Hašovací funkce

Milan Radojčić

SSPŠaG – KBB

15. března 2026

Hlavní body

Hlavní body

Hašovací funkce

- Obecně hašovací funkce produkují pro jakkoliv dlouhý vstup výstup dané délky.
- Od kryptografických hašovacích funkcí navíc vyžadujeme
  - **jednosměrnost**[1] – pro výstup y je těžké najít vstup x, aby $h(x) = y$,
  - **bezkoliznost**[2] – je obtížné najít dvě zprávy m, m', aby $h(m) = h(m')$.
- Ne všechny hašovací funkce jsou určeny pro kryptografické účely.
  - např. CRC nebo hašovací funkce používané pro hašovací tabulky

Poznámka k názvosloví

Pokud pro nějakou funkci h platí, že $h(x) = y$, tak x nazýváme **vzorem** a y **obrazem**.

V angličtine se používají výrazy *preimage* a *image*.

[1] angl. (first) preimage resistance
[2] angl. collision resistance

Návrh jednoduché hašovací funkce

- Chceme navrhnout jednoduchou funkci pro kontrolu integrity.
- Spolu se zprávou m pošleme krátký kontrolní součet $k = h(m)$.
  - S tímto může druhá strana ověřit, že se zpráva nezměnila kvůli nějaké chybě.
- Jak to ale uděláme?

Návrh jednoduché hašovací funkce

- Jeden z nejjednodušších způsobů, jak tuto funkci realizovat, je pomocí modulární aritmetiky.
- $h(m) = m \mod n$, kde $n \in \mathbb{N}$ můžeme zvolit.
- Jsou všechna n stejně dobrá?
  - Co kdyby bylo n mocnina dvojky?

Návrh jednoduché hašovací funkce

- Kdyby například $n = 8$, tak v některých případech mají zprávy stejný hash i při změně jen jednoho bitu.
  - Například $h(11011) = h(10011)$.
- Na volbě n záleží efektivita naší funkce.
- Na velice podobném principu, jako naše vymyšlená funkce, funguje i CRC.

Hlavní body

Narozeninový paradox

Kolik náhodných lidí se musíme zeptat na jejich narozeniny, aby jsme měli aspoň 50% šanci, že alespoň dva z nich budou mít narozeniny ve stejný den?

Narozeninový paradox

Mějme krabičku s m různýma kuličkama. Pravděpodobnost, že poté co vytáhneme n kuliček, tak jsme vytáhli alespoň dvě stejné je

$$1 - \frac{m \cdot (m-1) \cdot (m-2) \cdot \ldots \cdot (m-n+1)}{m^n}$$

Narozeninový paradox

Kolik náhodných lidí se musíme zeptat na jejich narozeniny, než se nám stane, že alespoň dva z nich budou mít narozeniny ve stejný den?

Odpověď je jen 23 lidí.

$$1 - \frac{365 \cdot 364 \cdot 363 \cdots 343}{365^{23}} \approx 0.507$$

Kolize 1. a 2. řádu

Definice

Kolize 1. řádu znamená nalezení dvou zpráv M a M' takových, že $h(M) = h(M')$.[3]

Definice

Kolize 2. řádu znamená nalezení M' pro danou zprávu M takovou, že $h(M) = h(M')$.

Pokud je výpočetně nemožné najít kolizi 1. (nebo 2.) řádu, tak o funkci říkáme, že je **odolná proti kolizím 1. (nebo 2.) řádu**.

[3] V obou případech předpokládáme, že $M \neq M'$.

Kolize 2. řádu

- Odolnosti proti kolizím 2. řádu se někdy říká *odolnost vůči modifikaci vzoru*.
  - angl. *second preimage resistance*

Cvičení

V jakém vztahu je odolnost proti kolizím 2. řádu s jednosměrností? Dokážeme říct, že jedno je prokazatelně těžší?

Cvičení

V jakém vztahu je odolnost proti kolizím 1. a 2. řádu?

Bezkoliznost

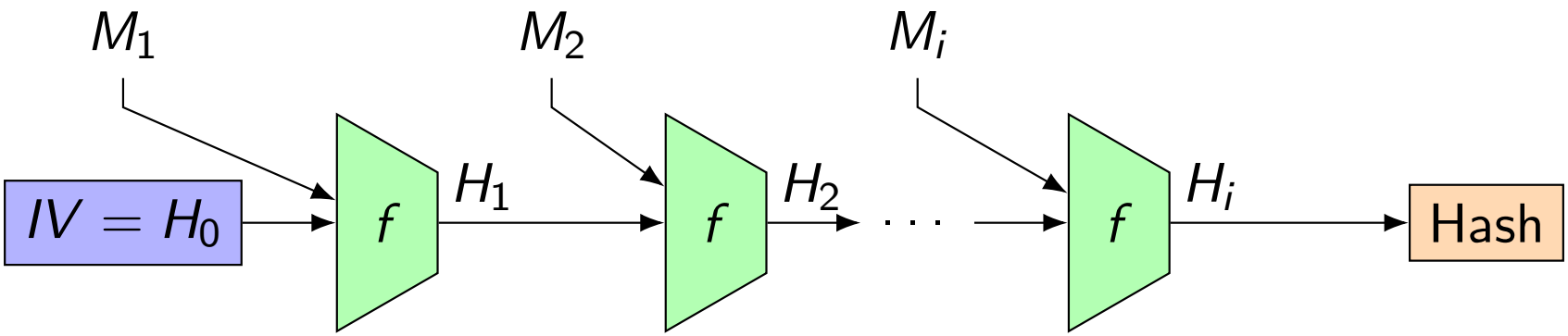
Pokud je funkce dobře navržená a nejde jednoduše vypočítat její inverze tak potřebujeme spočítat

- $2^{n/2}$ vstupů, abychom měli 50% šanci, že najdeme kolizi 1. řádu a
- 2^n vstupů, abychom nalezli kolizi 2. řádu, kde n je počet bitů výstupu.

Hlavní body

Merkleova–Damgårdova konstrukce

- Obecná struktura pro konstrukci hašovacích funkcí.
- Byla použita např. v MD4, MD5, SHA1 a SHA2.
- Je založená na postupném opakování **kompresní funkce** f na bloky dat a vnitřní stav.



- M_i jsou jednotlivé bloky vstupu M, tedy $M = M_1 M_2 \ldots M_n$.

Kompresní funkce

- Pro kompresní funkci platí $H_i = f(H_{i-1}, M_i)$.
- Lze ukázat, že pokud je kompresní funkce bezkolizní, tak bude celá hašovací funkce bezkolizní.
- Jednosměrnou kompresní funkci může například tvořit symetrická šifra: $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$.
  - Tomuhle se říká *Davies-Mayerova* konstrukce.

Padding

Cvičení

Představte si, že chceme zprávu 0101 doplnit na 8 bitů. Uděláme to tak, že přidáme samé 0 a získáme 01010000.

Jaký je zde problém?

- V praxi chceme tedy **zakódovat do paddingu délku originální zprávy**.
  - Tomuhle se také říká *Merkleovo-Damgårdovo zesílení*.

Hlavní body

MD4

- Hašovací funkce publikovaná roku 1990.
- Navrhl ji Ronald Rivest.
- V současné době není považována za bezpečnou.
- Je založena na Merkleově–Damgårdově konstrukci.
  - Stačí nám tedy popsat jen kompresní funkci f.

MD4: kompresní funkce

- MD4 používá následující logické funkce:
  - $f(u, v, w) = (u \wedge v) \vee ((\neg u) \wedge w)$
  - $g(u, v, w) = (u \wedge v) \vee (u \wedge w) \vee (v \wedge w)$
  - $h(u, v, w) = u \oplus v \oplus w$
- Dále stav je reprezentován jako 4 32-bitová slova.
  - Stav $H = (H_1, H_2, H_3, H_4)$.
  - Vnitřní stav má tedy 128 bitů a tím pádem i podpis bude mít 128 bitů.

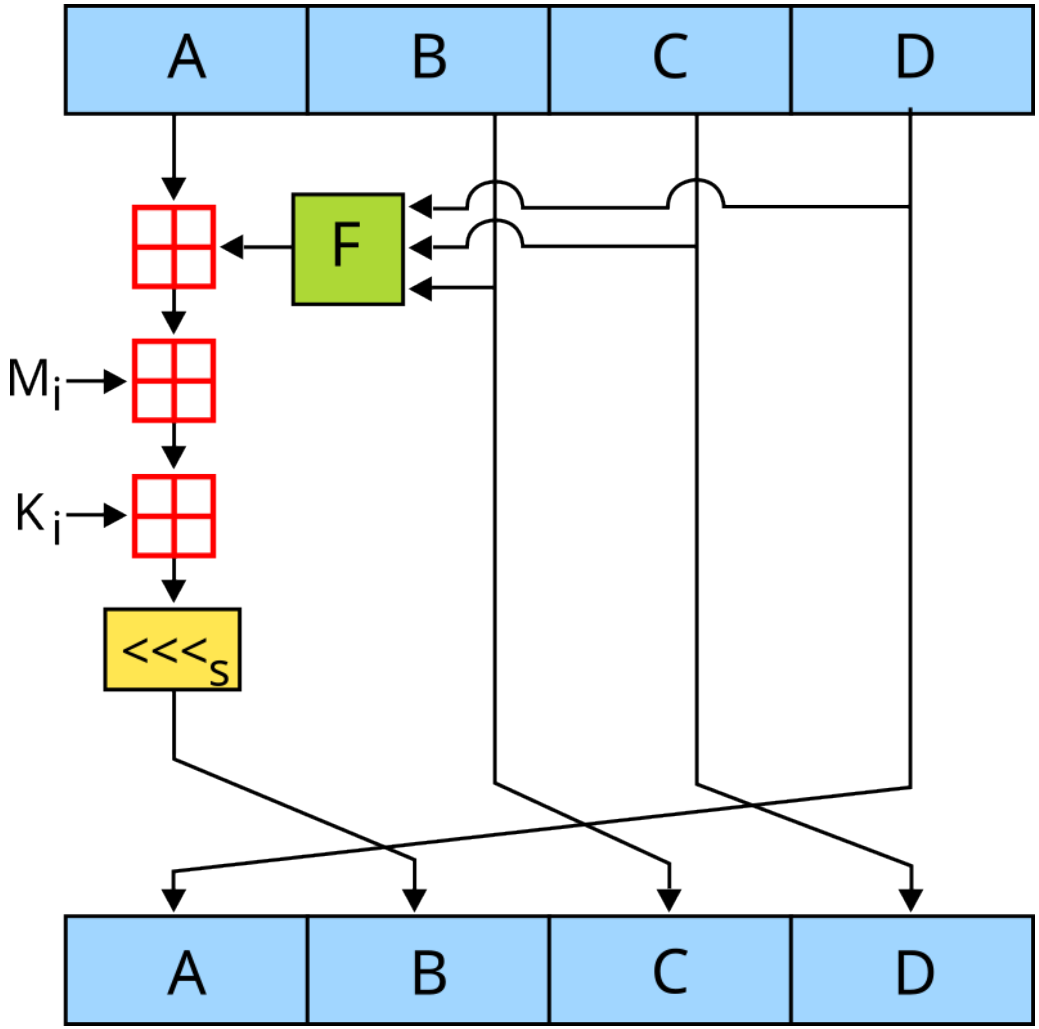
MD4: kompresní funkce

- y_i, z_i, w_i jsou konstanty, které závisí na rundě i.
- Kromě vnitřního stavu vstupuje do kompresní funkce i blok dat.
  - MD4 pracuje s bloky o velikosti 512 bitů.
  - Ty rozdělíme na 16 slov o velikosti 32 bitů.
  - X_0 bude značit prvních 32 bitů, X_1 druhých 32 bitů atd.

MD4: kompresní funkce

1: $(A, B, C, D) \leftarrow (H_1, H_2, H_3, H_4)$
2: **for** $j \leftarrow 0$ až 15 **do** ▷ 1. runda
3: $\quad t \leftarrow A + f(B, C, D) + X_{z_j} + y_j$
4: $\quad (A, B, C, D) \leftarrow (D, t \lll w_j, B, C)$
5: **end for**
6: **for** $j \leftarrow 16$ až 31 **do** ▷ 2. runda
7: $\quad t \leftarrow A + g(B, C, D) + X_{z_j} + y_j$
8: $\quad (A, B, C, D) \leftarrow (D, t \lll w_j, B, C)$
9: **end for**
10: **for** $j \leftarrow 32$ až 47 **do** ▷ 3. runda
11: $\quad t \leftarrow A + h(B, C, D) + X_{z_j} + y_j$
12: $\quad (A, B, C, D) \leftarrow (D, t \lll w_j, B, C)$
13: **end for**
14: $(H_1, H_2, H_3, H_4) \leftarrow (H_1 + A, H_2 + B, H_3 + C, H_4 + D)$

MD4: kompresní funkce



https://en.wikipedia.org/wiki/File:MD4.svg

Hlavní body

Využití hashů

1. ukládání hesel
2. integrita
3. digitální podpisy, autentizace
4. proof-of-work systémy
5. odvozování klíčů
6. hašovací tabulky

Příklady používaných hashů

Kryptografické

- SHA-2, SHA-3
- `bcrypt`

Nekryptografické

- CRC
- různé funkce pro hašovací tabulky