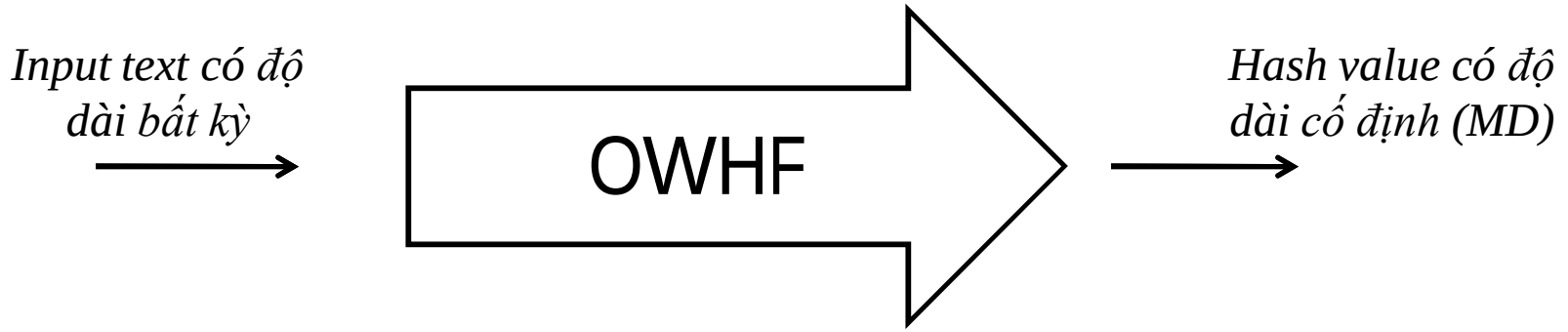
CHƯƠNG 5
HÀM BĂM MỘT CHIỀU VÀ CÁC THUẬT GiẢI CHỮ KÝ SỐ

5.1 .Hàm băm một chiều (One Way Hash function)

5.1.1. Khái niệm

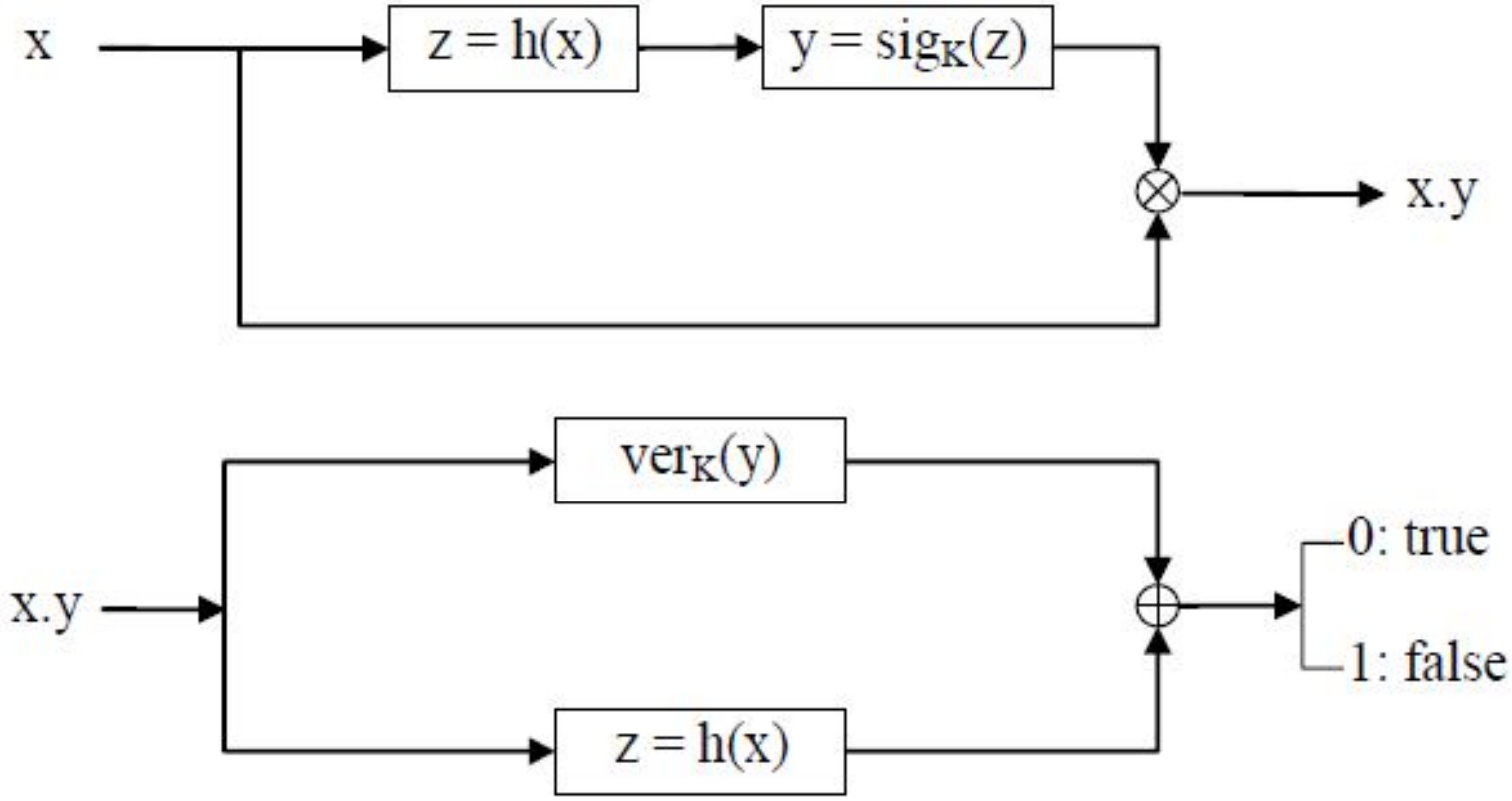
- ONE-WAY-HASH được sử dụng rộng rãi trong việc bảo mật , chứng thực các văn bản điện tử như chữ ký điện tử , “lấy dấu tay” và kiểm tra toàn vẹn dữ liệu.
- ONE-WAY-HASH tạo một trị số băm (hash value) có độ dài cố định từ một văn bản có độ dài bất kỳ.
- Bài toán ngược lại hầu như không thể thực hiện được.Hai văn bản khác nhau không thể có cùng trị số băm và mỗi văn bản chỉ có thể có một trị số băm duy nhất.Còn gọi là hàm lấy dấu tay hay Message Digist (MD).
- Ta sẽ phân tích kỹ thuật toán SHA-1.Đây là thuật toán được sử dụng rộng rãi và được coi là một chuẩn của hàm ONE-WAY-HASH .



Hình 5.1 Sơ đồ OWHF

- MD5 - MD có độ dài 128 bit
- SHA1 - MD có độ dài 160 bit
- SHA2 - MD có độ dài 256/512 bit

5.1.2. Sơ đồ DS dùng OWHF



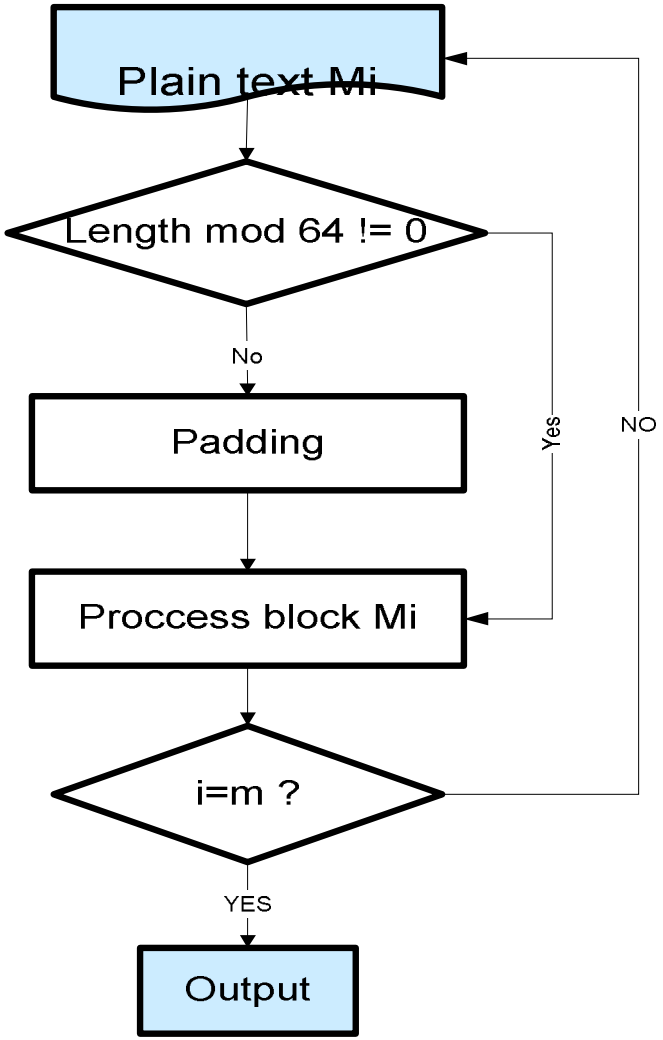
Hình 5.2: Sơ đồ chữ ký sử dụng hàm Băm

5.1.3. SHA1

1. Giới thiệu chung

- Thuật toán SHA-1 được sử dụng để tính bản tóm lược –còn gọi là trị số băm (MD) hoặc “dấu tay” của một văn bản có độ dài thay đổi.
- Khi văn bản có độ dài $\leq 2^{64}$ thuật giải SHA-1 sẽ tạo ra một trị số băm có độ dài 160 bit.

2. Sơ đồ SHA1


Plain text Mi
Length mod 64 != 0
No
Yes
NO
Padding
Proccess block Mi
i=m ?
YES
Output

3.Một số định nghĩa

a. Chuỗi bit và các số nguyên :

- Các số hexa nằm trong tập [0,1...,F], được biểu diễn bởi 4 bit ví dụ 6 = 0110 ; A =1010.
- Mỗi từ 32 bit được thể hiện bởi 4 chuỗi 8 bit , mỗi số 8 bít sẽ gồm hai số hexa. Ví dụ :

1110 1010 0110 1100 1101 0010 1111 1011 = EA6CDFB

- Mọi số nguyên từ 0 đến (2^{32})-1 đều có thể biểu diễn thành các từ 8 bit mỗi từ 8 bít gồm hai số hexa

  Ví dụ $291 = 2^8 + 2^5 + 2^1 + 2^0 = 256+32+2+1 = 0000\ 0001\ 0010\ 0011$ được biểu diễn thành $(00000123)_H$ trong đó mỗi số là các số hexa.
- Khối 512 bit sẽ gồm 16 từ 32 bit.

b.Các phép toán trong giải thuật SHA1

- Các phép toán logic (X,Y là các từ 32 bit)
  - X ^ Y = bitwise logical "and"
  - X $\vee$ Y = bitwise logical "inclusive-or".
  - X XOR Y = bitwise logical "exclusive-or" .
  - ~ X = bitwise logical "complement" .
- Phép dịch trái : S^n(X) = (X << n) OR (X >> 32-n).

  Trong đó X là một từ 32 bít và n là một số nguyên dương $0 \leq n \leq 32$.
- Chèn bít : Mục đích của việc chèn bít là tạo ra khối bít M_i có độ dài là bội của 512 bit.Để thực hiện được điều này ta thêm các bít “1” hoặc “0” vào sau các từ còn thiếu.

c. Chèn bít

- Chèn bít được sử dụng khi độ dài input block < 448 bit.
- Kỹ thuật chèn "bit" được mô tả như sau:

Vi du : Giả sử khối M_i có độ dài l = 40 bit

0110 0001 0110 0010 0110 0011 0110 0100 0110 0101

1. Chèn "1" ta có

0110 0001 0110 0010 0110 0011 0110 0100 0110 0101 1

l = 40 ,ta chèn 407 các bít "0" vào vị trí từ 41 → l= 448.

(61626364 65800000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000)$_{Hexa}$

2. Biểu diễn độ dài gốc l dưới dạng hai từ 32 bit (64 bít)

Ví dụ : Với l = 40 (l phải được tính trước khi chèn) biểu diễn của 40 dưới dạng 2 từ 32 bít sẽ là (0000000000000028)$_{HEXA}$.

Như vây thông điệp sau khi chèn sẽ là :

61626364 65800000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000028.

•

d. Các hàm trong SHA1

i . Các hàm logic

Mỗi hàm f(t), $0 \le t \le 79$, làm việc với 3 từ 32 bít (BCD)và tạo ra một từ 32-bit ở đầu ra. Hàm f(t)(B,C,D) được định nghĩa :

- For $(0 \le t \le 19)$ $f_t(B,C,D)$ =(B AND C)OR((NOT B) AND D)
- For $(20 \le t \le 39)$ $f_t(B,C,D)$ = B XOR C XOR D
- For $(40 \le t \le 59)$ f_t (B,C,D)=(B AND C)OR(B AND D) OR(C AND D)
- For ($60 \le t \le 79$) f_t (B,C,D) = B XOR C XOR D

ii. Các hằng số

- Hằng số dùng trong SHA-1 là các từ K(0), K(1), ... , K(79) được biểu diễn dưới dạng HEXA

$K(t) = 5A827999 \ (0 \leq t \leq 19)$

$K(t) = 6ED9EBA1 \ (20 \leq t \leq 39)$

$K(t) = 8F1BBCDC \ (40 \leq t \leq 59)$

$K(t) = CA62C1D6 \ (60 \leq t \leq 79).$

4.Tính MESSAGE DIGEST

i . Tạo hai buffer. Mỗi buffer gồm 5 từ 32 bit (A,B,C,D,E) và (H_0, H_1, H_2, H_3, H_4).Tạo một chuỗi 80 từ 32 bít (W_0,W_1,...,W_{79}) và một buffer đơn lẻ TEMP được sử dụng để hỗ trợ.

ii . Trước khi xử lý từng khối M_i,(H_i) được nạp trước các véc tơ khởi tạo sau:

H_0 = 67452301 H_3 = 10325476

H_1 = EFCDAB89 H_4 = C3D2E1F0.

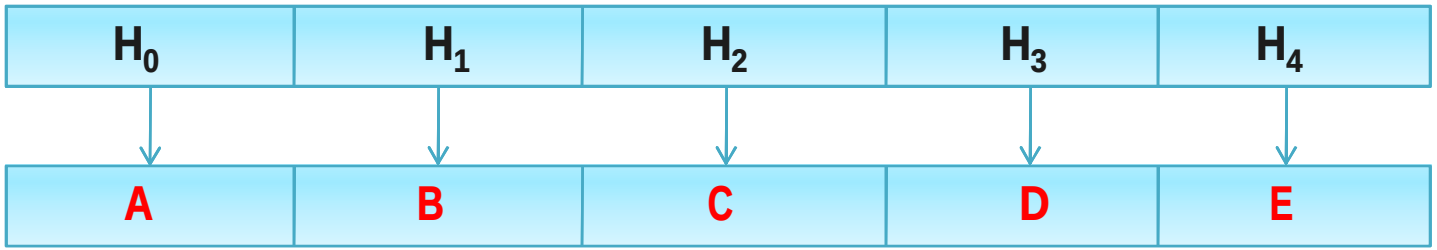
H_2 = 98BADCFE

iii .Xử lý các khối $M_1, M_2, ... M_i ... , M_n$

- Input text được mở rộng từ 16 word 32-bit (M_0 đến M_{15}) thành 80 word 32-bit (W_0 đến W_{79}) bằng việc sử dụng thuật toán mở rộng:
  - $W(t) = M(t)$ với $0 \le t \le 15$
  - For t = 16 to 79 do :

$W(t) = S^1(W(t-3)$ XOR $W(t-8)$ XOR $W(t-14)$ XOR $W(t-16))$.

- Set $A = H_0$, $B = H_1$, $C = H_2$, $D = H_3$, $E = H_4$.



iii. Xử lý một vòng lặp

For t = 0 to 79

{

TEMP = S^5(A) + f_t(B,C,D) + E + W_t + K_t;

E = D ;

B = A ;

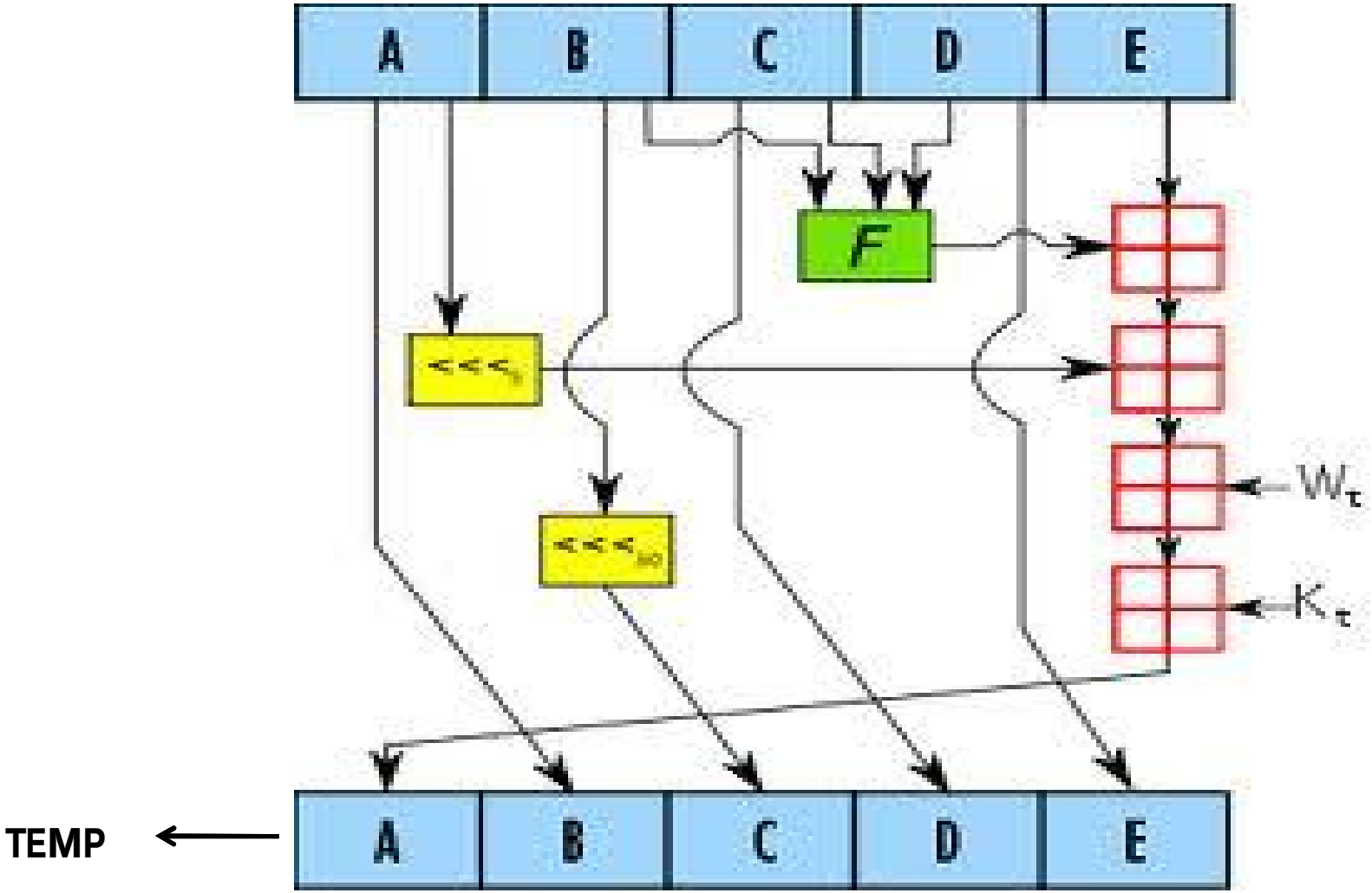
D = C ;

C = S^30(B);

A = TEMP;

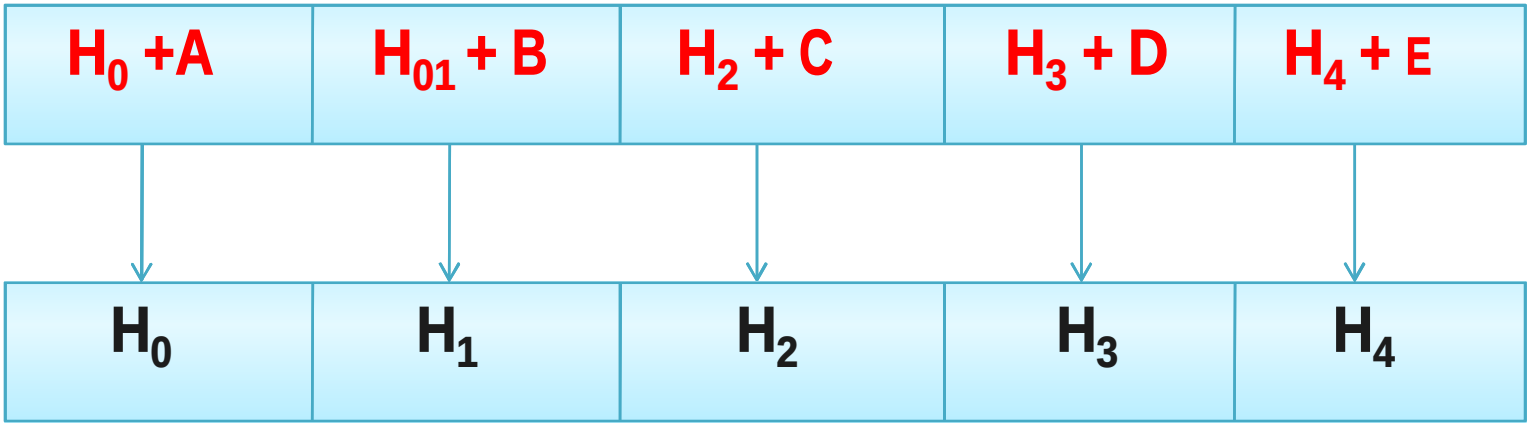
}

Sơ đồ một vòng lặp cuả SHA1



iv.Xử lý output

- Set $H_0 = H_0+A$, $H_1 = H_1+B$,$H_2 = H_2+C$,
  $H_3 = H_3+D$, $H_4 = H_4+E$.



Sau khi xử lý M_n khối, MD là một chuỗi 160-bit được biểu diễn bởi 5 từ $H_0\ H_1\ H_2\ H_3\ H_4$.

5.1.4. Một số hàm băm khác

- MD4 : Không còn sử dụng
- MD5

Output size	Internal state size	Block size	Length size	Word size	Collision
128	128	512	64	32	Có

- **SHA-256/224**

Output size	Internal state size	Block size	Length size	Word size	Collision
256/224	256	512	64	32	Không

- **SHA-512/384**

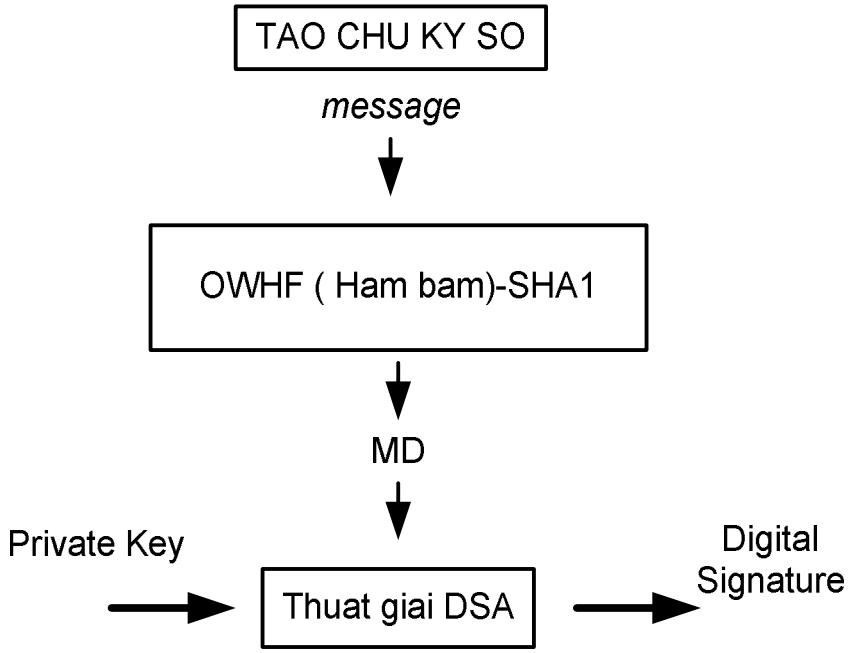
Output size	Internal state size	Block size	Length size	Word size	Collision
512/384	512	1024	128	64	Không

5.2. Chữ ký số (Digital signature)

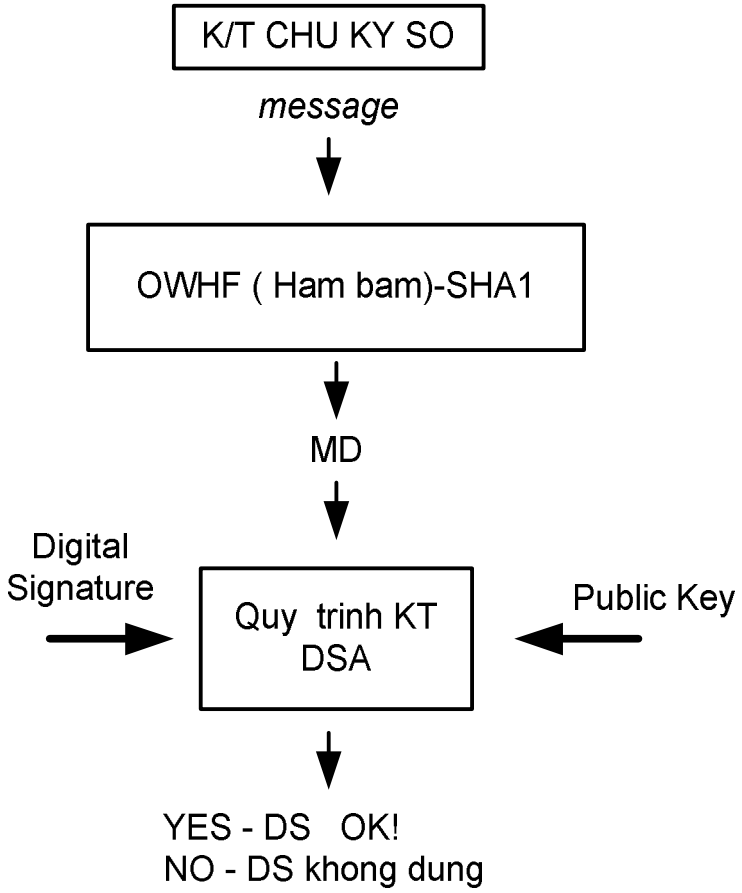
5.2.1.Khái niệm về chữ ký số

1. Đảm bảo tính xác thực
   - Chứng minh tính hợp pháp của người gửi
   - Chứng minh tính toàn vẹn của dữ liệu
2. Chữ ký số là hàm của các tham số
   - Thông báo giao dịch (văn bản gốc)
   - Thông tin bí mật của người gửi (Khóa riêng của sender)
   - Thông tin công khai trên mạng (Khóa công khai)
   - Mã xác thực : Đảm bảo tính toàn vẹn của thông điệp

5.2.2. Tạo và kiểm tra chữ ký số

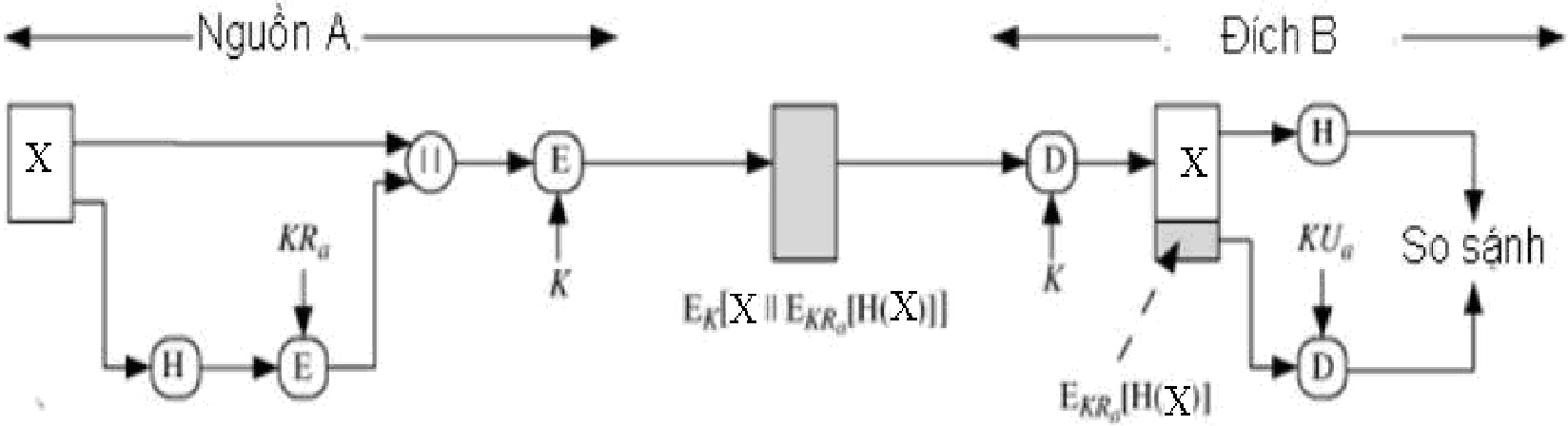


a. Tao chu ky so



b. Kiem tra chu ky so

Digital Signatures



KRa, KUa : khóa bí mật và công khai của bên A

K : khóa phiên đối xứng dùng chung của A và B

X : Thông báo gửi

H : Hàm băm

E : Mã hóa

D : Giải mã

5.2.3. Thuật giải DSA – Chuẩn chữ ký số

a. Họat động

DSA sử dụng các tham số sau:

- p là số nguyên tố với $2^{L-1}<p<2^{L}$ và $512 \le L \le 1024$
- q là một số nguyên tố và là ước số của p-1 với $2^{159}<q<2^{160}$
- $g = h^{(P-1)/q} \bmod p$; trong đó h là một số ngẫu nhiên và $1<h<p-1$; $h^{(p-1)/q} \bmod p > 1$;
- x là một số ngẫu nhiên hoặc là một số biết trước với điều kiện $0<x<q$ (x là khoá cá nhân)
- $y = g^{x} \bmod p$ (y là khóa công khai)

b.Tạo chữ kí số

- Chữ kí của văn bản là 1 cặp số r và s được tính theo công thức sau:
  - $r = g^k (\mathrm{mod} p) \mathrm{mod} q$
  - $s = k^{-1}(H(M) + xr)) \mathrm{mod} q$
- Dữ liệu được gửi đi là Văn bản M, chuỗi số r, và s.

c.Xác thực chữ ký

- Dữ liệu nhận được sẽ là văn bản M, số r và s (hay còn gọi là M',r' và s'). Văn bản trên được xác thực như sau:

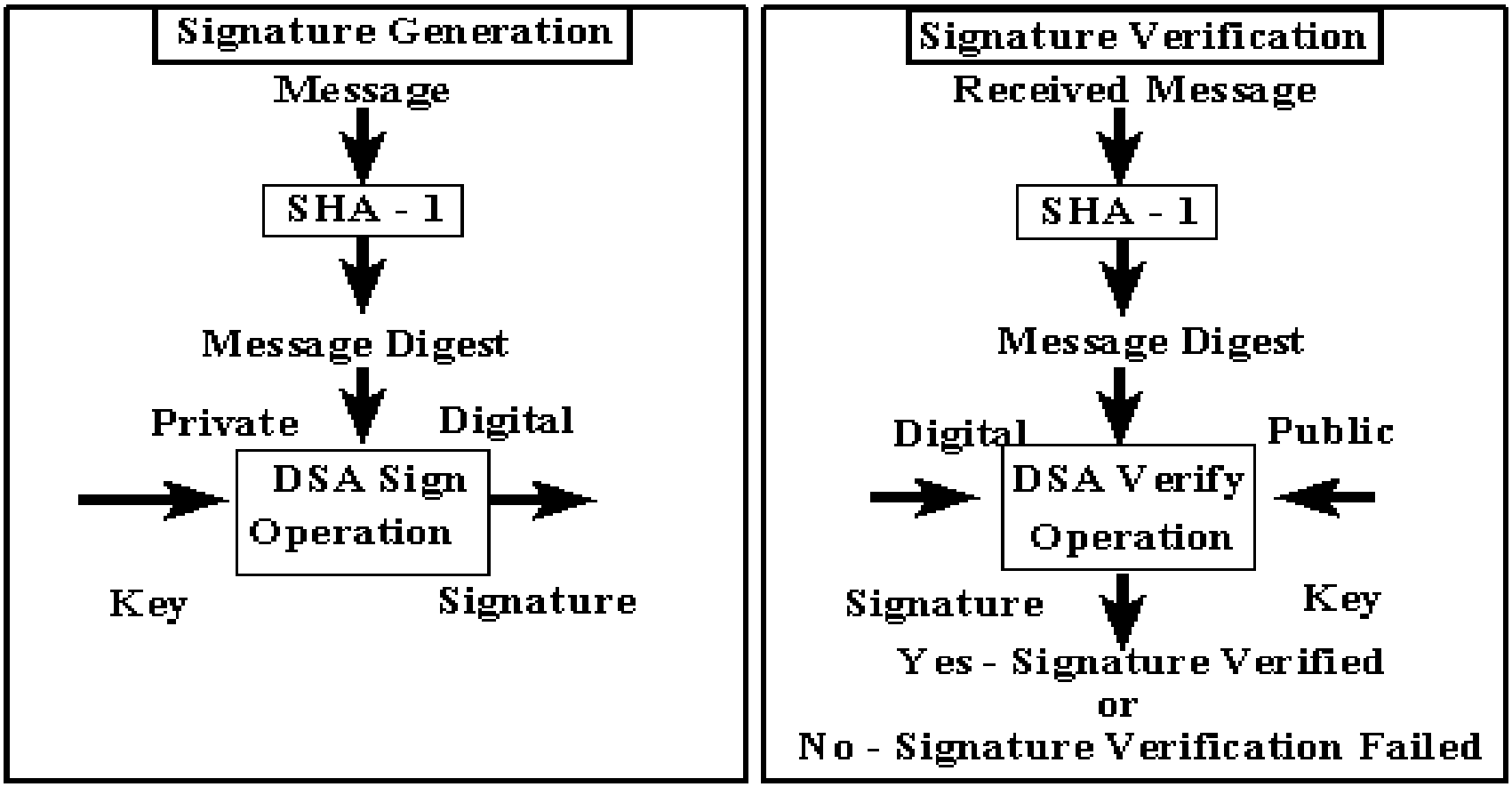
  Đặt w = modq

  u_1=((H(M')w)modq

  u_2=((r')w)modq

  v=((g^{u1})((y^{u2})modp)modq (Hàm kiểm chứng)

- Nếu v=r' chữ kí được xác thực.
- Nếu v != r' văn bản có thể đã được sửa đổi trên đường truyền hoặc khóa cá nhân mã hóa văn bản không khớp với khóa công khai mà người nhận đang giữ (người gửi mạo danh)


Signature Generation
Message
SHA - 1
Message Digest
Private
Digital
DSA Sign
Operation
Key
Signature
Signature Verification
Received Message
SHA - 1
Message Digest
Digital
Public
DSA Verify
Operation
Signature
Key
Yes - Signature Verified
or
No - Signature Verification Failed

5.2.4. Thuật giải RSA trong vai trò chữ ký số

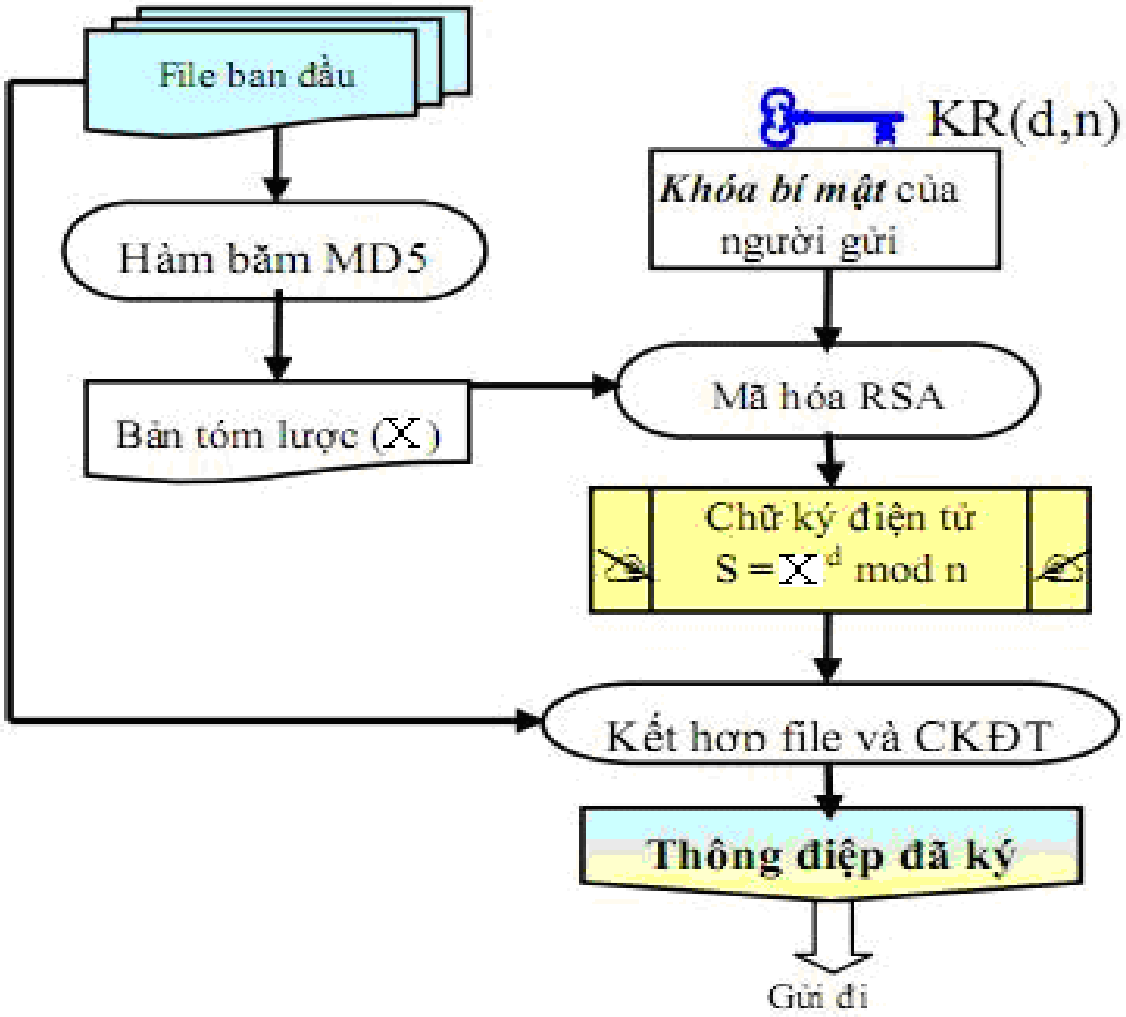
1. Bên gửi

- Tạo bản MD của thông báo $M \rightarrow H(M)$
- Dùng khóa riêng (d_s) của người gửi mã hóa H(M): $E(d_s H(M))$
- Truyền $(M, E(d_s H(M), e_s)$ trong đó k_p là khóa công khai của người gửi

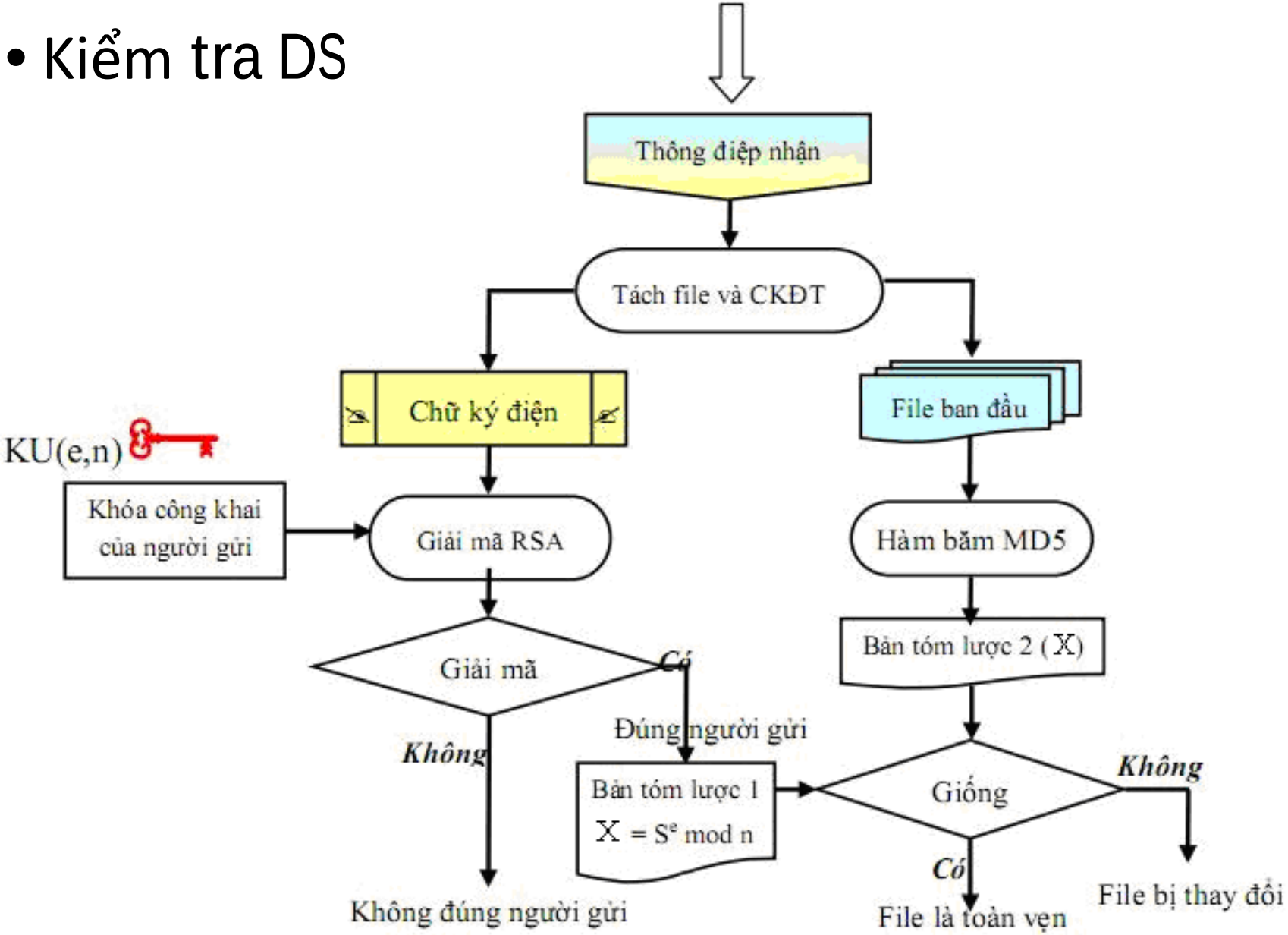
2. Bên nhận

- Tính MD của thông báo nhận được $M^r \rightarrow H(M^r)$
- Dùng khóa công khai của bên gửi (e_s) giải mã thông điệp $D(E(e_s H(M))$ và so sánh kết quả với $H(M^r)$
- Nếu kết quả trùng : xác thực đúng chữ ký của bên gửi . Ngược lại không phải chữ ký bên gửi

- Quá trình ký và gửi các tệp văn bản dựa vào thuật toán băm SHA-1(MD5) và thuật toán RSA



- Kiểm tra DS


Thông điệp nhận
Tách file và CKĐT
Chữ ký điện
File ban đầu
KU(e,n)
Khóa công khai của người gửi
Giải mã RSA
Hàm băm MD5
Giải mã
Có
Bản tóm lược 2 (X)
Đúng người gửi
Không
Bản tóm lược 1
X = S^e mod n
Giống
Không
Có
Không đúng người gửi
File là toàn vẹn
File bị thay đổi

5.2.5. Chuyển giao dữ liệu nhờ RSA

- Chuẩn PKCS#1 :
  - Là một trong 15 chuẩn PKCS do RSA lab đề xuất
  - Điểm quan trọng trong chuẩn PKCS#1 là sử dụng thuật giải RSA trong truyền DATA bao gồm cả quy trình tạo và quản lý Private & Public Keys
- Chuẩn PKCS#1 được sử dụng để mã hóa dữ liệu và lấy chữ ký số của thông điệp

HẾT CHƯƠNG 5