BÀI 2.
MẬT MÃ VÀ ỨNG DỤNG

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

1

1

Nội dung

- Mật mã (cipher) là gì?
- Nguyên tắc chung của các hệ mật mã
- Hệ mật mã khóa đối xứng
- Hệ mật mã khóa bất đối xứng
- Mã xác thực thông điệp
- Hàm băm mật mã
- Giao thức mật mã

2

2

1. MẬT MÃ LÀ GÌ?

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

Khái niệm mật mã

- Mã hóa (code): biến đổi cách thức biểu diễn thông tin
- Mật mã (cipher): mã hóa để che giấu, giữ mật thông tin
  - Lưu trữ
  - Truyền tin
- Mật mã học (cryptography): ngành khoa học nghiên cứu các phương pháp toán học để mã hóa giữ mật thông tin
- Thám mã (cryptoanalysis): nghiên cứu các phương pháp toán học để phá vỡ hệ mật mã
- Là công cụ hiệu quả giải quyết bài toán ATBM, là cơ sở cho nhiều cơ chế khác (xác thực, nhận dạng)
  - ✓Nhưng không vạn năng

Xây dựng mô hình (mật mã khóa đối xứng)

- Alice và Bob đã chia sẻ thông tin bí mật K gọi là khóa
- Alice cần gửi cho Bob một thông điệp M (bản rõ). Nội dung thông điệp cần giữ bí mật trước quan sát của Eve (kẻ tấn công, thám mã)

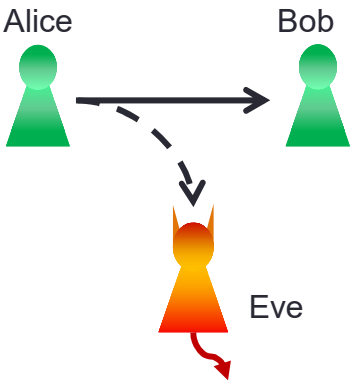
Mã hóa: C = E(K, M)

C: bản mã

- Alice gửi bản mã lên kênh truyền. Bob và Eve đều thu được thông điệp này. Chỉ có Bob giải mã để thu được bản rõ

Giải mã: M = D(K, C)

- Mật mã khóa đối xứng: dùng khóa K trong cả hai quá trình mã hóa và giải mã



5

5

Định luật Kerckhoffs

- Một hệ mật mã cần an toàn ngay cả khi mọi thông tin về hệ, trừ khóa bí mật, là công khai
- Tại sao?

6

6

Lý thuyết Shannon

- Hệ mật hoàn hảo: độ an toàn của hệ mật không phụ thuộc vào số bản mã và thời gian kẻ tấn công sử dụng để thám mã (An toàn vô điều kiện)
- Lý thuyết Shannon: Một hệ mật mã là hoàn hảo thì
  - Độ dài của khóa tối thiểu bằng độ dài bản tin rõ
  - Khóa chỉ sử dụng một lần
  - → Tại sao khó đạt được trên thực tế?

Điều kiện cần

- An toàn theo tính toán: thỏa mãn đồng thời 2 điều kiện
  - Thời gian để thám mã thành công lớn hơn thời gian cần giữ mật thông tin
  - Chi phí để thám mã thành công lớn hơn giá trị thông tin thu được

7

7

Thông tin tham khảo – Kích thước khóa

- Khóa có kích thước bao nhiêu?
  - Mật mã được coi là an toàn khi phương pháp vét cạn (brute-force) là cách nhanh nhất để bẻ khóa
  - Mục tiêu: giảm thiểu nguy cơ bị tấn công vét cạn (đạt độ an toàn theo tính toán)
- Chi phí để bẻ khóa DES (năm 2008)
  - 64 bit: $10.000
  - 87 bit: $10.000.000.000 (thời gian bẻ khóa không đổi)
- Tuy nhiên, vét cạn là phương pháp tấn công tầm thường.
- Tham khảo kích thước khóa nên sử dụng trong tương lai tại địa chỉ http://csrc.nist.gov/groups/ST/toolkit/key_management.html

8

8

Thông tin tham khảo – Kích thước khóa

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
2010 (Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1** SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
2011 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
> 2030	128	AES-128	3072	256	3072	256	SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
>> 2030	192	AES-192	7680	384	7680	384	SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512
>>> 2030	256	AES-256	15360	512	15360	512	SHA-512	SHA-256 SHA-384 SHA-512

http://www.keylength.com

9

9

Thời hạn khóa

- Thời hạn khóa (Cryptoperiod): thời gian khóa có hiệu lực sử dụng
  - Giảm xác suất bẻ khóa thành công của thám mã
  - Giảm thiệt hại khi xảy ra tình trạng khóa bị tấn công/sử dụng sai
- Các yếu tố ảnh hưởng đến thời hạn khóa:
  - Độ an toàn của hệ mật
  - Môi trường triển khai và vận hành ứng dụng
  - Lượng thông tin được mã hóa
  - Thời hạn giữ mật thông tin
  - Mục đích sử dụng khóa
  - Số phần tử chia sẻ khóa/Số bản sao được tạo ra
  - Sự phát triển của các hệ thống tính toán…

10

10

Thông tin tham khảo – Thời hạn khóa

Key Type *Move the cursor over a type for description*	Cryptoperiod Originator Usage Period (OUP)	Recipient Usage Period
Private Signature Key	1-3 years	
Public Signature Key	Several years (depends on key size)	
Symmetric Authentication Key	<= 2 years	<= OUP + 3 years
Private Authentication Key	1-2 years	
Public Authentication Key	1-2 years	
Symmetric Data Encryption Key	<= 2 years	<= OUP + 3 years
Symmetric Key Wrapping Key	<= 2 years	<= OUP + 3 years
Symmetric and asymmetric RNG Keys	Upon reseeding	
Symmetric Master Key	About 1 year	
Private Key Transport Key	<= 2 years (1)	
Public Key Transport Key	1-2 years	
Symmetric Key Agreement Key	1-2 years	
Private Static Key Agreement Key	1-2 years (2)	
Public Static Key Agreement Key	1-2 years	
Private Ephemeral Key Agreement Key	One key agreement transaction	
Public Ephemeral Key Agreement Key	One key agreement transaction	
Symmetric Authorization Key	<= 2 years	
Private Authorization Key	<= 2 years	
Public Authorization Key	<= 2 years	

11

11

2. Hệ mật mã khóa đối xứng

- Symmetric cryptography, Secret-key cryptography: sử dụng cùng một khóa khi mã hóa và giải mã.
- Được phát triển từ rất sớm
- Thuật toán mã hóa: phối hợp các toán tử
  - Thay thế
  - Đổi chỗ
  - XOR ⊕ (Tại sao dùng XOR?)
- Tốc độ thực hiện các thuật toán nhanh, có thể thực hiện bằng dễ dàng bằng phần cứng
- Một số hệ mật mã khóa đối xứng hiện đại: AES, Salsa20, DES, 2DES, 3DES, RC4, RC5,

12

12

2.1. Sơ đồ chung

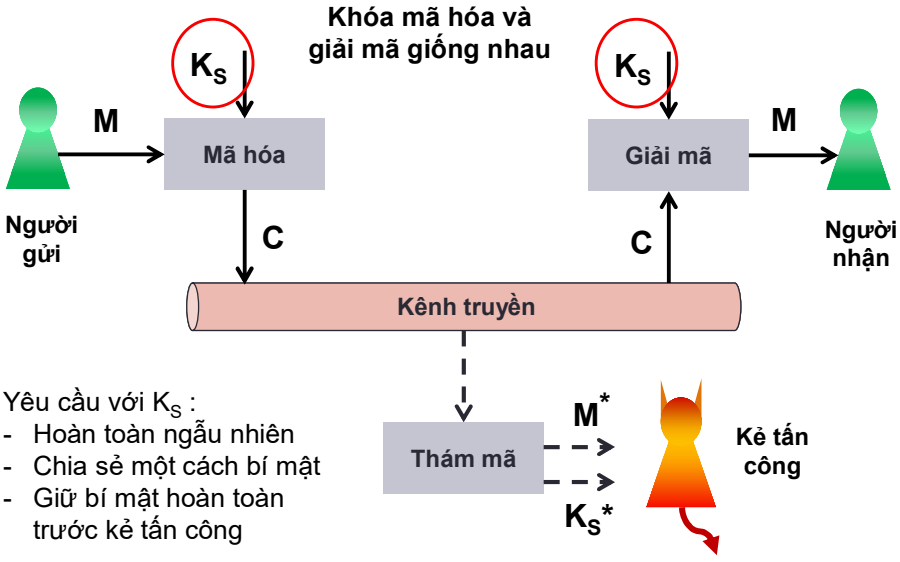
Hệ mật mã gồm:

- Bản rõ (plaintext-m): chứa thông tin không được che dấu
- Bản mật (ciphertext-c): chứa thông tin được che dấu
- Khóa (key- k_S): giá trị đã được chia sẻ bí mật
- Mã hóa (encrypt-E): $c = E(k_S, m)$
  - E là hàm ngẫu nhiên
- Giải mã (decrypt): $m = D(k_S, c)$
  - D là hàm xác định
- Tính đúng đắn $D(k_S, E(k_S, m)) = m$

13

13

Sơ đồ chung



Yêu cầu với K_S :

- Hoàn toàn ngẫu nhiên
- Chia sẻ một cách bí mật
- Giữ bí mật hoàn toàn trước kẻ tấn công

14

14

Mật mã dòng (stream cipher)

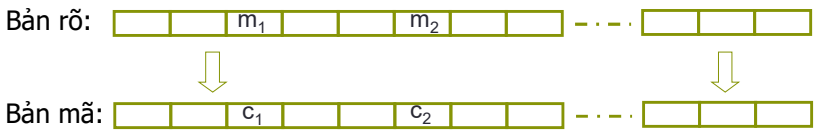
- Xử lý văn bản rõ theo dòng byte, thời gian thực
  - RC4 (900 Mbps), SEAL (2400 Mbps), RC5(450 Mbps)
- Phù hợp với các hệ thống truyền dữ liệu thời gian thực trên môi trường mạng máy tính
- An toàn nếu khóa hoàn toàn ngẫu nhiên, chỉ dùng 1 lần (one-time-pad)
  - Trên thực tế: khóa giả ngẫu nhiên dùng nhiều lần (n-time-pad)
  - → một số phương pháp mật mã dòng không còn an toàn (RC4)

15

15

Mật mã khối

- Chia văn bản gốc thành các khối có kích thước như nhau
- Xử lý mã hóa và giải mã từng khối độc lập
- ECB - Electronic Code Book (Chế độ mã từ điển)

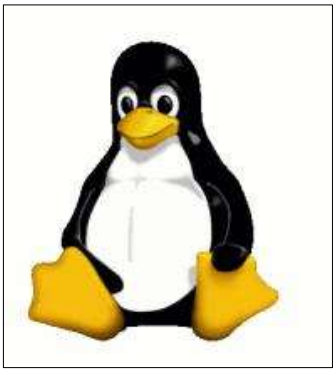


- Hạn chế?

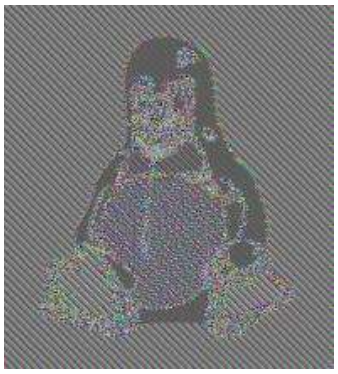
16

16

Ví dụ về mật mã khối chế độ ECB



Ảnh gốc



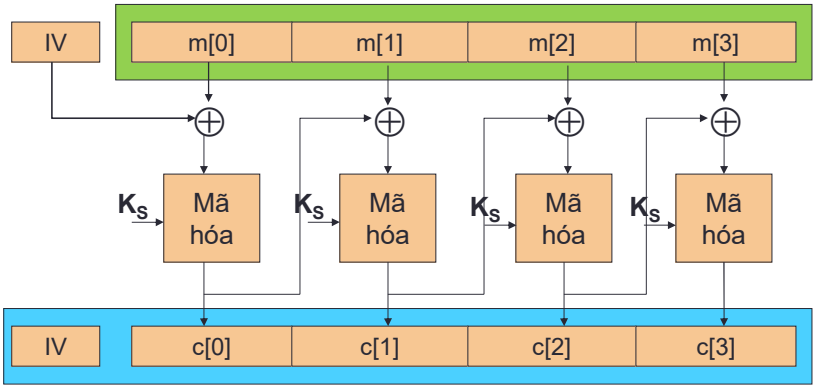
Ảnh được mã hóa ở chế độ ECB

17

17

Chế độ CBC - Cipher Block Chaining

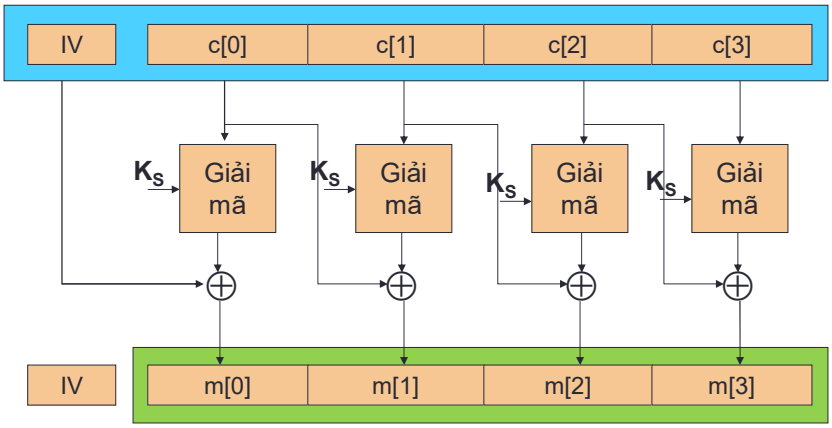
- Chế độ mã móc xích
- Có thể thay thế mã dòng: Ưu điểm? Hạn chế?



18

18

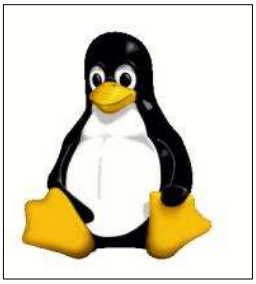
CBC – Giải mã



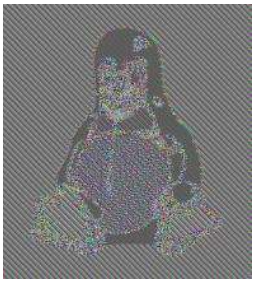
19

19

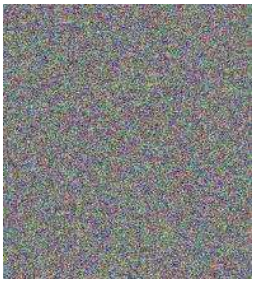
CBC – So sánh với ECB



Ảnh gốc



Mã hóa ở chế độ ECB



Mã hóa ở chế độ CBC

20

20

Những hạn chế của mật mã khóa đối xứng

- Cần kênh mật để chia sẻ khóa bí mật giữa các bên
  - Làm sao để chia sẻ một cách an toàn cho lần đầu tiên
- Số lượng khóa lớn: n(n-1)/2
- Khó ứng dụng trong các hệ thống mở (E-commerce)
- Không dễ dàng để xác thực đối với thông tin quảng bá (Chúng ta sẽ quay trở lại vấn đề này trong những bài sau)

21

21

3. Hệ mật mã khóa bất đối xứng

- Asymmetric key cryptography, Public key cryptography
- Tháng 11/1976, Diffie và Hellman giới thiệu ý tưởng về một kịch bản chia sẻ khóa bí mật (của hệ mật mã khóa đối xứng) mới mà không truyền trực tiếp giá trị của khóa.
- Độ an toàn dựa trên độ khó khi giải một số bài toán:
  - Phân tích một số thành thừa số nguyên tố
  - Tính logarit rời rạc
- Các thuật toán dựa trên các hàm toán học
- Một số hệ mật mã khóa công khai: RSA, El-Gamal

22

22

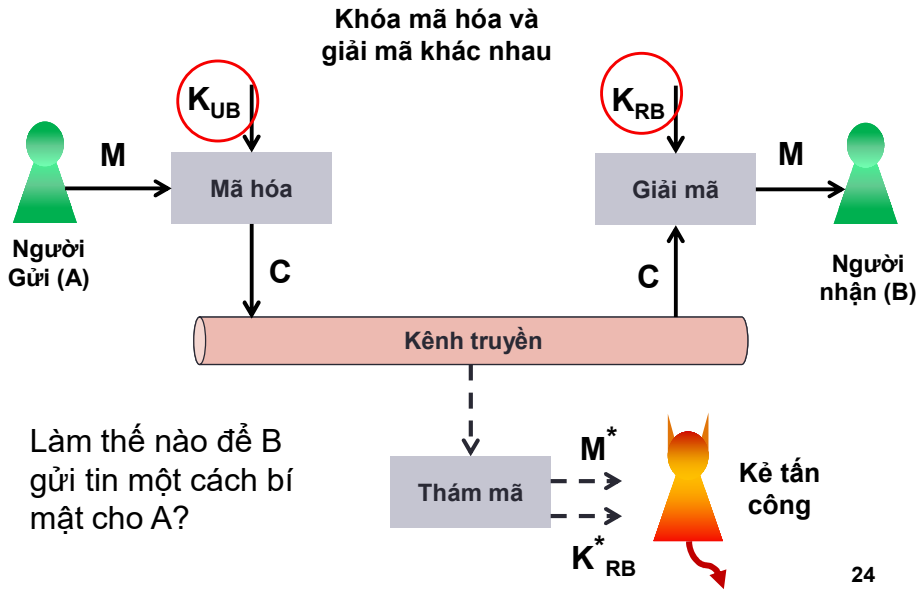
Sơ đồ bí mật

- Hệ mật mã gồm:
  - Bản rõ (plaintext-M): bản tin được sinh ra bởi bên gửi
  - Bản mật (ciphertext-C): bản tin che giấu thông tin của bản rõ, được gửi tới bên nhận qua một kênh không bí mật
  - Khóa: Bên nhận có **1 cặp** khóa:
    - ✓Khóa công khai K_{UB} : công bố cho tất cả biết (trong đó có cả kẻ tấn công)
    - ✓Khóa cá nhân K_{RB} : bên nhận giữ bí mật, không chia sẻ
  - Mã hóa (encrypt-E): $C = E(K_{UB}, M)$
  - Giải mã (decrypt): $M = D(K_{RB}, C) = D(K_{RB}, E(K_{UB}, M))$

23

23

Sơ đồ bí mật

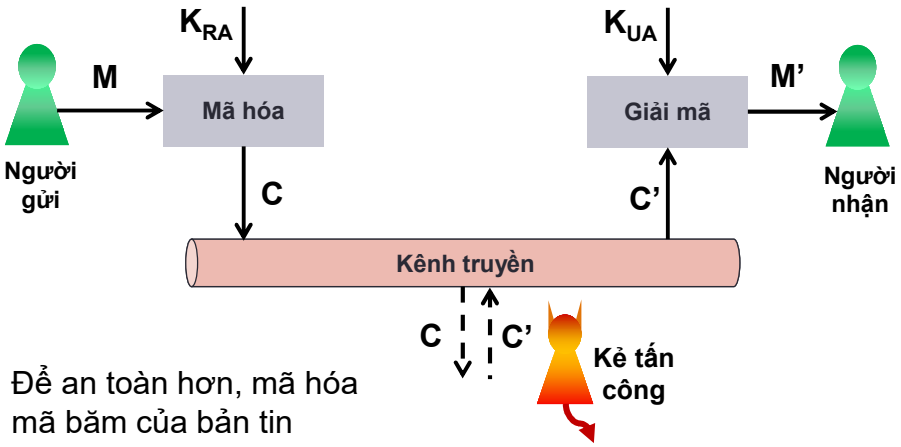


24

24

Sơ đồ xác thực

- Sơ đồ xác thực: mã hóa bằng khóa cá nhân của người gửi



Để an toàn hơn, mã hóa mã băm của bản tin

25

25

Kết hợp mật mã khóa công khai và mật mã khóa đối xứng

- Ưu điểm của mật mã khóa công khai:
  - Không cần chia sẻ khóa mã hóa K_{UB} một cách bí mật
    - ✓ Dễ dàng ứng dụng trong các hệ thống mở
  - Khóa giải mã K_{RB} chỉ có B biết:
    - ✓ An toàn hơn
    - ✓ Có thể sử dụng K_{RB} để xác thực nguồn gốc thông tin (Chúng ta sẽ quay lại vấn đề này trong bài sau)
  - Số lượng khóa để mã mật tỉ lệ tuyến tính với số phần tử (n phần tử → n cặp khóa)
- Nhưng...

26

26

Kết hợp mật mã khóa công khai và mật mã khóa đối xứng

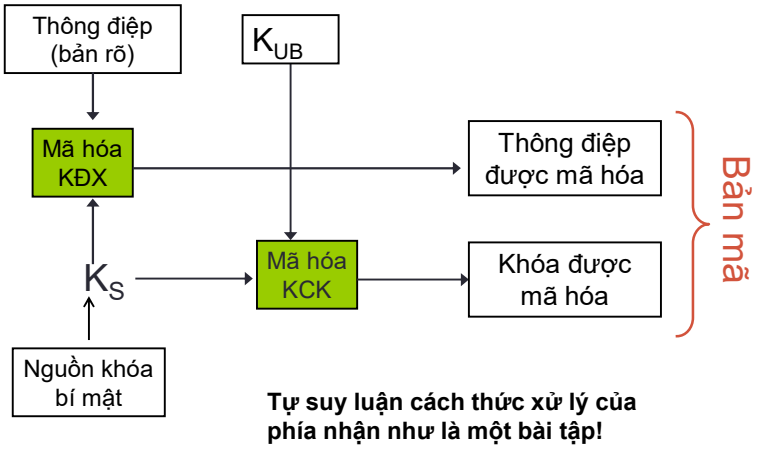
- Hạn chế của mật mã khóa công khai so với mật mã khóa đối xứng:
  - Kém hiệu quả hơn: khóa có kích thước lớn hơn, chi phí tính toán cao hơn
  - Có thể bị tấn công toán học
  - → Kết hợp 2 hệ mật mã

27

27

Sơ đồ "lai"

- Phía gửi



Tự suy luận cách thức xử lý của phía nhận như là một bài tập!

28

28

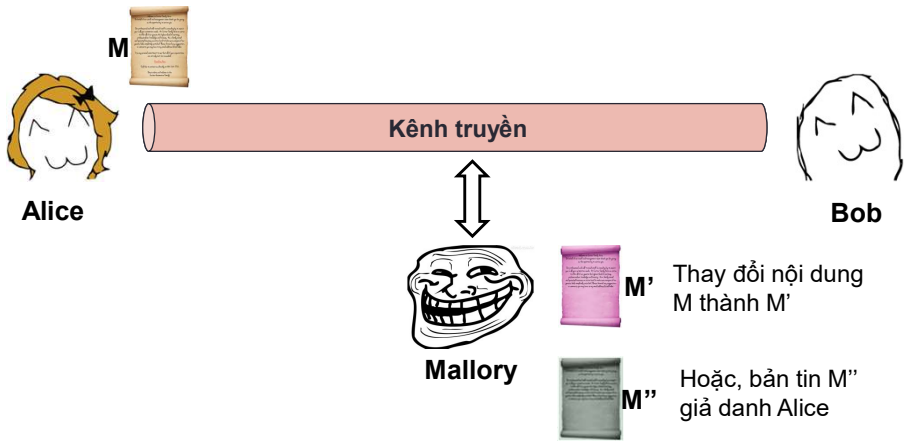
4. XÁC THỰC THÔNG ĐIỆP

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

29

29

Xác thực thông điệp



30

30

Xác thực thông điệp

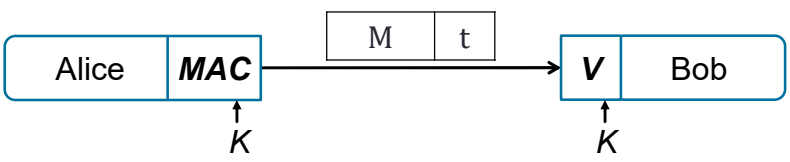
- Bản tin phải được xác minh:
  - Nội dung toàn vẹn: bản tin không bị sửa đổi
    - ✓Bao hàm cả trường hợp Bob cố tình sửa đổi
  - Nguồn gốc tin cậy:
    - ✓Bao hàm cả trường hợp Alice phủ nhận bản tin
    - ✓Bao hàm cả trường hợp Bob tự tạo thông báo và "vu khống" Alice tạo ra thông báo này
  - Đúng 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)

31

31

Message Authentication Code

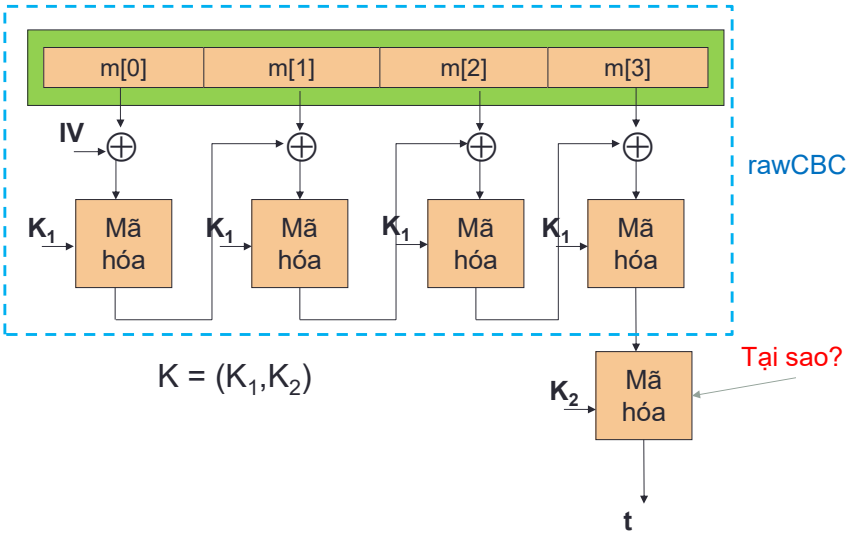
- Xây dựng trên cơ sở hệ mật mã khóa đối xứng:
  - Hai bên đã trao đổi một cách an toàn khóa mật K
  - Sử dụng các thuật toán mã hóa khối ở chế độ CBC-MAC
- Bên gửi:
  - Tính toán t = MAC(K, M) : kích thước cố định, không phụ thuộc kích thước của M
  - Truyền (M||t)
- Bên nhận: xác minh Verify(K, M,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



32

32

CBC-MAC



33

33

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

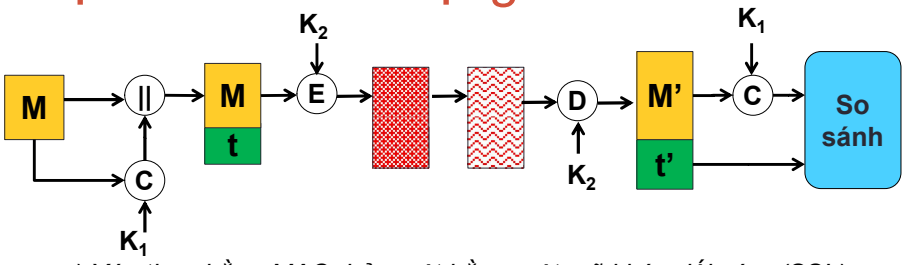
Từ chối giải mã các bản mã không hợp lệ

- 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

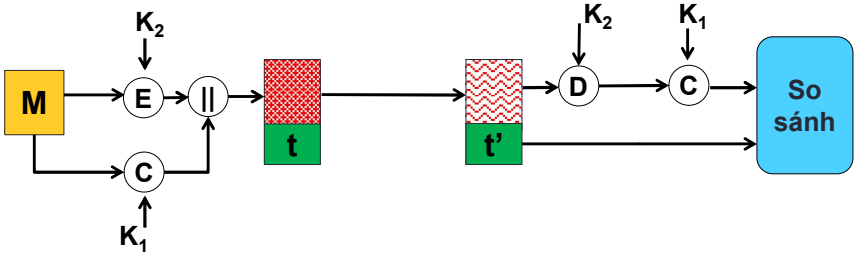
34

34

Một số sơ đồ sử dụng mã MAC



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

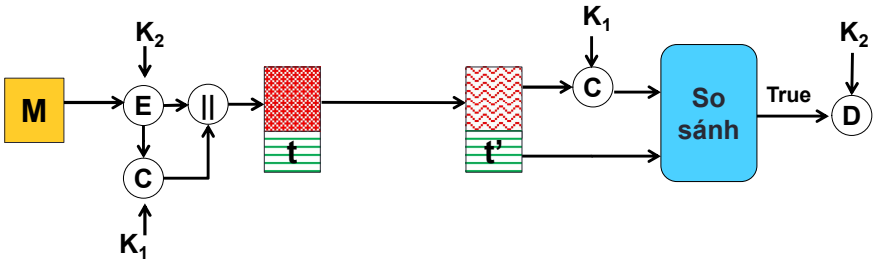


b) Xác thực bằng MAC, bảo mật bằng mật mã khóa đối xứng (SSH)

35

35

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

36

36

Hàm băm

- *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

37

37

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

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

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

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

$$\begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} = H(m)$$

38

38

Một số hàm băm phổ biến

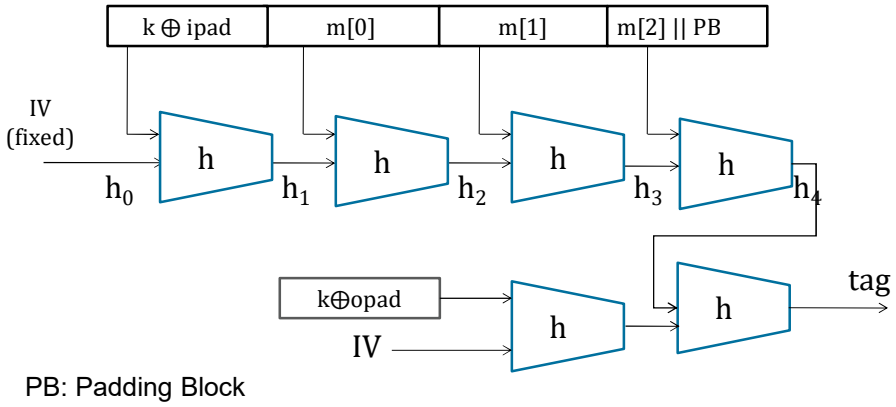
- MD5
  - Kích thước digest: 128 bit
  - Công bố thuật toán tấn công đụng độ (collision attack) vào 1995
  - Năm 2005 tấn công thành công
- SHA-1
  - Kích thước digest: 160 bit
  - Công bố tấn công thành công vào năm 2015
  - Hết hạn vào năm 2030
- SHA-2: 224/256/384/512 bit
- SHA-3: 224/256/384/512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

39

39

HMAC – Hashed MAC

- Hàm băm sử dụng khóa để xác thực thông điệp



PB: Padding Block

40

40

Khái niệm – Digital Signature

- Chữ kí 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 gửi
  - 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
  - Chống từ chối: người gửi không thể phủ nhận được hành động ký vào văn bản
- Đề nghị của Diffie-Hellman: Sử dụng khóa cá nhân trong mật mã công khai để tạo chữ ký.

41

41

Chữ ký số

- Hàm sinh khóa: $\mathrm{Gen}()$
- Hàm ký $\mathrm{S}(\mathrm{sk}, \mathrm{m})$
  - Đầu vào:
    - sk: Khóa ký
    - m: Văn bản cần ký
  - Đầu ra: chữ ký số S
- Hàm kiểm tra: $\mathrm{V}(\mathrm{pk}, \mathrm{m}, \mathrm{sig})$
  - Đầu vào:
    - pk: Khóa thẩm tra
    - m, sig
  - Đầu ra: True/False

- Hàm ký phải có tính ngẫu nhiên
- Bất kỳ ai có khóa sk đều có thể tạo chữ ký
- Bất kỳ ai có khóa pk đều có thể kiểm tra chữ ký

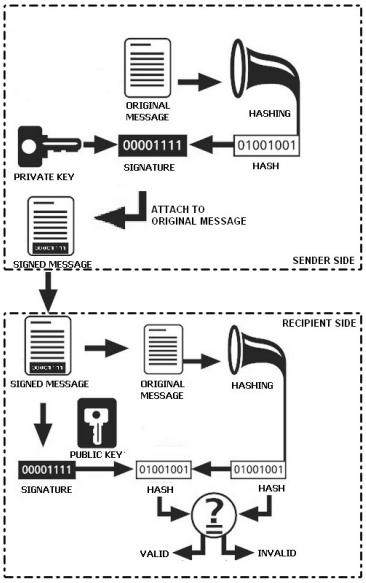
- Tính đúng đắn: $\mathrm{V}(\mathrm{pk}, \mathrm{m}, \mathrm{S}(\mathrm{sk},\mathrm{m})) = \mathrm{True}$

42

42

Chữ ký số dựa trên hàm băm

- **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.



43

Chữ ký số RSA

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

$D(k_U, H(m))$

44

44

5. PHÂN PHỐI KHÓA

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

Yêu cầu Perfect Forward Secrecy

- Một giao thức cần đảm bảo an toàn cho khóa phiên ngắn(short-term key) trong các phiên làm việc trước là an toàn khi khóa phiên dài (long-term key) không còn an toàn

Tấn công khóa đã biết (known-key)

- Sử dụng sự mất an toàn của khóa phiên trong các phiên làm việc trước để tấn công các phiên làm việc tới.
- Khi kẻ tấn công biết giá trị khóa phiên của các phiên cũ, nó có thể dùng khóa này để tính toán khóa phiên trong các phiên truyền thông mới

47

47

5.1. PHÂN PHỐI KHÓA BÍ MẬT

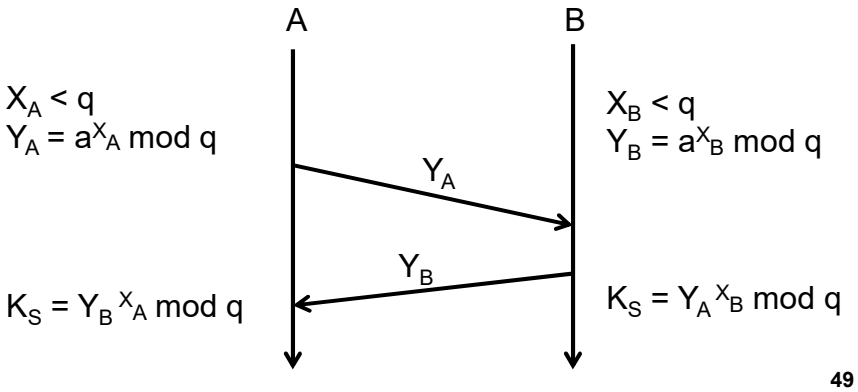
Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

48

48

Kịch bản trao đổi khóa bí mật của Diffie-Hellman

- Alice và Bob cùng chia sẻ một khóa nhóm (q, a). Trong đó
  - q là một số nguyên tố
  - 1< a < q thỏa mãn: $(a^i \bmod q) \neq a^j \bmod q \ \forall\ 1 < i \neq j < q$

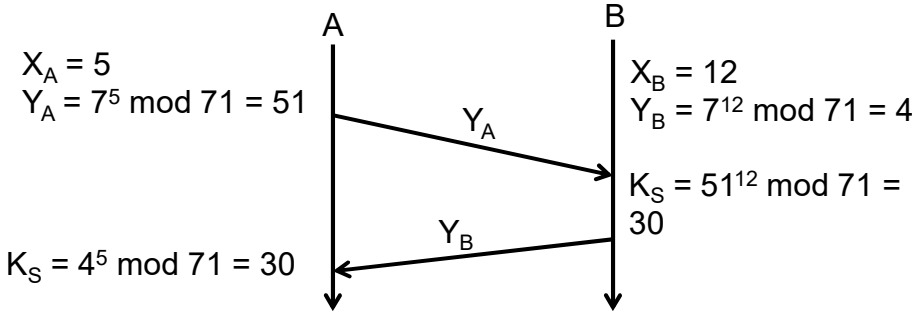


49

49

Ví dụ

- Khóa chung của nhóm q = 71, a = 7
  - Hãy tự kiểm tra điều kiện thỏa mãn của a
- A chọn $X_A = 5$, tính $Y_A = 7^5 \bmod 71 = 51$



- Vấn đề an toàn của sơ đồ này sẽ được xem xét đến sau. Rút ra được điều gì từ sơ đồ này?

50

50

Đặc điểm của sơ đồ

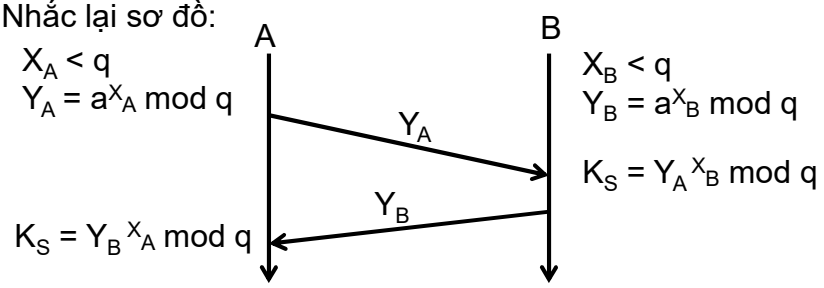
- Ưu điểm:
  - Thỏa mãn yêu cầu PFS
  - Chống lại tấn công khóa đã biết
  - Không cần kênh truyền bí mật
- Hạn chế:
  - Kích thước khóa lớn, sử dụng hàm toán học → chi phí tính toán cao
  - Cần xác thực cho các giá trị công khai

51

51

Tấn công sơ đồ trao đổi khóa Diffie-Hellman

- Nhắc lại sơ đồ:



- Kịch bản tấn công man-in-the-middle
  - C sinh 2 cặp khóa (X'_A, Y'_A) và (X'_B, Y'_B)
  - Tráo khóa Y_A bằng Y'_A, Y_B bằng Y'_B
  - Hãy suy luận xem tại sao C có thể biết được mọi thông tin A và B trao đổi với nhau

52

52

Giao thức phân phối khóa tập trung (Giao thức Needham-Schroeder)

(1) A → KDC: ID_A || ID_B || **N_1**

(2) KDC → A: $E(K_A, K_S$ || ID_A || ID_B || **N_1** || $E(K_B, ID_A$ || $K_S))$

(3) A giải mã, kiểm tra N_1 thu được K_S

(4) A → B: $E(K_B, ID_A$ || $K_S)$ ← B giải mã, thu được K_S

(5) B → A: $E(K_S, N_2)$ ← A giải mã, có được N_2, tính $f(N_2)$

(6) A → B: $E(K_S, f(N_2))$ ← B giải mã kiểm tra $f(N_2)$

(7) A ↔ B: $E(K_S$, Data)

E(K, .): mã hóa có xác thực

N_1, N_2: giá trị dùng 1 lần (nonce)

f(x): hàm biến đổi dữ liệu bất kỳ(Tại sao cần?)

- Hãy xem xét lại tính an toàn của giao thức này!

53

53

5.2. PHÂN PHỐI KHÓA CÔNG KHAI

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

54

54

Phân phối khóa công khai

- Bên thứ 3 được tin cậy – CA(Certificate Authority)
  - Có cặp khóa (K_{UCA}, K_{RCA})
  - Phát hành chứng thư số cho khóa công khai của các bên có dạng

  Cert = E(K_{RCA}, ID || K_U || Time)

  ID: định danh của thực thể

  K_U: khóa công khai của thực thể đã được đăng ký tại CA

  Time: Thời hạn sử dụng khóa công khai. Thông thường có thời điểm bắt đầu có hiệu lực và thời điểm hết hiệu lực.

55

55

Phân phối khóa công khai

(1) A → CA: ID_A || K_{UA} || $Time_A$

(2) CA → A: $Cert_A$= E(K_{RCA}, ID_A || K_{UA} || $Time_A$)

(3) B → CA: ID_B || K_{UB} || $Time_B$

(4) CA → B: $Cert_B$= E(K_{RCA}, ID_B || K_{UB} || $Time_B$)

(5) A → B: $Cert_A$

(6) B → A: $Cert_B$

- Làm thế nào để A và B có thể yên tâm sử dụng khóa công khai của nhau?
- Hãy cải tiến lại các giao thức trong các khâu cần đến xác thực thông điệp (sử dụng MAC hoặc hàm băm)
- Đọc thêm về PKI và chứng thư số theo chuẩn X.509

56

56

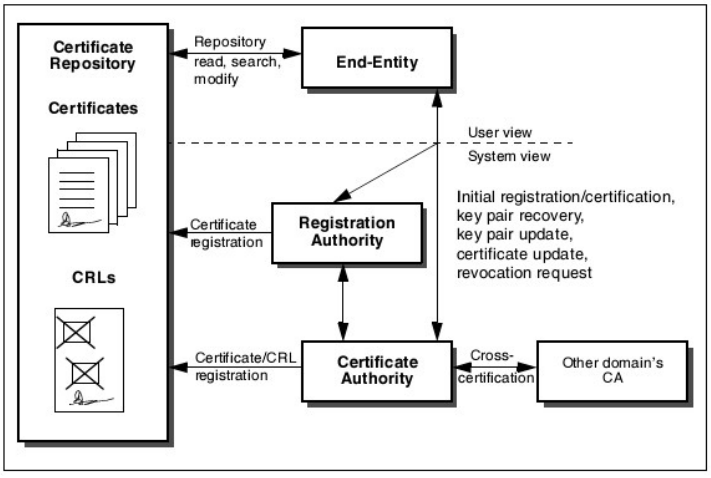
Hạ tầng khóa công khai (PKI)

- Public Key Infrastructure
- Khái niệm: hệ thống cấp phát, quản lý và chứng thực chứng thư số bao gồm
  - Phần cứng
  - Phần mềm
  - Chính sách
  - Thủ tục
- Một số sản phẩm mã nguồn mở: OpenCA, EJBCA

57

57

Các thành phần của PKI



58

58

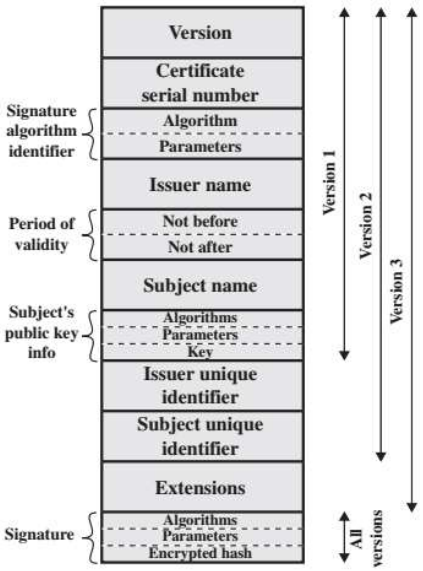
Các thành phần của PKI

- End Entity: đối tượng sử dụng chứng thư số(người dùng, thiết bị, phần mềm, dịch vụ)
- Certificate Authority(CA)
  - Phát hành chứng thư số
  - Phát hành danh sách chứng thư bị thu hồi(CRL)
  - Triển khai tập trung
- Registration Authority(RA)
  - Chứng thực thông tin đăng ký của người dùng
  - Có thể triển khai phân tán để giảm chi phí khi mở rộng hệ thống
- Certificate Reposiroty(CR): Lưu trữ, chứng thực chứng thư số

59

59

Chứng thư số X.509



60

60

Chứng thực chứng thư số

Chứng thư số cần được kiểm tra tính tin cậy:

- Kiểm tra tên thực thể sử dụng có khớp với tên đăng ký trong chứng thư số
- Kiểm tra hạn sử dụng của chứng thư số
- Kiểm tra tính tin cậy của CA phát hành chứng thư số
- Kiểm tra trạng thái thu hồi chứng thư số
- Kiểm tra chữ ký trên chứng thư số để đảm bảo chứng thư không bị sửa đổi, làm giả

61

61

Những sai lầm khi sử dụng mật mã

- Lỗ hổng trên HĐH Android được phát hiện vào năm 2013 cho thấy quá trình sinh khóa không đủ ngẫu nhiên
  - Các ứng dụng sử dụng cơ chế mã hóa bị ảnh hưởng, trong đó có các ứng dụng sử dụng Bitcoin để thanh toán
- Lỗ hổng trên Chromebooks: sinh giá trị ngẫu nhiên chỉ có 32 bit thay vì 256 bit
- Coi mật mã là giải pháp vạn năng (những bài sau chúng ta sẽ phân tích kỹ hơn)
- Sửa đổi/Thêm một vài yếu tố bí mật vào giải thuật, hệ mật mã sẽ an toàn hơn
- Sử dụng các hàm ngẫu nhiên của ngôn ngữ lập trình

62

62

Những sai lầm khi sử dụng mật mã

- Không thay đổi giá trị IV(Initial Vector)
- Sử dụng chế độ mã từ điển (ECB)
- Case study: Lỗi sử dụng mật mã trong các ứng dụng Android (2013)
  - Phân tích 11.748 ứng dụng

	# apps	violated rule
48%	5,656	Uses ECB (BouncyCastle default) (R1)
31%	3,644	Uses constant symmetric key (R3)
17%	2,000	Uses ECB (Explicit use) (R1)
16%	1,932	Uses constant IV (R2)
	1,636	Used iteration count < 1,000 for PBE(R5)
14%	1,629	Seeds SecureRandom with static (R6)
	1,574	Uses static salt for PBE (R4)
12%	1,421	No violation

63

63

Một số lưu ý khác

- Chỉ sử dụng thuật toán chuẩn và các thư viện lập trình được phê chuẩn: OpenSSL, Bouncy Castle, Libgcrypt, RSA BSAFE, wolfCrypt
- Nếu có thể, sử dụng các thuật toán mạnh nhất
- Nếu phải sinh khóa từ một giá trị cho trước, sử dụng hàm PBKDF2()
- Đừng tự thiết kế hệ mật mã cho riêng mình:
  - Nếu không thể sử dụng các hệ mật mã đã có, hãy xem lại hệ thống
  - Nếu bắt buộc phải sử dụng hệ mật mã mới hoàn toàn, hãy đánh giá một cách cẩn thận

64

64