Jegyzőkönyv

Biztonság és védelem az informatikában c. tárgy

12. feladat

Készítette:

Csomor Bence, TVIK4I

2021.05.05.

**Hash algoritmusok**

A hash eljárás bármely olyan eljárás, amely felhasználható az tetszőleges méretű adatok rögzített méretű adatra történő leképezésére. A hash eljárás által visszaadott értékeket hash értékeknek, vagy hash kódoknak nevezik. A hash algoritmusokat lehet generált tábla indexelésre használni és lehet üzemet hitelesítésre is. Esetünkben ez a felhasználás érdekes.

Elterjedt hash algoritmusok: MD5, SHA (SHA-512), Whirlpool, Üzenethitelesítő algoritmusok: HMAC, CMAC

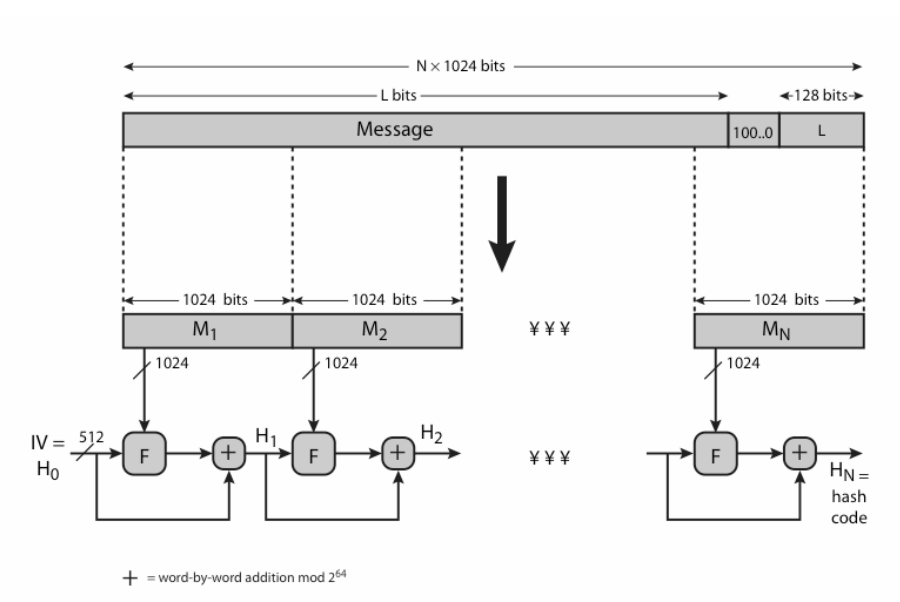
Az MD5:

* Az MD5 az egyik legelterjedtebb hash függvény
* 128 bites hash értéket ad
* Letöltéseknél ellen l ellenőrző összegként használt
  + persze ha a hash érték ugyanabból a forrásból származik (szintén letöltött), akkor csak véletlen hibák ellen véd, rosszindulatú hamisítások ellen nem
* Jelszavaknak hash értékként való tárolására is használják
* Ez a 128 bites hash ma már nagyon egyszerűen legenerálható

SHA (Secure Hash Algorithm):

* 160 bites hash értéket ad
* Ezt az algoritmust felülvizsgálták a kétezres években ütközés miatt, amit 269 művelettel lehetett találni. Helyette az SHA-2 lett használva, amelynek több változata és hosszabb hash értékei vannak
* Az SHA-2 három új változata, bithossz szerint: SHA-256, SHA-384, SHA-512
* Az SHA-2 már jóval biztonságosabb elődjénél

SHA-512:

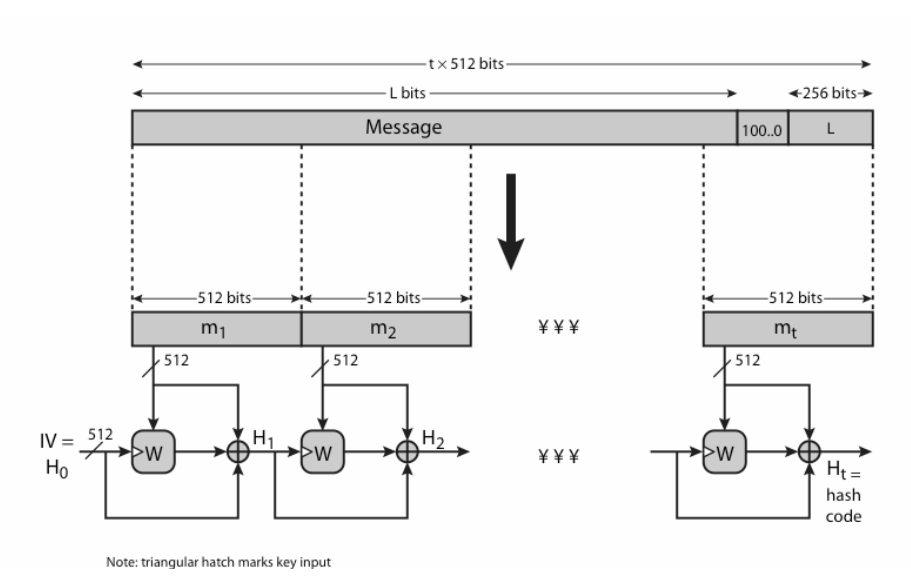


Whirlpool:

* A NESSIE = New European European Schemes for Schemes for

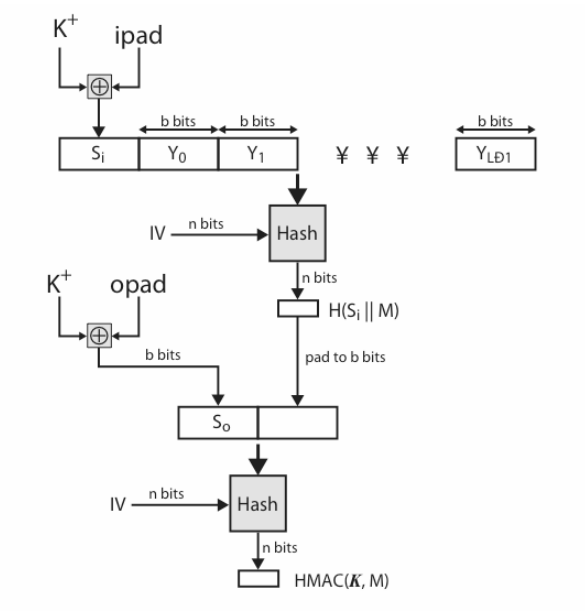
Signatures, Integrity, and Encryption Signatures, Integrity, and Encryption európai projekt által támogatott hash függvény

* tömörítő függvényként az AES módosított változatát használja
* teljesítménye és biztonsága összevethető, (ha nem jobb), mint az MD (ha nem jobb), mint az MD és SHA hasfüggvény családnak
* vizsgálatok szerint: nagyobb a hardverigénye, mint az SHA-nak, de az általa nyújtott teljesítmény is jobb



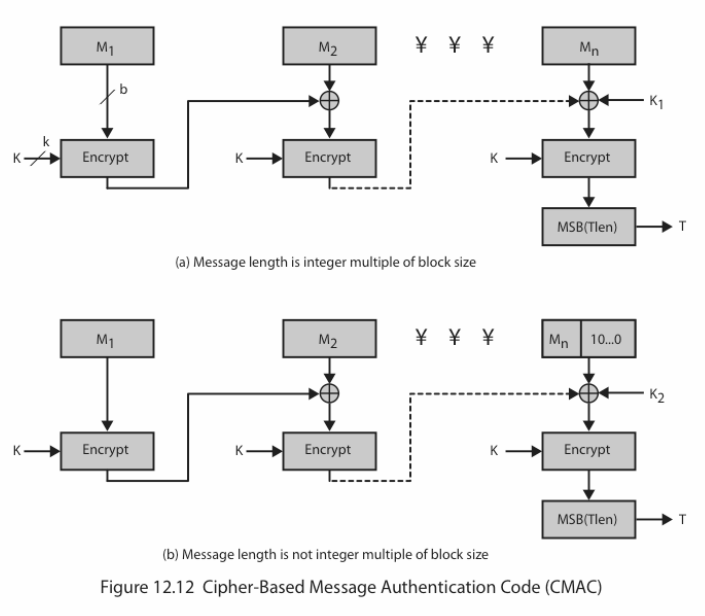
HMAC:

* HMACK = Hash [ (K+ XOR opad) || Hash [ (K+ XOR ipad) || M) ] ]
* Itt a K+ a kulcs feltöltve, az opad és ipad speciális feltöltő konstansok
* Ez bármely hash függvényre alkalmazható
* Bizonyítható, hogy a biztonsága arányos a használt hash algoritmus biztonságával
* Biztonságát egy üzenet HMAC kódjának sikeres hamisításához szükséges üzenet – HMAC érték párok számában mérik
* Támadási típusok:
  + brute force a használt kulcs ellen (2n)
  + születésnapi támadások (2n/2 de ehhez nagyon sok (2n) üzenet – HMAC érték pár kell ugyanazzal a kulccsal)
* Igy még a 128 bites MD5-tel is biztonságos



CMAC:

* CMAC = titkosító alapú üzenethitelesítő kód Cipher-based Message Authentication Code
* Széles körben használt (USA)
* Az üzenet mérete korlátozott, de ez kiváltható 2 kulcs használatával és feltöltéssel



Egy hash eljárásnak következő tulajdonságokat kell teljesítenie:

1. egy megfelelő hash-függvénynek a lehetséges bemeneteket a lehető legegyértelműbben kell leképezni a kimeneti tartományba,

Ez azt jelenti, hogy a kimeneti tartomány minden hash értékét nagyjából azonos valószínűséggel kell, hogy tartalmazza. Ha néhány hash-érték nagyobb valószínűséggel fordul elő, mint mások, akkor az eredmények között nagyobb valószínűséggel fordulhatnak elő azonos értékek.

Azt a jelenséget, amely két különböző bemeneti értékből azonos hash értéket generál, ütközésnek nevezzük.

2. Ennek következménye:

* a kiindulási adathoz tetszőleges bájt hozzáírásával a hash érték véletlenszerűen megváltozik,
* a kiindulási adatból tetszőleges bájt törlésével a hash érték véletlenszerűen megváltozik,
* két nem azonos kódú bájt felcserélésével a hash érték véletlenszerűen megváltozik,
* tetszőleges bájt átírásával a hash érték véletlenszerűen megváltozik
* egy ismert tartalom nem vagy nehezen változtatható meg algoritmikus úton úgy, hogy a kulcsa a kívánt értékre változzék meg.

Az azonos kimeneti értékek vagy ütközések lehetővé teszik az adott algoritmussal védett tanúsítványok hamisítását.

Hash algoritmusok használata:

Adódhat programozás során olyan eset, amikor a feladatunk az, hogy egy bemenetről megmondjuk, hogy része-e egy tárolt adatstruktúrának, adatbázisnak. Ilyen problémák megoldására találták ki a keresési algoritmusokat. Ezekből is annyi van, mint égen a csillag. Jellemzőjük azonban, hogy időt vesznek igénybe és drámaian tudják lassítani egy program működését, ha viszonylag sokat végzünk el belőlük és nem optimálisan vannak megírva.

Olyan esetekben, mikor sűrűn kell keresni, célszerű egy olyan tárolási módszert bevezetni, amely eleve megkönnyíti a keresést. Programozásban ilyen tárolási szerkezet a HashSet, amely egy olyan tömb, amiben az elemek indexét maga az elemből képzett hash adja. Így a keresés igen egyszerű, mert csupán azt kell megvizsgálni keresés esetén, hogy a képzett indexen a tömb tárol-e értelmes adatot. A módszer “hátránya”, hogy a kollekció többször nem tartalmazhatja ugyanazt az elemet.

MD5 algoritmus részletes bemutatása:

Az MD5 (Message-Digest algorithm 5) egy 128 bites, egyirányú kódolási algoritmus. Az RFC 1321-es internet szabványban foglaltak szerint használják internetes adatok kódolására, illetve titkosítási kulcsokban. Az MD5-kódolást biztonsági alkalmazások széles skálája használja adatellenőrzésre, például fájlok eredetiségének (sértetlenségének) vizsgálatára.

Az MD5-kódolást az RSA algoritmus megalkotóinak egyike, Ronald L. Rivest professzor fejlesztette ki 1991-ben az elavult MD4 lecserélésre. 2005 óta elektronikus aláírás területen használata nem javasolt, és 2010. december 31-ével az utódja, az SHA-1 algoritmus is kiváltandó az SHA-256 algoritmussal.

Az MD5 kódolás bármilyen adatból egy 16 bájt hosszú bináris hasht hoz létre. A kódolás egyirányú, így nem lehet visszafejteni. Éppen emiatt alkalmas eredetiség ellenőrzésre.

Például egy beléptető rendszer nem kell, hogy eltárolja az egyes felhasználók jelszavait, elég, ha csak azok vissza nem fejthető hashét rögzíti. A bejelentkezési kísérletnél a megadott jelszó hashét összehasonlítva az eredetileg tárolttal megkapjuk annak helyességét. A gyakorlatban a hash-t nem magából a jelszóból számoljuk ki, hanem egy véletlenszerű adat (a só) és a jelszó összekapcsolásával előállított stringre, hogy megnehezítsük a szótáralapú támadásokat.

Az MD5 hashek visszafejtése nem lehetséges, léteznek azonban különböző online is elérhető adatbázisok, melyek kódolt adatokat, és a hozzájuk tartozó hasheket tartalmazzák.

Az alábbi két szó csak egyetlen karakteren különbözik (a kódszavak Hamming-távolsága egy), a kapott MD5-hash azonban teljesen más:

MD5("majom"): bcb559cd9d05046da8ec6ea3175a834c

MD5("bajom"): e20c0bddf6416a2021f18b6b05784e88

források: <http://www.uni-obuda.hu/users/schuster.gyorgy/sha1.pdf>

<http://www.inf.u-szeged.hu/~zlnemeth/crypto/OLD-SLIDES/crypt10.pdf>

<https://www.webmaster442.hu/hash-algoritmusok/>

<https://hu.wikipedia.org/wiki/MD5>