Chương III

3

TOÀN VẸN DỮ LIỆU

PHẦN I: HÀM BĂM MẬT MÃ HỌC (CRYTOGRAPHIC HASH FUNCTIONS)

Nội dung chính

1. Tính toàn vẹn và tính bí mật
2. Tổng quan về hàm băm (Hash Function)
3. Ứng dụng của hàm băm
4. Kiến trúc hàm băm
5. Hai hàm băm MD5 và SHA1

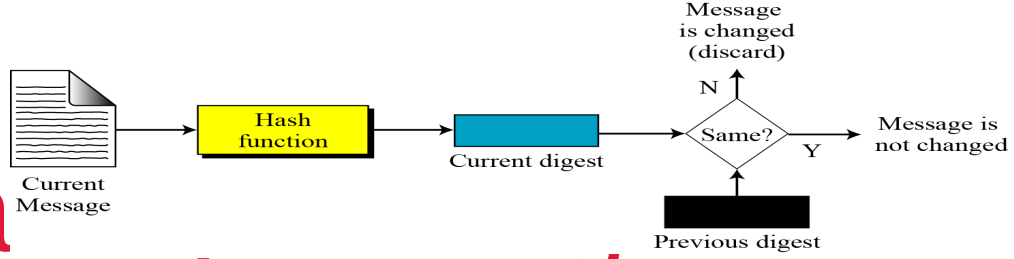
(Cryptography and Network Security: Principles and Practices (3rd Ed.) – Chapter 11)

(*Cryptography & Network Security*. McGraw-Hill, Inc., 2007., Chapter 12)

1. Tính toàn vẹn và tính bí mật

- **Tình toàn vẹn (Integrity):** người tấn công không thể can thiệp để sửa đổi nội dung thông điệp
- Mã hóa chỉ nhằm đảm bảo tính bí mật, không giúp đảm bảo tính toàn vẹn thông tin
- →Người tấn công có thể sửa đổi nội dung thông điệp đã được mã hóa mà không cần biết nội dung thật sự của thông điệp

1. Nhu cầu toà

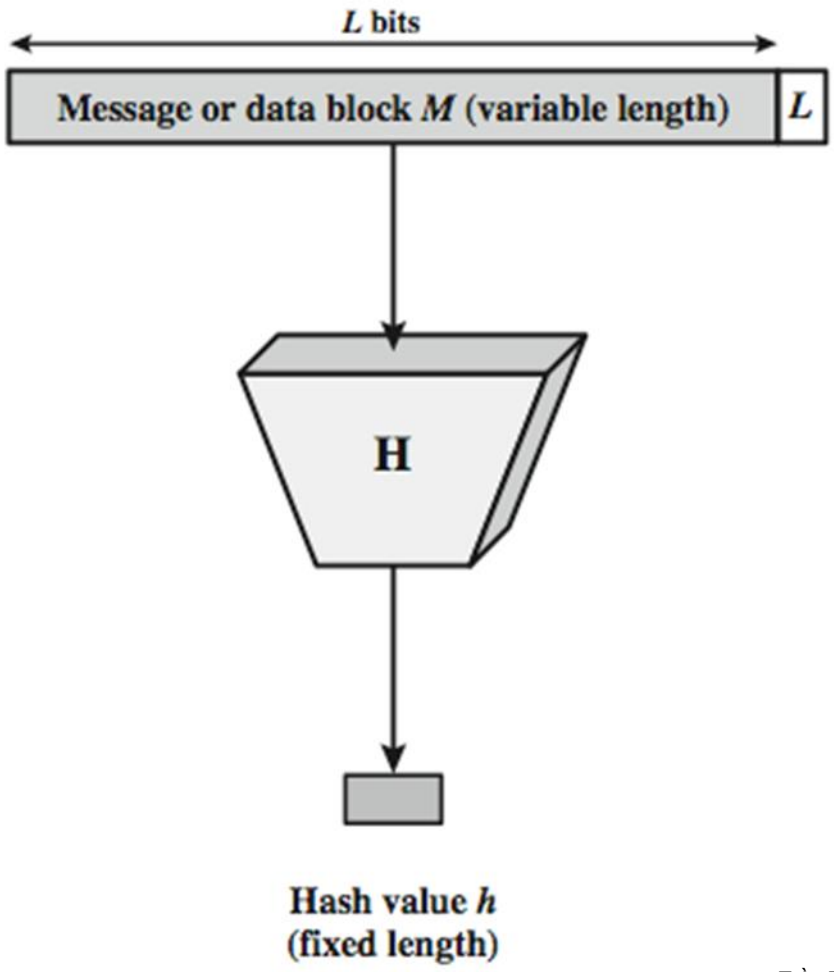


- Các ứng dụng chú trọng mục tiêu Toàn vẹn
  - Tài liệu được sử dụng giống hệt tài liệu lưu trữ
  - Các thông điệp trao đổi trong một hệ thống an toàn không bị thay đổi/sửa chữa
- "Niêm phong" tài liệu/thông điệp
  - "Niêm phong" không bị sửa đổi/phá hủy đồng nghĩa với tài liệu/thông điệp toàn vẹn
  - "Niêm phong": băm (hash), tóm lược (message digest), đặc số kiểm tra (checksum)
  - Tạo ra "niêm phong": hàm băm
- Ví dụ: Trong đấu giá trực tuyến, có thể thay đổi giá đặt của đối thủ mà không cần biết nội dung thật sự của giá đặt

2. Hash Function

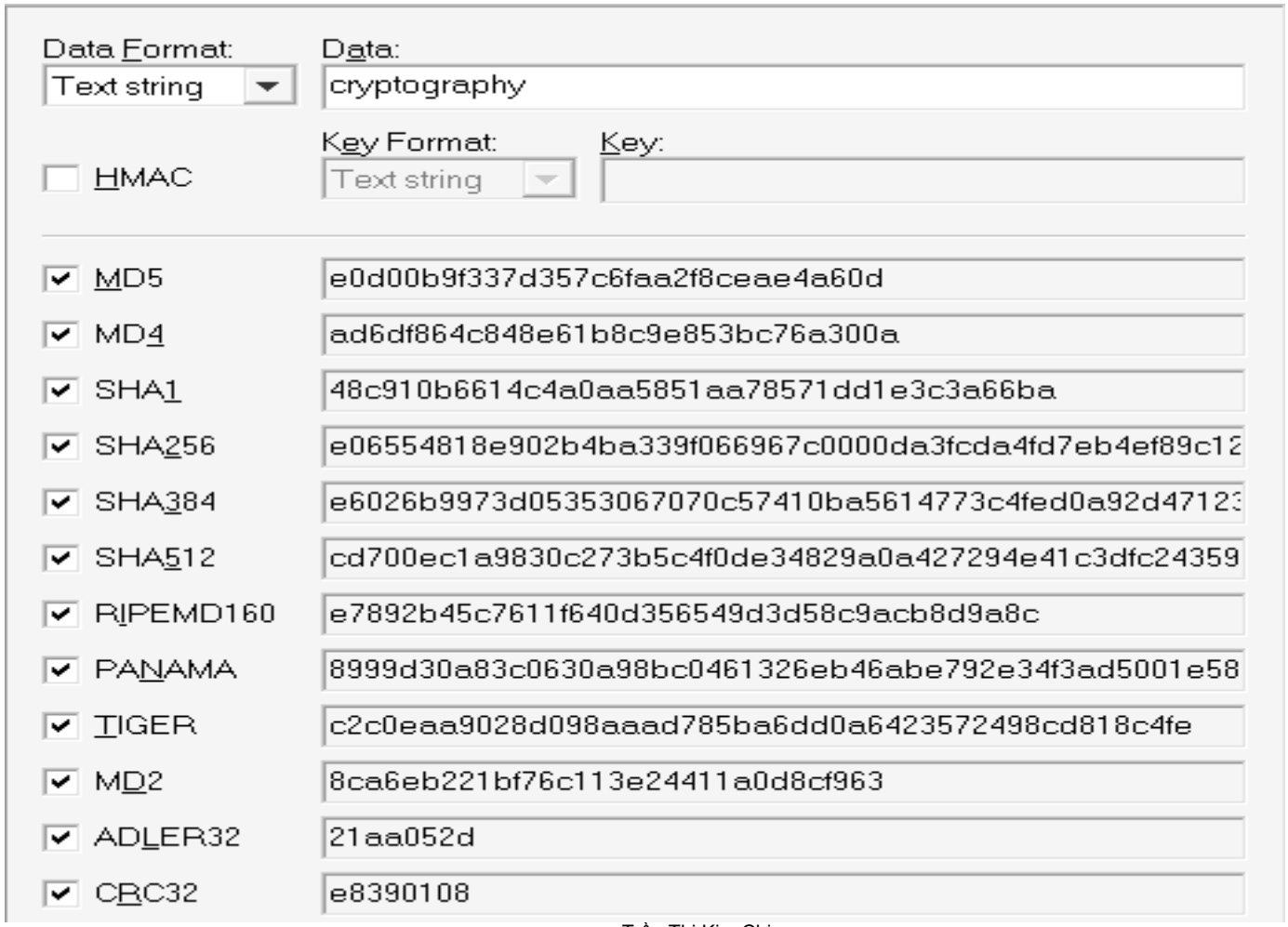
- **Hàm băm** là các thuật toán nhằm phát sinh ra các giá trị băm ứng với mỗi khối dữ liệu/thông điệp.
- Giá trị băm đóng vai trò gần như là một khóa để phân biệt các khối dữ liệu/thông điệp.
- Hàm băm có nhiệm vụ băm thông điệp được đưa vào theo một thuật toán ***h* một chiều nào đó**, rồi đưa ra một giá **trị băm – bản băm – văn bản đại diện – cốt thông điệp**
- Giá trị băm **có kích thước ngắn & cố định & duy nhất & không trùng & không thể suy ngược lại** thông điệp ban đầu .

2. Hash Function

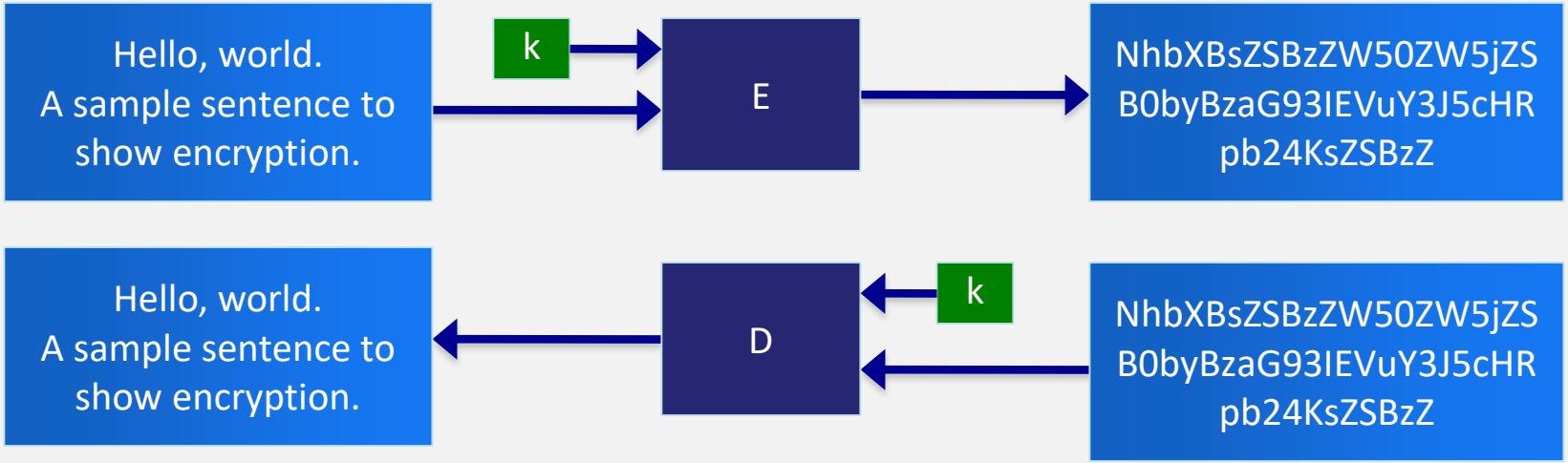


- Input: ***M có kích thước bất kỳ***
- Output – ***giá trị h có kích thước cố định, ngắn.***
- **H(x)** – ***hàm một chiều ("Khó để tính nghịch đảo")***

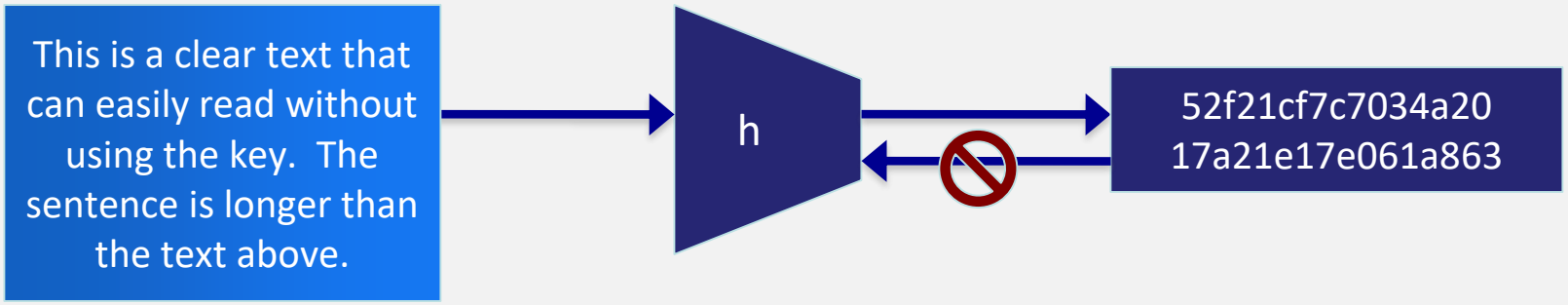
2. Hash Function



2. Hash Function

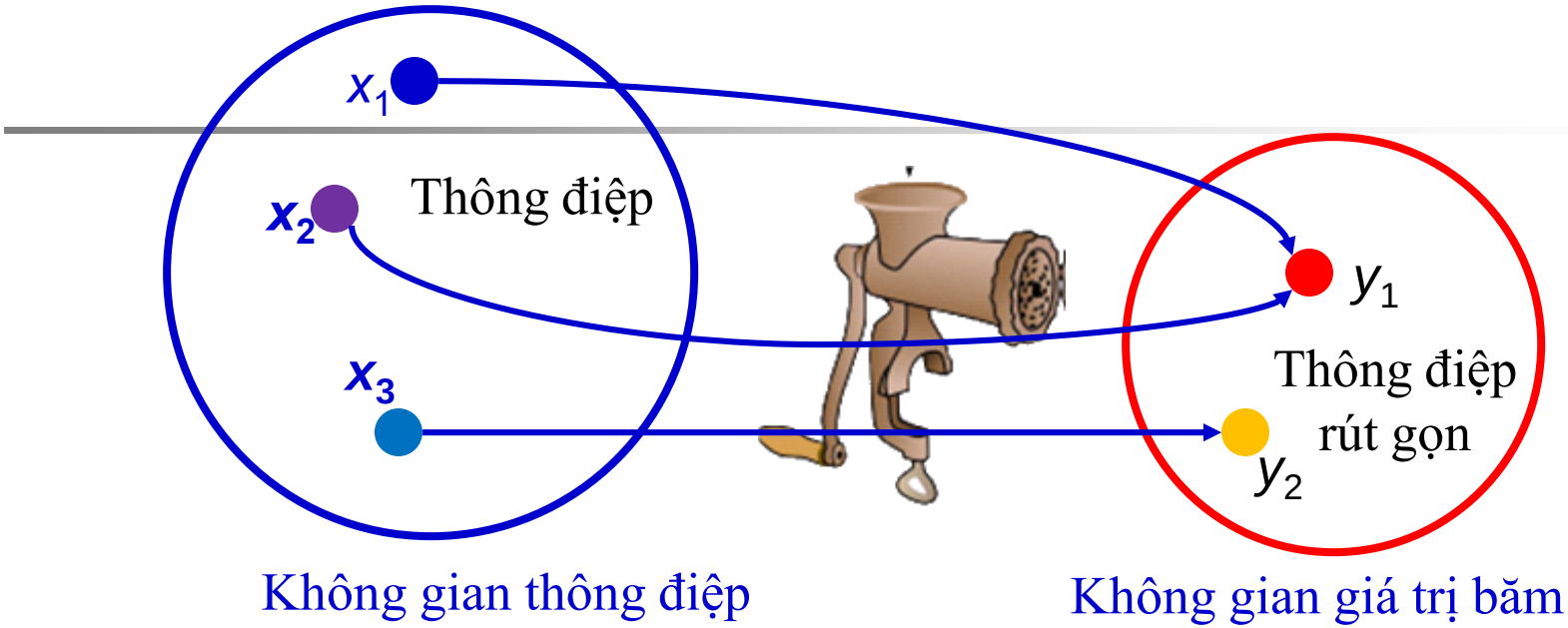


- Encryption is two way, and requires a key to encrypt/decrypt



- Hashing is one-way. There is no 'de-hashing'

2. Hash Function



→Không gian giá trị Băm nhỏ hơn rất nhiều so với Không gian thông điệp về mặt kích thước →chắc chắn sẽ tồn tại đụng độ (trùng), nghĩa là **có hai tin *x* và *x"* mà giá trị Băm của chúng là giống nhau, tức là *h(x) = h(x")***

Tính chất hàm băm

1. **Tính 1 chiều** (Preimage resistant – one-way property):

   Cho trước giá trị băm h việc tìm x sao cho H(x)=h là rất khó

2. **Tính kháng đụng độ yếu** (Second preimage resistant – weak collision resistance – Tính chống trùng yếu):

   Cho thông điệp đầu vào x, việc tìm một thông điệp x' với (x'≠ x) sao cho h(x')=h(x) là rất khó

3. **Tính kháng đụng độ mạnh-tính chống trùng mạnh** (Strong Collision resistance):

   Không thể tính toán để tìm được hai thông điệp đầu vào

   $x_1 \neq x_2$ sao cho chúng có cùng giá trị băm

   (Nghịch lý ngày sinh – Birthday paradox)

Tính chất hàm băm

1. **Tính 1 chiều** (Preimage resistant – one-way property):

   Cho trước giá trị băm h việc tìm x sao cho H(x)=h là rất khó

- Dạng tấn công thứ nhất là người C bắt đầu với một bức điện được ký có giá trị (x, y), trong đó y = sigK(h(x)) (cặp (x, y) có thể là bất kỳ bức điện trước đó mà B đã ký). Sau đó, C tính z = h(x) và cố gắng tìm x‟ # x để h(x‟) = h(x). Nếu C làm được điều này thì cặp (x‟, y) sẽ là một bức điện được ký có giá trị (một bức điện giả mạo có giá trị). Để ngăn cản việc này, hàm Băm h phải thoả mãn tính chất 1

Tính chất hàm băm

2. Tính kháng đụng độ yếu (Second preimage resistant – weak collision resistance – Tính chống trùng yếu):

Cho thông điệp đầu vào x, việc tìm một thông điệp x' với (x'≠ x) sao cho h(x')=h(x) là rất khó

- Một dạng tấn công khác mà người C có thể làm là: đầu tiên anh ta tìm 2 bức điện $x \neq x''$ sao cho $h(x) = h(x'')$. Sau đó C đưa bức điện x cho B và thuyết phục B ký vào cốt bức điện h(x); và vì vậy, anh ta tìm được y. Như vậy, cặp (x'', y) là một cặp chữ ký giả có giá trị. Điều này là nguyên nhân mà việc thiết kế hàm Băm phải thoả mãn tính chất 2.

Tính chất hàm băm

3. Tính kháng đụng độ mạnh-tính chống trùng mạnh (Strong Collision resistance):

Không thể tính toán để tìm được hai thông điệp đầu vào $x_1 \neq x_2$ sao cho chúng có cùng giá trị băm

(Nghịch lý ngày sinh – Birthday paradox)

Dạng tấn công thứ 3 là chọn một giá trị cốt z ngẫu nhiên. Người C sẽ tính một chữ ký với một giá trị ngẫu nhiên z, sau đó anh ta tìm một bức điện x sao cho z = h(x). Nếu anh ta làm được điều này thì cặp (x, y) là cặp chữ ký giả có giá trị. Như vậy một tính chất nữa mà h cần thoả mãn là tính một chiều.

Nghịch lý ngày sinh (birthday paradox) – Chứng minh trang 84 (BaiGiangATTT)

Bài toán 1: *Giả sử trong phòng có M sinh viên. Vậy xác suất để có hai SV có cùng ngày sinh là bao nhiêu phần trăm? (1 năm 365 ngày khác nhau)*

- *Theo nguyên lý chuồng bồ câu Dirichlet: cần có 365+1 = 366 người để tìm thấy 2 người có cùng ngày sinh với xác suất 100%. Vì vậy với 30 người thì xác xuất này rất nhỏ.*
- *Tính theo xác suất thống kê toán học thì*

$$M(M-1) >= 2 \times 365 \times \log_e 2 \quad (*)$$

chỉ cần 23 người là đủ để xác suất hơn 50%. Vì vậy bài toán này gọi là ***nghịch lý ngày sinh***

Nghịch lý ngày sinh (birthday paradox)

Điều này muốn nói lên rằng, *trong nhiều trường hợp xác suất để hai mẩu tin có cùng bản Hash là không nhỏ như chúng ta nghĩ.*

→Tính chống trùng mạnh

Nghịch lý ngày sinh (birthday paradox)

Bài toán 2: Giả sử bạn đang ở trong một lớp học với **M** sinh viên. Hỏi M tối thiểu là bao nhiêu để tồn tại một bạn khác có cùng ngày sinh với bạn với xác suất (XS) lớn hơn 50%?

- XS để 1 người khác ngày sinh với bạn là 364/365.
- →XS để M người đều khác ngày sinh với bạn là $(364/365)^M$.
- →XS để tồn tại ít nhất một người có cùng ngày sinh với bạn là: $1-(364/365)^M$
- Để XS này >50% ➔ **M>=253 người**

→Tính chống trùng yếu

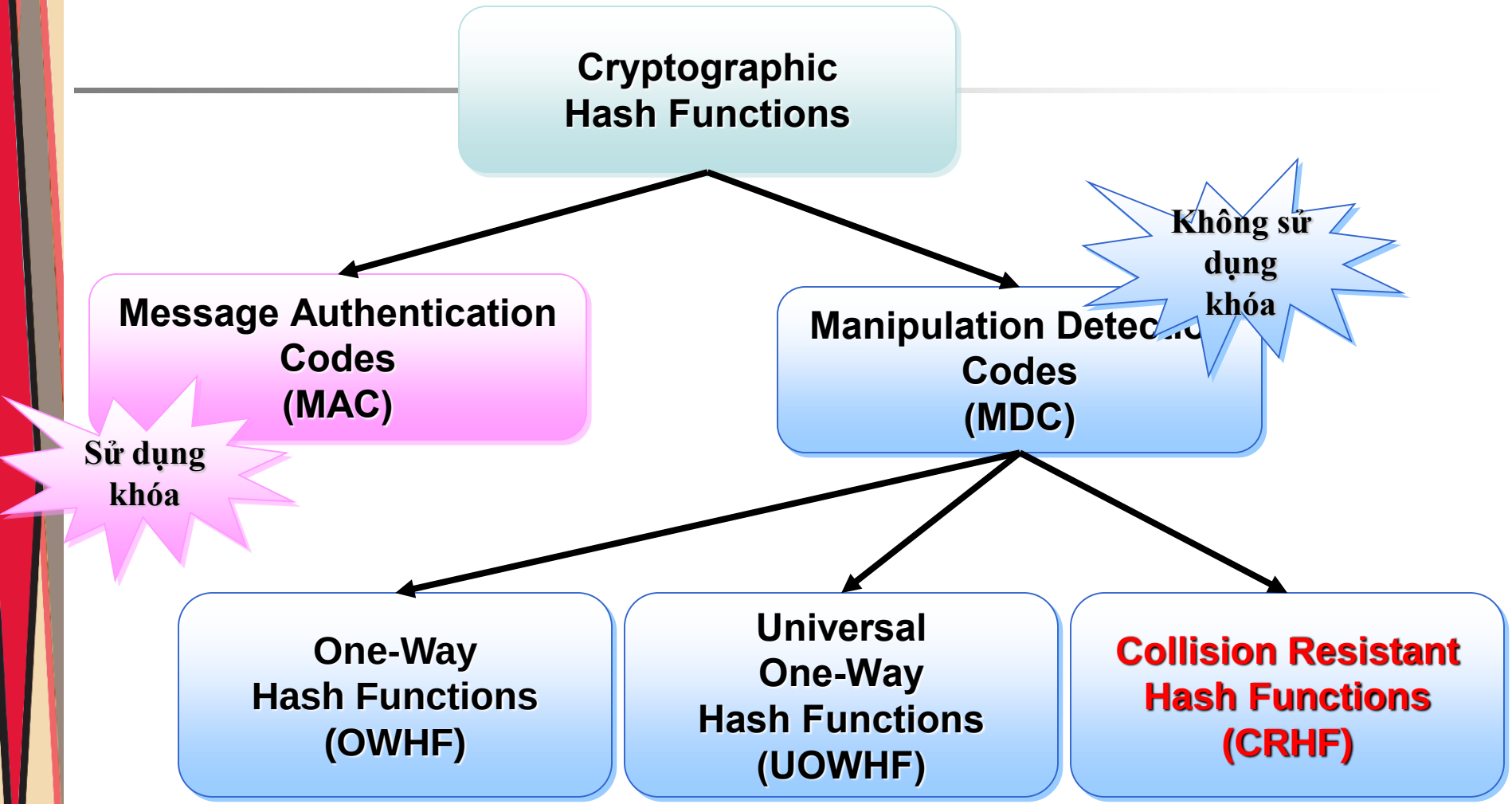
Nghịch lý ngày sinh (birthday paradox)

- Để tìm ra **hai thông điệp có cùng giá trị băm** (vét cạn) thì phải thử bao nhiêu thông điệp khác nhau?

  →Phải thử khoảng $2^{n/2}$ thông điệp khác nhau (xác suất > 50%)

- Ví dụ: Nếu **n=128** thì phải thử 2^{64} thông điệp (khá lớn), nghĩa là hàm hăm đạt được tính chống trùng mạnh (tương đương tấn công vét cạn khóa của DES)

Phân Loại hàm băm mật mã



3. Ứng dụng hàm băm

- **Lưu trữ mật khẩu**
- **Chứng thực thông điệp**
  (Message Authentication)
- **Chữ ký số**
  (Digital Signatures)
- **Các ứng dụng khác**
  (Other Applications)

3. 1 Lưu trữ mật khẩu

- Mật khẩu bao gồm chuỗi các chữ cái (hoa, thường), chữ số và các ký tự đặc biệt (@, # …).
- Do tính chất của hàm toán học một chiều, mật khẩu của tài khoản được bảo vệ ngay cả trong trường hợp file lưu trữ mật khẩu hệ thống bị sao chép.

Username	Password
admin	@123!Fitluh
trandung	@123!Tran
Lưu trữ không mã hóa mật khẩu	

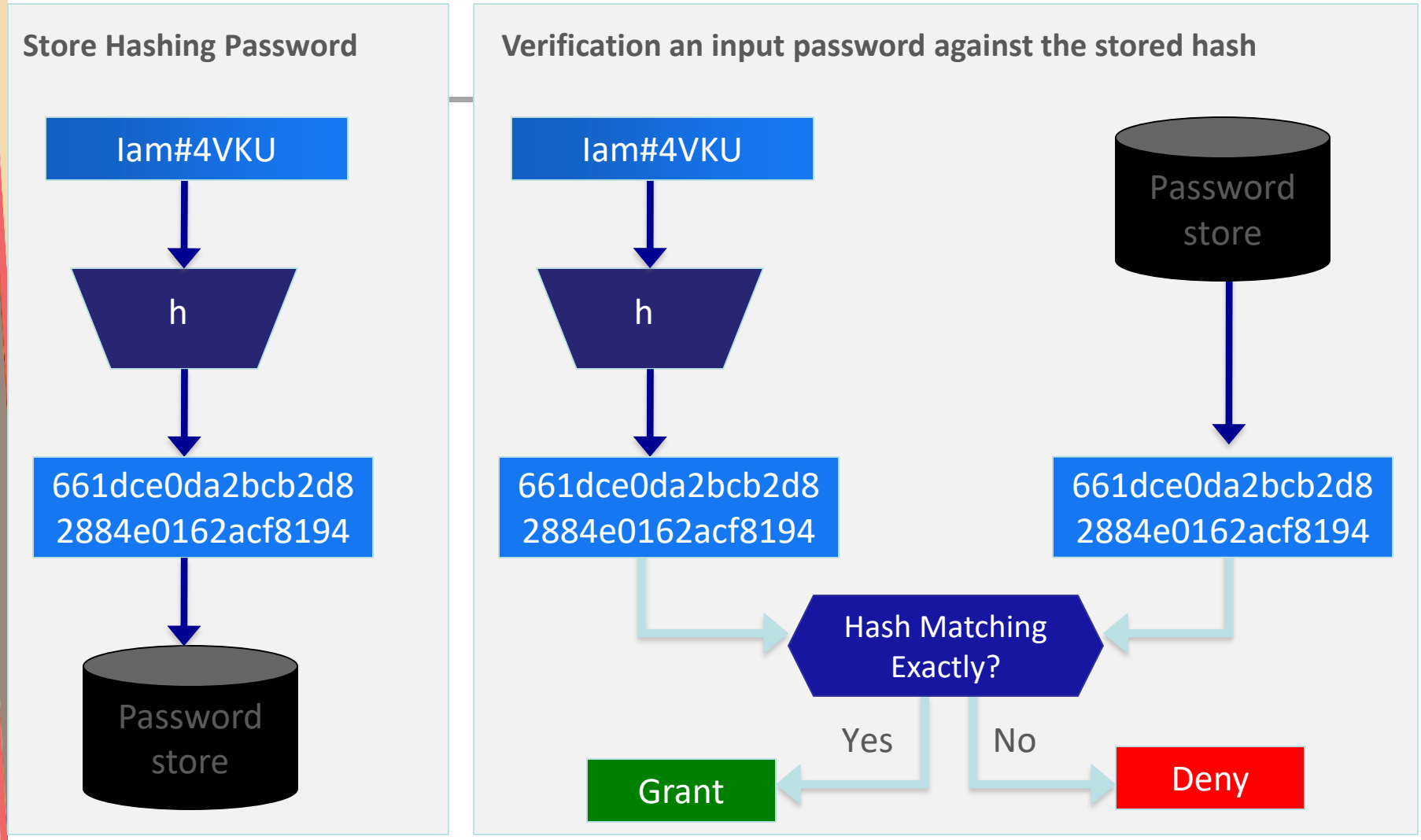
Username	Password
admin	69c919ee4881666e4c90d51d4a2ed505
trandung	168838fe639bd5d8b660d5b008978759
Lưu trữ mã hóa mật khẩu với MD5	

3. 1 Lưu trữ mật khẩu

Dùng lưu trữ mật khẩu (băm password):

- Hàm băm được dùng để tạo **one-way password file**, trong cơ chế này giá trị băm của password được lưu, điều này tốt hơn là lưu chính bản rõ password. → password không bị truy xuất bởi kẻ tấn công nơi chứa password.
- Khi user nhập vào một password, thì giá trị băm của password được so với giá trị băm được lưu để kiểm tra.

Password Verification



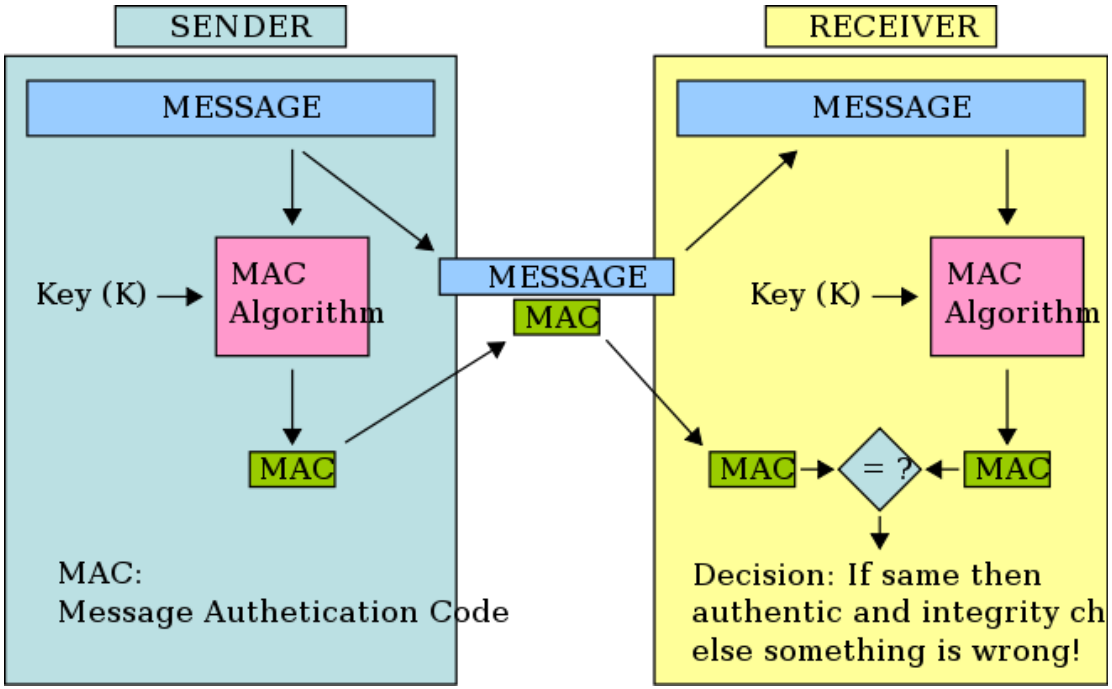
3.2 Message Authentication

- Xác thực thông điệp liên quan đến các khía cạnh sau khi truyền tin trên mạng
  - **Bảo vệ tính toàn vẹn của thông điệp:** bảo vệ thông điệp không bị thay đổi hoặc có các biện pháp phát hiện nếu thông điệp bị thay đổi trên đường truyền.
  - **Kiểm chứng danh tính, nguồn gốc**: xem xét thông điệp có đúng do người xưng tên gửi không hay một kẻ mạo danh nào khác gửi.
  - **Không chối từ bản gốc:** trong trường hợp cần thiết, bản thân thông điệp chứa các thông tin chứng tỏ chỉ có người xưng danh gửi, không một ai khác có thể làm điều đó => Người gửi không thể từ chối hành động gửi, thời gian gửi và nội dung của thông điệp.
- Hàm băm dạng này, giá trị băm (h) được gọi là **tóm tắt thông điệp (*message digest*)**

3.2 Message Authentication

- **Các yêu cầu bảo mật khi truyền mẩu tin:**
  - **Để lộ bí mật**: giữ bí mật nội dung mẩu tin, chỉ cho người có quyền biết.
  - **Thám mã đường truyền**: không cho theo dõi hoặc làm trì hoãn việc truyền tin.
  - **Giả mạo**: lấy danh nghĩa người khác để gửi tin.
  - **Sửa đổi nội dung**: thay đổi, cắt xén, thêm bớt thông tin.
  - **Thay đổi trình tự** các gói tin nhỏ của mẩu tin truyền.
  - **Sửa đổi thời gian**: làm trì hoãn mẩu tin.
  - **Từ chối gốc**: không cho phép người gửi từ chối trách nhiệm của tác giả mẩu tin.
  - **Từ chối đích**: không cho phép người nhận phủ định sự tồn tại và đến đích của mẩu tin đã gửi.

3.2 Message Authentication



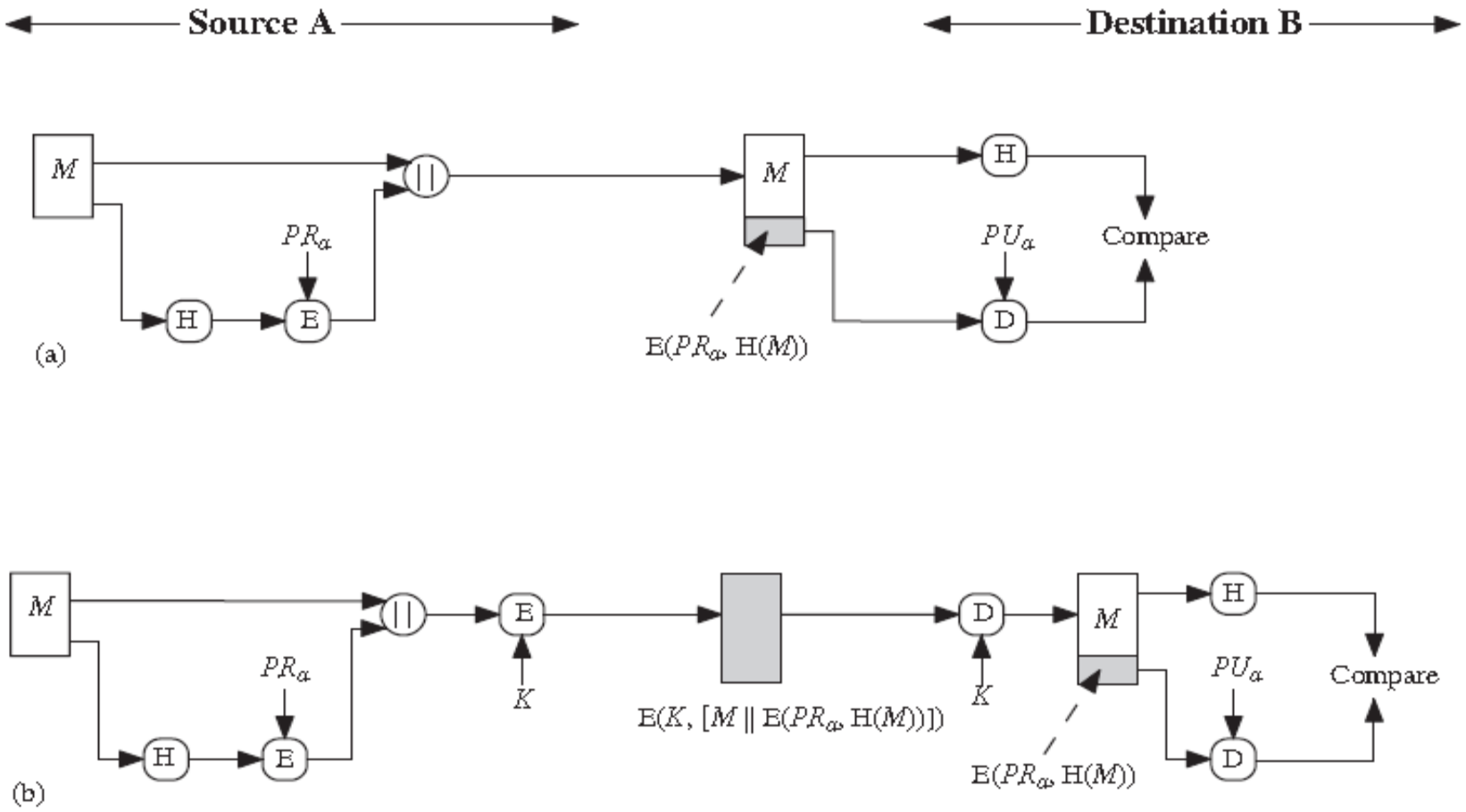
- bảo vệ toàn vẹn thông điệp và cả tính xác thực thông điệp bằng cách cho phép kiểm định (cũng là người sở hữu khóa bí mật) phát hiện bất kỳ thay đổi trong nội dung thông điệp

3.3 Digital Signatures

- Giá trị băm của thông điệp được mã hóa bằng **private key** của user, bất kỳ ai biết **public key** của user thì có thể thẩm tra thông điệp mà được gắn kết với chữ ký số.

- Kẻ tấn công muốn hiệu chỉnh thông điệp thì sẽ cần phải biết private key của user.

3.3 Digital Signatures



3.4 Other Applications

Dùng nhận diện xâm hại (*intrusion detection*) và nhận diện virus (*virus detection*).

- Tính, lưu và bảo mật giá trị băm H(F) của các tập tin trong hệ thống (có thể lưu trên CD-R)
- Kẻ xâm hại cần phải hiệu chỉnh F mà không thay đổi H(F)

3.4 Other Applications

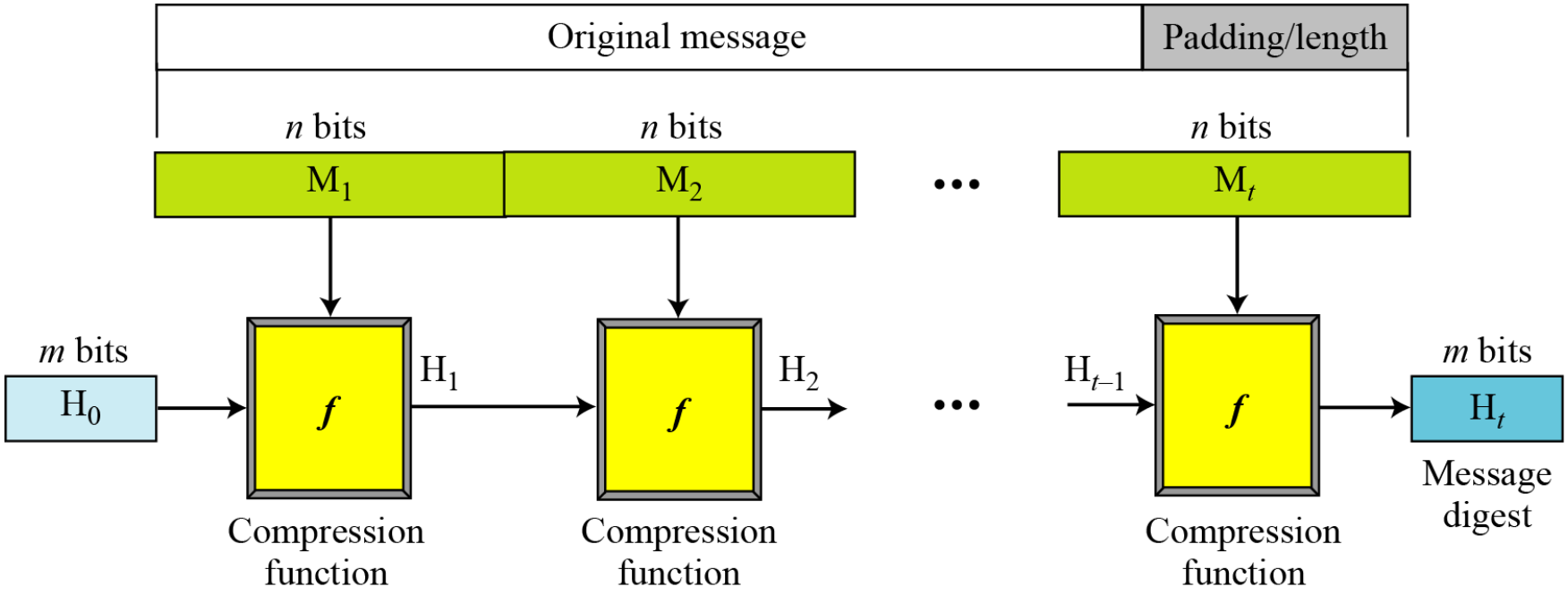
- **Dùng:**

  Xây dựng hàm ngẫu nhiên giả (*pseudorandom function - PRF*)

  hoặc

  Phát sinh số ngẫu nhiên giả (*pseudorandom number generator - PRNG*)

4. Kiến trúc hàm băm an toàn



- Tác giả: Ralph Merkle, Ivan Damgård
- Hầu hết các hàm băm đều sử dụng cấu trúc này
- Ví dụ: SHA-1, MD5

5. Hàm băm MD5, SHA1

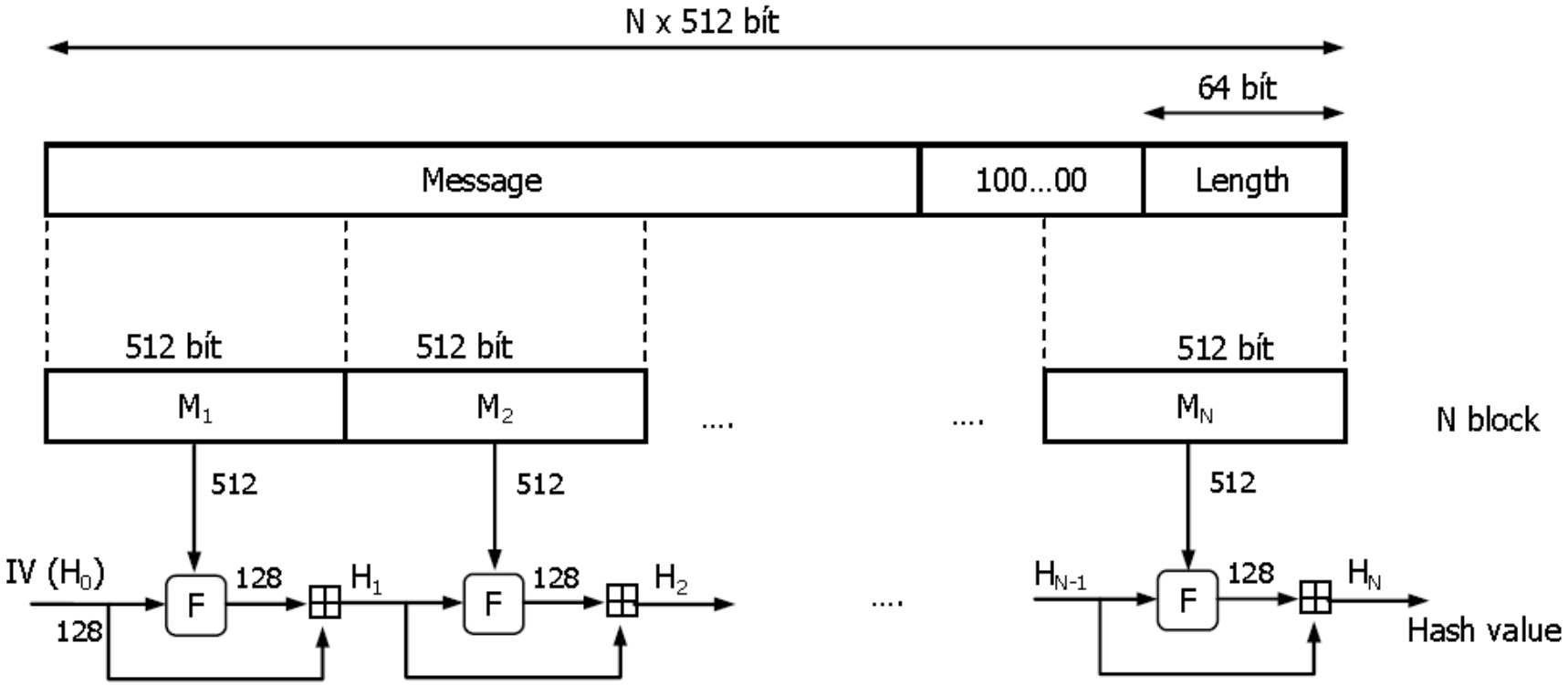
- MD5 (Message Digest)
  - Phát minh bởi Ron Rivest (RSA)
  - Phát triển từ MD4, trước đó MD2 (không an toàn). Hiện tại có MD6
  - Kích thước giá trị băm là 128-bit
- Được sử dụng rộng rãi
  - Truyền tập tin
  - Lưu trữ mật khẩu

5. Hàm băm MD5, SHA1

- SHA (Secure Hash Algorithm)
  - Được phát triển bởi NIST 1993 (SHA-0)
  - 1995: SHA-1 - Chính phủ Mỹ chọn làm chuẩn quốc gia. Kích thước giá trị băm 160-bit
  - Được sử dụng rộng rãi trong các giao thức SSL, PGP, SSH, S/MIME, IPSec

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

5.1 Hàm băm MD5 (128-bit, $\leq 2^{64}$-bit)

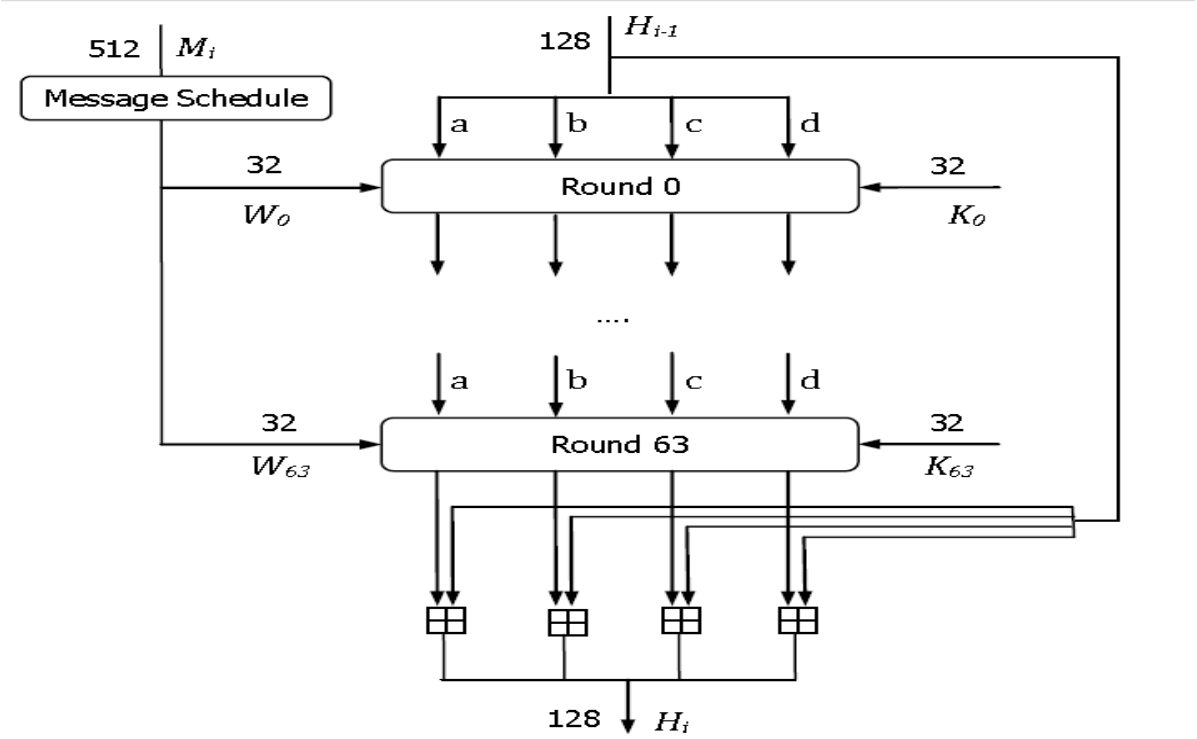


H_0 – 128-bit, chia thành 4 từ 32-bit, ký hiệu a,b,c,d – hằng số (thập lục phân)

a=01234567; b=89abcdef; c=fedbca98; d=76543210

5.1 Hàm băm MD5 (128-bit, $\leq 2^{64}$-bit)

- **Cấu trúc hàm F tại mỗi bước lũy tiến**



K_j: phần nguyên của 2^{32}abs(sin(i)) với *i* biểu diễn radian

M_i được biến đổi qua hàm **message schedule** cho ra W_0, W_1, .., W_{63}

5.1 Hàm băm MD5 (128-bit, $\leq 2^{64}$-bit)

Cấu trúc của một vòng trong F

Ở đây: b→c, c→d, d→a, b được tính qua hàm

$$t=a+f(b,c,d)+W_i+K_i$$

$$b=b+ROTL(t,s)$$

Hàm f(x,y,z):

f(x,y,z)=(x∧y)∨(¬x∧z) nếu vòng 0 – 15

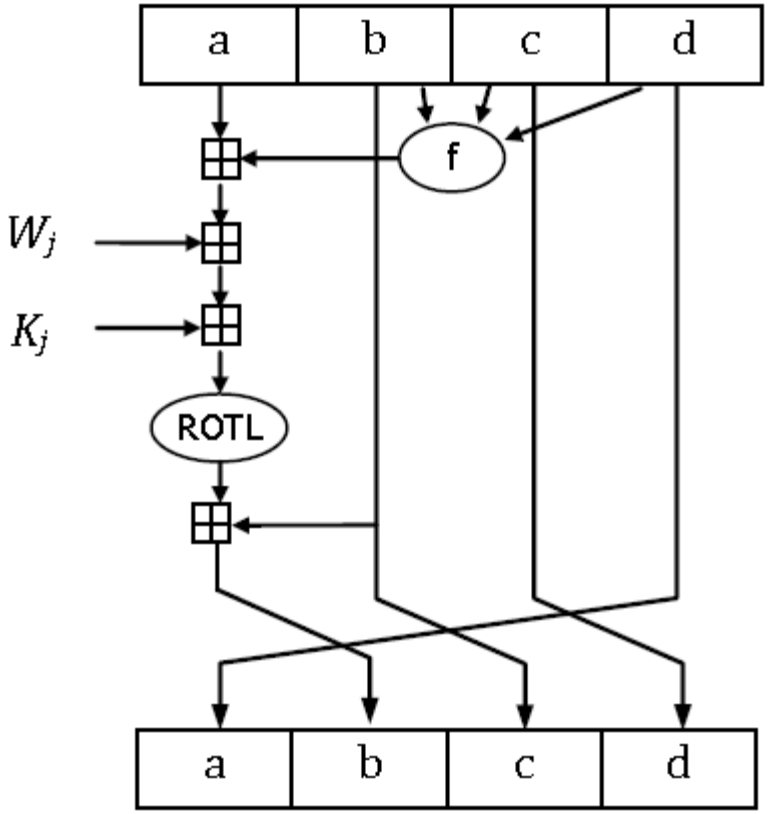
f(x,y,z)=(z∧x)∨(¬z∧y) nếu vòng 16 – 31

f(x,y,z)=x⊕y ⊕x nếu vòng 32 – 48

f(x,y,z)=y⊕ (x∨¬z) nếu vòng 49 – 63

Hàm ROTL(t,s): t được dịch vòng trái s-bit, với s là các hằng số cho vòng thứ i

Phép + (hay ⧺) là phép cộng modulo 2^{32}

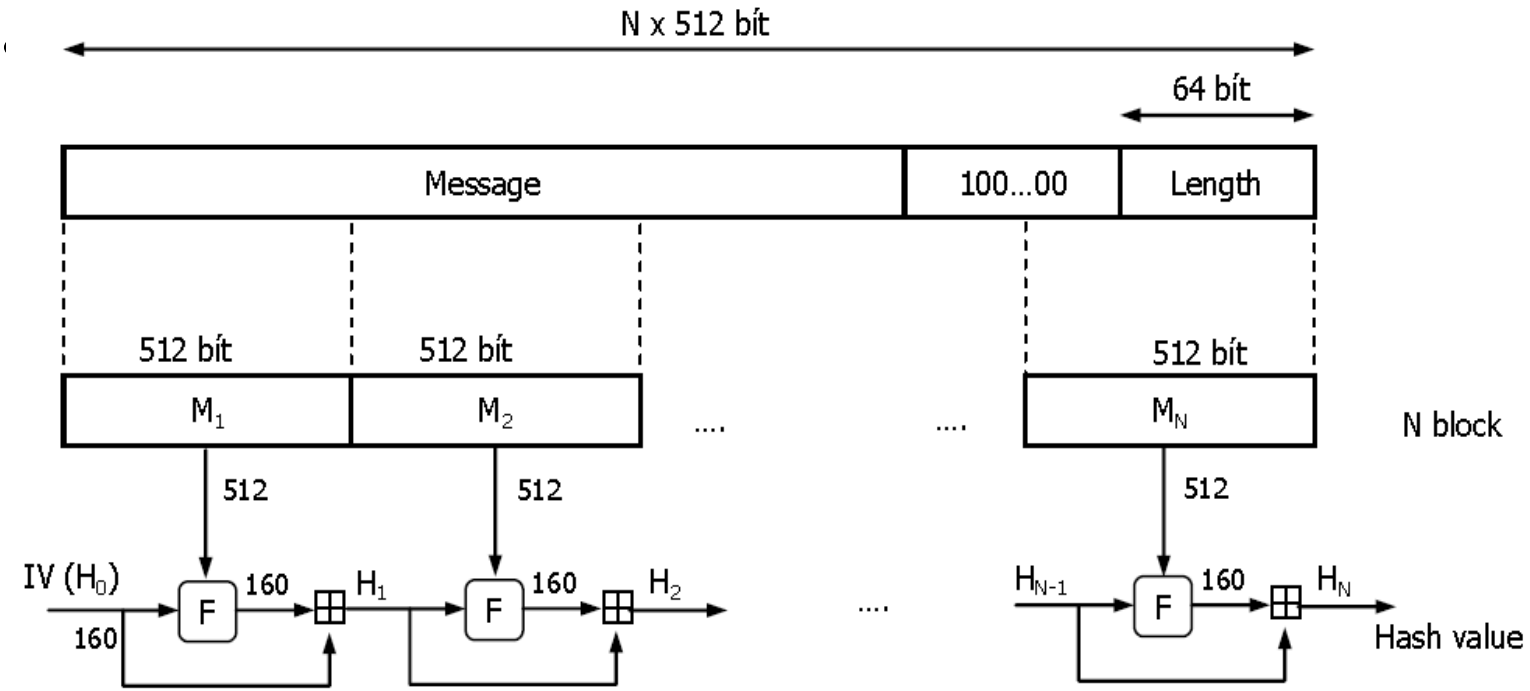


5.1 Hàm băm MD5 (128-bit, $\leq 2^{64}$-bit)

- **Hàm ROTL(t,s):** t được dịch vòng trái s-bit, với s là các hằng số cho vòng thứ i

i	*s*
0, 4, 8, 12	7
1, 5, 9, 13	12
2, 6,10, 14	17
3, 7, 11, 15	22
16, 20, 24, 28	5
17, 21, 25, 29	9
18, 22, 26, 30	14
19, 23, 27, 31	20
32, 36, 40, 44	4
33, 37, 41, 45	11
34, 38, 42, 46	16
35, 39, 43, 47	23
48, 52, 56, 60	6
49, 53, 57, 61	10
50, 54, 58, 62	15
51, 55, 59, 63	21

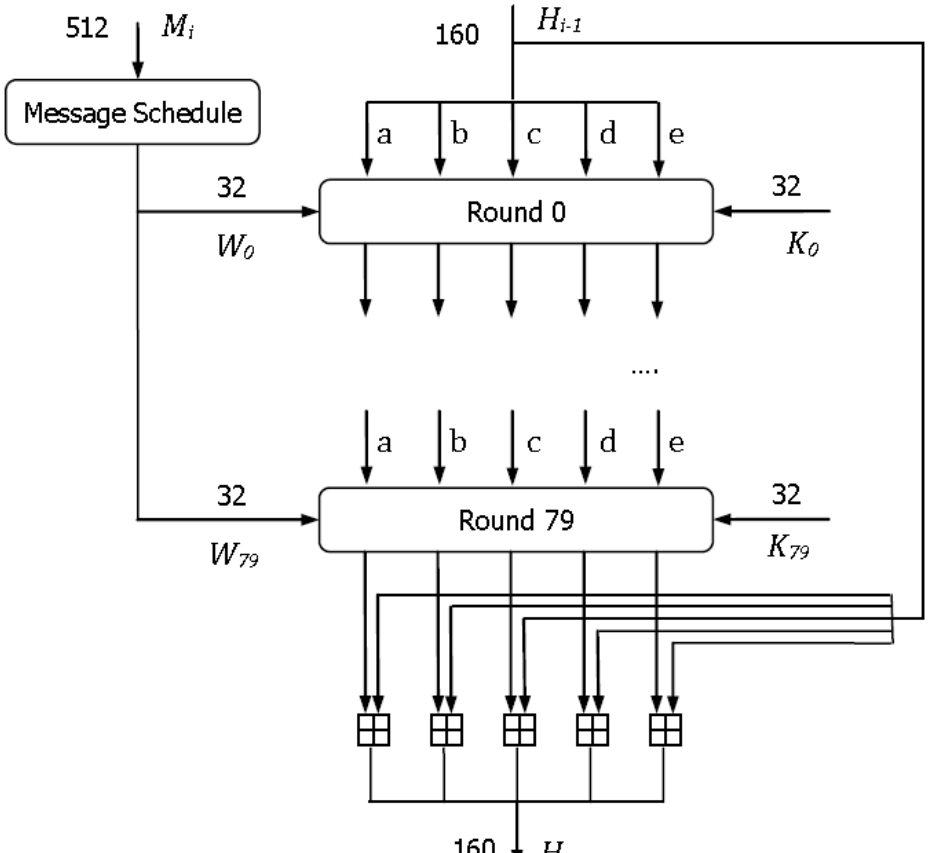
5.2 Hàm băm SHA-1 (160-bit, 2^{64}-bit)



H_0 – 160-bit, chia thành 5 từ 32-bit, ký hiệu a,b,c,d,e – hằng số
a=67452301; b=efcdab89; c=98badcfe; d=10325476; e=c3d2e1f0

5.2 Hàm băm SHA-1 (160-bit, 2^{64}-bit)

- Cấu trúc hàm F cũng tương tự MD5, nhưng được thực hiện **80 vòng**



K_i là hằng số

K_i=5A827999 với $0 \le i \le 19$

K_i=6ED9EBA1 với $20 \le i \le 39$

K_i=8F1BBCDC với $40 \le i \le 59$

K_i=CA62C1D6 với $60 \le i \le 79$

Giá trị M_i – biến đổi qua message schedule cho ra 80 giá trị như sau:

- M_i chia thành 16 block 32-bit ứng với $W_0, W_1, \ldots, W_{15}$.
- Các W_t ($16 \le t \le 79$) được tính:

W_t=ROTL($W_{t-3}+W_{t-8}+W_{t-14}+W_{t-16}$,1)

với phép cộng modulo 2^{32}

5.2 Hàm băm SHA-1 (160-bit, 2^{64}-bit)

Cấu trúc của một vòng trong F

Ở đây: a→b, c→d, d→e. Giá trị a và c được tính:

$$a=ROTL(a,5)+f(b,c,d)+e+W_i+K_i$$

$$c=ROTL(b,30)$$

Hàm f(x,y,z):

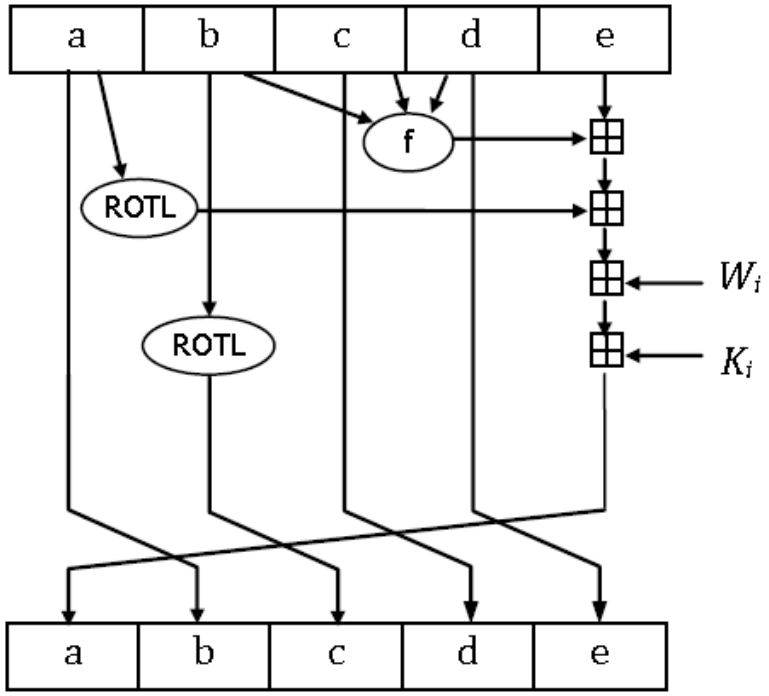
f(x,y,z)=Cf(x,y,z)=(x∧y) ⊕(–|x∧z) nếu vòng 0 – 19

f(x,y,z)=Parity(x,y,z)= x⊕y ⊕z nếu vòng 20 – 39

f(x,y,z)=Maj(x,y,z)=(x∧y) ⊕(y∧z)⊕(z∧x) nếu vòng 40 – 59

f(x,y,z)=Party(x,y,z)= x⊕y ⊕z nếu vòng 60 – 79

- Hàm Maj: giải sử x_i, y_i, z_i là bit thứ i của x,y,z th bit thứ i của hàm Maj là giá trị nào chiếm đa số, 0 hoặc 1
- Hàm Ch: bit thứ i của hàm Ch là phép chọn: if x_i then y_i else x_i



So sánh giữa MD5 và SHA-1

- Khả năng chống lại tấn công brute-force:
  - Để tạo ra thông điệp có giá trị băm cho trước, cần 2^{128} thao tác với MD5 và 2^{160} với SHA-1
  - Để tìm 2 thông điệp có cùng giá trị băm, cần 2^{64} thao tác với MD5 và 2^{80} với SHA-1
- Khả năng chống lại thám mã: cả 2 đều có cấu trúc tốt
- Tốc độ:
  - Cả hai dựa trên phép toán 32 bit, thực hiện tốt trên các kiến trúc 32 bit
  - SHA-1 thực hiện nhiều hơn 16 bước và thao tác trên thanh ghi 160 bit nên tốt độ thực hiện chậm hơn
- Tính đơn giản: cả hai đều được mô tả đơn giản và dễ dàng cài đặt trên phần cứng và phần mềm

Ứng dụng Hàm băm

- **Xác thực thông điệp** (Message Authentication)
- **Nâng hiệu quả của chữ ký số** (Digital Signatures)
- **Bảo mật mật khẩu** (Password hashing)

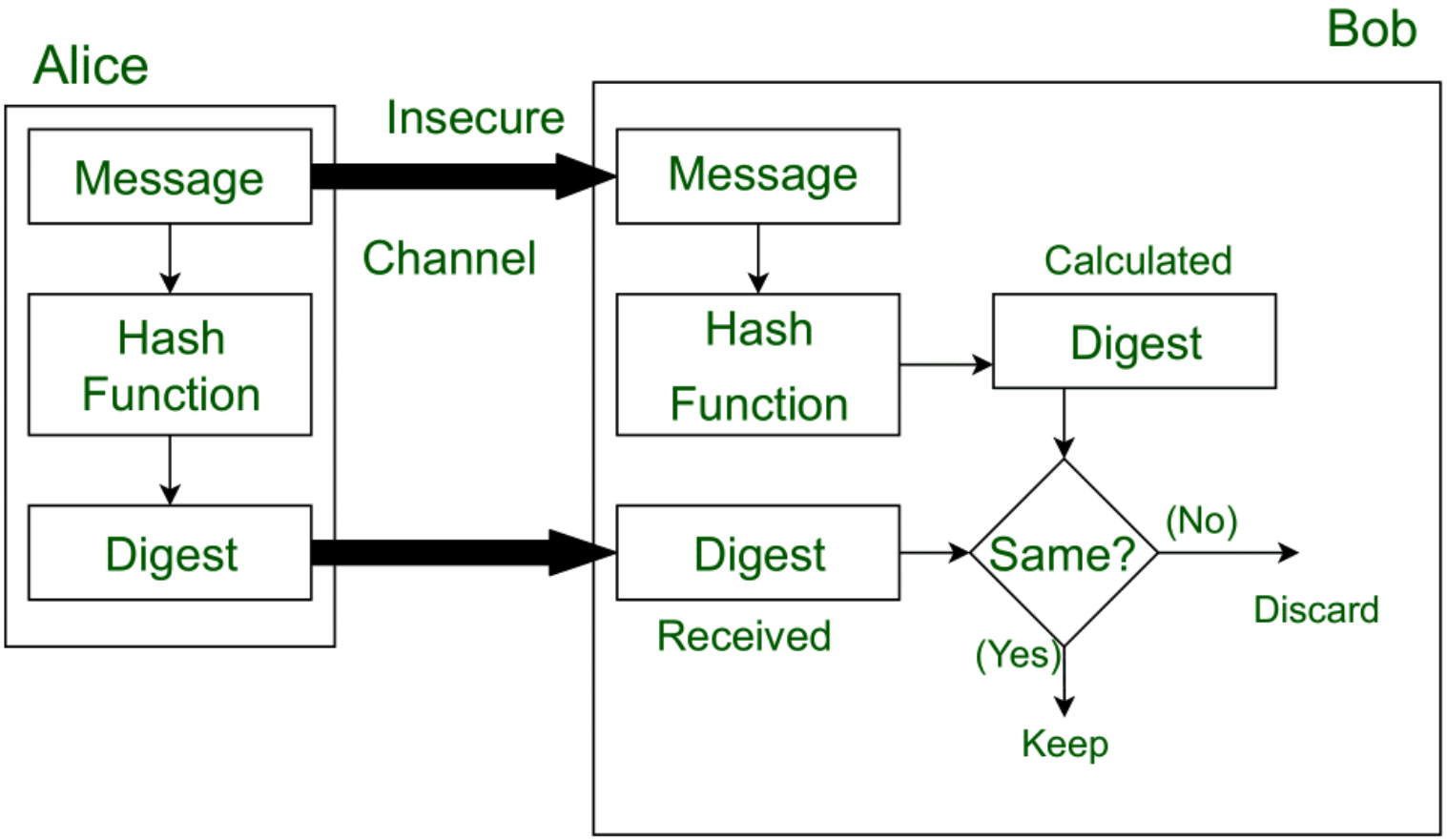
Một số ứng dụng MD5 và SHA-1

Xác thực thông điệp (Message Authentication Code – MAC)

- Là một **cơ chế/dịch vụ** được dùng để kiểm tra:
  - **Tính toàn vẹn** của một thông điệp - Đảm bảo thông điệp nhận được là chính xác như khi được gửi (không bị chỉnh sửa, chèn, hoặc thay thế)
  - **Nguồn gốc** của thông điệp – Xác định thông điệp đến từ ai (sender là hợp pháp).
- Hàm băm dạng này, giá trị băm (h) được gọi là **tóm tắt thông điệp hoặc cốt thông điệp (*message digest*)**

Một số ứng dụng MD5 và SHA-1

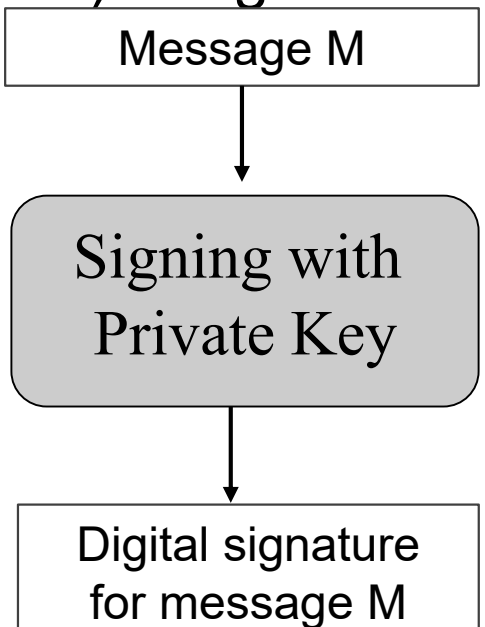
Xác thực thông điệp (Message Authentication Code – MAC)



Một số ứng dụng MD5 và SHA-1

Chữ ký điện tử (Digital signature) đảm bảo

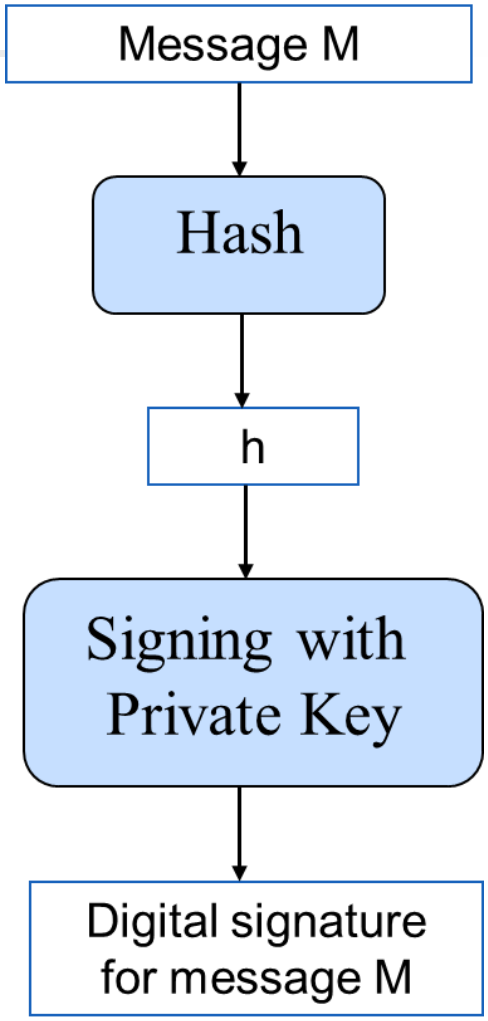
- Chữ ký số của một người dùng trên một thông điệp chính là mã hóa thông điệp bằng khóa riêng phần (Private Key) của người dùng đó.
- Khi đó chữ ký có kích thước (chiều dài) bằng chiều dài thông điệp.



Một số ứng dụng MD5 và SHA-1

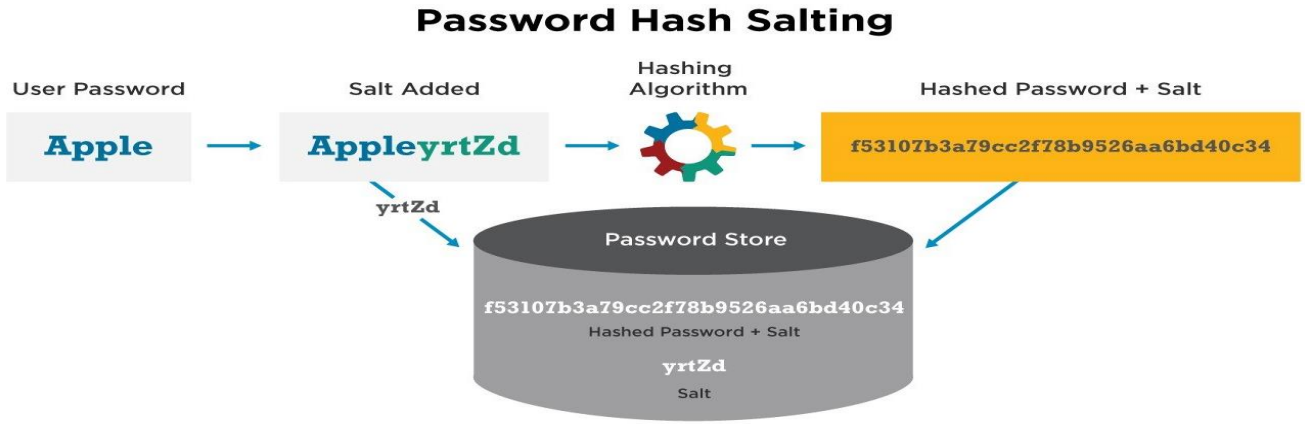
Chữ ký điện tử (Digital signature) đảm bảo

- Chữ ký số được tạo ta trên giá trị băm của thông điệp thay vì tạo trên thông điệp
- → Chữ ký trên giá trị băm sẽ có kích thước ngắn hơn rất nhiều trên chữ ký trên thông điệp
- → tăng tốc cho các ứng dụng chữ ký số



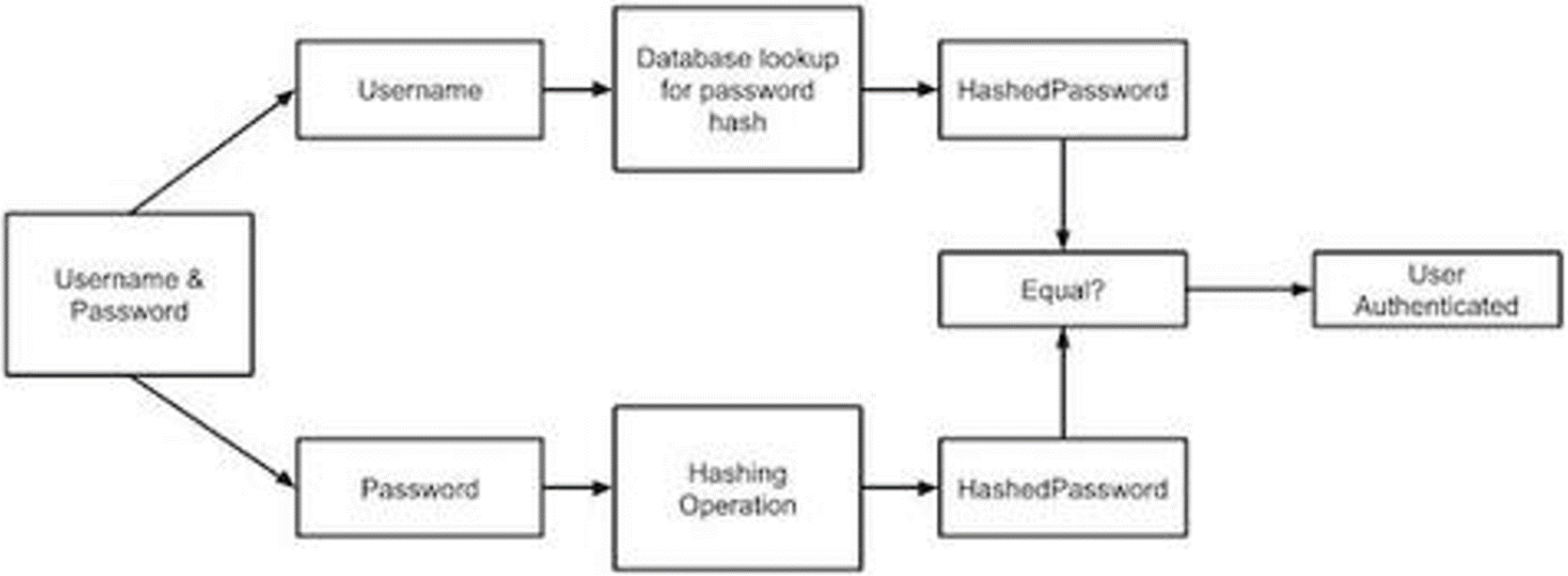
Bảo mật mật khẩu người dùng

- Nếu quản lý và lưu trữ các mật khẩu (password) của người dùng bằng cách lưu trữ bản rõ của password thì thông tin về user trong đó có password sẽ bị lộ khi nơi lưu trữ bị xâm phạm.
- → Thay vì lưu password dưới dạng bản rõ thì password sẽ được đưa vào một hàm băm (hash function) để sinh ra giá trị băm đại diện cho password.



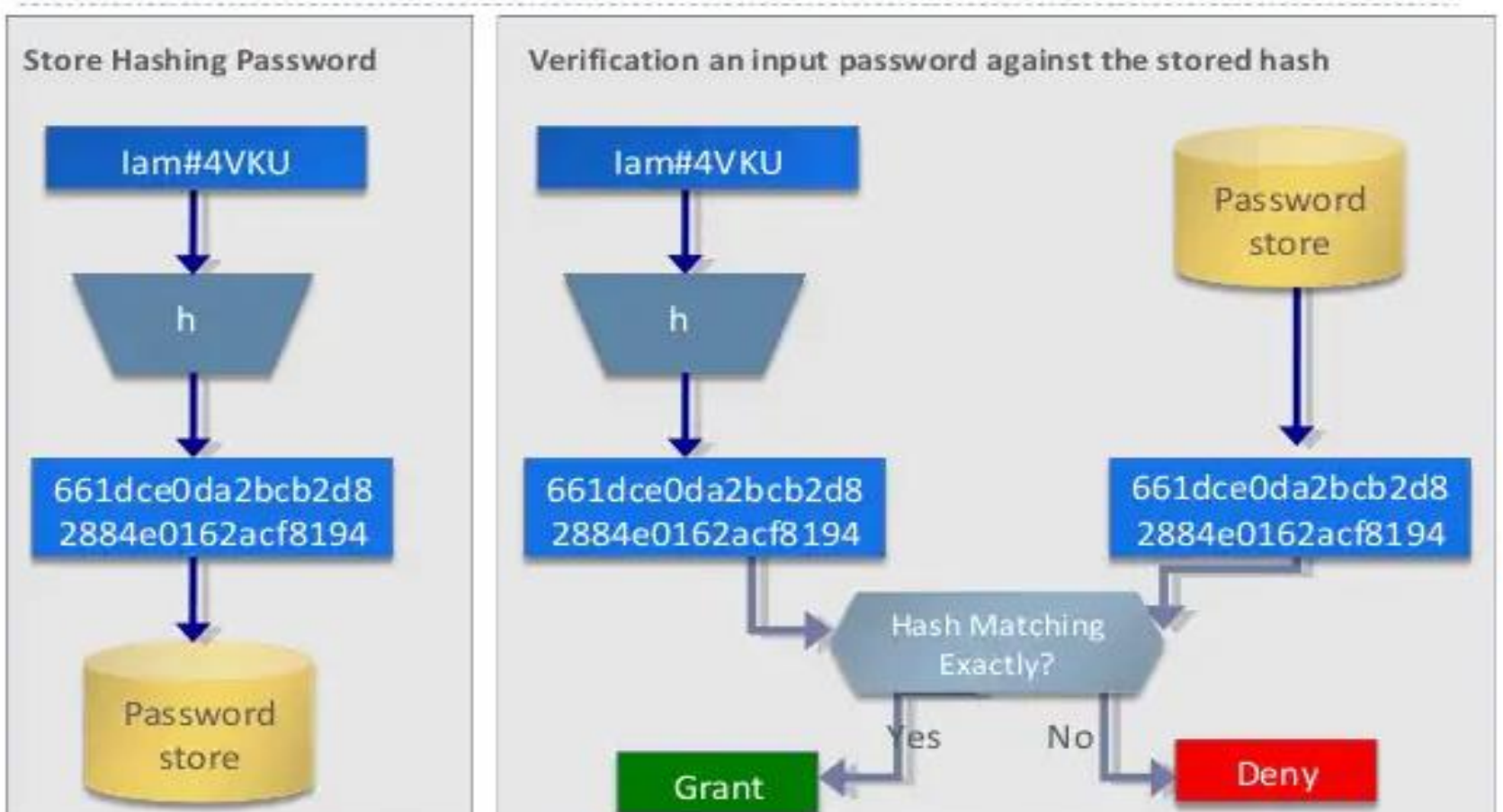
Bảo mật mật khẩu người dùng

Xác thực người dung khi đăng nhập



Bảo mật mật khẩu người dùng

Xác thực người dung khi đăng nhập



Bảo mật mật khẩu người dùng

1. Lưu trữ mật khẩu

	username	password	email
1	admin	nhx64312	nguyen@yahoo.com
2	devil	kin32xz	nam@hotmail.com
3	vampire	62ntt34	hung@gmail.com

Lưu trữ password không mã hóa

	username	password	Salt	email
1	admin	23dacd8cd768c95ad2e63cdc399c0535	64f84a9bdec1999e43c97ab12f8d9a36	nguyen@yahoo.com
2	devil	d400c472ab7a09ba87bf5c9715bbe118	66436db921568ae452f1e76a44e40aac	nam@hotmail.com
3	vampire	0d7b12ce9cf7ef3534aa5fee204eb0f5	1c798886f01910bad5a85fbec1c1bfea	hung@gmail.com

Lưu trữ password mã hóa bằng hàm hash MD5

Bảo mật mật khẩu người dùng

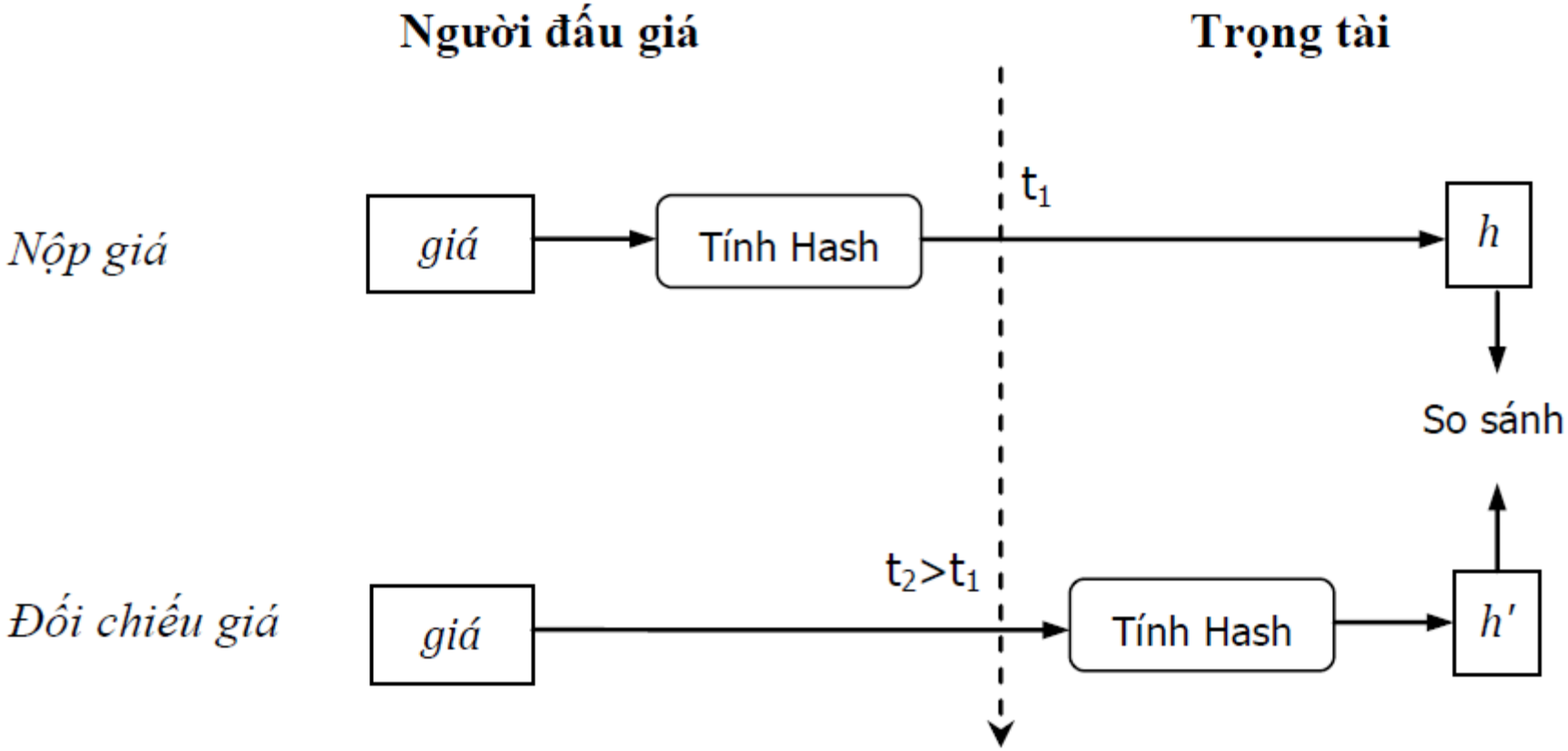
1. Lưu trữ mật khẩu
   - Đầu vào:
     - Một *password*
     - Một giá trị *salt* (ngẫu nhiên)
     - Số lần lặp *iteration*
   - Giải thuật :

```
key = hash(password + salt)
for 1 to iteration - 1 do
    key = hash(key + salt)
```

   - *Chú ý:* phép + ở đây là phép kết nối.

Bảo mật mật khẩu người dùng

2. Đấu giá trực tuyến (Trang 94)



Hình 5-6. Đấu giá bí mật

Bảo mật mật khẩu người dùng

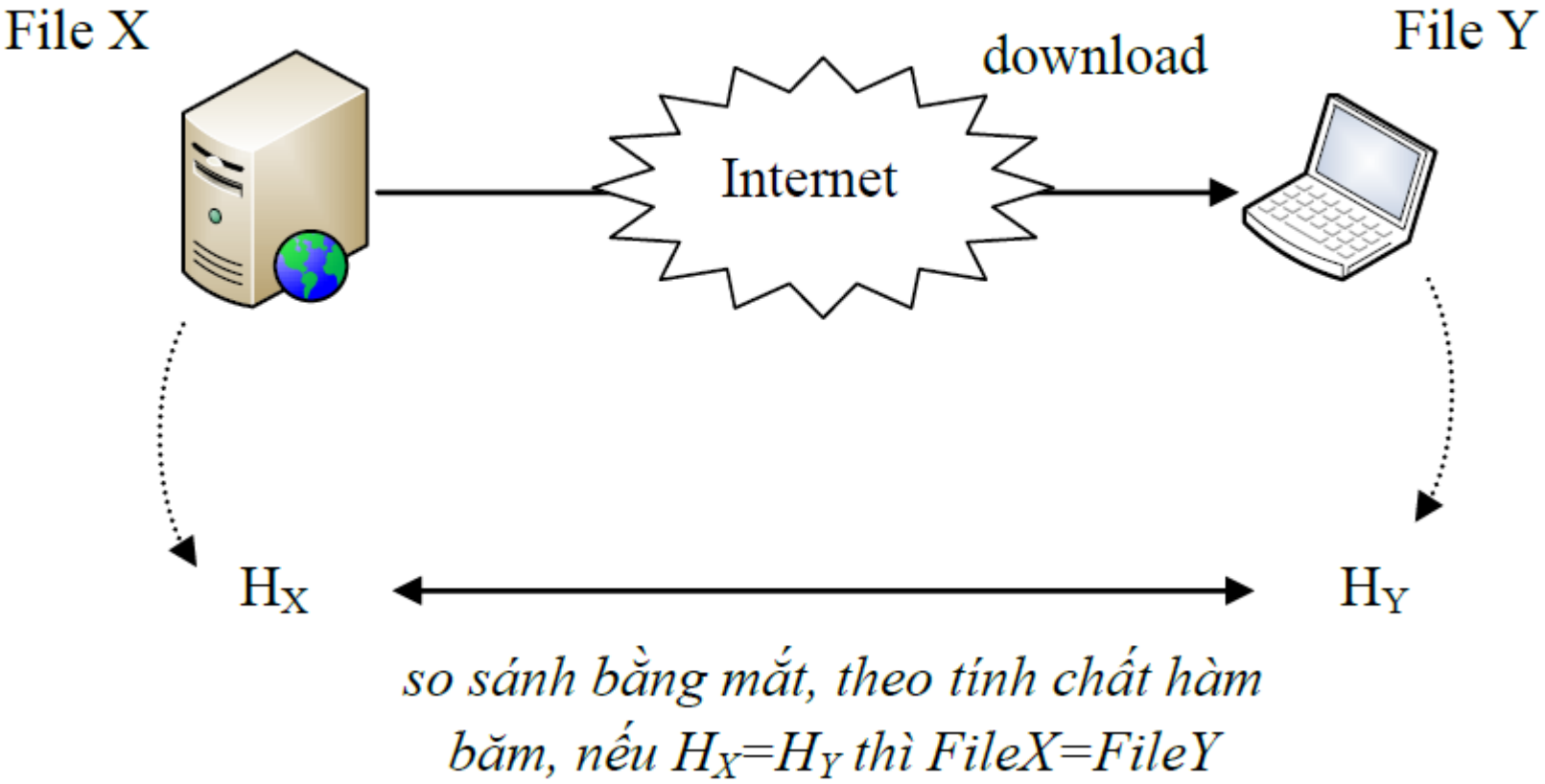
2. Đấu giá trực tuyến (Trang 94): Giả sử Alice, Bob và Trudy cùng tham gia đấu giá, họ sẽ cung cấp mức giá của mình cho trọng tài.

- Giả sử mức giá của Alice là 100, mức giá của Bob là 110, nếu Trudy thông đồng với trọng tài và biết được giá của Alice và Bob, Trudy có thể đưa ra mức giá 111 và thắng thầu.
- Có thể tránh những hình thức lừa đảo như vậy bằng cách sử dụng hàm băm. Từ mức giá bỏ thầu, Alice và Bob sẽ tính các giá trị băm tương ứng và chỉ cung cấp cho trọng tài các giá trị băm này.
- Vì hàm băm là một chiều, nếu trọng tài và Trudy bắt tay nhau thì cũng

không thể biết được giá của Alice và Bob là bao nhiêu. Đến khi công bố, Alice, Bob và Trudy sẽ đưa ra mức giá của mình. Trọng tài sẽ tính các giá trị băm tương ứng và so sánh với các giá trị băm đã nộp để bảo đảm rằng mức giá mà Alice, Bob và Trudy là đúng với ý định ban đầu của họ. Vì tính chống trùng của hàm băm nên Alice, Bob và Trudy không thể thay đổi giá so với ý định ban đầu.

Bảo mật mật khẩu người dùng

2. DownLoad File



Nhận diện sự xâm hại

Dùng nhận diện xâm hại (*intrusion detection*) và nhận diện virus (*virus detection*).

- Tính, lưu và bảo mật giá trị băm H(F) của các tập tin trong hệ thống (thể lưu trên CD-R)
- Kẻ xâm hại cần phải hiệu chỉnh F mà không thay đổi H(F)

Phát sinh số ngẫu nhiên giả

- **Dùng:**

  Xây dựng hàm ngẫu nhiên giả

  (*pseudorandom function - PRF*)

  hoặc

  Phát sinh số ngẫu nhiên giả (*pseudorandom number generator - PRNG*)

Tấn Công Hàm Băm

Kỹ thuật tấn công vét cạn: kẻ tấn công tạo ra một lượng lớn các văn bản và lần lượt tính toán, so sánh giá trị băm của chúng để tìm ra đụng độ. Trên thực tế, kích thước bảng băm theo các thuật toán thông dụng là 64 bit, 128 bit …, do đó thời gian tính toán để tìm được đụng độ theo phương pháp là rất lớn.

Tấn Công Hàm Băm

Kỹ thuật phân tích mã:

- Hạn chế của kỹ thuật tấn công vét cạn là số phép tính lớn, khi độ dài của bảng băm tương đối lớn (>=128 bit) thì việc tìm va chạm mất rất nhiều thời gian. Hiện nay có nhiều phương pháp cho ta kết quả tốt hơn kỹ thuật vét cạn, có thể kể đến như:
- Tấn công theo kiểu gặp nhau ở giữa (meet – in – the – middle attack)
- Tấn công khác biệt giữa các module (The modular differential attack)
- Boomerang Attacks …

Câu hỏi và bài tập

1. Để bảo đảm tính chứng thực dùng mã hóa đối xứng hay mã hóa khóa công khai, bản rõ phải có tính chất gì? Tại sao?
2. Nếu bản rõ là một dãy bít ngẫu nhiên, cần làm gì để bản rõ trở thành có cấu trúc?
3. Sử dụng MAC để chứng thực có ưu điểm gì so với chứng thực bằng mã hóa đối xứng?
4. Về mặt lý thuyết, giá trị Hash có thể trùng không? Vậy tại sao nói giá trị Hash có thể xem là “dấu vân tay của thông điệp”?
5. Tại sao để chứng thực một thông điệp M, người ta chỉ cần mã hóa khóa công khai giá trị Hash của M là đủ? Thực hiện như vậy có lợi ích gì hơn so với cách thức mã hóa toàn bộ M
6. Tìm hiểu phương pháp sử dụng hàm băm MD5 và SHA trong thư viện .NET, viết chương trình mã hóa password lưu trữ và kiểm tra password.

Câu hỏi và bài tập

1. Hãy xem xét hàm hash sau. Thông điệp có dạng là một dãy các số thập phân *M* = (*a1, a2, …, an*). Hàm hash được tính bằng công thức:

$$h = (\sum_{i=1}^{n} a_i) mod\ n.$$

2. Hàm hash trên có thỏa mãn các tính chất của một hàm hash như đã nêu trong phần 5.2 hay không? Giải thích lý do.
3. Giả sử Alice và Bob muốn tung đồng xu qua mạng (Alice tung và Bob đoán). Giao thức thực hiện như sau:
   - i. Alice chọn giá trị *X*=0 hay 1.
   - ii. Alice sinh một khóa *K* ngẫu nhiên gồm 256 bít
   - iii. Dùng AES, Alice tính *Y* = *E*(*X*||*R, K*) trong đó *R* gồm 255 bít bất kỳ
   - iv. Alice gửi *Y* cho Bob
   - v. Bob đoán *Z* là 0 hay 1 và gửi *Z* cho Alice
   - vi. Alice gửi khóa *K* cho Bob để Bob tính *X*||*R* = *D*(*Y, K*)
   - vii. Nếu *X*=*Z*, Bob đoán trúng. Nếu không Bob đoán sai.

Chứng tỏ rằng Alice có thể lừa Bob (chẳng hạn, Alice chọn *X*=1, thấy Bob đoán *Z*=1 thì Alice sẽ lừa như thế nào để Bob giải mã Y thì có *X*=0). Dùng hàm hash,hãy sửa đoạn giao thức trên để Alice không thể lừa được.

Câu hỏi và bài tập

- Trình bày ưu điểm, hạn chế của mã hóa đối xứng
- Trình bày ưu điểm, hạn chế của mã hóa bất đối xứng
- Nêu nguyên tắc của mã hóa khóa công khai? Tại sao khi truyền khóa sử dụng mã hóa khóa công khai không cần sử dụng kênh an toàn?
- Cho ví dụ về hàm một chiều.
- Cho ví dụ về hàm cửa lật một chiều.
- Trình bày nghịch lý ngày sinh nhật.

THANKS YOU