

MD5 hešovací funkce

a kryptografické hešovací funkce obecně

David Machač

FJFI ČVUT v Praze

29.4.2010

Definice (Hešovací funkce)

Nechť \mathcal{M} je množina řetězců libovolné délky a \mathcal{C}^ℓ množina řetězců délky ℓ . Pak zobrazení

$$h : \mathcal{M} \rightarrow \mathcal{C}^n \quad (1)$$

nazveme hešovací funkce. Obrazu hešovací funkce říkáme heš.

Definice (Jednocestnost (Preimage resistance))

Hešovací funkce se nazývá jednocestná, pokud není (jednoduše a rychle) možné najít $m \in \mathcal{M}$ takové, že $h(m) = N$ pro nějaké $N \in \mathcal{N}^\ell$.

Definice (Jednocestnost (Preimage resistance))

Hešovací funkce se nazývá jednocestná, pokud není (jednoduše a rychle) možné najít $m \in \mathcal{M}$ takové, že $h(m) = N$ pro nějaké $N \in \mathcal{N}^\ell$.

Definice (Slabá odolnost vůči kolizi (Second preimage resistance))

Hešovací funkce je slabě odolná vůči kolizi, pokud k zadanému $m \in \mathcal{M}$ není (jednoduše a rychle) možné nalézt $m' \in \mathcal{M}$ takové, že $h(m') = h(m)$.

Definice (Jednocestnost (Preimage resistance))

Hešovací funkce se nazývá jednocestná, pokud není (jednoduše a rychle) možné najít $m \in \mathcal{M}$ takové, že $h(m) = N$ pro nějaké $N \in \mathcal{N}^\ell$.

Definice (Slabá odolnost vůči kolizi (Second preimage resistance))

Hešovací funkce je slabě odolná vůči kolizi, pokud k zadanému $m \in \mathcal{M}$ není (jednoduše a rychle) možné nalézt $m' \in \mathcal{M}$ takové, že $h(m') = h(m)$.

Definice (Silná odolnost vůči kolizi (Collision resistance))

Hešovací funkce je silně odolná vůči kolizi, pokud není (jednoduše a rychle) možné nalézt dva různé vstupní řetězce $m, m' \in \mathcal{M}$ takové, že $h(m) = h(m')$.

Definice (Kryptografická hešovací funkce)

Hešovací funkce se nazývá kryptografická, pokud je jednocestná a zároveň slabě nebo silně odolná vůči kolizi.

Definice (Kryptografická hešovací funkce)

Hešovací funkce se nazývá kryptografická, pokud je jednocestná a zároveň slabě nebo silně odolná vůči kolizi.

Příklad (matematická hádanka)

Alice tvrdí, že vyřešila složitý matematický problém. Bob by ho chtěl zkusit vyřešit sám, ale zároveň chce mít jistotu, že Alice neblafuje. Proto mu Alice dá heš jejího řešení, takže Bob může poté co problém sám vyřeší ověřit, že Alice mluvila pravdu.

- Jak dlouhý má být heš?

- Jak dlouhý má být heš?
- odpověď poskytuje tzv. narozeninový útok

- Jak dlouhý má být heš?
- odpověď poskytuje tzv. narozeninový útok
- \mathcal{A} - aspoň dva lidé mají narozeniny ve stejný den

$$P(\mathcal{A}) = 1 - P(\bar{\mathcal{A}}) = 1 - \frac{365!}{(365 - n)!} \cdot \frac{1}{365^n}$$

- Jak dlouhý má být heš?
- odpověď poskytuje tzv. narozeninový útok
- \mathcal{A} - aspoň dva lidé mají narozeniny ve stejný den

$$P(\mathcal{A}) = 1 - P(\bar{\mathcal{A}}) = 1 - \frac{365!}{(365 - n)!} \cdot \frac{1}{365^n}$$

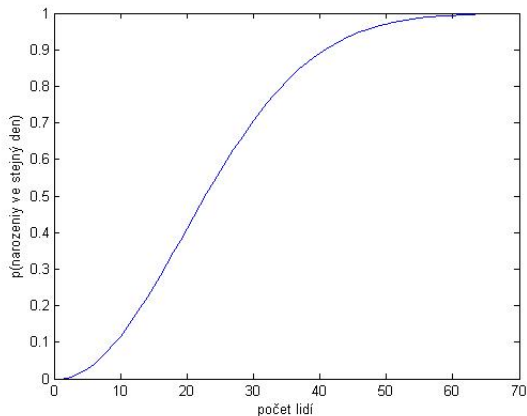
- $P(\mathcal{A}) = 1$ když $n > 365$

- Jak dlouhý má být heš?
- odpověď poskytuje tzv. narozeninový útok
- \mathcal{A} - aspoň dva lidé mají narozeniny ve stejný den

$$P(\mathcal{A}) = 1 - P(\bar{\mathcal{A}}) = 1 - \frac{365!}{(365 - n)!} \cdot \frac{1}{365^n}$$

- $P(\mathcal{A}) = 1$ když $n > 365$
- $P(\mathcal{A}) = 0.508$ když $n = 23$

Narozeninový útok



Obrázek: Pravděpodobnost, že mezi n lidmi najdu dva s narozeninami ve stejný den

- Pokud vyberu jednoho člověka, a chci najít dalšího s narozeninami ve stejný den s poloviční pravděpodobností, potřebuji skupinu $\lceil 365/2 \rceil = 183$ lidí

- Pokud vyberu jednoho člověka, a chci najít dalšího s narozeninami ve stejný den s poloviční pravděpodobností, potřebuji skupinu $\lceil 365/2 \rceil = 183$ lidí
- V terminologii hešových funkcí: dosáhnout silné odolnosti vůči kolizi je mnohem těžší než dosáhnout slabé odolnosti vůči kolizi

- Pokud vyberu jednoho člověka, a chci najít dalšího s narozeninami ve stejný den s poloviční pravděpodobností, potřebuji skupinu $\lceil 365/2 \rceil = 183$ lidí
- V terminologii hešových funkcí: dosáhnout silné odolnosti vůči kolizi je mnohem těžší než dosáhnout slabé odolnosti vůči kolizi
- pokud máme 64-bitový heš, můžeme dosáhnout přibližně $1,8 \times 10^{19}$ výstupů, pak potřebujeme $5,1 \times 10^9$ pokusů k nalezení kolize

- Pokud vyberu jednoho člověka, a chci najít dalšího s narozeninami ve stejný den s poloviční pravděpodobností, potřebuji skupinu $\lceil 365/2 \rceil = 183$ lidí
- V terminologii hešových funkcí: dosáhnout silné odolnosti vůči kolizi je mnohem těžší než dosáhnout slabé odolnosti vůči kolizi
- pokud máme 64-bitový heš, můžeme dosáhnout přibližně $1,8 \times 10^{19}$ výstupů, pak potřebujeme $5,1 \times 10^9$ pokusů k nalezení kolize
- pro n -bitový heš: $2^{n/2}$

- Většina používaných KHF je založena na Merkleho–Damgårdově konstrukci
- Ralph Merkle (US) a Ivan Damgård (DK) nezávisle, CRYPTO '89
- Hešovací funkce musí ze zprávy libovolné délky vytvořit zprávu konstantní délky
- Zprávu m délky q rozdělíme na n bloků stejné délky a na ně použijeme *jednocestnou kompresní funkci*
- často bloková šifra - ze dvou řetězců délky ℓ a b dostaneme jeden délky ℓ

$$f: \mathcal{M}^{\ell+b} \rightarrow \mathcal{C}^{\ell} \quad (2)$$

- klíč+otevřený text -> ŠIFROVÝ TEXT

- iterovaná hešovací funkce h vznikne opakovanou aplikací funkce f na všech n bloků zprávy m
- ℓ bývá většinou 128 bitů a b 512 bitů
- první blok zprávy m_2 zašifrujeme klíčem H_0
- vzniklý šifrový text budeme brát jako klíč H_1 , který použijeme na m_2 atd.
- na poslední výstup H_n se častou používá další šifrovací funkce g .

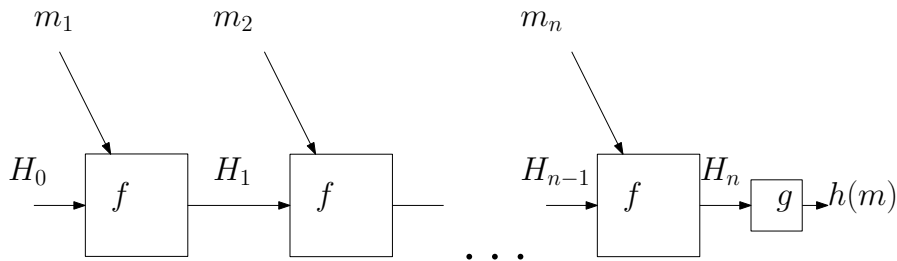
Merkleho–Damgårdova konstrukce KHF

- iterovaná hešovací funkce h vznikne opakovanou aplikací funkce f na všech n bloků zprávy m
- ℓ bývá většinou 128 bitů a b 512 bitů
- první blok zprávy m_2 zašifrujeme klíčem H_0
- vzniklý šifrový text budeme brát jako klíč H_1 , který použijeme na m_2 atd.
- na poslední výstup H_n se častou používá další šifrovací funkce g .

Schéma

$$\begin{aligned}H_0 &= pp, \\ H_i &= f(H_{i-1}, m_i), \quad i \in \hat{n}, \\ h(m) &= g(H_n)\end{aligned}$$

Merkleho–Damgårdova konstrukce KHF



Obrázek: Merkleho–Damgårdova konstrukce KHF

Doplnění zprávy tak, aby $b|q$

Příklad (8 bajtů)

Doplnění zprávy

HashInpu t

na $8 \times n$ bajtů:

HashInpu t00000000.

- Merkle: doplníme na konec zprávy její délku

Příklad

HashInpu t1000000 00000009.

Věta

Silná odolnost vůči kolizi se přenáší z f na h .

- Ověřování, zda-li byla zpráva nezměněna (stejný heš před i po přenosu)
- Ukládání hesel - útočník nemůže zjistit heslo, hash je mu k ničemu
- Ověřování integrity dat - např. v p2p sítích
- Hešovací tabulky

- *Message Digest*
- Ron Rivest, 1992
- využívá Merkleho–Damgårdovu konstrukci
- $l = 128$ bitů, $b = 512$ bitů
- vylepšená verze MD4 - praktický jediný rozdíl jsou 4 průchody hešovacího algoritmu místo 3 (o 30% výpočetně náročnější).

- mějme zprávu m délky s bitů

- mějme zprávu m délky s bitů
- vytvořme n -rozměrný vektor M , jehož prvky představují 32-bitová slova a $n \equiv 0 \pmod{16}$

- mějme zprávu m délky s bitů
- vytvoříme n -rozměrný vektor M , jehož prvky představují 32-bitová slova a $n \equiv 0 \pmod{16}$
- když spojíme prvky M , rovná se délka tohoto řetězce násobku 512 bitů

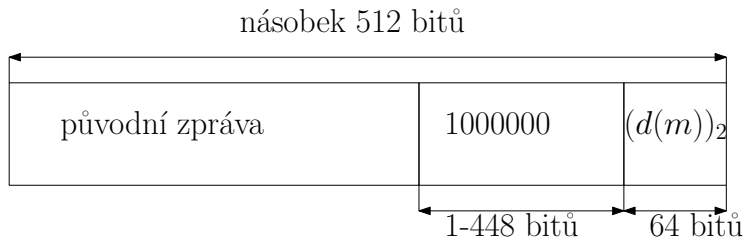
- mějme zprávu m délky s bitů
- vytvoříme n -rozměrný vektor M , jehož prvky představují 32-bitová slova a $n \equiv 0 \pmod{16}$
- když spojíme prvky M , rovná se délka tohoto řetězce násobku 512 bitů
- tvorba M : doplníme m tak, aby

$$\text{délka } M \equiv 448 \pmod{512}$$

- mějme zprávu m délky s bitů
- vytvoříme n -rozměrný vektor M , jehož prvky představují 32-bitová slova a $n \equiv 0 \pmod{16}$
- když spojíme prvky M , rovná se délka tohoto řetězce násobku 512 bitů
- tvorba M : doplníme m tak, aby

$$\text{délka } M \equiv 448 \pmod{512}$$

- zbývajících 64 bitů použijeme k zápisu délky m



Obrázek: Zpráva M , na kterou použijeme MD5

vytvoř M

$A := 0x67452301$

$B := 0xEFCDAB89$

$C := 0x98BADCFE$

$D := 0x10325476$

for $i := 0$ to $n/16 - 1$ **do**

for $j := 0$ to 15 **do**

$X[j] := M[i \cdot 16 + j]$

end for

$(A..D)' := (A..D)$

 ALG1..4

$(A..D) := (A..D) + (A..D)'$

end for

$h(m) := A || B || C || D$

1. $A \leftarrow (A + g(B, C, D) + X[1] + T[17]) \leftarrow 5$
2. $D \leftarrow (D + g(A, B, C) + X[6] + T[18]) \leftarrow 9$
3. $C \leftarrow (C + g(D, A, B) + X[11] + T[19]) \leftarrow 14$
4. $B \leftarrow (B + g(C, D, A) + X[0] + T[20]) \leftarrow 20$
5. $A \leftarrow (A + g(B, C, D) + X[5] + T[21]) \leftarrow 5$
6. $D \leftarrow (D + g(A, B, C) + X[10] + T[22]) \leftarrow 9$
7. $C \leftarrow (C + g(D, A, B) + X[15] + T[23]) \leftarrow 14$
8. $B \leftarrow (B + g(C, D, A) + X[4] + T[24]) \leftarrow 20$
9. $A \leftarrow (A + g(B, C, D) + X[9] + T[25]) \leftarrow 5$
10. $D \leftarrow (D + g(A, B, C) + X[14] + T[26]) \leftarrow 9$
11. $C \leftarrow (C + g(D, A, B) + X[3] + T[27]) \leftarrow 14$
12. $B \leftarrow (B + g(C, D, A) + X[8] + T[28]) \leftarrow 20$
13. $A \leftarrow (A + g(B, C, D) + X[13] + T[29]) \leftarrow 5$
14. $D \leftarrow (D + g(A, B, C) + X[2] + T[30]) \leftarrow 9$
15. $C \leftarrow (C + g(D, A, B) + X[7] + T[31]) \leftarrow 14$
16. $B \leftarrow (B + g(C, D, A) + X[12] + T[32]) \leftarrow 20$

Obrázek: Algoritmus ALG2

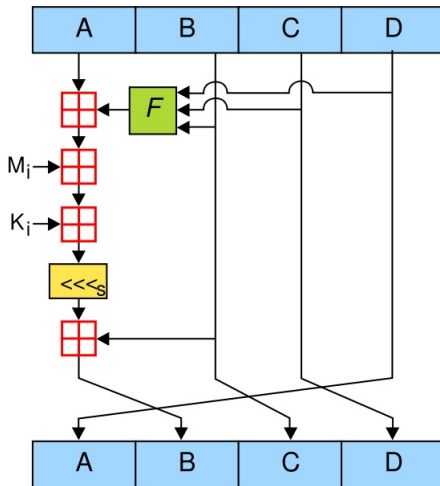
- AND, OR, XOR, NOT, sčítání modulo 2^{32} , posun
- jediná aritmetická operace je sčítání modulo 2^{32}
- $g(X, Y, Z) = ((X \wedge Z) \vee (X \wedge (\neg Z)))$
- $T[j] = \lfloor 4\,294\,967\,296 \cdot |\sin j| \rfloor, j \in \hat{64}$

MD5 - hešovací algoritmus: pravdivostní tabulka

X	Y	Z	f	g	h	i
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

Obrázek: Pravdivostní tabulka funkcí použitých v MD5

MD5 - hešovací algoritmus: přehled



Obrázek: Jedna MD5 operace se skládá z 64 zobrazených operací (4x16)

- MD5("The quick brown fox jumps over the lazy dog") = 9e107d9d372bb6826bd81d3542a419d6
- MD5("The quick brown fox jumps over the lazy dog.") = e4d909c290d0fb1ca068ffaddf22cbd0

- MD5 je dnes z bezpečnostního hlediska historie

- MD5 je dnes z bezpečnostního hlediska historie
- nalezení kolize: složitost $2^{24,1}$

- MD5 je dnes z bezpečnostního hlediska historie
- nalezení kolize: složitost $2^{24,1}$
- 3/2005 - A. Lenstra a X. Wang popsali způsob, jak vytvořit dva certifikáty X.509 se stejným md5 hešem

- MD5 je dnes z bezpečnostního hlediska historie
- nalezení kolize: složitost $2^{24,1}$
- 3/2005 - A. Lenstra a X. Wang popsali způsob, jak vytvořit dva certifikáty X.509 se stejným md5 hešem
- 3/2005 - V. Klíma představil algoritmus, který zvládne totéž na obyčejném PC

- MD5 je dnes z bezpečnostního hlediska historie
- nalezení kolize: složitost $2^{24,1}$
- 3/2005 - A. Lenstra a X. Wang popsali způsob, jak vytvořit dva certifikáty X.509 se stejným md5 hešem
- 3/2005 - V. Klíma představil algoritmus, který zvládne totéž na obyčejném PC
- 12/2008 - A. Sotirov, M. Stevens a kol. pomocí kolizí vytvořili falešný certifikát SSL, pomocí kterého je možné vytvořit libovolný jiný

- MD5 je dnes z bezpečnostního hlediska historie
- nalezení kolize: složitost $2^{24,1}$
- 3/2005 - A. Lenstra a X. Wang popsali způsob, jak vytvořit dva certifikáty X.509 se stejným md5 hešem
- 3/2005 - V. Klíma představil algoritmus, který zvládne totéž na obyčejném PC
- 12/2008 - A. Sotirov, M. Stevens a kol. pomocí kolizí vytvořili falešný certifikát SSL, pomocí kterého je možné vytvořit libovolný jiný
- 4/2009 - Y. Sasaki a K. Aoki prolomili jednocestnost (pouze teoreticky, složitost $2^{123,4}$)

- GOST

- GOST
- kolize: 2^{105} , jednocestnost: 2^{192}

- GOST
- kolize: 2^{105} , jednocestnost: 2^{192}
- heš: 256 bitů, 32 průběhů

- GOST
- kolize: 2^{105} , jednocestnost: 2^{192}
- heš: 256 bitů, 32 průběhů

- GOST
- kolize: 2^{105} , jednocestnost: 2^{192}
- heš: 256 bitů, 32 průběhů
- Tiger (TTH)

- GOST
- kolize: 2^{105} , jednocestnost: 2^{192}
- heš: 256 bitů, 32 průběhů
- Tiger (TTH)
- kolize: 2^{62} , jednocestnost: 2^{184}

- GOST
- kolize: 2^{105} , jednocestnost: 2^{192}
- heš: 256 bitů, 32 průběhů
- Tiger (TTH)
- kolize: 2^{62} , jednocestnost: 2^{184}
- heš: 192, 128, 160 bitů, 24 průběhů

-  Rolf Oppliger, *Contemporary Cryptography*. Artech House, Londýn, 2005.
-  M.M.J Stevens, *On Collisions for MD5*. Eindhoven University of Technology, 2007.

Děkuji za pozornost