MẬT MÃ ỨNG DỤNG TRONG AN TOÀN THÔNG TIN

(Applied Cryptography In Secure Information System)

ThS. Bùi Hữu Đông
buihuudong19@gmail.com
0903.82.36.46

Học viện Kỹ thuật Mật mã, khoa An toàn thông tin

Ngày 26 tháng 11 năm 2020

Buổi 06

HÀM BĂM & CHỨNG THỰC THÔNG ĐIỆP

Tổng quan về Mã chứng thực

- Giả sử một message có ý nghĩa thì phải có một cấu trúc nào đó.

  Thí dụ: một câu văn có ý nghĩa khi các các chữ kết hợp với nhau theo quy tắc và cấu trúc từ vựng & ngữ pháp của ngôn ngữ cụ thể đó.
- Khi đó, kẻ phá mã can thiệp sửa đổi bản mã thì bản giải mã sẽ là một chuỗi bit vô nghĩa $=>$ người nhận biết được bản tin ban đầu đã bị sửa.

Nhận xét về Mã chứng thực

Ta có hai nhận xét sau về tính chứng thực của mã hóa (gồm mã hóa đối xứng và bất đối xứng):

- Kẻ phá mã không thể tìm ra bản mã C_T sao cho khi *người nhận* giải mã bằng khóa K_{AB} (hay khóa K_{UA} với mã công khai) cho ra bản rõ P_T có ý nghĩa theo ý muốn của *kẻ phá mã*
- *Kẻ thám mã* không thể tìm ra C_T sao cho P_T là một bản tin có ý nghĩa mà là dãy bít hỗn loạn và không có cấu trúc.

Nhận xét về Mã chứng thực

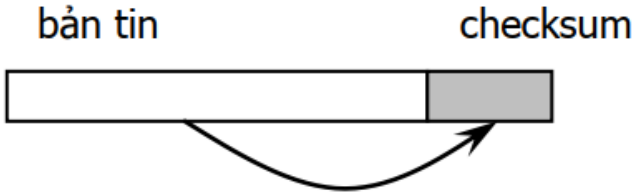
- Trong thực tế, có nhiều loại data có các bít gần như là *random*. Thí dụ: dữ liệu hình ảnh bitmap, âm thanh hay video.

  => *việc nhận ra dãy bít thế nào là ý nghĩa thì rất khó khăn*.
- Vì thế, trong thực tế ta luôn coi dãy bít nào cũng có thể có ý nghĩa
- Như thế, với phương pháp mã đối xứng và bất đối xứng không thể đảm bảo tính chứng thực.

=> Để giải quyết vấn đề này, mã hóa phải vận dụng khái niệm *redundancy* của lĩnh vực truyền số liệu.

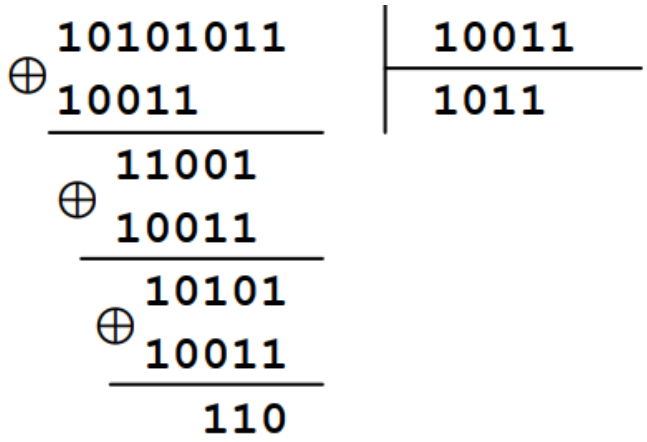
Phương pháp redundancy - Checksum

- Phương pháp redundancy thực hiện việc thêm ít dữ liệu (checksum) để biến bản tin từ dãy bít ngẫu nhiên => thành dãy bít có cấu trúc
- Phương pháp checksum phổ biến là *cyclic redundancy check* (CRC)



Phương pháp checksum CRC

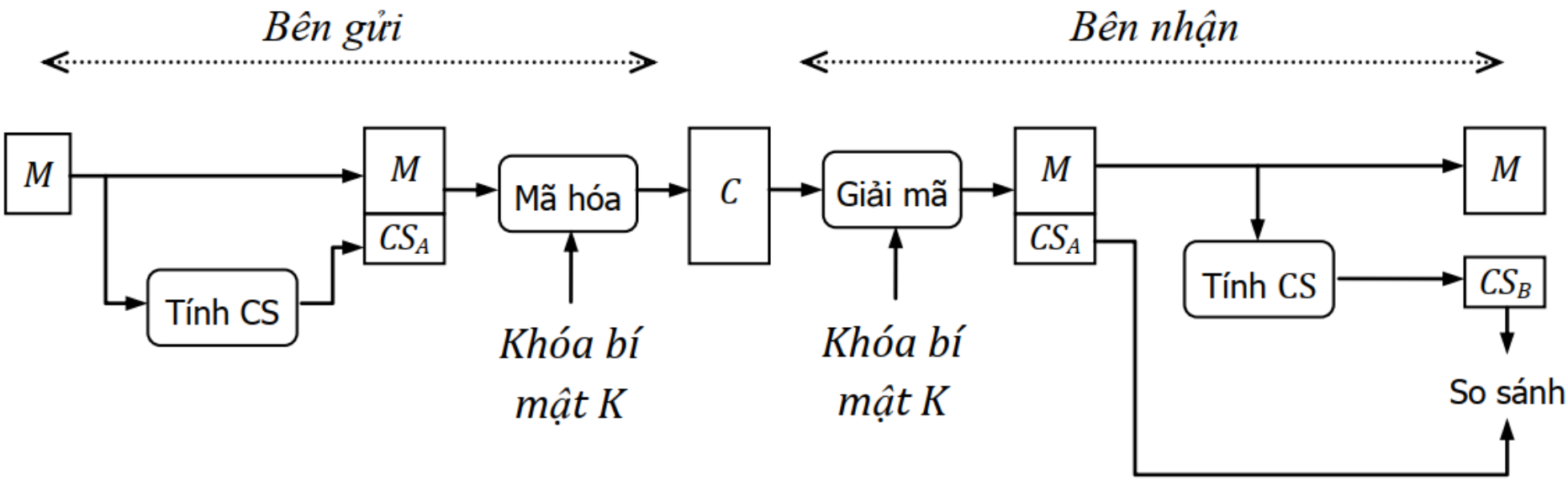
- *Mô tả*:
  - Theo CRC một đoạn bít ngắn được chọn làm số chia, lấy dãy bít của message chia cho số này và phần dư của phép chia này được gọi là giá trị checksum CRC.
  - Phép chia này khác phép chia thông thường là dùng phép XOR thay cho phép trừ.
- *Thí dụ*: Giả sử thông điệp là 10101011 và số chia là 10011, quá trình tính như sau:



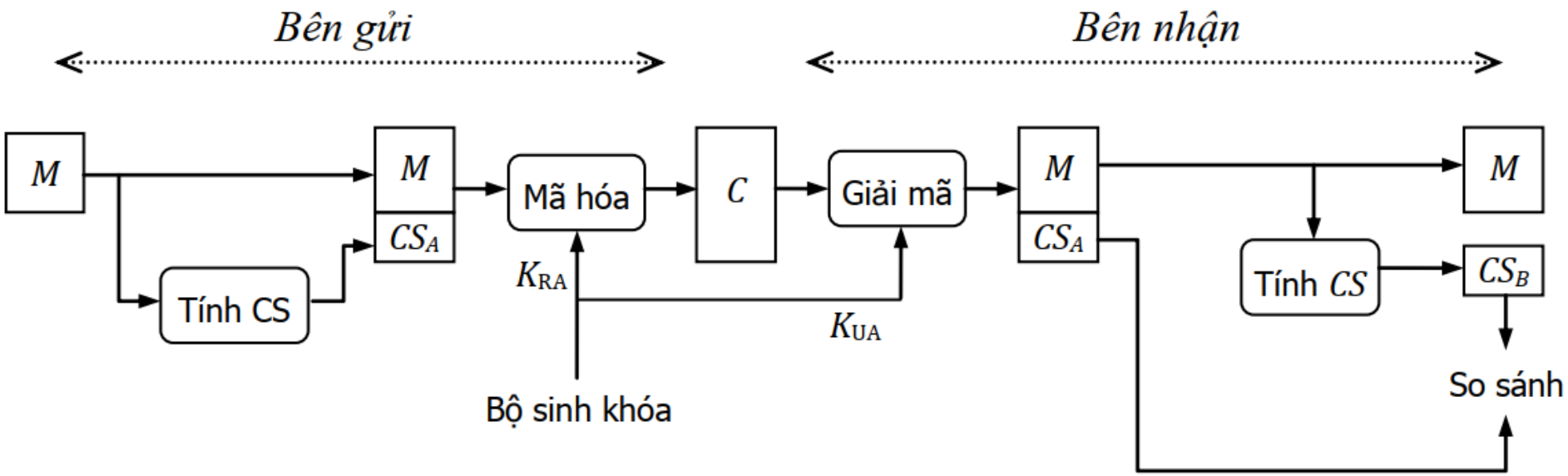
Phương pháp checksum CRC

- *Quá trình chứng thực*:
  - Như thí dụ trên, phần dư 0110 sẽ được gửi kèm tới người nhận, người nhận cũng thực hiện CRC như lúc gửi. Nếu so sánh hai giá trị *checksum* mà trùng khớp => thông điệp không bị lỗi khi truyền tin
  - Nếu ta áp dụng cơ chế checksum vào việc chứng thực thông điệp, sau đó append checksum này vào dãy bit ban đầu => được dãy bít có cấu trúc.
  - Thực hiện mã hóa (đối xứng hoặc bất đối xứng) trên dãy bít mới (bao gồm checksum)
  - Size của checksum nhỏ => không ảnh hưởng tới hiệu năng mã hóa và băng thông truyền tin.

Mô hình mã hóa đối xứng và bảo mật với checksum



Mô hình mã hóa bất đối xứng với checksum



Nhận xét chứng thực với checksum

- Kẻ phá mã nếu có được và thực hiện sửa bản mã C thì bản giải mã của người nhận là M_T và CS_T sẽ không còn tính cấu trúc
- Khi hai giá trị checksum khác nhau => bản tin ban đầu bị thay đổi
- Nếu hàm checksum có độ phức tạp cao thì xác suất để $CS_B = CS_T$ là rất thấp.

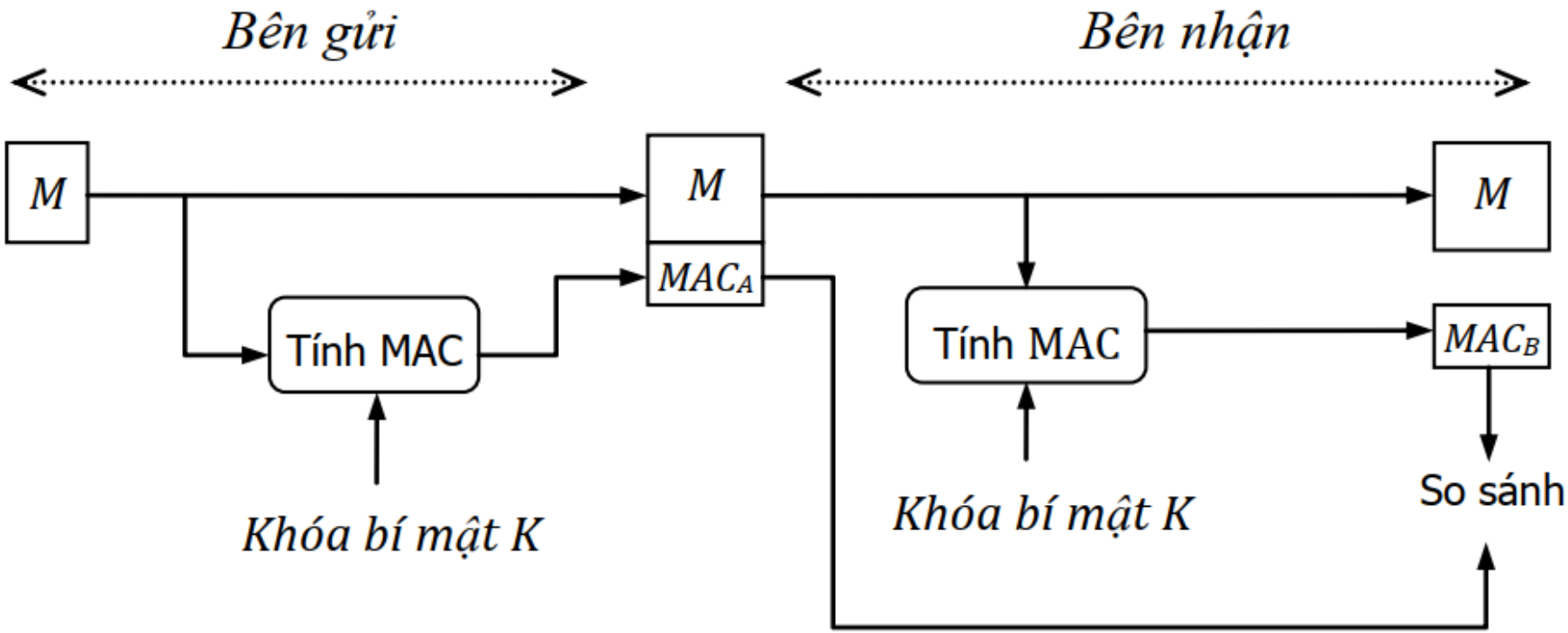
Phương pháp chứng thực khác

- Chứng thực thông điệp - Message Authentication Code (MAC)
- Hàm băm (Hash function)

Mã chứng thực thông điệp - MAC

- Mã chứng thực thông điệp (MAC) có thể coi là một dạng checksum của mã hóa, được tính theo công thức $MAC = C(M, K)$, trong đó:
  - M là thông điệp cần tính MAC
  - K là khóa bí mật được chia sẽ giữa người gởi và người nhận
  - C là hàm tính MAC
- Với MAC cả người gửi và người nhận đều biết khóa bí mật K

Mô hình chứng thực với MAC



Mô hình chứng thực với MAC

- Nhận xét với MAC:
  - Kẻ thám mã, nếu chỉ sửa M thành M_T thì giá trị MAC_B sẽ khác MAC_A và người nhận phát hiện được
  - Nếu kẻ phá mã muốn sửa thông điệp mà người nhận không biết, thì cần sửa luôn MAC_A thành MAC_T tính được từ M_T
  - Tuy nhiên kẻ phá mã không biết khóa K, do đó không tính được MAC_T cần thiết
  - Mô hình trên không đảm bảo tính bảo mật. Để có tính bảo mật, M và MACA cần được mã hóa trước khi truyền đi

Mô hình chứng thực với MAC

Tại sao không dùng mã hóa đối xứng mà cần dùng *MAC*?

Hàm Băm - Hash Function

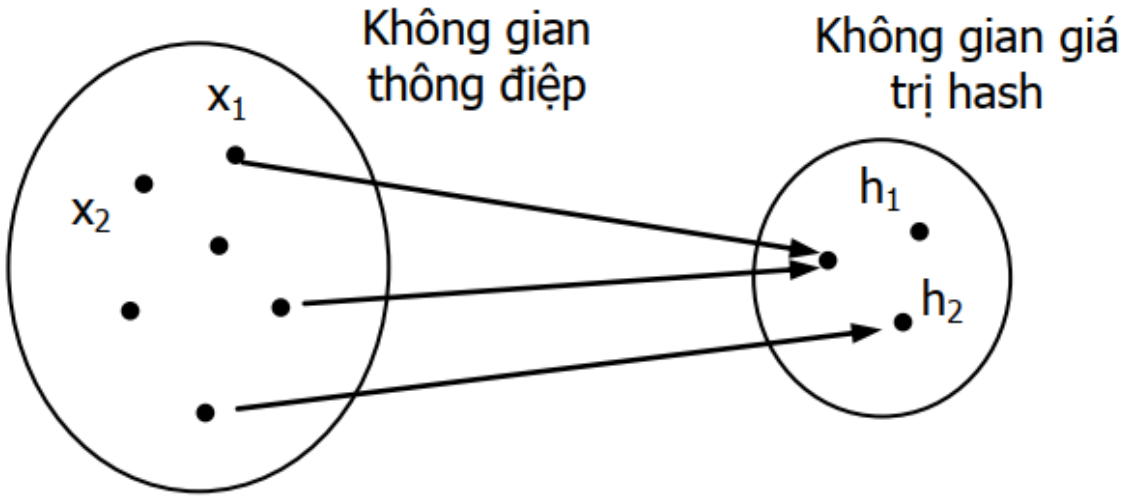
Giới thiệu Hàm băm

Với checksum *CRC* cho phép hai dãy bít có cùng mã checksum, hàm băm $H(x)$ là hàm tính checksum mạnh, nó thỏa mãn các yêu cầu:

- H có thể áp dụng cho các thông điệp x với các độ dài khác nhau
- Kích thước của output $h = H(x)$ là cố định và nhỏ
- Là hàm cửa sập - với một h cho trước, không thể tìm lại được x sao cho $h = H(x)$ (về mặt thời gian tính toán)
- Tính chống trùng yếu: cho trước một x, không thể tìm $y \neq x$ sao cho $H(x) = H(y)$
- Tính chống trùng mạnh: không thể tìm ra cặp x, y bất kỳ $(x \neq y)$ sao cho $H(x) = H(y)$, hay nói cách khác nếu $H(x) = H(y)$ thì có thể chắc chắn rằng $x = y$.

Nhận xét Hàm băm

- Kích thước của input x là bất kỳ còn kích thước của h là nhỏ
- Thí dụ: kích thước của x là 512 bít còn kích thước của h là 128 bít $=>$ Như vậy trung bình có khoảng 2^{384} giá trị x mà có cùng giá trị h và ta không thể loại bỏ được sự trùng lặp này.



Thí dụ Hàm băm

Thí dụ với con người chúng ta:

- Hàm lấy khuôn mặt ?
- Hàm lấy dấu vân tay?

Yêu cầu của Hàm băm

- Giá trị băm $h = H(x)$ không được quá lớn, nếu h lớn thì có thể giải quyết việc chống trùng => Tốn dung lượng truyền tin
- Vậy kích thước h là bao nhiêu ? để thực hiện việc chống trùng hiệu quả?
- Ta xem xét ví dụ về bài toán ngày sinh nhật để hiểu rõ hơn !

Bài toán ngày sinh nhật

- *Bài thứ nhất*: Giả sử trong phòng có 30 người. Vậy xác suất để có hai người có cùng ngày sinh là bao nhiêu phần trăm?
  - ◇ Theo nguyên lý Dirichlet thì cần $365 + 1 = 366$ người để tìm thấy 2 người có cùng ngày sinh với xác suất 100% => ta nghĩ với 30 người thì xác suất 2 người trùng ngày sinh là nhỏ (nhỏ hơn 50%)
  - ◇ Thực tế đã chứng minh với 23 người là đủ để xác suất >= 50%.

Bài toán ngày sinh nhật

- Ta đánh số thứ tự M người lần lượt là 0 đến $M-1$
- Xác suất để người thứ 1 khác với người thứ 0 là 364/365, tiếp theo xác suất để người thứ 2 có ngày sinh khác với người thứ 0 và thứ 1 là 363/365. Tiếp tục như vậy người thứ $M-1$ xác suất để người này khác ngày sinh với với tất cả người trước là $(365-M+1)/365$
- Vậy xác suất để M người này có ngày sinh khác nhau là:

$$p(M) = \left(\frac{364}{365}\right)\left(\frac{363}{365}\right)\ldots\left(\frac{365-M+1}{365}\right) = \left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\ldots\left(1-\frac{M-1}{365}\right)$$

Bài toán ngày sinh nhật

- Xét hàm lũy thừa e^x, chúng ta đã biết một xấp xỉ của e^x khi x nhỏ là $e^x = 1 + x$. Do đó $p(M)$ là:

$$p(M) \approx e^{\frac{-1}{365}} e^{\frac{-2}{365}} e^{\frac{-M+1}{365}} = e^{-\frac{1+2+3+\ldots+(M-1)}{365}} = e^{-\frac{M(M-1)}{2\ x\ 365}}$$

- Nên, xác suất để tồn tại ít nhất hai người có ngày sinh giống nhau là:

$$1 - p(M) \approx 1 - e^{-\frac{M(M-1)}{2\ x\ 365}}$$

- Để xác suất này $\geq 50\%$ ta có:

$$1 - e^{-\frac{M(M-1)}{2\ x\ 365}} \geq \frac{1}{2}$$

hay (*):

$$e^{-\frac{M(M-1)}{2\ x\ 365}} \leq \frac{1}{2} <=> M(M-1) \geq 2\ x\ 365\ x\ log_e 2 => M \geq 23$$

Bài toán ngày sinh nhật

• *Bài thứ 2*: Giả sử bạn đang ở trong một căn phòng với M người khác. Hỏi M tối thiểu là bao nhiêu để tồn tại một người có cùng ngày sinh với bạn với xác suất lớn hớn 50% ?

- ◇ Xác suất để một người không có cùng ngày sinh với bạn là $\frac{364}{365}$. Theo đó, xác suất để M người đều khác ngày sinh với bạn là: $\frac{364}{365}^M$
- ◇ Từ đó ta có xác suất để tồn tại ít nhất một người có cùng ngày sinh với bạn là: $1 - (\frac{364}{365})^M$
- ◇ Để xác suất này lớn hơn 50% thì $M \geq 253$. Vậy tối thiểu phải có 253 người.

Bài toán ngày sinh nhật và Hàm băm

• Ta thấy tính chống trùng mạnh giống như bài toán 1, còn chống trùng yếu như bài toán 2

• Ta giả sử, $h = n$ bit, số lượng giá trị có của h là $N = 2^n$ và ta giả sử 2^n giá trị băm này đều là ngẫu nhiên, có khả năng xuất hiện như nhau. Ta thay giá trị 365 của (*) bằng 2^n:

$$M(M-1) \geq 2\ x\ 2^n\ x\ log_e 2$$

Giải bất phương trình này ta có xấp xỉ:

$$M \geq \sqrt{2^n} = 2^{\frac{n}{2}}$$

Bài toán ngày sinh nhật và Hàm băm

- Ta phải thử khoảng $2^{\frac{n}{2}}$ thông điệp khác nhau để tìm ra hai thông điệp có cùng giá trị băm (với xác suất $\geq 50\%$)
- Nếu $n = 128$ thì phải thử khoảng 2^64 thông điệp, một con số khá lớn, nghĩa là hàm băm này đạt được tính chống trùng mạnh
- Do đó việc phá hàm băm cũng khó giống như là việc tấn công vét cạn khóa của mã hóa đối xứng DES
- Tóm lại, tính chất chống trùng của hàm băm có thể phát biểu:

$$\nexists x \neq y \mid H(x) = H(y)$$

Nói cách khác: $\forall x, y$ nếu $H(x) = H(y)$ thì $x = y$

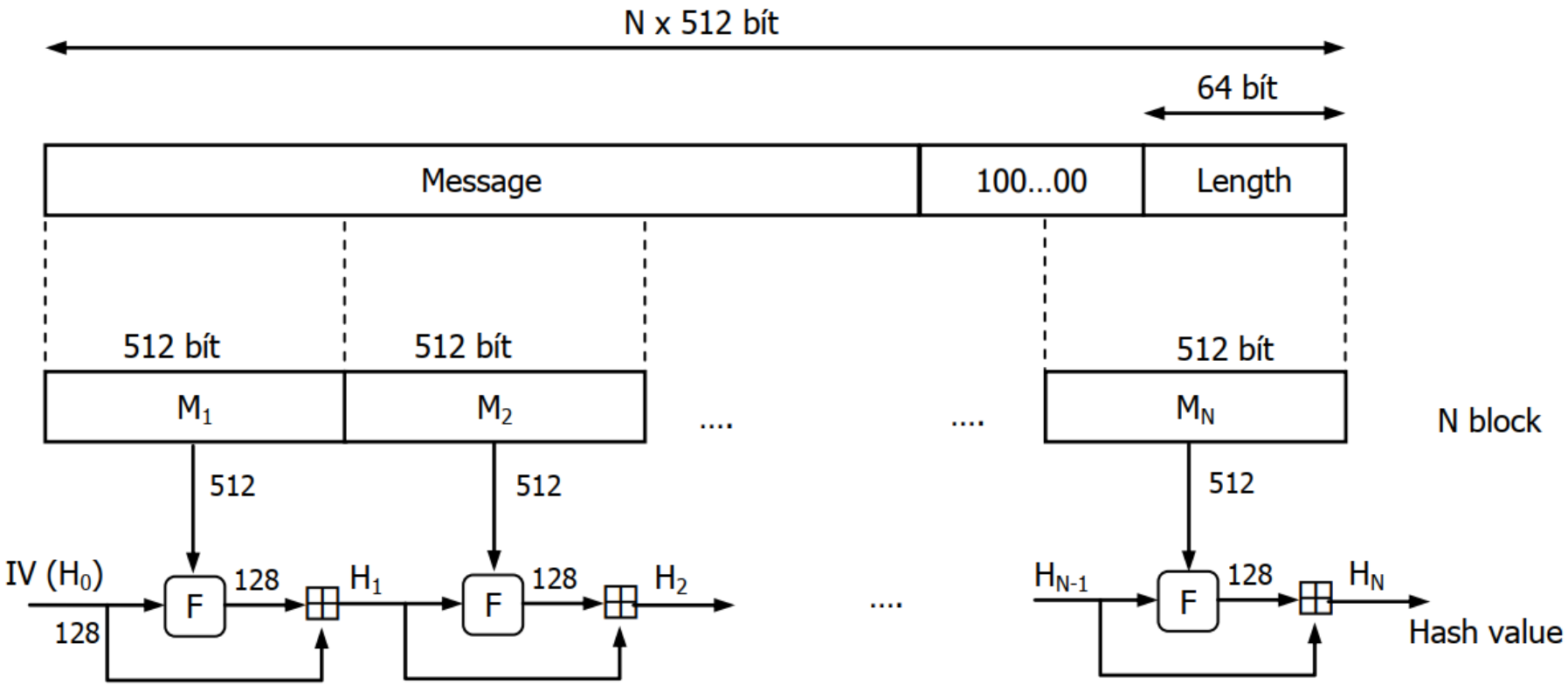
Hàm băm phổ biến

- MD-5 (Message Digest - 128 bit): Được phát minh bởi Ron Rivest
- SHA-1 (Secure Hash Algorithm): SHA-1 đã được chính phủ Mỹ chọn làm chuẩn quốc gia. SHA-1 có kích thước giá trị băm là 160 bít.

Ngày nay còn có ba phiên bản khác của SHA là SHA-256, SHA-384, SHA-512 mà có kích thước giá trị băm tương ứng là 256, 384 và 512 bít.

Hàm băm MD5

- MD5 cho ra kích thước của giá trị băm là 128 bit và kích thước đầu vào tối đa là 2^{64} bit. Sơ đồ MD5 như sau:



Hàm băm MD5

Theo sơ đồ của MD5 ta thấy:

- Trước tiên, thông điệp M được thêm dãy bit padding 100....00
- Sau đó thêm vào chiều dài (trước khi padding) của thông điệp được biểu diễn bằng 64 bít
- Như vậy chiều dài của dãy bít padding được chọn sao cho cuối cùng thông điệp có thể chia thành N block 512 bít $M_1, M_2, ..., M_N$
- Quá trình tính giá trị băm của thông điệp là quá trình lũy tiến

Quá trình Băm của Hàm MD5

- Quá trình tính giá trị băm của thông điệp là quá trình lũy tiến
- Trước tiên block M_1 kết hợp với giá trị khởi tạo H_0 thông qua hàm F để tính giá trị hash H_1
- Sau đó block M_2 được kết hợp với H_1 để cho ra giá trị hash là H_2
- Cứ như vậy cho đến block M_N thì ta có giá trị băm của toàn bộ thông điệp là H_N

Mô tả các hàm trong MD5

- Hàm H_0: là một dãy 128 bít được chia thành 4 từ 32 bít, ký hiệu 4 từ 32 bít trên là $abcd$. a, b, c, d là các hằng số như sau (viết dưới dạng thập lục phân):

$$a = 01234567$$

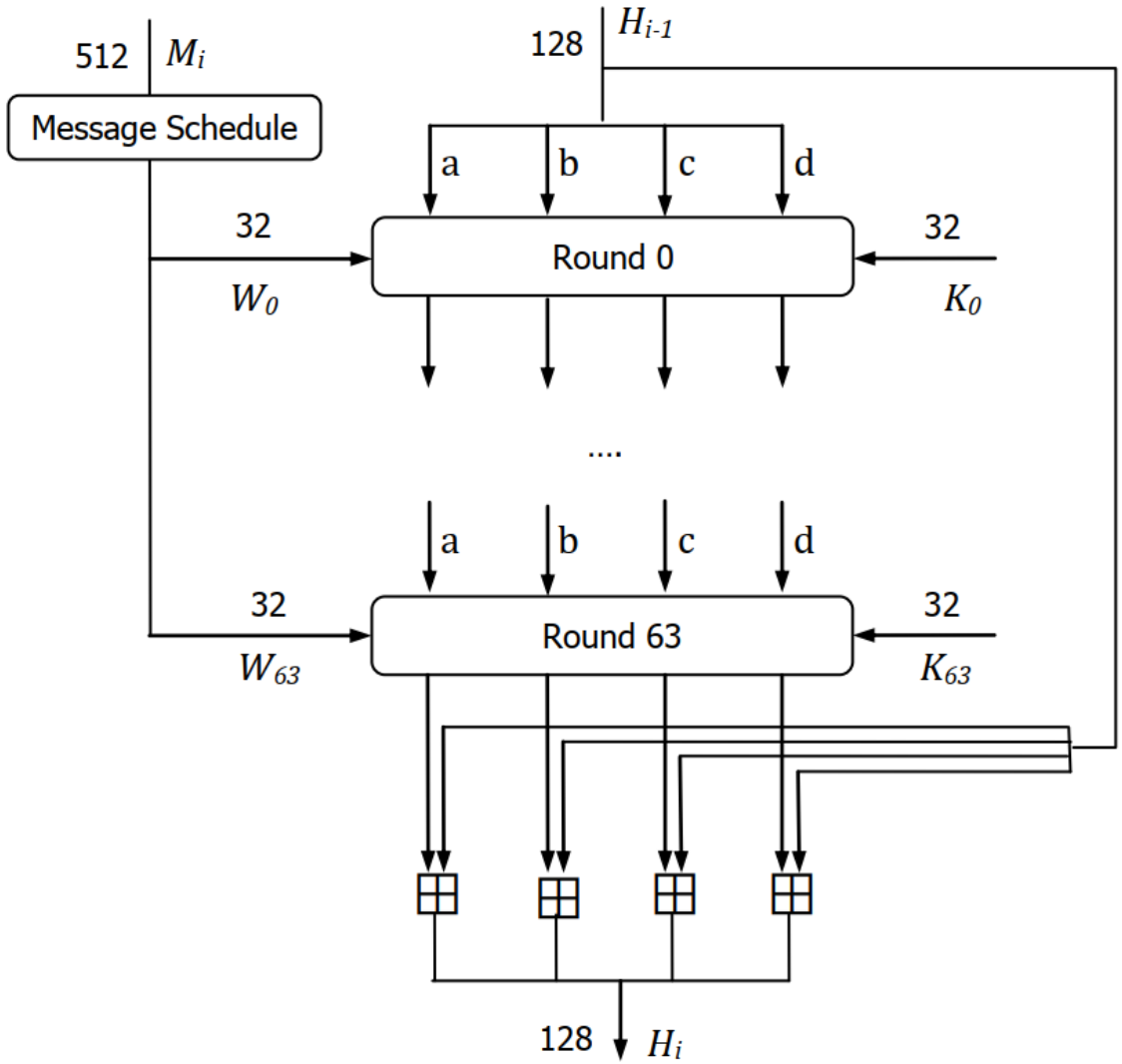
$$b = 89abcdef$$

$$c = fedbca98$$

$$d = 76543210$$

Mô tả các hàm trong MD5

- Cấu trúc hàm F:



Mô tả các hàm trong MD5

- Các thành phần của hàm F:
  - Tại mỗi bước lũy tiến, các giá trị $abcd$ của giá trị hash H_{i-1} được biến đổi qua 64 vòng từ 0 đến 63.
  - Tại vòng thứ j sẽ có 2 tham số là K_j và W_j đều có kích thước 32 bít. Các hằng số K_j được tính từ công thức:

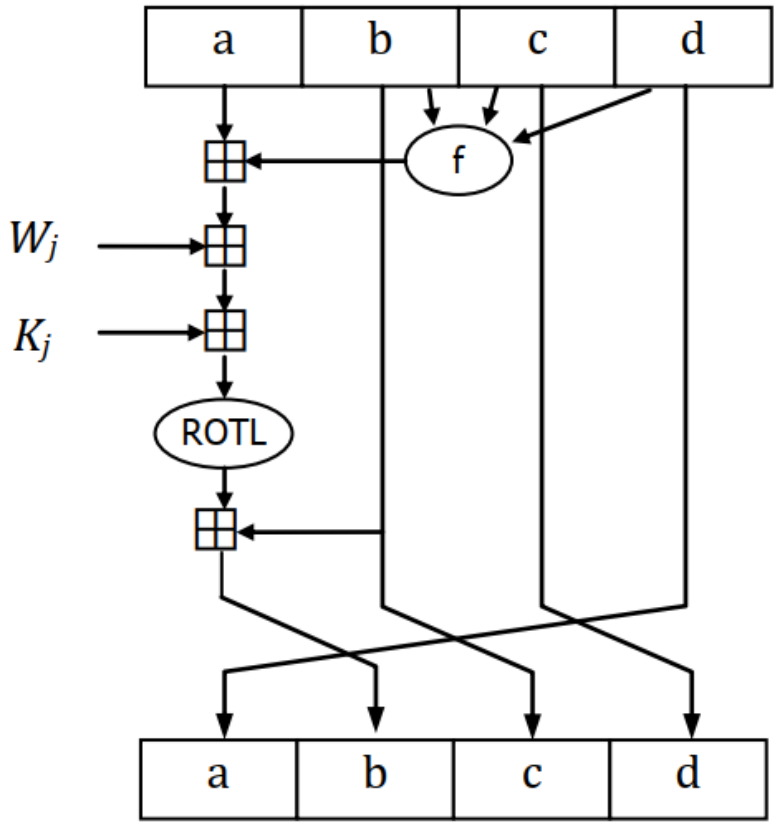
    K_j là phần nguyên của số $2^{32} abs(sin(i))$ với i biểu diễn theo radian.
  - Giá trị block M_i 512 bít được biến đổi qua một hàm *message schedule* cho ra 64 giá trị $W_0, W_1, ..., W_{63}$ mỗi giá trị 32 bít
  - Block M_i 512 bít được chia thành 16 block 32 bít ứng với các giá trị $W_0, W_1, ..., W_{15}$ $(16\ x\ 32 = 512)$

Mô tả các hàm trong MD5

- Các thành phần của hàm F <tiếp>:
  - Tiếp theo, 16 giá trị này được lặp lại 3 lần tạo thành dãy 64 giá trị
  - Sau vòng cuối cùng, các giá trị *abcde* được cộng với các giá trị abcd của H_{i-1} để cho ra các giá trị *abcd* của H_i. Phép cộng ở đây là phép cộng modulo 2^{32}

Mô tả các thành phần trong MD5

- Cấu trúc của một vòng:
  - Việc biến đổi các giá trị *abcd* trong vòng thứ *i* được thể hiện trong hình bên dưới:



Mô tả các thành phần trong MD5

- Cấu trúc của một vòng <tiếp>:
  - Ở đây $b \rightarrow c, c \rightarrow d, d \rightarrow a$. Giá trị b được tính qua hàm:

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

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

Trong đó:

- Hàm $f(x, y, z)$:

$$f(x,y,z) = (x \wedge y) \vee (\neg x \wedge z) \quad \text{nếu là vòng 0 đến 15}$$
$$f(x,y,z) = (z \wedge x) \vee (\neg z \wedge y) \quad \text{nếu là vòng 16 đến 31}$$
$$f(x,y,z) = x \oplus y \oplus z \quad \text{nếu là vòng 32 đến 48}$$
$$f(x,y,z) = y \oplus (x \vee \neg z) \quad \text{nếu là vòng 49 đến 63}$$

Mô tả các thành phần trong MD5

- Cấu trúc của một vòng <tiếp>:
  - Hàm $ROTL(t, s)$: t được dịch vòng trái s bít, với s là các hằng số cho vòng thứ i như sau:

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

Mô tả các thành phần trong MD5

- Cấu trúc của một vòng <tiếp>:
  - Phép $+$ là phép cộng modulo 2^{32}

Phương pháp hàm băm - HMAC

◦ HMAC: Hàm băm cũng có thể dùng để tính MAC bằng cách truyền thêm khóa bí mật K vào hàm băm. Lúc này, giá trị kết xuất được gọi là HMAC

$$HMAC = H(M||K)$$

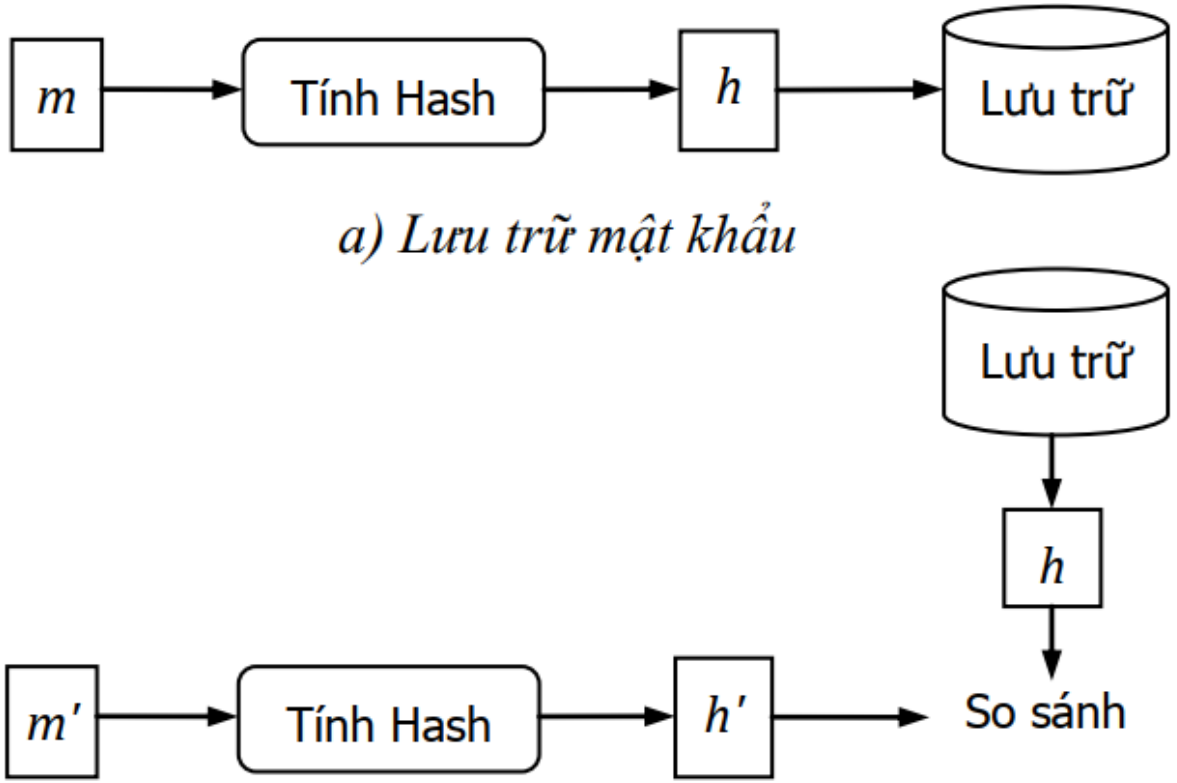
Phương pháp hàm băm SHA

SV tự tìm hiểu về:

- SHA-1
- SHA-512

Một số ứng dụng của hàm băm

- Lưu trữ mật khẩu



Một số ứng dụng của hàm băm

- Lưu trữ mã hóa mật khẩu

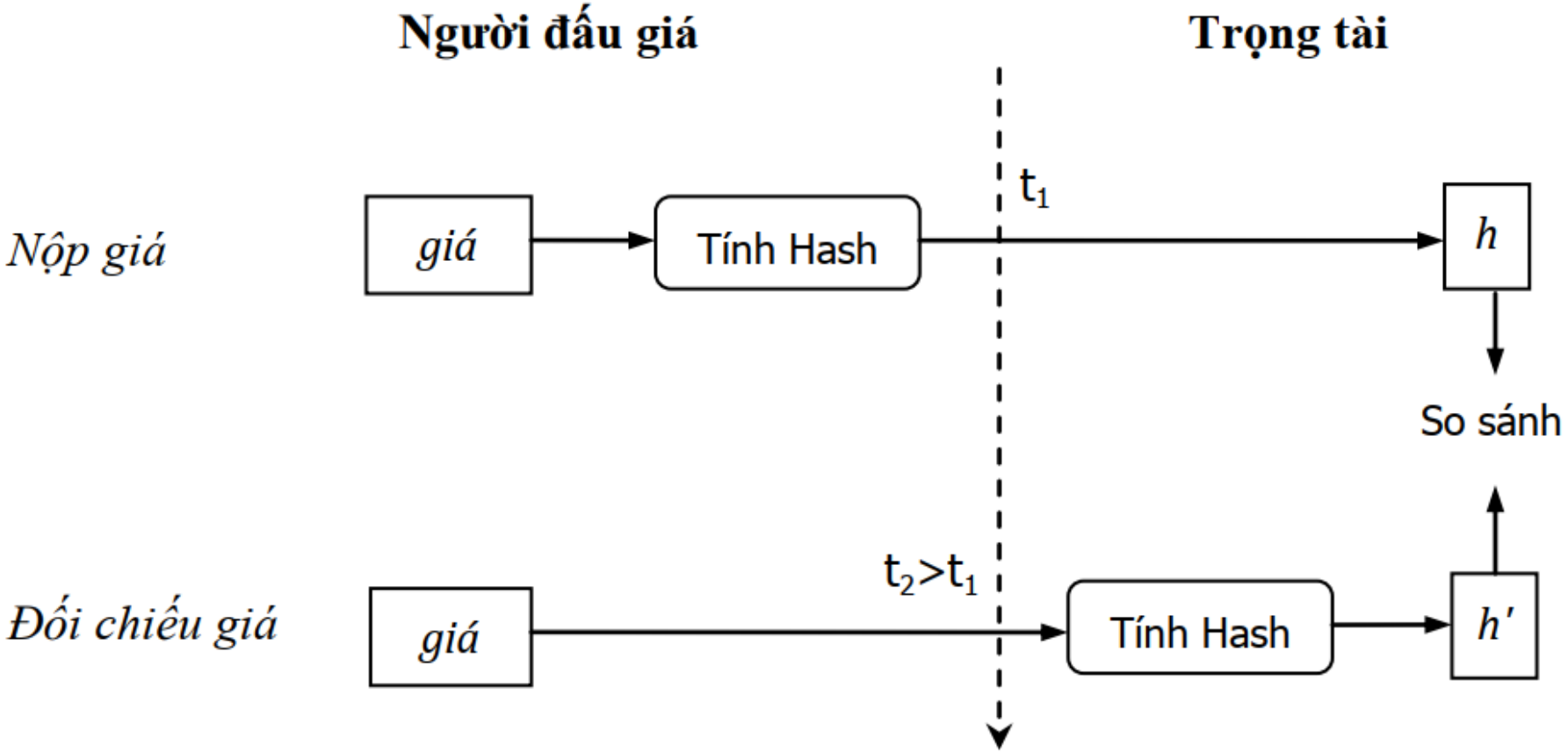
	username	password	email
1	admin	nhx64312	nguyen@yahoo.com
2	devil	kin32xz	nam@hotmail.com
3	vampire	62ntt34	hung@gmail.com

Lưu trữ password không mã hóa

	username	password	Salt	email
1	admin	23dacd8cd768c95ad2e63cdc399c0535	64f84a9bdec1999e43c97ab12f8d9a36	nguyen@yahoo.com
2	devil	d400c472ab7a09ba87bf5c9715bbe118	66436db921558ae452f1e76a44e40aac	nam@hotmail.com
3	vampire	0d7b12ce9cf7ef3534aa5fee204eb0f5	1c798886f01910bad5a85fbec1c1bfea	hung@gmail.com

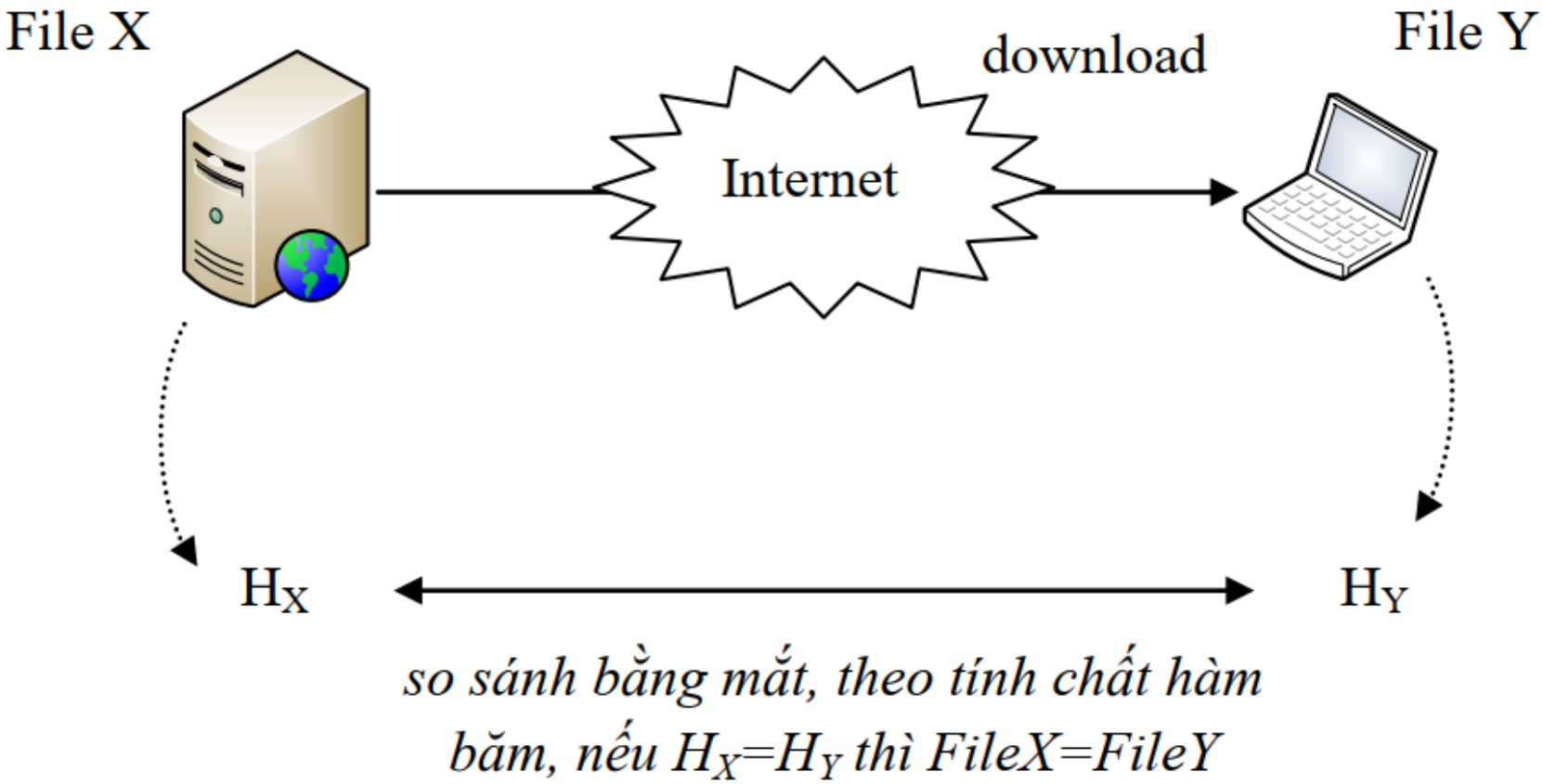
Một số ứng dụng của hàm băm

- Đấu giá trực tuyến



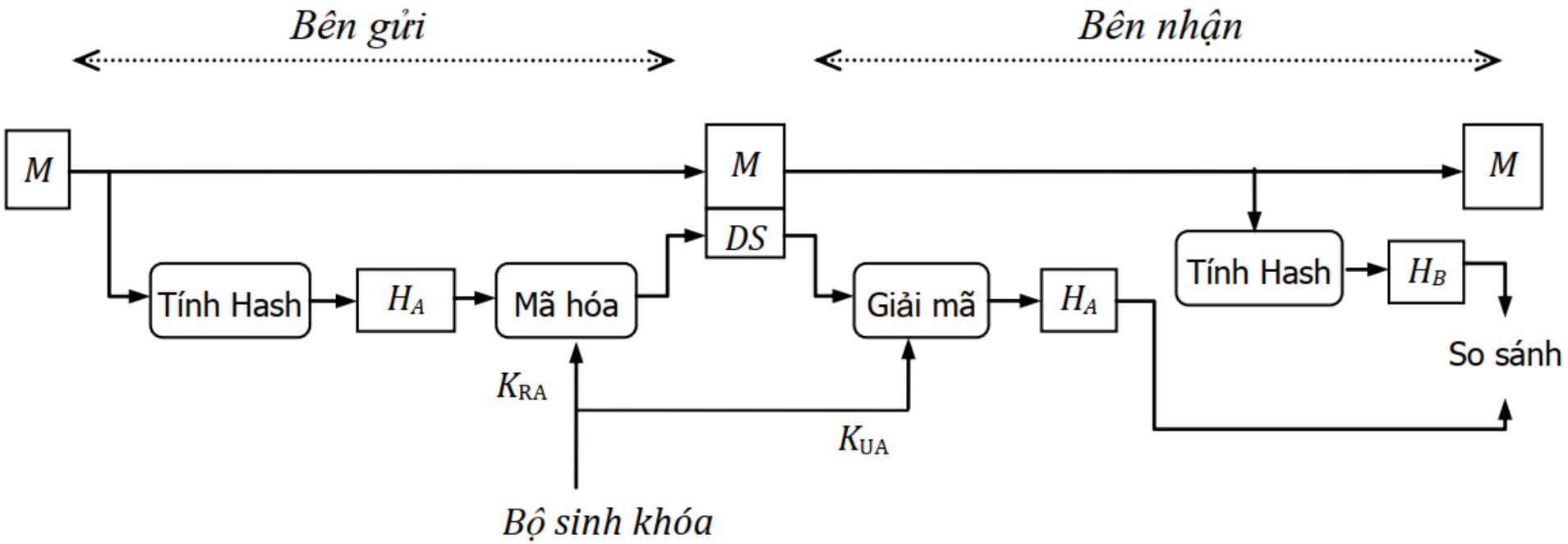
Một số ứng dụng của hàm băm

- Download file



Một số ứng dụng của hàm băm

- Chữ ký điện tử



DS: Data signature – chữ ký điện tử