Kryptológia - úvod

Úvod do informačnej bezpečnosti

Michal Rjaško

LS 2022/2023

rjasko@dcs.fmph.uniba.sk

Plán kurzu

http://www.dcs.fmph.uniba.sk/~rjasko/uib2022.html

- Manažment informačnej bezpečnosti
  1. Úvod, základné pojmy (Daniel Olejár)
  2. Bezpečnosť v organizácii (Daniel Olejár)
  3. Legislatíva v IB a normy (Daniel Olejár)
- Kryptológia
  4. Úvod, základné kryptografické prvky – šifrovanie (Michal Rjaško)
  5. Hašovacie funkcie, Autentizačné kódy, Digitálne podpisy (Michal Rjaško)
  6. Dĺžky kľúčov, štandardy, Kryptológia v kontexte - príklady zraniteľností a pod. (Michal Rjaško)
  7. PKI a digitálne podpisy (Daniel Olejár)
  8. Kryptografické protokoly, heslá, identifikácia a autentizácia (Michal Rjaško)
- Bezpečnosť software
  9. Základné zraniteľnosti (Michal Rjaško)
  10. Bezpečnosť OS (Jaroslav Janáček / Michal Rjaško)
  11. Malware, vírusy a iná háveď (Peter Košinár – ESET)
  12. Analýza škodlivého kódu, penetračné testovanie, sieťová bezpečnosť (CSIRT.sk?)

Krypto: obsah

- Úvod
- Základné kryptografické prvky
  - ✓Symetrické šifry
  - ✓Asymetrické šifry
  - ✓Hašovacie funkcie
  - ✓Autentizačné kódy
  - ✓Digitálne podpisy
- Dĺžky kľúčov
- Kryptológia v kontexte
- Štandardy

Informačná bezpečnosť a kryptológia

Čo krypto dokáže a aké sú jeho limity

- Ukradnutý / zabudnutý notebook s dôvernými údajmi
  - Disk bol hardvérovo šifrovaný
  - Cold Boot útok
  - Evil maid útok
- Šifrované USB kľúče (certifikované FIPS 140-2)
  - https://www.schneier.com/blog/archives/2010/01/fips_140-2_leve.html
- WPS (WiFi Protected setup)
- Náhodnosť kryptografických kľúčov
  - Debian Linux 2008
  - DUAL_EC_DRBG
- Timing útoky / padding oracle útoky na SSL/TLS
  - Drownattack, Heartbleed, Beast, CRIME, Lucky 13, Poodle útoky na SSL a TLS protokoly
- Meltdown / spectre, …

Kryptológia

- dôležitá súčasť informačnej bezpečnosti
- informačná bezpečnosť - kryptológia = ???
- kryptológia = kryptografia + kryptoanalýza
- Požiadavky: dôvernosť, integrita, autentickosť, nepopretie autorstva/doručenia a pod.
- Kryptografia nie je odpoveďou na všetky bezpečnostné požiadavky
  - Dostupnosť (redundancia), bezpečný softvér a pod.
- Kvalitná kryptografia je nutná, ale nie postačujúca
  - Nepoužiteľná alebo zraniteľná bez ďalších opatrení: správa kľúčov, riadenie prístupu, kvalita implementácie a pod.

Kryptológia

- Matematika
- Detaily sú podstatné
- Implementácia a použitie

- 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 256-bit TLS encryption. This ensures that your payment information is secure and safe.

Ako sú uložené heslá používateľov v databáze?
Ukladajú sa na serveri čísla kariet? Ak áno, ako?
Kto má prístup k platobným informáciám?

Kryptografické konštrukcie

- Obvykle používané:
  - Šifrovanie, digitálne podpisy, autentizačné kódy, protokoly pre autentizáciu a distribúciu kľúča
- Menej obvyklé:
  - Zdieľanie tajomstva, onion routing, volebné protokoly, elektronické peniaze a pod.
- „Exotické“:
  - plne homomorfné šifrovanie, obfuskácia a pod.

- Očakávame: efektívnosť, korektnosť a bezpečnosť

Symetrické šifrovanie

- klasický cieľ kryptografie – dôverný prenos dát
- komunikujúce subjekty zdieľajú tajný kľúč
- kľúč je rovnaký pre odosielateľa aj príjemcu ⇒ symetrické šifrovanie

Označenie:

- otvorený text (plain text) = pôvodná správa, text, dokument, dáta
- šifrový text (cipher text) = zašifrovaný text, výstupné dáta šifrovacieho algoritmu

Symetrické šifrovanie

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

- čo očakávame od E, D?

Symetrické šifrovanie

Očakávame:

- efektívnosť: E aj D sú efektívne (polynomiálny algoritmus)
- korektnosť:

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

- 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?

- 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 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
  - tzv. sémantická bezpečnosť

Vernamova šifra (one-time pad)

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

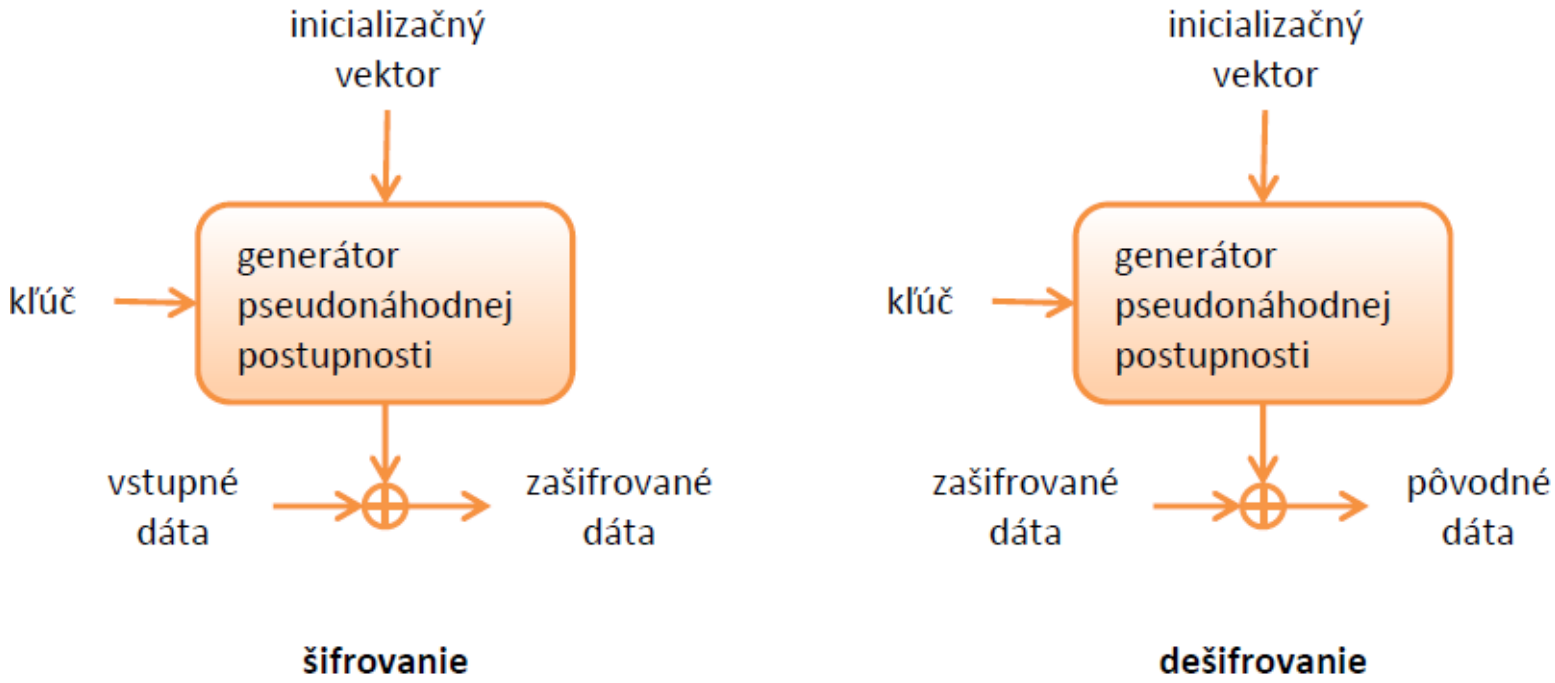
Vernamova šifra (one-time pad)

Nevýhody:

- kľúč rovnako dlhý ako otvorený text
- neposkytuje integritu
  - útočník nemôže získať otvorený text, ale môže ho cestou pozmeniť
- „jednorázový“ 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)



Prúdové šifry

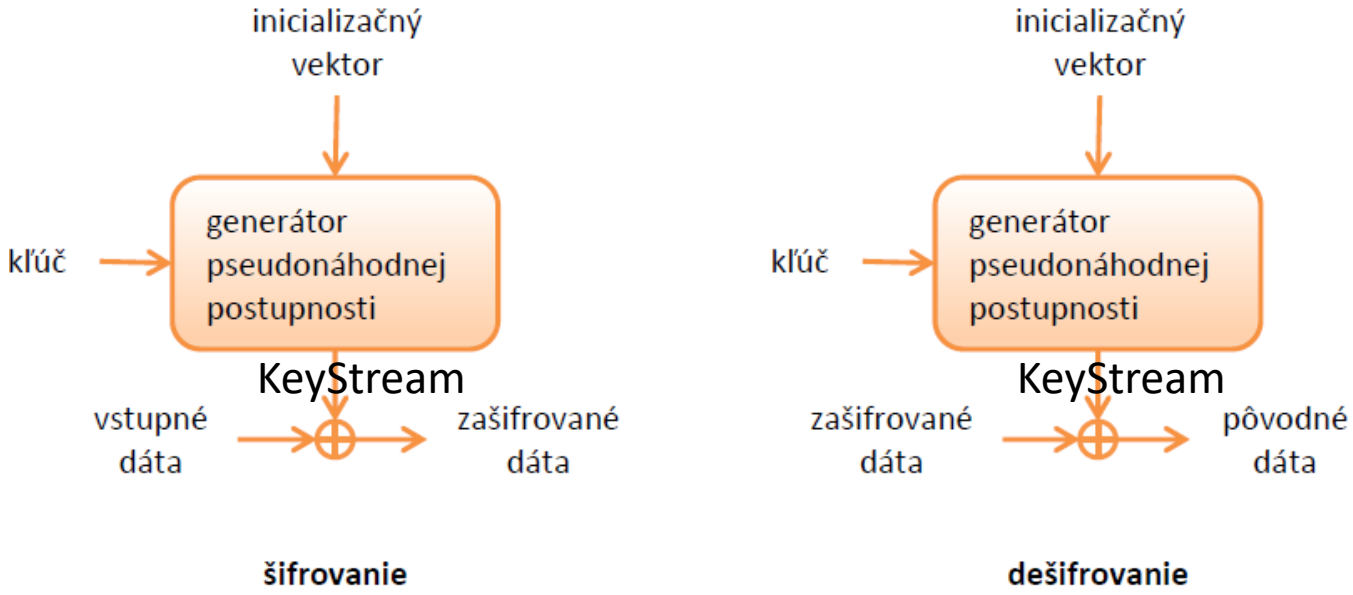
Zvyčajné výhody (oproti blokovým šifrám):

- 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
  - (KeyStream $\oplus$ M_1) $\oplus$(KeyStrem $\oplus M_2$) = $M_1 \oplus M_2$
- 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: $\{0,1\}^n \rightarrow \{0,1\}^n$
- E_k a D_k sú inverzné bijekcie
- **Mód** – spôsob šifrovania dlhých OT, napr:
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)
  - CTR (Counter)
- Najpoužívanejšie blokové šifry: AES, (3)DES
- Spôsob konštrukcie blokových šifier: iterácia viacerých kôl

Blokové šifry

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 – AES, 3DES

- Oba algoritmy schválené NIST
- „Novšie“ procesory s HW podporou pre AES
  - podstatne vyšší výkon ako pri SW implementácii
- Rastúca dĺžka kľúča $\Rightarrow$ mierne klesá výkon AES (viac kôl)
- 3DES – sekvenčné zreťazenie 3 DES transformácií
  - 3 nezávislé kľúče – 168 bitov dlhý kľúč (3 x 56)
  - Efektívna dĺžka kľúča (len) 112 bitov

	dĺžka bloku	dĺžka kľúča	počet kôl
AES	128	128, 192, 256	10, 12, 14
3DES	64	168 (112)	3 x 16

Blokové šifry - Štandardy

- 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 bitov pri 112 bitovej bezpečnosti
    - Meet in the middle útok

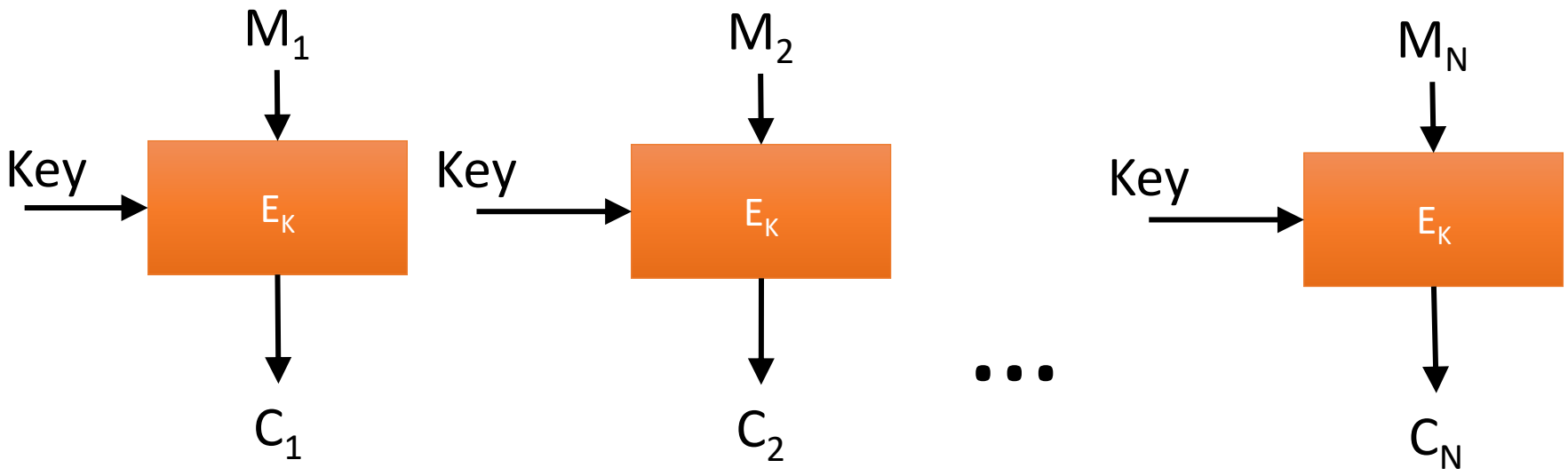
Operačné módy blokových šifier

- Flexibilné použitie blokových šifier
- Rôzne módy pre rôzne bezpečnostné požiadavky (módy schválené NIST):
  - dôvernosť (celkovo 5 módov: ECB, CBC, OFB, CFB, CTR)
  - autentickosť (CMAC)
  - autentizované šifrovanie (CCM) a autentizované šifrovanie s vysokou priepustnosťou (GCM)
  - dôvernosť pre blokové úložiská dát, napr. disky (XTS)
  - dôvernosť a integrita kľúčov (KW, KWP, TKW)

  ... a ďalšie
- Rôzne módy majú rôzne vlastnosti a predpoklady

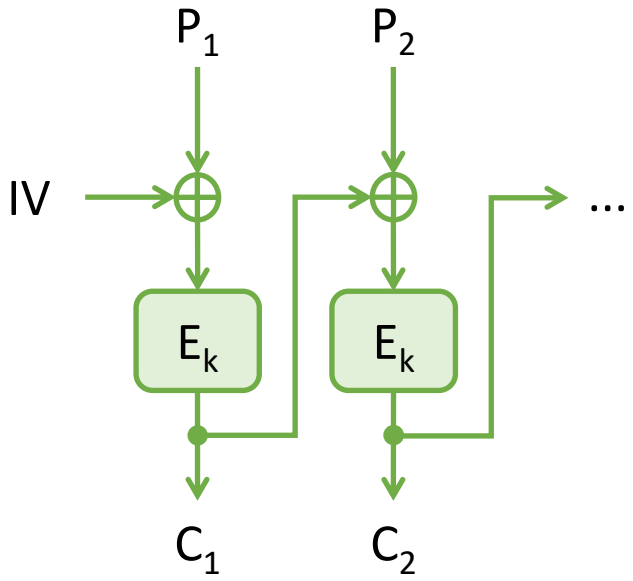
Operačné módy blokových šifier

- ECB (Electronic Code Book)

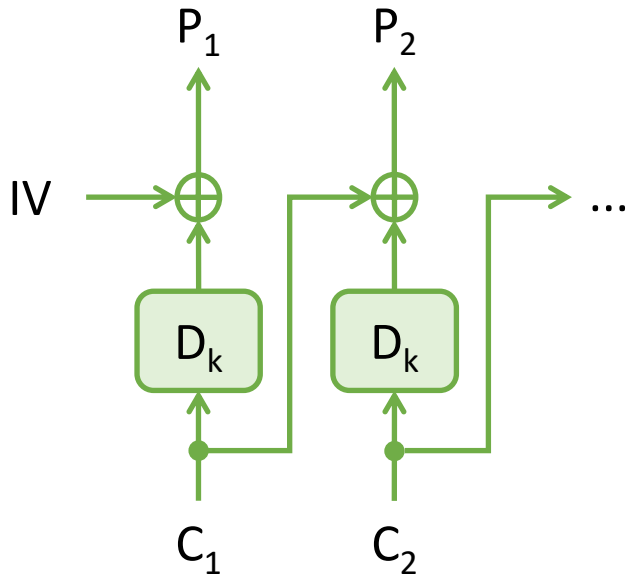


Operačné módy blokových šifier

- CBC (Cipher Block Chaining)



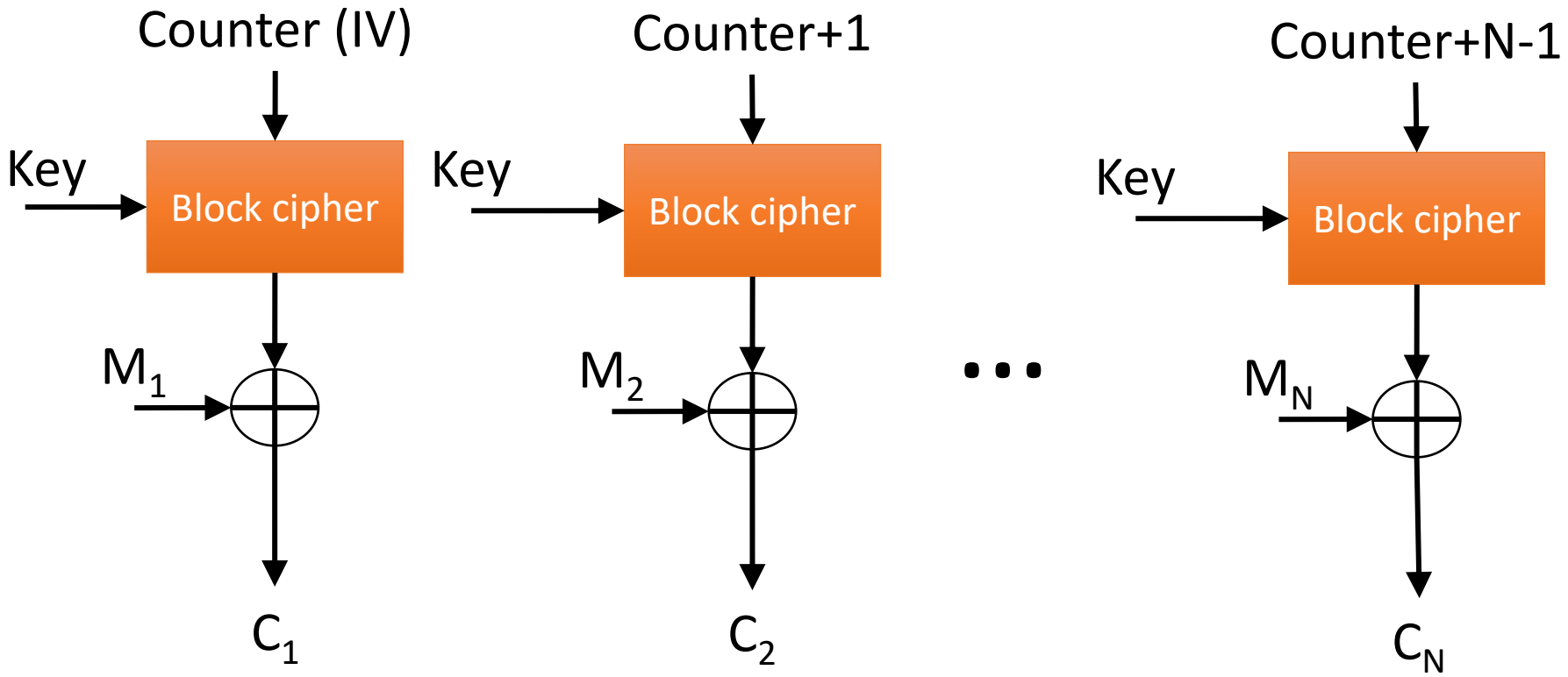
šifrovanie v CBC móde



dešifrovanie v CBC móde

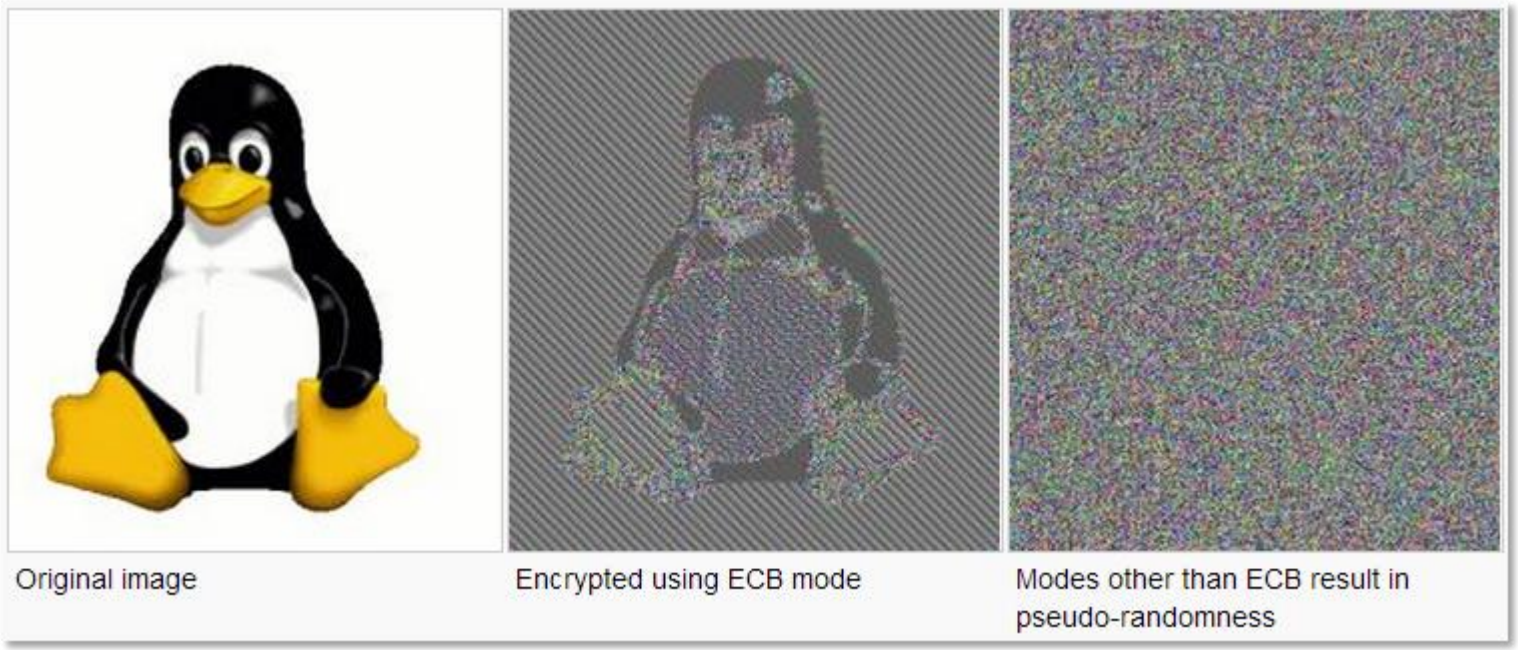
Operačné módy blokových šifier

- CTR (Counter)



Operačné módy blokových šifier

- ECB vs. CBC



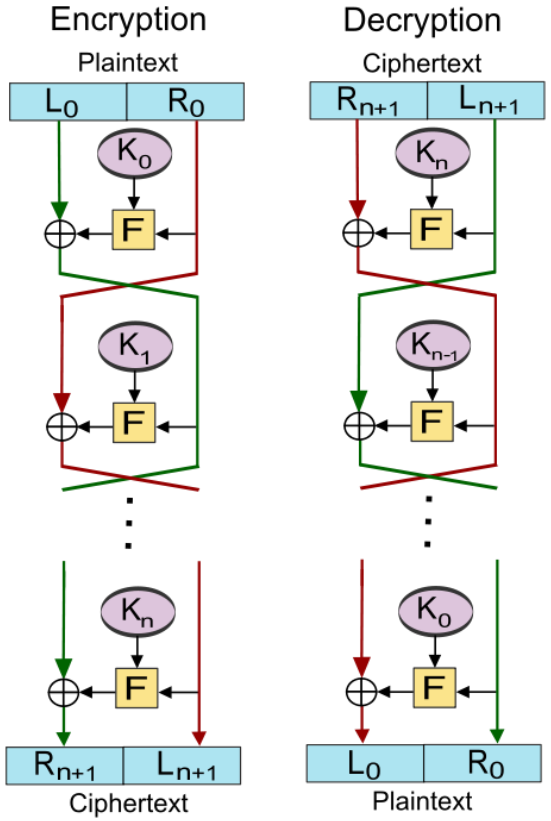
Original image

Encrypted using ECB mode

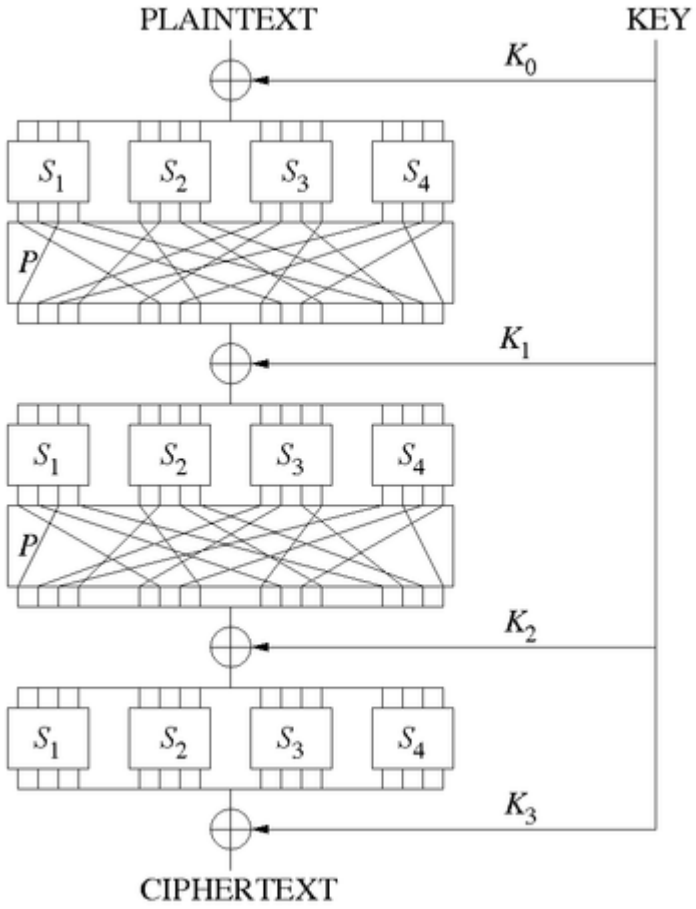
Modes other than ECB result in pseudo-randomness

Blokové šifry – konštrukcia

- Konštrukcia blokových šifier - iterácia kolových funkcií



Feistelovská sieť
napr. DES



Substitučno – permutačná sieť
napr. AES

Blokové šifry – poznámky

- Na dĺžke (bloku) záleží
  - Krátky blok uľahčuje kryptoanalýzu (hľadanie diferencií a pod.)
  - Veľmi krátky blok: max. $(2^n)!$ permutácií
  - Dlhý blok zvyšuje nároky na HW a implementáciu

- Kľúč štandardne volený ako náhodný prvok z $\{0,1\}^k$
- Úplné preberanie priestoru kľúčov $\sim 2^k$ operácií
  - V priemernom prípade (očakávaná zložitosť) $\sim 2^{k-1}$
  - Platí len pre náhodne (uniformne) volené kľúče

Blokové šifry – poznámky

- Možné problémy v skutočnosti:
  - Kľúč odvodený z hesla, nekvalitný generátor a pod.
  - Preberanie kľúčov od najpravdepodobnejších

- Čokoľvek lepšie ako úplné preberanie je (teoreticky) úspešný útok – hoci môže byť stále nepraktický
  - Priveľká zložitosť, napr. 2^{118}
  - Nerealistické predpoklady, napr. 2^{90} zvolených OT šifrovaných rovnakým kľúčom

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
  - vypočítať zvolenú funkciu nad otvoreným textom
  - **rozlíšiť, ktorému z dvoch (útočníkom vybraných) OT zodpovedá daný ŠT**
    - (IND-CPA / IND-CCA)
    - IND-CPA bezpečnosť je ekvivalentná sémantickej bezpečnosti

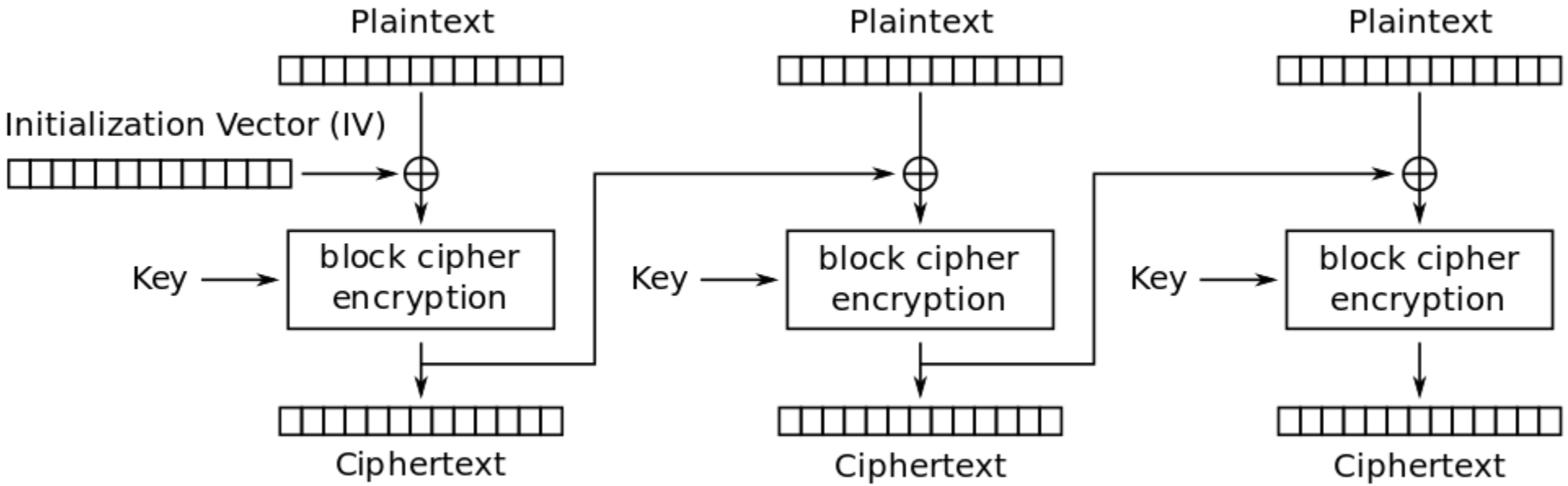
Symetrické šifry - výkon

	SW impl. [MB/s]	HW podpora AES-NI (encrypt) [MB/s]	HW podpora AES-NI (decrypt) [MB/s]
AES-128-CTR		4 125	4 121
AES-128-CBC	127	749	4 064
AES-192-CBC	106	623	3 509
AES-256-CBC	90	537	3 055
3DES-CBC	28		
RC4	891		

i7-2600, 3.40GHz, Ubuntu 12.04 LTS 64-bit, openssl 1.0.1, 8kB bloky

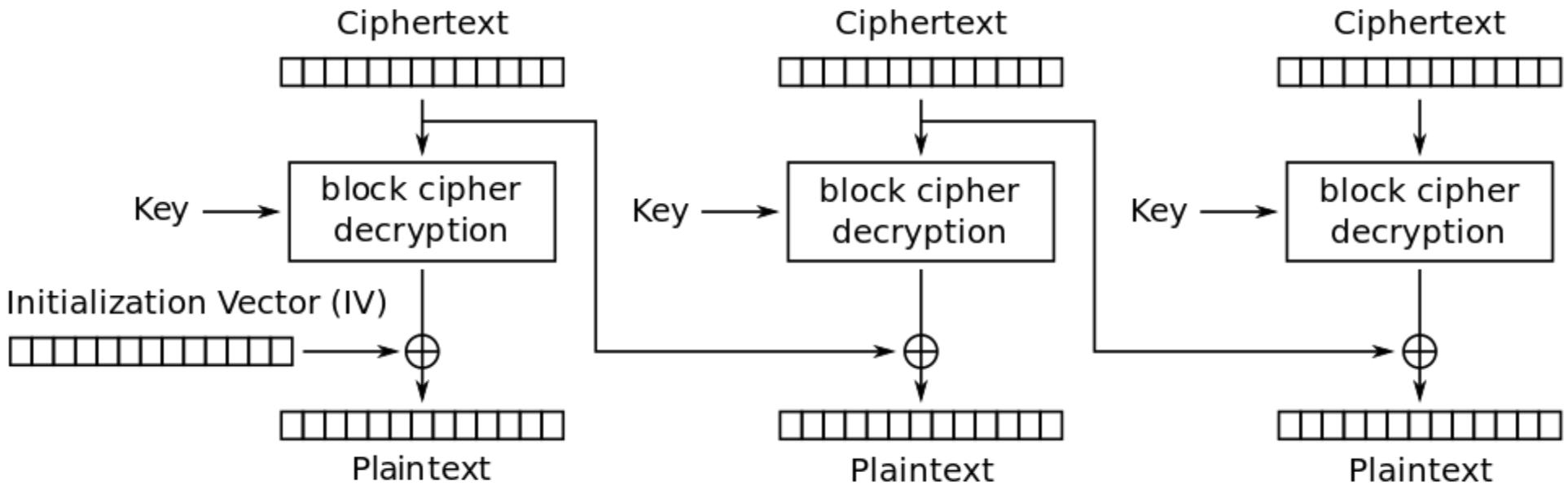
CBC šifrovanie / dešifrovanie

šifrovať môžeme iba sekvenčne

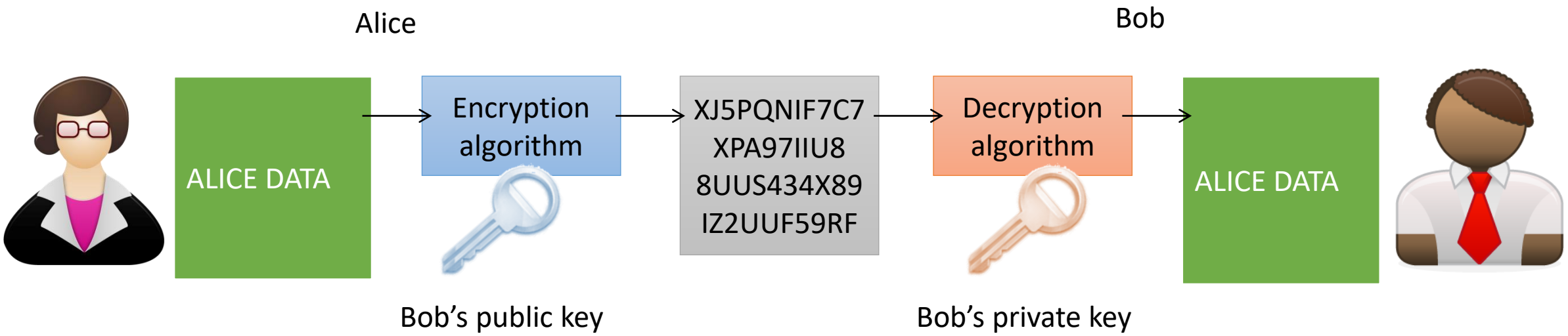


CBC šifrovanie / dešifrovanie

dešifrovať môžeme paralelne



Asymetrické šifrovanie



dvojica rôznych kľúčov:

- Verejný – šifrovanie ⇒ ktokoľvek vie šifrovať
- Súkromný – dešifrovanie ⇒ len vlastník vie dešifrovať

- šifrovanie: $E: P \times K_{\text{pub}} \rightarrow C$
- dešifrovanie: $D: C \times K_{\text{priv}} \rightarrow P$

Asymetrické šifrovanie

- Jednoduchšia správa kľúčov
  - Ako dôveryhodne distribuovať verejný kľúč (autentickosť)?
    - Osobne / protokolárne
    - Prostredníctvom dôveryhodnej tretej strany – PKI – samostatná prednáška
- Bezpečnosť:
  - CPA útok je vždy možný
  - Verejný kľúč ⇏ algoritmus na dešifrovanie
- Bezpečnosť sa opiera o zložitosť matematických problémov
  - faktorizácia, diskrétny logaritmus a iné
- Najznámejšie systémy: RSA, ElGamal

Problémy pre asymetrické konštrukcie

- Faktorizácia veľkých čísel
  - RSA problém, QR problém, počítanie druhých odmocnín a pod.
- Diskrétny logaritmus v rôznych grupách
  - DLOG, rozhodovací a výpočtový DH problém, Gap DH problém, Twin DH problém a pod.
- Mriežky
  - SVP (Shortest vector problem), CVP (Closest vector problem), LWE (Learning with errors), NTRU a pod.
  - Zatiaľ odolné voči útokom kvatovými počítačmi
- Iné
  - SDP (Syndrome Decoding Problem) a pod.

Faktorizácia

- Úloha: vypočítať rozklad čísla n
  - zvyčajne $n = p \cdot \mathrm{q}$ (súčin dvoch veľkých prvočísel)
- Najlepší všeobecný algoritmus: General Number Field Sieve (GNFS)
- Zložitosť:
  - $\exp\left(\left(\left(\frac{64}{9}\right)^{\frac{1}{3}} + o(1)\right)(\ln n)^{\frac{1}{3}}(\ln \ln n)^{\frac{2}{3}}\right)$
- Efektívnejšie algoritmy existujú pre niektoré voľby p, q
- Napr. p, q blízko pri sebe, $(p - 1)$ alebo $(q - 1)$ bez veľkého prvočíselného faktora a pod

Diskrétny logaritmus

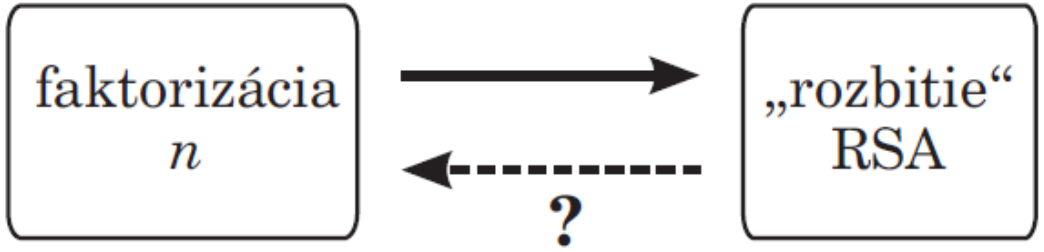
- Úloha: pre dané g a y vypočítať číslo x také, že $g^x = y$
  - *g* je generátor nejakej grupy *G* a „násobenie“ príslušná operácia
- Ľahký/ťažký problém v závislosti na *G*
- V kryptografii sú obvykle používané:
  - Multiplikatívna grupa (podgrupa) modulo prvočíslo, pričom operácia je násobenie v modulárnej aritmetike
  - Grupa bodov eliptickej krivky (operácia je sčítanie bodov krivky)
- Všeobecný generický algoritmus (pre ľubovoľnú konečnú cyklickú grupu): $O(n^{1/2})$ (n je počet prvkov grupy)
  - Baby-step giant-step, Pollard ρ
- Pre modulárnu aritmetiku: Number Field Sieve pre DLOG
  - Rovnaká zložitosť ako GNFS pre faktorizáciu
- Existujú efektívne kvantové algoritmy na faktorizáciu a počítanie DLOG (Shorov algoritmus)

RSA

- Verejný kľúč: (e, n)
  - $n = p \cdot q$, kde p, q sú veľké prvočísla
  - e je nesúdeliteľné s $(p-1)(q-1)$
  - najčastejšia voľba $e = 65537$
- Súkromný kľúč: d
  - pričom platí $e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$
  - pre urýchlenie súkromnej transfromácie sú súčasťou obvykle aj ďalšie parametre
- Šifrovanie ($E\colon Z_n \rightarrow Z_n$): $E(m) = m^e \bmod n$
  - pre vstup $m \in \{0,1, \ldots, n-1\}$
- Dešifrovanie ($D\colon Z_n \rightarrow Z_n$): $D(c) = c^d \bmod n$
- Šifrovanie rýchlejšie ako dešifrovanie

RSA

- (1978) Rivest, Shamir, Adleman
- Bezpečnosť súvisí s problémom faktorizácie
- Faktorizácia ⇔ znalosť d

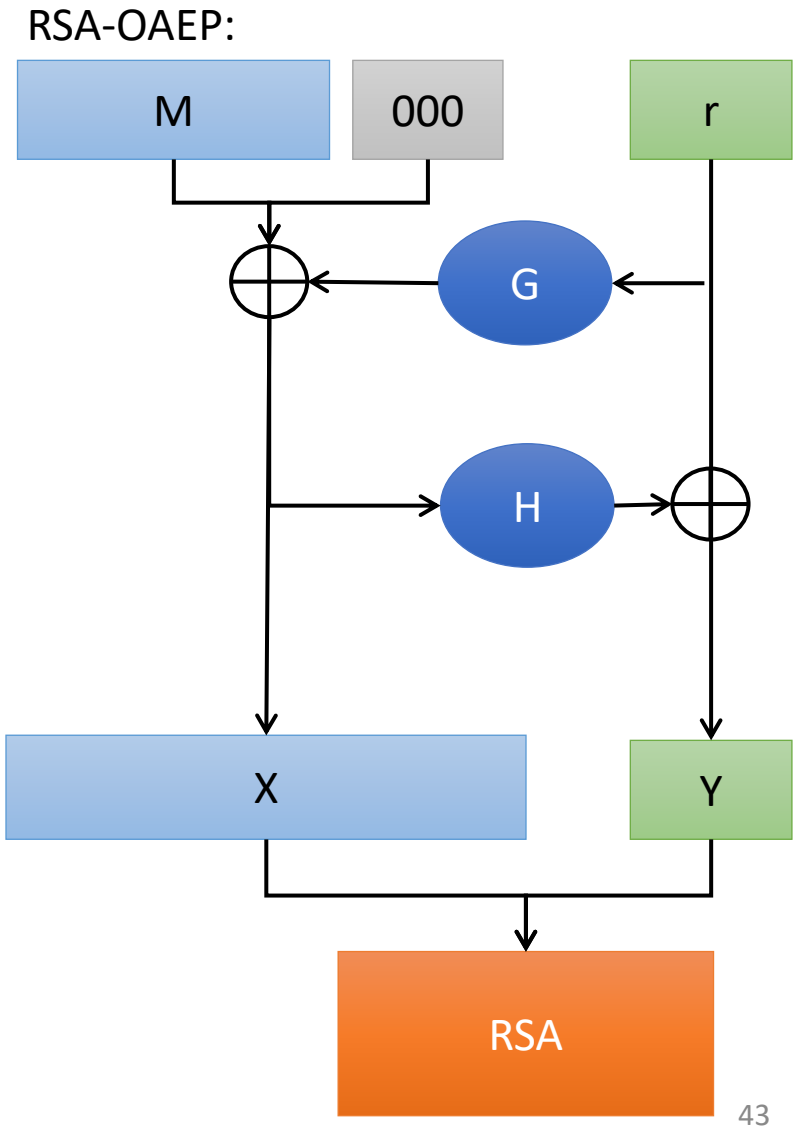


- RSA je bijekcia (nezvyčajné pri asymetrických systémoch)
- RSA je deterministické (to je zlé) ⇒ v praxi sa znáhodňuje
  - Každému OT zodpovedá len jeden ŠT (možnosť testovať OT)
  - Problém s malým priestorom správ („áno“/„nie“, výplata a pod.)

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 – implementačné poznámky

- Verejný exponent volený najčastejšie ako 65537
  - prvočíslo ⇒ vysoká pravdepodobnosť nesúdeliteľnosti s $(p-1)(q-1)$
  - ľahké testovanie nesúdeliteľnosti už pri generovaní p, q
  - krátke a binárny zápis s malým počtom 1 ⇒ rýchly výpočet E
  - príliš malý exponent (napr. $e = 3$) môže byť problematický z hľadiska bezpečnosti
- Súkromný exponent je jednoznačne určený
  - dĺžka d približne rovnaká ako dĺžka n
  - matematicky je možné najskôr zvoliť d a dopočítať e (krátke d je náchylné na útoky, pre $d < n^{0,292}$)
  - D výpočtovo náročnejšia ako E

    napr. RSA-2048: $E \sim 27496$ ops/s, $D \sim 857$ ops/s

RSA - výkon

	dešifrovanie [operácie/s]	šifrovanie [operácie/s]
RSA-1024	6 100	93 281
RSA-2048	857	27 496
RSA-4096	118	7 370

i7-2600, 3.40GHz, Ubuntu 12.04 LTS 64-bit, openssl 1.0.1, 8kB bloky

Hybridné šifrovanie

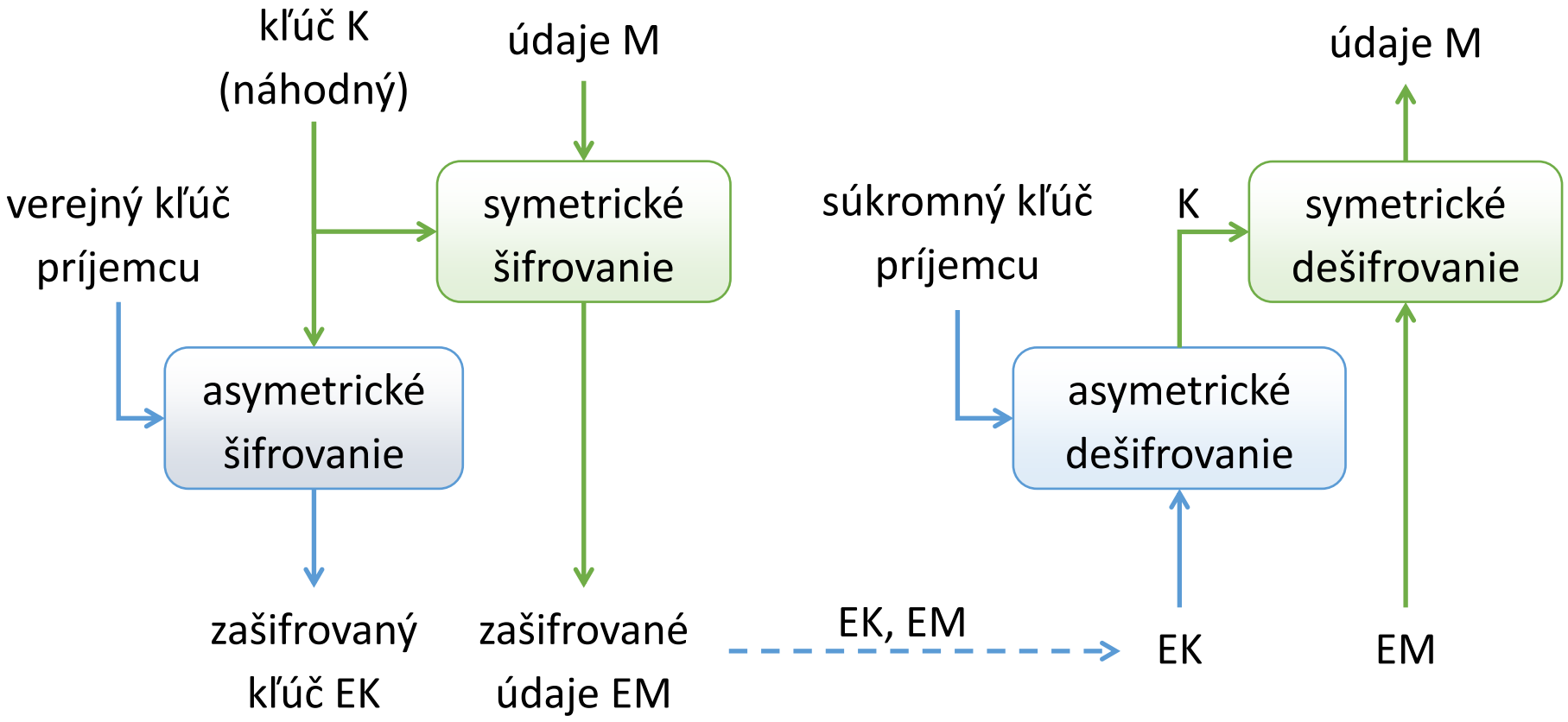
- Asymetrické šifry sú pomalé (v porovnaní so symetrickými)
- Čo s prenosom objemných dát?
- Riešenie:
  - šifrujme symetricky s **náhodným** kľúčom k
  - kľúč k zašifrujeme **asymetricky** pre adresáta

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

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

Hybridné šifrovanie

- Kombinácia asymetrického a symetrického šifrovania



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 $\mathrm{h}: \mathrm{X} \rightarrow Y$, zvyčajne $X = \{0,1\}^*, Y = \{0,1\}^n$
- „Odtlačok“ správy, dokumentu
- Žiadny kľúč
- Kontrola integrity (detekcia náhodnej/neúmyselnej zmeny dát), digitálne podpisy
- Rôznorodé použitie
  - digitálne podpisy, paddingové schémy, autentizačné kódy správ, PBKDF (odvodenie symetrických kľúčov z hesiel, uloženie hesiel)

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

Hašovacie funkcie

- Kryptografické vlastnosti:
  - jednosmernosť: pre dané y nájsť x: $h(x) = y$
  - odolnosť voči kolízám: nájsť $x \neq x'$: $h(x) = h(x')$
- Efektívnosť
- Najznámejšie hašovacie funkcie: MD5, SHA1, SHA-(224,256,384,512), SHA3
- SHA-3 najnovší štandard
  - Verejná súťaž vyhlásená NISTom (2007)
  - Keccak – víťazný algoritmus
  - Štandard od 5. augusta 2015
  - Predpokladaná koexistencia s rodinou SHA-2 funkcií

Hašovacie funkcie

Generické útoky:

- Hľadanie vzoru (útok na jednosmernosť – pre dané $y \in \{0,1\}^n$ hľadáme vzor)
  1. Zvolíme náhodne alebo systematicky $x \in \{0,1\}^m$
  2. Ak $h(x) = y$ tak sme našli vzor, inak postup opakujeme
  - Očakávaná zložitosť $\sim 2^n$
- Hľadanie kolízií – Narodeninový útok:
  - využíva tzv. „narodeninový“ paradox
  - Očakávaná zložitosť $\sim 2^{n/2}$

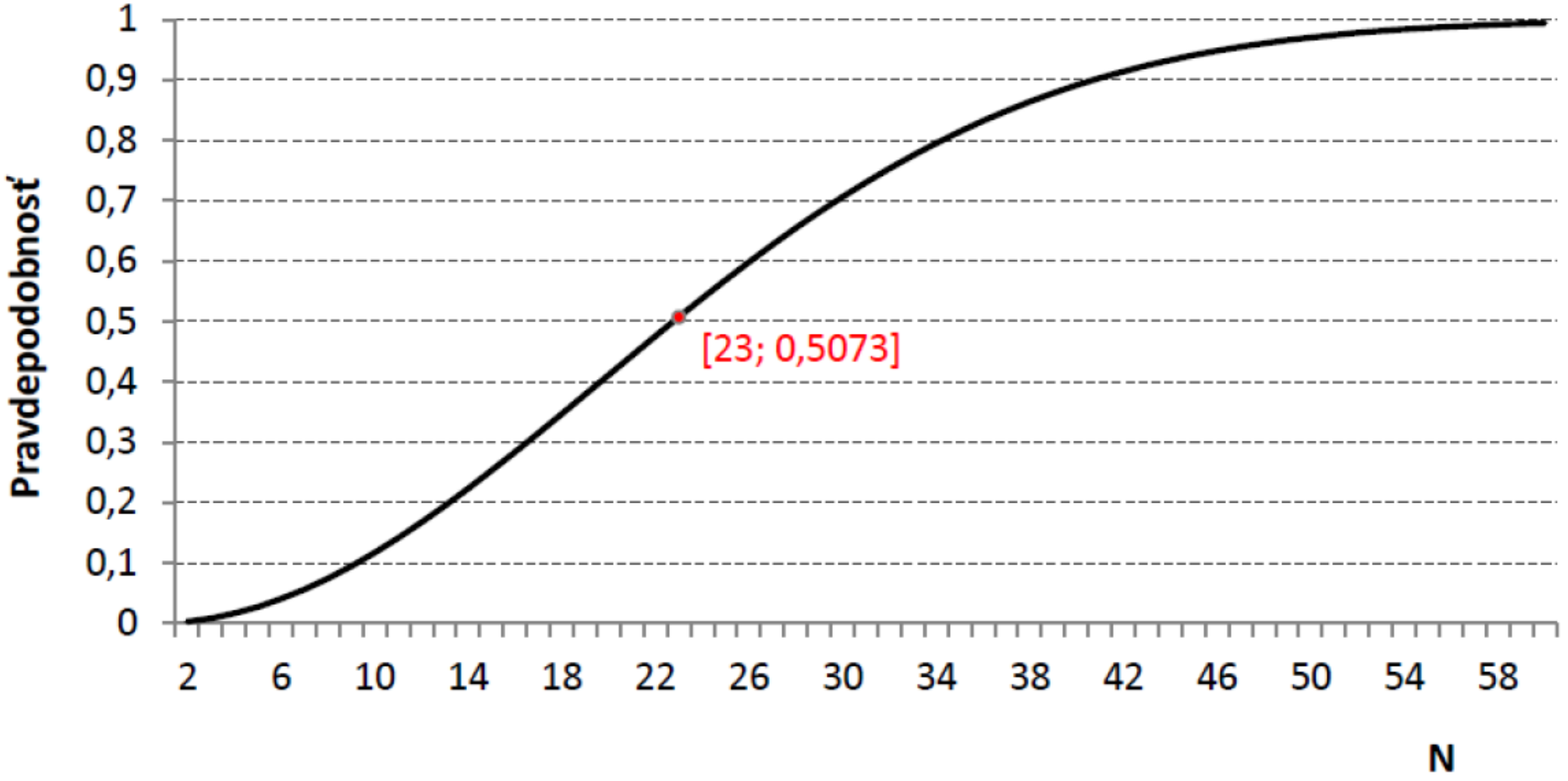
Narodeninový útok - idea

- Aká je pravdepodobnosť, že aspoň dve osoby v miestnosti majú narodeniny v ten istý deň?

$$\Pr_k = 1 - \frac{365 \cdot 364 \cdot \ldots \cdot (365 - k + 1)}{365^k}$$

- 23 osôb stačí na dosiahnutie pravdepodobnosti aspoň ½
- H.f. zobrazuje ľudí na dni v roku, |Y| = 365

Narodeninový paradox – pravdepodobnosť pre N ľudí



Narodeninový útok

- Priebeh:
  1. Zvolíme k rôznych hodnôt $x_1, x_2, \dots, x_k \in X$
  2. Vypočítame odtlačky $y_i = h(x_i)$ pre $i = 1, \dots, k$ pričom hľadáme zhodu (kolíziu)
- Predpokladajme $Y = \{0,1\}^n$
- Očakávané k ak chceme pravdepodobnosť úspechu aspoň:
  - 50% $k \approx 1{,}177 \cdot 2^{n/2}$
  - 90% $k \approx 2{,}146 \cdot 2^{n/2}$
  - 99% $k \approx 3{,}035 \cdot 2^{n/2}$
- Čím je h.f. „nerovnomernejšia“, tým je pravdepodbnosť úspechu vyššia
  - pri rovnakom počte k

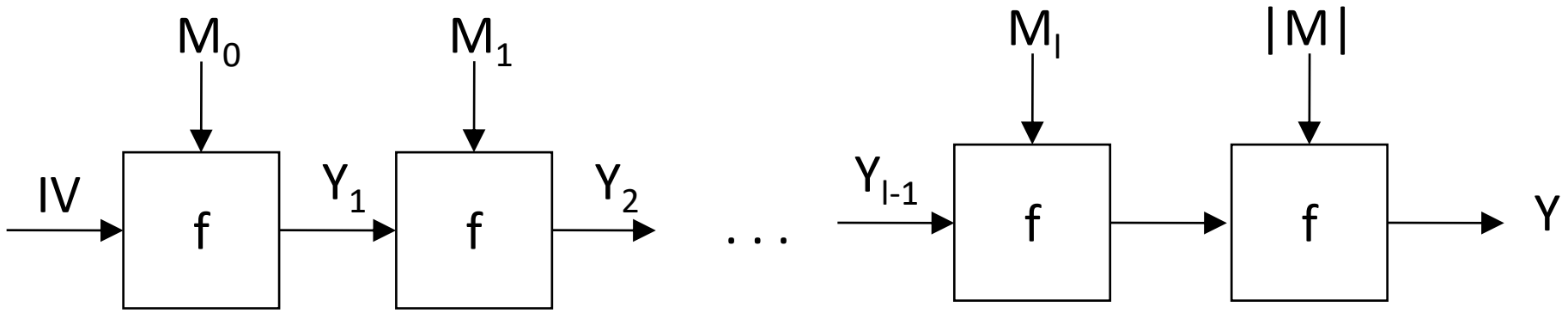
Výkonové porovnanie

	SW impl. [MB/s]	HW podpora AES-NI (encrypt) [MB/s]	HW podpora AES-NI (decrypt) [MB/s]
AES-128-CTR		4 125	4 121
AES-128-CBC	127	749	4 064
AES-192-CBC	106	623	3 509
AES-256-CBC	90	537	3 055
3DES-CBC	28		
RC4	891		
SHA-1	717		
SHA-256	215		
SHA-512	335		

i7-2600, 3.40GHz, Ubuntu 12.04 LTS 64-bit, openssl 1.0.1, 8kB bloky

Hašovacie funkcie - konštrukcie

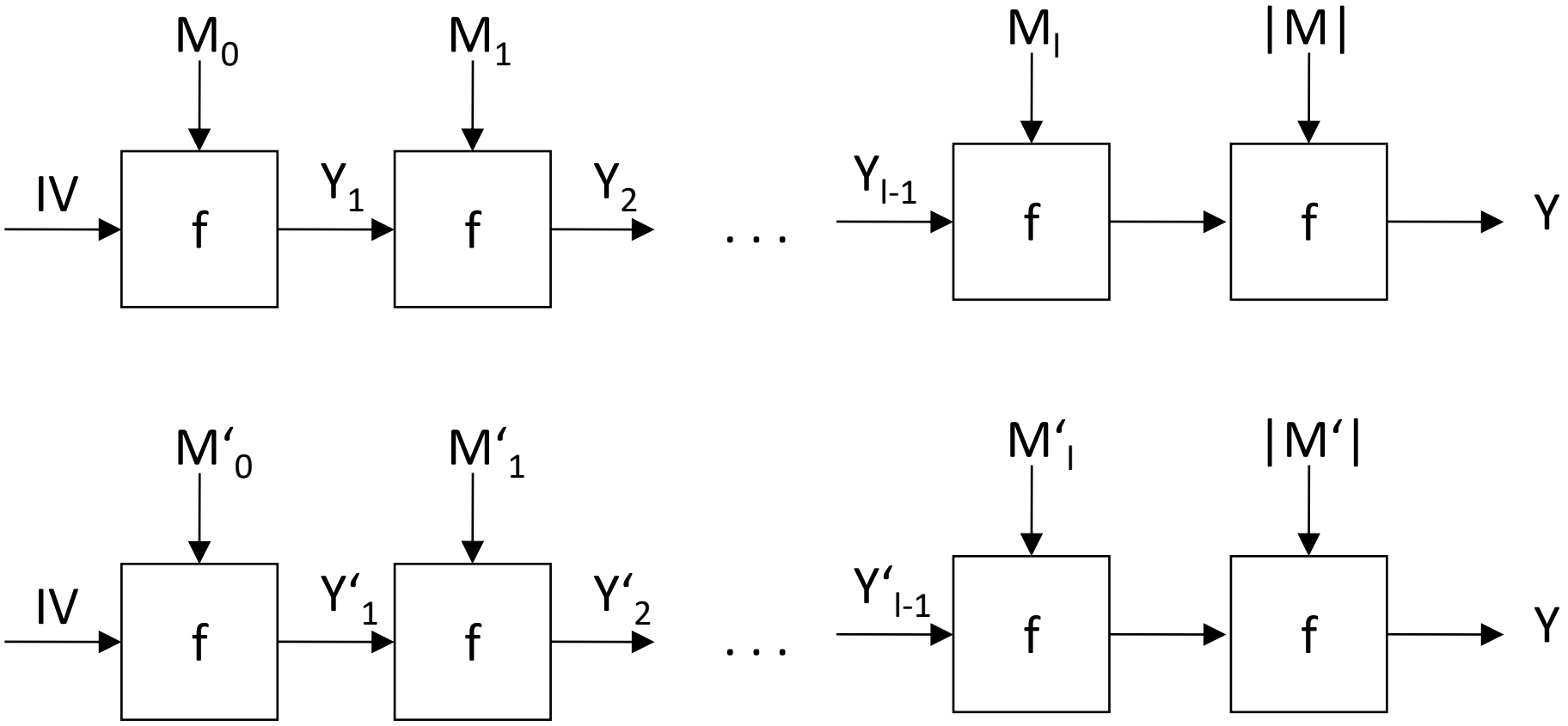
- Najznámejšou je Merkleho-Damgårdova iteratívna konštrukcia
  - Správa M je zarovnaná a rozdelená na bloky $M_0, M_1, \ldots M_l$ fixnej dĺžky
  - $f: \{0,1\}^{n+d} \rightarrow \{0,1\}^n$ je „kompresná“ funkcia



- Kľúčová vlastnosť MD-konštrukcie: zachováva odolnosť voči kolíziám
- H.f. MD, SHA1, SHA2 majú takúto konštrukciu

Merkleho-Damgårdova konštrukcia:

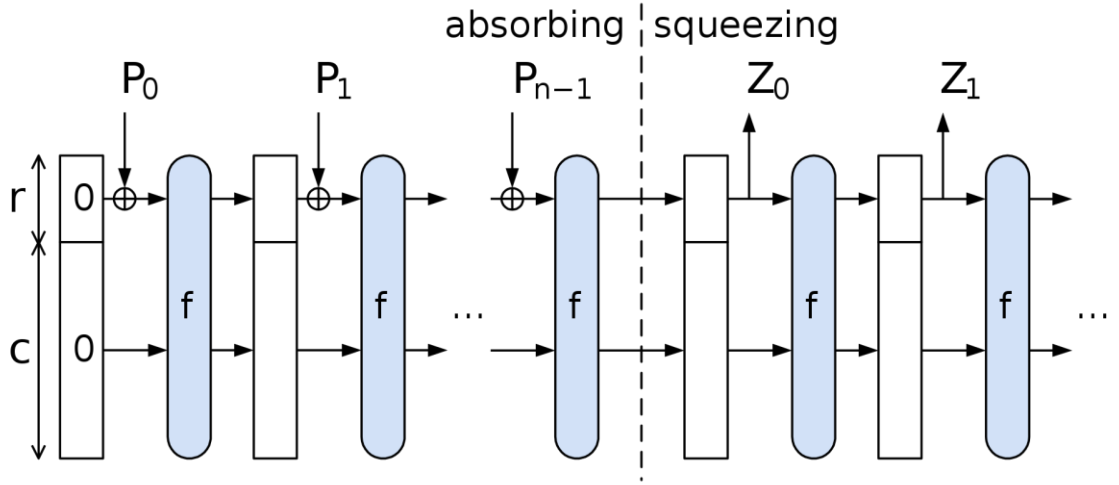
Odolnosť voči kolíziám



Hašovacie funkcie – ďalšie konštrukcie

- Konštrukcia z blokových šifier
  - Postupné spracovávanie blokov
  - Napr. Davies, Mayer $h_i = E_{M_i}(h_{i-1}) \oplus h_{i-1}$
  - Výstup $h(M) = h_l$ (posledný medzivýsledok)

- „Špongiová” konštrukcia
  - SHA-3



Zdroj: http://sponge.noekeon.org/, https://commons.wikimedia.org/w/index.php?curid=13463547

Bezpečnosť hašovacích funkcií

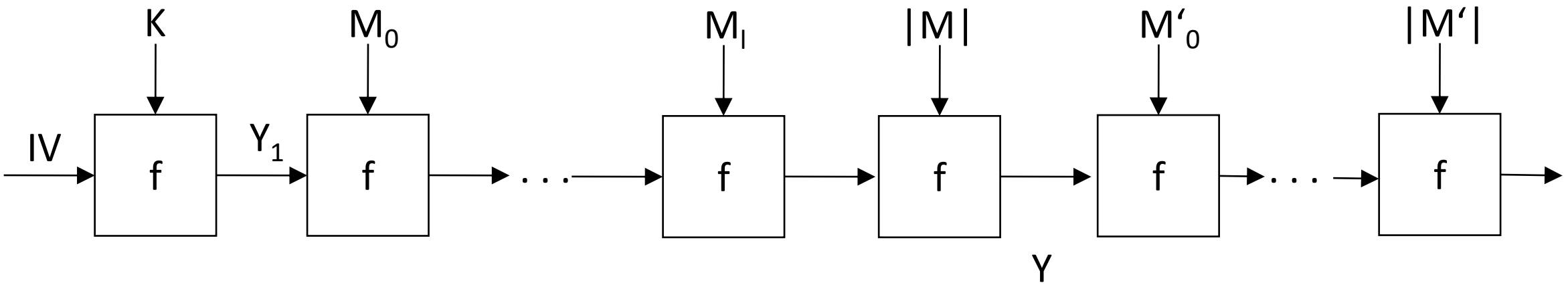
- MD5
  - Ľahké generovanie kolízií (v podstate okamžité)
  - Kolidujúce dokumenty (PS, XLS, ...) – aj jedna kolízia stačí
  - Generovanie falošného X.509 certifikátu CA (chosen prefix collision attack)
- SHA-1
  - Hľadanie kolízií odhadovaná zložitosť 2^{61}
    - *SHAttered* attack (2017) – našli kolidujúce PDF - potrebovali 2^{63} operácií SHA1
    - The attack required "the equivalent processing power of 6,500 years of single-CPU computations and 110 years of single-GPU computations"
  - Neodporúča sa používať (ak je potrebná odolnosť voči kolíziám)
- Sada SHA-2
  - Zatiaľ len útoky na redukované verzie (menší počet kôl)

- Zlyhanie konkrétnej h.f. vzhľadom na nejakú vlastnosť neznamená rozbitie všetkých konštrukcií, kde je použitá.
  - Bezpečnosť HMAC nezávisí na odolnosti použitej h.f. voči kolíziám

Merkleho-Damgårdova konštrukcia:

Length-extension attack

- Útočník len zo znalosti M a $Y = H(K \parallel M)$, kde K je tajná informácia, vie vypočítať $H(K \parallel M \parallel |M| \parallel M')$



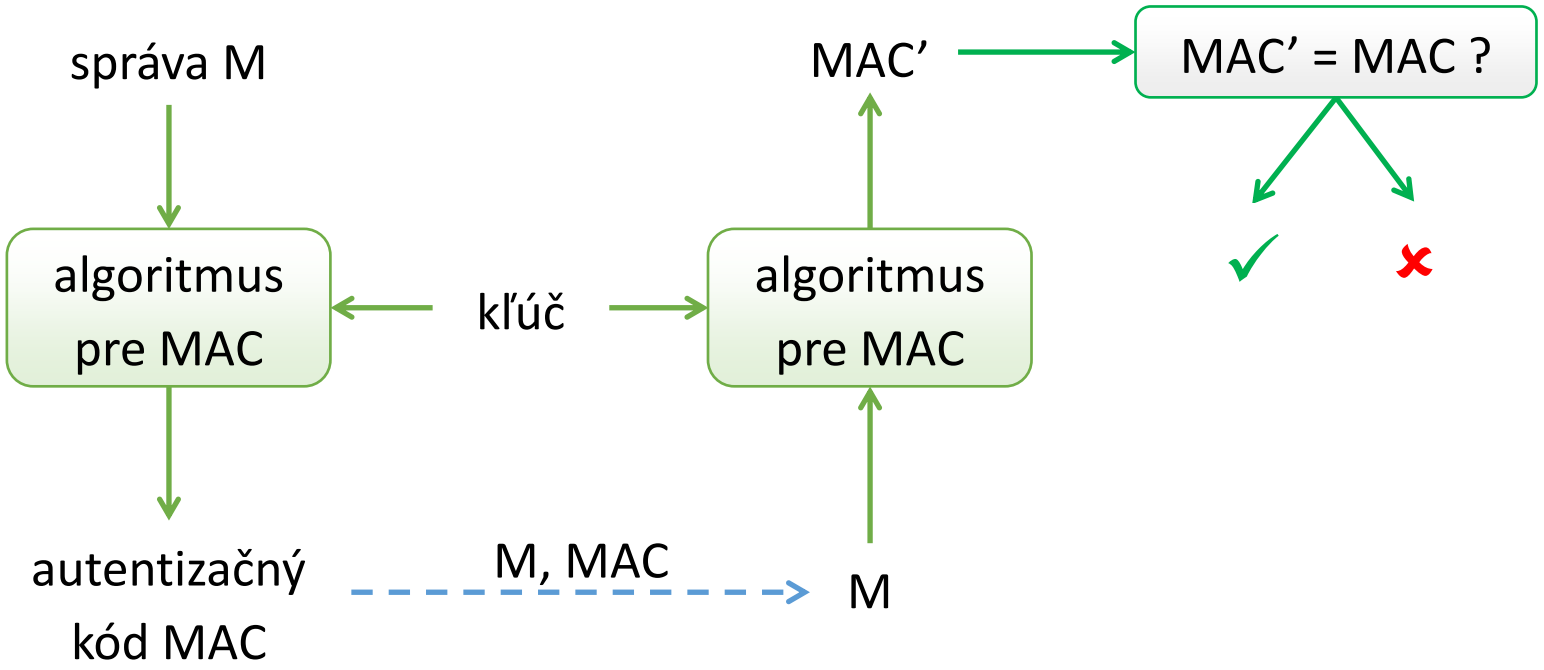
$H(K \parallel M)$

$H(K \parallel M \parallel |M| \parallel M'\_0)$

Autentizačné kódy správ (MAC)

- Symetrický kľúč (odosielateľ a príjemca)
- Zabezpečenie integrity a autentickosti správ (bez nepopierateľnosti !)
- Rýchle (oproti digitálnym podpisom)
  - v sieťových protokoloch sa počíta MAC pre každý paket
- Použitie napr. SSL/TLS, IPSec
- Konštrukcia - najčastejšie pomocou hašovacej funkcie parametrizovanej kľúčom

Autentizačné kódy správ



Autentizačné kódy správ - HMAC

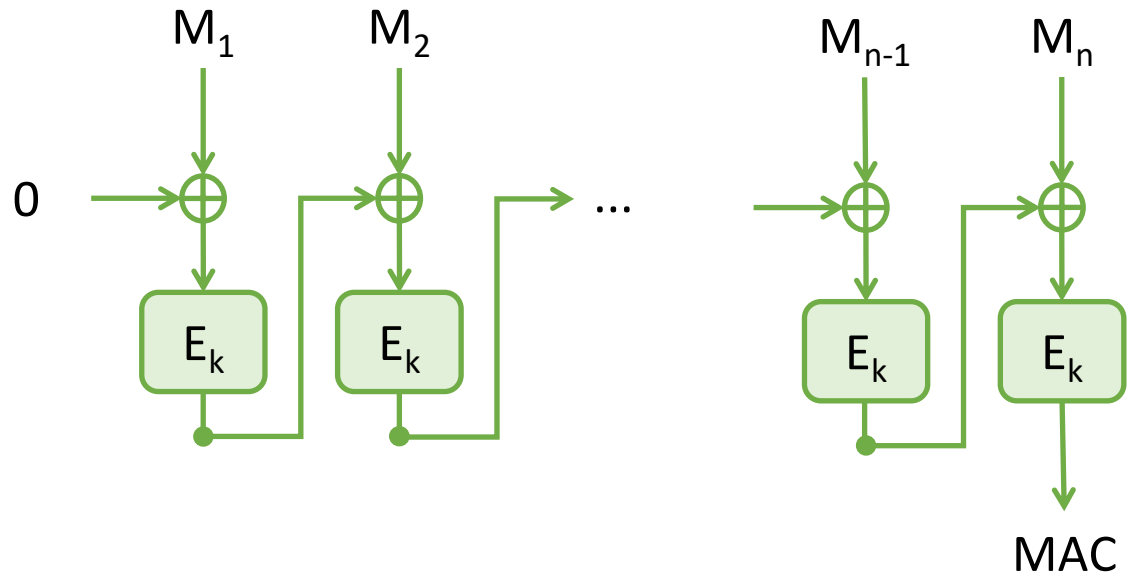
- najznámejšia konštrukcia: HMAC (RFC 2104)
  - H – hašovacia funkcia
  - *k* – kľúč – rovnako dlhý ako blok h.f. (512 bitov pre SHA1, 512/1024 bitov SHA2)
  - *m* – správa
  - opad/ipad – konštanty

$$MAC_k(m) = H\big(k \oplus opad \parallel H(k \oplus ipad \parallel m)\big)$$

- pre HMAC-MD5/SHA1 je opad = $(0x5C)^{64}$, ipad = $(0x36)^{64}$
- 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)



Bezpečný komunikačný kanál

- Dôvernosť & autentickosť ~ symetrické šifrovanie & MAC
- **Šifruj potom MAC (napr. IPSec)**:

$$E_{k1}(m), MAC_{k2}(c)$$

  - Bezpečný spôsob „vo všeobecnosti“ (za predpokladu, že E je IND-CPA bezpečná a MAC je bezpečný, potom takáto schéma je IND-CCA bezpečná)
- MAC potom šifruj (napr. SSL/TLS):

$$E_{k1}(m, MAC_{k2}(m))$$

- Šifruj a MAC (napr. SSH2):

$$E_{k1}(m), MAC_{k2}(m)$$

  - Vo všeobecnosti nebezpečné
  - Existujú E a MAC schémy, pre ktoré MAC a šifruj resp. šifruj a MAC nie sú bezpečné ani za predpokladu, že E a MAC sú bezpečné

Digitálne podpisy

- Autentickosť, integrita, nepopierateľnosť pôvodu
- Ekvivalent „vlastnoručného“ podpisu v elektronickom prostredí
- Asymetrická schéma:
  - súkromný kľúč – podpisovanie ⇒ len vlastník vie podpísať
  - verejný kľúč – overovanie ⇒ ktokoľvek vie overiť
- Podpis musí závisieť na podpisovanom dokumente/správe
- Zvyčajne sa podpisuje odtlačok dokumentu
  - výkonové dôvody
  - bezpečnostné dôvody (falšovanie náhodnej správy)
- Najpoužívanejšie konštrukcie: RSA, DSA, ECDSA

Digitálne vs. vlastnoručné podpisy

- „elektronické podpisy“ v legislatíve
- „digitálne podpisy“ v kryptológii
  - „digitálne podpisy“ $\subset$ „elektronické podpisy“

Iba nejaký elektronický „tag“
skoro žiadne garancie – rozhoduje súd

Dosiahnuteľné len s pomocou kryptológie, poskytujú určité „matematické“ garancie

Digitálne vs. vlastnoručné podpisy

- Digitálny podpis – ekvivalent vlastnoručného podpisu?
- Vlastnoručný podpis:
  - Identifikácia podpisujúceho, nepopierateľnosť autorstva
  - Nefalšovateľnosť
  - Potvrdenie dokumentu podpisujúcim (informovaný súhlas)
- V digitálnom svete je
  - jednoduché kopírovanie
  - jednoducho pozmeníme dokument

    ⇒ Digitálny podpis musí závisieť na dokumente

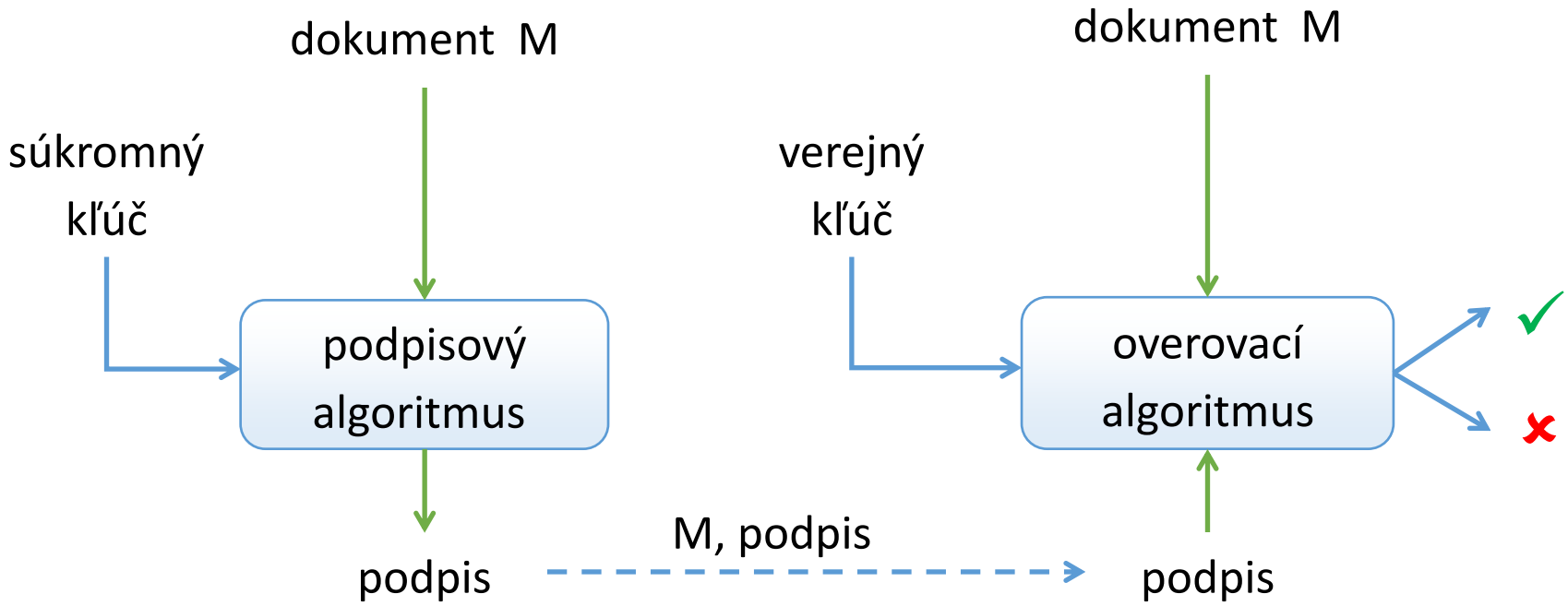
Digitálne vs. vlastnoručné podpisy

- Digitálne podpisy:
  - Identifikácia podpisujúceho, nepopierateľnosť autorstva
  - Nefalšovateľnosť
  - Potvrdenie dokumentu podpisujúcim (informovaný súhlas) ?
  - Autentickosť a integrita podpísaných údajov
  - Každý môže overiť podpis (zvyčajne)
- Niektoré vlastnosti nie je možné dosiahnuť len pomocou podpisových schém
  - PKI, zákony, predpisy (nie sú predmetom tejto prednášky)

Digitálne podpisy

- Identifikácia podpisujúceho
  - Crypto + PKI + bezpečný hardware
- Nefalšovateľnosť
  - Crypto
- Potvrdenie dokumentu podpisujúcim, nepopierateľnosť autorstva
  - Ťažké dosiahnuť v praxi
  - Crypto + zákon + predpisy + dôveryhodný harware a software
- Autentickosť a integrita podpísaných údajov
  - Crypto
- Každý môže overiť podpis (zvyčajne)
  - Crypto

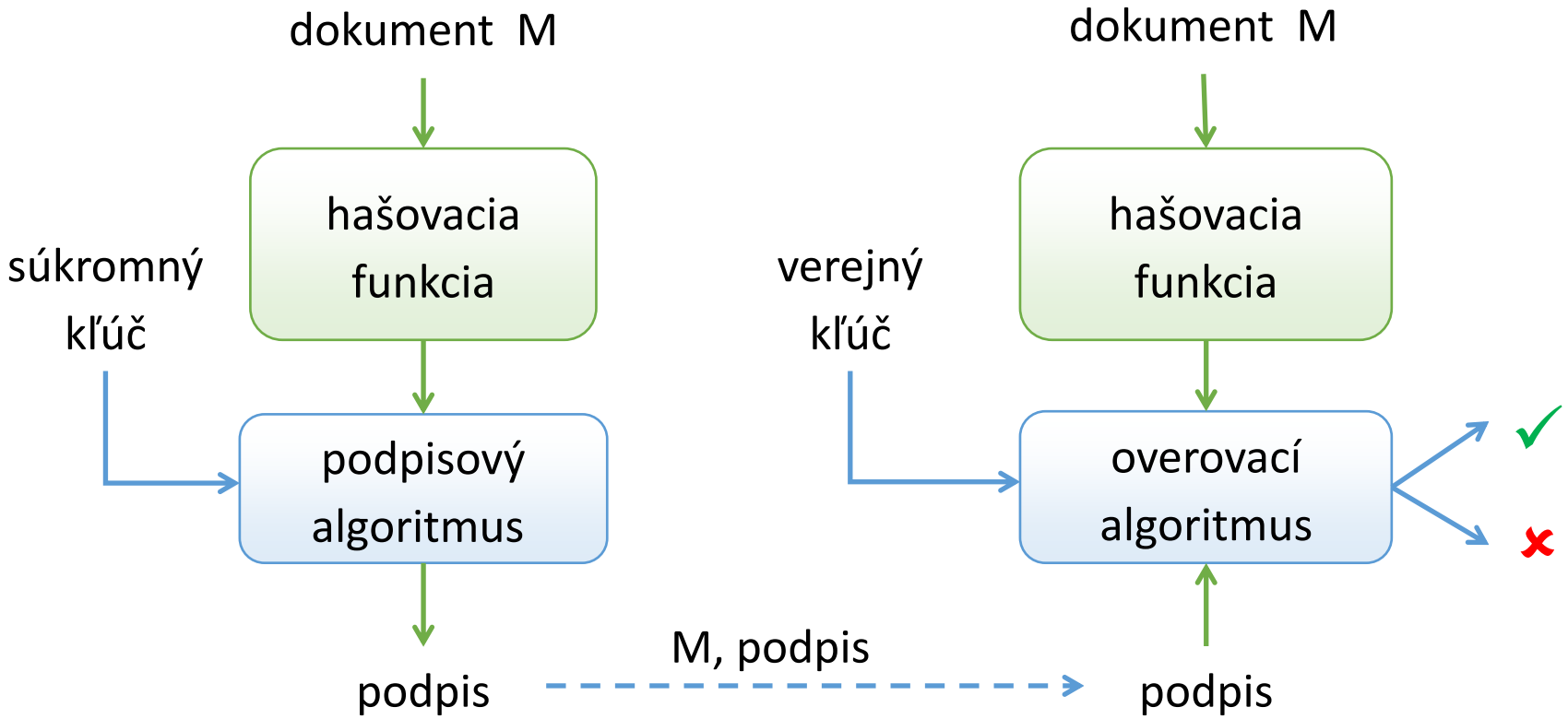
Schémy digitálnych podpisov



Asymetrická konštrukcia

- Súkromný kľúč – podpisovanie
- Verejný kľúč - overovanie

Schémy digitálnych podpisov
najčastejšia forma



Asymetrická konštrukcia

- Súkromný kľúč – podpisovanie
- Verejný kľúč - overovanie

Schémy digitálny podpisov

Podpisová schéma: (Gen, Sig, Vrf)

Gen efektívny (pravdepodobnostný polynomiálny) algoritmus, generuje verejný a súkromný kľúč (pk, sk)

Sig efektívny algoritmus, ktorý na základe správy a tajného kľúča podpisujúceho vytvorí podpis: $\sigma = Sig_{sk}(\mathrm{m})$

Vrf zvyčajne deterministický polynomiálny algoritmus, ktorý overuje podpis danej správy: $Vrf_{pk}(m, \sigma) \in \{\mathrm{true}, \mathrm{false}\}$

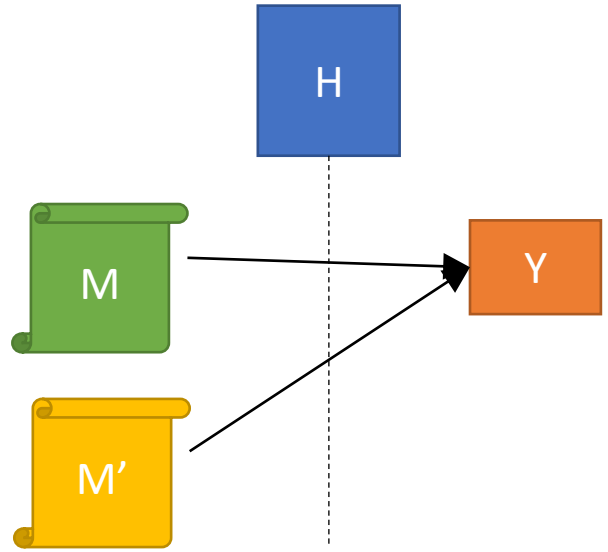
Korektnosť podpisovej schémy:

$$\forall (pk, sk) \leftarrow Gen\left(1^k\right) : \forall m : Vrf_{pk}\left(m, Sig_{sk}(m)\right) = true$$

Schémy digitálnych podpisov – poznámky

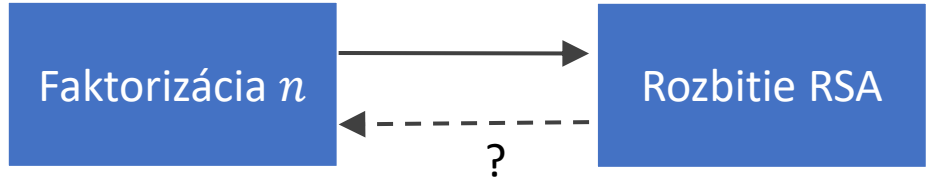
- Použitie hašovacej funkcie
  - Rýchlejšie podpisovanie – podpisuje sa len krátky odtlačok
  - Zabraňuje určitým druhom útokov, napr. falšovanie náhodnej správy

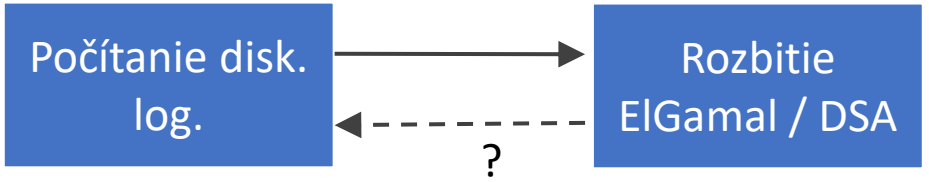
- Bezpečnosť schémy závisí aj od vlastností použitej h.f.
  - napr. potrebujeme odolnosť voči kolíziám



V praxi používané podpisové schémy

- RSA
  - Postavené na RSA probléme, ktorý súvisí s problémom faktorizácie
- ElGamal
  - postavená na probléme diskrétneho logaritmu
- DSA
  - štandardizovaný NISTom v roku 1994
  - postavený na probléme diskrétneho logaritmu
  - obmena ElGamalovej schémy





*Bezpečnosť podpisových schém

- Útočník má prístup k verejnému kľúču

1. Cieľ – čo chce útočník dosiahnuť
   - čo najjednoduchší
2. Možný scenár útoku – ako prebieha útok
   - čo najsilnejší
3. Zdroje – výpočtová sila útočníka
   - čo najväčšie, ale dosiahnuteľné

*Bezpečnosť podpisových schém

1. Cieľ
   a) Odhaliť súkromný kľúč
   b) Možnosť vytvoriť falošný podpis pre akúkoľvek správu
   c) Možnosť vytvoriť falošný podpis danej správy m
   d) Možnosť vytvoriť falošný podpis útočníkom zvolenej správy m
      - aj keď m nedáva zmysel
      - “Existential forgery”

*Bezpečnosť podpisových schém

2. Scenár útoku
   a) Len so znalosťou verejného kľúča
   b) Útočník pozná niekoľko dvojíc (m, σ), kde σ je podpis správy m
   c) Útočník si môže zvoliť niekoľko správ, ku ktorým následne dostane ich podpis
   d) Adaptívny útok, kde má útočník prístup k podpisovaciemu orákulu $Sig_{sk}(\cdot)$ – môže si dať podpísať akúkoľvek správu
      - „Adaptive Chosen Message Attack“ (CMA)

3. Zdroje – cca 2^{80} operácií v rozumnom čase, resp. polynomiálny algoritmus

*Bezpečnosť podpisových schém

- EUF-CMA (existentially unforgeable under the chosen message attack)
  - CMA – útočník má prístup k orákulu $Sig_{sk}(\cdot)$
  - EUF – úlohou útočníka je vytvoriť správu m (na ktorú sa nepýtal orákula) a jej podpis σ, kde $Vrf_{pk}(m, \sigma) = true$
- Podpisová schéma je EUF-CMA, ak pravdepodobnosť úspechu akéhokoľvek efektívneho útočníka je zanedbateľná
  - Za predpokladu, že nejaký matematický problém je ťažký (faktorizácia, diskrétny logaritmus, DH-problém)
- Efektívny útočník:
  - $< 2^{80}$ operácií, alebo
  - pravdepodobnostný polynomiálny algoritmus

RSA podpisová schema

„učebnicová“ verzia

- Veľmi podobná šifrovacej schéme RSA – „obrátené“ transformácie

Inicializácia

- $n = p \cdot q$, kde p, q sú veľké prvočísla
- $e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$
- Verejný kľúč: $pk = (n, e)$
- Súkromný kľúč: $sk = d$

Nepoužívať!

Podpisovanie: $\sigma = m^d \bmod n$

Overovanie: $\sigma^e \bmod n == m$?

- Korektnosť vyplýva z vlastností (bijektívnosti) RSA

RSA podpisová schema

„učebnicová“ verzia – **Problémy**

- Vieme podpisovať iba krátke správy ($m \in \mathbb{Z}_n$)
- Falšovanie náhodnej správy:
  - Útočník vie vygenerovať dvojicu (správa, podpis) – Existential forgery
  - $(\underbrace{\sigma^e \ mod \ n}_{m}, \sigma)$, kde $\sigma \overset{\$}{\leftarrow} \mathbb{Z}_n$
  - Útočník nemá kontrolu nad správou m
- Z dvojíc $(m_1, \sigma_1), (m_2, \sigma_2)$ vie útočník vytvoriť validný podpis $(m_1 m_2 \bmod n, \sigma_1 \sigma_2 \bmod n)$

RSA podpisová schema

2. pokus – „hašovaná“ verzia

Inicializácia: rovnaké ako v predošlom prípade

Podpisovanie: $\sigma = H(m)^d \bmod n$

Overovanie: $\sigma^e \bmod n \; == H(m)$?

✓Môžeme podpisovať ľubovoľne dlhé správy

✓Odolnosť H voči nájdeniu vzoru ⇒ odolnosť schémy voči falšovaniu náhodnej správy

- H musí byť odolná voči kolíziám
- RSA-FDH (Full Domain Hash) schéma využívajúca h.f. H, ktorej obraz ∈ $\mathbb{Z}_n$
  - Dokázateľná bezpečnosť v modeli s náhodným orákulom
- Obraz H je zvyčajne kratší ako n ⇒ padding

PKCS #1 v 1.5 (EMSA-PKCS1-v1_5)

- Konštrukcia štandardizovaná v roku 1998
  - Bez formálneho dôkazu bezpečnosti
- Často používaná v praxi, napr. v X.509 certifikátoch:
  - „sha1RSA“ alebo „PKCS #1 sha1 with RSA encryption“
- Padding pre odtlačok H(M):

$$0\text{x}00 \parallel 0\text{x}01 \parallel 0\text{xff} \parallel \cdots \parallel 0\text{xff} \parallel 0\text{x}00 \parallel H(m)$$

“Moreover, while no attack is known against the EMSA-PKCS-v1_5 encoding method, a gradual transition to EMSA-PSS is recommended as a precaution against future developments.” (RFC 3447)

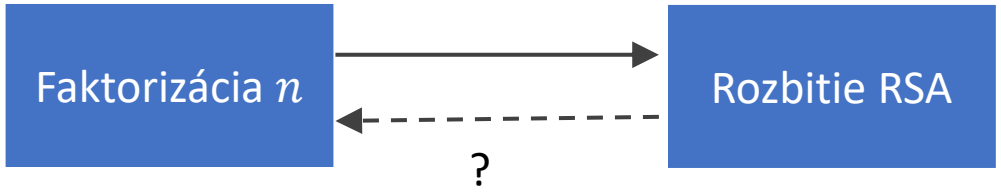
RSA-PSS

- Probabilistic Signature Scheme – štandardizovaná v roku 2002 PKCS #1 v 2.1, RFC 3447
- Dokázateľne bezpečná v modeli s náhodným orákulom

Úspešný útočník (EUF-CMA)
$\Rightarrow$ vieme riešiť RSA problém (počítať $\sqrt[e]{\cdot} \mod n$)

! Za predpokladu, že H je modelované ako náhodné orákulum

- dôkaz nehovorí o bezpečnosti schémy ak je použitá štandardná h.f.

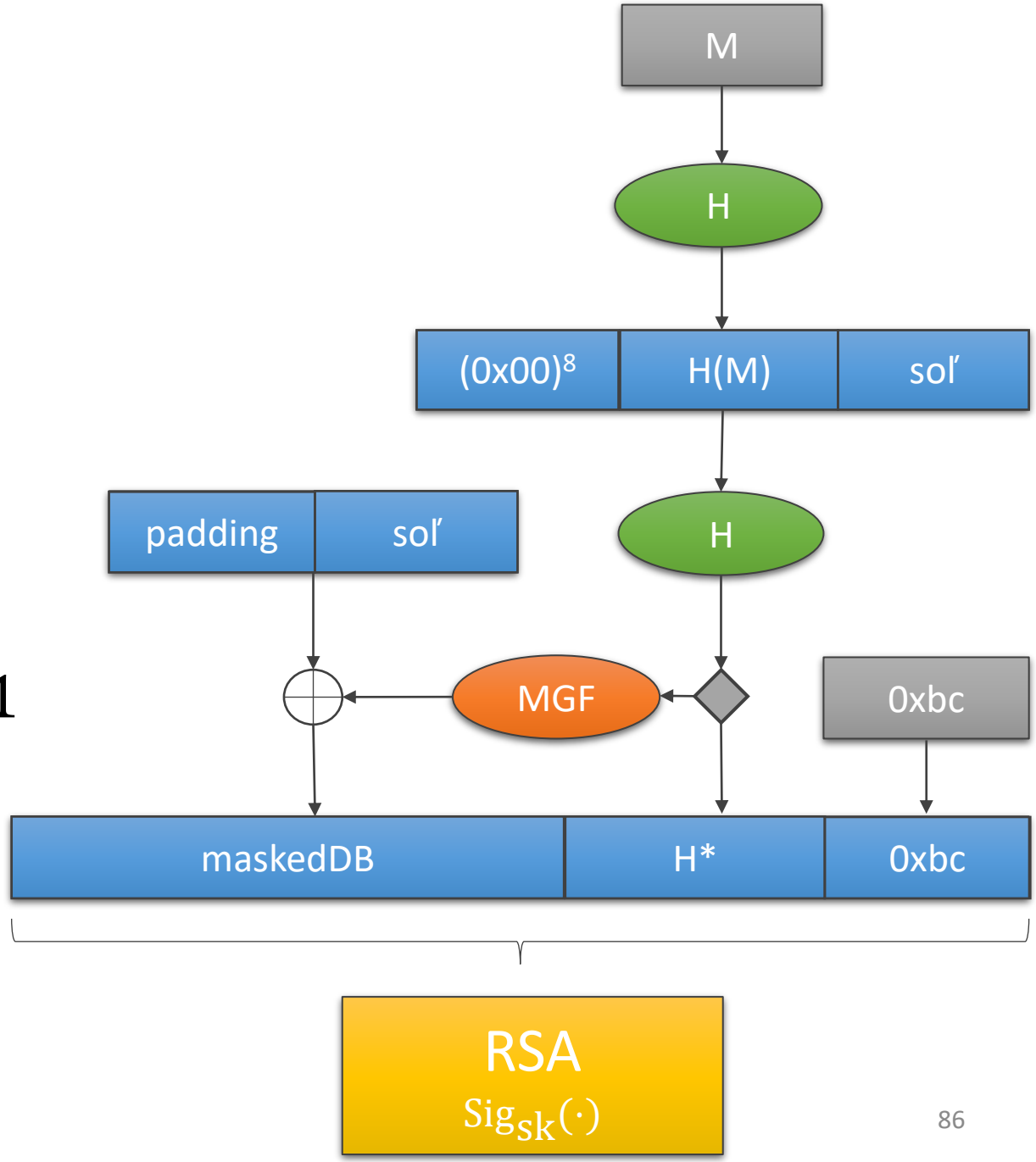


RSA-PSS - podisovanie

MGF - Mask Generation Function
(použité aj v RSA-OAEP)

soľ - sekvencia náhodných bajtov

padding=$0x00 \| \cdots \| 0x00 \| 0x01$



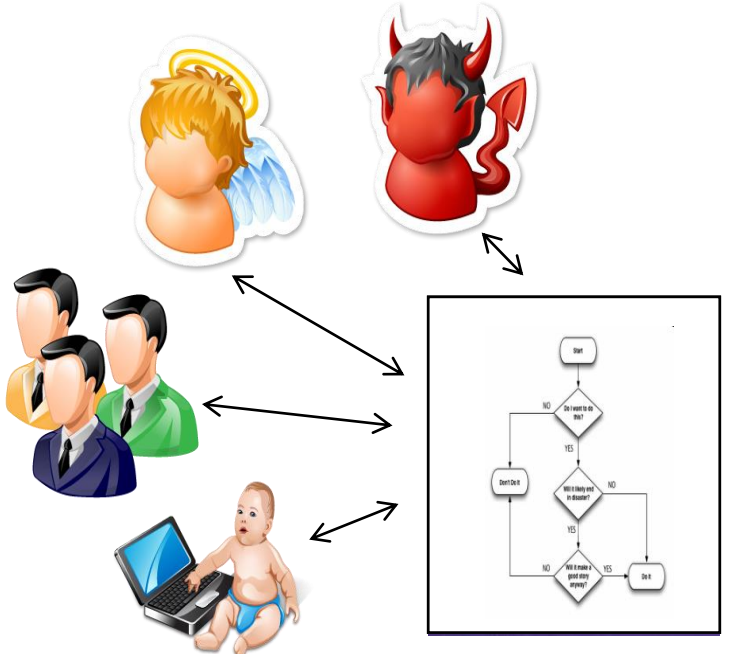
*Náhodné orákulum

- Kompikovanú h.f. nahradíme jednoduchým matematickým objektom – náhodnou funkciou
- T.j. odtlačok Y = H(M) je vybraný uniformne náhodne pre každé M
  - Y je nezávislé na M
  - Pre rôzne M_1,M_2, ..., odtlačky $Y_i = H(M_i)$ sú navzájom úplne nezávislé
  - Je ťažké nájsť kolíziu
  - Je ťažké hľadať predobrazy
  - atď.
- Náhodnú funkciu v praxi nevieme implementovať, keďže jej popis by si vyžadoval obrovskú tabuľku vzorov a príslušných obrazov.

*Model s náhodným orákulom

[Bellare-Rogaway, 1993]

1. „Predstieraj “, že H je náhodná
2. Navrhni kryptosystém, ktorý bude využívať takú (náhodnú) H
   - Dokáž bezpečnosť systému vzhľadom na „náhodné orákulum“
3. Nahraď náhodné orákulum skutočnou h.f.
   - Dúfaj, že systém ostane bezpečný



*Model s náhodným orákulom: odôvodnenie

- Nech S je schéma (napr. podpisová) využívajúca h.f. H
- Ak dokážeme, že S je bezpečná za predpokladu, že H je náhodná, potom akýkoľvek útok na S musí využiť nejakú „nonrandom“ vlastnosť hašovacej funkcie H
  - T.j. mali sme si vybrať lepšiu H, bez tejto „nonrandom“ vlastnosti
- Problém: ako vieme, ktoré „nonrandom“ vlastnosti sú dôležité / využiteľné útočníkom?
- Problém: Existujú podpisové schémy, ktoré
  - Sú bezpečné vzhľadom na náhodné orákulum
  - Sú ľahko prelomiteľné pre každú skutočnú h.f.

DSA

- Súčasť štandardu DSS - FIPS 186-4 (DSA, RSA, ECDSA)
- Bezpečnosť súvisí s problémom diskrétneho logaritmu
- Vychádza z ElGamalovej schémy

Inicializácia :

1. Vygeneruj prvočísla p, q (napr. $|p| = 2048, |q| = 256$), pričom $q|(p-1)$
2. Vygeneruj $h \xleftarrow{\$} \mathbb{Z}_{p-1}$, vypočítaj $g = h^{p-1/q} \bmod p$
3. Súkromný kľúč: $x \xleftarrow{\$} \mathbb{Z}_q^*$
4. Verejný kľúč: $y = g^x \bmod p$ a parameter p, q, g

DSA

Podpisovanie - $\mathrm{Sig}_{sk}(m)$:

1. $r = g^k \bmod p \bmod q$, kde $k \xleftarrow{\$} \mathbb{Z}_q^*$
2. $s = k^{-1}(H(m) + xr) \bmod q$
3. Ak $r = 0$ alebo $s = 0$ začni znova krokom 1 (veľmi málo pravdepodobné)
4. $\sigma = (r, s)$

Overovanie - $\mathrm{Vrf}_{pk}(m, (r, s))$

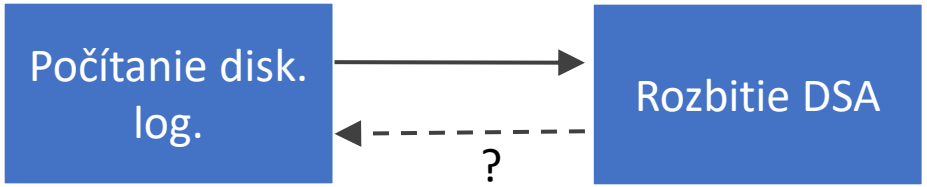
1. Over, či $r, s \in \mathbb{Z}_q^*$
2. $u_1 = H(m) \cdot s^{-1} \bmod q$
   $u_2 = r \cdot s^{-1} \bmod q$
3. Over, či $(g^{u_1} \cdot y^{u_2} \bmod p) \bmod q = r$

DSA

- Korektnosť:
  - $(g^{u_1} \cdot y^{u_2} \bmod p) \bmod q = g^{H(m)s^{-1}+xrs^{-1}} \bmod \text{p} \bmod \text{q}$
    $= g^{s^{-1}(H(m)+xr)} \bmod p \bmod q = g^k \bmod p \bmod q$
    $= r$
- Ak $r = 0$, potom podpis nezávisí na súkromnom kľúči x (podpis možno ľahko falšovať)
- Ak $s = 0$, potom $s^{-1} \bmod q$ neexistuje
- Predpovedateľné k vedie ku kompromitácii súkromného kľúča
  - (2010) Sony PS3 ECDSA s konštantným k
  - Existuje variant deterministickej voľby k odvodenej zo s.k. a $H(m)$ (RFC 6979)

DSA

- Parametre p, q, g môžu byť zdieľané medzi viacerými používateľmi systému
  - Málo používané, potreba zabezpečiť, že neboli vyberané útočníkom
  - Existuje overiteľná procedúra na generovanie týchto parametrov (súčasť štandardu)



Bezpečnosť DSA:

- Súvisí s problémom diskrétneho logaritmu
  - Vieme počítať disk. log. ⇒ vieme rozbiť DSA
- Neexistuje formálny dôkaz bezpečnosti DSA
  - Ani v modeli s náhodným orákulom

DSA, RSA

n	Dĺžka hašu	Dĺžka podpisu	Bezpečnosť
RSA-FDH			
1024	1024	1024	80
2048	2048	2048	112
3072	3072	3072	128
RSA-PSS			
1024	160-512	1024	80
2048	160-512	2048	112
3072	160-512	3072	128

p	q	Dĺžka hašu	Dĺžka podpisu	Bezpečnosť
DSA				
1024	160	160	320	80
2048	224	224	448	112
3072	256	256	512	128

ECDSA

- Variant DSA nad eliptickými krivkami
- Problém diskrétneho logaritmu je nad eliptickými krivkami ťažší
  ⇒ môžeme použiť kratšie kľúče
- Dĺžka kľúča 160-256 bitov ekvivalentná 1024-3072 bitovému RSA
  - Ekvivalentné 80-128 bitovej bezpečnosti symetrických schém

Dĺžka kľúča	Dĺžka podpisu	Dĺžka hašu	Bezpečnosť
160	320	160	80
224	448	224	112
256	512	256	128

Výkonové porovnanie

	podpisovanie [operácie/s]	overovanie [operácie/s]
RSA-1024	6 100	93 281
RSA-2048	857	27 496
RSA-4096	118	7 370
ECDSA-224 (nistp224)	15 375	7 349
ECDSA-256 (nistp256)	9 024	3 697
ECDSA-521 (nistp521)	3 252	1 501

i7-2600, 3.40GHz, Ubuntu 12.04 LTS 64-bit, openssl 1.0.1, 8kB bloky

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

(Ne)determinizmus

- Nederministické schémy
  - Pre jednu správu existuje veľa podpisov
  - ElGamal, DSA, RSA-PSS
  - Generovanie náhodných čísel je ťažké, hlavne na smart kartách
  - Existujú metódy ako ich zmeniť na deterministké schémy bez straty bezpečnosti
    - RFC 6979 pre DSA
- Deterministické schémy
  - Pre každú správu existuje len jeden podpis
  - RSA-FDH, RSA-PKCS #1 v 1.5

Timestamping

- Časom môže dôjsť ku kompromitácii podpisovej schémy / súkromného kľúča podpisujúceho
- Eva odhalí súkromný kľúč Boba ⇒ môže falšovať Bobov podpis na akejkoľvek správe
- Autentickosť všetkých Bobových podpisov pred kompromitáciou kľúča je teda taktiež otázna
- Problém je, že nevieme určiť kedy bola správa podpísaná
- Timestamping – dôkaz, že správa bola podpísana v určitom čase

Timestamping

Ako na to

Nech pub je nejaká nepredpovedateľná verejne dostupná informácia (napr. hodnota akcií na burze)

Postup: nech m je správa, ktorú chce Bob podpísať

1. Bob vypočíta $z = H(m)$
2. Bob vypočíta $z' = H(z \parallel \text{pub})$
3. Bob vypočíta $y = Sig_{sk}(z')$
4. Bob zverejní (z, pub, y) v novinách

Zjavne, podpis musel byť vytvorený najneskôr vtedy, ako bolo (z, pub, y) zverejnené

Nepredpovedateľnosť pub znamená, že podpis nemohl byť vytvorený ani predtým

Timestamping

Dôveryhodný poskytovateľ časových pečiatok

- Autorita (TSA), ktorá garantuje časovú pečiatku dokumentu
- Alica chce opečiatkovať dokument m
  1. A vypočíta $z = H(m)$ a pošle ho TSA
  2. TSA vygeneruje pečiatku ts
     - vypočíta $z' = H(z \parallel ts)$
     - podpíše $y = Sig_{sk_{TSA}}(z')$
  3. A udržuje (ts, y) ako dôkaz
- Každý môže overiť podpis a pečiatku
- TSA nevidí samotný dokument

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

Digitálne podpisy - zhrnutie

Bez ohľadu na podpisovú schému, digitálne podpisy vyžadujú nasledovné

1. **Kvalitné algoritmy**
   - Bezpečnosť niektorých podpisových schém bola kompromitovaná
2. **Kvalitná implementácia**
   - Dobrý algoritmus implementovaný s chybou nefunguje
3. **Súkromný kľúč musí zostať utajený**
   - V prípade jeho odhalenia môže útočník falšovať podpisy
4. **Identita držiteľa verejného kľúča musí byť overiteľná**
   - PKI
5. **Používatelia (a software) musia dodržiavať postup podpisovacieho protokolu**

Digitálne podpisy - zhrnutie

- Digitálne podpisy vs. elektronické podpisy vs. ručné podpisy
- Digitálny podpis: autenticita, integrita, nefalšovateľnosť, nepopierateľnosť pôvodu
- Najznámejšie podpisové schémy
  - RSA-FDH, RSA-PKCS #1 v1.5, RSA-PSS
  - ElGamal, DSA, ECDSA
- Použitie rovnakých kľúčov na šifrovanie a podpisovanie sa neodporúča
- Časové pečiatky
- Bezpečný hardware je nevyhnutný

Obsah

- Úvod
- Základné kryptografické prvky
  - ✓Symetrické šifry
  - ✓Asymetrické šifry
  - ✓Hašovacie funkcie
  - ✓Autentizačné kódy
  - ✓Digitálne podpisy
- **Dĺžky kľúčov**
- Kryptológia v kontexte
- Štandardy

Kryptografické kľúče

- Správa kľúčov (ako – algoritmy a postupy)
  - Generovanie
  - Distribúcia
  - Ukladanie a prístup
  - Ničenie kľúčov
  - Postupy pri kompromitácii kľúčov

- Dĺžka kľúčov – nutná ale nepostačujúca podmienka bezpečnosti
  - generický útok – úplné preberanie množiny *K*

Bezpečnosť – dĺžka kľúča

Aká dĺžka kľúča je dostatočná?

- Deep Crack – DES, $200k, 1 kľúč ~22 hodín, 1998
  - … (2007) 1 DES kľúč ~ 3 dni, $12 000
- Odhady:
  - 80 bitový kľúč ~ 1 rok, $8M, 2006
  - 80 bitový kľúč ~ 1 mesiac, $33M, 2010
  - 80 bitový kľúč ~ 1 deň, $1M, 2015

- 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

- https://www.blockchain.com/explorer/charts/hash-rate

- Tažba bitcoinu
- Spoločne minery vykonajú zhruba
  $\approx 230 \cdot 10^{18}$ Hash / s

- $\approx 2^{94}$ SHA-256 za rok

Bezpečnosť – dĺžka kľúča

Aká dĺžka kľúča je dostatočná?

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

NSA Suite B Cryptography

- Odporúčania 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. Šifry	Eliptické krivky	hašovanie
Secret	128/256	256	256
Top secret	256	384	384

ECRYPT Report

- ECRYPT – európska sieť excelencie v kryptológii
- Pravidelný report o doporučených dĺžkach kľúčov
- Level 1-8
  - level 4 – najmenšia všeobecná ochrana (do 2020)
    „veľmi krátkodobá ochrana voči agentúram“
  - level 7 – dlhodobá ochrana (cca. 30 rokov)
  - level 8 – „predvídateľná budúcnosť“

	Sym. Šifry	RSA	Eliptické krivky	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 ECRYPT II (2012)
- Porovnanie rôznych metód: www.keylength.com
- Bezpečnosť vs. výpočtové nároky

Obsah

- Úvod
- Základné kryptografické prvky
  - ✓Symetrické šifry
  - ✓Asymetrické šifry
  - ✓Hašovacie funkcie
  - ✓Autentizačné kódy
  - ✓Digitálne podpisy
- Dĺžky kľúčov
- **Kryptológia v kontexte**
- Štandardy

Kryptológia v kontexte

Kryptografia je obvykle použitá v niečom „väčšom“

- operačný systém, čipové karty, databázový systém, mail, webová aplikácia, e-commerce, sieťové protokoly, ...

- Aká dôveryhodná je implementácia?
- Akým spôsobom sú spravované kľúče:
  - generovanie?
  - distribúcia?
  - backup?
  - riadenie prístupu?
  - ničenie?
- Postranné kanály (čas, spotreba zdrojov, hyperthreading, virtualizácia, chybové hlášky, ...)?

...je ľahké urobiť chybu

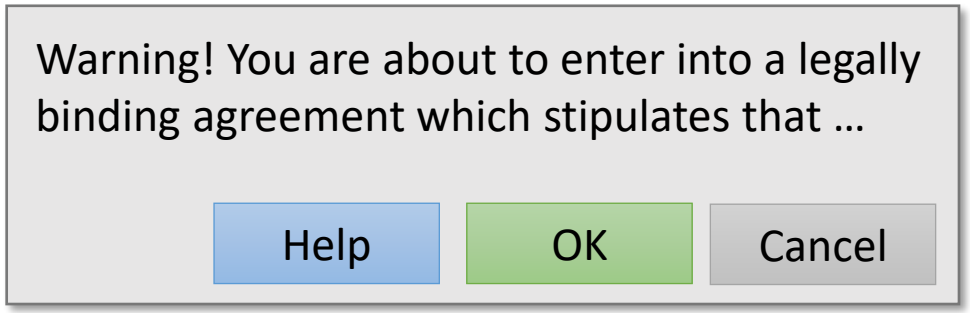
Kryptológia v kontexte:

Nepopierateľnosť autorstva v praxi

- Teoreticky môžeme hovoriť, že digitálne podpisy poskytujú nepopierateľnosť autorstva
  - Bez znalosti súkromného kľúča nikto nevie vytvoriť validný podpis
- V praxi je to však veľmi ťažké dosiahnuť
  - Existencia manuálneho podpisu znamená, že podpisujúci mal dočinenia a videl podpísaný dokument
  - Existencia dig. podpisu znamená, že niekedy niečo vykonalo matematickú operáciu nad nejakými dátami

Nepopierateľnosť autorstva v praxi

- Validný digitálny podpis môže byť veľmi jednoducho popretý
  - „Software ma nedostatočne upozornil na dôsledky vykonaných akcií“
    - „babička klikla na zlé tlačidlo a prišla o dom“
  - „Urobil to vírus“
    - Univerzálna výhovorka
  - Používateľ zverejnil svoj súkromný kľúč
    - Môžeme ho potrestať za ľahostajnosť, ale nie za obsah podpísaného dokumentu
  - Ak má podpis časovú pečiatku, používateľ bude tvrdiť, že v čase podpisovania ešte nevedel o kompromitácii svojho súkromného kľúča
- Potrebujeme vytvoriť ekvivalent „informovaného súhlasu” (WYSIWYS)



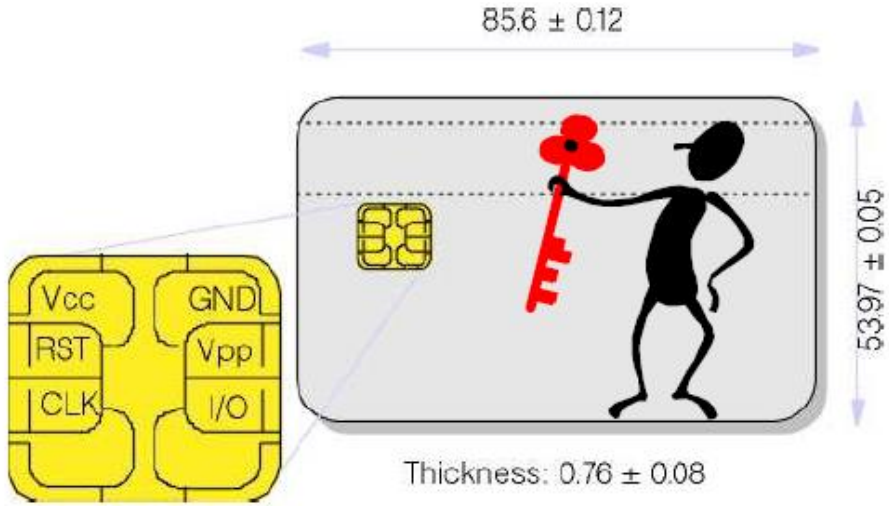
Smart karty, bezpečný hardware

Súkromný kľúč musí zostať utajený!

- Aj pred podpisujúcim

Musí byť bezpečne

- Generovaný
- Uložený
- Používaný
- Zálohovaný
- Vymazaný



Počítač neposkytuje dostatočnú bezpečnosť

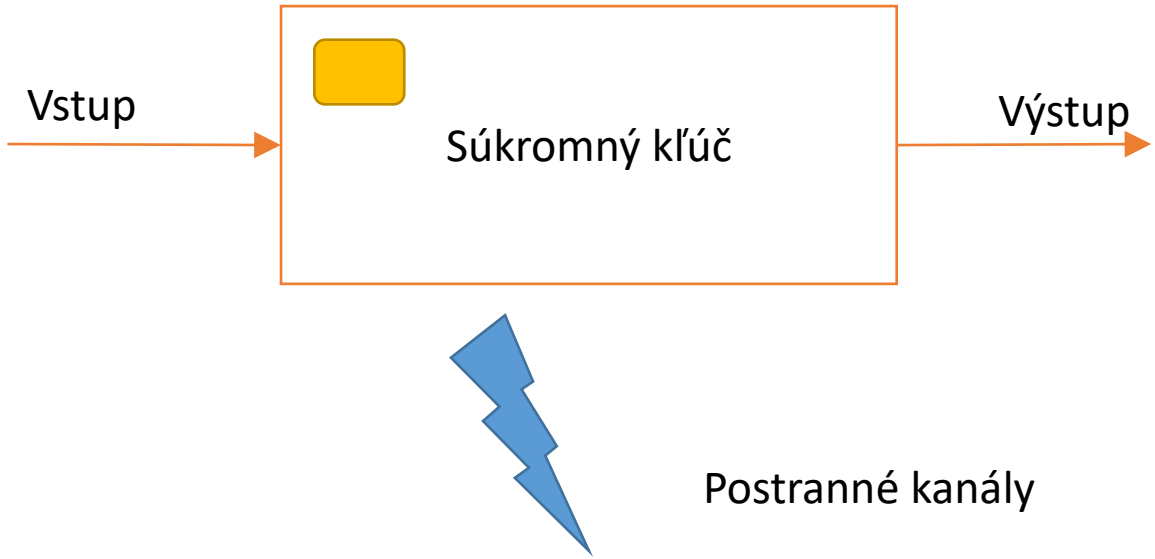
Riešenie: smart karty

- Pre zavedenie nepopierateľnosti autorstva je bezpečný hardware kľúčový

Kryptológia v kontexte:

Útoky postrannými kanálmi

- Timing útoky
- Meranie napätia
- EM merania – TEMPEST
- Meranie hlučnosti
- Chybové kanály
- ...

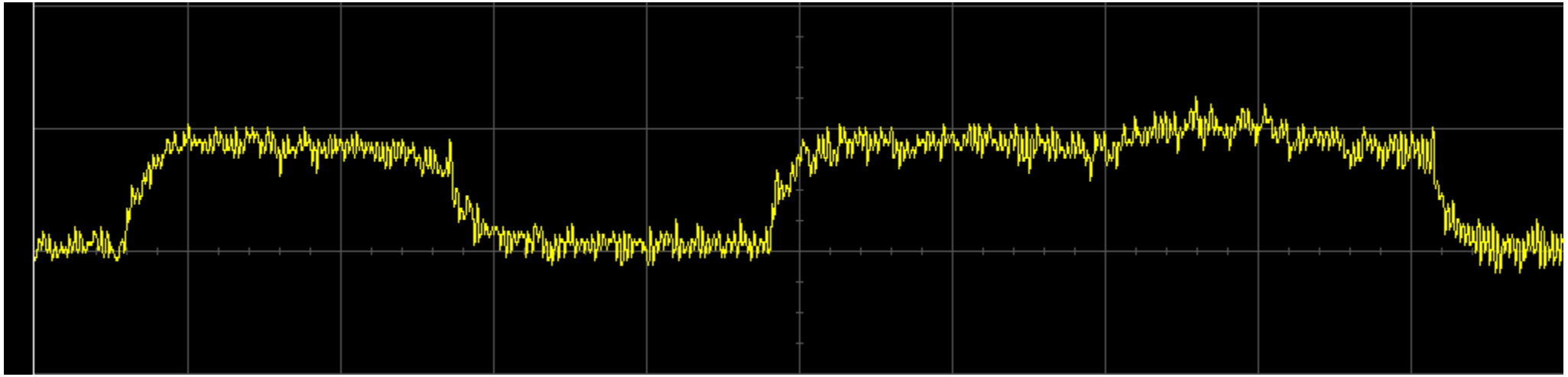


Timing útoky

- Čas behu nejakého kroku algoritmu môže závisieť od dát
- Čas beh algoritmu „square-and-multiply“ na počítanie modulárneho umocňovania závisi od počtu 1 v kľúči
- 2003, Boneh, Brumley – timing útok na SSL využívajúci slabinu v implementácii RSA využívajúcu optimalizáciu cez čínsku zvyškovú vetu
  - Na odhalenie súkromného kľúča stačilo niekoľko hodín

- Ochrana: „maskovacie” techniky v implementácii algoritmov

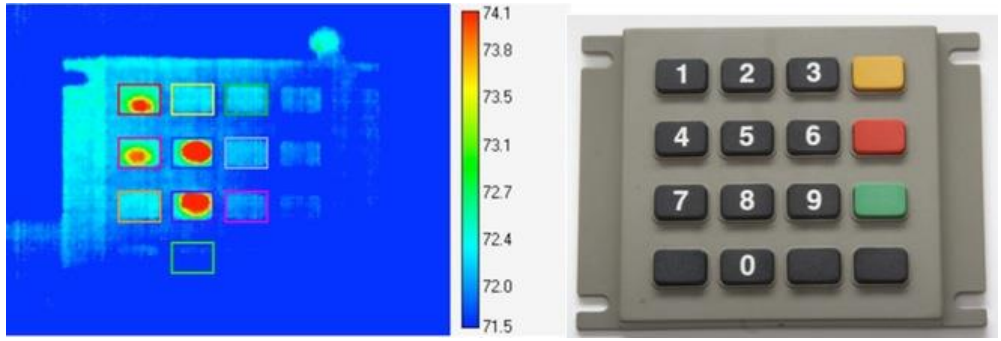
Meranie napätia



- Priebeh napätia počas výpočtu RSA podpisu na smart karte
- Ľavý vrchol – priebeh napätia CPU počas kroku algoritmu bez násobenia
- Pravý vrchol – napätie CPU počas kroku s násobením
- Môžeme tak rozoznať bity súkromného kľúča

Ďalší útok: Bankomaty a PIN

- Pomocou termokamery je možné čítať PIN



- Niekedy je dokonca možné určiť poradie stlačenia kláves
- Kovové klávesy sú náchylnejšie

Politika a kryptografia

Post-Snowden éra

- Snowden, 2013
- Bývalý zamestnanec CIA, pracoval aj pre firmu pracujúcu pre NSA (subkontraktor)
- Zverejnil tajné dokumenty o sledovacích projektoch NSA
- NSA sa snažila špehovať a analyzovať veľkú časť globálnej internetovej komunikácie
- Potenciálny útočníci už nie sú len kyber-kriminálnici
  - Ale organizácie s miliardovým rozpočtom

Politika a kryptografia

Post-Snowden éra

- Veľká časť v praxi používaných kryptografických algoritmov je štandardizovaných NISTom
  - AES, SHA-1-2-3, ECDSA, …
- Odhalenia Snowdena o praktikách NSA dávajú do popredia otázku, či tieto štandardy neobsahujú zadné vrátka
  - Spomeňme si na pseudo-náhodný generátor DUAL_EC_DRBG
  - AES a SHA-3 boli vyberané verejnou kryptografickou súťažou

Štát vs. Krypto - Lavabit

- Lavabit - šifrovaná web-mailová služba
  - Využíval ju aj Snowden na komunikáciu s novinármi a právnikmi
  - Služba navrhnutá programátormi, ktorým vadila bezpečnostná politika Gmailu
  - Poskytovala silnú kryptografickú ochranu emailov svojich používateľov
    - Na úrovni, ktorá by mala byť odolná aj voči spravodajským službám
- August 2013 – Lavabit skončil svoju prevádzku potom, ako bol požiadaný súdom o poskytnutie svojich súkromných kľúčov
- Následne skončila svoju prevádzku aj ďalšia podobná služba Silent Circle
  - Pre istotu vymazali všetky šifrovacie kľúče

YOU ARE COMMANDED to appear and testify before the United States district court at the time, date, and place shown below to tesify before the court's grand jury. When you arrive, you must remain at the court until the judge or a court officer allows you to leave.

Place:	UNITED STATES DISTRICT COURT 401 Courthouse Square Alexandria, Virginia 22314	Date and Time:	July 16, 2013	9:30 AM

You must also bring with you the following documents, electronically stored information, or objects (blank if not applicable):

In addition to your personal appearance, you are directed to bring to the grand jury the public and private encryption keys used by lavabit.com in any SSL (Secure Socket Layer) or TLS (Transport Security Layer) sessions, including HTTPS sessions with clients using the lavabit.com web site and encrypted SMTP communications (or Internet communications using other protocols) with mail servers;

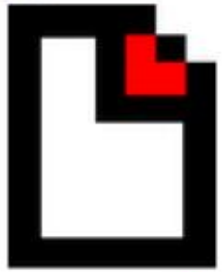
Any other information necessary to accomplish the installation and use of the pen/trap device ordered by Judge Buchanan on June 28, 2013, unobtrusively and with minimum interference to the services that are accorded persons with respect to whom the installation and use is to take place;

If such information is electronically stored or unable to be physically transported to the grand jury, you may provide a copy of the information to the Federal Bureau of Investigation. Provision of this information to the FBI does not excuse your personal appearance.

Date: July 11, 2013

CLERK OF COURT [redacted]

August 2013



Lavabit

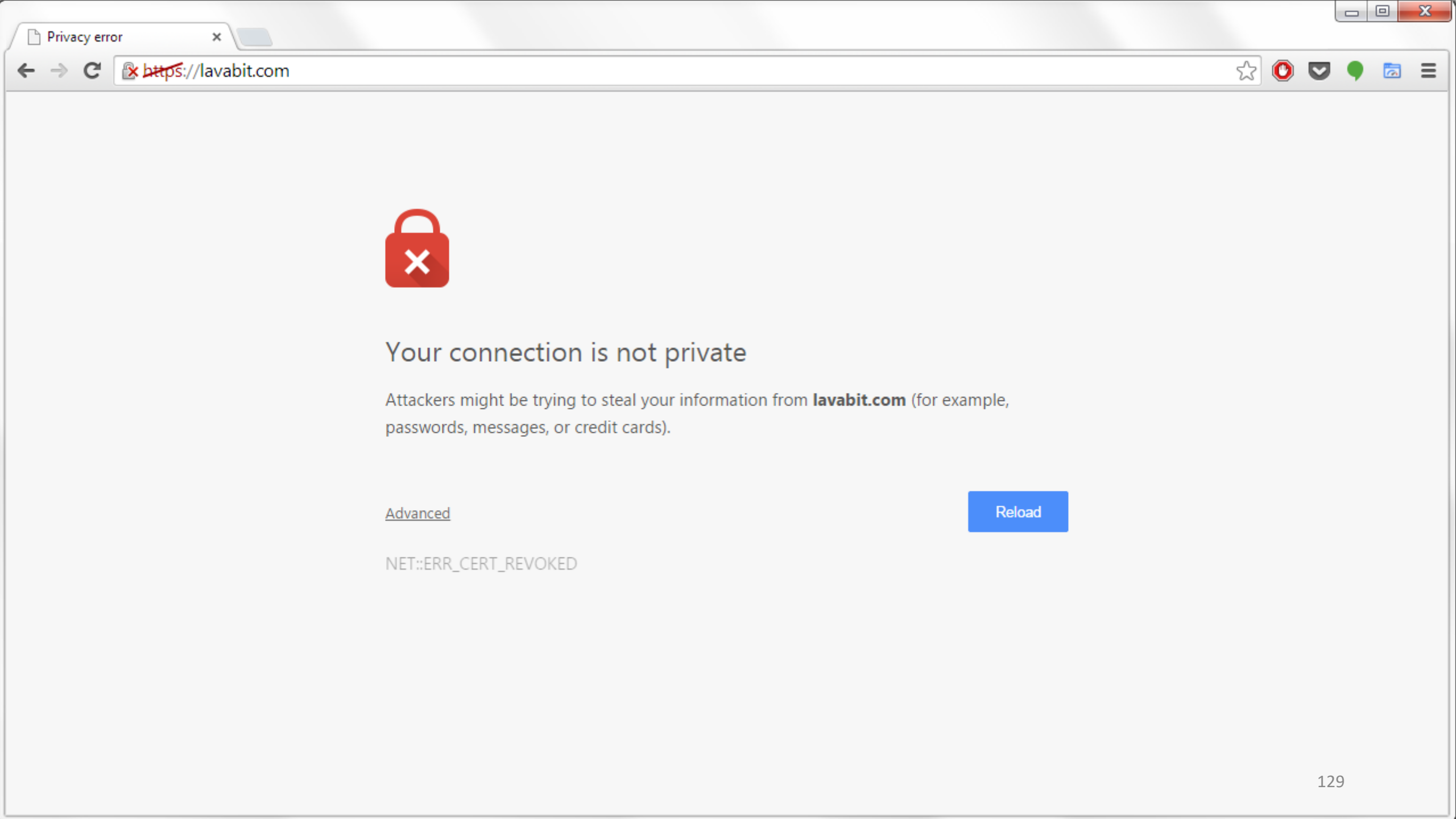
My Fellow Users,

I have been forced to make a difficult decision: to become complicit in crimes against the American people or walk away from nearly ten years of hard work by shutting down Lavabit. After significant soul searching, I have decided to suspend operations. I wish that I could legally share with you the events that led to my decision. I cannot. I feel you deserve to know what's going on--the first amendment is supposed to guarantee me the freedom to speak out in situations like this. Unfortunately, Congress has passed laws that say otherwise. As things currently stand, I cannot share my experiences over the last six weeks, even though I have twice made the appropriate requests.

What's going to happen now? We've already started preparing the paperwork needed to continue to fight for the Constitution in the Fourth Circuit Court of Appeals. A favorable decision would allow me resurrect Lavabit as an American company.

This experience has taught me one very important lesson: without congressional action or a strong judicial precedent, I would _strongly_ recommend against anyone trusting their private data to a company with physical ties to the United States.

Sincerely,
Ladar Levison


Privacy error
https://lavabit.com
Your connection is not private
Attackers might be trying to steal your information from lavabit.com (for example, passwords, messages, or credit cards).
Advanced
Reload
NET::ERR_CERT_REVOKED

Generátory náhodných čísel

- Častá príčina zlyhania kryptografických systémov
  - Generovanie skutočne náhodných čísel je ťažké
- (Ne)kryptografické generátory pseudo-náhodných čísel bývajú predvídateľné
- Určite nepoužívajte “rand()” funkciu zabudovanú v programovacom jazyku
  - srand(seed) inicializuje generátor, nastaví `state=seed`
  - `rand()`:
    - `state=f(state)`, kde f je nejaká lineárna funkcia
    - `return state;`
  - Generovanie 128-bitového kľúča KEY=`rand()‖rand()‖rand()‖rand()`
  - Entropia kľúča je iba 32 bitov!

Generátory náhodných čísel

- Generátor pseudo-náhodných čísel použiteľný v kryptografii
  - Výstup neodlíšiteľný od úplne náhodného akýmkoľvek efektívnym algoritmom
  - Pokiaľ možno, zakaždým reinicializovaný novým zdrojom entropie
  - Malo by byť ťažké uhádnuť interný stav generátora
    - Napr. entropia by nemala pochádzať iba z času (súborov)

- „Cold boot“ problémy
  - Server práve naštartoval a potrebuje zdroj náhodnosti ... je možné získať dosť entropie, ak server beží len pár sekúnd?

Generátory náhodných čísel

najznámejšie zraniteľnosti

- Netscape, implementácia SSL, 1995
  - Pseudonáhodný generátor inicializovaný na základe času, ID procesu a ID nadradeného procesu – všetko ľahko predvídateľné hodnoty
  - Generátor nebol verejne dostupný („security through obscurity“), na analýzu využili reverzné inžinierstvo
- Windows 2000 / XP, 2007
  - Leo Dorrendorf - *Cryptanalysis of the Random Number Generator of the Windows Operating System,* http://eprint.iacr.org/2007/419.pdf
  - Vážne nedostatky vstavaného generátora
  - Ak sa útočníkovi podarilo získať stav generátora (napr. cez buffer overflow), mohol predpovedať všetky predchádzajúce aj nasledujúce vygenerované hodnoty (napr. SSL šifrovacie kľúče)
  - Opravené v XP SP3

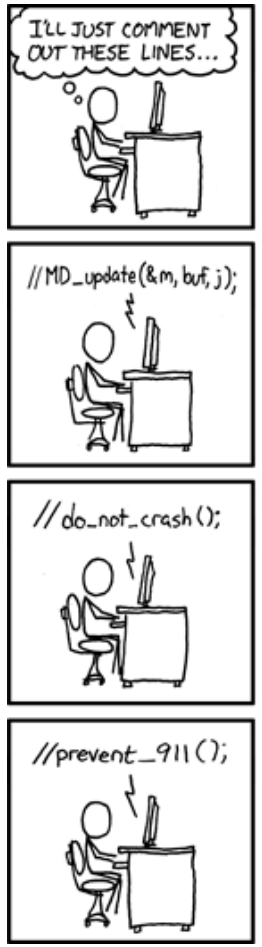
Generátory náhodných čísel

najznámejšie zraniteľnosti

- Debian OpenSSL, 2008
  - Debian distribúcie Linuxu
  - Zmeny v kóde pseudo-náhodného generátora drasticky znížili entropiu
  - Chyba bola spôsobená vývojárom, ktorý na základe upozornení kompilátora odstránil na pohľad zbytočný kód

```
MD_Update(&m,buf,j); /* neinicializovaná hodnota */
[ .. ]
MD_Update(&m,buf,j); /* neinicializovaná hodnota */
```

    - Odstránený kód zabezpečoval zvýšenie entropie
    - Po jeho odstránení bol generátor inicializovaný len na základe ID procesu (max. 32 768 hodnôt)
  - Chyba umožnila odhaliť vygenerované súkromné kľúče
  - Veľké množstvo kľúčov a certifikátov muselo byť vygenerovaných znovu



Generátory náhodných čísel

najznámejšie zraniteľnosti

- PlayStation 3, 2010
  - Sony využíva ECDSA algoritmus na podpisovanie softvéru pre PlayStation 3
  - ECDSA vyžaduje dobrý PRNG
  - Opakované použitie k vedie k odhaleniu súkromného kľúča
  - SONY použilo zakaždým tú istú hodnotu k

Podpisovanie v ECDSA $\mathrm{Sig}_x(m)$:

1. $k \xleftarrow{\$} \{1, \ldots, n-1\}$
2. $(x_1, y_1) = k \times G$
3. $r = x_1 \bmod n$
4. $s = k^{-1}(H(m) + rx) \bmod n$
5. Ak $r = 0$ alebo $s = 0$ začni znova krokom 1
6. $\sigma = (r, s)$

Generátory náhodných čísel

najznámejšie zraniteľnosti

- Implementácia Bitcoinov v Androide, 2013
  - Chyba v Java triede SecureRandom - možné kolízie v hodnote k pri použití ECDSA
  - Kolízia vedie k odhaleniu súkromného kľuča – možnosť ukradnúť Bitcoiny z Androidovej peňaženky
- DUAL_EC_DRBG, 2007, 2013
  - NIST Special Publication 800-90 – kolekcia pseudo-náhodných generátorov
  - DUAL_EC_DRBG – odporúčaný / navrhovaný aj NSA
    - Kryptografia nad eliptickými krivkami - štandard obsahuje aj sadu odporúčaných kriviek / konštánt
    - 2007, Shumow, Ferguson ukázali, že konštanty mohli byť skonštruované tak, aby umožňovali „zadné vrátka“ k náhodnému generátoru
  - 2013, REUTERS – Snowden: NSA zaplatilo firme RSA $10 mil., aby bol predvolený generátor práve DUAL_EC_DRBG

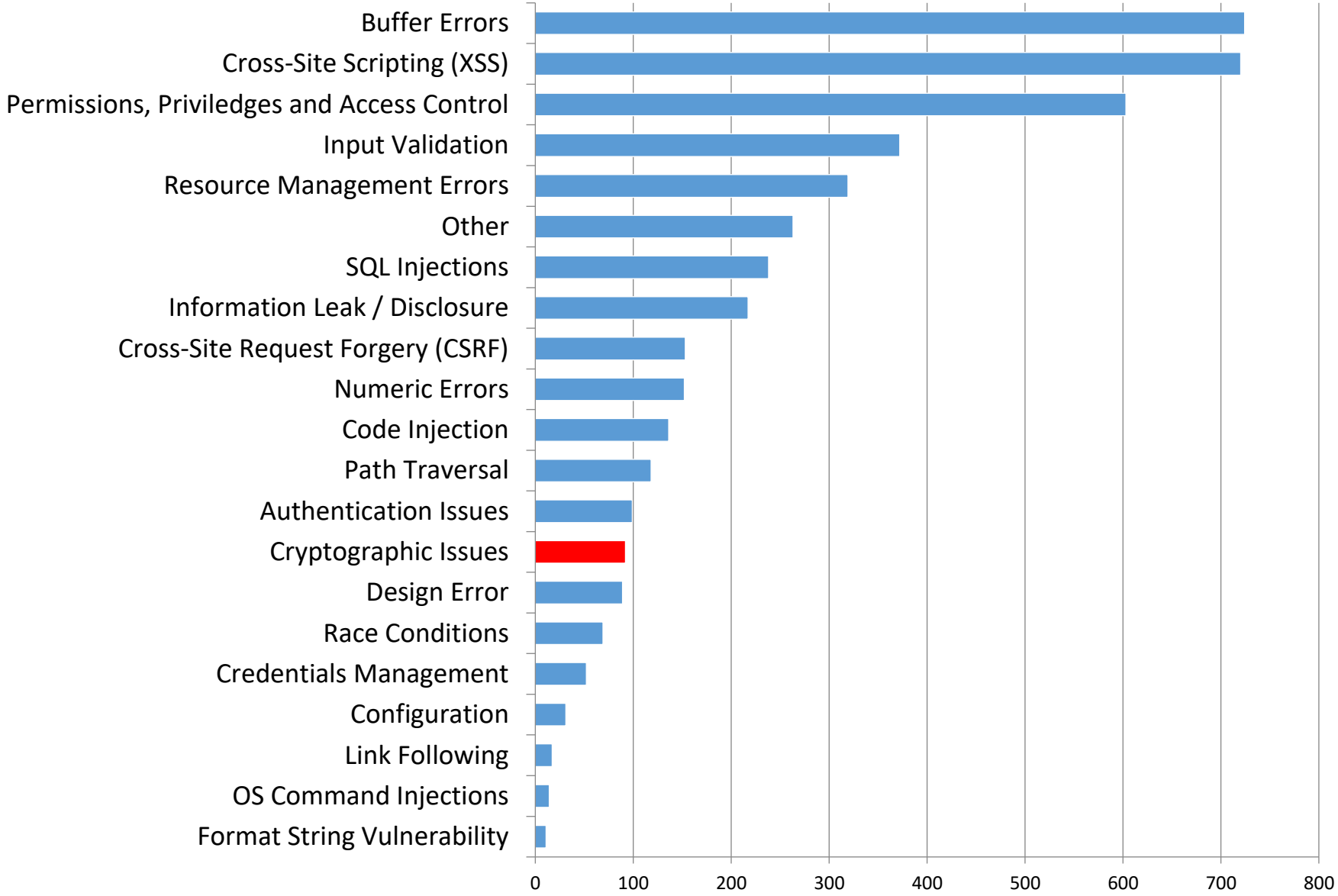
Kryptografia a zraniteľnosti

- Útoky na kryptografické mechanizmy
  - Obvykle sú slabiny v správe kľúčov a v implementácii
  - Protokoly – zvyčajne slabiny v protokole, bez ohľadu na algoritmy
- Niektoré implementačné slabiny/útoky
  - Útok postrannými kanálmi (napr. timing útok)
  - Nesplnenie bezpečnostných predpokladov (napr. náhodnosť)
  - Slabiny v protokoloch (napr. útoky na SSL/TLS)
  - Slabé algoritmy (napr. proprietárne algoritmy ako CCS)

Kryptografia a zraniteľnosti

- NIST: NVD (National Vulnerability Database)
  - SW zraniteľnosti a ich klasifikácia (typ, závažnosť a pod.)
- Najčastejšie zraniteľnosti v „Cryptographic Issues“:
  - použitie nekvalitného zdroja náhodnosti pri generovaní kľúčov,
  - nedostatočná (neúplná) kontrola certifikátov,
  - nekorektná implementácia kryptografických algoritmov alebo protokolov,
  - fixné heslá servisných účtov alebo heslá odvodené z verejne známych údajov

Počty zraniteľností publikovaných v roku 2012 podľa NVD



Štandardy

- Kryptografické algoritmy (šifry, podpisové schémy, hašovacie funkcie)
  - Primárne NIST, široká akceptácia
- Protokoly
  - Zvyčajne RFC
- Štandardy v IB riešia kryptografiu len okrajovo
- ISO/IEC 27000:
  - Politika používania kryptografických opatrení
  - Riadenie kľúčov
- ISO/IEC 15408 (Common Criteria):
  - Správa kryptografických kľúčov
  - Prevádzka kryptografie

Common Criteria

- ISO/IEC 15408 – Evaluation criteria for IT security
- Použitie: jednotný jazyk pre
  - používateľov: špecifikujú požiadavky
  - výrobcov: popíšu vlastnosti produktov
  - nezávislé testovanie: vyhodnotí

- EAL: evaluation assurance level (EAL1 - EAL7)
- Flexibilné – hodnotenie veľkých systémov (napr. operačné systémy), aj malých (napr. čipové karty, aplikácie)

Common Criteria

- 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.

FIPS PUB 140-2

- Security Requirements for Cryptographic Modules
- 4 bezpečnostné úrovne
- Oblasti: špecifikácia modulu, role, služby, autentizácia, fyzická bezpečnosť modulu, samotestovanie, správa kľúčov, elektromagnetické vyžarovanie a ďalšie
- Certifikované moduly v roku 2012
  - 68 osvedčení na úrovni 1, 95 na úrovni 2, 37 na úrovni 3 a žiadne na úrovni 4
- Od roku 2019 FIPS PUB 140-3
- Certifikácia nie je zárukou bezpečnosti!
  - Príklad: certifikované USB kľúče: https://www.schneier.com/blog/archives/2010/01/fips_140-2_leve.html

Záver

- Bez kryptológie je tažké (nemožné?) dosiahnuť bezpečnosť IS

- Niekedy sú použité kryptografické konštrukcie zlé
- Niekedy sú kryptografické konštrukcie použité zle

Odporúčania 1

- ✓ Používajte štandardné kryptografické algoritmy, schémy a protokoly
- ✓ Používajte dostatočné dĺžky kľúčov
- ✓ Pravidelne meňte kľúče a heslá
- ✓ Dbajte na kvalitné generovanie kľúčov a voľbu hesiel
- ✓ Majte premyslené, čo robiť po kompromitácii kľúčov alebo hesiel
- ✓ Ak môžete, použite certifikované riešenia
- ✓ Poznajte konfiguračné možnosti kryptografických riešení a ich bezpečnostné dopady

Odporúčania 2

- ✓ Uprednostnite AES-256 a SHA-512 pred inými symetrickými šiframi a hašovacími funkciami
- ✓ Ak môžete, uprednostnite schémy založené na eliptických krivkách
- ✓ Ak používate RSA schémy, uprednostnite RSA-OAEP pre šifrovanie a RSA-PSS pre podpisovanie
- ✓ Použite hašovaciu funkciu s dvojnásobnou dĺžkou odtlačku ako je dĺžka symetrického kľúča
- ✓ V praxi často krát požiadavky na funkčnosť a pohodlie víťazia nad bezpečnosťou – dbajte, aby to nebolo K.O.

Varovania

- Kryptografia nie je miesto na kreativitu a ad-hoc riešenia
- Dlhodobo nezmenené kľúče považujte za prezradené
- Šifrovanie nezabezpečuje integritu ani autentickosť údajov
- Autentizačné kódy ani digitálne podpisy nezabezpečujú dôvernosť
- Obvykle je heslo najslabším „kľúčom“ v systéme
- Certifikácia nie je náhradou bezpečného používania
- Kryptografia nenahradí iné organizačné a technické bezpečnostné opatrenia

Ďakujem za pozornosť