BÀI 2.
CÁC HỆ MẬT MÃ

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

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

1. MẬT MÃ LÀ GÌ?

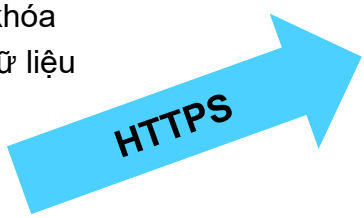
Bùi Trọng Tùng,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

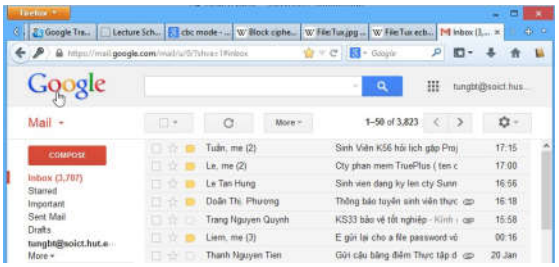
1.1. Khái niệm mật mã

- Mã hóa (code): biến đổi cách thức biểu diễn thông tin
- Mật mã (cipher): mã hóa để che giấu, giữ mật thông tin
- 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 AT-ANTT
  - Nhưng không phải là công cụ vạn năng
- Trong học phần này, chỉ đề cập đến khái niệm cơ bản và cách thức sử dụng các phương pháp mật mã

Truyền tin bí mật

- Bước 1: Trao đổi khóa
- Bước 2: Mã hóa dữ liệu





5

5

Lưu trữ thông tin mật

Thiết bị lưu trữ

Alice “hôm nay” truyền tin bí mật cho Alice “ngày mai”

6

6

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)

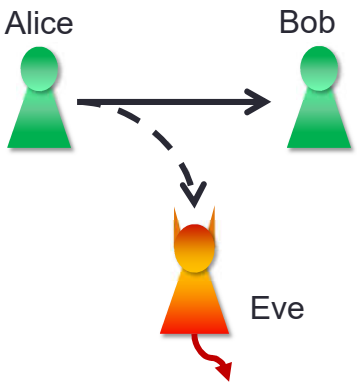
Mã hóa: C = E(k, m)

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



7

7

Ứng dụng của mật mã

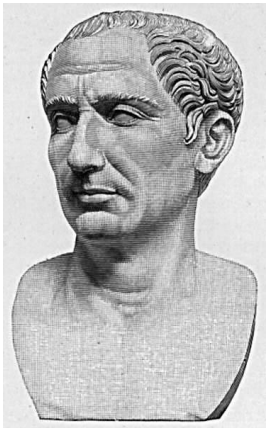
- Giữ bí mật cho thông tin,
- …và không chỉ vậy…
- Chữ ký số(Digital Signature)
- Liên lạc ẩn danh (Anonymous Communication)
- Tiền ẩn danh (Anonymous digital cash)
- Bầu cử điện tử (E-voting)

8

8

Một ví dụ - Mật mã Caesar

- Julius Caesar đưa ra vào thế kỷ thứ 1 trước CN, sử dụng trong quan sự
- Ý tưởng: thay thế một ký tự (bản rõ) trong bảng chữ cái bằng ký tự (bản mật) đứng sau nó 3 (khóa) vị trí.
  - Sử dụng bảng chữ cái vòng
  - A → D, B → E, C → F,..., X → A, Y → B, Z → C
- Mô hình hóa bằng toán học:
  - Khóa k = 3
  - Mã hóa: c = (m + 3) mod 26
  - Giải mã: m = (c − 3) mod 26
- Dễ dàng bị phá ngay cả khi K thay đổi các giá trị khác



Gaius Julius Caesar

9

9

Lịch sử phát triển của mật mã học

- Năm 300 TCN, Euclid phát hiện ra số nguyên tố, thuật toán tìm UCLN của 2 số
- Mật mã Hy Lạp

- Năm 1640 ra đời định lý Fermat nhỏ:

$$a^{p-1} = 1\ (mod\ p)\ \forall\ p\ là\ số\ nguyên\ tố, 1 \leq a < p$$

a^{p-1} và p là 2 số nguyên tố cùng nhau

10

10

Lịch sử phát triển của mật mã học

- Năm 1798, Gauss tiên đoán về sự quan trọng của việc phân tích hợp số thành các thừa số nguyên tố
- Năm 1874, William Stanley Jevons (Anh) đưa ra lời thách thức phân tích hợp số 8616460799.
  - Năm 1903 Derrick Lehmer (Mỹ) có đáp án

11

11

Lịch sử phát triển của mật mã học

- Năm 1917, Vernam cipher đưa ra ý tưởng mật mã one-time-pad sử dụng phép XOR nhưng chưa được chú ý
- Chiến tranh TG lần 1: sử dụng các biện pháp can nhiễu sóng radio khi trao đổi thông tin
- Chiến tranh thế giới lần 2: máy Enigma được quân phát xít sử dụng
  - Bị phá mã bởi lực lượng đồng minh

12

12

Lịch sử phát triển của mật mã học

- Năm 1945, Claude Shannon xuất bản sách "Communication Theory of Secrecy Systems"
- Năm 1949, Claude Shannon công bố lý thuyết Shannon về mật mã hoàn hảo
- Năm 1976 mật mã DES ra đời
- Tháng 11/1976 Diffie và Hellman công bố bài báo "New Directions in Cryptography" đặt nền móng cho hệ mật mã khóa bất đối xứng
- Năm 1977, Ron Rivest, Adi Shamir, Len Adleman giới thiệu mật mã RSA
  - Fun fact: Hai nhân vật Alice và Bob được giới thiệu

13

13

1.2. Một số nguyên lý chung của các hệ mật mã

- 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?

14

14

Lý thuyết Shannon

- Hệ mật hoàn hảo: $\forall m_0, m_1$ có độ dài như nhau, $\forall c$

$$Pr[E(k, m_0) = c] = Pr[E(k, m_1) = c]$$

E: Hàm ngẫu nhiên

- 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

15

15

Lý thuyết Shannon (tiếp)

- Độ dư thừa của ngôn ngữ: Sự xuất hiện của n ký tự (n-gram) cho phép đoán nhận đúng các ký tự xuất hiện tiếp theo với xác xuất *p* nào đó.
  - Nếu *p = 0 ∀n:* ngôn ngữ không có dư thừa
  - Nếu *p > 0*: ngôn ngữ có dư thừa (một số ký tự là không cần thiết sau khi *n ký tự đã xuất hiện*)
  - Định lượng: sử dụng lý thuyết thông tin
  - Ví dụ: tiếng Việt
- Đối với thám mã: sử dụng phương pháp vét cạn, cần phải thu được tối thiểu *u* ký tự mật mã để tìm được chính xác khóa.

u: khoảng cách unicity (unicity distance)

→ *u* càng lớn độ an toàn của hệ càng cao

16

16

Lý thuyết Shannon (tiếp)

- Tính toán khoảng cách unicity

$$u = \frac{l_K H(k)}{H(c) - H(m)}$$

l_K: Kích thước khóa

$H(k), H(m), H(c)$: entropy của ký tự. Ví dụ

$H(m) = -\sum p(m_i) \times log_2(p(m_i))$: entropy của ký tự bản rõ

$p(m_i)$: xác suất xuất hiện của ký tự trong không gian bản rõ

- Nếu khóa và bản mật xuất hiện hoàn toàn ngẫu nhiên, và chung bảng chữ cái:

$$u = \frac{l_K log_2(N)}{log_2(N) - H(m)}$$

N: số ký tự của bảng chữ cái

- Làm thế nào để tăng độ an toàn khi sử dụng mật mã?

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)
- Bạn nghe ở đâu đó, "dễ dàng" bẻ khóa mật mã DES có kích thước khóa 64 bit?
  - Năm 1999, hệ thống phá mã EFF DES (trị giá 250K$) bẻ khóa DES trong khoảng 1 ngày
  - Năm 2008, hệ thống phá mã COPACOBANA (trị giá 10K$) bẻ khóa DES trong 6,4 ngày

  Sử dụng định luật Moore để tính thời gian bẻ khóa trong năm 2016 với chi phí 10K$?

Thông tin tham khảo – Kích thước khóa

- Chi phí để bẻ khóa DES (năm 2008)
  - 64 bit: $10.000
  - 87 bit: $100.000.000.000 (thời gian bẻ khóa không đổi)
- Cần giữ thông tin mật trong bao lâu khi hệ thống phá mã là COPACOBANA? (năm 2008)
  - 64 bit: 6.4 ngày
  - 128 bit: ?
- 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

19

19

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

20

20

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	

21

21

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
- 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: DES, 2DES, 3DES, AES, RC4, RC5

22

22

2.1. Sơ đồ nguyên lý

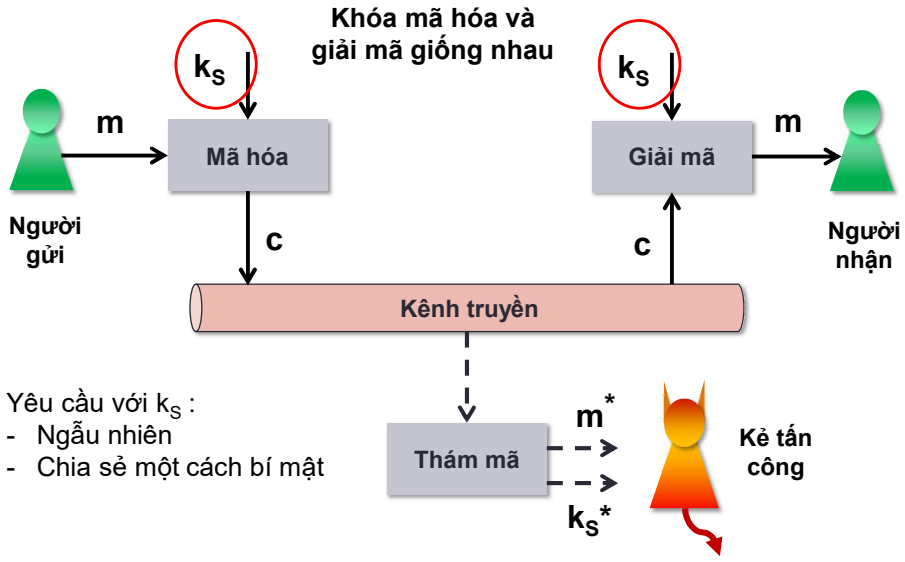
Hệ mật mã gồm:

- Bản rõ (plaintext-m): thông tin không được che dấu
- Bản mật (ciphertext-c): 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 $D(k_S, E(k_S, M)) = M$

23

23

Sơ đồ chung



Yêu cầu với k_S :
- Ngẫu nhiên
- Chia sẻ một cách bí mật

24

24

Thám mã

- Nhắc lại 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"
  - Kẻ thám mã đã biết giải thuật mã hóa, giải mã
- Tấn công chỉ biết bản mật:
  - Kẻ thám mã có các bản mật (ciphertext-only attack)
  - Phương pháp phá mã: thử tất cả các tổ hợp khóa có thể để tìm ra tổ hợp khóa thích hợp. Trong trường hợp không gian khóa lớn thì phương pháp này không thực hiện được.
  - Đối phương cần phải phân tích văn bản mật, thực hiện các kiểm nghiệm thống kê để giảm số lượng trường hợp cần thử.

25

25

Thám mã (tiếp)

- Tấn công đã biết bản rõ (known-plaintext attack):
  - Kẻ thám mã đã có một số cặp (m,c) của những phiên truyền tin trước đó. Mục đích: đoán khóa mật k.
  - Phương pháp tấn công: phân tích thuộc tính thống kê của ngôn ngữ trên văn bản gốc
- Tấn công chọn trước bản rõ (chosen-plaintext attack): kẻ thám mã lừa người gửi mã hóa một số bản tin đặc biệt do hắn chọn
- Tấn công chọn trước bản mật(chosen-ciphertext attack): kẻ thám mã lừa người nhận giải mã một số bản tin đặc biệt do hắn chọn
- Tấn công chọn trước bản rõ, bản mật
  - Thuật toán được thiết kế để chống lại dạng tấn công này

26

26

2.2. MẬT MÃ CỔ ĐIỂN

27

27

Mật mã thay thế(Substitution cipher)

- Một/một mẫu ký tự được thay thế bằng một/một mẫu ký tự khác.
- Mật mã Ceasar
- Mật mã dịch vòng (Shift Cipher): mã từng ký tự
  - Khóa: $1 \le k \le 25$
  - Mã hóa: $c = (m + k) \bmod 26$
  - Giải mã: $m = (c - k) \bmod 26$

28

28

Mật mã thay thế(Substitution cipher)

- Mật mã Vigener: mã 1 khối ký tự

k = C R Y P T O C R Y P T O C R Y P T

(+ mod 26)

m = W H A T A N I C E D A Y T O D A Y

---

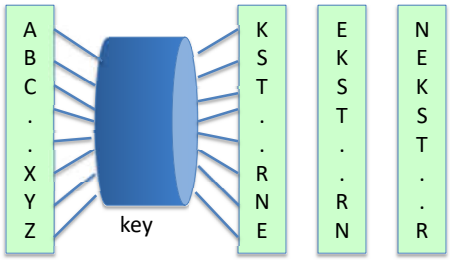
c = Z Z Z J U C | L U D T U N | W G C Q S

29

29

Mật mã thay thế(Substitution cipher)

- Máy rotor (Rotor machine)



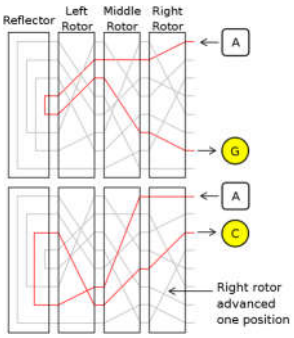
Hebern machine

30

30

Mật mã thay thế(Substitution cipher)

- Máy rotor (Rotor machine)



Số lượng khóa?

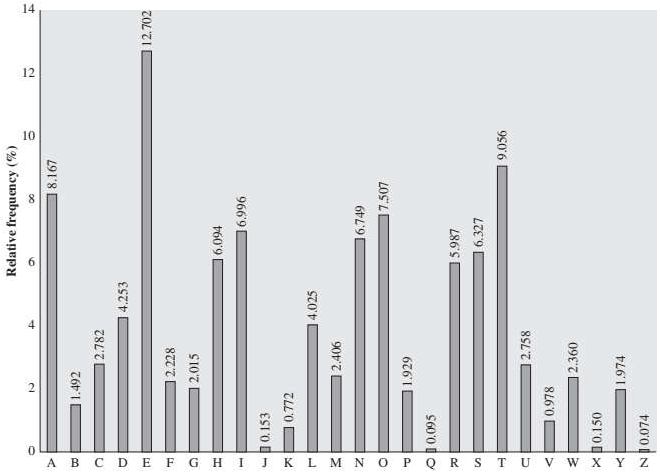
Enigma

31

31

Phá mã hệ mật mã thay thế

- Chỉ có bản mã: Dựa trên phương pháp thống kê
- Ví dụ: tiếng Anh



32

32

Thuộc tính thống kê của tiếng Anh

- Phân nhóm ký tự theo tần suất

I	e
II	t,a,o,i,n,s,h,r
III	d,l
IV	c,u,m,w,f,g,y,p,b
V	v,k,j,x,q,z

- Một vài mẫu ký tự có tần suất xuất hiện cao
  - Bigrams: th, he, in, an, re, ed, on, es, st, en at, to
  - Trigrams: the, ing, and, hex, ent, tha, nth, was eth, for, dth

33

33

Ví dụ: Phá mã dịch vòng

```
YKHLBA JCZ SVIJ JZB TZVHI JCZ VHJ DR IZXKHLBA VSS
RDHEI DR YVJV LBXSKYLBA YLALJVS IFZZXC CVI
LEFHDNZY EVBLRDSY JCZ FHLEVHT HZVIDB RDH JCLI CVI
WZZB JCZ VYNZBJ DR ELXHDZSZXJHDBLXI JCZ XDEFSZQLJT
DR JCZ RKBXJLDBI JCVJ XVB BDP WZ FZHRDHEZY WT JCZ
EVXCLBZ CVI HLIZB YHVEVJLXVSST VI V HXXIKSJ DR
JCLI HZXZBJ YZNZXDFEZBJ LB JZXCBDSDAT EVBT DR JCZ
XLFCZH ITIJZEIJCVJ PZHZ DBXZ XDBILYXHZYIZKHZ
VHZBDP WHZVMVWSZ
```

34

34

Ví dụ: Phá mã dịch vòng

Ký tự:	*A*	*B*	*C*	*D*	*E*	*F*	*G*
Tần suất:	*5*	*24*	*19*	*23*	*12*	*7*	*0*
Ký tự:	*H*	*I*	*J*	*K*	*L*	*M*	*N*
Tần suất:	*24*	*21*	*29*	*6*	*21*	*1*	*3*
Ký tự:	*O*	*P*	*Q*	*R*	*S*	*T*	*U*
Tần suất:	*0*	*3*	*1*	*11*	*14*	*8*	*0*
Ký tự:	*V*	*W*	*X*	*Y*	*Z*		
Tần suất:	*27*	*5*	*17*	*12*	*45*		

Z → e
f_J=29, f_V = 27
f_{JCZ} = 8
⇒J → t, C → h
V đứng riêng: V → a

Nhóm: {J, V, B, H, D, I, L, C} → {t, a, o, i, n, s, h, r}
t a h
JZB → te? {teo, tei, ten, tes, ter}: B → n

35

35

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKHLnA the SaIt ten TeaHI the aHt DR IeXKHLnA aSS
RDHEI DR Yata LnXSKYLnA YLALtaS IFeeXh haI
LEFHDNeY EanLRDSY the FHLEaHT HeaIDn RDH thLI haI
Ween the aYNent DR ELXHDeSeXtHDnLXI the XDEFSeQLtT
DR the RKnXtLDnI that Xan nDP We FeHRDHEeY WT the
EaXhLne haI HLIen YHaEatLXaSST aI a HXXIKSt DR
thLI HeXent YeNeXDFEent Ln teXhnDSDAT EanT DR the
XLFheH ITIteEIthat PeHe DnXe XDnILYXHeYIeKHe
aHenDP WHeaMaWSe
```

Nhóm: {J, V, B, H, D, I, L, C} → {t, a, o, i, n, s, h, r}
t a n h
aI → a? {ao, ai, as, ar}: I → s

36

36

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKHLnA the Sast ten TeaHs the aHt DR seXKHLnA aSS
RDHEs DR Yata LnXSKYLnA YLALtaS sFeeXh has
LEFHDNeY EanLRDSY the FHLEaHT HeasDn RDH thLs has
Ween the aYNent DR ELXHDeSeXtHDnLXs the XDEFSeQLtT
DR the RKnXtLDns that Xan nDP We FeHRDHEeY WT the
EaXhLne has HLsen YHaEatLXaSST as a HXXsKSt DR
thLs HeXent YeNeXDFEent Ln teXhnDSDAT EanT DR the
XLFheH sTsteEsthat PeHe DnXe XDnsLYXHeYseKHe
aHenDP WHeaMaWSe
```

Nhóm: {J, V, B, H, D, I, L, C} → {t, a, o, i, n, s, h, r}
t a n s h
Rút gọn: {H, D, L} → {o, i, r}
thLs = th?s {thos, this, thrs}: L → i

37

37

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKHinA the Sast ten TeaHs the aHt DR seXKHinA aSS
RDHEs DR Yata inXSKYinA YiAitaS sFeeXh has
iEFHDNeY EaniRDSY the FHiEaHT HeasDn RDH this has
Ween the aYNent DR EiXHDeSeXtHDniXs the XDEFSeQitT
DR the RKnXtiDns that Xan nDP We FeHRDHEeY WT the
EaXhine has Hisen YHaEatiXaSST as a HXXsKSt DR
this HeXent YeNeXDFEent in teXhnDSDAT EanT DR the
XiFheH sTsteEsthat PeHe DnXe XDnsiYXHeYseKHe
aHenDP WHeaMaWSe
```

Nhóm: {H, D} → {o, r}
aHt = a?t {aot, art}: H → r, D → o

38

38

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKrinA the Sast ten Tears the art oR seXKrinA aSS
RorEs oR Yata inXSKYinA YiAitaS sFeeXh has
iEFroNeY EaniRoSY the FriEarT reason Ror this has
Ween the aYNent oR EiXroeSeXtroniXs the XoEFSeQitT
oR the RKnXtions that Xan noP We FerRorEeY WT the
EaXhine has risen YraEatiXaSST as a rXXsKSt oR
this reXent YeNeXoFEent in teXhnoSoAT EanT oR the
XiFher sTsteEsthat Pere onXe XonsiYXreYseKre
arenoP WreaMaWSe
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	n	h	o				r	s	t		i										a				e

reason Ror this has Ween → reason for this has been
this reXent → this recent
R → f, W → b, X → c

39

39

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKrinA the Sast ten Tears the art of secKrinA aSS
forEs of Yata incSKYinA YiAitaS sFeech has
iEFroNeY EanifoSY the FriEarT reason for this has
been the aYNent of EicroeSectronics the coEFSeQitT
of the fKnctions that can noP be FerforEeY bT the
Eachine has risen YraEaticaSST as a rccsKSt of
this recent YeNecoFEent in technoSoAT EanT of the
ciFher sTsteEsthat Pere once consiYcreYseKre
arenoP breaMabSe
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	n	h	o				r	s	t		i						f				a	b	c		e

of the fKnctions → of the functions
of the ciFher → of the cipher
K → u, F → p

40

40

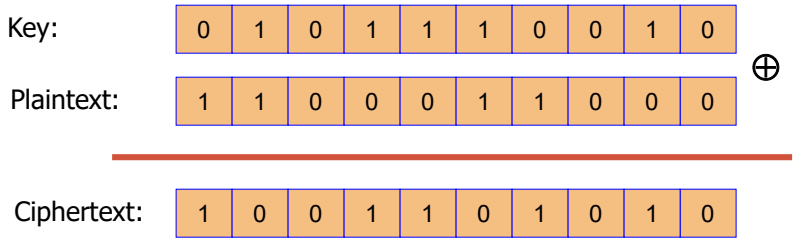
2.3. MẬT MÃ HIỆN ĐẠI

41

41

Mật mã one-time-pad (OTP)

- Vernam (1917)



- Kích thước của khóa bằng kích thước của bản rõ
- Khóa chỉ dùng 1 lần
- Shannon : mật mã OTP là hệ mật hoàn hảo. Tuy nhiên, mã Vernam không khả thi trên thực tế(Tại sao?)

42

42

Mật mã OTP

- Nếu khóa được dùng nhiều hơn 1 lần → mật mã two-time-pad không còn an toàn (Tại sao?)

$$c_1 \leftarrow m_1 \oplus k$$

$$c_2 \leftarrow m_2 \oplus k$$

Nếu kẻ tấn công có được bản mã:

$$c_1 \oplus c_2 \rightarrow m_1 \oplus m_2$$

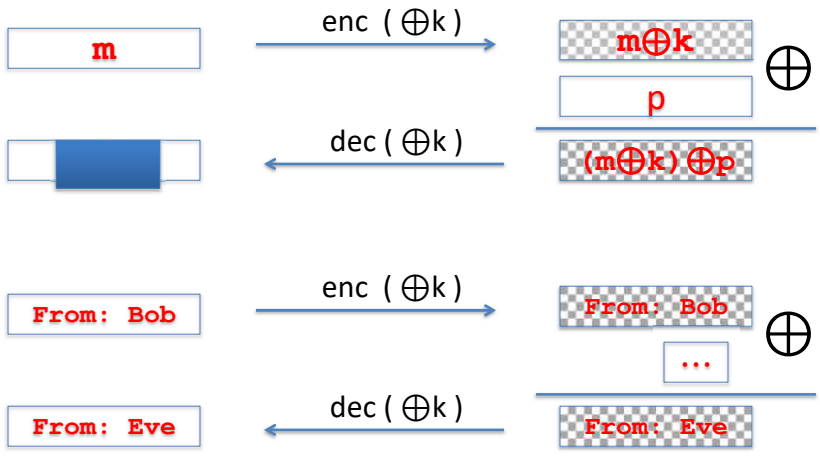
Nếu kích thước bản tin đủ dài

$$m_1 \oplus m_2 \rightarrow m_1 , m_2$$

43

43

Tấn công vào tính toàn vẹn của OTP



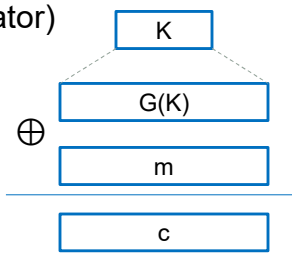
44

44

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 chỉ dùng 1 lần (one-time-pad)
- Trên thực tế, sử dụng hàm sinh khóa giả ngẫu nhiên (PRG - Pseudo Random Generator)

  G: $\{0, 1\}^s \rightarrow \{0, 1\}^n$ ($s << n$)

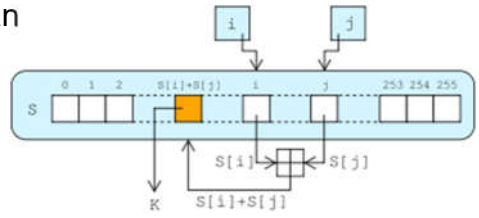


45

45

Mã RC4 (Rivest Cipher 4)

- Rivest Cipher 4: ra đời năm 1987
- Kích thước khóa: 40 đến 128 bit
- Hoạt động: gồm 2 thuật toán chính
  - Key-scheduling algorithm (KSA): mở rộng khóa mã hóa thành 1 giá trị S có kích thước 256 byte
  - Pseudo-random generation algorithm (PRGA): lựa chọn 1 byte K từ S để XOR 1 byte thông điệp
- Hiện không còn an toàn



46

46

Mã eStream

- Phương pháp mật mã dòng mới nhất được thiết kế để thay thế cho các phương pháp mã dòng cũ
- Hiện đang được phát triển, chưa công bố thành tiêu chuẩn
- Hàm sinh khóa giả ngẫu nhiên:

$$\text{PRG:} \quad \{0,1\}^s \times R \longrightarrow \{0,1\}^n$$

R: giá trị chỉ dùng 1 lần, không lặp lại

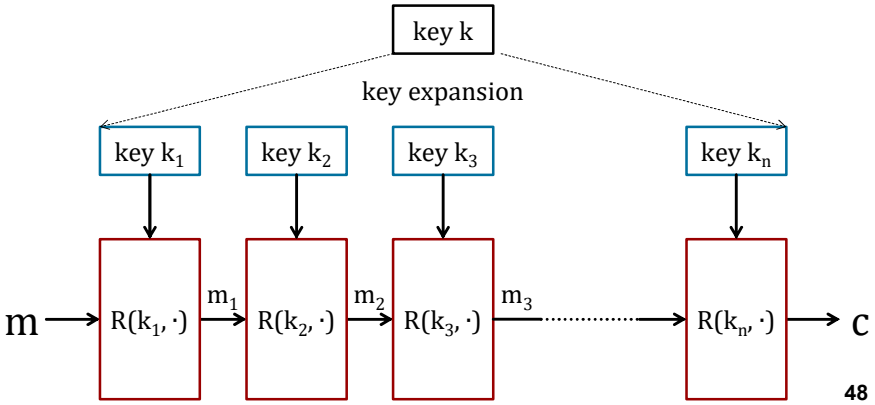
- Mã hóa: $E(k, m ; r) = m \oplus PRG(k ; r)$
- Ví dụ: Salsa20 có s = 128 hoặc 256 bit, R có kích thước 64 bit

47

47

Mật mã khối (Block Cipher)

- Chia văn bản gốc thành các khối có kích thước như nhau
- Xử lý mã hóa và giải mã từng khối
- Nguyên lý chung: sử dụng các hàm lặp $R(k_i, \cdot)$

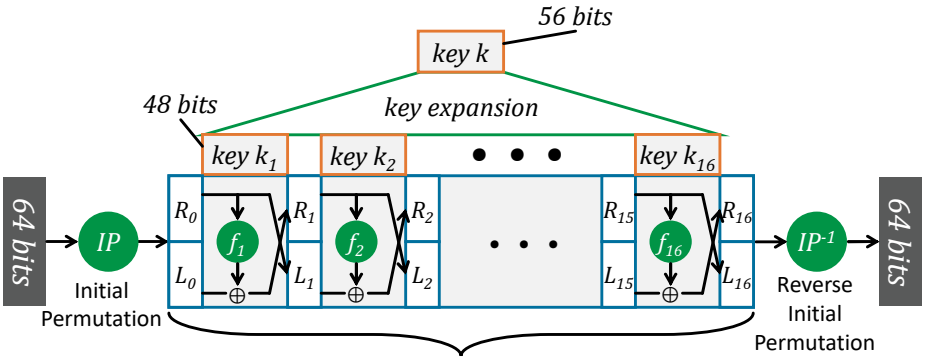


48

48

Mật mã DES - Data Encryption Standard

- Kích thước khóa: 56 bit
- Kích thước khối dữ liệu: 64 bit
- Không còn an toàn để sử dụng

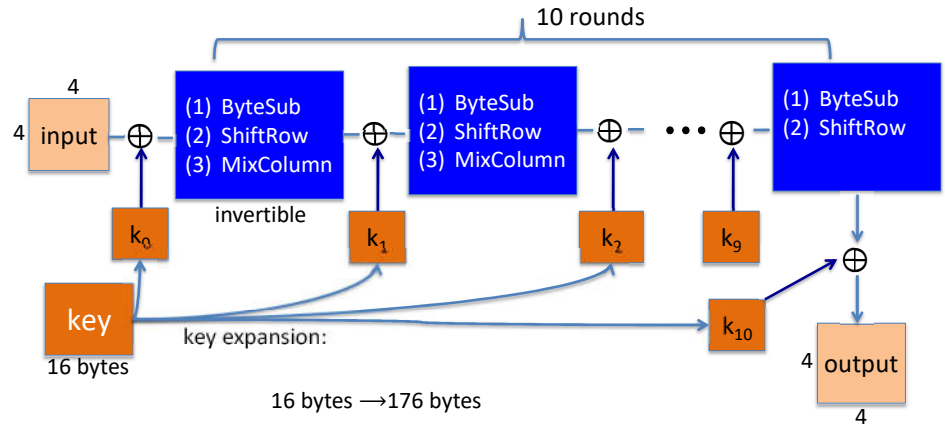


Mạng Fiestel có 16 vòng lặp

49

Mật mã AES – Advanced Encryption Standard

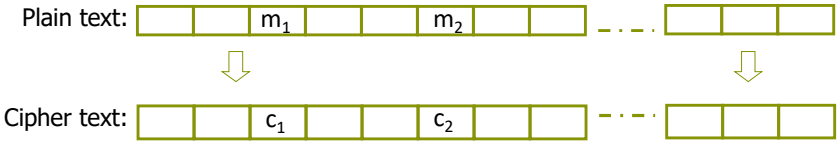
- Kích thước khóa: 128, 192, 256 bit
- Kích thước khối: 128 bit



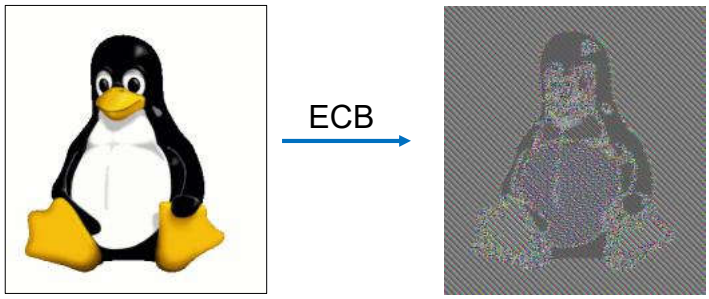
50

Các chế độ mã khối

- Electronic Code Book (ECB): Mã từ điển



- Hạn chế

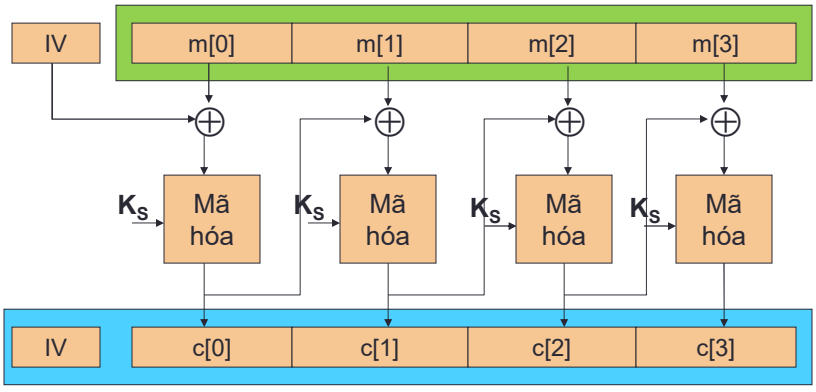


51

51

Chế độ CBC - Cipher Block Chaining

- Chế độ mã móc xích



IV (Initial Vector) cần phải ngẫu nhiên và dùng 1 lần (nonce)

52

52

Một ví dụ về sử dụng hàm mã hóa(OpenSSL)

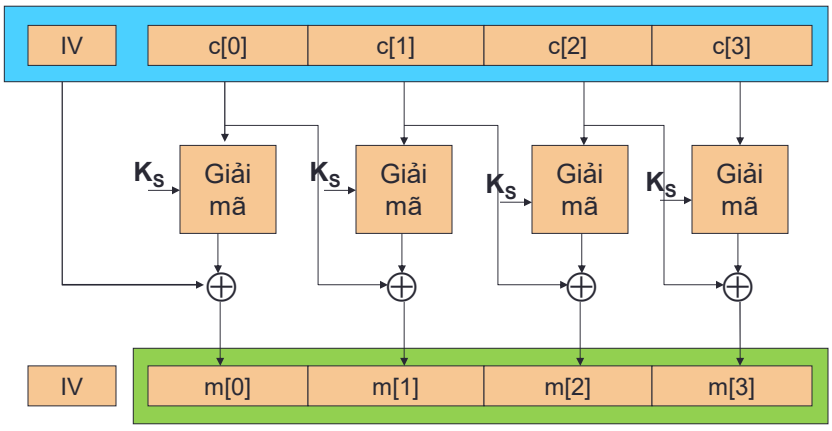
```
void AES_cbc_encrypt(
      const unsigned char *in,
      unsigned char *out,
      size_t length,
      const AES_KEY *key,
      unsigned char *ivec,
      AES_ENCRYPT or AES_DECRYPT);
```

- Nếu `ivec` không ngẫu nhiên, cần phải được mã hóa trước khi sử dụng

53

53

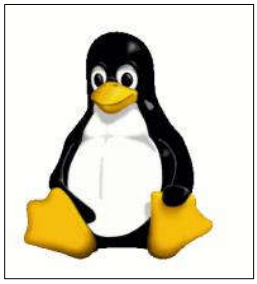
CBC – Giải mã



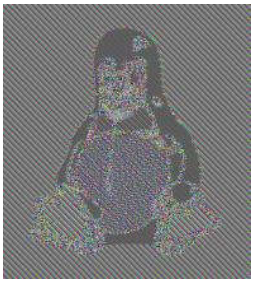
54

54

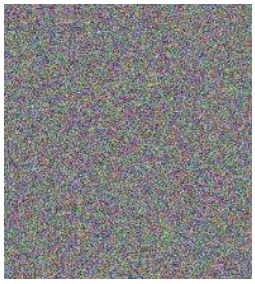
CBC – So sánh với ECB



Ảnh gốc



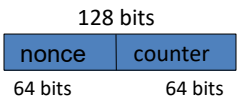
Mã hóa ở chế độ ECB



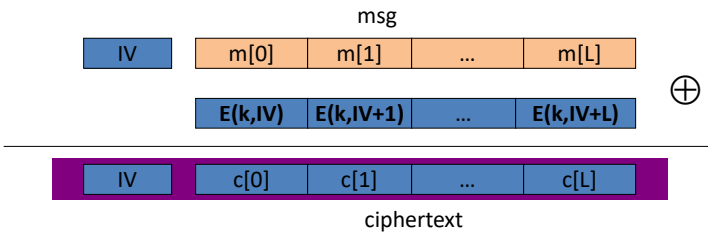
Mã hóa ở chế độ CBC

Chế độ CTR – Counter Mode

- Initial Vector: 2 phương pháp sử dụng
  - Giá trị ngẫu nhiên
  - Sử dụng giá trị dùng 1 lần (nonce)



- Mã hóa



Tấn công vào mật mã khối

- Tấn công vét cạn (Exhaustive Search): Kẻ tấn công thử mọi giá trị khóa k khi có được một vài cặp (m_i, c_i)
  - DES: Với 2 cặp, xác suất tìm được đúng khóa k là ~ $1 - 1/2^{71}$ với thời gian vét cạn 2^{56} giá trị
  - AES-128: Với 2 cặp, xác suất tìm được đúng khóa k là ~ $1 - 1/2^{128}$ với thời gian vét cạn 2^{128} giá trị
  - Sử dụng tính toán lượng tử: thời gian vét cạn còn $T^{1/2}$ → sử dụng AES-256

1976	DES adopted as federal standard		
1997	Distributed search	3 months	
1998	EFF deep crack	3 days	$250,000
1999	Distributed search	22 hours	
2006	COPACOBANA (120 FPGAs)	7 days	$10,000

57

57

Tấn công vào mật mã khối

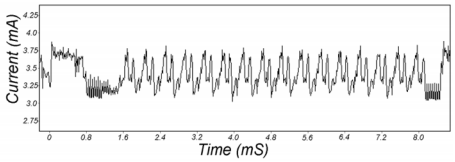
- Tấn công vét cạn (Exhaustive Search): Kẻ tấn công thử mọi giá trị khóa k khi có được một vài cặp (m_i, c_i)
  - DES: Với 2 cặp, xác suất tìm được đúng khóa k là ~ $1 - 1/2^{71}$ với thời gian vét cạn 2^{56} giá trị
  - AES-128: Với 2 cặp, xác suất tìm được đúng khóa k là ~ $1 - 1/2^{128}$ với thời gian vét cạn 2^{128} giá trị
  - Sử dụng tính toán lượng tử: thời gian vét cạn còn $T^{1/2}$ → sử dụng AES-256
- Tấn công tuyến tính (Linear Attack): Kẻ tấn công tính toán khóa k khi có rất nhiều cặp (m_i, c_i)
  - DES: Với 2^{42} cặp có thể tìm thấy khóa K trong thời gian 2^{43}
  - AES-256: Với 2^{99} cặp có thể tìm thấy khóa K trong thời gian 2^{99}

58

58

Tấn công vào mật mã khối

- Tấn công kênh bên (side-channel attack): phán đoán giá trị các bit khóa bằng cách ước lượng thời gian, lượng điện năng tiêu thụ, bức xạ điện từ… khi mã hóa, giải mã
  - Ví dụ: phương pháp của Kocher và Jaffe năm 1998



- Tấn công dựa vào lỗi (Fault attacks): lỗi xảy ra ở vòng lặp cuối cùng sẽ làm lộ thông tin về khóa

59

59

2.4. 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)

60

60

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, Eliptic Curve

61

61

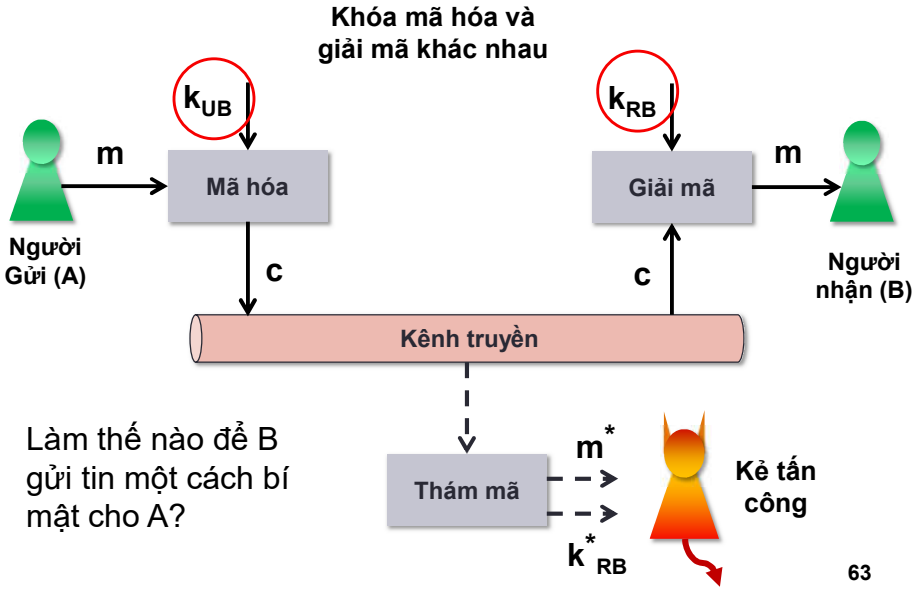
Sơ đồ nguyên lý

- Hệ mật mã gồm:
- Bản rõ (plaintext-m): thông tin không được che dấu
- Bản mật (ciphertext-c): thông tin được che dấu
- 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): $C = E(k_{UB}, m)$
  - Là hàm ngẫu nhiên
- Giải mã (decrypt): $m = D(k_{RB}, c)$
  - Là hàm xác định
- Tính đúng đắn: $D(k_{RB}, E(k_{UB}, m)) = m$

62

62

Sơ đồ nguyên lý (tiếp)



63

63

Một ví dụ - Hệ mật RSA

- Sinh khóa:
  - Chọn p,q là hai số nguyên tố
  - Tính n = p × q , Φ(n) = (p-1)×(q-1)
  - Chọn e sao cho UCLN(Φ(n), e) = 1 ;1< e < Φ(n)
  - Tính d sao cho (e × d) mod Φ(n) =1.
  - Khóa công khai : k_U = (e,n)
  - Khóa riêng : k_R = (d,n)
- Mã hóa : $c = m^e \bmod n$
- Giải mã: $m = c^d \bmod n$

64

64

Một ví dụ - Hệ mật RSA

- Sinh khóa:
  - Chọn p = 5, q = 11
  - Tính n = p × q = 55, Φ(n) =(p-1)×(q-1) = 40
  - Chọn e sao cho UCLN(Φ(n), e) = 1 và 1 < e < Φ(n)

    VD: e = 7
  - Tính d sao cho (e × d) mod Φ(n) = 1, 1 < d < Φ(n)

    d = 23

    Cặp khóa : k_U = (7,55), k_R = (23,55)
- Mã hóa: m = 6 → c = 41
- Giải mã: c = 41 → m = 6

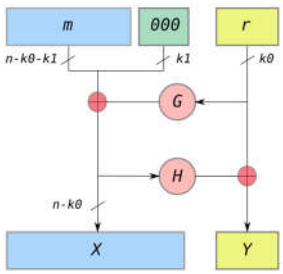
Nếu kẻ tấn công có k_U, làm thế nào để tính k_R?

65

65

RSA-OEAP

- Optimal Asymetric Encryption Padding
- Nếu bản tin m được mã 2 lần với cùng khóa k thì nội dung bản mã không thay đổi → không đảm bảo yêu cầu tính ngẫu nhiên của hàm mã
- RSA-OEAP: sử dụng thêm khối đệm(padding) và giá trị ngẫu nhiên trong quá trình mã hóa
- Xử lý bản m trước khi mã hóa:
  - R: giá trị ngẫu nhiên
  - G, H: hàm băm
- Mã hóa: E(X||Y, .)



66

66

Độ an toàn của RSA

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

67

67

Tấn công vào RSA

- Tấn công kênh bên: quan sát quá trình giải mã
  - Phân tích thời gian [Kocher et al. 1997]: quá trình giải mã có thể lộ thông tin về khóa riêng
  - Phân tích mức độ tiêu thụ năng lượng [Kocher et al. 1999]
  - Phân tích tiếng ồn phát ra từ CPU [Daniel Genkin et al. 2013]
- Tấn công dựa vào lỗi tính toán
- Tấn công do sinh khóa không ngẫu nhiên:
  - Giả sử quá trình sinh khóa sử dụng p1 = p2 nhưng q1 ≠ q2 → UCLN(N_1, N_2) = p
  - Thực tế: 0.4% số lần sinh khóa ra trong giao thức HTTPS gặp lỗi trên

```
x = C
for j = 1 to n
  x = mod(x2, N)
  if dj == 1 then
    x = mod(xC, N)
  end if
return x
```

68

68

3.3. 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...

69

69

3.3. 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

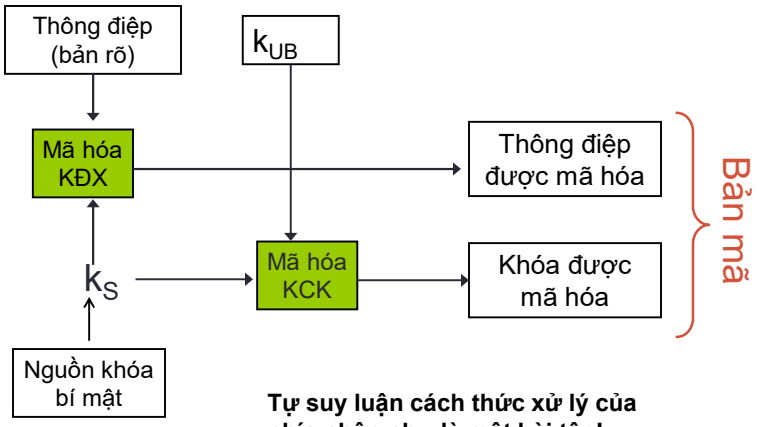
  → Kết hợp 2 hệ mật mã

70

70

Sơ đồ "lai"

- Phía gửi



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

71

71

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, tTrong 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
- 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

72

72

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
- 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
- Mật mã chưa đáp ứng yêu cầu về toàn vẹn
  - Khi sử dụng mật mã hãy thêm vào các sơ đồ đáp ứng toàn vẹn nội dung thông tin và xác thực nguồn gốc thông tin (sẽ đề cập đến trong những bài sau)

73