1 HAŠOVACIE (JEDNOCESTÉ) FUNKCIE

1.1 ÚVOD

Kryptografické hašovacie funkcie patria medzi základné stavebné bloky moderných kryptografických algoritmov. Niektoré ich vlastnosti sú koncepčne porovnateľné s funkciami používanými na integritu údajov (napr. CRC kódmi) – t.j. mapovanie veľkej množiny dát (správy) na podstatne menšiu množinu (hašovacia hodnota, otlačok, ...), ktorá umožňuje kontrolovať integritu pôvodnej veľkej množiny dát. Na kryptografické hašovacie funkcie (ďalej len hašovacie funkcie) sú však kladené aj ďalšie dodatočné požiadavky ako **jednocestnosť** (praktická nemožnosť z hašovanej hodnoty určiť pôvodnú správu), **odolnosť proti kolíziám** (praktická nemožnosť[1] nájsť dve rovnaké správy, ktoré poskytujú zhodnú hašovaciu hodnotu) a samozrejme **jednoduchosť výpočtu** hašovacej funkcie.

Existuje množstvo rôznych hašovacích funkcií – napr. MD2, MD4, MD5, SHA, SHA-1, SHA-256, SHA-384, SHA-512, SNERFU-128, SNERFU-256, RIPEMD, RIPEMD-128, RIPEMD-256 a pod. Hašovacie funkcie sú v kryptografickej praxi používané predovšetkým v schémach digitálneho podpisu [1], existujú však aj iné aplikácie hašovacích funkcií – napr. v kryptograficky bezpečných generátoroch pseudonáhodných čísel [1], [2]. V ďalšej časti opíšeme hašovaciu funkciu SHA-1 a jej použitie na generovanie pseudonáhodných čísel.

1.2 HAŠOVACIA FUNKCIA SHA-1

Funkcia SHA-1 (Secure Hash Algorithm) vychádza z funkcie MD4 a bola navrhnutá Americkým úradom (National Institute of Standards and Technology – NIST) pre použitie vo vládnych inštitúciách. Pôvodný návrh funkcie SHA [3] bol v roku 1995 modifikovaný[2] na SHA-1 [4], [5] a v súčasnosti je SHA-1 považovaná za bezpečnú hašovaciu funkciu a široko akceptovaná aj mimo USA.

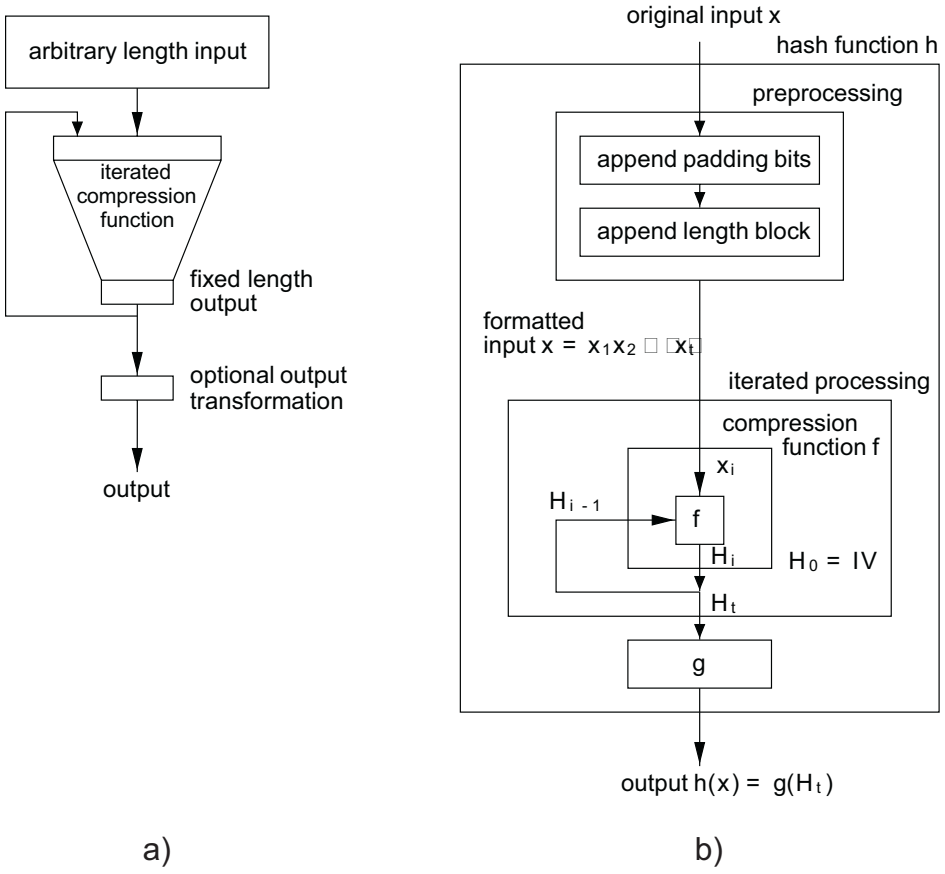
---

[1] Vzhľadom na uvedený princíp vytvárania hašovacej hodnoty (t.j. mapovanie veľkej množiny dát na podstatne menšiu množinu), existuje veľké množstvo kolízií (čím dlhšia správa, tým existuje viac možných kolízii). Pod praktickou nemožnosťou rozumieme vlastnosť, že pri náhodnom výbere správ a použití n-bitovej hašovacej hodnoty je pravdepodobnosť náhodného výberu dvoch správ s rovnakou hašovacou hodnotou rovná 2^{-n}. Samozrejmou požiadavkou je aj skutočnosť, že neexistuje efektívnejšia metóda hľadania správa spôsobujúcich kolízie ako náhodné prehľadávanie!

[2] Zmena v algoritme SHA-1 je veľmi malá. V hlavnej slučke algoritmu bola pridaná jedno-bitová rotácia. Z pohľadu bezpečnosti však táto zmena má veľký význam. V čase publikovania [4] nebola táto zmena vôbec komentovaná a až o niekoľko rokov neskôr nezávislí krytológovia publikovali, že táto zmena zvýšila odolnosť pôvodnej SHA voči kolíziám. Pôvodná SHA teda nie je bezpečná.

1.2.1 ITERAČNÉ HAŠOVACIE FUNKCIE

SHA-1 patrí medzi tzv. iteračné hašovacie funkcie, ktorých základný model je uvedený na obr.1 [6].



Obr.1 Všeobecný model iteračnej hašovacej funkcie a) základný princíp b) detailná štruktúra

Vstup hašovacej funkcie x má prakticky ľubovoľnú dĺžku[3]. Vstupná správa x je v procese predspracovania (preprocessing) rozdelená na r-bitové bloky x_i. Toto predspracovanie zahrňuje aj pripojenie dodatočných bitov (padding) tak, aby celková dĺžka správy bola presným násobkom dĺžky bloku r. V tomto kroku sa tiež doplňuje informácia o dĺžke pôvodnej správy x. Každý r-bitový blok x_i je privedený v iteračnom procese na vstup iteračnej hašovacej funkcie f, ktorá vypočíta v každom iteračnom kroku výstupnú n-bitovú hodnotu H_i (tzv. zreťazená premenná). Hodnota H_i je funkciou predchádzajúcej výstupnej hodnoty H_{i-1} a aktuálnej vstupnej hodnoty x_i. Iteračné spracovanie správy $x = x_1 x_2 \ldots x_t$ je možné zapísať v tvare

[3] Konkrétne hašovacie funkcie majú obmedzenú maximálnu dĺžku vstupnej správy. V prípade SHA-1 je maximálna dĺžka správy $2^{64} - 1$ bitov.

$$\begin{aligned} &H_0 = IV; \\ &H_i = f\left(H_{i-1}, x_i\right), \quad 1 \le i \le t; \\ &h(x) = g\left(H_t\right) \end{aligned} \tag{1.1}$$

pričom IV je preddefinovaná štartovacia hodnota. Voliteľná výstupná transformácia g môže byť využitá na mapovanie n-bitovej premennej H_t na m-bitovú výstupnú hodnotu $g\left(H_t\right)$. Veľmi často je $g\left(H_t\right) = H_t$.

1.2.2 ITERAČNÉ ROVNICE SHA-1

ALGORITMUS: Secure Hash Algorithm – SHA-1
VSTUP: bitový reťazec x dĺžky $0 \le b < 2^{64}$
VÝSTUP: 160-bitová hašovacia hodnota správy x

1. **Definícia konštánt**
   Definujme päť 32-bitových IV (prvé štyri sú z algoritmu MD4)

$$\begin{aligned} h_1 &= 0x67452301 \\ h_2 &= 0xefcdab89 \\ h_3 &= 0x98badcfe \\ h_4 &= 0x10325476 \\ h_5 &= 0xc3d2e1f0 \end{aligned}$$

   a 32-bitové rundové aditívne konštanty

$$\begin{aligned} y_1 &= 0x5a827999 \\ y_2 &= 0x6ed9eba1 \\ y_3 &= 0x8f1bbcdc \\ y_4 &= 0xca62c1d6 \end{aligned}$$

2. **Predspracovanie**
   Doplňme x tak, že bitová dĺžka je násobkom 512 podľa nasledujúceho pravidla. Najskôr doplňme bit 1 a za ním $r-1$ (≥ 0) bitov 0 pre najmenšiu možnú hodnotu r tak, aby celková bitová dĺžka správy s doplnenými bitmi bola celočíselným násobkom 512 zmenšeným o 64. Nakoniec doplňme 64-bitovú hodnotu, ktorá vyjadruje hodnotu $b \bmod 2^{64}$. Túto 64-bitovú hodnotu doplňme ako dve 32-bitové hodnoty, pričom menej významové slovo je doplnené ako prvé. Nech m je počet 512-bitových blokov vo výslednom bitovom reťazci (t.j. platí $b + r + 64 = 512m = 32 \cdot 16m$). Upravený vstup sa tak skladá z $16m$ 32-bitových slov $x_0, x_1 \ldots x_{16m-1}$. Inicializujme zreťazené premenné

$$\left(H_1, H_2, H_3, H_4, H_5\right) \leftarrow \left(h_1, h_2, h_3, h_4, h_5\right)$$

3. Iteračná slučka

Pre $i = 0,1\ldots m-1$ skopírujme i-ty blok šestnástich 32-bitových slov do dočasných premenných $X[j] \leftarrow x_{16i+j}$, $0 \le j \le 15$ a tieto premenné spracujme v nasledujúcich štyroch 20-krokových rundách:
Expandujme 16-slovný blok na 80-slovný blok, označme $X_j = X[j]$ a vykonajme[4]

$$X_j \leftarrow \left(\left(X_{j-3} \oplus X_{j-8} \oplus X_{j-14} \oplus X_{j-16} \right) <<< 1 \right), \quad j = 16,17\ldots 79$$

Inicializujme pracovné premenné A, B, C, D, E

$$(A, B, C, D, E) \leftarrow (H_1, H_2, H_3, H_4, H_5)$$

a vykonajme nasledujúce rundy[5]
Runda 1 $j = 0,1\ldots 19$ **:**

$$t \leftarrow \left((A <<< 5) + f(B, C, D) + E + X_j + y_1 \right)$$
$$(A, B, C, D, E) \leftarrow (t, A, B <<< 30, C, D)$$

Runda 2 $j = 20,21\ldots 39$ **:**

$$t \leftarrow \left((A <<< 5) + h(B, C, D) + E + X_j + y_2 \right)$$
$$(A, B, C, D, E) \leftarrow (t, A, B <<< 30, C, D)$$

Runda 3 $j = 40,41\ldots 59$ **:**

$$t \leftarrow \left((A <<< 5) + g(B, C, D) + E + X_j + y_3 \right)$$
$$(A, B, C, D, E) \leftarrow (t, A, B <<< 30, C, D)$$

Runda 4 $j = 60,61\ldots 79$ **:**

$$t \leftarrow \left((A <<< 5) + h(B, C, D) + E + X_j + y_4 \right)$$
$$(A, B, C, D, E) \leftarrow (t, A, B <<< 30, C, D)$$

Zmena zreťazených premenných

$$(H_1, H_2, H_3, H_4, H_5) \leftarrow (H_1 + A, H_2 + B, H_3 + C, H_4 + D, H_5 + E)$$

[4] Označenie $X <<< s$ označuje cyklickú rotáciu slova W o s bitov doľava.
[5] Iteračná slučka využíva funkcie $f(u,v,w) = u \times v + \overline{u} \times w$, $g(u,v,w) = u \times v + u \times w + v \times w$, $h(u,v,w) = u \oplus v \oplus w$, pričom $\times, +, \oplus$ sú operácie logického súčinu, súčtu resp. xor operácie vykonávané po jednotlivých bitoch.

4. Výstup

Výstupná 160-bitová hašovaná hodnota je (operátor $X \| Y$ označuje spojenie dvoch slov):

$$H_1 \| H_2 \| H_3 \| H_4 \| H_5$$

1.2.3 FUNKCIE SHA-256, SHA-384, SHA-512

Aj keď funkcia SHA-1 je široko používaná v praxi, s príchodom nového šifrovacieho štandaru AES - Rijndael [7],[8] zrejme dôjde k širšiemu využívaniu funkcií SHA-256, SHA-384, SHA-512 , ktoré poskytujú výstupnú hašovaciu hodnotu, ktorá má 256, 384 resp. 512 bitov. Základným dôvodom je poskytnutie porovnateľnej zložitosti prelomenia[6] hašovacích funkcií so zložitosťou prelomenia jednotlivých módov algoritmu AES s využitím lúštenia hrubou silou. Táto skutočnosť je dokumentovaná v nasledujúcej tabuľke [10].

Kombinácia Rijndael-yyy a SHA-xxx	Zložitosť lúštenia hrubou silou	Zložitosť hľadania kolízií
Rijndael-128 a SHA-256	2^{127}	$2^{128,5}$
Rijndael-198 a SHA-384	2^{191}	$2^{192,5}$
Rijndael-256 a SHA-512	2^{255}	$2^{256,5}$

1.3 PRÍKLADY POUŽITIA SHA-1

1.3.1 DIGITÁLNY PODPIS

Existuje niekoľko prakticky využiteľných schém digitálneho podpisu [6]. V prípade ich praktického využitia sa namiesto podpisu pôvodnej digitálnej správy x, ktorá môže mať prakticky neobmedzenú dĺžku podpisuje podstatne kratšia hašovacia hodnota (tzv. Message Digest), čo v prípade funkcia SHA-1 budeme zapisovať v tvare

$$\text{Message Digest} = SHA(x) \tag{1.2}$$

1.3.2 GENERÁTOR PSEUDONÁHODNÝCH ČÍSEL

Hašovacie funkcie sú vzhľadom na ich jednocestnosť obľúbeným stavebným blokom pri vytváraní kryptograficky bezpečných generátorov pseudonáhodných čísel. Jednocestnosť zabezpečí nemožnosť spustiť spätný chod generátora aj pri zistení interného stavu generátora.

[6] V prípade hašovacích funkcií je uvažovaná náročnosť hľadania kolízií, pričom sa uvažuje tzv. narodeninový paradox.

Typickým príkladom je PRNG použitý v štandarde DSS (Digital Signature Standard) [1], [2]. DSA PRNG realizuje všetky aritmetické operácie modulo 2^N, pričom $160 \le N \le 512$ a je ho možné opísať takto:

1.) PRNG udržiava stavovú premennú X_i.
2.) PRNG môže využiť voliteľný vstup W_i. V prípade, že tento vstup neexistuje, je použitá hodnota $W_i = 0$.
3.) PRNG generuje výstup s využitím vzťahov

$$\begin{aligned} vystup[i] &= SHA\left(W_i + X_i \bmod 2^N\right) \\ X_{i+1} &= X_i + vystup[i] + 1 \left(\bmod 2^N\right) \end{aligned} \tag{1.3}$$

1.4 ZHRNUTIE

V rámci cvičenia sme prebrali významný stavebný blok – iteračnú hašovaciu funkciu SHA-1. Funkcia SHA-1 je v súčasnosti jediná hašovacia funkcia schválená organizáciou NIST a v praxi sa s ňou budeme stretávať predovšetkým v aplikáciách využívajúcich digitálny podpis.

LITERATÚRA

[1] Digital Signature Standard - FIBS PUB 186-2, Federal Information Processing Standards Publications, January 2000. U.S. Department of Commerce/National Institute of Standards and Technology. Dostupné v elektronickej forme **– FIPS186-2.pdf**.

[2] Kelsey, J – Schneier, B. – Wagner, D.: Cryptanalytic Attacks on Pseudorandom Number Generators. Counterpane system, dostupné v elektronickej forme – **PRNG.pdf**.

[3] Secure Hash Standard - FIBS PUB 180, Federal Information Processing Standards Publications, May 1993. U.S. Department of Commerce/National Institute of Standards and Technology.

[4] Secure Hash Standard - FIBS PUB 180-1, Federal Information Processing Standards Publications, April 1995. U.S. Department of Commerce/National Institute of Standards and Technology. Dostupné v elektronickej forme **– FIPS180-1.pdf**.

[5] Klíma, V.: Hašovací funkce a kódy: Výživná haše. CHIP 3/1999, s.40-43.

[6] Menezes, J.A. – Oorschot, P.C. – Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, New York 1997, ISBN 0-8493-8523-7.

[7] Advanced Encryption Standard, http//www.nist.gov/aes

[8] Advanced Encryption Standard (AES) - FIBS PUB 197, Federal Information Processing Standards Publications, November 26, 2001 U.S. Department of Commerce/National Institute of Standards and Technology. Dostupné v elektronickej forme **– FIPS-197.pdf**.

[9] Descriptions of SHA-256, SHA-384, and SHA-512. U.S. Department of Commerce/National Institute of Standards and Technology. Dostupné v elektronickej forme **– sha256-384-512.pdf**.

[10] Rosa, T.: Schéma digitalního podpisu: Vybrané problémy podpisových schémat. CHIP 1/2001, s.134-137.