

KRIPTOGRAFIJA SA PRIMJENOM

VJEŽBA 6

LABORATORIJSKE VJEŽBE U MATLABU

Roko Rogulj

SECURE HASH ALGORITHM

SHA (eng. Secure Hash Algorithm) je vrsta kriptografske hash funkcije kojom ulaz proizvoljne veličine reduciramo na izlaz određene veličine. Zavisno o veličini izlaza i način obrade podataka, hash funkcije su podijeljene na četiri skupa : **SHA-0**, **SHA-1**, **SHA-2** i **SHA-3**. Skupovi se dalje dijele na standarde pa tako primjerice SHA-2 sadrži dva hash algoritma **SHA-256** i **SHA-512**. Razlika ova dva algoritma je u veličini njihovih izlaza (256 bitni i 512 bitni izlazi) te postoje male razlike u izvršavanju samih algoritama.

Hash funkcije **$H(x)$** generiraju reprezentativni otisak (fingerprint) ulaza stoga hash funkcije moraju zadovoljavati sljedeća svojstva:

- Hash funkcija se može primijeniti na poruku proizvoljne duljine
- Hash funkcija daje vrijednost fiksne duljine
- Jednostavno je izračunati hash vrijednost za proizvoljni ulaz **x**
- Nemoguće je pronaći ulaz funkcije **x** za proizvoljni **h** tako da **$H(x)=h$** (*One way property*)
- Za bilo koji **x** , praktično je nemoguće naći **y** tako da je **$H(x)=H(y)$**
- Praktično je nemoguće naći uređeni par **(x,y)** tako da je **$H(x)=H(y)$**

SHA-1

SHA-1 je hash funkcija koja dani ulaz pretvara u 160-bitnu vrijednost. Izbačen je iz upotrebe 2005. godine zbog slabe otpornosti na napade od jakih računala. Hash funkcije u ovoj vježbi će biti obrađene na primjeru SHA-1 zbog jednostavnosti algoritma. Ostale hash funkcije dijele slične principe rada gdje je glavna razlika u korištenim operacijama redukcije i veličini blokova nad kojima se operacije redukcije izvode. Ispod je dan primjer izvršavanja SHA-1 nad zadanim textom:

Input: 'Sveučilišni odjel za stručne studije'

SHA -1 (HEX) : 'e2e55d7915ca1b84664a1d7837bcf94623cdf535'

Minimalnom promjenom ulaza hash funkcije izrazito se mijenja izlaz (*avalanche effect*).

Input: 'Pveučilišni odjel za stručne studije'

SHA -1 (HEX) : 'a0b4b33cebe15efbf5f432bfb4bcbaaf45fe0dd5'

SHA-1 PSEUDOKOD

%Inicijalizacija varijabli

```
h0 = 0x67452301
h1 = 0xEFCDAB89
h2 = 0x98BADCFE
h3 = 0x10325476
h4 = 0xC3D2E1F0
m1 = duljina poruke (bit)
```

%Obrada ulazne poruke

Dodajte '0' na početak poruke tako da je ukupna duljina poruke djeljiva na dijelove od 512 bitova

%Obrada poruke po dijelovima od 512 bitova.

Rastavite poruku na segmente od 512 bitova

for i=0: broj segmenta od 512 bitova

 podijelite svaki segment u 16 dijelova od 32 bita

for i=0:16

%Inicijalizacija hash vrijednosti za ovaj blok

```
a = h0
b = h1
c = h2
d = h3
e = h4
```

%Glavna petlja

for i=0:79

if $0 \leq i \leq 19$

 f = (b **and** c) **or** ((**not** b) **and** d)

 k = 0x5A827999

else if $20 \leq i \leq 39$

 f = b **xor** c **xor** d

 k = 0x6ED9EBA1

else if $40 \leq i \leq 59$

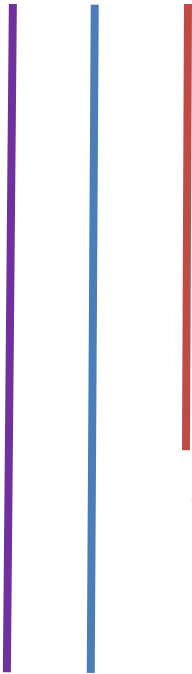
 f = (b **and** c) **or** (b **and** d) **or** (c **and** d)

 k = 0x8F1BBCDC

else if $60 \leq i \leq 79$

 f = b **xor** c **xor** d

 k = 0xCA62C1D6



$\text{temp} = (a \ll 5) + f + e + k + (\text{i-ti 32-bitni segment})$

$e = d$
 $d = c$
 $c = b \ll 30$
 $b = a$
 $a = \text{temp}$

Dodavanje vrijednosti na hash izlaz

$h0 = h0 + a$
 $h1 = h1 + b$
 $h2 = h2 + c$
 $h3 = h3 + d$
 $h4 = h4 + e$

Konačna vrijednost

$\text{HASH} = h0 \ll 128 \text{ or } h1 \ll 96 \text{ or } h2 \ll 64 \text{ or } h3 \ll 32 \text{ or } h4$

VJEŽBE U MATLABU

ZADATAK 1

Napišite program koji generira jednostavnu hash funkciju predstavljenu pseudokodom navedenim ispod. Neka ova hash funkcija za ulaz prima poruke veličine 64 bita te daje vrijednost izlaza od 20 bitova. Primjer:

PORUKA: '0x3C76284A8115ADCB'

HASH: '0x55003'

PSEUDOKOD

%Inicijalizacija varijabli

h0 = 0x6
h1 = 0xE
h2 = 0x9
h3 = 0x1
h4 = 0xC

%Obrada poruke po dijelovima od 4 bita.

Rastavite poruku **m** na 16 segmenata od 4 bita **m[i]**

%Inicijalizacija hash vrijednosti

a = h0
b = h1
c = h2
d = h3
e = h4

%Glavna petlja

for i=1:16

```
    if 0 < i ≤ 4
        f = (b and c) or ((not b) and d)
        k = 0x5
    else if 4 < i ≤ 8
        f = b xor c xor d
        k = 0x6
    else if 8 < i ≤ 12
        f = (b and c) or (b and d) or (c and d)
        k = 0x8
    else if 12 < i ≤ 16
        f = b xor c xor d
        k = 0xC
```

```
temp = (a << 1) + f + e + k + m(i)
```

```
e = d
```

```
d = c
```

```
c = b << 1
```

```
b = a
```

```
a = temp
```

Dodavanje vrijednosti na hash izlaz

```
h0 = h0 + a
```

```
h1 = h1 + b
```

```
h2 = h2 + c
```

```
h3 = h3 + d
```

```
h4 = h4 + e
```

Konačna vrijednost

```
HASH = h0 <<16 or h1 << 12 or h2 <<8 or h3 <<4 or h4
```

ZADATAK 2

Unaprijedite algoritam iz prethodnog zadatka tako da prima poruke proizvoljne veličine. Program treba na početak poruke dodati '0' tako da je broj bitova poruke djeljiv s 64.

KORISNE FUNKCIJE

bitand(a,b) – funkcija koja daje vrijednost bit and operacije nad varijablama ***a*** i ***b***

```
>> bitand(5,8)
```

```
ans = 0
```

bitor(a,b)

bitxor(a,b)

bitshif(a,b) – pomak bitova vrijednost ***a*** u lijevo za ***b*** (***a << b***)

append(a,b,c,d...) – spajanje više tekstualnih vrijednosti u jednu

```
>> append ('hash','funkcija')
```

```
ans = 'hashfunkcija'
```

hex2dec() – pretvaranje tekstualne heksadecimalne vrijednosti u decimalnu

dec2hex() – pretvaranje decimalne vrijednosti u heksadecimalnu