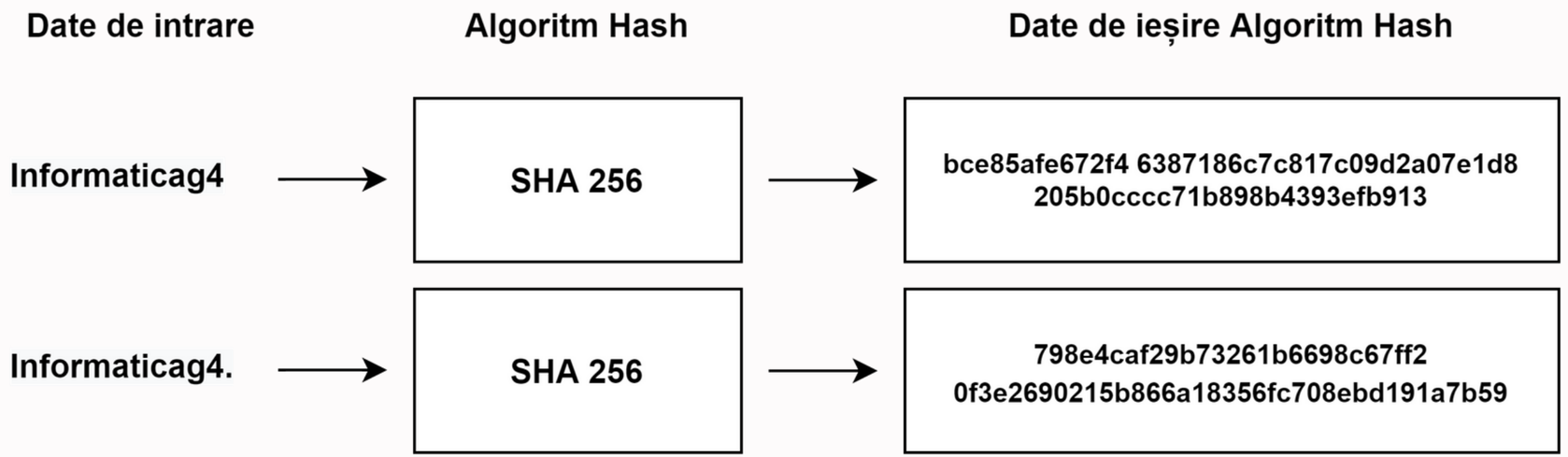
Mecanism de autentificare **utilizând funcția SHA-256**

Student: Pavel Alexandru Daniel
Facultatea de Matematică și Informatică
Anul 1 Seria 2 Grupa 4

Cuprins

CE REPREZINTA FUNCȚIA HASH 256?

- SHA256 este utilizată pentru securizarea datelor.
  Spre exemplu: tranzacțiile Blockchain și Bitcoin
- Funcție matematică ce convertește datele într-un șir de caractere de 256 de bits.
- Secure Hash Algorithm 256
- Proces:



REZULTATUL ACESTEIA POATE SĂ FIE DECRIPTAT?

-> SHA256 reprezintă o funcție hash, nu o funcție de criptare. Din acest motiv nu poate fi decriptat rezultatul pe care il prezinta datele de ieșire SHA256.

-> Nu poate să fie inversat deoarece este o **funcție unidirecțională**. Este ușor de calculat la fiecare intrare, dar greu de calculat inversa. Acesta ar putea provoca un atac care nu și-ar atinge obiectivul de proiectare.

CARACTERISTICI

- Date de intrare sunt imprevizibile
- Se efectueaza întotdeauna un șir de 512 biți
- Prezintă un efect excelent de avalanșă (reprezintă procesul când se schimbă un singur bits, suma hash devine complet diferită).

ÎNAINTE DE A ÎNCEPE PROCESUL DE HASHING 1/2

Se declară *8 valori inițiale hash*

h0 = 0x6a09e667; h4 = 0x510e527f;

h1 = 0xbb67ae85; h5 = 0x9b05688c;

h2 = 0x3c6ef372; h6 = 0x1f83d9ab;

h3 = 0xa54ff53a; h7 = 0x5be0cd19

-> Acestea prezintă 32 de biți din rădăcina pătrată a primelor 8 numere prime.

ÎNAINTE DE A ÎNCEPE PROCESUL DE HASHING 1/2

Pasul 1: Se calculeaza radacina pătrată din numărul prim 2.
sqrt(2) - (1.41421356237309504880168872420 97)

Pasul 2: Se retine partea sa zecimala.
(0.41421356237309504880168872420 97)

Pasul 3: Este înmulțița cu 2 la puterea 32.
2^32 (0.41421356237309504880168872420 97)
Rezultat: (1779033703.952099384902777060052 6)

Pasul 4: Valorile sunt prezentate de tip hexazecimal.
(6A09E667) = h0 = 0x6a09e667;

ÎNAINTE DE A ÎNCEPE PROCESUL DE HASHING 2/2

Se declară *64 de constante hash*

```
numar k[64] = {
    0x428a2f98,0x71374491,0xb5c0fbcf,0xe9b5dba5,0x3956c25b,0x59f111f1,0x923f82a4,0xab1c5ed5,
    0xd807aa98,0x12835b01,0x243185be,0x550c7dc3,0x72be5d74,0x80deb1fe,0x9bdc06a7,0xc19bf174,
    0xe49b69c1,0xefbe4786,0x0fc19dc6,0x240ca1cc,0x2de92c6f,0x4a7484aa,0x5cb0a9dc,0x76f988da,
    0x983e5152,0xa831c66d,0xb00327c8,0xbf597fc7,0xc6e00bf3,0xd5a79147,0x06ca6351,0x14292967,
    0x27b70a85,0x2e1b2138,0x4d2c6dfc,0x53380d13,0x650a7354,0x766a0abb,0x81c2c92e,0x92722c85,
    0xa2bfe8a1,0xa81a664b,0xc24b8b70,0xc76c51a3,0xd192e819,0xd6990624,0xf40e3585,0x106aa070,
    0x19a4c116,0x1e376c08,0x2748774c,0x34b0bcb5,0x391c0cb3,0x4ed8aa4a,0x5b9cca4f,0x682e6ff3,
    0x748f82ee,0x78a5636f,0x84c87814,0x8cc70208,0x90befffa,0xa4506ceb,0xbef9a3f7,0xc67178f2
};
```

-> Acestea prezintă 32 de biți din radicalul a primelor 64 numere prime. Diferența față de calculul celor 8 valori inițiale hash o reprezintă calcularea părți fracționare ale rădăcinii cubice.

ETAPE DE CALCUL 1/4

Pasul 1: Se declară 8 valori inițiale hash

Pasul 2: Se declară 64 de constante hash

Pasul 3: Spre exemplu, datele de intrare "Informaticag4"

Codul ASCII pentru acestea îl reprezintă 731101021111141099711610599710352

Pasul 4: Se declară un array w[0] - w[63] de tip dată *unsigned long 32 bits*

ETAPE DE CALCUL 2/4

Pasul 5:

Se mută 15 octeți ASCII în matricea de programare a mesajelor, începând de la w [0] așa mai departe, apoi adăugați un bit '1' și '0' biți ca mai jos cu w [15] = lungimea datelor de intrare în biți (120 = 0x78)

	MSB..LSB		MSB..LSB		MSB..LSB		MSB..LSB
W[0]	54686973	W[4]	00000000	W[8]	00000000	W[12]	00000000
W[1]	20697320	W[5]	00000000	W[9]	00000000	W[13]	00000000
W[2]	59524331	W[6]	00000000	W[10]	00000000	W[14]	00000000
W[3]	35313780	W[7]	00000000	W[11]	00000000	W[15]	00000078

ETAPE DE CALCUL 3/4

Pasul 6:

Se calculează restul w [16] la w [63]

Pasul 7: Se declară variabilele de lucru (a-h) și se inițializează la valoarea hash curentă

```
a = h0    b = h1    c =  h2    d = h3    e = h4    f = h5    g = h6    h = h7
```

Pasul 8: Se creeaza funcția de compresie bucla principală

ETAPE DE CALCUL 4/4

Pasul 9: Se adăugă hashul comprimat la valoarea de hash curentă

Pasul 10: Se produce valoarea final hash
h0 add h1 add h2 add h3 add h4 add h5 add h6 add h7

Dacă pașii sunt executați corect, datele de intrare "Informaticag4" trebuie să prezinte valorea hash:

bce85afe672f46387186c7c817c09d2a07e1d8205b0cccc71b898b4393efb913
---ho--- ---h1--- ---h2--- ---h3--- ---h4--- ---h5--- ---h6--- ---h7---

Vă mulțumesc pentru atenție!