BÀI 3.
XÁC THỰC THÔNG ĐIỆP

Bùi Trọng Tùng,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

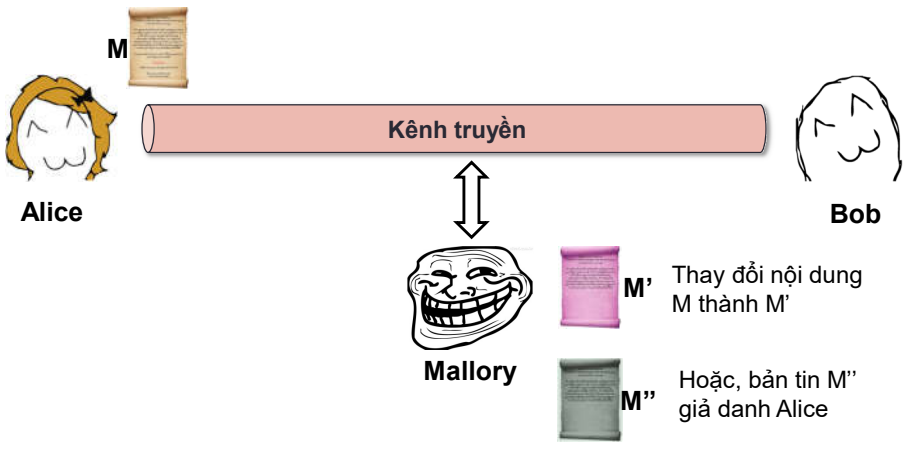
Nội dung

- Các vấn đề xác thực thông điệp
- Mã xác thực thông điệp (MAC)
- Hàm băm và hàm băm mật HMAC

1. ĐẶT VẤN ĐỀ

3

1. Đặt vấn đề



4

Xác thực thông điệp

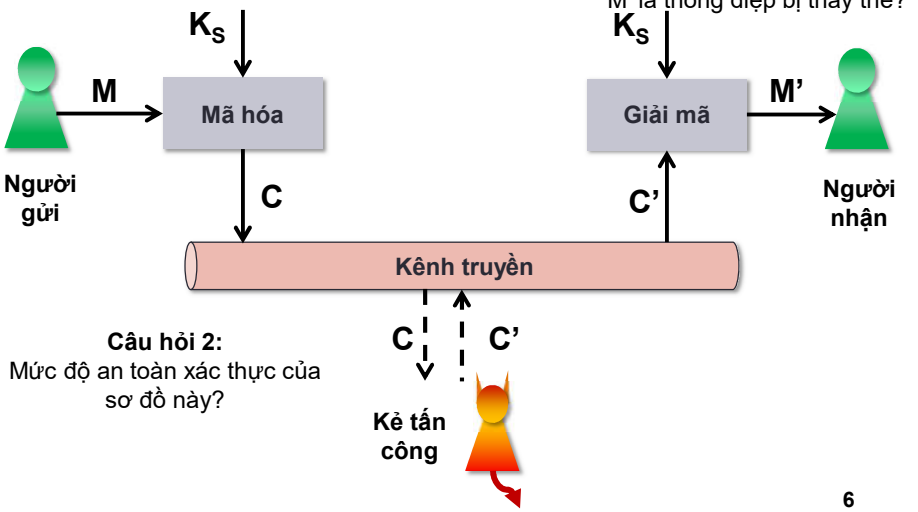
- Bản tin phải được xác minh:
  - ➢Nội dung toàn vẹn: bản tin không bị sửa đổi
    - ✓Bao hàm cả trường hợp Bob cố tình sửa đổi
  - ➢Nguồn gốc tin cậy:
    - ✓Bao hàm cả trường hợp Alice phủ nhận bản tin
    - ✓Bao hàm cả trường hợp Bob tự tạo thông báo và "vu khống" Alice tạo ra thông báo này
  - ➢Đúng thời điểm
  - →Các dạng tấn công điển hình vào tính xác thực: Thay thế (Substitution), Giả danh (Masquerade), tấn công phát lại (Replay attack), Phủ nhận (Repudiation)

5

5

Xác thực bằng mật mã khóa đối xứng

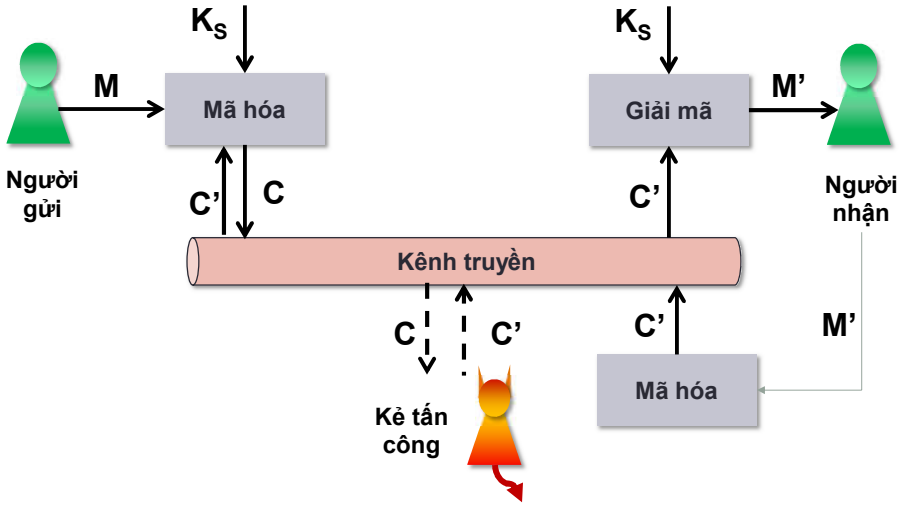
- Nhắc lại sơ đồ mật mã khóa đối xứng



6

6

Kịch bản minh họa

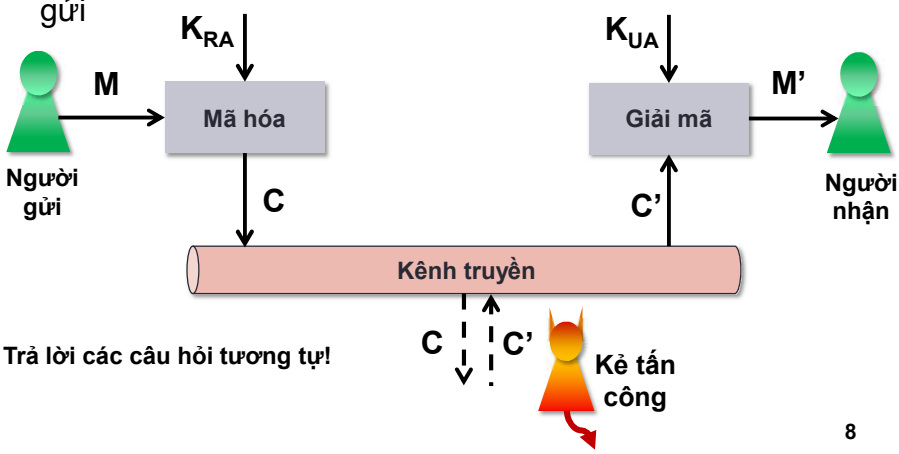


7

7

Xác thực bằng mật mã khóa công khai

- Chúng ta đã biết sơ đồ bí mật: mã hóa bằng khóa công khai của người nhận
- Sơ đồ xác thực: mã hóa bằng khóa cá nhân của người gửi



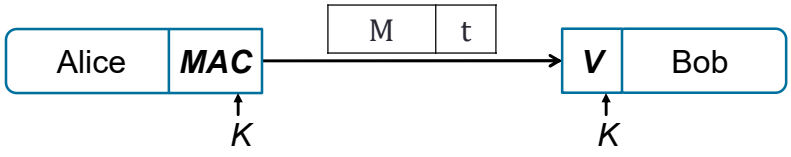
8

8

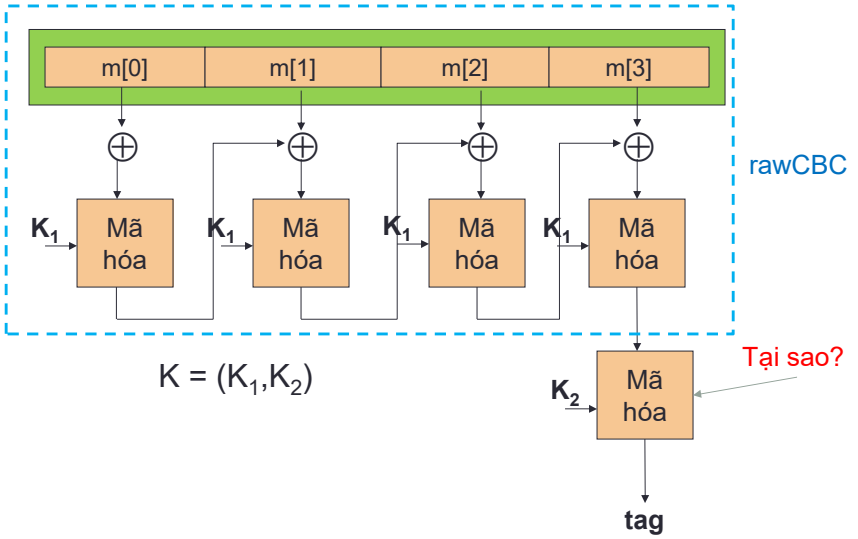
2. MÃ XÁC THỰC THÔNG ĐIỆP (MAC)

Message Authentication Code

- Xây dựng trên cơ sở hệ mật mã khóa đối xứng:
  - Hai bên đã trao đổi một cách an toàn khóa mật K
  - Sử dụng các thuật toán mã hóa khối ở chế độ CBC-MAC
- Bên gửi:
  - Tính toán t = MAC(K, M) : kích thước cố định, không phụ thuộc kích thước của M
  - Truyền (M||t)
- Bên nhận: xác minh Verify(K, M, t)
  - Tính t' = MAC(K, M)
  - So sánh: nếu t' = t thì Verify(K, M, t) = 1, ngược lại Verify(K, M, t) = 0



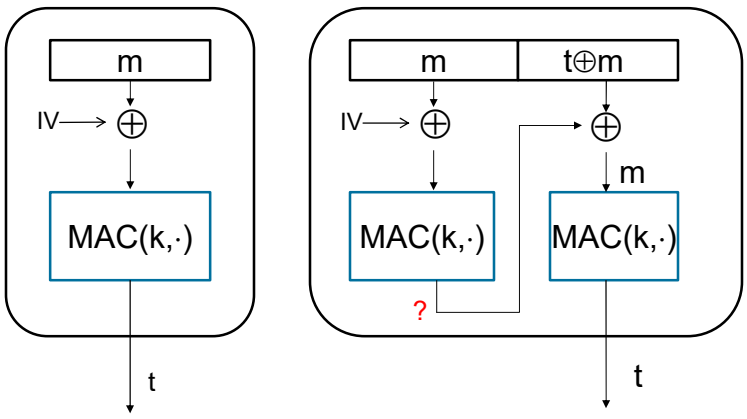
CBC-MAC



11

11

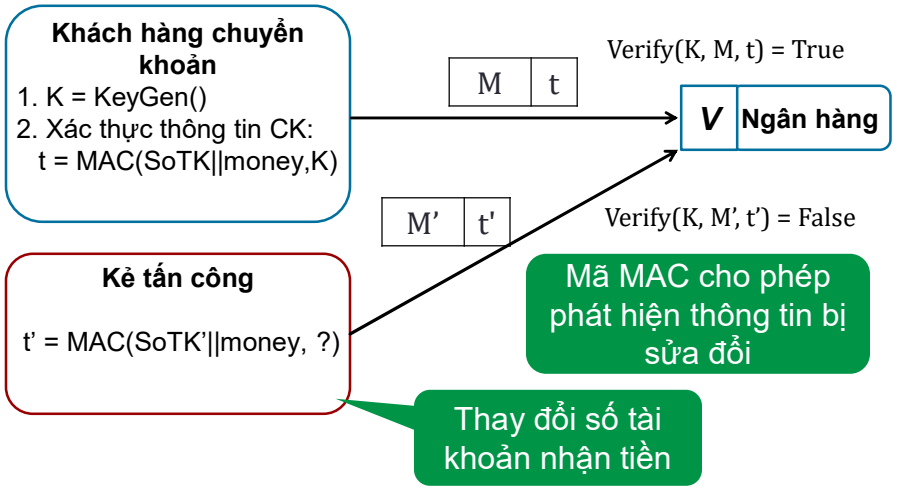
rawCBC-Tấn công chọn trước bản rõ



Vấn đề: MAC(k, m|| t⊕m) = MAC(k, MAC(k,m)⊕(t⊕m)) =
MAC(k, t⊕(t⊕m)) = MAC(k,m) = t

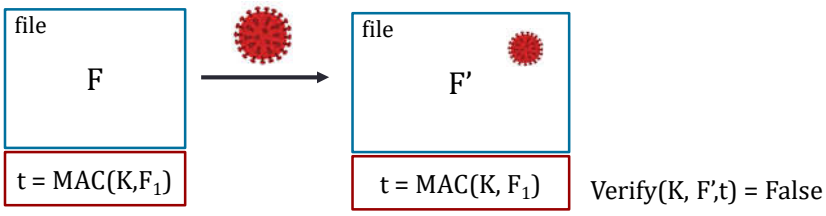
12

MAC – Ví dụ 1



MAC – Ví dụ 2: Phần mềm TripeWire

- Khi cài đặt, tính giá trị MAC của các file cần bảo vệ



- Khi máy tính khởi động, các file được kiểm tra mã MAC

→ Cho phép phát hiện các file bị sửa đổi (ví dụ do nhiễm virus)

Độ an toàn của MAC

- Giả sử M_1 và M_2 là hai bản tin có mã MAC giống nhau:

$MAC(M_1, K) = MAC(M_2, K)$

→$MAC(M_1||W, K) = MAC(M_2||W, K)$ với W bất kỳ

- Kịch bản tấn công:

1. Kẻ tấn công tính toán $t_x = MAC(M_x, K)$ với x = 1, …, N
2. Tìm cặp bản tin (M_i, M_j) có $t_i = t_j$. Nếu không tìm thấy thực hiện lại bước 1
3. Chọn bản tin W và tính $t = MAC(M_i ||W, K)$
4. Thay M_i || W bằng M_j || W có lợi cho kẻ tấn công

15

15

Ví dụ tấn công vào tính đụng độ

(1) Kẻ tấn công(Mr. Tung) chọn được 2 bản tin có mã MAC giống nhau:

M1: 'I will pay 1'

M2: 'I will pay 2'

Chọn W = '000$ to Mr.Tung'

M1 || W = 'I will pay 1000$ to Mr.Tung'

M2 || W = 'I will pay 2000$ to Mr.Tung'

(2) Đánh lừa người dùng gửi bản tin 'I will pay 1000$ to Mr.Tung' || MAC(K, 'I will pay 1000$ to Mr.Tung') cho ngân hàng

(3) Thay thế bằng 'I will pay 2000$ to Mr.Tung' || MAC(K, 'I will pay 1000$ to Mr.Tung') → Ngân hàng chấp nhận

16

16

Độ an toàn của MAC (tiếp)

- Kích thước bản tin: L_M
- Kích thước mã MAC: L_t
- Nếu $L_M \leq L_t$ và L_M không đổi: Mã MAC an toàn
- Nếu L_M thay đổi: $|M| > |t|$ nên tồn tại $M_2 \neq M_1$ sao cho $MAC(M_2) = MAC(M_1)$

→MAC bị giảm tính an toàn

- Yêu cầu với giải thuật tạo MAC có đầu ra n bit
  - Nếu biết trước (M_1,t_1), xác suất tìm M_2 sao cho $MAC(M_2) = t_1$ không lớn hơn đáng kể 2^{-n}
  - Xác suất tìm được cặp bản tin M_1 và M_2 sao cho $t_1 = t_2$ không lớn hơn đáng kể 2^{-n}
  - Giả sử M' là một dạng biến đổi của M, xác suất để t' = t không lớn hơn đáng kể 2^{-n}

17

17

Tấn công phát lại (Replay attack)

- Kẻ tấn công phát lại bản tin M đã được chứng thực trong phiên truyền thông trước đó
- Thiết kế MAC không chống được tấn công phát lại

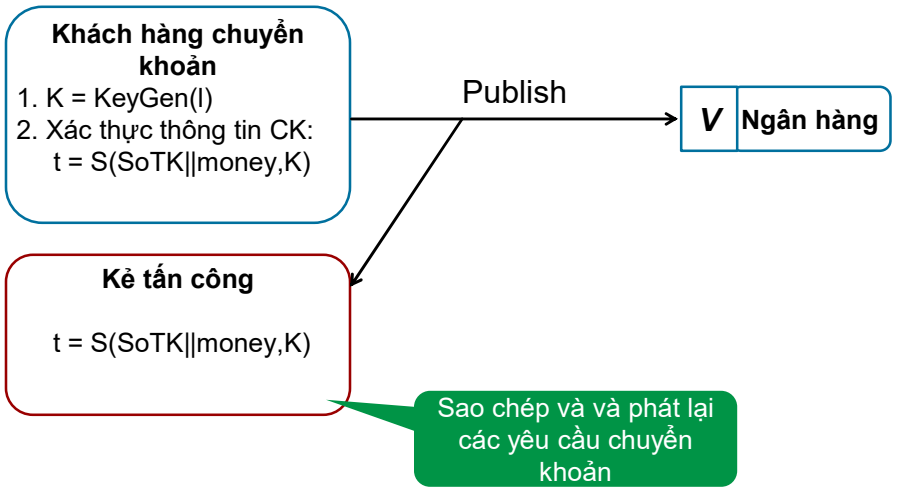
→ cần thêm các yếu tố chống tấn công phát lại trong các giao thức truyền thông sử dụng MAC

- Một số kỹ thuật chống tấn công phát lại:
  - Giá trị ngẫu nhiên dùng 1 lần: $MAC[(MAC(M, K_1) \| Random), K_2]$
  - Tem thời gian: $MAC[(MAC(M, K_1) \| Timestamp), K_2]$

18

18

Tấn công phát lại



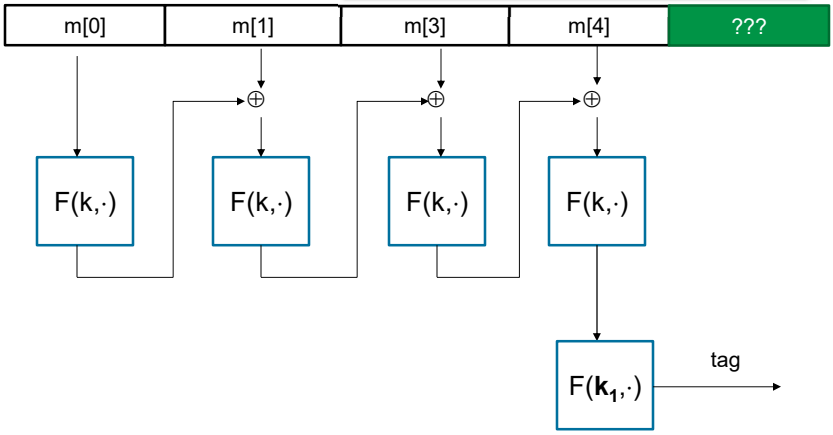
19

19

Padding cho MAC

CBC-MAC

Xử lý thế nào nếu kích thước thông điệp không chia hết cho kích thước một khối?

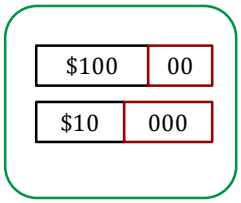


20

Padding cho MAC

- Ý tưởng 1: Thêm vào các bit 0

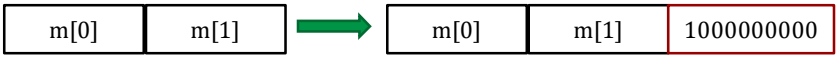
- Không an toàn. Ví dụ:



21

Padding cho MAC

- Yêu cầu: $M_i \neq M_j$ thì $pad(M_i) \neq pad(M_j)$
- Chuẩn ISO:
  - Sử dụng chuỗi padding bắt đầu bởi bit 1
  - Nếu kích thước thông điệp là bội số kích thước của khối, luôn thêm 1 khối padding



22

Mật mã có xác thực

- Một hệ mật mã có xác thực (E, D) là một hệ mật mã mà

Hàm mã hóa E: K x M x N → C

Hàm giải mã D: K x C x N → $M \cup \{\perp\}$

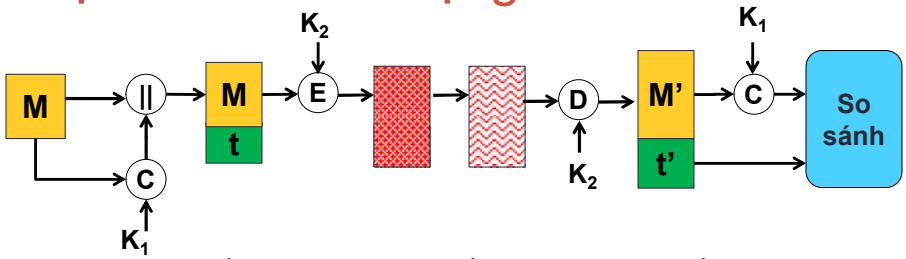
- Trong đó N là một dấu hiệu sử dụng để xác thực

Từ chối giải mã các bản mã không hợp lệ

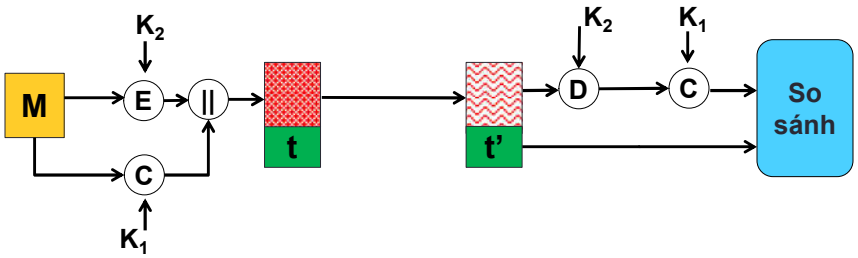
- Yêu cầu:
  - Chống tấn công chọn trước bản rõ, và
  - Kiểm tra được tính toàn vẹn của bản mật: xác suất kẻ tấn công tạo ra được một bản mật có thể giải mã là rất nhỏ
- Giải phát: Kết hợp mật mã và mã MAC

23

Một số sơ đồ sử dụng mã MAC



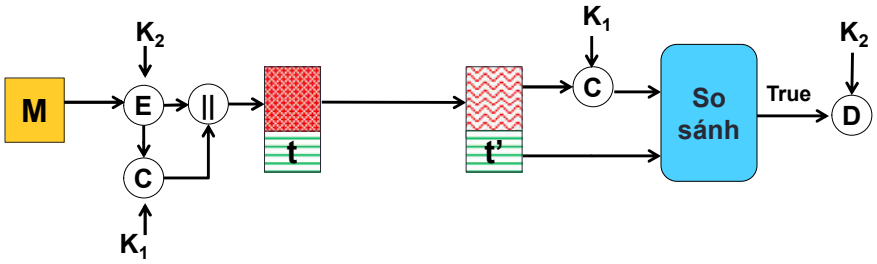
a) Xác thực bằng MAC, bảo mật bằng mật mã khóa đối xứng(SSL)



b) Xác thực bằng MAC, bảo mật bằng mật mã khóa đối xứng (SSH)

24

Một số sơ đồ sử dụng mã MAC(tiếp)



c) Xác thực bằng MAC, bảo mật bằng mật mã khóa đối xứng(IPSec)

- Một số chuẩn:

GCM: Mã hóa ở chế độ CTR sau đó tính CW-MAC

CCM: Tính CBC-MAC sau đó mã hóa ở chế độ CTR (802.11i)

EAX: Mã hóa ở chế độ CTR sau đó tính CMAC

25

25

Nhận xét

Sơ đồ a

- Xác thực toàn vẹn bản rõ
- Không xác thực toàn vẹn bản mật(không phát hiện tấn công thay thế bản mật)
- Không có thông tin về bản rõ từ MAC
- Chỉ đảm bảo an toàn khi mã ở chế độ rand-CBC hoặc rand-CTR

Sơ đồ b

- Xác thực toàn vẹn bản rõ
- Không xác thực toàn vẹn bản không phát hiện bản mật bị thay thế)
- MAC chứa thông tin bản rõ
- Chỉ đảm bảo an toàn khi mã ở chế độ rand-CBC hoặc rand-CTR

Sơ đồ c

- Xác thực toàn vẹn bản rõ
- Xác thực toàn vẹn bản mật(có thể phát hiện bản mật bị thay thế)
- MAC không chứa thông tin bản rõ
- Luôn đảm bảo an toàn

26

3.HÀM BĂM

27

27

Khái niệm

- *Hàm băm H*: thực hiện phép biến đổi:
  - Đầu vào: bản tin có kích thước bất kỳ
  - Đầu ra: giá trị *digest h = H(M)*có kích thước *n* bit cố định (thường nhỏ hơn rất nhiều so với kích thước bản tin đầu vào)
- Chỉ thay đổi 1 bit đầu vào, làm thay đổi hoàn toàn giá trị đầu ra
- Ví dụ:
  - Đầu vào: “The quick brown fox jumps over the lazy dog”
  - Mã băm: 2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
  - Đầu vào: “The quick brown fox jumps over the lazy cog”
  - Đầu ra: de9f2c7fd25e1b3afad3e85a0bd17d9b100db4b3

28

28

Một hàm băm đơn giản

- Chia thông điệp thành các khối có kích thước n-bit
  - Padding nếu cần
- Thực hiện XOR tất cả các khối → mã băm có kích thước n bit
- Tất nhiên, hàm băm này không đủ an toàn để sử dụng trong bài toán xác thực thông điệp

$$m = \begin{bmatrix} m_1 \\ m_2 \\ \dots \\ m_l \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{l1} & m_{l2} & \cdots & m_{ln} \end{bmatrix}$$

$$\oplus \quad \oplus \quad \oplus \quad \oplus$$

$$\Downarrow \quad \Downarrow \quad \Downarrow \quad \Downarrow$$

$$[c_1 \quad c_2 \quad \dots \quad c_n] = \text{H(m)}$$

29

29

Yêu cầu đối với hàm băm

1. Có thể áp dụng với thông điệp M với độ dài bất kỳ
2. Tạo ra giá trị băm h có độ dài cố định
3. H(M) dễ dàng tính được với bất kỳ M nào
4. Từ h rất khó tìm được M sao cho h = H(M): tính một chiều
5. Biết trước M_1 rất khó tìm được M_2 sao cho $H(M_1) = H(M_2)$
6. Rất khó tìm được cặp (M_1,M_2) sao cho $H(M_1)=H(M_2)$

30

30

Một số hàm băm phổ biến

- MD5
  - Kích thước digest: 128 bit
  - Công bố thuật toán tấn công đụng độ (collision attack) vào 1995
  - Năm 2005 tấn công thành công
- SHA-1
  - Kích thước digest: 160 bit
  - Công bố tấn công thành công vào năm 2015
  - Hết hạn vào năm 2030
- SHA-2: 224/256/384/512 bit
- SHA-3: 224/256/384/512 bit

31

31

MD5

- Bước 1: Padding dữ liệu sao cho bản tin đầu vào có độ dài L sao cho L mod 512 = 448
- Bước 2: Biểu diễn độ dài của dữ liệu ban đầu dưới dạng 64 bit. Thêm giá trị độ dài này vào khối dữ liệu.
  - Coi khối dữ liệu là một chuỗi các khối 512 bit: $Y_0, Y_1, \ldots, Y_{K-1}$
  - Hoặc là một chuỗi các khối 32 bit : $M_0, M_1, \ldots, M_N$
- Bước 3: Khởi tạo các giá trị hằng số A, B, C, D
  - A = 0x67 45 23 01
  - B = 0xEF CD AB 89
  - C = 0x98 BA DC FE
  - D = 0x10 32 54 76

32

MD5

- Bước 4: Thực hiện vòng lặp xử lý các khối 512 bit
  - Xử lý khối dữ liệu 512 bit thứ q: thực hiện 4 vòng lặp. Mỗi vòng lặp áp dụng hàm nén với T[1..64] là mảng hằng số xác định trước
  - Cộng modulo 2^{32} mỗi khối với giá trị CV_q để có CV_{q+1}
- Bước 5: Kết quả xử lý khối 512 bit cuối cùng là giá trị băm của thông điệp

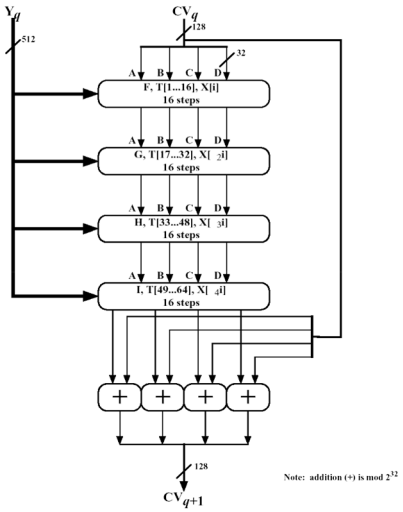


Figure 9.2 MD5 Processing of a Single 512-bit Block (MD5 Compression Function)

Hàm nén trong MD5

- Đầu vào:
  - CV: Khối 128 bit
  - M_i: khối dữ liệu 32-bit
  - T_i: Hằng số
- ⊞ Cộng modulo 2^{32}
- <<<s: dịch trái s bit
- ∧: AND, v: OR, ¬: NOT

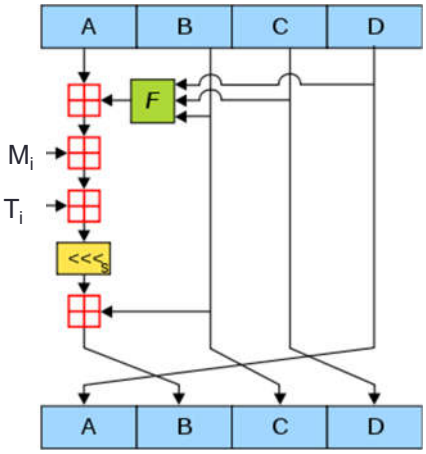
F1 = (B∧C)∨(¬B ∧ D)

F2 = (B ∧ D) ∨(C ∧ ¬D)

F3 = B ⊕ C ⊕ D

F4=C ⊕ (B ∨ ¬D)

- Thực hiện vòng lặp 16 bước



SHA-1

- Bước 1: Padding dữ liệu sao cho bản tin đầu vào có độ dài L sao cho L mod 512 = 448
- Bước 2: Biểu diễn độ dài của dữ liệu ban đầu dưới dạng 64 bit. Thêm giá trị độ dài này vào khối dữ liệu.
  - Coi khối dữ liệu là một chuỗi các khối 512 bit: Y_0, Y_1, …, Y_{K-1}
  - Hoặc là một chuỗi các khối 32 bit : M_0, M_1, …,M_N
- Bước 3: Khởi tạo các giá trị hằng số A, B, C, D, E
  - A = 0x67 45 23 01
  - B = 0xEF CD AB 89
  - C = 0x98 BA DC FE
  - D = 0x10 32 54 76
  - E = 0xC3 D2 E1 F0

35

SHA-1

- Bước 4: Thực hiện vòng lặp xử lý các khối 512 bit
  - Xử lý khối dữ liệu 512 bit thứ q: thực hiện 4 vòng lặp. Mỗi vòng lặp áp dụng hàm nén với K là hằng số xác định trước
  - Cộng modulo 2^{32} mỗi khối với giá trị CV_q để có CV_{q+1}
- Bước 5: Kết quả xử lý khối 512 bit cuối cùng là giá trị băm của thông điệp

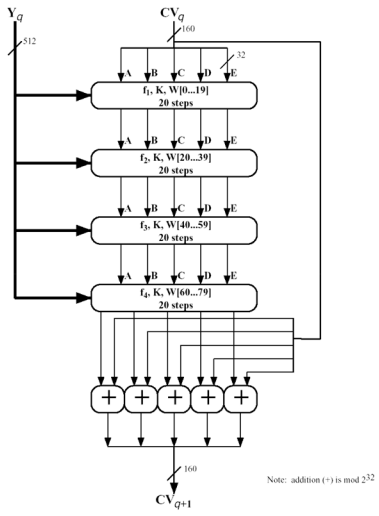


Figure 9.5 SHA-1 Processing of a Single 512-bit Block (SHA-1 Compression Function)

36

Hàm nén trong SHA-1

- Đầu vào:
  - CV: Khối 160 bit
  - W_t: Khối dữ liệu mở rộng 32 bit
  - K_t: Hằng số
- ⊞ Cộng modulo 2^{32}
- <<<5(30): dịch trái 5(30) bit
- ∧: AND, v: OR, ¬: NOT

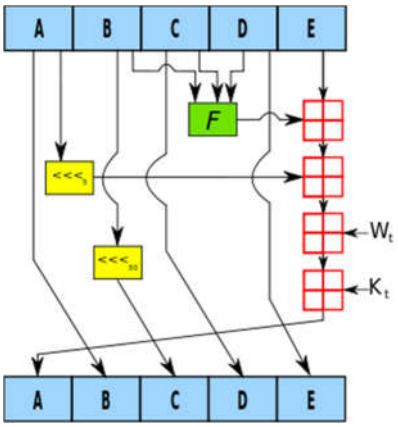
F1 = (B∧C)∨(¬B ∧ D)

F2 = B ⊕ C ⊕ D

F3 = (B ∧ C) ∨ (B ∧ D) ∨ (C ∧ D)

F4 = B ⊕ C ⊕ D

- Thực hiện vòng lặp 20 bước



37

Tấn công vét cạn

- h = H(M): kích thước n bit
  - n << L_M → luôn tồn tại $M_2 \neq M_1$ sao cho $H(M_2) = H(M_1)$

    → kẻ tấn công muốn được bản tin M_2 có lợi cho anh ta để thay thế M_1 đã được xác thực
- Phương pháp: vét cạn → số bản tin cần tính tối thiểu là bao nhiêu sẽ chắc chắn thành công?

38

Tấn công ngày sinh

- Bài toán: Khi chọn n người bất kỳ, xác suất để có tối thiểu 2 người có trùng ngày sinh là bao nhiêu?
- Số cách chọn ra n người bất kỳ: 365^n
- Số cách chọn ra n người không có cặp nào trùng ngày sinh: 365 x 364 x … x (365-(n-1)) = C^n_{365}
- Xác suất để chọn ra n người không có cặp nào trùng ngày sinh

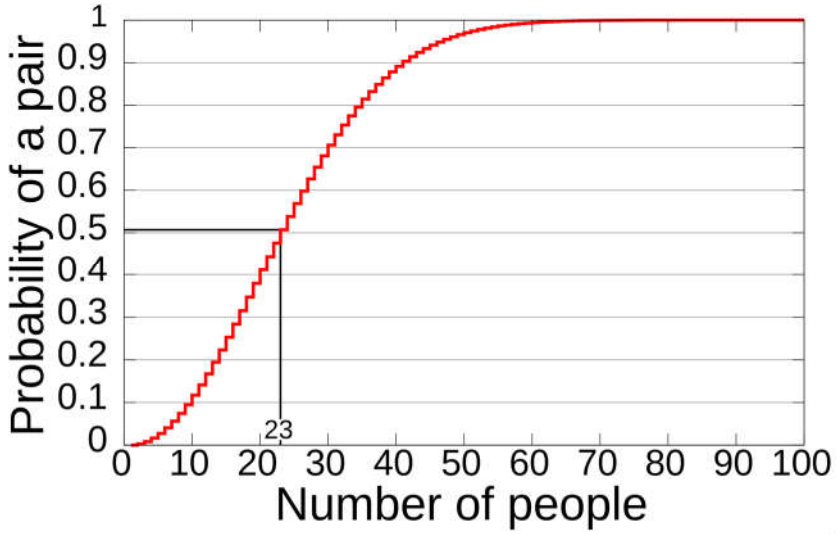
$$Q = \frac{365 \times 364 \times \cdots \times (365 - (n-1))}{365^n}$$

- Xác suất cần tính: P = 1 – Q
- n = ? để P > 0.5 (cứ 2 lần chọn thì có 1 lần thỏa mãn)

39

39

Xác suất trong tấn công ngày sinh



40

40

Tấn công ngày sinh (Birthday paradox attack)

- h = H(M): kích thước n bit
  - n << L_M → luôn tồn tại $M_2 \neq M_1$ sao cho $H(M_2) = H(M_1)$

  → kẻ tấn công muốn được bản tin M_2 có lợi cho anh ta để thay thế M_1 đã được xác thực
- Phương pháp: vét cạn → số bản tin cần tính tối thiểu là bao nhiêu sẽ chắc chắn thành công?
- Cải tiến bằng tấn công ngày sinh: cho phép giảm số bản tin xuống chỉ còn $2^{n/2}$ với xác suất thành công là ≥ 0.5:
  - Công thức gần đúng tính xác suất thành công:

$$P(N,k) > 1 - e^{\frac{-k(k-1)}{2N}}$$

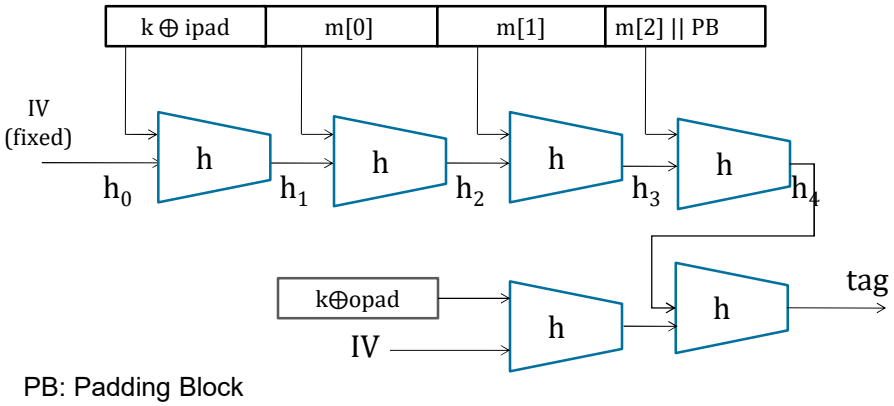
N: số giá trị *h*
k: số bản tin cần kiểm tra

41

41

HMAC

- Hashed MAC: kết hợp MAC và hàm băm để tăng cường an toàn cho hàm băm

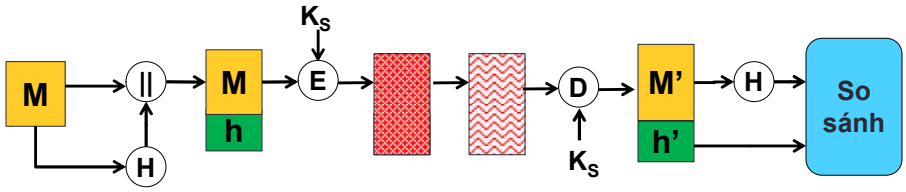


PB: Padding Block

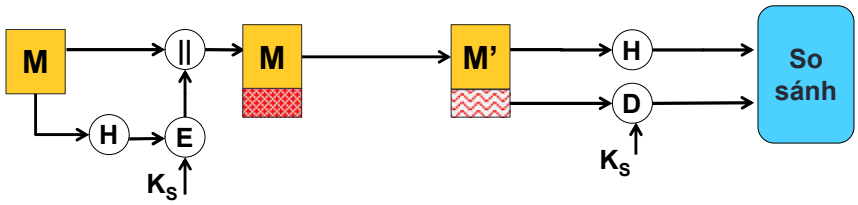
42

42

Một số sơ đồ sử dụng hàm băm để xác thực



a) Xác thực thông điệp và bảo mật bằng mật mã khóa đối xứng

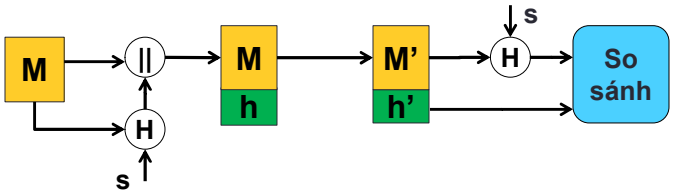


b) Xác thực thông điệp, mã băm được bảo vệ bằng mật mã khóa đối xứng

43

43

Một số sơ đồ sử dụng hàm băm để xác thực



c) Xác thực thông điệp sử dụng HMAC

Bài tập:

1. Kiểm tra những sơ đồ trên đáp ứng được yêu cầu nào về xác thực
2. Kết hợp sử dụng hệ mật mã khóa công khai để tạo ra một số sơ đồ mới

44

44

Tổng kết

- MAC: Message Authentication Code
  - Sử dụng các thuật toán mật mã khóa đối xứng ở chế độ CBC-MAC
  - Hai bên cần chia sẻ trước khóa bí mật K
  - Hàm tạo mã MAC: CBC-MAC(K, Data)
- Hàm băm:
  - Không sử dụng khóa
  - Phải kết hợp với MAC → HMAC
  - Sử dụng trong các sơ đồ chữ ký số
  - Các hàm băm an toàn: ~~MD5~~, SHA-1/2/3
- Tấn công ngày sinh
  - Giảm số bản tin cần kiểm tra để tạo ra các bản tin đụng độ (có mã MAC hoặc mã băm giống nhau)