



Bài 3.
Mật mã học
(Phần 2 – Xác thực thông điệp)


ONE LOVE. ONE FUTURE.

1

1

Nội dung

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

 **ĐẠI HỌC BÁCH KHOA HÀ NỘI**
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

2

2

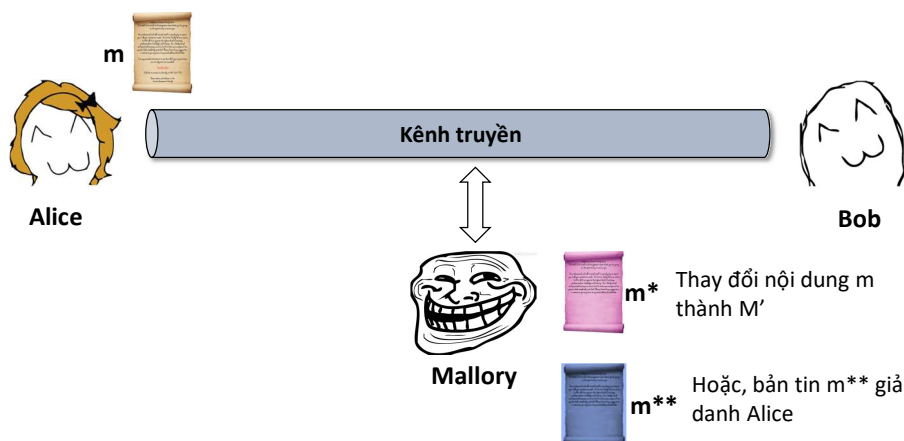
1. Đặt vấn đề

ONE LOVE. ONE FUTURE.

3

3

Đặt vấn đề



4

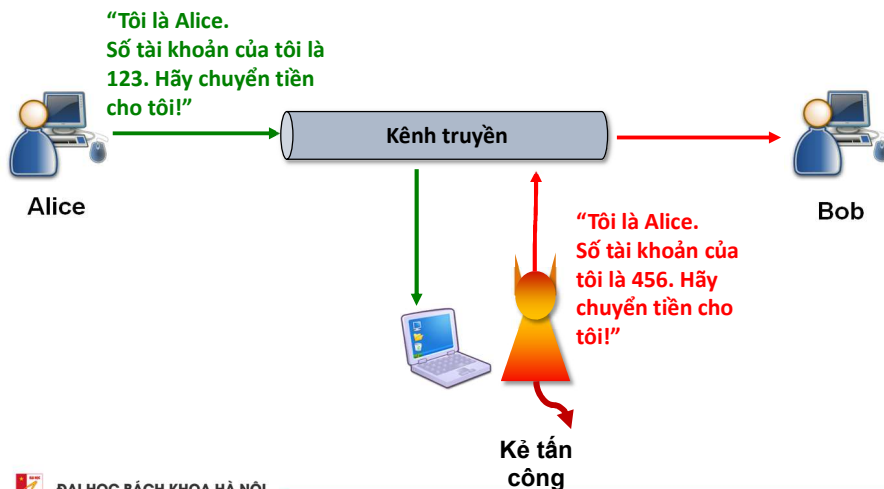
4

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)

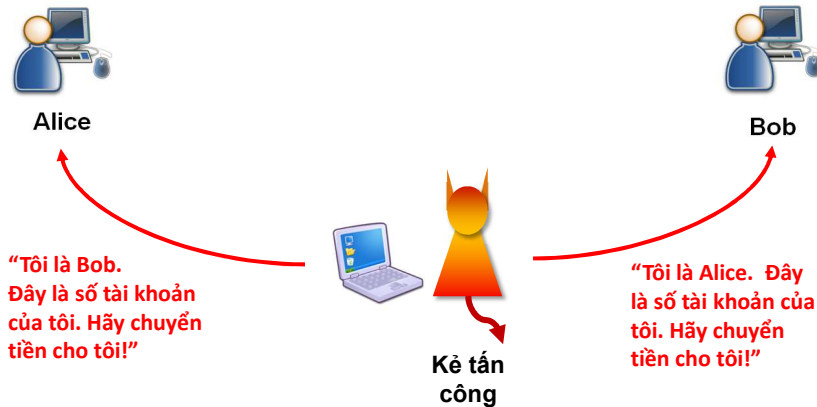
Tấn công thay thế

- Chặn thông điệp, thay đổi nội dung và chuyển tiếp cho bên kia



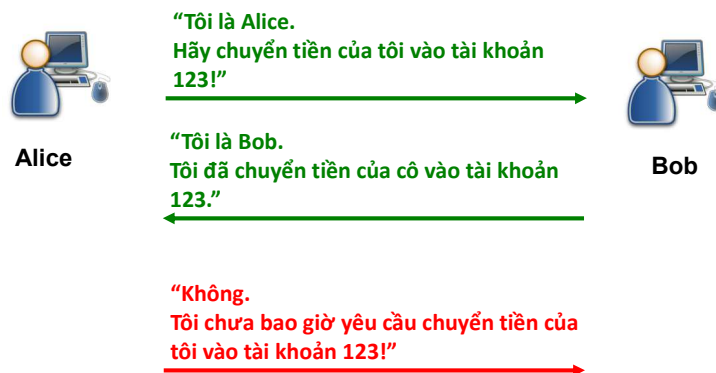
Tấn công giả danh

- Kẻ tấn công mạo danh một bên và chuyển các thông điệp cho bên kia.



Tấn công phủ nhận gửi

- Bên gửi phủ nhận việc đã gửi đi một thông tin



2. Mã xác thực thông điệp (MAC)

ONE LOVE. ONE FUTURE.

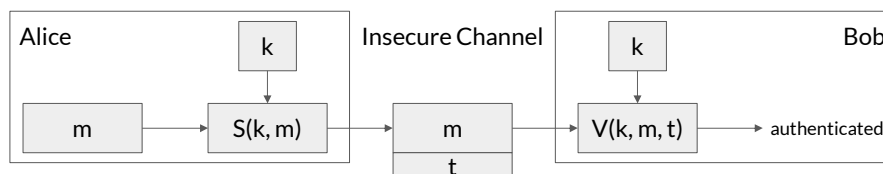
9

9

t ora 1 chu i mã MAC có th xác th c thông i p (ch nh ng ng b t khóa k m s t o c)

Message Authentication Code

- Hai bên đã trao đổi một cách an toàn khóa mật k
- Hàm $MAC = (S, V)$ là một cặp thuật toán
- Sinh mã: $t = S(k, m)$
 - Đầu ra: kích thước cố định, không phụ thuộc kích thước của bản tin đầu vào m
- Xác minh: $V(k, m, t)$
 - Tính $t' = S(k, m)$
 - Nếu $t' = t$ thì $V = \text{true}$, ngược lại $V = \text{false}$



10

k t n công m u n t n công p h i t o r a c l c p m * t * t h a m ă n
 -> khó vì ko bt khóa k

MAC – Ví dụ 1

Khách hàng chuyển khoản

1. Chia sẻ khóa k
2. $m = \text{account} || \text{money}$
3. $t = S(k, m)$

m t

$$V(k, m, t) = \text{True}$$

V Ngân hàng

m* t*

$$V(k, m^*, t^*) = \text{False}$$

Kẻ tấn công

1. Không biết k
2. Tạo $m^* = \text{account}^* || \text{money}$
3. Tạo t^* trong khi không biết k

Mã MAC cho phép phát hiện thông tin bị sửa đổi

Thay đổi số tài khoản nhận tiền

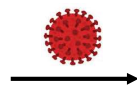
11

MAC d a trên nguyên lý khóa i x n g
 MAC còn an toàn thì khóa k còn c g i b i m t
 m t an toàn khi khóa k b l

MAC – Ví dụ 2: Phần mềm Tripwire

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

file
 F
 $t = S(k, F)$



file
 F*
 $t = S(k, F)$

$$V(k, F^*, t) = \text{False}$$

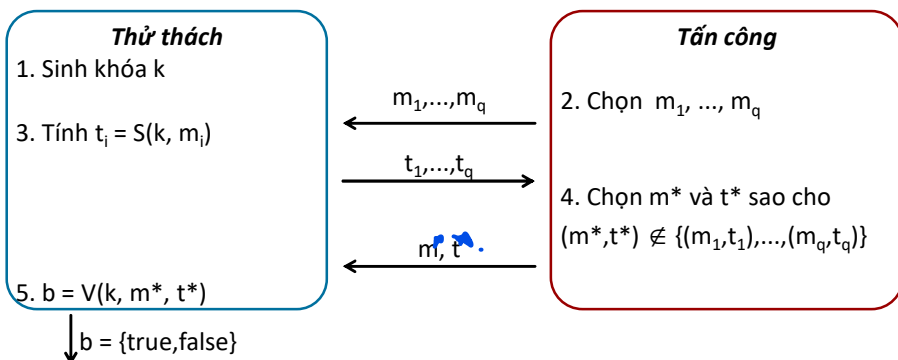
- Khi máy tính khởi động, các file được kiểm tra mã MAC
 → Cho phép phát hiện các file bị sửa đổi (ví dụ do nhiễm virus)

12

Đặc tính của MAC

- **Tính xác định:** nếu $m_1 = m_2 \rightarrow S(k, m_1) = S(k, m_2)$
 - **Tính đúng đắn:** $V(k, m, S(k, m)) = \text{True}$
 - **Tính hiệu quả:** tính toán được với mọi m trong thời gian chấp nhận được (th i giant ng i tùy thu ch th ng)
 - **Tính an toàn:** nếu đối phương không biết k
 - Không thể tính được $t = S(k, m)$
 - Với t cho trước, không thể tìm được m sao cho $S(k, m) = t \rightarrow$ hàm c as p
 - Không thể tìm được m và m^* sao cho $S(k, m) = S(k, m^*)$
- t p mã MAC b g i h n -> luôn t n t i các b n tin khác nhau có MAC g ng nhau

An toàn của MAC



- MAC là an toàn nếu với mọi thuật toán tấn công hiệu quả thì xác suất $P(b = \text{true}) \leq \epsilon$
- kẻ tấn công không thể tạo giá trị t hợp lệ nếu không có khóa k

An toàn của MAC

- MAC còn an toàn không nếu tồn tại thuật toán hiệu quả cho một trong các tình huống sau:
 - (1) Tìm được m^* sao cho $S(?, m^*) = t$ với t chọn trước
 - (2) Tìm được m^* sao cho $S(?, m^*) = S(?, m)$ với m chọn trước
 - (3) Tìm được m và m^* sao cho $S(?, m^*) = S(?, m)$ ch abt mã MAC
- Hoặc giá trị t có kích thước 10 bit

- 1, ko an toàn -> vì có m^* và t th a mã -> b luôn = True
- 2, ko an toàn -> t n t i 2b n tin khác nhau nh ng có cùng mã MAC
-> ng -> ko an toàn (ch n m^* và $S(?, m)$) -> luôn True
- 3 ko an toàn -> t ng t 2
- 4, ko an toàn -> vết c n mã MAC

Một ví dụ khác

- Một hàm MAC được tạo ra như sau:

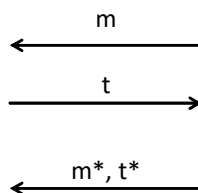
$$S'(k, m_1 || m_2) = S(k, m_1) || S(k, m_2)$$
 Trong đó S là hàm tạo mã MAC an toàn. S' có phải là hàm MAC an toàn hay không?

Thử thách

1. Sinh khóa k
3. Tính $t = S'(k, m) = S(k, m_1) || S(k, m_2) = t_1 || t_2$
5. $b = V(k, m^*, t^*) = \text{true}$

Tấn công

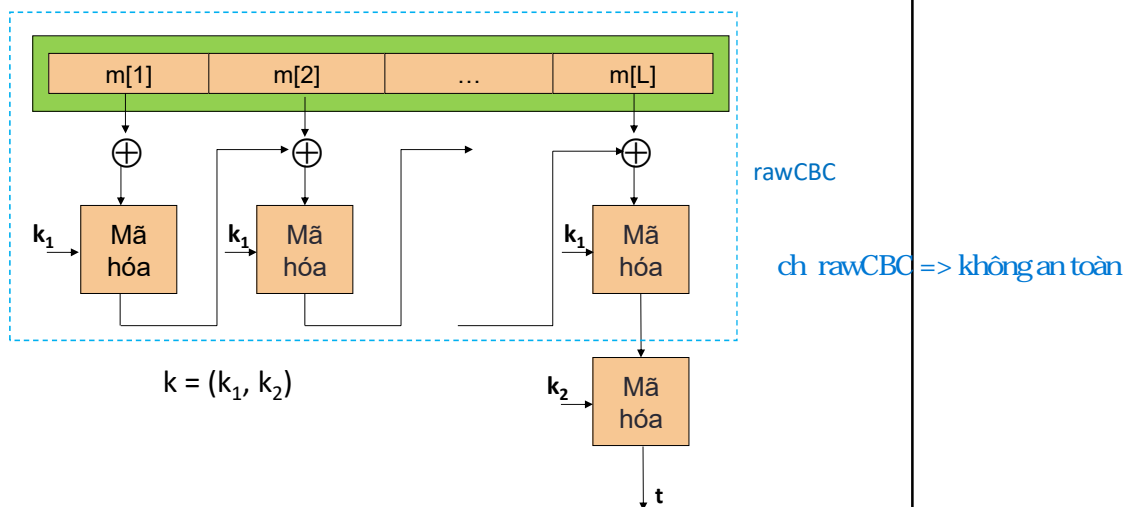
2. Chọn $m = m_1 || m_2$
4. Chọn $m^* = m_2 || m_1$
 $t^* = t_2 || t_1$



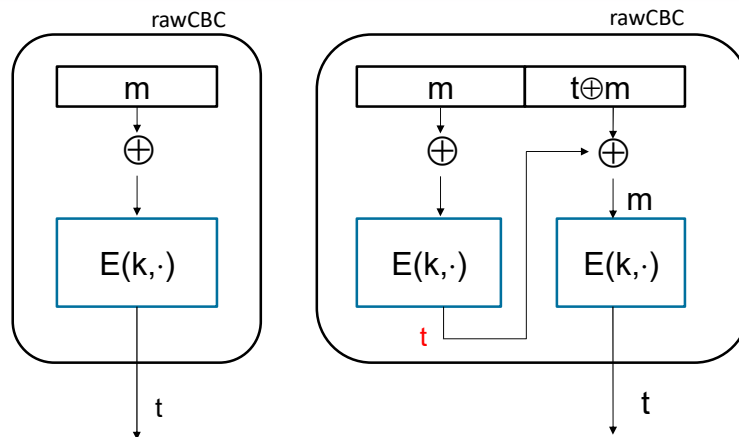
MAC cung cấp dịch vụ nào?

- **Toàn vẹn:** đối phương không thể thay đổi bản tin mà không bị phát hiện
- **Xác thực danh tính:** Nếu Alice nhận được một bản tin có mã MAC hợp lệ với khóa bí mật đã chia sẻ với Bob thì Alice có thể xác định được người tạo bản tin là Bob
- **Không có khả năng chống từ chối:** Bob không có cách nào thuyết phục được với bên thứ ba rằng Alice là người tạo bản tin
- **Không có khả năng giữ bí mật:**
 - Nếu khóa tạo MAC không thay đổi, kẻ tấn công có thêm thông tin về bản tin gốc

Xây dựng MAC: CBC-MAC



rawCBC-Tấn công chọn trước bản rõ



$m \oplus m + t \text{ xor } m \Rightarrow 2b \text{ n}$
 tin ng \Rightarrow hàm MAC
 không an toàn

Vấn đề: $S(k, m \parallel t \oplus m) = S(k, E(k, m) \oplus (t \oplus m)) =$
 $S(k, t \oplus (t \oplus m)) =$
 $S(k, m) = t$

ko an toàn tr t n công CPA,
 k t n công k c n b t khóa k
 v n tìm ra cm^*, t^* hợp lý

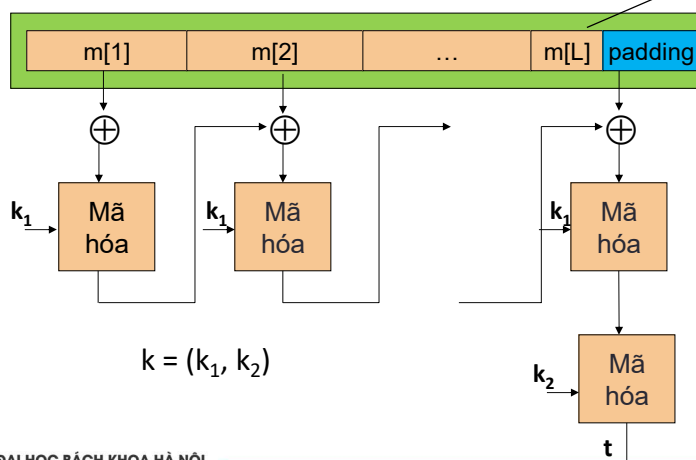
19

hàm ngẫu nhiên: v i vào xác nh, urang u nhiên
 hàm xác nh: u vào xác nh, uraxác nh
 \rightarrow hàm MAC là hàm xác nh

Xây dựng MAC: CBC-MAC

không s d ng vector IV \Rightarrow 2 vector IV ng u nhiên
 \Rightarrow 2b n tin g ng nhau có mã MAC khác nhau
 \Rightarrow ko xác nh

Kích thước thông điệp không chia
 hết cho kích thước một khối?



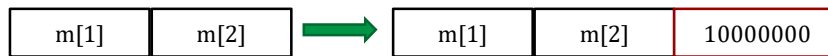
20

Padding cho CBC-MAC

- Yêu cầu: $m_i \neq m_j$ thì $\text{pad}(m_i) \neq \text{pad}(m_j)$
- Chuẩn ISO/IEC 9797-1:
 - 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



ko h t v n thêm padding

Độ an toàn của CBC-MAC

- Giả sử m và m^* là hai bản tin có mã MAC giống nhau: $S(k, m) = S(k, m^*)$
 - $S(k, m || W) = S(k, m^* || W)$ với W bất kỳ
- Kịch bản tấn công:
 1. Kẻ tấn công nhận được $t_x = S(k, m_x)$ với $x = 1, \dots, N$
 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 $t = S(k, m_i || W)$
 4. Thay $m_i || W$ bằng $m_j || W$ có lợi cho kẻ tấn công

Ví dụ tấn công vào tính độn độ

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

m: 'I will pay 1'

m*: 'I will pay 2'

Chọn W = '000\$ to Mr.Tung'

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

m* || W = 'I will pay 2000\$ to Mr.Tung'

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

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



An toàn của CBC-MAC

- Khóa được dùng nhiều lần → giảm độ an toàn
- Nếu gọi:
 - q: số bản tin được tính MAC cùng với khóa không đổi
 - X: Số lượng giá trị có thể của t
- Xác suất tấn công thành công $\leq 2^*q^2 / X$
- Để xác suất tấn công là không đáng kể ($\leq 2^{-80}$) thì sau bao nhiêu lần tính MAC phải đổi khóa?
- Ví dụ: Sử dụng AES-CBC-MAC:

$$2^*q^2 / 2^{128} \leq 2^{-80} \rightarrow q = ?$$



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 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 dùng 1 lần (nonce): $S(k, m || \text{nonce})$
- Tem thời gian: $S(k, m || \text{timestamp})$ thời gian thực

-> mỗi thời gian khác nhau -> mã MAC khác nhau -> chống tấn công phát lại

25

Tấn công phát lại

Khách hàng chuyển khoản

1. $K = \text{KeyGen}(I)$
2. Xác thực thông tin CK:
 $t = S(k, \text{SoTK} || \text{money})$

Publish

V Ngân hàng

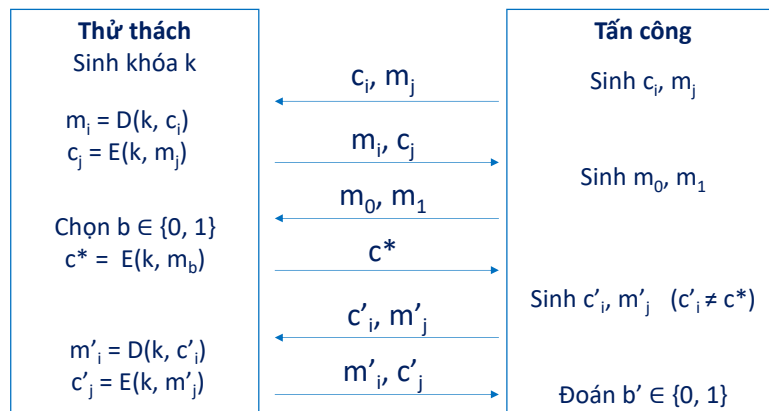
Kẻ tấn công

$t = S(k, \text{SoTK} || \text{money})$

Sao chép và phát lại các yêu cầu chuyển khoản

26

Tấn công CCA – Nhắc lại

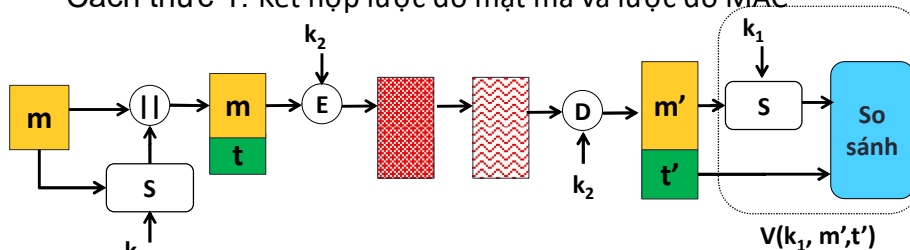


- Hệ mật chống lại được tấn công CCA (độ an toàn IND-CCA) nếu với mọi thuật toán tấn công hiệu quả thì $P(b' = b) \leq \frac{1}{2} + \epsilon$

27

Mật mã có xác thực(Authenticated Encryption)

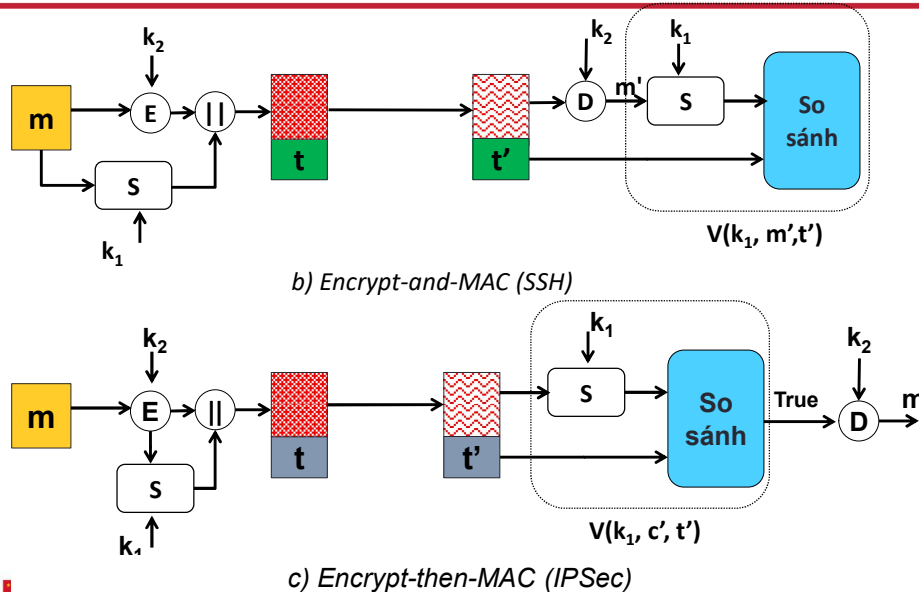
- Các sơ đồ mật mã đã xem xét không chống lại được tấn công CCA(chosen-cipher attack)
 - Cách thức chung: kẻ tấn công sửa bản mã c^* thành c'_i và yêu cầu giải mã. Dựa trên kết quả giải mã, hắn có thể thu được thêm thông tin.
- Mật mã có xác thực: sơ đồ mật mã đảm bảo đồng thời tính bí mật và toàn vẹn
- Cách thức 1: Kết hợp lược đồ mật mã và lược đồ MAC



a) MAC-then-encrypt (SSL)

28

Một số sơ đồ sử dụng mã MAC(tiếp)



29

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
- Không an toàn với tấn công kênh bên

Sơ đồ b

- Xác thực toàn vẹn bản rõ
- Không xác thực toàn vẹn bản mật(không phát hiện bản mật bị thay thế)
- MAC chứa thông tin bản rõ
- Có thể giảm sự an toàn mã mật

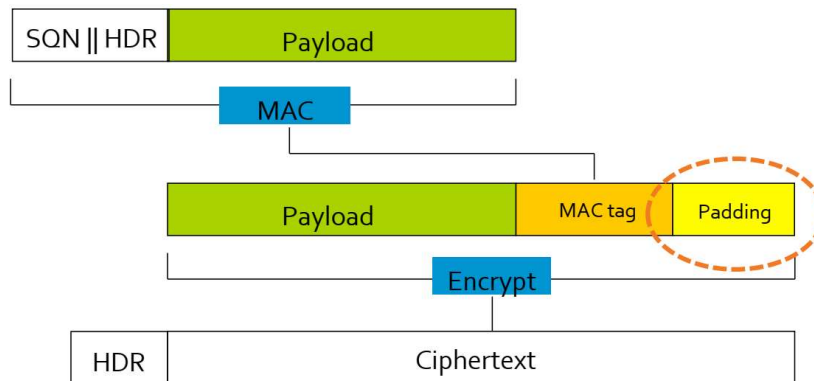
Sơ đồ c

- 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 CCA

30

Đọc thêm: Tấn công Lucky 13 (1)

- Giao thức SSL/TLS sử dụng lược đồ MAC-then-encrypt với nhiều bộ thuật toán, trong đó phổ biến là AES-CBC-HMAC



31

Đọc thêm: Tấn công Lucky 13 (2)

- Kết hợp 2 kỹ thuật tấn công kênh bên
 - Padding oracle attack: sử dụng kết quả kiểm tra phần đệm khi giải mã trong chế độ mã CBC
 - Timing attack: sử dụng sự khác biệt về thời gian tính toán HMAC với các bản tin có kích thước khác nhau:
 - Bản tin 55 byte: Cần 4 chu kỳ CPU
 - Bản tin 56 byte: Cần 5 chu kỳ CPU
- Chi tiết:

<https://www.ieee-security.org/TC/SP2013/papers/4977a526.pdf>

32

Tái sử dụng khóa

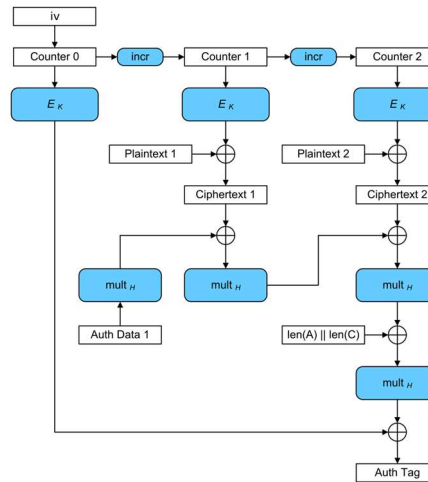
- Tái sử dụng khóa(Key reuse): sử dụng cùng khóa cho cả mã mật và MAC
- Hậu quả của tái sử dụng khóa:
 - Tăng rủi ro: lộ khóa khiến sơ đồ mất cả tính bí mật và toàn vẹn
 - Giảm an toàn của sơ đồ. Ví dụ sử dụng CBC-MAC và CBC-Encryption với khóa giống nhau
- **Dùng khóa khác nhau cho mục đích khác nhau:**
 - Khóa mã mật và khóa MAC khác nhau
 - Mở rộng: khóa khác nhau cho các chiều truyền tin khác nhau
 - Mở rộng: khóa khác nhau cho các loại dữ liệu khác nhau
 - An toàn hơn nhưng tốn nhiều chi phí hơn

Cách thức 2: AEAD Encryption

- Authenticated encryption with additional data (AEAD)
 - Là các phương pháp mật mã
- Hàm mã hóa $E: K \times M \rightarrow C \parallel \text{Auth Tag}$
- Hàm giải mã $D: K \times C \parallel \text{Auth Tag} \rightarrow M \cup \{\perp\}$
- Từ chối giải mã các bản mã không hợp lệ
- Trong đó t là mã MAC được tính toán trong quá trình mã hóa
 - 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
 - Ưu điểm:
 - Tốc độ tính toán nhanh hơn
 - Luôn chống được CCA khi được dùng đúng

AEAD-Ví dụ: GCM(Đọc thêm)

- Hạn chế: tái sử dụng IV gây mất an toàn



3. Hàm băm

Khái niệm

- **Hàm băm H**: thực hiện xử lý
 - Đầu vào: bản tin có kích thước bất kỳ
 - Đầu ra: giá trị mã băm(*digest*) $h = H(m)$ có kích thước n bit (thường nhỏ hơn rất nhiều so với kích thước bản tin đầu vào)
- **Đặc tính của hàm băm mật mã**:
 - **Tính đúng đắn**: đầu vào giống nhau thì đầu ra giống nhau
 - **Tính hiệu quả**: thực hiện trong thời gian chấp nhận được
 - **Tính một chiều**: không thể xác định bản tin từ mã băm
 - **Tính chống đụng độ**: xác suất tìm ra 2 bản tin giống nhau là 0
 - **Tính ngẫu nhiên**: không thể đoán được sự thay đổi của đầu ra khi thay đổi đầu vào
 - Chỉ thay đổi 1 bit đầu vào, làm thay đổi hoàn toàn giá trị đầu ra
 - Xác suất xuất hiện của mọi giá trị mã băm là như nhau ($= 2^{-n}$)



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

37

37

tính ngẫu nhiên: luôn tìm ra 2 bản tin khác nhau có mã băm giống nhau

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

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

$$\begin{matrix} \oplus & \oplus & \oplus & \oplus \\ \downarrow & \downarrow & \downarrow & \downarrow \\ [c_1 & c_2 & \cdots & c_n] = H(m) \end{matrix}$$



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

38

38

Tính chống đụng độ (Collision Resistance)

- Đụng độ: Hai bản tin $m_1 \neq m_2$ có $H(m_1) = H(m_2)$
 - Trong hàm băm, luôn tồn tại các bản tin đụng độ
- Tính chống đụng độ: với mọi thuật toán tấn công hiệu quả, xác suất tìm ra hai bản tin bất kỳ mà chúng đụng độ là không đáng kể

Tấn công vào hàm băm

- Tấn công vào tính 1 chiều
 - Mục tiêu: Tìm ra bản tin gốc m từ mã băm h cho trước
 - Cách thức: tấn công vét cạn
 - (1) Chọn 2^n bản tin ngẫu nhiên
 - (2) Tính mã băm cho mỗi bản tin
 - (3) Nếu h không phải là mã băm của một trong số 2^n bản tin đã chọn, quay lại bước 1
 - Vì xác suất xuất hiện của mọi giá trị băm là như nhau nên theo kỳ vọng, tấn công thành công với 1 lần thử
 - Nhận xét: kích thước mã băm (n -bit) càng lớn thì hàm băm càng an toàn

Ví dụ

- Đối phương muốn tấn công vét cạn để tìm bản tin gốc của mã băm. Giả sử hắn có khả năng tính toán 1.000.000 mã băm mỗi giây. Kích thước mã băm là bao nhiêu để khi thực hiện tấn công trong 100 năm, xác suất thành công cao nhất là 2^{-80}
- Lời giải:
 - Kích thước mã băm: n bit \rightarrow số mã băm có thể là 2^n
 - Số mã băm kẻ tấn công tính được trong 100 năm là: X
 - Thỏa mãn: $X/2^n \leq 2^{-80} \rightarrow 132$ bit

Tấn công vào hàm băm (tiếp)

- Tấn công vào tính đựng độ (1)
 - Mục tiêu: Tìm ra bản tin m^* có mã băm trùng với m cho trước
 - Cách thức: tấn công vét cạn
 - (1) Tính $h = H(m)$
 - (2) Chọn 2^n bản tin ngẫu nhiên
 - (3) Tính mã băm cho mỗi bản tin
 - (4) Nếu $H(m_i) = h$ thì m_i là bản tin cần tìm
 - (5) Nếu không thấy m^* trong số 2^n bản tin đã chọn, quay lại bước 2
 - Vì xác suất xuất hiện của mọi giá trị băm là như nhau nên theo kỳ vọng, tấn công thành công với 1 lần thử
 - Nhận xét: kích thước mã băm (n -bit) càng lớn thì hàm băm càng an toàn

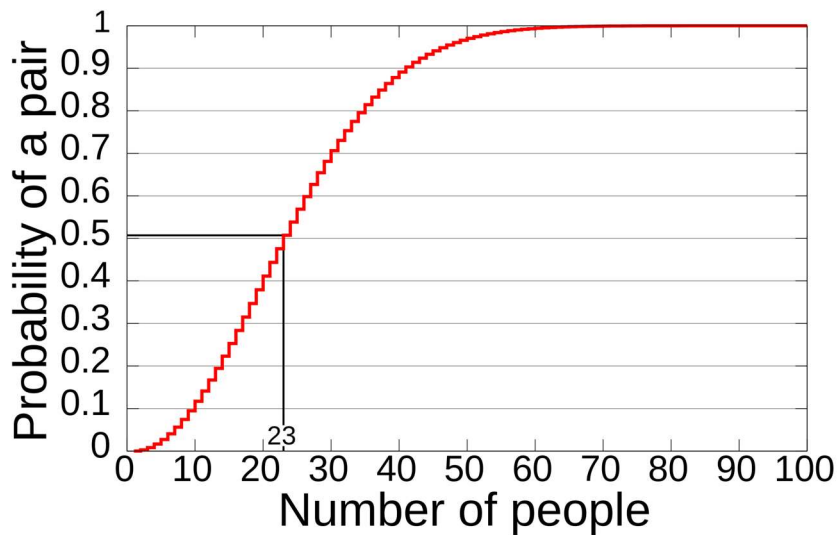
Tấn công vào hàm băm (tiếp)

- Tấn công vào tính đựng độ (2)
 - Mục tiêu: Tìm ra cặp bản tin (m, m^*) có mã băm trùng nhau
 - Cách thức: tấn công vét cạn
 - (1) Chọn $2^n + 1$ bản tin ngẫu nhiên
 - (2) Tính mã băm cho mỗi bản tin
 - (3) Luôn tìm thấy (m, m^*) thỏa mãn
 - Nhận xét: kích thước mã băm (n -bit) càng lớn thì hàm băm càng an toàn
 - Cách thức tấn công khác dựa trên nghịch lý ngày sinh hiệu quả hơn (được trình bày sau)

Nghịch lý ngày sinh (Birthday paradox)

- 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: $365 \times 364 \times \dots \times (365 - (n - 1)) = C^n_{365}$
- Xác suất để chọn ra n người không có cặp nào trùng ngày sinh
$$Q = \frac{365 \times 364 \times \dots \times (365 - (n - 1))}{365^n}$$
- Xác suất cần tính: $P = 1 - Q$
- $n = ?$ để $P > 0.5$ (mỗi 2 lần chọn thì có 1 lần thỏa mãn)

Nghịch lý ngày sinh

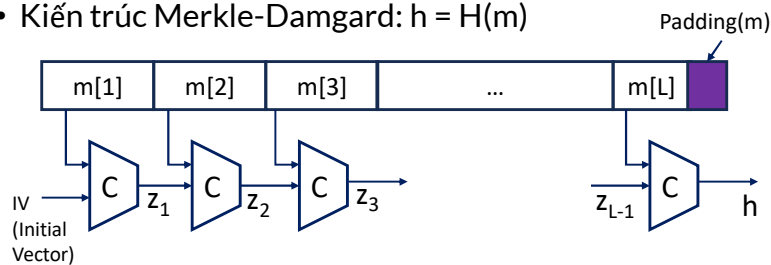


Tấn công dựa trên nghịch lý ngày sinh (Birthday paradox attack)

- Kiểm tra $2^{n/2}$ bản tin có thể tìm ra các bản tin đụng độ với xác suất ~ 0.5
- Cách thức tấn công:
 - Bước 1: Chọn ra $2^{n/2}$ bản tin ngẫu nhiên
 - Bước 2: Tính mã băm cho các bản tin
 - Bước 3: Kiểm tra sự tồn tại của các bản tin đụng độ. Nếu không có, quay lại bước 1.
 - Kỳ vọng: thành công sau 2 lần thử
 - Nhận xét: kích thước mã băm (n-bit) càng lớn thì hàm băm càng an toàn

Kiến trúc hàm băm Merkle-Damgard

- Bản tin đầu vào được chia thành các khối có kích thước s bit
 - Thêm phần đệm nếu cần
- Hàm nén: $z = C(x)$ cung cấp đầu ra z có kích thước n bit
- Kiến trúc Merkle-Damgard: $h = H(m)$



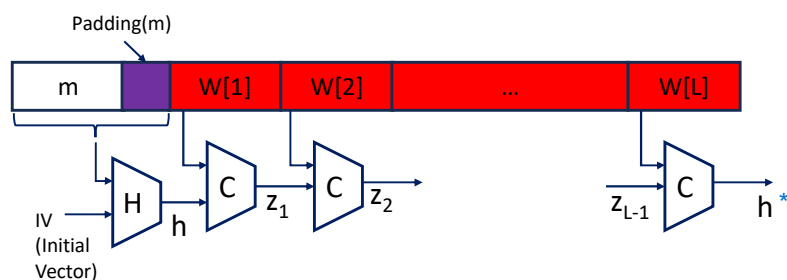
- Các hàm băm sử dụng: MD5, SHA-1, SHA-2

kích thước khối vào có kích thước n bit
IV là hằng số thay đổi

47

Hệ quả của kiến trúc Merkle-Damgard

- Hệ quả 1: Nếu biết độ dài bản tin m và $h = H(m)$ thì có thể tính $h^* = H(m \parallel \text{Padding}(m) \parallel W)$
 - Dễ dàng tính toán $\text{padding}(m)$ nếu biết độ dài của m



- Hệ quả 2: Nếu $H(m) = H(m^*)$ thì $H(m \parallel W) = H(m^* \parallel W)$, với mọi W

có thể padding

48

Một số hàm băm phổ biến

- MD5
 - Kích thước digest: 128 bit
 - Công bố thuật toán tấn công đụng độ (collision attack) vào 1995
 - Đã bị tấn công thành công vào năm 2005
- SHA-1
 - Kích thước digest: 160 bit
 - Đã bị tấn công thành công vào năm 2017
 - Hết hạn vào năm 2030
- SHA-2: 224/256/384/512 bit
- SHA-3: 224/256/384/512 bit

224bits -> v n ch p nh n s d, s h t h n vào 2030
m b m 224bits có an toàn t ng ng vs
khóa i x ng 112bits



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

49

49

SHA-1 (Đọc thêm)

- Bước 1: Padding dữ liệu sao cho bản tin đầu vào có độ dài L sao cho $L \bmod 512 = 448$
- 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 dữ liệu là một chuỗi các khối 512 bit: Y_0, Y_1, \dots, Y_{K-1}
- 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



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

50

SHA-1

- Bước 4: Thực hiện vòng lặp xử lý các khối 512 bit

$$CV_{q+1} = F(Y_q, CV_q)$$

- Xử lý khối dữ liệu Y_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

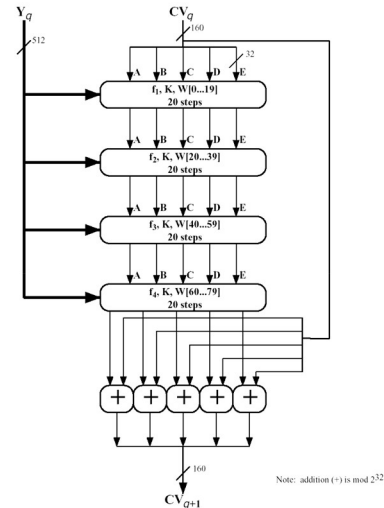
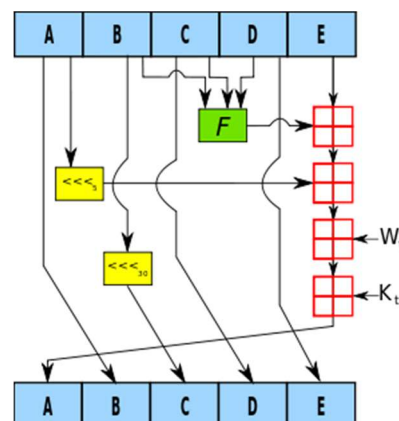


Figure 9.5 SHA-1 Processing of a Single 512-bit Block (SHA-1 Compression Function)

Hàm nén trong SHA-1

- Đầu vào:
 - CV : Khối 160 bit
 - W_t : Khối dữ liệu mở rộng 32 bit
 - K_t : Hằng số
- \boxplus Cộng modulo 2^{32}
- $\ll 5(30)$: dịch trái 5(30) bit
- \wedge : AND, \vee : OR, \neg : NOT
- $F1 = (B \wedge C) \vee (\neg B \wedge D)$
- $F2 = B \oplus C \oplus D$
- $F3 = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
- $F4 = B \oplus C \oplus D$
- Thực hiện vòng lặp 20 bước



MD5 (Đọc thêm)

- Bước 1: Padding dữ liệu sao cho bản tin đầu vào có độ dài L sao cho $L \bmod 512 = 448$
- 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 dữ liệu là một chuỗi các khối 512 bit: Y_0, Y_1, \dots, Y_{K-1}
- Bước 3: Khởi tạo các giá trị hằng số A, B, C, D
 - $A = 0x67\ 45\ 23\ 01$
 - $B = 0xEF\ CD\ AB\ 89$
 - $C = 0x98\ BA\ DC\ FE$
 - $D = 0x10\ 32\ 54\ 76$

MD5

- Bước 4: Thực hiện vòng lặp xử lý các khối 512 bit

$$CV_{q+1} = F(Y_q, CV_q)$$
 - Xử lý khối dữ liệu Y_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

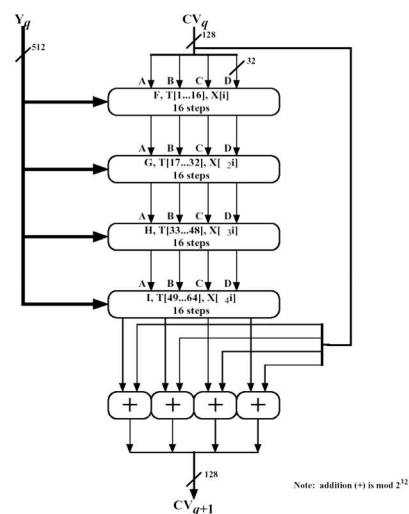
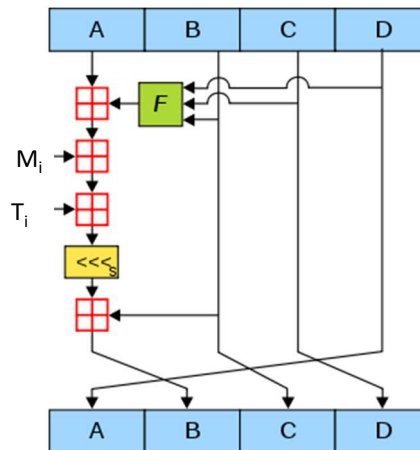


Figure 9.2 MD5 Processing of a Single 512-bit Block (MD5 Compression Function)

Hàm nén trong MD5 (Tham khảo)

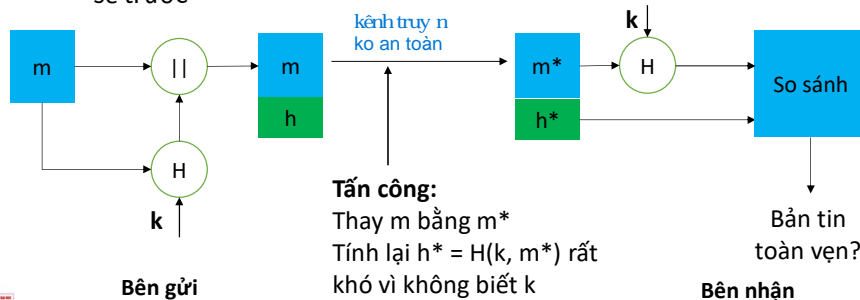
- Đầu vào:
 - CV: Khối 128 bit
 - M_i : khối dữ liệu 32-bit
 - T_i : Hằng số
- \boxplus : Cộng modulo 2^{32}
- $\ll s$: dịch trái s bit
- \wedge : AND, \vee : OR, \neg : NOT
- $F1 = (B \wedge C) \vee (\neg B \wedge D)$
- $F2 = (B \wedge D) \vee (C \wedge \neg D)$
- $F3 = B \oplus C \oplus D$
- $F4 = C \oplus (B \vee \neg D)$
- Thực hiện vòng lặp 16 bước



cách 1 (ch b n tin m và hàm b m public): g i mã b m qua 1 kênh an toàn
cách 2 (s d ng thêm khóa k bí m t): g i c trên kênh không an toàn

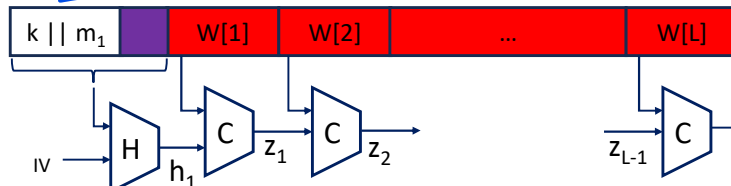
Sử dụng hàm băm

- Xác thực toàn vẹn dữ liệu khi công bố
 - Công bố mã băm của dữ liệu. Ví dụ: công bố trên website
 - Yêu cầu: truy cập dịch vụ web phải an toàn
- Xác thực toàn vẹn dữ liệu khi truyền:
 - Chỉ phát hiện được lỗi ngẫu nhiên trong quá trình truyền
 - Khi có tấn công chủ động: băm bản tin cùng với khóa bí mật đã chia sẻ trước



H(k || m) có an toàn?

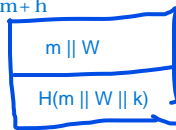
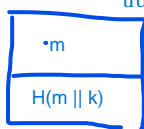
- Tấn công mở rộng kích thước (Length extension attack)
- Có thể tính được $H(k || (m_1 || m_2))$ nếu biết $\text{length}(k)$, $\text{length}(m_1)$ và $h_1 = H(k || m_1)$
 - Kẻ tấn công thay m_1 bằng $m_1 || m_2$ và tính được $H(k || m_1 || m_2)$ dù không biết k



- Các hàm băm bị ảnh hưởng: MD5, SHA-1, SHA-2 vi dùng kí n trúc merkle damgard
- Sửa đổi: sử dụng $H(m || k)$
- An toàn hơn: sử dụng HMAC

57

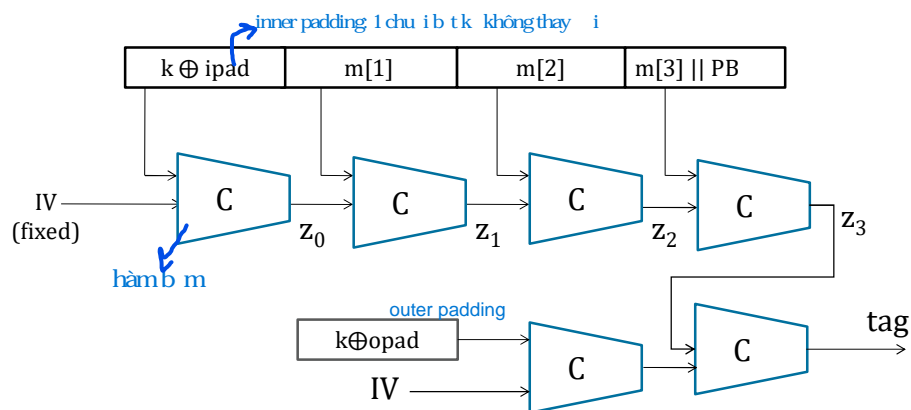
truy n b n tìm m + h



k t n công ko bit khóa k tính

HMAC

- Hashed MAC: xây dựng MAC dựa trên hàm băm

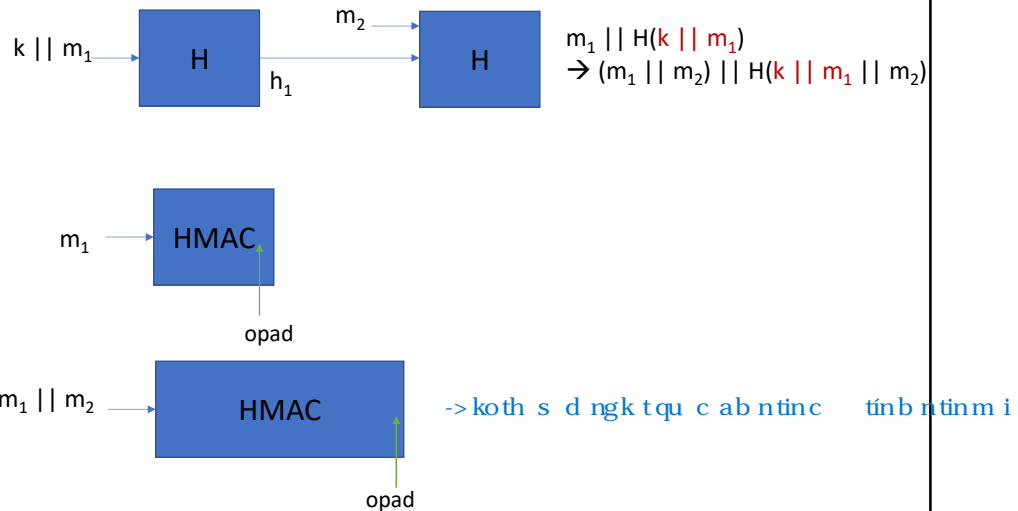


PB: Padding Block

outer padding có tính ch t t ng t inner padding nh ng giá tr khác

58

HMAC chống lại length extension attack

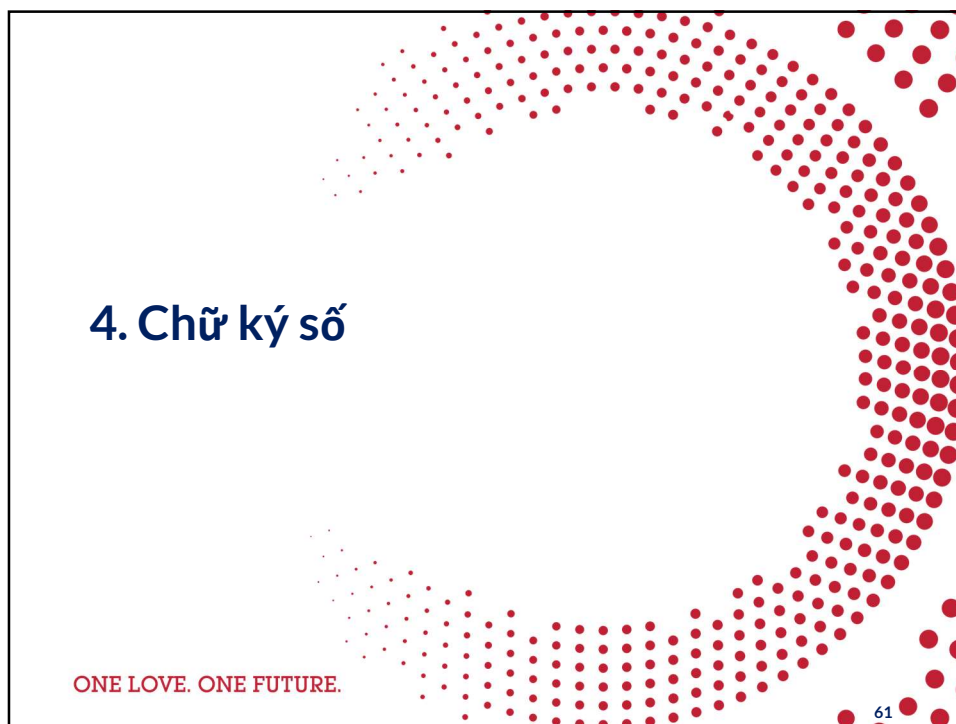


HMAC và MAC

- HMAC có tính chống đụng độ chắc chắn hơn do sử dụng hàm băm
- Tốc độ tính toán của HMAC nhanh hơn
- Kích thước giá trị tag được tạo bởi HMAC lớn hơn
→ an toàn hơn trước các tấn công

HMAC có kích thước lớn hơn và tính linh hoạt hơn

→ xác thực toàn vẹn dữ liệu và bí mật (khóa k)



61

hình thức của MAC: trong bài toán multicast, broadcast -> mỗi phiên trao đổi cần 1 khóa
 -> sử dụng khóa để tính toán và tính toán
 không cần tính chi tiết: 2 bên đều có khóa để tính MAC

Khái niệm – Digital Signature

- Chữ ký số (Digital Signature) hay còn gọi là chữ ký điện tử là đoạn dữ liệu được bên gửi gắn vào văn bản gốc để chứng thực nguồn gốc và nội dung của văn bản
- Yêu cầu:
 - Tính xác thực: người nhận có thể chứng minh được văn bản được ký bởi người gửi
 - Chống từ chối: người gửi không thể phủ nhận được hành động ký vào văn bản
 - Tính toàn vẹn: người nhận có thể chứng minh được không có ai sửa đổi văn bản đã được ký
 - Không thể tái sử dụng: mỗi chữ ký chỉ có giá trị trên 1 văn bản
 - Không thể giả mạo
- Đề nghị của Diffie-Hellman: Sử dụng khóa cá nhân trong một mã công khai để tạo chữ ký.



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

62

62

Chữ ký số

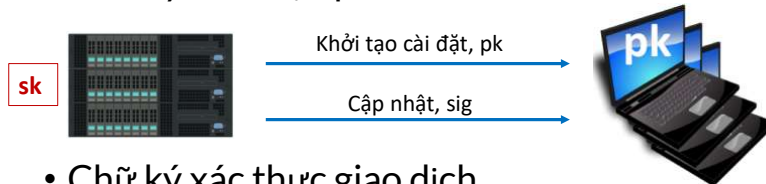
- Hàm sinh khóa: Gen()
- Hàm ký $S(sk, m)$
 - Đầu vào:
 - sk: Khóa ký
 - m: Văn bản cần ký
 - Đầu ra: chữ ký số sig
- Hàm kiểm tra: $V(pk, m, sig)$
 - Đầu vào:
 - pk: Khóa thẩm tra
 - m, sig
 - Đầu ra: True/False
- Tính đúng đắn: $V(pk, m, S(sk, m)) = \text{True}$
- Hàm ký phải có tính ngẫu nhiên
- Chỉ ai có khóa sk mới tạo được chữ ký
- Bất kỳ ai có khóa pk đều có thể kiểm tra chữ ký

Tấn công vào chữ ký số

- Kẻ tấn công chọn trước một số bản tin m_1, m_2, \dots, m_q và có chữ ký của bản tin đó $sig_i \leftarrow S(sk, m_i)$
- Mục tiêu: Tạo ra chữ ký cho bản tin m^*
 $m^* \notin \{m_1, m_2, \dots, m_q\}$
- Yêu cầu đối với chữ ký số: Xác suất tấn công thành công là không đáng kể
- Quiz: Nếu kẻ tấn công tìm được 2 bản tin m_1, m_2 sao cho $V(pk, m_1, sig) = V(pk, m_2, sig) \forall (sk, pk)$ thì chữ ký số đó có an toàn không?

Một số ứng dụng của chữ ký số

- Chữ ký xác thực phần mềm



- Chữ ký xác thực giao dịch



- Chữ ký xác thực thư điện tử: **DKIM**

Một số ứng dụng của chữ ký số (1)

- Ký văn bản trên hệ thống dooffice.hust.edu.vn

Đăng ký phát hành văn bản đi ký số

Ngôn Ngữ: Tiếng Việt | Loại văn bản công tác *: Công tác Chuyên môn

Loại văn bản *: | Hạn trả lời: |

Độ khẩn *: Bình thường | Độ mật *: Bình thường

Trích yếu nội dung *: | **Trình/báo cáo HDTV**
☐ Lưu ý: Chỉ những văn bản trình và xin ý kiến HDTV thì mới sử dụng, các trường hợp gửi HDTV để báo cáo hoặc xem để biết thì không sử dụng lựa chọn này.

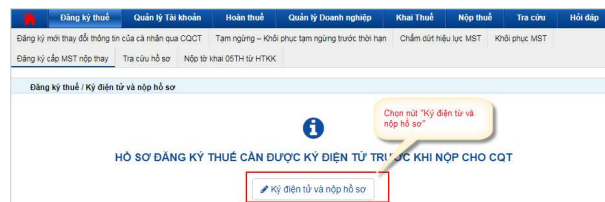
Lãnh đạo ký ban hành *: | Lãnh đạo cơ quan ký tắt: |

Lãnh đạo phòng ban ký trình: | Phòng ban phối hợp đồng trình: |

File văn bản | Mẫu thẻ thư văn bản tham khảo: 829/QĐ-EVN | 1080/QĐ-EVN | File phụ lục

Một số ứng dụng của chữ ký số (2)

- Nộp hồ sơ khi sử dụng dịch vụ công trực tuyến
 - Ví dụ: doanh nghiệp nộp thuế điện tử



- Nghị quyết 130/NQ-CP vào tháng 09/2022: Cung cấp chữ ký số cá nhân cho người dân khi sử dụng dịch vụ công trực tuyến

Một số ứng dụng của chữ ký số (3)

- Chữ ký số văn bản điện tử



Digitally signed
 by Duong
 Minh
 Độ
 Date:
 2020.04.17
 17:46:54 +07'00'

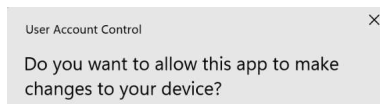
Ký bởi: **CÔNG TY CỔ PHẦN MISA**
 Ký ngày: **11/02/2020**



Digitally signed
 by Duong Minh Độ
 TC
 Date:
 2020.04.17
 17:46:54 +07'00'

Một số ứng dụng của chữ ký số (4)

- Chữ ký số phần mềm
 - Công ty phát hành phần mềm ký lên bộ cài đặt
 - Hệ điều hành chứng thực khi người dùng cài đặt

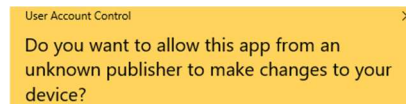


FileZilla FTP Client

Verified publisher: Tim Kosse
File origin: Hard drive on this computer

[Show more details](#)

Phần mềm có chữ ký số tin cậy



UserAccountControlSettings.exe

Publisher: Unknown
File origin: Hard drive on this computer

[Show more details](#)

Phần mềm không có chữ ký số tin cậy



69

69

Một số ứng dụng của chữ ký số (5)

- Chữ ký số xác thực đăng nhập



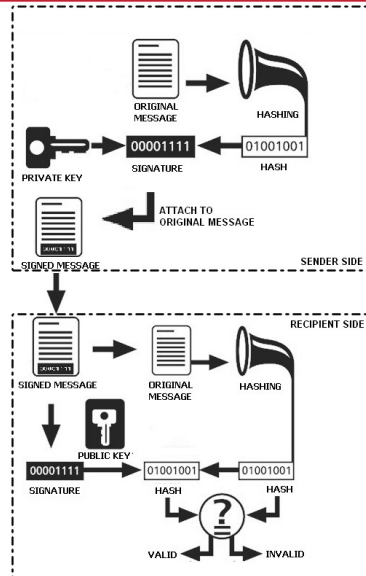
70

70

g i: b m b n tin -> h -> mã v i khóa private -> sig -> g n vào b n tin -> m || sig
 nh n: tách m và sig -> b m m -> h -> g i mã sig vs public key -> h' -> so sánh h, h'

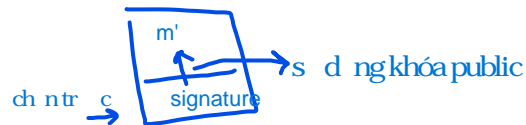
Chữ ký số sử dụng mật mã KCK

- **Phía gửi : hàm ký**
 1. Băm bản tin gốc, thu được giá trị băm h
 2. Mã hóa giá trị băm bằng khóa riêng → chữ kí số sig
 3. Gắn chữ kí số lên bản tin gốc (m || sig)
- **Phía nhận : hàm xác thực**
 1. Tách chữ kí số sig khỏi bản tin.
 2. Băm bản tin m, thu được giá trị băm h
 3. Giải mã sig với khóa công khai của người gửi, thu được h'
 4. So sánh : h và h'. Kết luận.



71

kos d ng h m b m -> k t n công có th t o ra 1 b n tin m' v i ch ký h pl



Chữ ký số RSA

- Sinh cặp khóa: $k_U = (n, e)$, $k_R = (n, d)$
- Chữ ký: $\text{sig} = E(k_R, H(m)) = H(m)^d \bmod n$
- Thẩm tra: nếu $H(m) = \underbrace{\text{sig}^e \bmod n}_{D(k_U, \text{sig})}$ thì chấp nhận

72

Chuẩn chữ ký số DSS (Đọc thêm)

- Digital Signature Standard
- Các tham số:
 - Hàm băm H
 - L: là bội số của 64, $N \leq$ Kích thước mã băm
- Tạo khóa nhóm $k_{UG} = (p, q, g)$:
 - Số nguyên tố q kích thước N bit
 - Số nguyên p kích thước L bit, sao cho p-1 là bội số của q
 - Chọn h là ngẫu nhiên $2 \leq h \leq p-2$
 - $g = h^{(p-1)/q} \bmod p$
- Khóa riêng: x ngẫu nhiên thỏa mãn $0 < x < q$
- Khóa công khai: $y = g^x \bmod p$



Chuẩn chữ ký số DSS

- Tạo chữ ký:
 - Chọn giá trị $0 < k < q$ ngẫu nhiên
 - Tính $r = (g^k \bmod p) \bmod q$; nếu $r = 0$ thì chọn lại k
 - Tính $s = [k^{-1} (H(m) + xr)] \bmod q$; nếu $s = 0$ thì chọn lại k
 - Chữ ký (r, s)
- Thẩm tra chữ ký:
 - $w = (s)^{-1} \bmod q$
 - $u1 = [H(m)w] \bmod q$
 - $u2 = rw \bmod q$
 - $v = [(g^{u1} y^{u2}) \bmod p] \bmod q$
 - Nếu $v = r$ thì chữ ký hợp lệ



Chữ ký số KCK cung cấp dịch vụ nào?

- **Toàn vẹn:** đối phương không thể thay đổi bản tin mà không bị phát hiện
- **Xác thực danh tính:** Nếu Alice xác thực chữ ký hợp lệ với khóa công khai của Bob thì Alice có thể xác định được người tạo bản tin là Bob
- **Chống từ chối:** Bob không thể phủ nhận được bản tin mà anh ta đã tạo ra
- Không có khả năng giữ bí mật:
 - Nếu chữ ký số không thay đổi, kẻ tấn công có thêm thông tin về bản tin gốc

Chữ ký mù (Blind Signature)

- Một số giao dịch điện tử yêu cầu cần che giấu thông tin cá nhân của các bên tham gia:
 - Thương mại điện tử
 - Bầu cử điện tử
- Chữ ký mù: người ký không biết nội dung của văn bản
 - Người kiểm tra tính hợp lệ của phiếu bầu không được phép biết nội dung của phiếu (tên cử tri, người được cử tri bầu...)
 - Sau khi xác minh và chấp nhận cho khách hàng rút tiền, ngân hàng không thể kiểm tra lại trên tờ tiền điện tử lưu thông có tên người rút là gì.

Chữ ký mù RSA cho Phiếu bầu điện tử

- Cơ quan bầu cử sử dụng cặp khóa $k_U = (e, n)$, $k_R = (d, n)$
 - Sau khi đã thực hiện xác thực với cơ quan bầu cử, Alice điền thông tin trên phiếu bầu. Thông tin này được ghi lên bản tin x:
 - Chọn 1 giá trị ngẫu nhiên r
 - Làm mù nội dung lá phiếu: $m' = (H(x).r^e) \bmod n$
 - Đưa cho cơ quan bầu cử ký
 - Cơ quan BC thực hiện ký mù

$$s' = (m')^d \bmod n = ((H(x))^d.r) \bmod n$$
 - Alice xóa mù chữ ký: $s = s'.r^{-1} \bmod n = (H(x))^d \bmod n$
- Lưu ý $1 < r^{-1} < n$ là giá trị sao cho $r.r^{-1} \bmod n = 1$
- Phiếu điện tử của Alice (x, s)
 - Làm thế nào để cơ quan kiểm phiếu tin tưởng đây là phiếu bầu do cơ quan bầu cử phát hành?

Chữ ký điện tử của cơ quan BC lên x

An toàn cho chữ ký số

- Tấn công phát lại thêm 1 y út ng unhiên -> 2b ntưng ng nhaut i 2th i i mớch ký khác nhau
- Sự an toàn của khóa cá nhân
 - Vấn đề: nếu khóa cá nhân bị kẻ tấn công đánh cắp, hắn có thể giả mạo chữ ký của người sở hữu khóa.
 - Giải pháp:
 - Bảo vệ bằng mật khẩu
 - Sử dụng thẻ thông minh (Smart Card)
 - Sử dụng thiết bị lưu trữ an toàn (USB Token)
- Sự tin cậy của khóa công khai.
 - Vấn đề: kẻ tấn công làm sử dụng khóa công khai giả mạo. Nếu người dùng bị đánh lừa, họ sẽ tin cậy vào chữ ký giả mạo
 - Giải pháp: sử dụng hệ thống PKI để phát hành khóa công khai dưới dạng chứng thư số

Bảo vệ khóa cá nhân(1)

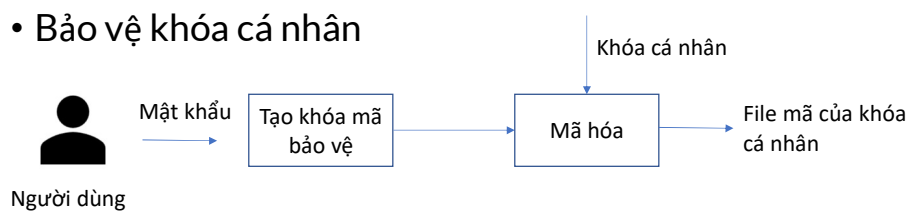
- Khóa cá nhân được đóng gói vào file(ví dụ .pfx), lưu trên thiết bị nhớ thông thường (ổ cứng, USB...)
- File được bảo vệ bởi mật khẩu dạng mã PIN
- Mức an toàn thấp nhất:
 - Dễ dàng sao chép file chứa khóa
 - Mã PIN có thể bị đoán nhận



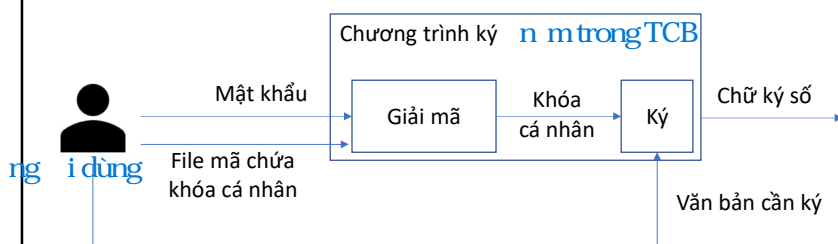
79

Mô hình sử dụng

- Bảo vệ khóa cá nhân



- Sử dụng khóa để tạo chữ ký

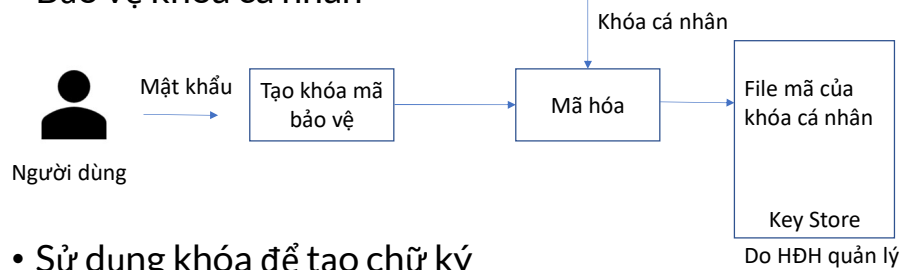


80

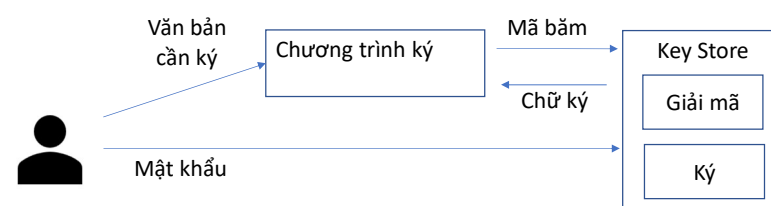
s an toàn khi ch ngtình ký yêu c um t kh ulà an toàn, áng tín c y, n m trong TCB

Cải tiến

- Bảo vệ khóa cá nhân



- Sử dụng khóa để tạo chữ ký



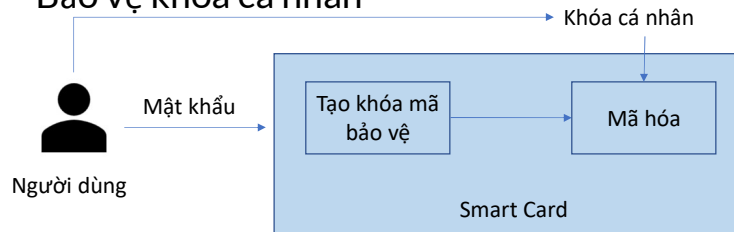
Bảo vệ khóa cá nhân(2)

- Khóa được lưu trữ trên chip điện tử (IC) của Smart Card
- Khi thực hiện ký số:
 - Giá trị băm được truyền vào chip IC
 - Chip IC mã hóa giá trị băm bằng khóa cá nhân (yêu cầu người dùng nhập mã PIN) → chữ ký số
 - Truyền chữ ký số từ Smart Card tới ứng dụng
- Yêu cầu:
 - Phải có đầu đọc chuyên dụng
 - Thư viện API để giao tiếp

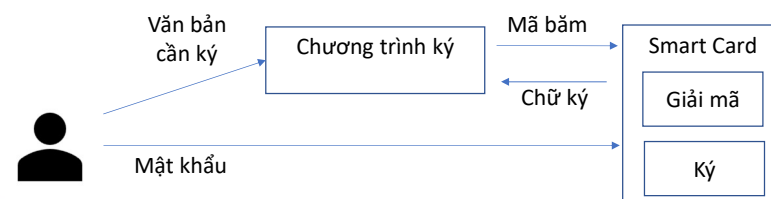


Smart card

- Bảo vệ khóa cá nhân



- Sử dụng khóa để tạo chữ ký



Bảo vệ khóa cá nhân (3)

- Khóa được lưu trữ trong thiết bị nhớ chuyên dụng, sử dụng giao tiếp USB
- Có nhiều mức độ giải pháp khác nhau:
 - Chỉ có chức năng lưu trữ khóa, cho phép ứng dụng truy xuất khóa cá nhân để sử dụng
 - Kịch bản sử dụng tương tự Smart Card
 - Khóa có thể được sinh ngay trên thiết bị



4. Mật mã hậu lượng tử

ONE LOVE. ONE FUTURE.

85

85

Giới thiệu về máy tính lượng tử

- Máy tính điện tử truyền thông thực hiện tính toán với đơn vị là bit
 - Mỗi bit có 2 trạng thái 0 và 1
 - Mỗi bit là độc lập với các bit khác
 - Máy tính lượng tử (Quantum Computer): được phát triển dựa trên lý thuyết lượng tử
 - qubit (bit lượng tử): đơn vị tính toán của máy tính lượng tử
 - Chồng chập lượng tử: 1 qubit thể biểu diễn rất nhiều trạng thái
 - Vướng víu lượng tử: Các qubit có liên kết với nhau
- Một số ít qubit có thể biểu diễn một số lượng khổng lồ các trạng thái.
- Tốc độ tính toán vượt xa máy tính điện tử
- IBM Osprey (2002): có thể sử dụng 433 qubits



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

86

86

Ảnh hưởng của máy tính lượng tử với mật mã

- Về lý thuyết, các thuật toán được thiết kế để chạy trên máy tính lượng tử có hiệu năng tốt hơn hoặc giải quyết rất nhiều bài toán khó hiện nay
- Bài toán tìm kiếm vết cạn: thuật toán Grover có độ phức tạp $O(\sqrt{n})$
- Tấn công vét cạn vào AES-128 chỉ còn trong 2^{64}
- Tấn công vét cạn vào SHA-256 chỉ còn trong 2^{64}
- Tấn công vét cạn vào AES-256 còn 2^{128}
 - Tương đương với tấn công vét cạn vào AES-128 bằng máy tính điện tử truyền thống
 - AES-256 là an toàn
- Tấn công ngẫu sinh vào SHA-512 còn trong 2^{128}
 - SHA-512 và SHA3-512 là an toàn



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

87

87

Ảnh hưởng của máy tính lượng tử với mật mã

- Bài toán phân tích số nguyên thành thừa số nguyên tố: thuật toán Shor có độ phức tạp $O((\log n)^2(\log \log n)(\log \log \log n))$
- Bài toán tính logarit rời rạc: các thuật toán hệ quả từ thuật toán Shor
- Nguy cơ với mật mã KCK: Thời gian gian tấn công là đa thức
 - Chỉ trong một vài phút có thể bẻ khóa thành công
- Yêu cầu mới: Xây dựng các hệ mật mã KCK mới (được gọi là mật mã hậu lượng tử hay an toàn lượng tử)
 - An toàn trước tấn công trên máy tính lượng tử
 - Có thể thực hiện trên máy tính điện tử truyền thống



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

88

88

Một số thuật toán mật mã hậu lượng tử

- 2022: NIST phê chuẩn 4 thuật toán vượt qua vòng cuối
 - CRYSTALS-KYBER: mã hóa và đóng gói khóa
 - CRYSTALS-Dilithium: Chữ ký số
 - FALCON: Chữ ký số
 - SPHINCS+: Chữ ký số
- 2024: NIST công bố bản thảo tiêu chuẩn cho các thuật toán
 - FIPS 203: Dẫn xuất từ CRYSTALS-KYBER
 - FIPS 204: Dẫn xuất từ CRYSTALS-Dilithium
 - FIPS 205: Dẫn xuất từ SPHINCS+
- Ngoài ra, một số thuật toán khác đã được phê chuẩn bước vào vòng kiểm tra cuối cùng