Hašovací funkce

Jan Legerský

jan.legersky@gmail.com

Theoretical Informatics GRoup

UKRY

29. dubna 2013

Hašovací funkce

Hašovací funkce $f : \{0,1\}^N \rightarrow \{0,1\}^n$ je zobrazení přiřazující zprávě M libovolné délky N řetězec pevně dané délky n, která má následující vlastnosti [3]:

Hašovací funkce

Hašovací funkce $f : \{0,1\}^N \to \{0,1\}^n$ je zobrazení přiřazující zprávě M libovolné délky N řetězec pevně dané délky n, která má následující vlastnosti [3]:

- Snadný výpočet $f(M)$
  - Čas $\mathcal{O}(N)$, paměť $\mathcal{O}(1)$.

Hašovací funkce

Hašovací funkce $f : \{0,1\}^N \to \{0,1\}^n$ je zobrazení přiřazující zprávě M libovolné délky N řetězec pevně dané délky n, která má následující vlastnosti [3]:

- Snadný výpočet $f(M)$
  - Čas $\mathcal{O}(N)$, paměť $\mathcal{O}(1)$.
- Odolnost proti kolizím
  - Je výpočetně nemožné nalézt různé zprávy M a M' takové, že $f(M) = f(M')$.

Hašovací funkce

Hašovací funkce $f : \{0,1\}^N \to \{0,1\}^n$ je zobrazení přiřazující zprávě M libovolné délky N řetězec pevně dané délky n, která má následující vlastnosti [3]:

- Snadný výpočet $f(M)$
  - Čas $\mathcal{O}(N)$, paměť $\mathcal{O}(1)$.
- Odolnost proti kolizím
  - Je výpočetně nemožné nalézt různé zprávy M a M' takové, že $f(M) = f(M')$.
- Odolnost proti nalezení 2. vzoru
  - K dané M_{target} je výpočetně nemožné nalézt zprávu M takovou, že $f(M) = f(M_{\text{target}})$.

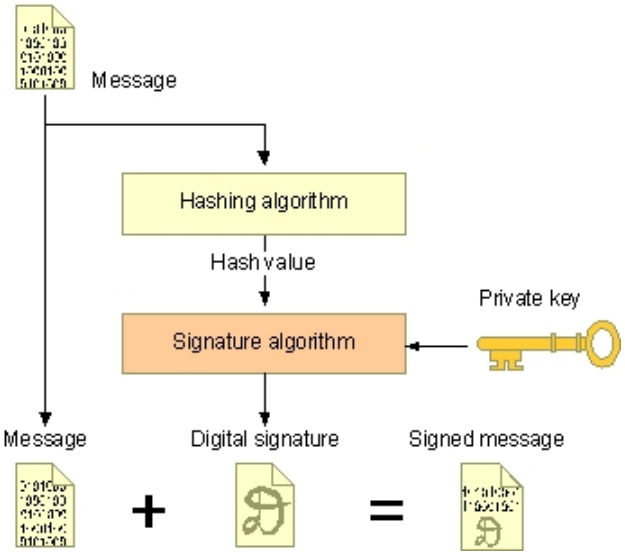
Hašovací funkce

Hašovací funkce $f : \{0,1\}^N \to \{0,1\}^n$ je zobrazení přiřazující zprávě M libovolné délky N řetězec pevně dané délky n, která má následující vlastnosti [3]:

- Snadný výpočet $f(M)$
  - Čas $\mathcal{O}(N)$, paměť $\mathcal{O}(1)$.
- Odolnost proti kolizím
  - Je výpočetně nemožné nalézt různé zprávy M a M' takové, že $f(M) = f(M')$.
- Odolnost proti nalezení 2. vzoru
  - K dané M_{target} je výpočetně nemožné nalézt zprávu M takovou, že $f(M) = f(M_{\text{target}})$.
- Odolnost proti hledání vzoru
  - K dané haši h_{target} je výpočetně nemožné nalézt zprávu M takovou, že $f(M_{\text{target}}) = h_{\text{target}}$.

Použití hašovacích funkcí

- Kontrola integrity dat
- Autentizace dat – HMAC
- Digitální podpisy
- Ověřování a ukládání hesel
- Identifikace dat nebo souborů
- Hašovací tabulky
- Generátory pseudonáhodných čísel
- Prokazování znalosti nebo autorství
- Derivace klíčů



Příklady hašovacích funkcí: MD5, SHA-1, SHA-512, Tiger, ...

Konstrukce

Merkleovo-Damgårdovo paradigma – iterativní konstrukce pomocí kompresní funkce F.

Haš $f(M)$ zprávy M spočteme následovně [6, 5]:

1. Doplníme zprávu M na bloky velikosti m:

$$M_1 M_2 \dots M_{l-1} M'_l ||1||00\dots0||\text{length}(M)\,.$$

2. Iterujeme l-krát kompresní funkci $F(h_i, M_i) : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$:

$$h_0 = IV\,,$$
$$h_i = F(h_{i-1}, M_i)\,.$$

3. Získáme otisk:

$$f(M) := g(h_l)\,.$$

Jako g se často používá identita.

Kompresní funkce

MD konstrukce a kompresní funkce odolná proti kolizím implikuje hašovací funkci odolnou proti kolizím.

Davies-Meyerova konstrukce využívá blokovou šifru $E_k(x)$:

$$F(h, M) := E_M(h) \oplus h\,.$$

Typy útoků

- Útok na kolizi
- Útok na k-kolizi
- Útok na druhý vzor ke zprávě délky 2^k

Útok na kolizi

- Pomocí narozeninového paradoxu – $2^{\frac{n}{2}}$ zpráv

Útok na kolizi

- Pomocí narozeninového paradoxu – $2^{\frac{n}{2}}$ zpráv
  Musíme vybrat l prvků z množiny velikosti N, abychom měli pravděpodobnost p, že dojde ke kolizi:

Útok na kolizi

- Pomocí narozeninového paradoxu – $2^{\frac{n}{2}}$ zpráv
  Musíme vybrat l prvků z množiny velikosti N, abychom měli pravděpodobnost p, že dojde ke kolizi:

$$l \doteq \sqrt{2 \ln(\frac{1}{1-p})\sqrt{N}}$$

Útok na kolizi

- Pomocí narozeninového paradoxu – $2^{\frac{n}{2}}$ zpráv
  Musíme vybrat l prvků z množiny velikosti N, abychom měli pravděpodobnost p, že dojde ke kolizi:

$$l \doteq \sqrt{2 \ln(\frac{1}{1-p})\sqrt{N}}$$

- Prolomení kompresní funkce
  - 17. 8. 2004 – Xiaoyun Wang, Dengguo Feng, Xuejia Lai a Hongbo Yu našli kolize MD5.
  - 18.3. 2006 – Vlastimil Klíma publikoval algoritmus hledající kolize MD5 během 1 minuty – tunelování.

Útok na multikolizi

k-kolize je k zpráv $M^{(1)}, M^{(2)}, \ldots, M^{(k)}$ takových, že

$$f(M^{(1)}) = f(M^{(2)}) = \cdots = f(M^{(s)}).$$

Útok na multikolizi

k-kolize je k zpráv $M^{(1)}, M^{(2)}, \ldots, M^{(k)}$ takových, že

$$f(M^{(1)}) = f(M^{(2)}) = \cdots = f(M^{(s)}).$$

- Teoreticky [7] – $(k!)^{\frac{1}{k}} 2^{\frac{n(k-1)}{k}}$ volání kompresní funkce.

Útok na multikolizi

k-kolize je k zpráv $M^{(1)}, M^{(2)}, \ldots, M^{(k)}$ takových, že

$$f(M^{(1)}) = f(M^{(2)}) = \cdots = f(M^{(s)}).$$

- Teoreticky [7] – $(k!)^{\frac{1}{k}} 2^{\frac{n(k-1)}{k}}$ volání kompresní funkce.
- Jouxův útok [4] – $\lceil \log k \rceil \cdot 2^{\frac{n}{2}}$

Jouxův útok ukazuje, že zřetězením dvou haší se nezlepší odolnost podle očekávání ($n_g 2^{n_f/2} + 2^{n_g/2}$ místo $2^{\frac{n_g+n_f}{2}}$).

Útok na druhý vzor

- Teoreticky

2^n volání kompresní funkce.

Útok na druhý vzor

- Teoreticky

$$2^n \text{ volání kompresní funkce.}$$

- Útok pomocí rozšiřitelné zprávy

$$2^{\frac{n}{2}+1} + 2^{n-k}, \text{ případně } k \cdot 2^{\frac{n}{2}+1} + 2^{n-k}$$

Útok na druhý vzor

- Teoreticky

$$2^n \text{ volání kompresní funkce.}$$

- Útok pomocí rozšiřitelné zprávy

$$2^{\frac{n}{2}+1} + 2^{n-k}, \text{ případně } k \cdot 2^{\frac{n}{2}+1} + 2^{n-k}$$

- Kolizní strom hloubky l

$$4\sqrt{2 \log 2} \cdot \sqrt{l} \cdot 2^{\frac{n+l}{2}} + 2^{n-k} + 2^{n-l}$$

Ditherování

Nekonečná slova

Nechť $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ je konečná abeceda písmen. Pak posloupnost $\mathbf{d} = d_1 d_2 d_3 \ldots$, kde $d_i \in \mathcal{A}$, nazveme nekonečným slovem nad abecedou $\mathcal{A}$.

Vlastnosti nekonečných slov

Slovo $\mathbf{d}$ je square-free, pokud neobsahuje žádný faktor u ve tvaru $u = ww$, kde w je libovolný neprázdný faktor slova $\mathbf{d}$.

Faktorovou komplexitou $C_{\mathbf{d}}(n) : \mathbb{N} \to \mathbb{N}$ slova $\mathbf{d}$ nazveme funkci přiřazující každému n počet faktorů slova $\mathbf{d}$ délky n.

Ditherování

Vylepšení odolnosti hašovací funkce proti útoku na druhý vzor.
Při každé iteraci přidáváme do vstupu kompresní funkce písmeno d_i nekonečného slova $\mathbf{d}$:

$$h_i = F(h_{i-1}, M_i, d_i)\,.$$

- Square-free slovo $\mathbf{d}$ znemožní útok pomocí rozšiřitelné zprávy.
- Slovo s vysokou komplexitou nebo velkou abecedou znesnadní další útoky.

Útoky na ditherované hašovací funkce

- Upravený kolizní strom hloubky l
  - $4\sqrt{2\ln 2}\cdot\sqrt{l}\cdot 2^{\frac{n+l}{2}} + C_{\mathbf{d}}(l+1)\cdot 2^{n-k} + 2^{n-l}$
- Kite generátor
  - offline fáze $|\mathcal{A}|\cdot(1+\sqrt{2})\cdot\sqrt{\ln 2}\sqrt{n-k}\cdot 2^{n}$
  - online fáze $2^{\frac{n-k}{2}}$
- Hellmanovy tabulky
  - offline fáze $|\mathcal{A}|\cdot 2^{n}$
  - online fáze $2^{\frac{2(n-k)}{3}}$

Oprava složitosti konstrukce kite generátoru

Kite generátor – struktura 2^{n-k} haší propojených vzájemnými kolizemi.
Složitost konstrukce uváděná autory v článku [1]:

$$\mathcal{O}(|\mathcal{A}| \cdot 2^n)\,,$$

kde $\mathcal{A}$ je abeceda ditherovací posloupnosti $\mathbf{d}$.
Námi vypočtená minimální složitost offline fáze pomocí teorie náhodných grafů:

$$|\mathcal{A}| \cdot (1+\sqrt{2}) \cdot \sqrt{\ln 2}\sqrt{n-k} \cdot 2^n\,.$$

Nevhodnost xorování ditherovací posloupnosti na zprávu

Aumasson a Phan v článku [2] navrhují blok zprávy a ditherovací písmeno xorovat na sebe:

$$F_d^{\oplus}(h_{i-1}, M_i, d_i) = F(h_{i-1}, M_i \oplus d_i)\,. \tag{1}$$

Nalezli jsme útok na druhý vzor se stejnou složitostí jako bez použití ditherovací posloupnosti:

1. Upravení zprávy pomocí ditherovací posloupnosti
2. Útok na klasickou konstrukci
3. Zpětná transformace nalezeného druhého vzoru pomocí ditherovací posloupnosti

Hledání multikolizí LAB módu

Xigen Yao navrhuje používat místo ditherovací posloupnosti předchozí blok zprávy (*Locking Abutting Blocks*):

$$h_i = F_{LAB}(h_{i-1}, M_{i-1}, M_i)\,.$$

Tvrdí, že tato konstrukce je odolná proti hledání multikolizí.

Nalezli jsme ale analogii Jouxova útoku, která má pouze dvojnásobnou složitost oproti klasické konstrukci.

- Složení dvou volání kompresní funkce.
- Pevné sudé bloky.

Vhodná ditherovací posloupnost

Umíme zkonstruovat:

- Square-free slovo nad abecedou $\{0, 1, 2\}$.
- Slovo s exponenciální komplexitou nad abecedou $\{0, 1\}$.
- Square-free slovo s exponenciální komplexitou nad malou abecedou.

Potřebujeme:

- Square-free slovo s exponenciální komplexitou s lepším koeficientem.
- Může být i nad větší abecedou.

Děkuji za pozornost.

Zdroje I


E. Andreeva, Ch. Bouillaguet, P.-A. Fouque, J. J. Hoch, J. Kelsey, A. Shamir, and S. Zimmer.
Second Preimage Attacks on Dithered Hash Functions.
Advances in Cryptology – EUROCRYPT 2008, Lecture Notes in Computer Science 4965, pages 270–288, 2008.


Jean-Philippe Aumasson and Raphael C.-W. Phan.
How (not) to efficiently dither blockcipher-based hash functions?
In *Progress in Cryptology – AFRICACRYPT 2008*, volume 5023 of *Lecture Notes in Computer Science*, pages 308–324. Springer Berlin Heidelberg, 2008.

Zdroje II


Menezes A. J., van Oorschot P. C., and Vanstone S. A.
Handbook of Applied Cryptography.
CRC Press, 1996.


Antoine Joux.
Multicollisions in iterated hash functions. application to cascaded constructions.
In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 306–316. Springer, 2004.

Zdroje III


John Kelsey and Bruce Schneier.
Second Preimages on n-bit Hash Functions for Much Less than 2^n Work.


Rivest R. L.
Abelian Square-free Dithering for Iterated Hash Functions.
presented at ECrypt Hash Function Workshop, June 21, 2005, Cracow, and at the Cryptographic Hash workshop, November 1, 2005, Gaithersburg, Maryland (August 2005), 2005.

Zdroje IV

Kazuhiro Suzuki, Dongvu Tonien, Kaoru Kurosawa, and Koji Toyota.
Birthday paradox for multi-collisions.
In *Information Security and Cryptology – ICISC 2006*, volume 4296 of *Lecture Notes in Computer Science*, pages 29–40. Springer Berlin Heidelberg, 2006.