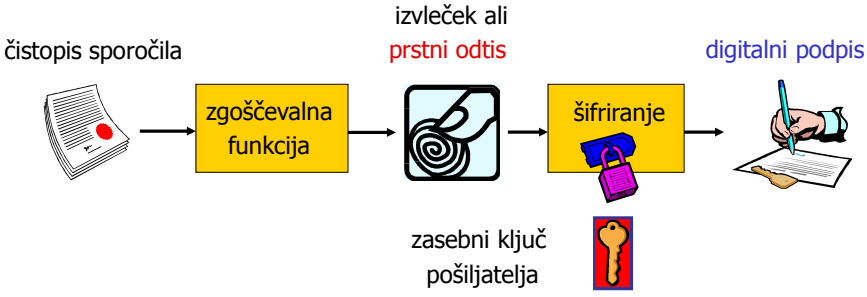
Varne komunikacije

Zgoščevalne funkcije

- Digitalni podpis in prstni odtis
- Zgoščevalne funkcije
  - MDC in MAC
  - razredi MDC
- Zgoščevalna funkcija na osnovi DES
- MD4
- MD5
- SHA-1, SHA-2 ...

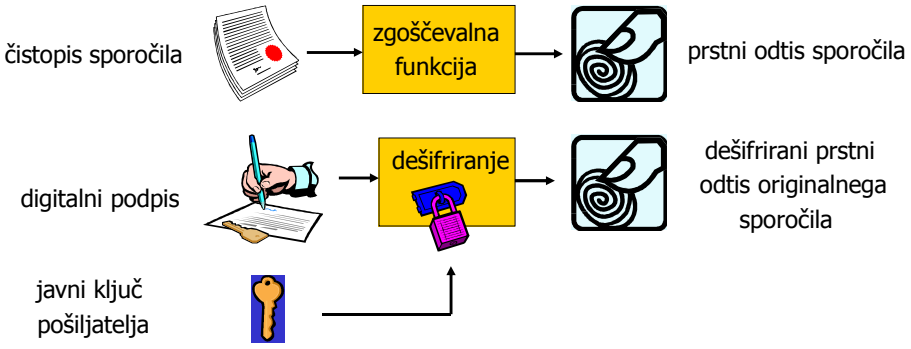
Digitalni podpis

- Digitalni podpis je s tajnim ključem šifrirani prstni odtis sporočila:



- Zgoščevalna funkcija je enosmerna funkcija in vsaka sprememba čistopisa spremeni tudi prstni odtis sporočila.
- Napadalec bi lahko spremenil sporočilo in dodal nov prstni odtis !
- Pošiljatelj zaščiti prstni odtis s šifriranjem!

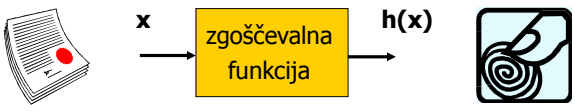
Preverjanje digitalnega podpisa



- Prejemnik preveri ujemanje prstnih odtisov in če sta enaka
  - je sporočilo verodostojno,
  - potrjena je identiteta pošiljatelja in
  - če velje oboje, potem pošiljatelj ne more zanikati sporočila.

Zgoščevalna funkcija

- Zgoščevalna funkcija (hash function) preslika poljubno dolgo sporočilo v blok podatkov končne dolžine. Izvleček (digest) imenujemo tudi prstni odtis (digital fingerprint) sporočila.
- Zgoščevalna funkcija je enosmerna funkcija.
- Verjetnost, da najdemo sporočilo z enakim prstnim odtisom mora biti zelo majhna **Pr(** $h(x_1)=h(x_2)$ **) --> 0** .



čistopis

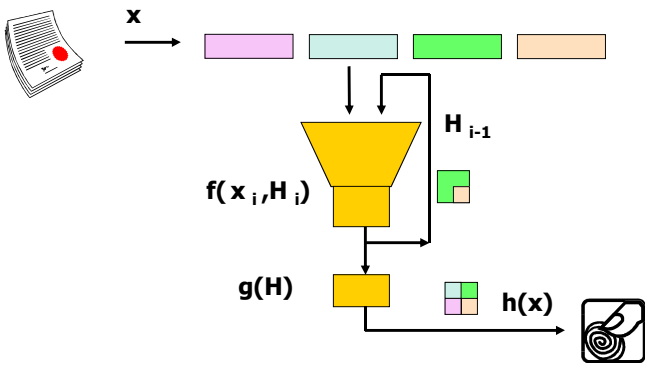
izvleček ali prstni odtis

Funkcionalna delitev zgoščevalnih funkcij

- MDC zgoščevalne funkcije (modification detection codes) omogočajo prepoznavo sprememb v sporočilu . Ključa ne potrebujemo .
- MAC zgoščevalne funkcije (message autentication codes) zagotavljajo verodostojnost sporočila in avtentičnost pošiljatelja. Zgoščevalna funkcija uporablja tajni ključ zato se uporablja za MAC tudi ime keyed hash function.

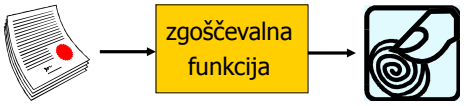
Model iteracijske zgoščevalne funkcije

- Sporočilo razdelimo na bloke dogovorjene dolžine.
- Postopek zgoščevanja ponavljamo in vsakič uporabimo izvleček predhodnih blokov.



Razredi MDC zgoščevalnih funkcij

- MDC zgoščevalne funkcije razdelimo v tri razrede:
  - zgoščevalne funkcije, ki uporabljajo blokovne šifrirne postopke; (npr. DES - Davies-Meyer)
  - zgoščevalne funkcije, ki uporabljajo modularno aritmetiko (MASH 1 - Modular Arithmetic Secure Hash - algorithm 1);
  - posebej prilagojene (namenske) zgoščevalne funkcije odlikuje mnogo manjša računska zahtevnost; (najbolj znane so MD4, MD5, SHA1)



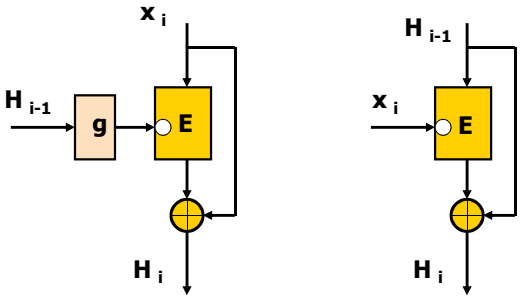
čistopis

izvleček ali prstni odtis

Zgoščevalna funkcija z blokovno šifro

- Množica zgoščevalnih funkcij temelji na enem od blokovnih šifrirnih postopkov, pogosto na DES-u.
- primer sta zgoščevalni funkciji z blokovno šifro:

(Matyas-Meyer-Oseas) (Davies-Meyer)



Zgoščevalne funkcije MD

- Namenske MDC zgoščevalne funkcije so računsko bistveno bolj učinkovite od blokovnih. Najbolj pogoste funkcije temeljijo na MD algoritmu, ki ga je razvil Ron Rivest:
  - MD2 (message digest algorithm 2) , 1989
  - MD4 (message digest algorithm 4), 1990
  - MD5 (message digest algorithm 5), 1991
  - SHA - 1 (Secure Hash Algorithm 1) je bil razvit v NITS v sodelovanju z NSA in objavljen 1994
- Pogosto sta uporabljena zgoščevalna algoritma MD5 in SHA-1:
  - dolžina sporočil je "omejena" na 2^{64} bitov.
  - MD5 je računsko manj zahteven od SHA-1 (razmerje hitrosti 7 : 3)
  - algoritem SHA-1 ima daljši izvleček (160bit vs 128 bit)
  - algoritem SHA-1 je del standarda (DSS)
- Danes se uporablja varnejši SHA-2 , dolžina je 256-512 bitov
- Na poti je še novejši SHA-3 !!

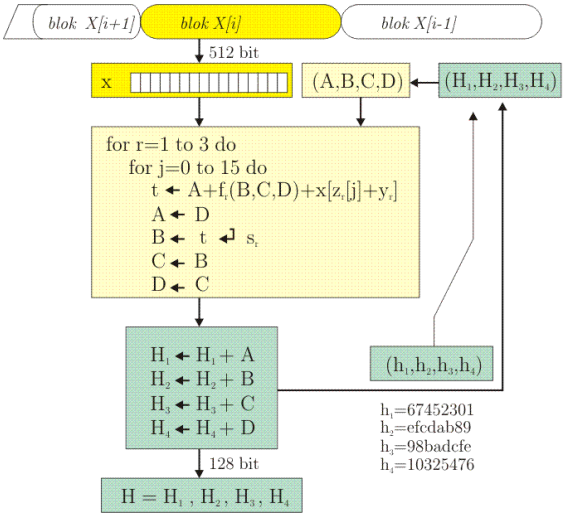
Posodobitve SHA-1, 2, ..

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)	Rounds	Operations	Collisions found	Example Performance (MiB/s)[11]
SHA-0		160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rot	Yes	-
SHA-1		160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rot	Theoretical attack (2^{51})[12]	153
SHA-2	*SHA-256/224*	256/224	256	512	$2^{64} - 1$	32	64	+,and,or,xor,shr,rot	None	111
	SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80	+,and,or,xor,shr,rot	None	99

MD4

- b - bitno sporočilo ***X*** razdelimo na 512- bitne bloke *X[i]*
- izvleček ***H*** ima dolžino 128 bitov (štiri 32- bitne besede)
- zgoščevanje vsakega bloka poteka v treh krogih
- v vsakem krogu uporabimo
  - različne funkcije: *f* , *g* , *h*
  - različni vrstni red branja besed v bloku *z(x)*
  - različne rotacije *s(x)*
- delni izvleček *H[i]* uporabimo pri obdelavi naslednjega bloka, v zadnjem bloku pa je to hkrati izvleček celotnega sporočila *H*

MD4 algoritem



$f_1 = f(B,C,D) = BC \vee \overline{B}C$
$f_2 = g(B,C,D) = BC \vee BD \vee CD$
$f_3 = h(B,C,D) = B \text{ xor } C \text{ xor } D$

$z_1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]$
$z_2 = [0,4,8,12,1,5,9,13,2,6,10,14,3,7,11,15]$
$z_3 = [0,8,4,12,2,10,6,14,1,9,5,13,3,11,7,15]$

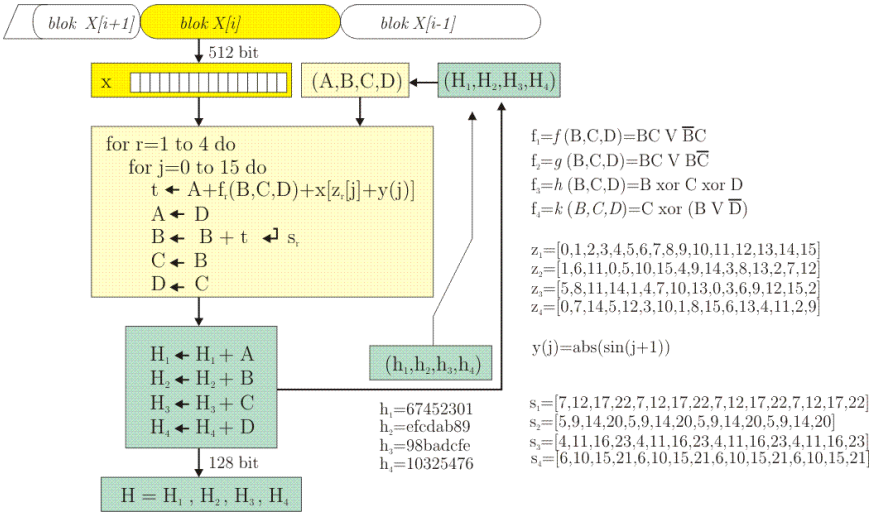
$y_1 = 0$
$y_2 = 5a827999$
$y_3 = 6ed9eba1$

$s_1 = [3,7,11,19,3,7,11,19,3,7,11,19,3,7,11,19]$
$s_2 = [3,5,9,13,3,5,9,13,3,5,9,13,3,5,9,13]$
$s_3 = [3,9,11,15,3,9,11,15,3,9,11,15,3,9,11,15]$

MD5

- b - bitno sporočilo ***X*** razdelimo na 512- bitne bloke *X[i]*
- izvleček ***H*** ima dolžino 128 bitov (štiri 32- bitne besede)
- zgoščevanje vsakega bloka poteka v štirih krogih
- v vsakem krogu uporabimo
  - različne funkcije: *f* , *g* , *h* , *e*
  - različni vrstni red branja besed v bloku *z(x)*
  - različne rotacije *s(x)*
- delni izvleček *H[i]* uporabimo pri obdelavi naslednjega bloka, v zadnjem bloku pa je to hkrati izvleček celotnega sporočila *H*

MD5 algoritem



Zgoščevalne funkcije na zgledu

testno sporočilo x : izvleček h(x):

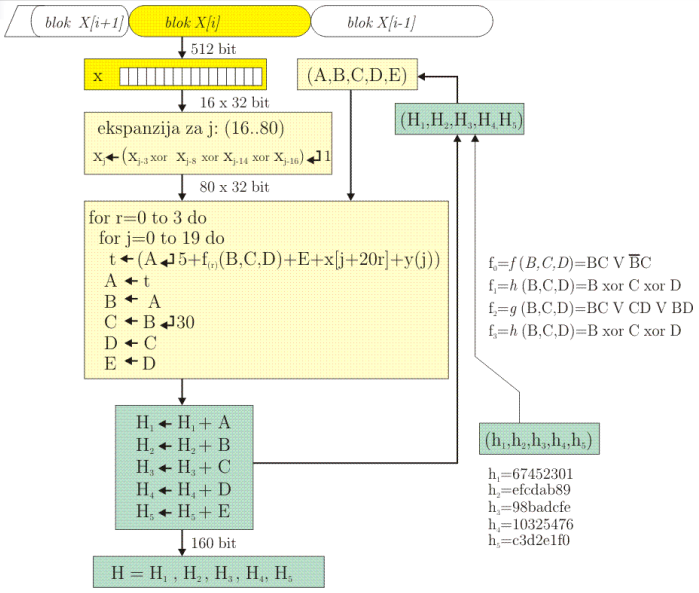
MD4	“” “a” “abc” “abcdefghijklmnopqrstuvwxyz”	31d6cfe0d16ae931b73c59d7e0c089c0 bde52cb31de33e46245e05fbdbd6fb24 a448017aaf21d8525fc10ae87aa6729d d79e1c308aa5bbcdeea8ed63df412da9
MD5	“” “a” “abc” “abcdefghijklmnopqrstuvwxyz”	d41d8cd98f00b204e9800998ecf8427e 0cc175b9c0f1b6a831c399e269772661 900150983cd24fb0d6963f7d28e17f72 c3fcd3d76192e4007dfb496cca67e13b
SHA-1	“” “a” “abc” “abcdefghijklmnopqrstuvwxyz”	da39a3ee5e6b4b0d3255bfef95601890afd80709 86f7e437faa5a7fce15d1ddcb9eaeaea377667b8 a9993e364706816aba3e25717850c26c9cd0d89d 32d10c7b8cf96570ca04ce37f2a19d84240d3a89

SHA-1

- b - bitno sporočilo ***X*** razdelimo na 512- bitne bloke *X[i]*
- blok 16 32-bitnih besed ekspandiramo na 80 besed (2560 bitov)
- zgoščevanje ekspandiranega bloka poteka s štirimi različnimi funkcijami: *f* , *g* , *h* , *e*
- delni izvleček *H[i]* uporabimo pri obdelavi naslednjega bloka, v zadnjem bloku pa je to hkrati izvleček celotnega sporočila *H*
- izvleček ***H*** ima dolžino 160 bitov (pet 32- bitnih besed)

- SHA-1 (Secure Hash Algorithm 1) je posodobitev prvotnega algoritma SHA

SHA-1 algoritem



Digitalni podpis

- Digitalni podpis je šifrirani izvleček sporočila. Potrebujemo ustrezno zgoščevalno funkcijo in asimetrični postopek šifriranja.
- Za digitalni podpis zelo pogosto uporabljamo kombinacijo:
  - zgoščevalna funkcija SHA-1
  - asimetrični šifrirni algoritem RSA
  - V standardnem formatu digitalnega certifikata X-509 je v enem polju zapisan tudi tip zgoščevalne funkcije in šifrirni postopek. V spletnem potrdilu sigen-ca je algoritem podpisa sha1RSA .
- NITS je leta 1991 postavil standard za digitalni podpis DSS.
- DSS predpisuje
  - zgoščevalno funkcijo SHA-1 in
  - šifrirni algoritem DSA (Digital Signature Algorithm).
- DSA je poseben primer ElGamal algoritma in varnost prav tako temelji na težavnosti računanja diskretnega logaritma.

DSA generacija ključev

- števila (p,q,g) so lahko skupna za več uporabnikov:
- izberemo veliko praštevilo p
  - dolžina zapisa L je od 512 do 1024, L je mnogokratnik 64
- število q izberemo tako, da je p-1 deljiv z q (160-bitno število)
- generiramo število $g=h^{(p-1)/q} \bmod p$,
  - izberemo h<p-1 , veljati mora g>1
- javni ključ y generiramo s pomočjo naključno izbranega tajnega ključa x:

  $y= g^x \bmod p$
- javni ključ sestavlja (p,q,g,y)
- tajni zasebni ključ je (p,q,g,x)

DSA digitalni podpis

- Pošiljatelj A želi podpisati sporočilo m
- H(m) je izvleček sporočila m (SHA-1, 160 bit)
- A izbere naključno število k<q in s pomočjo tajnega ključa (p,q,g,x) izračuna dvodelni digitalni podpis:

  $r=(g^k \bmod p) \bmod q$

  $s=(k^{-1} (H(m)+x\ r)) \bmod q$
- A pošlje podpisano sporočilo (m, r, s) prejemniku B
- Prejemnik B ima tudi dostop do javnega ključa pošiljatelja (p,q,g,y) in izračuna verifikacijsko sporočilo v po korakih:

  $w=s^{-1} \bmod q$

  $u_1=(H(m)\ w) \bmod q$

  $u_1=(r\ w) \bmod q$

  $v =((g^{u1}\ y^{u2}) \bmod p) \bmod q$
- Prejemnik B preveri, če velja v=r ?

DSA na zgledu

- Generirana so števila
  - p=1291,
  - q=215, p-1=6 q
  - g=1003 , (h=1230)
- Pošiljatelj A želi podpisati izvleček sporočila H(m)=1155
- A izbere naključno število k=973 in izračuna podpis:

  $r=(g^k \bmod p) \bmod q = 52$

  $s=(k^{-1} (H(m)+x\ r)) \bmod q = 118$
- A pošlje sporočilo in podpis (r=52, s=118) prejemniku B
- Prejemnik B izračuna H(m) in verifikacijsko sporočilo v po korakih:

  $w=s^{-1} \bmod q = 82$

  $u_1=(H(m)\ w) \bmod q = 110$

  $u_1=(r\ w) \bmod q = 179$

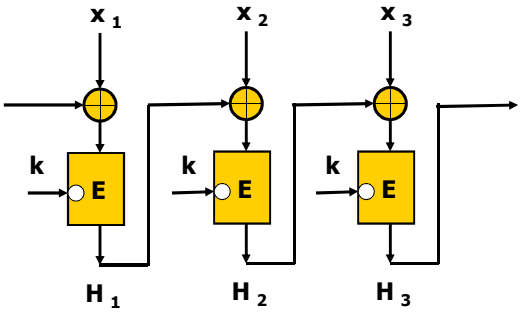
  $v = ((g^{u1}\ y^{u2}) \bmod p) \bmod q = 52$
- Prejemnik B potrdi veljavnost sporočila, saj velja v=r=52

Zgoščevalne funkcije s ključem (MAC)

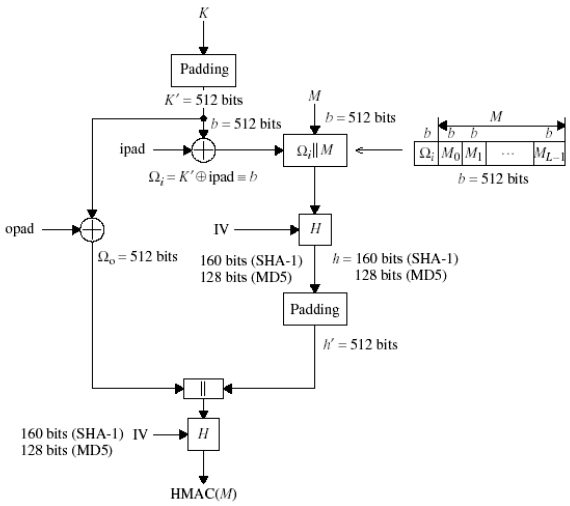
- Namen zgoščevalnih funkcij s ključem (keyed hash functions) je avtentikacija sporočila, ki nam zagotavlja integriteto sporočila in avtentikacijo izvora za datoteke poslane med dvema uporabnikoma (primer uporabe je SSL);
- Poznamo dve vrsti MAC funkcij:
  - MAC funkcije, ki temeljijo na bločnih šifrah (CBC-MAC)
  - MAC funkcije ki temeljijo na MD funkcijah. V praksi se v glavnem uporabljata dve vrsti teh funkcij:
    - HMAC–SHA-1 in
    - HMAC-MD5
    - Glavna razlika v primerjavi z bločnimi MAC funkcijami je prilagoditev funkcije stiskanja, ki zavisi od ključa k. Vse posredovalne iteracije vključujejo skrivni ključ. To zagotavlja dodatno varnost v primeru odkritja slabosti osnovne zgoščevalne funkcije

MAC zgoščevalne funkcije z bločno šifro

- Avtentikacijske zgoščevalne funkcije s ključem (MAC) pogosto uporabljajo veriženje blokov - CBC (Cipher Block Chaining).
- Šifrirni algoritem je lahko DES:



HMAC



$$HMAC = H\left[\left(K \oplus opad\right) \square H\left[\left(K \oplus ipad\right) \square M\right]\right]$$