Teoretický základ a prehľad kryptografických hashovacích funkcií

Hashovacia funkcia je funkcia (predpis) pre prevod vstupného reťazca dát na krátky výstupný reťazec. Tento reťazec sa označuje ako haš (ang. hash), charakteristika, odtlačok vstupných dát. Dĺžka hashu je závislá od zvolenej hashovacej funkcie, má fixnú dĺžku napr. MD5 je dlhý 128 bitov – 32 znakov, SHA-1 je dlhý 160 bitov - 40 znakov. Z hľadiska bezpečnosti sa doporučuje používať hashovacie algoritmy, ktoré majú výstup 160 a viac bitov. Pre zadanú funkciu h a vstup x, výpočet hashu h(x) musí byť dostatočne rýchly.

Hashovacie funkcie sa využívajú predovšetkým pre autentizáciu dát, kontrolu integrity dát (napr. samoopravné kódy), na zrýchlenie vyhľadávania. Je dôležitou súčasťou kryptografických systémov pre digitálne podpisy.

Podmienky kladené na hashovacie funkcie:

- Jednocestnosť – pre odtlačok y musí byť výpočtovo nezvládnuteľné nájsť taký výstup x, aby platilo h(x) = y. Podmienka hovorí o jednocestnosti hashovacej funkcie, to znamená, že hashovacia funkcia nesmie byť dešifrovateľná alebo aspoň nesmie byť výpočtovo zvládnuteľné nájsť k výstupu jeho vstup. Ak by táto podmienka nebola splnená, daná hashovacia funkcia nie je použiteľná, pretože útočník poznajúci hash správy by mohol jeho analýzou zistiť znenie pôvodnej správy.

- Slabá bezkolíznosť – pre vstup x musí byť výpočtovo nezvládnuteľné nájsť iný vstup $y \neq x$, tak aby platilo h(x) = h(y). Pri nesplnení tejto podmienky je hashovací algoritmus nepoužiteľný. Útočník by mohol k správe, ktorej hash získal, vytvoriť inú správu a vydávať ju za pravú, práve na základe rovnakého digitálneho odtlačku oboch správ.

- Silná bezkolíznosť - podmienka hovorí, že musí byť výpočtovo nezvládnuteľné nájsť dva rôzne vstupy x a y, pre ktoré platí h(x) = h(y).

Hashovacia funkcia, ktorá je jednocestná a spĺňa podmienky slabej a silnej bezkolíznosti hovoríme, že je kryptograficky dobrá hashovacia funkcia.

Ďalšou vlastnosťou je lavínovitosť, to znamená, že vytvorený hash musí úzko súvisieť s jednotlivými bitmi vstupu, aby aj pri jeho malej zmene došlo k rapídnym zmenám pri výstupnom hashi. Teda ak by sme zmenili len jedno písmeno, tak na výstupe dostaneme úplne iný hash.

Narodeninový paradox

Hovorí o tom koľko správ by sa muselo hashovať, aby sa našla kolízia, teda dve rôzne správy s rovnakým digitálnym odtlačkom. Odpoveď sa skrýva v nasledujúcom tvrdení. Je daná množina A o n rôznych prvkoch. Ak je n dostatočne veľké a k sa rovná približne hodnote $(2.n.ln2)^{0,5}$ potom v množine s k prvkami, ktoré sa vyberajú z množiny A náhodne, sa približne s 50% pravdepodobnosťou nájdu dva rovnaké prvky. Narodeninovým paradoxom sa nazýva z dôvodu, že sa dá toto tvrdenie aplikovať na problém nájdenia dvoch ľudí s rovnakým dátumom narodenia. Pre n=365 postačí skupina náhodne vybraných 23 ľudí k tomu aby sa medzi nimi našla asi s 50% pravdepodobnosťou dvojica oslavujúca narodeniny v rovnaký deň v roku. Podobne je to s hashovacími funkciami, kde A je množina všetkých možných výsledkov danej hashovacej funkcie. Ak budeme vychádzať z hashovacej funkcie MD5 hashovací kód je 128-bitový, v tom prípade n=2128 , a postačí hashovať 264 správ aby sme s 50% pravdepodobnosťou našli kolíziu. Tento počet sa dá označiť za výpočtovo nezvládnuteľný. Pokiaľ sa však dá povedať, že je možné nachádzať kolízie jednoduchšie ako pomocou narodeninového paradoxu, potom sa dá prehlásiť, že hashovacia funkcia je prelomená, pretože je výpočtovo zvládnuteľné nájsť kolíziu silnej bezkolíznosti(prvého radu).

Náhodné orákulum

Náhodné orákulum je určitá matematická fikcia. Podstatou je, že pri zadaní vstupnej správy je výstupom náhodná hodnota. Toto priradenie si náhodné orákulum zapamätá. Pri zadaní iných správ vždy vygeneruje náhodnú hodnotu výstupu to znamená nezávisle na tom, aké hodnoty už vygeneroval. Pokiaľ však zadáme správu, pre ktorú náhodné orákulum už výstupnú správu generovalo, potom siahne do svojej pamäte a vygeneruje rovnakú hodnotu ako predtým. Ideálna hashovacia funkcia by mala mať matematicky zhodné vlastnosti aké má náhodné orákulum.

MD4

MD4 je hashovací algoritmus vyvinutý v roku 1990 Ronaldom Rivestom. Je zdokumentovaný v RFC 1320 - The MD4 Message-Digest Algorithm. Výsledný hash má dĺžku 128 bitov. Bezpečnosť MD4 bola vážne ohrozená. Prvý plný kolízny útok proti MD4 bol uverejnený v roku 1995 a niekoľko novších útokov bolo zverejnených od tej doby. Od vtedy nie je považovaný za bezpečný.

Algoritmus MD4:

- Doplnenie správy na jednotnú dĺžku – doplň správu tak, aby jej dĺžka bola (n + počet zarovnávacích bitov) MOD 512 = 448 (jedna jednotka a zvyšok sú nuly), pripojenie týchto zarovnávacích sa uskutočňuje vždy, dokonca aj v prípade že dĺžka pôvodnej správy n bez zarovnávacích bitov spĺňa našu podmienku
- Doplnenie dĺžky správy – pridaj 64 bitovú reprezentáciu b(dĺžka správy v bitoch) , dĺžka správy je zarovnaná na celých 512 bitov, a teda máme bloky po šestnástich 32-bitových slovách

- Inicializácia bufferu – pracujeme s buffermi A, B, C, D, každý 32-bitové slovo a ich iniciálne hodnoty sú: A: 01 23 45 67
  B: 89 AB CD EF
  C: FE DC BA 98
  D: 76 54 32 10
- Spracovanie správy po blokoch (jeden blok je 16 32-bitových slov) - najprv definujeme 3 funkcie, ktoré použijeme v nasledujúcom algoritme:
  F(X; Y;Z) = (X AND Y) OR (NOT(X) AND Z)
  G(X; Y;Z) = (X AND Y) OR (X AND Z) OR (Y AND Z)
  H(X; Y;Z) = X XOR Y XOR Z

  skopíruj blok i do X;
  For j = 0 to 15 do
  Set X[j] to M[i * 16 + j].
  End

  skopíruj A do AA, B do BB, C do CC, D do DD;
  AA = A
  BB = B
  CC = C
  DD = D

- Výstupom algoritmu je zreťazenie bufferov AA, BB, CC a DD

MD5

Hashovací algoritmus MD5 bol vynájdený v roku 1992 Ronom Rivestom na univerzite MIT (Massachusetts Institute of Technology). Definovaný je v RFC 1321. MD5 je vylepšenou verziou svojho predchodcu algoritmu MD4, ktorý bol v tom čase síce veľmi rýchlym algoritmom ale jeho bezpečnosť bola narušená úspešnými kryptografickými útokmi. Pri tvorbe MD5 nebol kladený taký veľký dôraz na rýchlosť výpočtu ale na bezpečnosť. Hashovacia funkcia MD5 je definovaná pomocou kompresnej funkcie f a inicializačnej hodnoty Ho, ktorá je tvorená štyrmi 32-bitovými registrami A, B, C, D.

Tento algoritmus sa stále využíva, aj keď už nie je považovaný za bezpečný. Významnejšie problémy boli nájdené a publikované v roku 2004 (popis, ako vytvoriť dva súbory, ktoré zdieľajú hash. Algoritmus je veľmi podobný svojmu predchodcovi a medzi hlavné rozdiely patrí:

- bolo pridané 4. kolo;
- v každom kroku pridávame unikátnu konštantu;
- funkcia G v druhom kole bola zmenená;
- rýchlejší lavínový efekt.

SHA - 1

Hashovacia funkcia SHA – 1 boal vydaná v roku 1995 v spolupráci National Institute of Standards and Technology (NIST) a National Security Agency (NSA). Najnovšia verzia je štandardizovaná ako FIPS 180-1 1995 a RFC3174 - US Secure Hash Algorithm 1 (SHA1). Pôvodná špecifikácia algoritmu bola publikovaná v roku 1993 ako Secure Hash Standard,

FIPS 180. Táto verzia je teraz často označovaná ako SHA-0. Bola stiahnutá NSA krátko po zverejnení, a bola nahradená prepracovaním verzie. SHA-1 je veľmi podobná SHA-0, ale opravuje chybu v pôvodnej špecifikácií tak, že bola odolná voči známym zraniteľnostiam SHA - 0. SHA-1 vyrába 160-bitový hash založený na princípoch podobných tým, ktoré používa Ronald L. Rivest z MIT v návrhu MD4 a MD5 správ prehľadových algoritmov, ale je komplikovanejšia ako tieto dve funkcie. V roku 2005 boli identifikované bezpečnostné chyby v SHA-1 a síce, že by mohla existovať matematická slabosť. Od roku 2010 nie je doporučené používať túto funkciu pre aplikácie, v ktorých sa vyžaduje väčšia bezpečnosť.

Schématický zápis algoritmu:

- doplnenie správy na jednotnú dĺžku - rovnako ako u MD4 a MD5
- doplnenie dĺžky správy – rovnako ako u MD4 a MD5
- inicializácia bufferov – používame 5 bufferov namiesto 4
- samotný algoritmus - tentokrát omnoho zložitejší, správu delíme po 512-bitoch. Tie uložíme ako 16 slov a ďalších 64 vygenerujeme. Vykonáme 4 kola rovnako ako u MD5, pričom v každom kole používame inú funkciu a iné konštanty. Každé kolo obsahuje 20 iterácií. Nakoniec sa updatujú buffery.
- výsledkom sú zreťazené buffery

SHA – 2

Hashovacia funkcia SHA – 2 je následníkom hashovacej funkcie SHA – 1. Bola vydaná opäť v spolupráci NIST a NSA v roku 2002. Nové verzie sú nasledujúce: SHA-256, SHA-384, SHA-512. Čísla v menách funkcie odpovedajú dĺžke výsledného hashu. Aktuálnym štandardom pre SHA-2 je FIPS 180-2 2002. Ďalšia verzia bola potom pridaná v roku 2007 a dĺžka výsledného hashu je v tomto prípade 224 bitov. Štandardom pre túto funkciu je FIPS 180-3 2007. SHA-256 a SHA-512 sú hashovacie funkcie počítajúce s 32 - a 64-bitovými slovami. Používajú rôzne zmeny čiastky a aditívne konštanty, ale ich štruktúry sú inak prakticky totožné, líšia sa iba počtom kôl. SHA-224 a SHA-384 sú proste skrátené verzie prvých dvoch, počítajúce s rôznymi počiatočnými hodnotami. SHA-2 funkcie nie sú tak široko používané ako SHA-1, a to napriek ich lepšiemu zabezpečeniu. Dôvody môžu zahŕňať nedostatok podpory pre SHA-2 na systémoch s operačným systémom Windows XP SP2 alebo starším. SHA-256 sa používa na overenie Debian Linux softvérových balíkov a v správe DKIM podpisu normy, SHA-512 je súčasťou systému na overenie archívneho videa z Medzinárodného trestného tribunálu pre rwandskú genocídu. SHA-256 a SHA-512 sú navrhnuté pre použitie v DNSSEC. Predajcovia Unix a Linux sa sťahujú do používania 256 a 512-bit SHA-2 pre bezpečné hashovanie hesla.

SHA – 3

Aj keď zatiaľ neboli objavené zraniteľnosti rodiny štandardov SHA – 2, z historických skúseností to môžeme predpokladať. Z tohoto dôvodu bola už v roku 2008 vyhlásená otvorená súťaž, ktorá má za úlohu nájsť nástupcu SHA-2, zatiaľ označovaného ako SHA-3. Súťaž ešte stále prebieha a víťaz sa očakáva behom tohto roku (2012).

Whirpool

Jedná sa o menej známu hashovaciu funkciu, ktorú vytvoril Vincent Rijmen a Paulo Barreto. Prvá verzia Whirpool-0 bola publikovaná v roku 2000. Druhá verzia Whirpool-T bola vybraná pre New European Schemes for Signatures, Integrity and Encryption (NESSIE) projekt. Tretia a posledná verzia bola vydaná a prijatá ako International Organization for Standardization (ISO) standard ISO/IEC ISO-10118-3:2004. Whirpool používa upravený AES algoritmus ako svoj základ a produkuje hash dĺžky 512 bitov. Jedná sa o relatívne nový návrh, takže stále nie je dostatok skúseností s touto funkciou. Výkon je o niečo lepší ako v prípade SHA funkcií, avšak aj za vyšších požiadaviek na hardware.

Tiger

Tiger je hashovacia funkcia, ktorou v roku 1995 navrhli Ross Anderson a Eli Biham. Táto funkcia produkuje kontrolný súčet, alebo hash o dĺžke 192 bitov, príp. 128 alebo 160 u verzií Tiger/128 a Tiger/160 (u týchto verzií sa hash získava skrátením z pôvodnej dĺžky 192 bitov). Používa sa pre kontrolu integrity súborov alebo ukladanie hesiel. Tiger 2 je varianta funkcie Tiger, ktorá používa rovnaké zakončenie vstupných dát, ako funkcie MD5 či SHA-1, oproti mierne odlišnému zakončeniu dát v pôvodnej funkcií Tiger. Oficiálna špecifikácia funkcie Tiger 2 doteraz nebola publikovaná. Tiger sa často používa v tzv. Merklovom hashovom strome, kde je označovaný ako TTH (Tiger Tree Hash). TTH sa používa napríklad v mnohých P2P aplikáciách, napr. Direct Connect alebo Gnutella.

Haval

Haval je kryptografická hashovacia funkcia. Na rozdiel od MD5, ale rovnako ako väčšina moderných kryptografických hashovacích funkcií, môže haval produkovať hashe rôznych dĺžok a to 128 bitov, 160 bitov, 192 bitov, 224 bitov a 256 bitov. Tiež umožňuje používateľom určiť počet kôl (3, 4 alebo 5), ktoré budú použité pre generovanie hashu. Bol vynájdený v roku 1992 a jeho autormi sú Yuliang Zheng, Josef Pieprzyk a Jennifer Seberry.

RIPEMD (RACE Integrity Primitives Evaluation Message Digest)

RIPEMD - 160 je kryptografická hashovacia funkcia vyvinutá v Leuven v Belgicku a jej autori sú Hans Dobbertin, Antoon Bosselaers a Bart Preneel. Prvýkrát bol publikovaný v roku 1996. Je to vylepšená verzia RIPEMD, ktorý bol založený na konštrukčných princípoch používaných v MD4, a je podobný vo výkone na viac populárne SHA-1. Existujú aj 128, 256 a 320-bitové verzie tohto algoritmu s názvom RIPEMD-128, RIPEMD-256 a RIPEMD-320. RIPEMD-160 bol navrhnutý v otvorenej akademickej obci. RIPEMD – 160 sa javí ako o niečo menej používaný oproti SHA – 1, čo môže spôsobiť, že je menej preskúmaný než SHA. V auguste 2004 bola hlásená kolízia pôvodného RIPEMD.