Hašovací funkce

- Hašovací funkce, charakteristika zprávy, jednocestná funkce, message digest, digest, hash, hash function, one way function
- je to funkce F taková, že
  - je aplikovatelná na argument libovolné velikosti
  - její výstupní hodnota má konstantní délku (zpravidla 128, 160 nebo 256 bitů)
  - lze rychle spočíst F(x)
  - pro dané *y* je výpočetně nezvládnutelné nalézt takové *x*, aby platilo *F(x)=y* *(first preimage resistance)*
  - pro dané *x* je výpočetně nezvládnutelné nalézt takové *x'<>x*, aby platilo *F(x')=F(x)* *(second preimage resistance)*
  - je výpočetně nezvládnutelné nalézt takové *x' a x, x'<>x*, aby platilo *F(x')=F(x)* *(collision resistance)*
- implementace
  - MD2, MD4, MD5
  - SHS (Secure Hash Standard), SHA

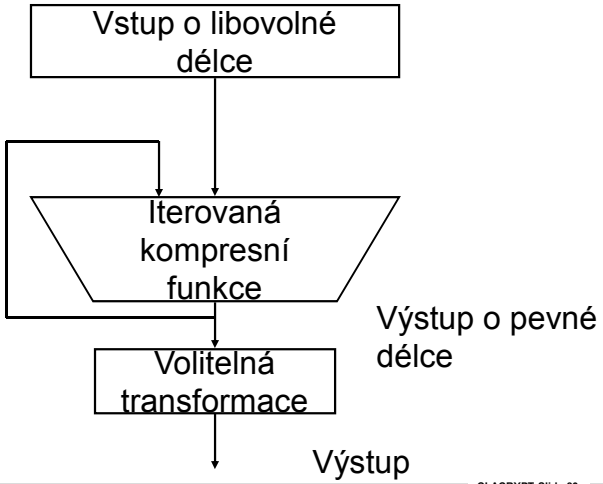
©Petr Hanáček CLACRYPT Slide 81

Vztah mezi vlastnostmi

- collision resistance⇒2nd preimage resistance
- collision resistance nezaručuje preimage resistance
- Pokud funkce h_k je MAC, pak h_k vzhledem k útoku se zvoleným textem (chosen-text attack) je:
  - 2nd preimage a collision resistant
  - preimage resistant

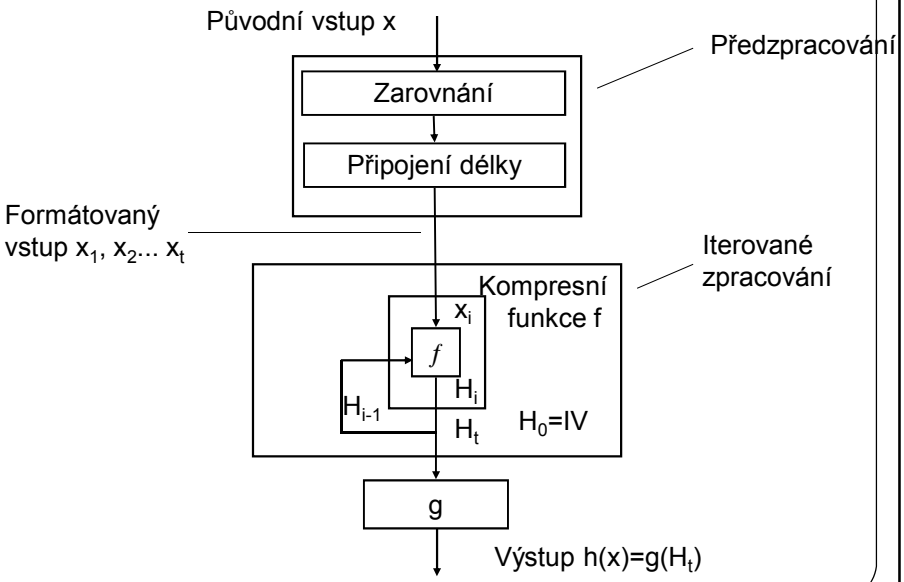
©Petr Hanáček CLACRYPT Slide 82

Obecný model iterované hašovací funkce



©Petr Hanáček CLACRYPT Slide 83

Detailní pohled



©Petr Hanáček CLACRYPT Slide 84

Merklova meta-metoda

- Jakákoli kompresní funkce f odolná proti kolizím se dá rozšířit na CRHF
- Merklova meta-metoda je efektivní způsob jak z f vytvořit CRHF
  - n bitový výstup, r bitová proměnná
  - Pokud existuje kolize pro h, pak to znamená, že vznikla kolize pro f v určitém kole i
  - Vložením délky bloku je zajištěno, že žádný vstup není prefizem jiného vstupu
    - » Merkle-Damgardovo zesílení

©Petr Hanáček CLACRYPT Slide 85

Zarovnání (Padding)

- Nejednoznačné zarovnání (Ambiguous Padding): připoj ke zprávě tolik nulových bitů, aby zpráva byla násobkem délky bloku
- Jednoznačné zarovnání (Unambiguous Padding)
  - Připoj ke zprávě 1
  - Proveď nejednoznačné zarovnání
  - Neintuitivní pro programátora

©Petr Hanáček CLACRYPT Slide 86

Bezpečnostní cíle

Typ funkce	Cíl návrhu	Ideální síla	Cíl útočníka
OWHF	preimage res; 2^{nd}-preimage res	2^n 2^n	Vytvořit preimage Vytvořit 2-preimage
CRHF	collision res	$2^{n/2}$	Vytvořit kolizi
MAC	key non-recovery; computation res	2^n Pf	Nalézt klíč Vytvořit nový MAC

* $Pf=\max(2^{-n}, 2^{-t})$ kde t je délka klíče

©Petr Hanáček CLACRYPT Slide 87

Základní útok

- **Základní útok na haš**
  - **n-bitový neklíčovaný haš má ideální bezpečnost, pokud splňuje požadavky na OWHF a CHRF**
- **Útok silou na klíč MAC (known-text attack),vyžaduje 2^t operací**
- **Uhodnutí MAC – vyžaduje 2^n operací**
- **Předvypočítání haše (memory-time tradeoff)**
- **Paralelizace 2nd-preimage**
- **Útoky na dlouhé zprávy pro 2nd-preimage. Pokud h je iterovaná funkce a nepoužívá se MD zesílení, pak 2nd-preimage může být nalezeno v čase $(2^n/s)+s$, v prostoru $n(s+\log s)$ bitů, pro $1 \le s \le \min(t, 2n/2)$**
  - **Narozeninový útok na mezivýsledky**

©Petr Hanáček CLACRYPT Slide 88

Birthday paradox

- The apparent paradox that, in a room of only 23 people, there is a 50 percent probability that at least two will have the same birthday. The "paradox" is that we have an even chance of success with just 23 of 365 possible days represented.
- Birthday paradox:
  $r1, \ldots, rn \in [0,1,\ldots,B]$ indep. random integers.
  When $n = 1.2\sqrt{B}$ then
  $Pr[\exists i \neq j : r_i = r_j] > \frac{1}{2}$
- msg-digest only 64 bits long $\Rightarrow$
  can find collision in 2^{32} tries.
- Typical digest size = 160 bits. (e.g. SHA-1)
  $\Rightarrow$collision time is 2^{80} tries.

©Petr Hanáček CLACRYPT Slide 89

Příklad Birthday Attack

- Předpokládejme hašovací funkci, která má n bitový výstup
- Útočník vytvoří dokument „přátelská dohoda“ a přibližně $2^{n/2+1}$ sémanticky ekvivalentních verzí
- Podobně útočník vytvoří dokument „nepřátelská dohoda“ a přibližně $2^{n/2+1}$ sémanticky ekvivalentních verzí
- S pravděpodobností ½ bude existovat verze „přátelské dohody“ a „nepřátelské dohody“, které budou mít stejný haš

©Petr Hanáček CLACRYPT Slide 90

Vyžadované délky

- OWHF n>=80
- CHRF n>=160 (birthday attack)
- MAC n>=64 s klíčem alespoň 64 bitů
  - Je vhodné omezit počet pokusů hádání

©Petr Hanáček CLACRYPT Slide 91

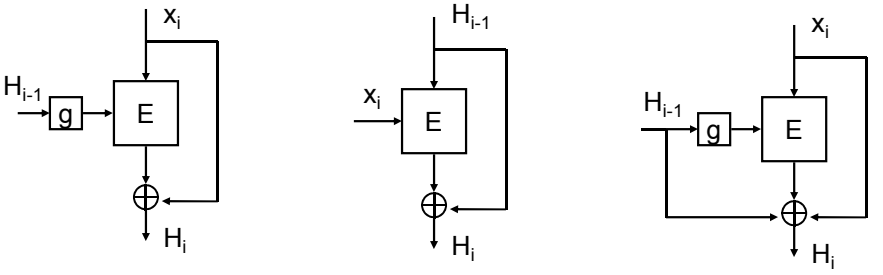
Některé hašovací algoritmy

	SHA-1	MD5 (MD4+)	RIPEMD-160
Velikost výstupu	160 bits	128 bits	160 bits
Základní velkost bloku	512 bits	512 bits	512 bits
Počet kroků	80 (4 rounds of 20)	64 (4 rounds of 16)	160 (5 paired rounds of 16)
Maximální velikost zprávy	2^{64}-1 bits	unlimited	unlimited

©Petr Hanáček CLACRYPT Slide 92

Hašovací funkce z blokové šifry

- Blokové šifry už existují (není třeba je navrhovat)
- Jednoduché (n bitů) nebo dvojité (2n bitů)
  - Jednoduché pro OWHF
  - Dvojité pro CHRF (obvykle n=64, pro odolnost proti kolizím potřebujem 128 bitů)

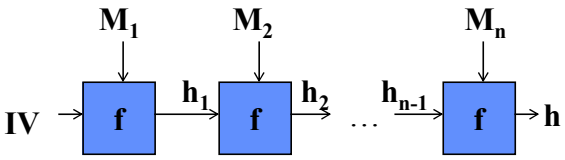


Matyas-Meyer-Oseas | Davies-Meyer | Miyaguchi-Preneel

©Petr Hanáček — CLACRYPT Slide 93

Konstrukce hašovacích algoritmů

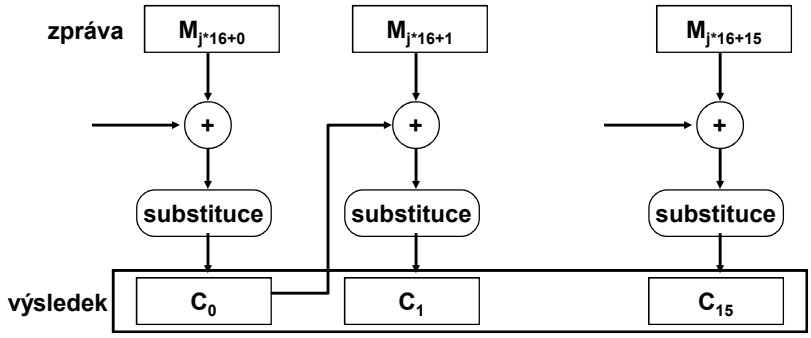
- Jsou obvykle založeny na kompresní funkci f, která pracuje nad bloky M



- Podobné blokovým šifrám v CBC režimu
- Vytvářejí hodnotu haše pro každý blok, která je závislá na hodnotě bloku a hodnotě haše předchozích bloků

©Petr Hanáček — CLACRYPT Slide 94

Algoritmus MD2



- MD2 dává 128-bitovou charakteristiku zprávy
- algoritmus je jednoduchý a poměrně velmi rychlý
- náhodná substituce je tabulka z číslic Ludolfova čísla, takže nemůže obsahovat skrytá vrátka

©Petr Hanáček CLACRYPT Slide 95

MD4

- Navrženo specielně jako hašovací funkce specielně pro softwarové implementace na 32-bitových strojích
- Startovací bod pro MD5, SHA-1 a RIPEMD
- 128 bitový výstup
  - Rozlomena jako CRHF, Dobbertin nalezl kolize pro smysluplné zprávy

©Petr Hanáček CLACRYPT Slide 96

RIPEMD-160

- **Kompresní funkce mapuje 21-slovní vstup (5-slovní stavová proměnná, 16-slovní blok zprávy, 32-bitová slova) na 5-slovní výstup**
- **Více kol než MD-4**
- **Bezpečnost porovnatelná s SHA-1**

©Petr Hanáček CLACRYPT Slide 97

MD5

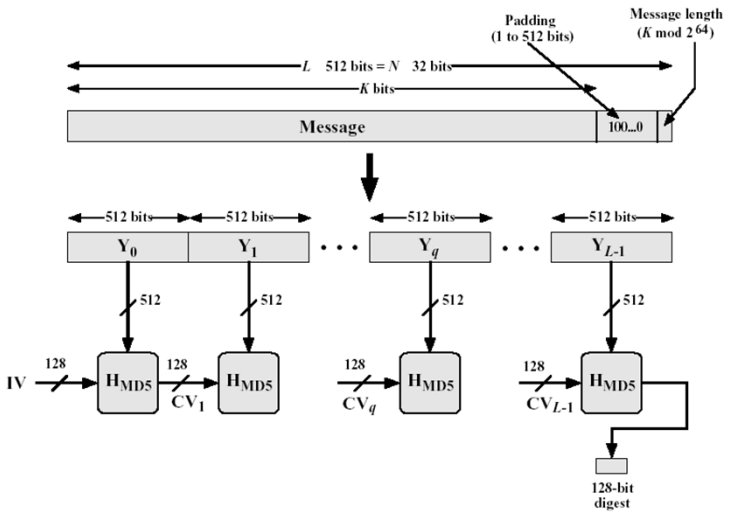
©Petr Hanáček CLACRYPT Slide 98

MD5

- Vyvinul ho Ron Rivest na MIT
- Zpracovává zprávu libovolné délky na haš o délce 128 bitů po blocích o délce 512 bitů
- Zpráva je doplněna na délku
  - » k = 448 mod 512
- Na konec zprávy je přidán 64bitový blok s délkou zprávy. Výsledná délka zprávy je násobkem 512 bitů.
- Detailní popis MD5 je v dokumentu RFC1321.
- Hans Dobbertin ukázal, že MD5 není odolné proti kolizím
- Používá se v IPSec a v jiných protokolech

©Petr Hanáček CLACRYPT Slide 99

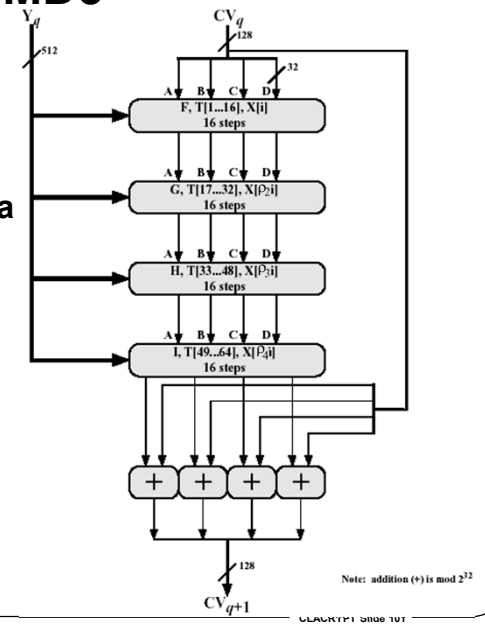
MD5



©Petr Hanáček CLACRYPT Slide 100

MD5

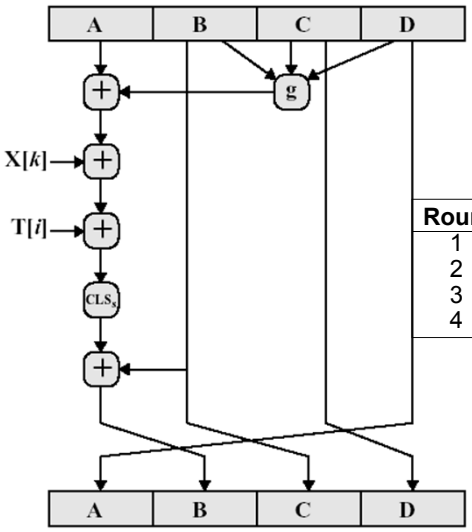
- Zpracování jednoho 512bitového bloku (kompresní funkce)
- F,G,H,I – funkce
- T – konstantní tabulka
- X – vstup Y



©Petr Hanáček

CLACRYPT Slide 101

MD5 – elementární operace



Round	Function g	g(b, c, d)
1	F(b, c, d)	$(b \wedge c) \vee (b' \wedge d)$
2	G(b, c, d)	$(b \wedge d) \vee (c \wedge d')$
3	H(b, c, d)	$b \oplus c \oplus d$
4	I(b c, d)	$c \oplus (b \vee d')$

©Petr Hanáček

CLACRYPT Slide 102

Principle of Most Surprise

- **https://news.ycombinator.com/item?id=9484757**

```
md5('240610708') == md5('QNKCDZO')
```


SHA

Secure Hash Algorithm (SHA)

- SHA byla vytvořena organizací NIST v roce 1993
- Podobná MD5
- Revidována v r. 1995 jako SHA-1
- Revidována v r. 2001 jako SHA-2
  - "SHA-256", "SHA-384", and "SHA-512"

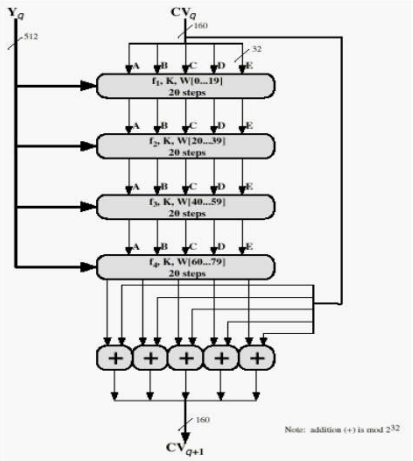
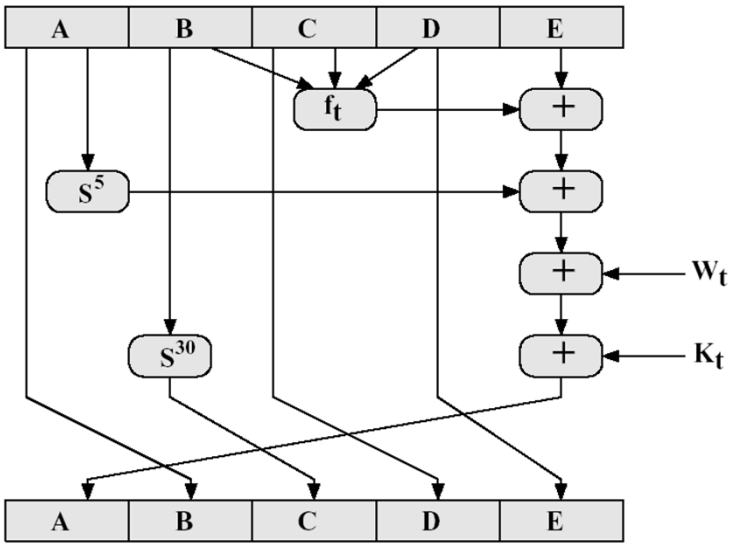


Figure 3.5 SHA-1 Processing of a Single 512-bit Block

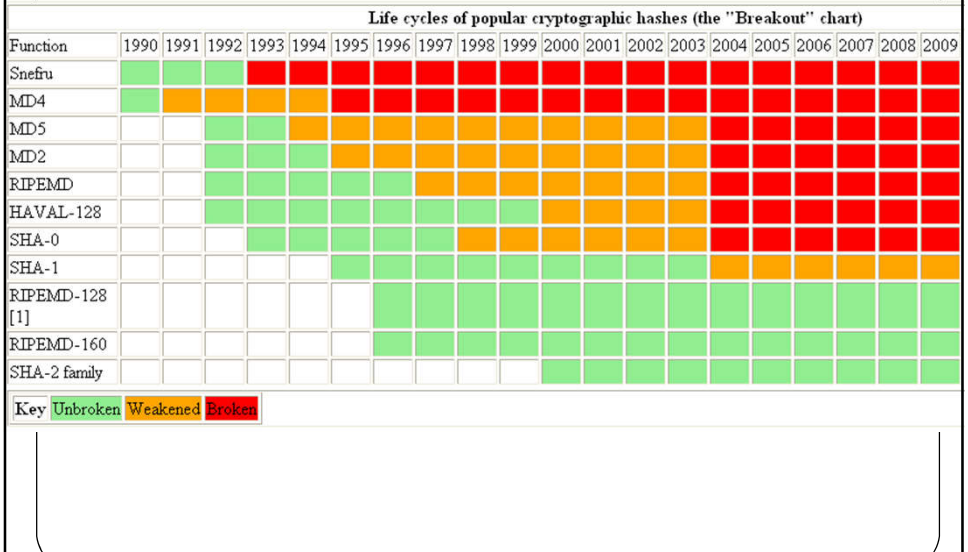
©Petr Hanáček CLACRYPT Slide 105

SHA – Elementární operace



©Petr Hanáček CLACRYPT Slide 106

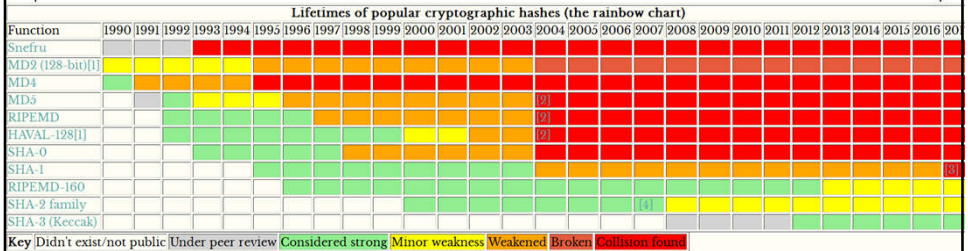
Breakout chart 2009



©Petr Hanáček CLACRYPT Slide 107

Rainbow chart 2017

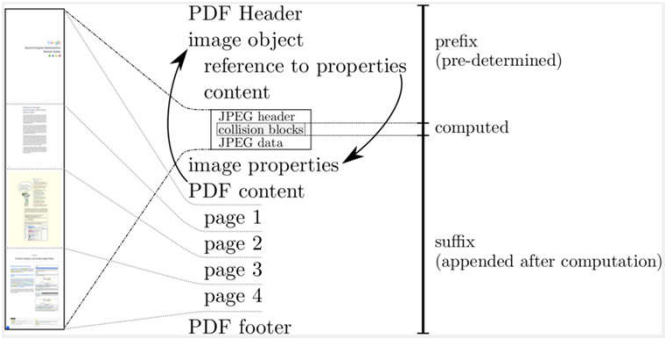
- **http://valerieaurora.org/hash.html**



©Petr Hanáček CLACRYPT Slide 108

23.2.2017 – zemřelo SHA-1

- **Announcing the first SHA1 collision**
  - **https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html**
  - **https://techcrunch.com/2017/02/23/security-researchers-announce-first-practical-sha-1-collision-attack/**
  - **https://shattered.it/static/shattered.pdf**



Soutěž o SHA-3

- **2005-2006: NIST přemýšlí o vyhlášení soutěže na SHA-3**
  - **MD5 a SHA-1 utrpěli těžké rány**
  - **SHA-2 je založen na stejných základech jako MD5 a SHA-1**
  - **Hledáme následníka SHA-2**
- **Říjen 2008: Deadline pro návrhy**
  - **Efektivnější než SHA-2**
  - **Délky výstupů: 224, 256, 384, 512 bitů**
  - **Bezpečnost: collision and (2nd) pre-image resistant**

Soutěž o SHA-3

- První kolo: Říjen 2008 až léto 2009
  - 64 návrhů, 51 přijato
  - 37 prezentováno na první konferenci kandidátů SHA-3 v Leuvenu, únor 2009
  - Mnoho z nich rozbito kryptoanalýzou
  - NIST zúžil výběr na 14 semifinalistů
- Druhé kolo: léto 2009 až podzim 2010
  - Analýzy prezentovány na druhé konferenci kandidátů SHA-3 v Santa Barbaře, srpen 2010
  - NIST zúžil výběr na 5 finalistů
- Třetí kolo: podzim 2010 až říjen 2012
  - Analýzy na třetí konferenci SHA-3 ve Washingtonu, březen 2012
- 2. říjen : NIST oznamuje, že vítězem SHA-3 se stal Keccak

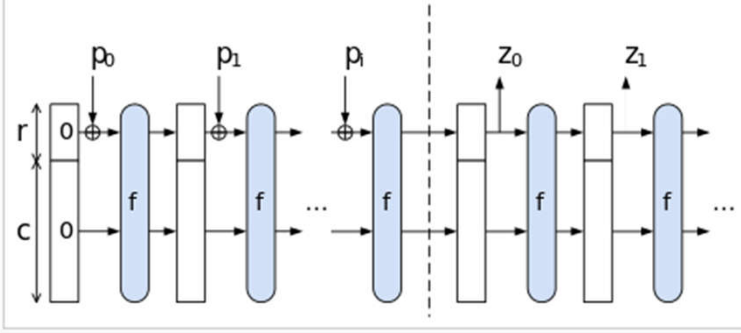
Keccak

- SHA-3 je kryptografická hašovací funkce, která byla určena v soutěži hledající nástupce starších funkcí SHA-1 a SHA-2 a organizované americkým NIST.
- Vítězná funkce byla do soutěže přihlášena pod svým původním jménem Keccak (výslovnost [kɛtʃak]), jejími autory jsou Guido Bertoni, Joan Daemen, Michaël Peeters a Gilles Van Assche
- Ostatní finalisty (i SHA-2) překonává v rychlosti v hardware.
- Při běhu na běžném procesoru Core 2 má rychlost zhruba 13 cyklů na bajt.
- Zcela odlišný princip od SHA-2, což znamená, že průlomový pokrok, který by ohrozil bezpečnost jedné z funkcí, pravděpodobně neohrozí druhou z nich.

•Wikipedia

SHA-3

- **Princip houby (sponge)**



The sponge construction for hash functions. p_i are input, z_i are hashed output. The unused "capacity" c should be twice the desired resistance to collision or preimage attacks.

https://en.wikipedia.org/wiki/SHA-3

MAC
Message Authentication Code

Vlastnosti MAC

- **MAC je rodina funkcí h_k (parametrizovaných tajným klíčem k)**
  - **Snadný výpočet (pokud je k známé)**
  - **Komprese, x má libovolnou délku, $h_k(x)$ má pevnou délku**
  - **Výpočetní bezpečnost, při znalosti páru $(x_i, h_k(x_i))$ je výpočetně nemožné spočíst novou dvojici $(x, h_k(x))$ pro nové $x \neq x_i$**

©Petr Hanáček CLACRYPT Slide 115

Message Authentication Code

- **Message Authentication Code (MAC)**
  - » **MAC=F(Message,Key)**

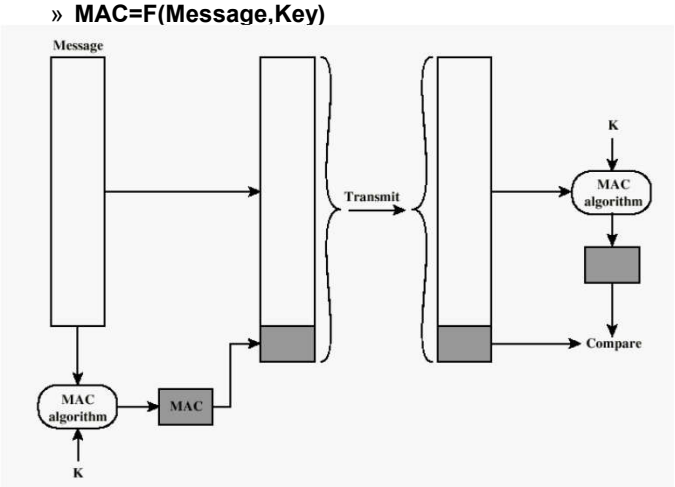
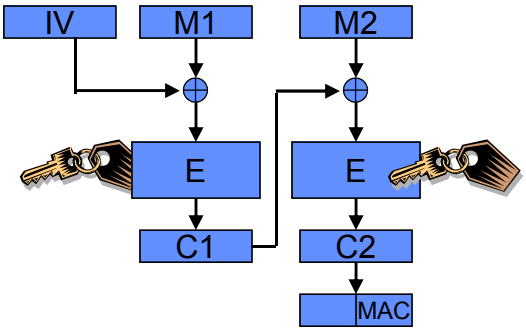


Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)

©Petr Hanáček lide 116

CBC MAC

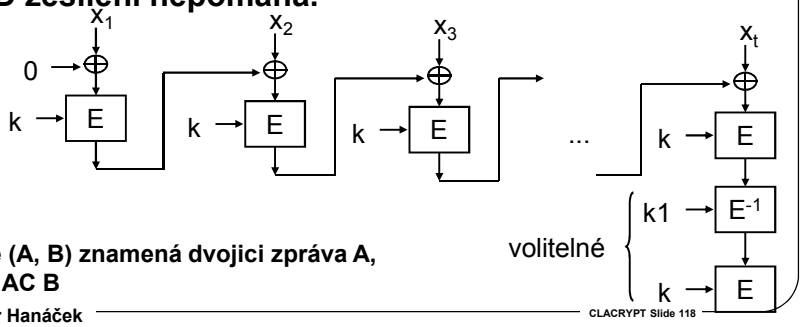


- Typicky 32 bitů z posledního bloku (48, 64)
- Je to dost?

©Petr Hanáček

Bezpečnost CBC-MAC

- Volitelný krok má zabránit útoku chosen-text existential forgery bez ovlivnění předchozích kroků
- Existential forgery: základní CBC-MAC je bezpečný jenom pro zprávy z pevným počtem bloků. Jinak pokud máme dvojice (x1, H1) a (x2, H2) a můžeme požadovat ((x1 || z), M) pak je možné zkonstruovat novou zprávu (x2 || (H1⊕z⊕ H2), M) která je platná. MD zesílení nepomáhá.



Notace (A, B) znamená dvojici zpráva A, a její MAC B

©Petr Hanáček

MAC vytvořené z MDC

- Velmi rozšířená konstrukce (např. IPSec, SSL)
- Tři různé strategie
  - secret prefix
  - secret suffix
  - enveloping

©Petr Hanáček CLACRYPT Slide 119

Secret prefix

- Mějme MDC funkci *h* s kompresní funkcí *f*: H_0=IV, $H_i=f(H_{i-1}, x_i)$, $h(x)= H_t$
- Konstrukce: na začátek zprávy se přidá tajný klíč *k* a MAC je potom M=h(k||x)
- Je zde útok, kdy je možné na konec zprávy přidat *y* a spočíst h(k||x||y) ze znalosti h(k||x) bez znalosti *k* !!
- Ani MD zesílení nepomáhá (i délka x se dá zahrnout do zprávy)
- Stejně tak není bezpečná ani varianta, kdy *k* použijeme jako H_0

©Petr Hanáček CLACRYPT Slide 120

Secret suffix

- MAC hodnoty *x* se spočte jako M=h(x||k)
- Možnost narozeninového útoku, útočník, lterý může zvolit *x* může také vytvořit *x'* pro které h(x)=h(x') se složitostí $O(2^{n/2})$ bez ohledu na délku klíče k
- Útočník tedy může zkonstruovat dvojici (x',M)
- Metoda v podstatě vypočte haš a v konečné fázi ho „zašifruje"
- Není to dobrý způsob

Enveloping

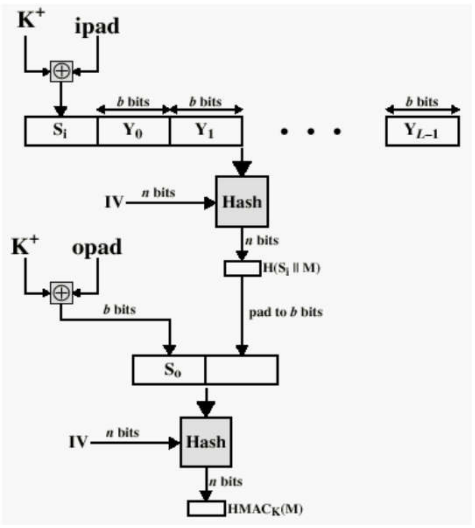
- $h_k(x)=h(k||p||x||k)$
- p je řetězec, použitý pro zarovnání klíče k na délku jednoho bloku
- Lepší než předchozí dvě metody, není to však nejlepší metoda
- Základ pro algoritmus HMAC

Hash Function MAC (HMAC)

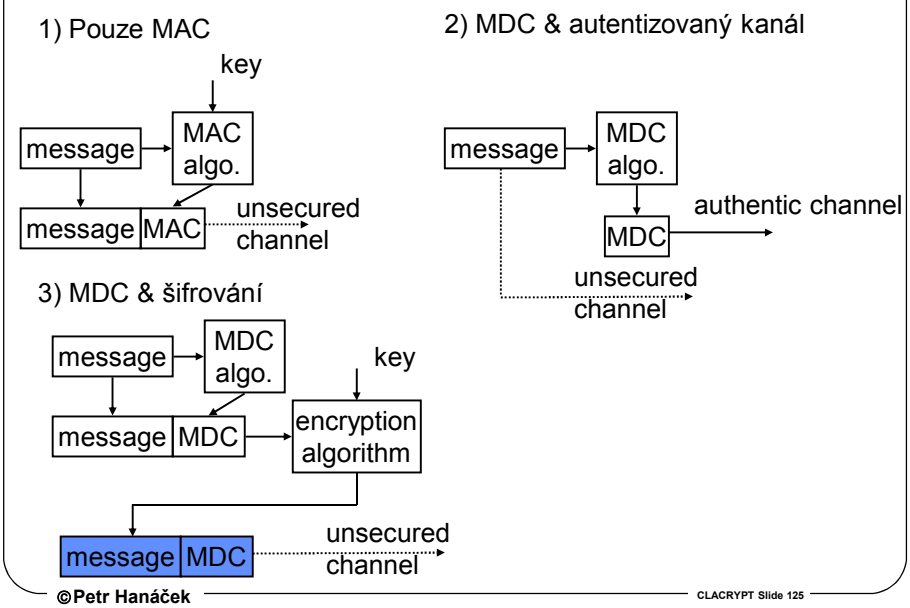
- „Klíčovaný haš“
- Myšlenka: vytvořit MAC z hašovací funkce
  - Dodání klíče
    - » „Přihašování klíče“
- Použití:
  - IPsec
  - Transport Layer Security (TLS)

HMAC

- Spočte se H1 jako haš konkatenace M a K1
- Pro zabránění útoku “dodatečný blok”, se spočte H2 jako haš konkatenace H1 a K2
- K1 a K2 používají polovinu bitů klíče K
- Vymaskování bitů:
  - K^+ = K doplněný nulami
  - Ipad = 00110110 x b/8
  - Opad = 01011100 x b/8



Metody pro integritu dat



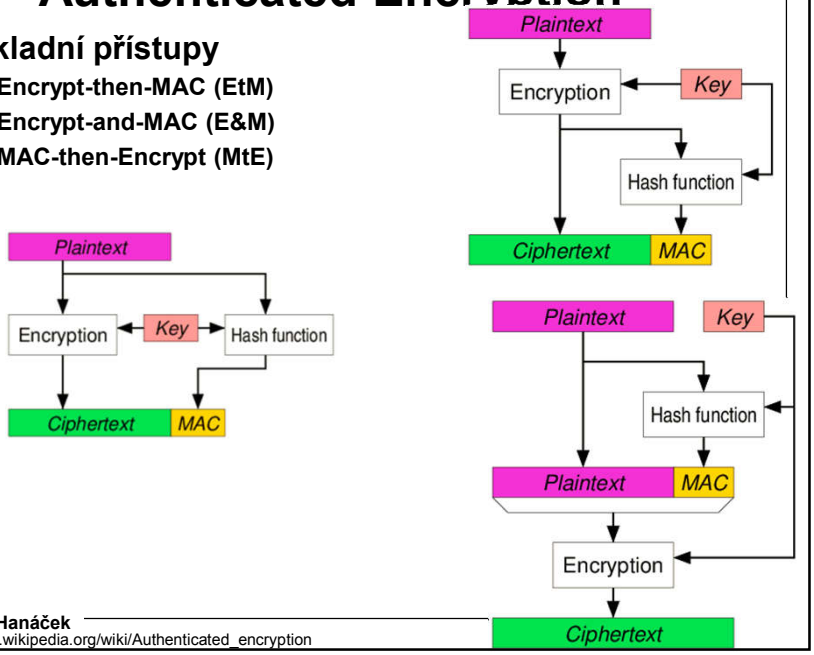
©Petr Hanáček CLACRYPT Slide 125

AE
Authenticated Encryption

©Petr Hanáček CLACRYPT Slide 126

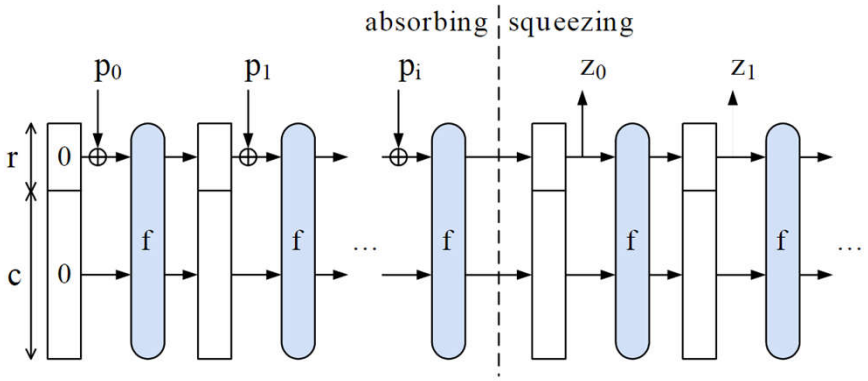
Authenticated Encryption

- **Základní přístupy**
  - **Encrypt-then-MAC (EtM)**
  - **Encrypt-and-MAC (E&M)**
  - **MAC-then-Encrypt (MtE)**



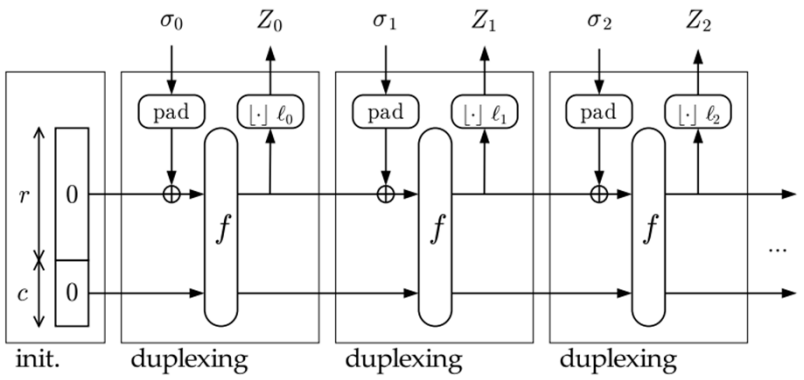
©Petr Hanáček
https://en.wikipedia.org/wiki/Authenticated_encryption

Sponge



©Petr Hanáček
CLACRYPT Slide 128
https://en.wikipedia.org/wiki/Sponge_function

Sponge duplexing



©Petr Hanáček

KONEC

©Petr Hanáček