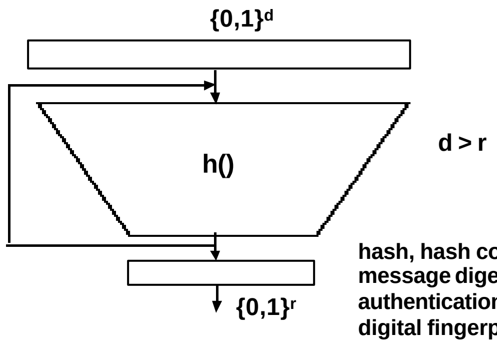
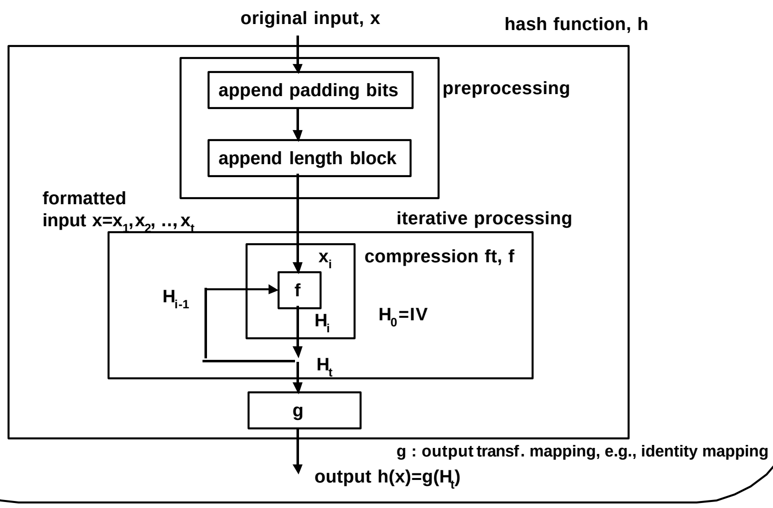
Hash function

- ❑Compress a binary string with an arbitrary length into a fixed short message
- ❑ Used for digital signature, integrity, authentication etc.



hash, hash code/value/result
message digest,checksum,MIC,
authentication tag, seal,compression
digital fingerprint, imprint

Configuration of Hash Function



Requirements of Hash function

- Compression
- One-wayness
  : If y=h(x) is given, it is computational infeasible to compute x
- Collision-free
  : It is computational infeasible to find a pair (x, x'), x^{1} x' satisfying h(x)=h(x').
- Efficiency
  – Easy to compute f(x) for a given x.

Classification of Hash ft

- Keyed hash : MAC (Message Authentication Code)
- Unkeyed hash : MDC (Manipulation Detection Code),
  – 1WHF(One Way Hash Function)
  – CFHF(Collision-Free Hash Function)

Unkeyed Hash Function

- h() must be publicly known and not require any secret information (extension of Kerckhoff’s principle)
- 1-way [Merkle] :
  computational infeasible
  to find x ¹ x’ s.t. h(x)=h(x’), |h(x)| ³ 64bit
  Þ weakly collision-free, weak 1WHF
- Collision-intractable [Damgard] :
  computational infeasible to find
  (x,x’) s.t. x ¹ x’ and h(x)=h(x’),|h(x)| ³ 128 bit
  Þ strongly collision-free, strong 1WHF

Keyed Hash Function

- H() must be publicly known and the only secret info. lies the key.(extension of Kerckhoff’s principle)
- |h(x,k)|=n, 32 £ n £ 64 bit
- Given x,k, hard to find h(x,k) with prob. $1/2^n$ without k.
- Hard to find k or to compute h(x’,k) for any x ¹ x’ even large set of pairs {x_i, h(x_i,k)} is known.

MAC forgery

- Universal forgery : adversary can find the equivalent algorithm as MAC function
- Selective forgery : adversary can create a pair of new text-MAC.
- Existential forgery : Even if adversary can’ t adjust the value of text, he can create a pair of new text-MAC.

Birthday Paradox

- Probability that 2 persons have the same birthday among r persons : p_r
- (Assumption) each birthday is independent and uniform in the range 1 to m.

  $p_r = 1-(m)^r/m^r = 1- m!/m^r(m-r)!$

  » Ö$e^{-r^2/(2m)}$

  where, $(m)^r$ = m(m-1) ..(m-r+1)
- if r= Öm, p_r » 0.5 e.g., m=365,r=23, p_r>0.5

  Û n-bit hash function will collide with probability 0.5 after Ö(2^n) times operation

Design Criteria

- All input value must affect to compute the hashed value.

(Ex) Crytanalysis of Snefru

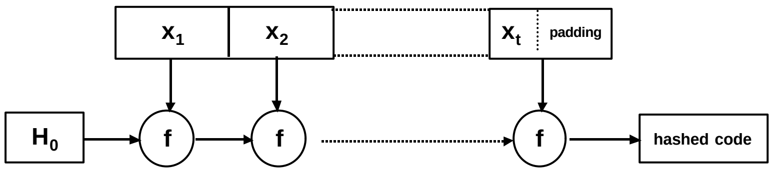
- No trapdoor
- The length of hashed value must be greater than 128 bit Þ guarantee breaking complexity 2^{64} by brute force attack.
- Maximum error propagation from input to output.

Classification

- Using block cipher
  - -Matyas-Meyer/Davies-Meyer scheme
  - – Merkle' s meta scheme
- Using modular operations
  - – quadratic congruential
- Dedicated hash functions
  - – MD2,MD4,MD5, SHA-1
  - – RIPE-MD, HAVAL, Snefru, N-hash

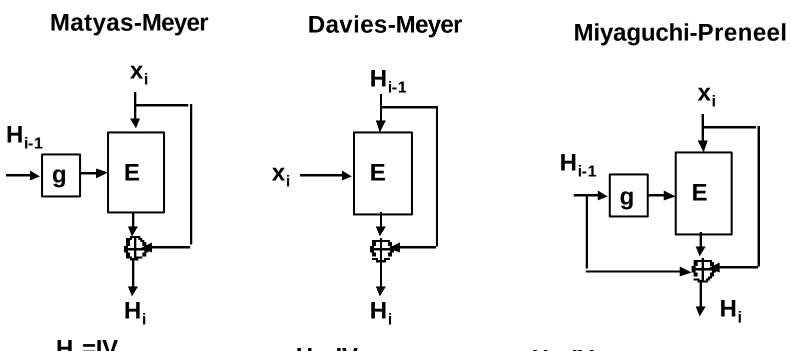
Meta scheme

- $H_0=IV$, $H_i=f(H_{i-1},x_i)$, $1 \le i \le t$, $h(x)=H_t$



f : h' s primitive hash function
H_i : connection variable from i-1 to i

Hash ft by block ciphers



Matyas-Meyer: $H_0=IV$, $H_i=E_{g(H_{i-1})}(x_i) \oplus x_i$

Davies-Meyer: $H_0=IV$, $H_i=E_{x_i}(H_{i-1}) \oplus H_{i-1}$

Miyaguchi-Preneel: $H_0=IV$, $H_i=E_{g(H_{i-1})}(x_i) \oplus x_i \oplus H_{i-1}$

Hash by modular operation

- **Quadratic Congruential**
  - $H_i = (x_i + H_{i-1})^2 \bmod N, H_0=0$
  - where N=Mersenne prime 2^{31}-1
- $H_i = (x_i$ Å $H_{i-1})^2 \bmod N$ Å x_i
- $H_i = (x_i$ Å $H_{i-1})^e \bmod N$

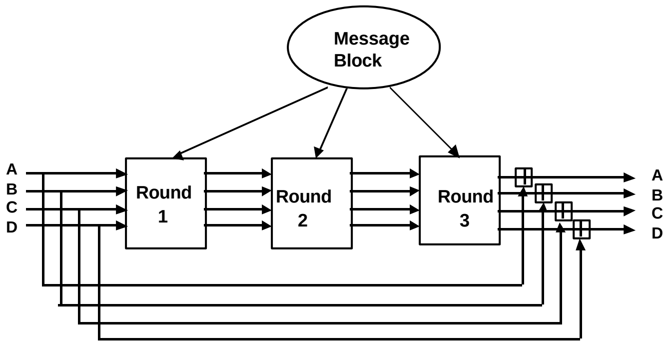
MD4(I)

- **Preprocessing a message, x**

1. padding : d =447 -(|x| mod 512)
2. Length of a message : n= |x| mod 2^{64},|n|= 64 bit
3. M = x ||1||0^d||n Ü multiple of 512
   ||: concatenation

* little-endian : $W=2^{24}B_4+2^{16}B_3+2^8B_2+B_1$
(B_1: lowest address)

MD4(II)



MD4(III)

1. A=67452301$_h$, B=efcdab89$_h$, C=98badcfe$_h$, D=10325476$_h$

2. for i=0 to N/16 -1 do (N mod 16=0)

3. for j=0 to 15 do

X[j] =M[16i+j] (M[i] : 32 bit string)

4. AA=A, BB=B, CC=C, DD=D

5..7. Round 1(for j=0..15), Round 2(for j=16..31), Round 3(j=32..47)

8. A=A+AA, B=B+BB, C=C+CC, D=D+DD

where + is modular addition over 2^{32}.

9. output A||B||C||D||

Round 1 in MD4

1. A=(A+f(B,C,D)+X[0])<<<3
2. D=(D+f(A,B,C)+X[1])<<<7
3. C=(C+f(D,A,B)+X[2])<<<11
4. B=(B+f(C,D,A)+X[3])<<<19
5. A=(A+f(B,C,D)+X[5])<<< 3

   .

   .

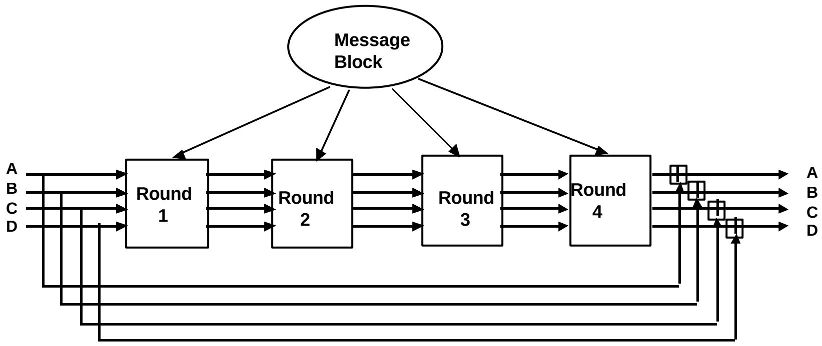
16. B=(B+f(C,D,A)+X[15])<<<19

where, f(X,Y,Z) = (X Ù Y) Ú ((ØX) Ù Z) , Ú: OR, Ù: AND,
Ø:complement, <<<s : circular left rotate by s

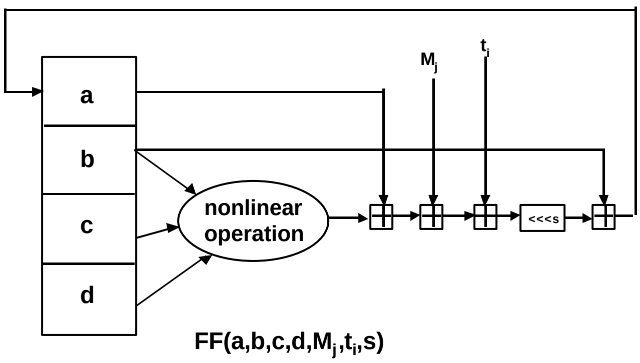
MD5(I)

- Add 4-th round
- Use unique constant per each step
- g ft in 2 round: change from symmetric ft (XÙY) v (XÙZ) v (YÙZ) to non-symmetric ft (XÙZ) v (YÙ(ØZ))
- Each step is added to the output of a previous step to achieve avalanche effect as earlier as possible.
- Change the value of rotation in round ft.

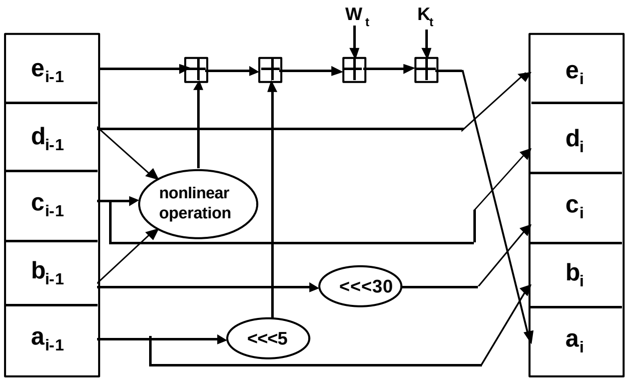
MD5(II)



MD5' s primitive ft



SHA(I)



FF(a,b,c,d,M_j,t_i,s)

SHA(II)

- 160 bit hashed value
- 4 round hash, each round has 20 step
- Change internal primitive ft
- big-endian

Summary

name	designer	year	charateristics	security
MD4	R.L.Rivest (USA)	‘91	Boolean ft 3R, 128bit	collision (‘95) 2^{20} operation
MD5	R.L.Rivest (USA)	‘92	Boolean ft 4R, 128bit	primitive ft’s collision(‘96)
HAVAL	Y.Zheng (Australia)	‘92	expand MD5 3,4,5R/128,160,192,224,256bit	
SHS	NIST	‘91	Boolean ft Modified MD4, 4R,160bit	
HAS -160	KISA (Korea)	‘98	Boolean ft 160bit	

Performance

486SX(33MHZ)

Algorithm	Length	Speed (Kb/s)
Davies-Meyer with DES	64	9
HAVAL (3 pass)	variable	168
HAVAL (4 pass)	variable	118
HAVAL (5 pass)	variable	95
MD2	128	23
MD4	128	236
MD5	128	174
N-Hash(12 round)	128	29
N-Hash(15 round)	128	24
RIPEMD	128	182
SHA	160	75

Application

- Used together with a signature scheme
- Integrity service for MIC (Message Integrity Code) (Ex: anti-virus)
- passwd ft in UNIX OS
- Keyed Hash Ft (MAC)
- Identification in Challenge-response protocol