľkých čísel



- konštrukcia:
  - $n = p \cdot q$, p, q sú veľké prvočísla
  - $e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$
  - verejný kľúč: (e, n) (najčastejšia voľba $e = 65537$).
  - súkromný kľúč: d
- šifrovanie $(E : Z_n \to Z_n)$: $E(m) = m^e \bmod n$
- dešifrovanie $(D : Z_n \to Z_n)$: $D(c) = c^d \bmod n$
- RSA je bijekcia
- RSA je deterministické (fuj!) $\Rightarrow$ v praxi sa znáhodňuje

RSA – štandardy

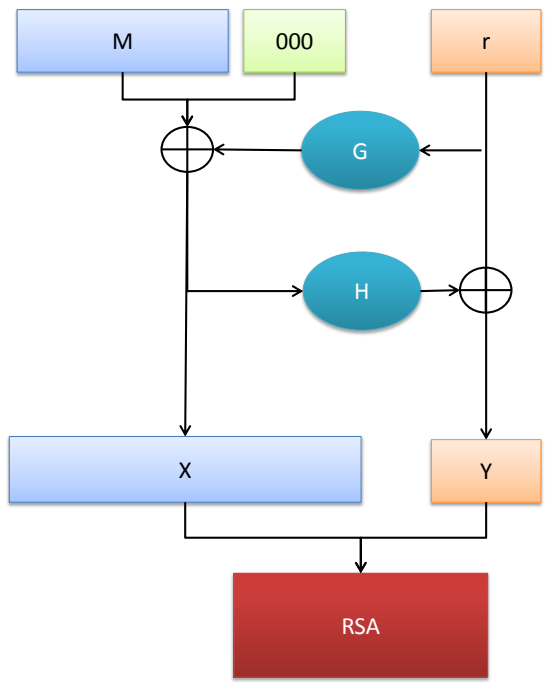
- RSA PKCS #1 v1.5 (žiadny dôkaz bezpečnosti)
  - veľmi rozšírené a v praxi používané
  - verí sa, že je CPA bezpečné
  - známy CCA útok
- RSA PKCS #1 v2.1 – RSA-OAEP
  - znáhodnený padding (zarovnanie)
  - „dôkaz“ bezpečnosti (v modeli s náhodným orákulom)



RSA – štandardy

- RSA PKCS #1 v1.5 (žiadny dôkaz bezpečnosti)
  - veľmi rozšírené a v praxi používané
  - verí sa, že je CPA bezpečné
  - známy CCA útok

- RSA PKCS #1 v2.1 – RSA-OAEP
  - znáhodnený padding (zarovnanie)
  - „dôkaz“ bezpečnosti (v modeli s náhodným orákulom)



Výkon – softvérové implementácie

- knižnica Crypto++ v.5.6.0, (Windows Vista)
- hardvér: Intel Core 2, 1.83 GHz

alg.	**ms/oper.**
RSA-1024 šifr.	0.08
RSA-1024 dešifr.	1.46
RSA-2048 šifr.	0.16
RSA-2048 dešifr.	6.08

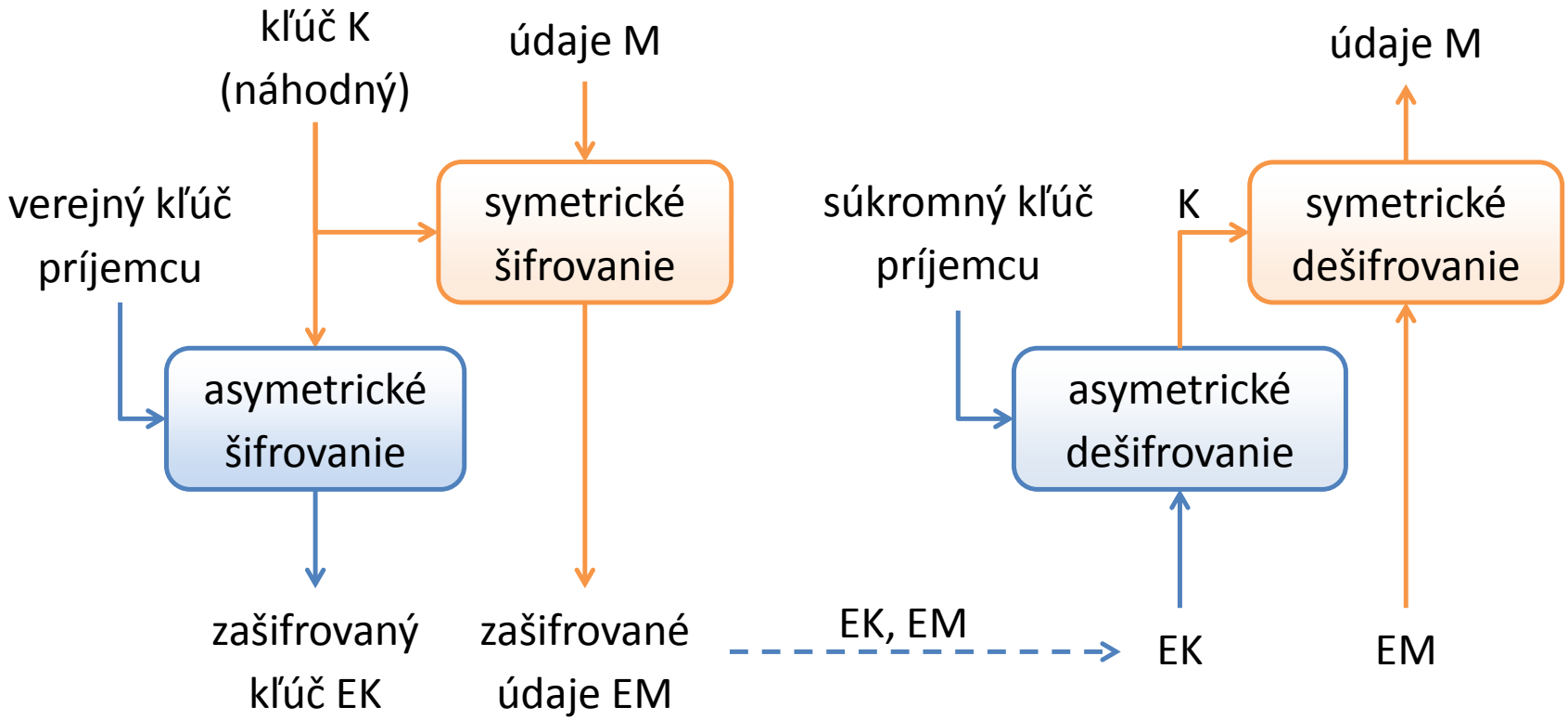
Hybridné šifrovanie

- asymetrické šifry sú pomalé (v porovnaní so symetrickými)
- čo s prenosom objemných dát?
- riešenie:
  - šifrujme symetricky s náhodným kľúčom k
  - kľúč k zašifrujeme asymetricky pre adresáta

$$\langle AES_k(m), E_A^{\mathrm{RSA}}(k) \rangle$$

- v praxi špeciálne schémy (KEM – Key Encapsulation Method), napr. RSA-KEM (ISO/IEC 18033-2)

Hybridné šifrovanie



Symetrické vs. asymetrické šifrovanie

	Symetrické šifrovanie	Asymetrické šifrovanie
Primárne použitie	dôvernosť údajov ľubovoľného rozsahu	dôvernosť krátkych dát (typicky napr. kľúče pre symetrické šifrovanie)
Komunikácia	1:1 – obvykle dvaja účastníci	N:1 – ľubovoľný počet odosielateľov (šifrovací kľúč je verejný), jeden príjemca (súkromný dešifrovací kľúč)
Efektívnosť	rýchle šifrovanie aj dešifrovanie	pomalé šifrovanie aj dešifrovanie
Dĺžka kľúčov	obvykle 112 až 256 bitov (náhodný reťazec bitov)	v závislosti na konkrétnom algoritme, niekoľko sto až niekoľko tisíc bitov

Hašovacie funkcie

- funkcia $h : \{0,1\}^* \to \{0,1\}^n$
- „odtlačok“ správy, dokumentu
- kontrola integrity (detekcia náhodnej/neúmyselnej zmeny dát), digitálne podpisy
- kryptografické vlastnosti:
  - jednosmernosť: pre dané y nájsť x: $h(x) = y$
  - odolnosť voči kolízám: nájsť $x \neq x'$: $h(x) = h(x')$
- najznámejšie hašovacie funkcie: MD5, SHA1, SHA-(224,256,384,512)

Hašovacie funkcie - načo sú dobré?

"Modern, collision resistant hash functions were designed to create small, fixed size message digests so that a digest could act as a proxy for a possibly very large variable length message in a **digital signature algorithm**, such as RSA or DSA. These hash functions have since been widely used for many other "ancillary" applications, including hash-based **message authentication codes**, **pseudo random number generators**, and **key derivation functions**."

"Request for Candidate Algorithm Nominations",
-- NIST, November 2007

Bezpečnosť hašovacích funkcií

- nedávne výsledky:
  - nájdené kolízie v MD5 (2005)
  - kolízie v certifikátoch s MD5 (!)
  - kolízie $\sim 2^{24}$ (2007)
  - SHA-1 (160 bitov) kolízie $\sim 2^{57}$ (2010)
- generický útok – narodeninový útok
  - hľadanie kolízií
  - využíva tzv. „narodeninový“ paradox
  - zložitosť útoku $O(2^{n/2})$

Výkon – softvérové implementácie

- knižnica Crypto++ v.5.6.0, (Windows Vista)
- hardvér: Intel Core 2, 1.83 GHz

alg.	**MB/s**
MD5	255
SHA-1	153
SHA-256	111
SHA-512	99
Whirlpool	57

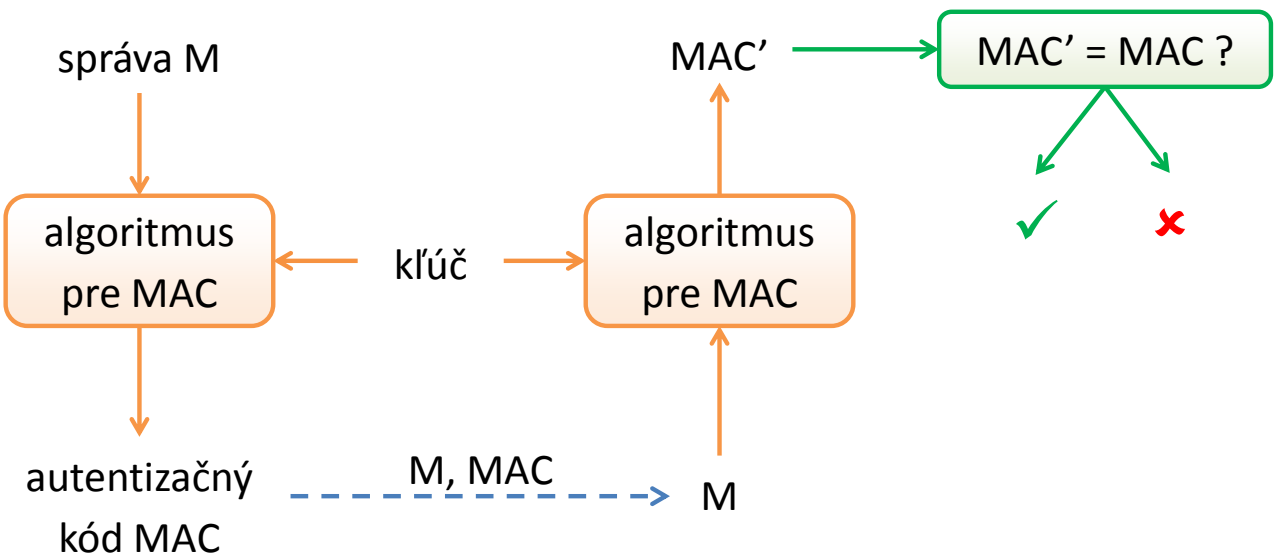
Budúcnosť

- od roku 2007 verejný výber novej sady algoritmov (SHA-3)
- organizoval NIST
- finalisti: BLAKE, Grøstl, JH, Keccak, Skein
- výherca Keccak vybraný 2. októbra 2012
- detaily: `csrc.nist.gov/groups/ST/hash/sha-3/index.html`
- štandard očakávaný v roku 2014

alg.	**MB/s**
MD5	255
SHA-1	153
Keccak-256	135
SHA-256	111
SHA-512	99
Whirlpool	57

Autentizačné kódy správ (MAC)

- symetrický kľúč (odosielateľ a príjemca)
- zabezpečenie integrity a autentickosti správ (bez nepopierateľnosti !)
- rýchle (oproti digitálnym podpisom)
- použitie napr. SSL/TLS, IPSec
- konštrukcia - najčastejšie pomocou hašovacej funkcie s kľúčom (symetrické)

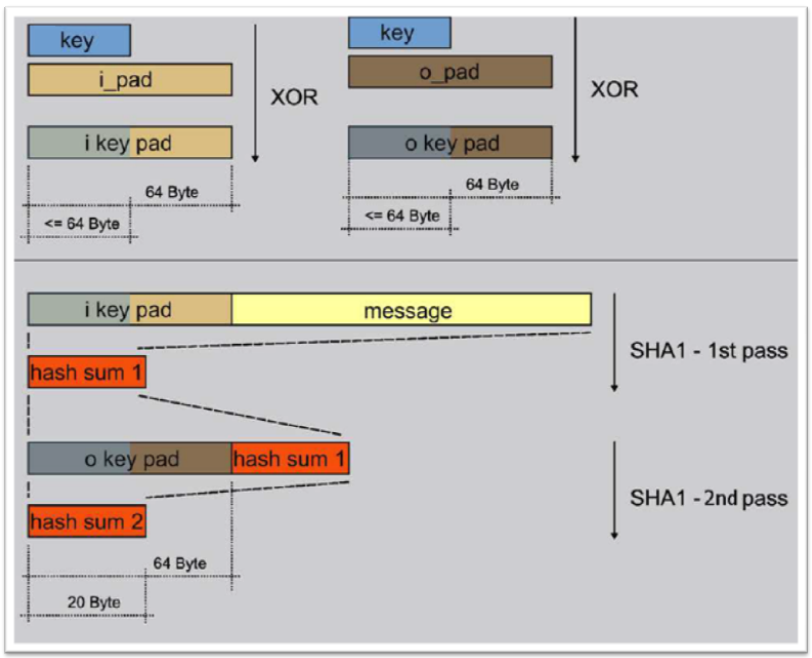


Autentizačné kódy správ - HMAC

- najznámejšia konštrukcia: HMAC (RFC 2104)

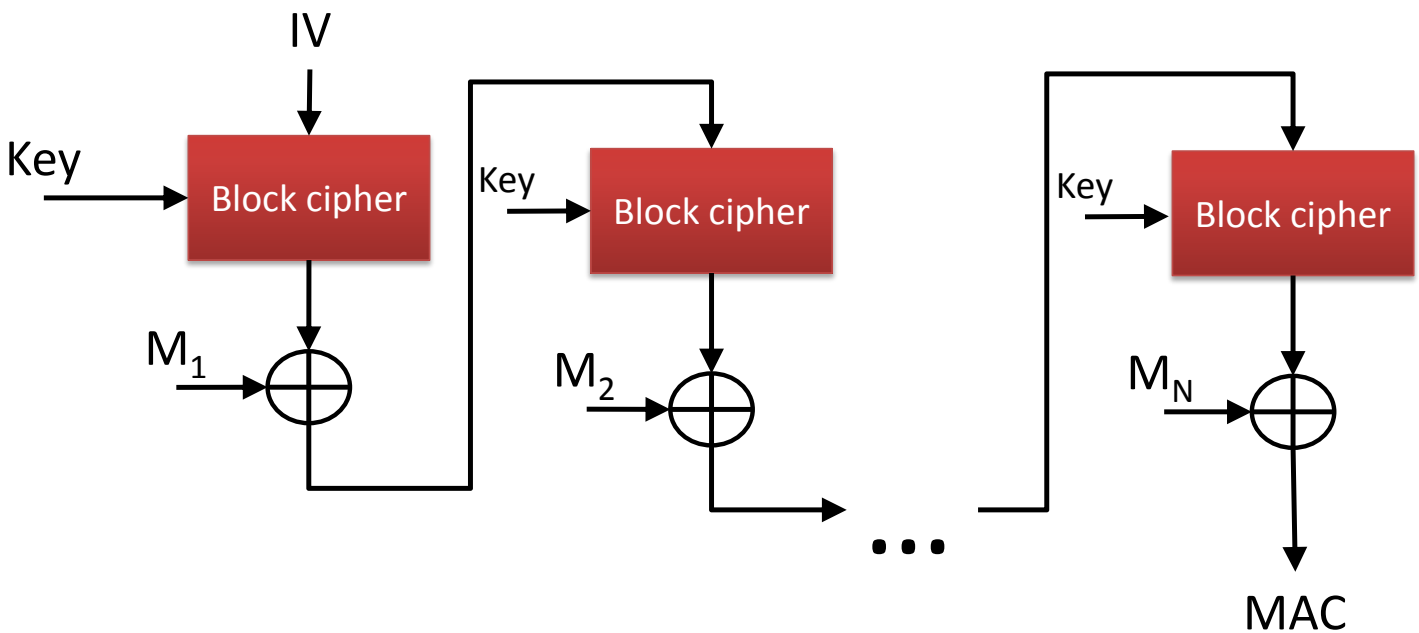
$$\mathrm{MAC}_k(m) = H(k \oplus \mathrm{opad} \,||\, H(k \oplus \mathrm{ipad} \,||\, x)),$$

- pre HMAC-MD5/SHA1 je opad $= (0\text{x}5\text{C})^{64}$, ipad $= (0\text{x}36)^{64}$
- použitie napr. SSL/TLS



Autentizačné kódy správ - CBC-MAC

- CBC-MAC: Konštrukcia MAC z blokovej šifry
- použitie napr. WiFi - WPA2 (CBC-MAC so šifrou AES)



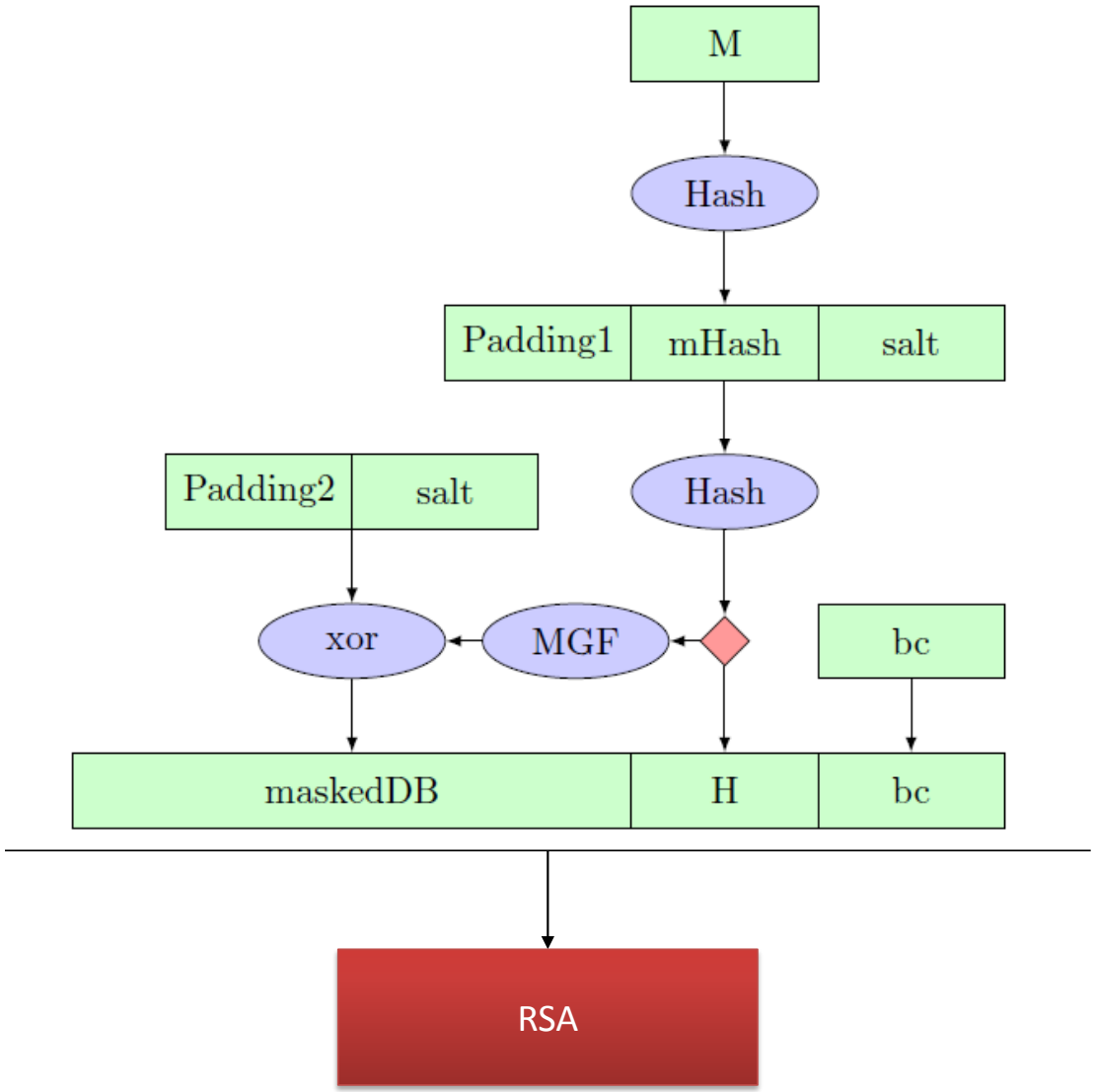
Digitálne podpisy

- autentickosť, integrita, nepopierateľnosť pôvodu . . .
- ekvivalent „vlastnoručného“ podpisu v elektronickom prostredí
- asymetrická schéma:
  - súkromný kľúč – podpisovanie $\Rightarrow$ len vlastník vie podpísať
  - verejný kľúč – overovanie $\Rightarrow$ ktokoľvek vie overiť
- podpis musí závisieť na podpisovanom dokumente/správe
- podpisuje sa odtlačok dokumentu ($H(m)$):
  - výkonové dôvody
  - bezpečnostné dôvody (falšovanie náhodnej správy)

RSA podpisy

- „prehodíme“ transformácie z klasického RSA:
  - podpisovanie: $s = H(m)^d \bmod n$
  - overovanie podpisu: platí $s^e \bmod n = H(m)$?
- (len tu!) bijektívnosť je zriedkavá vlastnosť asym. systémov
- v štandardoch zvyčajne so znáhodneným zarovnaním, napr. RSA-PSS (Probabilistic Signature Scheme) v RFC 3447, resp. v PKCS #1 v2.1
  - PKCS #1 v1.5 stále široko používané

RSA-PSS (PKCS #1 v2.1)



DSA

- DSA – Digital Signature Algorithm
- súčasť štandardu DSS (aktuálne FIPS 186-3)
- bezpečnosť súvisí s problémom diskrétneho logaritmu
- parametre:
  - p, q – prvočísla, $q \mid (p-1)$
    (kde bitové dĺžky prvočísel sú $(1024, 160)$, $(2048, 224)$, $(2048, 256)$ a $(3072, 256)$)
  - g – vypočítame $g = h^{(p-1)/q} \bmod p$, kde $h \in_R \{2, 3, \ldots, p-2\}$
  - súkromný (podpisový) kľúč: $x \in_R Z_q^*$
  - verejný (overovací) kľúč: $y = g^x \bmod p$ a parametre p, q, g

DSA (pokračovanie)

- podpisovanie:
  1. $k \in_R \{1, \ldots q-1\}$
  2. $r = (g^k \bmod p) \bmod q$
  3. $s = k^{-1}(H(m) + xr) \bmod q$
- overovanie:
  1. $u_1 = H(m) \cdot s^{-1} \bmod q$
  2. $u_2 = r \cdot s^{-1} \bmod q$
  3. platí $(g^{u_1} \cdot y^{u_2} \bmod p) \bmod q = r$?

Porovnanie

	Hašovacie funkcie	Autentizačné kódy	Digitálne podpisy
Integrita	áno	áno	áno
Autentickosť	nie	áno	áno
Nepopierateľnosť autorstva	nie	nie	áno
Kľúče	žiadne	symetrické	asymetrický pár kľúčov
Efektívnosť	rýchle	rýchle	pomalé
Typická aplikácia	kontrola integrity statických dát	autentickosť jednotlivých paketov v sieti	autentickosť dokumentov

PKCS

- Public-Key Cryptography Standards
- implementačné štandardy, napr.:

PKCS #1: RSA Cryptography Standard
PKCS #5: Password-Based Cryptography Standard
PKCS #7: Cryptographic Message Syntax Standard
PKCS #10: Certification Request Syntax Standard
PKCS #11: Cryptographic Token Interface Standard
PKCS #12: Personal Information Exchange Syntax Standard

Bezpečnosť – dĺžka kľúča

- generický útok – úplné preberanie množiny K
- dostatočná veľkosť $|K|$ (dĺžka kľúča)
  - nutná, ale
  - nie postačujúca podmienka bezpečnosti
- aká dĺžka kľúča je dostatočná?
  - *Deep Crack – DES, \$200k,* 1 *kľúč* $\sim$ 22 *hodín, 1998 . . . (2007)* 1 *DES kľúč* $\sim$ 3 *dni, \$12 000*
  - *odhady:* 80 *bitový kľúč* $\sim$ 1 *rok, \$8M, 2006*
  - *odhady:* 80 *bitový kľúč* $\sim$ 1 *mesiac, \$33M, 2010*

  - ako dlho má šifra (ŠT) odolať („cena“ dát)?
  - aký progres v kryptoanalýze predpokladáme?
  - aký bude progres v technológii (Mooreov zákon)?
  - aký silný (ekonomicky) je/bude útočník?

Bezpečnosť – dĺžka kľúča 2

- rôzne doporučenia, rôzne metodiky výpočtu
- NSA Suite B Cryptography (2010)
- ECRYPT Report (2011)
- NIST Recommendations (2011)
- . . . a ďalšie (`www.keylength.com`)

NSA Suite B Cryptography (2010)

- pre komerčne dodávané systémy
- Suite A – neverejné algoritmy, neznáme dĺžky kľúčov
- algoritmy:
  - šifrovanie: AES (FIPS 197) v GCM móde
  - podpisy: ECDSA (FIPS 186-3)
  - hašovanie: SHA-2 (FIPS 180-3)
  - výmena kľúčov: ECDH

	sym. šifr.	EC (GF(p))	hašovanie
Secret	128/256	256	256
Top Secret	256	384	384

ECRYPT II Report (2011)

- ECRYPT – európska sieť excelencie v kryptológii
- rôzne aktivity
- pravidelný report o doporučených dĺžkach kľúčov
- level 1-8
  - level 4 – najmenšia všeobecná ochrana (do 2015)
    „veľmi krátkodobá ochrana voči agentúram“
  - level 7 – dlhodobá ochrana (cca. 30 rokov)
  - level 8 – „prevídateľná budúcnosť“

	sym. šifr.	RSA	EC	hašovanie
level 4	80	1 248	160	160
level 7	128	3 248	256	256
level 8	256	15 424	512	512

Útok prehľadávaním priestoru kľúčov

Čas útoku	Individuálny útočník 1 procesor	Stredne veľká firma 500 procesorov	Príjmy SR za 1 rok (53,8 mil. procesorov)
1 minúta	33,7	42,6	59,3
1 hodina	39,6	48,5	65,2
1 deň	44,1	53,1	69,8
30 dní	49,1	58,0	74,7
1 rok	52,7	61,6	78,3
100 rokov	59,3	68,3	85,0

- Ilustračný príklad pre konkrétny procesor (i7-2600)

Generické útoky

Konštrukcia	Generický útok (k dĺžka kľúča, n veľkosť odtlačku/výstupu)
Symetrická šifra	Prehľadávanie priestoru všetkých kľúčov $\sim 2^k$
Hašovacia funkcia	Hľadanie kolízií: narodeninový útok $\sim 2^{n/2}$ Hľadanie vzoru: prehľadanie a vyskúšanie vzorov $\sim 2^n$
MAC	Prehľadávanie priestoru všetkých kľúčov $\sim 2^k$, resp. uhádnutie korektného autentizačného kódu k správe $\sim 2^n$.
Asymetrická šifra	Riešenie konkrétneho ťažkého problému (faktorizácia, výpočet diskrétneho logaritmu a pod.)
Podpisová schéma	Riešenie konkrétneho ťažkého problému, resp. útok na hašovaciu funkciu.

Ekvivalentné dĺžky kľúčov

Ochrana	Symetrický kľúč	Výstup hašovacej funkcie	RSA modul	Eliptická krivka
~ 4 roky	80	160	1248	160
~ 20 rokov	112	224	2432	224
~ 30 rokov	128	256	3248	256
	256	512	15424	512

Podľa správy Podľa správy ECRYPT II (2012)

Protokoly

- rôzne typy protokolov (účel):
  - výmena (distribúcia/dohoda) kľúča
  - autentizácia subjektu
  - slepé podpisy, voľby, peniaze, ...
- bezpečnosť závisí na schopnosti útočníka:
  - odpočúvať / modifikovať ľubovoľné správy
  - legitímny subjekt prostredia / mimo
- (zvyčajne) chceme protokol odolný voči najsilnejšiemu útočníkovi
- Najznámejšie protokoly: SSL/TLS, IPSec, SSH a pod.
  - Viaceré varianty – algoritmy, spôsoby autentizácie
- Prostriedky autentizácie účastníka protokolu:
  - Zdieľaná tajná informácia (heslo/kľúč)
  - Znalosť súkromného kľúča k verejnému kľúču uvedenom v certifikáte

Diffieho-Hellmanov protokol

- protokol na dohodnutie kľúča
- súvisí s problémom diskrétneho logaritmu (DH problém)

$$A \to B: \quad X = g^x$$
$$B \to A: \quad Y = g^y$$
$$\text{výsledný kľúč: } K = X^y = Y^x = g^{xy}$$

- man-in-the-middle útok (len pre aktívneho útočníka)
- schopnosť overiť autentickosť dát (napr. dig. podpismi) znemožní MIM útok

SSL/TLS 1

- SSL – Secure Socet Layer (pôvodne Netscape)
- TLS – Transport Layer Security (TLS 1.0 $\sim$ SSL v3.1)
- protokol nad TCP/IP, zabezpečuje intergitu a dôvernosť
- v podstate ľubovoľný protokol nad SSL (FTP, SMTP)
- najčastejšie: HTTP/SSL (https)

SSL/TLS 2

- SSL protokoly:
  - Record Protocol – spodná vrstva (šifrovanie, MAC, kompresia[1])
  - Handshake Protocol – autentizácia (jednostranná – server, alebo vzájomná – aj klient) dohoda o kryptografických algoritmoch, dohoda o šifrovacom kľúči a MAC kľúči
  - Alert Protocol – oznamovanie chybových hlášok (napr. `certificate_expired`)
  - Change Cipher Spec Protocol – „prepnutie“ aloritmov
- kryptografia v SSL, napr.:

_KeyExchange_WITH_Cipher_MAC
SSL_DHE_DSS_WITH_DES_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA

[1] ktorú nikto nepoužíva

IPSec

- bezpečnostný „doplnok“ k IP vrstve
- oblasti pôsobnosti: dôvernosť, autentickosť, správa kľúčov
- výhody „nízkoúrovňového“ protokolu:
  - zabezpečená celá komunikácia nad IP
  - transparentné pre aplikácie
  - možnosť vytvoriť VPN
  - integrácia do sieťových zariadení (smerovače a pod.)
- nevýhody:
  - SW implementácia (v operačnom systéme) zaťažuje server
  - identita zariadania, nie používateľa/aplikácie

IPSec (2)

- základné protokoly – AH, EPS
- AH (Authentication Header) – len autentickosť/integrita
- ESP (Encapsulating Security Payload) – šifrovanie a voliteľne autentickosť/integrita
- transportný mód (spracúvajú sa vybrané časti IP paketu) a tunelovací mód (zabalenie celého IP paketu do nového) protokolov
- algoritmy: HMAC-MD5/SHA-1 (96), 3DES, Blowfish, ...
- správa kľúčov: manuálna, automatizovaná (ISAKMP/Oakley)

Kryptológia v kontexte

- aká dôveryhodná je implementácia?
- akým spôsobom sú spravované kľúče:
  - generovanie?
  - distribúcia?
  - backup?
  - riadenie prístupu?
  - ničenie?
- postranné kanály (čas, spotreba zdrojov, hyperthreading, chybové hlášky, . . .)?

. . . je ľahké urobiť chybu.

Kryptológia v kontexte (2)

- kryptografia je obvykle použitá v niečom „väčšom“
- operačný systém, čipové karty, databázový systém, mail, webová aplikácia, e-commerce, sieťové protokoly, . . .

ISO/IEC 27002

- Code of practice for information security management
- 12.3 Cryptographic controls

  12.3.1 Policy on the use of cryptographic controls
  12.3.2 Key management

Key management – hlbšie

1. Generating keys for different cryptographic systems
2. Generating and obtaining public key certificates
3. Distributing keys to intended users
4. Storing keys and how to obtain access to keys
5. Changing or updating keys
6. Dealing with compromised keys
7. Revoking and deactivating keys
8. Recovering keys that are lost or corrupted as part of BCM
9. Archiving keys
10. Destroying keys
11. Logging and auditing key management activities

Common Criteria

- ISO/IEC 15408 – Evaluation criteria for IT security
- použitie: jednotný jazyk pre
  - používateľov: špecifikujú požiadavky
  - výrobcov: popíšu vlastnosti produktov
  - nezávislé testovanie: vyhodnotí
- časti:
  - Part 1: Introduction and general model
  - Part 2: Security functional requirements
  - Part 3: Security assurance requirements
- EAL: evaluation assurance level (EAL1 - EAL7)
- flexibilné – hodnotenie veľkých systémov (napr. operačné systémy), aj malých (napr. čipové karty, aplikácie)

Common Criteria (2)

- CC a kryptografické algoritmy:

  The subject of criteria for the assessment of the inherent qualities of cryptographic algorithms is not covered in ISO/IEC 15408. Should independent assessment of mathematical properties of cryptography embedded in a TOE be required, the evaluation scheme under which ISO/IEC 15408 is applied must make provision for such assessments.

Common Criteria (3)

- požiadavky na bezpečnostné funkcie
- trieda FCS (Cryptographic support):
  - Správa kryptografických kľúčov
    - generovanie kľúčov
    - distribúcia kľúčov
    - prístup ku kľúčom
    - deštrukcia kľúčov
  - Kryptografická činnosť
- pre jednotlivé komponenty sa vyžaduje súlad s definovanými algoritmami, metódami, dĺžkami kľúčov, štandardmi.

FIPS 140-2

- Security Requirements for Cryptographic Modules
- požiadavky na kryptografické moduly (SW aj HW)
- 4 úrovne bezpečnosti:

  Level 1: najnižšia úroveň
  Level 2: + detekcia fyzickej manipulácie, ...
  Level 3: + odolnosť voči fyzickej manipulácii, ...
  Level 4: + ...
- Level 1,2 – najčastejšie úrovne
  - (za rok 2011) 64 certifikátov L1, 92 L2, 26 L3, 3 L4
- FIPS 140-2 $\rightsquigarrow$ ISO/IEC 19790:2006 Security requirements for cryptographic modules
- (žiadna) certifikácia negarantuje bezpečnosť

FIPS 140-2 (oblasti)

1. Špecifikácia (dokumentácia) modulu
2. Časti modulu a rozhrania (vrátane segregácia)
3. Role, služby a autentizácia
4. Konečnostavový model
5. Fyzická bezpečnosť
6. Operačné prostredie (operačný systém)
7. EMI/EMC
8. Správa kryptografických kľúčov
9. Samotestovanie
10. Záruky pre kvalitu návrhu a implementácie
11. Ochrana pred útokmi (napr. TEMPEST)

Záver

- bez kryptológie je ťažké (nemožné?) dosiahnuť bezpečnosť IS
- niekedy sú použité kryptografické konštrukcie zlé
- niekedy sú kryptografické konštrukcie použité zle