DUOMENŲ STRUKTŪROS IR ALGORITMAI

MARIUS GŽEGOŽEVSKIS

TERMINAI

Kriptografija (iš gr. *κρυπτός, kryptó s* 'paslėptas' + gr. *γράφειν, graphein* 'rašyti') – mokslas, tiriantis informacijos užšifravimo ir iššifravimo metodus. Kriptografija yra sudėtinė kriptologijos mokslo dalis.

Kriptografijos uždaviniai ir priemonės

Apžvelkime gerokai supaprastintą padėtį, į kurią patenka šiuolaikinės informacinės visuomenės žmonės:

- A (Algis) siunčia informaciją B (Birutei), perdavimo kanalą kontroliuoja Z (Zigmas) ir jaučiasi padėties šeimininkas. Jis gali pasyviai stebėti perdavimo kanalą, skaityti siunčiamus pranešimus ir kaupti dosjė; jis gali veikti aktyviai - pakeisti dalį siunčiamos informacijos, o kartais - apsimesti A ir siųsti jo vardu pranešimus B arba apsimesti B. Taigi dėl Zigmo veiksmų gali būti pažeidžiamos šios siunčiamų duomenų savybės:

Kriptografijos uždaviniai ir priemonės

Slaptumas (konfidencialumas, *confidentiality*)
- Šifravimas

Vientisumas (integralumas, *integrity*)
- Kriptografinės maišos funkcijos
- Skaitmeninis parašas

Autentiškumas (tapatumo nustatymas, *authenticity*)
- MAC
- Skaitmeninis parašas

Informacijos vientisumas

Vientisumas *(integrity)* – tai garantija, kad bus išsaugotos teisingos duomenų reikšmės. Tai užtikrinama draudžiant neautorizuotiems vartotojams kokiu nors būdu pakeisti, modifikuoti, sunaikinti, arba kurti duomenis.

Maišos funkcija ir santrauka

Pranešimo vientisumui užtikrinti naudojama **pranešimo santrauka** (*message digest*, kartais dar vadinama *Modification Detection Code* (*MDC*)).

Pranešimo santrauka skaičiuojama, naudojant maišos funkciją. Pranešimo santrauka vadiname maišos funkcijos reikšmę.

Maišos funkcija ir santrauka

Maišos funkcija (angl. *hash function*) vadiname funkciją, kuri bet kokio baigtinio ilgio ženklų eilutei priskiria *fiksuoto ilgio* eilutę.

Maišos funkcijos naudojamos ne tik informacijos vientisumo patikrinimui, bet ir dokumento santraukai gauti skaitmeninio parašo schemose, slaptažodžių saugojimui, paieškos raktų formavimui duomenų bazėse ir panašiai.

Maišos funkcijų rezultatų pavyzdžiai

- **CRC32** (32 bitai): 9468ffc5
- **MD5** (128 bitai): 5ccb2201a7ee633d2b2dc1ff527d4c79
- **SHA-1** (160 bitų):

  09035a1b2703a81b7a86e1abe9965b5756666591
- **SHA-256** (256 bitai):

  7765253a2cf3996d41c2a54af094a80bfcb8f9c4b4b8d4352c3e1ab8e3c7bbfb

Santraukos ilgis yra fiksuotas

Pavyzdys. Skaičiuosime duoto pranešimo MD5 maišos funkcijos reikšmę:

Tuščias pranešimas:

- d41d8cd98f00b204e9800998ecf8427e

***A*:**

- 7fc56270e7a70fa81a5935b72eacbe29

Ilgesnis pranešimas iš kelių žodžių:

- cd894604746deba896568cb8f2f6f197

3,5 MB dydžio failas:

- 9659f0218ab08598f7f53edec512479e

Maišos funkcijų savybės

Maišos funkcijos yra ***determinuotosios***, t. y. ne atsitiktinės:

- Skaičiuojant maišos reikšmę tai pačiai įvesčiai kelis kartus, visada bus gaunamas tas pats rezultatas.

Kadangi galimų įvesčių yra daugiau, negu galimų išvesčių, tai maišos funkcijos nėra ***injektyvios***:

- Gali būti, kad skirtingoms įvestims maišos reikšmės bus vienodos. Tai vadinama **sutapimu** (kolizija, angl. *collision*).

MAIŠOS ALGORITMAI

Maišos algoritmai yra naudojami duomenims, esantiems sistemose su laisvu priėjimu, adresuoti. Jų gamyba prasidėjo 1950 metų viduryje. Maišos procesas iš pradžių buvo vadinamas randomizacija (angl. randomizing).

Nuo 1968 m. šis procesas vadinamas maiša (angl. **hashing**), nes vėliau nustatyta, kad rakto pritaikymo adresui procesas neturi būti atsitiktinis.

KRIPTOGRAFIJOS MAIŠOS ALGORITMAI

Daugelis sistemų analitikų vengia išmaišymo algoritmų, laikydami juos per daug sudėtingus. Iš tikrųjų išmaišymo algoritmai ir jų naudojimas yra pakankamai paprasti. Taip pat jie turi 2 aiškius privalumus, palyginti su indeksavimo metodais:

- pirmiausia daugumą įrašų galima rasti vienu kreipiniu į išorinį atminties įrenginį;
- antra, gana lengvai daromas įrašų įterpimas ir išmetimas.

KRIPTOGRAFIJOS MAIŠOS ALGORITMAI

Indeksavimas sėkmingai taikomas bylose, išrikiuotose pagal pradinį raktą, ir duoda didžiausią efektą daugelyje paketinio apdirbimo sistemų. Dabar yra žinoma daug maišos algoritmų (Rompay 2004).

MAIŠA (angl. hashing)

Maiša (angl. hashing). Tai yra matematinė transformacija, atliekama tam tikro ilgio simbolių eilutę verčiant į fiksuoto ilgio – dažniausiai trumpesnę – reikšmę.

Ši reikšmė atstovauja originaliai eilutės reikšmei. Dažniausiai maiša naudojama duomenų įrašų duomenų bazėje indeksuoti ar ištraukti, nes taip operacija yra atliekama greičiau – surasti trumpesnę reikšmę yra mažiau sąnaudų reikalaujanti operacija, negu surasti originalią simbolių eilutę. Maiša taip pat naudojama ir kodavimo algoritmuose.

MAIŠOS FUNKCIJA

Maišos funkcija (angl. **hash function**). Tai funkcija f = f (x), priskirianti argumentui x pseudoatsitiktinį skaičių, vadinamą maišos kodu (angl. **hash code**). Tam pačiam argumentui funkcija visada turi duoti tokį patį rezultatą, taigi ji nėra atsitiktinė. Dažniausiai funkcijos reikšmių sritis (t. y. f) yra, palyginti su apibrėžimo sritimi (t. y. x), nedidelė.

MAIŠOS ALGORITMAS (angl. hash algorithm).

Maišos algoritmas (angl. hash algorithm). Tam tikrų maišos funkcijų iš anksto taisyklėmis nustatytas ir apibrėžtas nuoseklus vykdymas.

MAIŠOS LENTELĖ (angl. hash table).

Tai yra duomenų struktūra, kurioje duomenys yra saugomi jiems priskiriant unikalų raktą (angl. **unique key**). Raktus generuoja įvairios maišos funkcijos.

Maišos lentelės yra ypač naudingos, kai yra vykdoma paieškos operacija – maišos funkcija pagal duomenis identifikuojančią informaciją generuoja unikalų raktą, kuris dėstymo lentelėje naudojamas įrašams rikiuoti ir aptikti.

DUOMENŲ SANTRAUKA (angl. message digest, cheksum).

Duomenų santrauka (angl. message digest, cheksum). Kriptografinės maišos funkcijos rezultatas, transformavus tam tikro fiksuoto dydžio duomenis (Preneel 2003).

SAVYBĖS IR TAIKYMO SRITYS

Pagrindinės maišos algoritmų savybės:

• apibrėžimo sritis yra didelė, palyginti su fiksuoto dydžio reikšmių sritimi;

• funkciją nesunku apskaičiuoti bet kokiam argumentui;

• daug kartų kviečiant funkciją su įvairiais argumentais, jos rezultatai pasiskirsto tolygiai;

• pagal rezultatą negalima vienareikšmiškai nustatyti argumento;

• yra gana nedidelė tikimybė, jog dviejų argumentų x ir y rezultatas f(x) = f(y). Tai itin svarbu kriptografijoje.

SAVYBĖS IR TAIKYMO SRITYS

Maišos algoritmai naudojami:

• duomenų struktūrose rikiuojant duomenis;

• duomenų bazėse vykdant užklausas, indeksavimą;

• programinėje įrangoje patikrinti, ar tam tikri duomenys (pvz., rinkmena) nėra pakeisti sistemos išorėje;

• kriptografijoje.

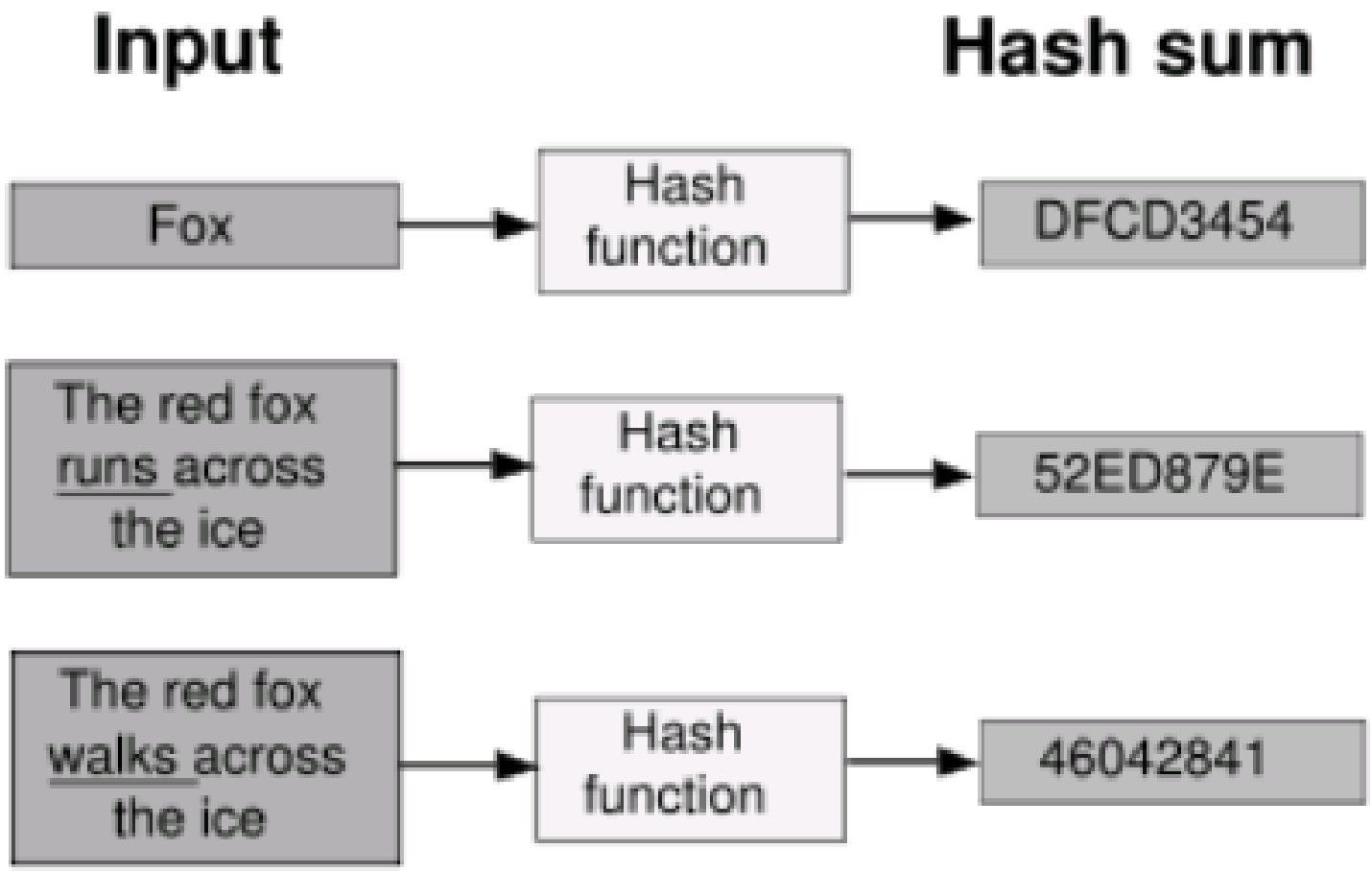
SAVYBĖS IR TAIKYMO SRITYS

Jei funkcijos apibrėžimo sritis yra objektas, kaip maišos funkcija neretai vartojamas fizikinis to objekto adresas kompiuterio atmintyje. Tuomet tariama, jog kiekvienas objektas gali būti lygus tik sau pačiam.

Jei pagal programos algoritmą du skirtingą padėtį atmintyje užimantys objektai gali būti lygūs tarpusavyje, būtina iš naujo apibrėžti ne tik palyginimo operaciją, bet ir tos klasės objektams taikomą maišos algoritmą, pakeičiant du sutartus, kiekvieno objekto turimus virtualius metodus (pvz., equals ir hashCode Java programavimo kalboje) (Rompay 2004).

SAVYBĖS IR TAIKYMO SRITYS

Maišos algoritmai, naudojami duomenų užklausoms formuoti, nebūtinai korektiškai veiks kriptografijoje dėl klaidų tikrinimo. Todėl kiekvienas panaudojimo aspektas turi savo specifines maišos funkcijas (Augot, Finiasz, Sendrier 2005).



Tipinis maišos algoritmas

MAIŠOS ALGORITMAI KRIPTOGRAFIJOJE

Kriptografijoje maišos algoritmai priima tam tikrą duomenų kiekį, juos transformuoja ir grąžina fiksuoto dydžio simbolių eilutę. Tai yra vadinama maišos reikšme. Maišos reikšmė yra tiksli didesnio duomenų kiekio ar dokumento reprezentacija, tai yra tam tikras skaitmeninis pirštų antspaudas.

Todėl, kad šie maišos algoritmai yra vienakrypčiai, jie plačiai paplitę kriptografijoje. Taip pat šiuos algoritmus galima taikyti ir duomenų integralumui ir korektiškumui tikrinti (Augot, Finiasz, Sendrier 2005).

SECURE HASH ALGORITHM (SHA).

Secure Hash Algorithm (SHA). Saugus maišos algoritmas iš pranešimo, kuris yra mažesnis už 264 bitų, generuoja 160 bitų kodą. Šis kodas yra skaičiuojamas taip:

• pradinis tekstas suskirstomas į N blokų po 512 bitų (64 baitus);

• jei paskutiniame M_n bloke trūksta informacijos iki 512 bitų, bloko gale pridedamas 1 ir tiek 0, kad būtų užpildytas blokas paliekant 64 bitus pradinio teksto ilgio išsaugojimui bitais;

• naudojamos funkcijos f0, f1, … f79. Kiekviena funkcija operuoja trimis 32 bitų žodžiais B, C, D ir grąžina vieną 32 bitų žodį.

SECURE HASH ALGORITHM (SHA).

Taip pat yra SHA-1 maišos algoritmo atmainų, skaičiuojančių skirtingo dydžio bitų kodus: SHA256, SHA384, SHA512, atitinkamai skaičiuoja 256, 384 ir 512 bitų santraukų kodus (Rompay 2004).

SHA-1

SHA-1 yra populiarus algoritmas, naudojamas įvairiose programose ir protokoluose, kaip antai TLS, SSL, PGP, SSH, S/MIME, IPSec. SHA-1 buvo laikomas pažeidžiamo MD5 algoritmo įpėdiniu. 2004 m. ir 2005 m. atrastas pažeidžiamumas skatina svarstyti, prieš tai šį algoritmą naudojant itin aukšto saugumo reikalaujančiose srityse.

MD2 (Message-Digest algorithm 2)

MD2 (Message-Digest algorithm 2). Pirmoji MD5 algoritmo versija, sukurta profesoriaus Ronaldo Rivesto ir skirta naudoti didelės apimties duomenų parašui sukurti. MD2 maišos algoritmas yra optimizuotas 8 bitų kompiuteriams, o naujesnės jo versijos MD4 ir MD5 – 32 bitų kompiuteriams.

MD5

MD5 (Message-Digest algorithm 5) šiuo metu yra populiariausias maišos algoritmas, apskaičiuojantis 128 bitų ilgio parašą. Nors MD5 algoritmas (kaip ir kiti algoritmai) gali apskaičiuoti parašą nuo begalinio skaičiaus įeinamų duomenų, galimų sugeneruotų kodų (parašų) skaičius yra baigtinis – 2 128, todėl jau seniai buvo žinoma, kad kolizijos egzistuoja.

Anksčiau tai nebuvo aktualu dėl reliatyviai silpnų kompiuterių pajėgumų, tačiau dabar MD5 algoritmas laikomas nesaugus. Tai parodė 2004 m. atrastas pažeidžiamumas. Pavyzdys: MD5(„Sveiki gyvi“) = 23f00289ee3181c0ab6ab3fcd20f983b.

RIPEMD-160

RIPEMD-160 (RACE Integrity Primitives Evaluation Message Digest) sukurtas 1996 m. akademinės bendruomenės, todėl nėra suvaržytas patentų. Šis algoritmas yra ne toks populiarus kaip MD5 ar SHA-1, todėl atitinkamai mažiau ir analizuotas. RIPEMD-160 apskaičiuoja 160 bitų parašą (Menezes, Oorschot, Vanstone 1996).

RMD160 („Labas rytas“) = **2755eae08ccb73b63872dded3982e50b47d47388**.

ALGORITMŲ PALYGINIMAS

Toliau pateikiami spartos rezultatai labiausiai paplitusiems kriptografijos algoritmams, nes jų sparta yra aktualiausias rodiklis iš maišos algoritmų. Visi algoritmai buvo programuoti naudojant C++ kalbą, sukompiliuoti su MS Visual C++ 2005. Naudotas centrinis procesorius buvo Intel Core2 1,83 GHz kartu su Windows XP SP2 32 bitų operacine sistema.

Remtąsi:
http://leidykla.vgtu.lt/conferences/jmk_grafika_2008/files/pdf/kavaliunas_31-37.pdf

Maišos algoritmų lyginimas

ALGORITMO PAVADINIMAS	MB/SEKUNDĘ	CIKLAI PER BAITĄ
AES-64	1489	1,2
AES-128	794	2,2
CRC-32	256	6,8
MD5	258	6,8
SHA-1	155	11,3
SHA-256	81	21,5
SHA-512	99	17,6
Tiger	217	8,0
Whirlpool	58	30,0
RIPEMD-160	108	16,1
RIPEMD-320	111	15,8
RIPEMD-128	155	11,3
RIPEMD-256	159	11,0