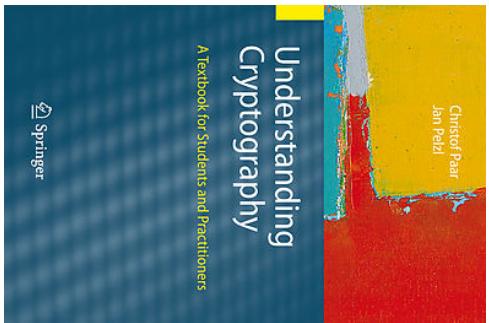


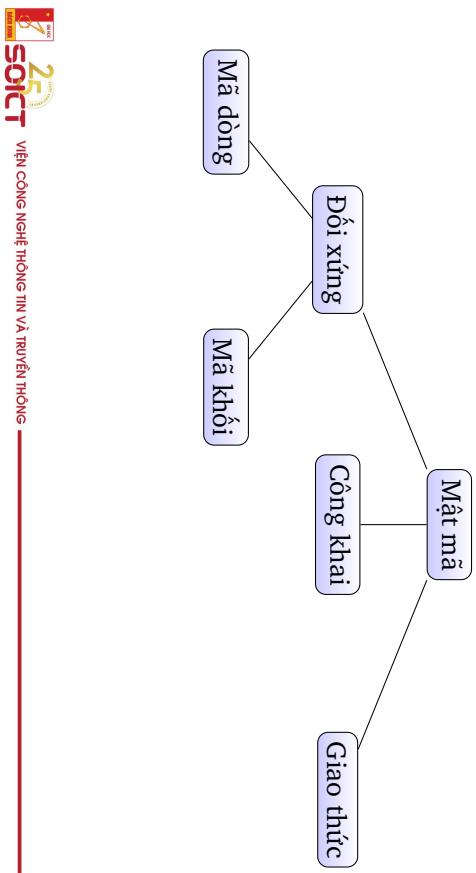
Mật mã

<https://www.crypto-textbook.com>

Tài liệu



2 / 30



4 / 30

Nội dung

- ① Giới thiệu
- ② Mã hóa và giải mã với hệ mã dòng
- ③ Bộ sinh số ngẫu nhiên
- ④ One Time Pad (OTP)
- ⑤ Hệ mã dòng trong thực tế

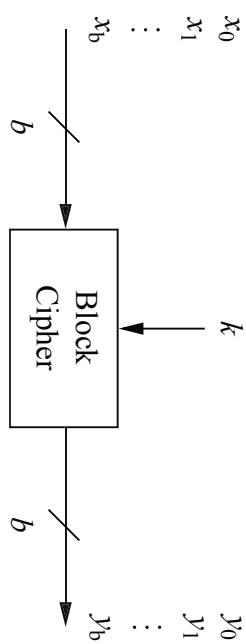
Nhập môn An toàn Thông tin

Mã dòng



1 / 30

Mã khối



Các hệ mã **khối** mã hoá mỗi lần cả một khối của bản rõ.

Ví dụ

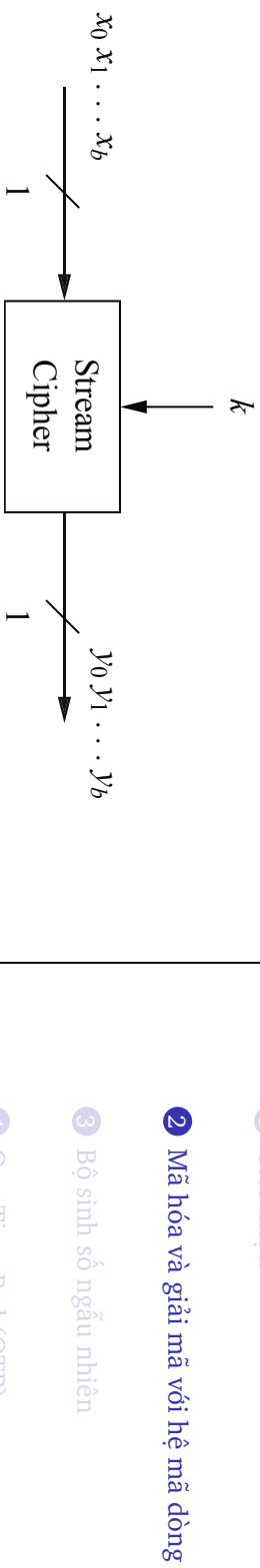
- Kích thước khối của AES (Advanced encryption standard) là 128 bit;
- DES (Data encryption standard) hoặc 3DES là 64 bit.

Mã dòng

Định nghĩa (Mã hóa và giải mã)
Dòng bản rõ, bản mã và khóa là các bit riêng rẽ. Cụ thể
 $x_i, y_i, s_i \in \{0, 1\}$.

- Mã hóa: $y_i = \text{Enc}(s_i, x_i) = x_i \oplus s_i$.
- Giải mã: $x_i = \text{Dec}(s_i, y_i) = y_i \oplus s_i$.

Mã dòng



Nội dung

Các hệ mã **dòng** mã hoá từng bit riêng rẽ.

1 Giới thiệu

2 Mã hóa và giải mã với hệ mã dòng

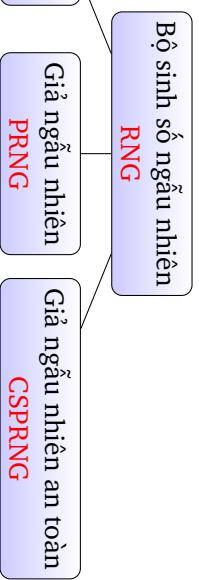
3 Bộ sinh số ngẫu nhiên

4 One Time Pad (OTP)

5 Hệ mã dòng trong thực tế

Khoá của mã dòng

Ba kiểu số ngẫu nhiên



- **Hỏi:** Làm thế nào để sinh dòng bit khoá s_i ?
- **Trả lời:** Liên quan đến việc sinh dãy số **ngẫu nhiên**.

Thuật ngữ:

- Random Number Generator
- True RNG
- Pseudo RNG
- Cryptographically Secure PRNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

10/30

- **Hỏi:** Làm thế nào để sinh dòng bit khoá s_i ?
- **Trả lời:** Liên quan đến việc sinh dãy số **ngẫu nhiên**.

- **Hỏi:** Làm thế nào để sinh dòng bit khoá s_i ?
- **Trả lời:** Liên quan đến việc sinh dãy số **ngẫu nhiên**.

Thuật ngữ:

- Random Number Generator
- True RNG
- Pseudo RNG
- Cryptographically Secure PRNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

12/30

Ví dụ

Alice muốn mã hoá ký tự A, biểu diễn bởi mã ASCII. Mã ASCII của 'A' là $65_{10} = 1000001_2$. Giả sử các bit đầu tiên của dòng khoá là $(s_0, \dots, s_6) = 0101100$.

Nội dung

1. Giới thiệu

1. Giới thiệu
2. Mã hóa và giải mã với hệ mã dòng

3. Bộ sinh số ngẫu nhiên

3. Bộ sinh số ngẫu nhiên
4. One Time Pad (OTP)
5. Hệ mã dòng trong thực tế

$$x_0, \dots, x_6 = 1000001 = A$$

\oplus

$$\begin{aligned} s_0, \dots, s_6 &= 0101100 \\ y_0, \dots, y_6 &= 1101101 = m \end{aligned}$$

$m=1101101$

$$y_0, \dots, y_6 = 1101101 = m$$

\oplus

$$\begin{aligned} s_0, \dots, s_6 &= 0101100 \\ x_0, \dots, x_6 &= 1000001 = A \end{aligned}$$

Sinh số giả ngẫu nhiên

Ví dụ dùng rand()

- Bộ sinh số giả ngẫu nhiên sinh dãy bằng cách tính toán từ một giá trị seed ban đầu.
- Thông thường chúng được tính theo công thức truy hồi:

$$\begin{aligned}s_0 &= \text{seed} \\ s_{i+1} &= f(s_i), \quad i = 0, 1, \dots\end{aligned}$$

- hoặc tổng quát hơn

$$s_{i+1} = f(s_i, s_{i-1}, \dots, s_{i-t})$$

với t là một giá trị cố định.



VIEN CONG NGHIEP THONG TIN VÀ TRUYEN THONG

14 / 30

Hàm `time(0)` trả về số giây tính từ thời điểm ban đầu kỷ nguyên máy tính (00:00:00 UTC, ngày 1 tháng 1 năm 1970); và dùng nó làm giá trị `seed`.



VIEN CONG NGHIEP THONG TIN VÀ TRUYEN THONG

16 / 30

Sinh số ngẫu nhiên thực sự

- Các số ngẫu nhiên thực sự phát sinh từ quá trình vật lý.
- Ví dụ:** tung đồng xu, tung xúc xắc, di chuyển chuột, thời gian ấn phím, Microphone, camera, sự thay đổi tốc độ của ô còng,...
- Trên Linux, nguồn ngẫu nhiên có thể lấy từ file `/dev/random`.

```
1 > hexdump -C -n 8 /dev/random
2
3 00000000 10 59 69 d4 dd 1e ad 66  | .yi....f |
4 00000008
5
```

Ví dụ

Hàm `rand()` trong ANSI C:

$$s_0 = 12345$$

$$s_{i+1} = 1103515245 \cdot s_i + 12345 \pmod{2^{31}}, \quad i = 0, 1, \dots$$

Nội dung

① Giới thiệu

② Mã hóa và giải mã với hệ mã dòng

③ Bộ sinh số ngẫu nhiên

④ One Time Pad (OTP)

⑤ Hệ mã dòng thực tế



20 / 30

Định nghĩa (One Time Pad)

One Time Pad (OTP) là hệ mã dòng trong đó

- ❶ dòng bit khoá s_i được sinh bằng bộ sinh số ngẫu nhiên thực sự.
- ❷ mỗi dòng bit khoá chỉ được sử dụng **một lần**.

Sinh số giả ngẫu nhiên an toàn

Bộ sinh số giả ngẫu nhiên an toàn là bộ sinh số giả ngẫu nhiên với tính chất: **Không thể dự đoán được**.

Định nghĩa (Không dự đoán được)

Cho n bit

$s_i, s_{i+1}, \dots, s_{i+n-1}$.

Không có thuật toán chạy trong thời gian đà thúc để dự đoán bit tiếp theo s_{i+n} với xác suất thành công lớn hơn 50%.

Xây dựng hệ mã “hoàn hảo”.

Mục đích

Định nghĩa (An toàn không điều kiện)

Một hệ mật là *an toàn không điều kiện* hoặc *an toàn theo lý thuyết thông tin* nếu nó không thể bị phá ngay cả khi kẻ tấn công có nguồn tài nguyên tính toán vô hạn.

Tính xác thực của OTP

Nội dung

1 Giới thiệu

- OTP có thể bị giả mạo.

Thay đổi các bit của bản mã sẽ làm cho các bit tương ứng của bản rõ bị thay đổi.

- OTP không cho phép xác thực nội dung thông điệp hay cho phép chống lại việc sửa đổi thông điệp.

4 One Time Pad (OTP)



22 / 30

5 Hệ mã dòng trong thực tế



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Nhược điểm của OTP

Người dùng phải

- sinh ra các khóa bí mật lớn,
- chia sẻ chúng an toàn,
- giữ chúng bí mật,
- tránh sử dụng lại khóa:

$$\begin{aligned}y_1 \oplus y_2 &= (x_1 \oplus s) \oplus (x_2 \oplus s) \\&= x_1 \oplus x_2\end{aligned}$$

có thể rút ra nhiều thông tin về x_1, x_2 .

Bài tập

Giả sử bạn biết mã hóa của thông điệp “attack at dawn” dùng mã hoá OTP là

6c73d5240a948cc86981bc294814d

(bản rõ ở dạng mã ASCII 8-bit và bản mã được viết ở dạng hexa).
Bản mã của thông điệp “attack at dusk” với cùng khóa OTP là
giì?

Mã dòng với PRNG

Tấn công hệ mã dựa trên LCG 2

Ví dụ (Đồng dư tuyến tính LCG)

$$S_0 = \text{seed}$$

$$S_i = A \cdot S_{i-1} + B \pmod{m}, \quad i = 0, 1, \dots$$

với m có kích thước khoảng 100 bit; và $S_i, A, B \in \{0, 1, \dots, m-1\}$.

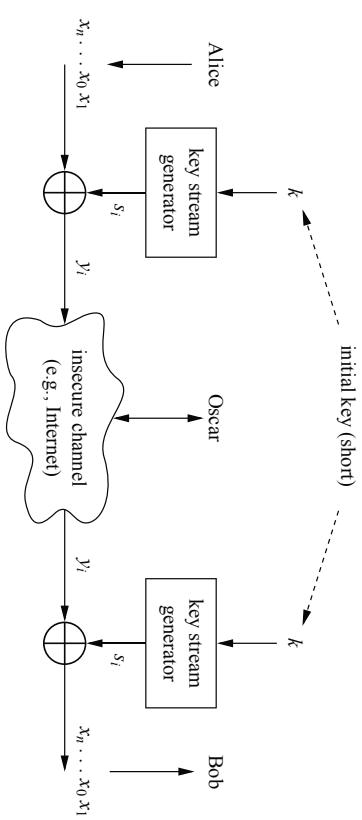
Khoá của hệ mã dòng là $k = (A, B)$.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

26 / 30

Mã dòng thực tế



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

27 / 30

Tấn công hệ mã dựa trên LCG

Giả sử Oscar biết x_1, x_2, x_3 .

- ➊ Oscar tính S_1, S_2, S_3 .
- ➋ Ta có

$$\begin{aligned} S_2 &= A \cdot S_1 + B \pmod{m} \\ S_3 &= A \cdot S_2 + B \pmod{m} \end{aligned}$$

Đây là hệ phương trình đồng dư tuyến tính với hai biến!

Cảm ơn!



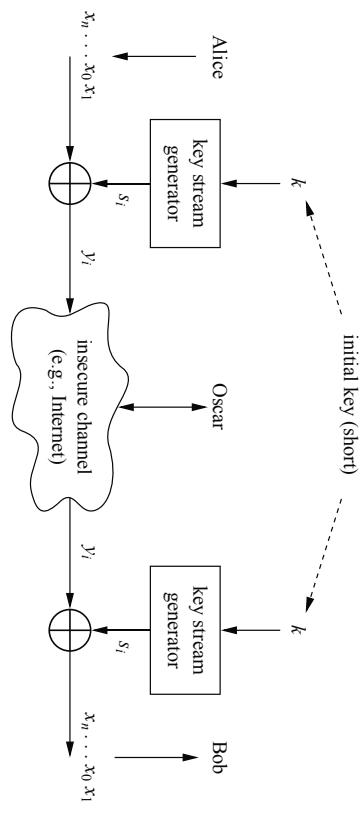
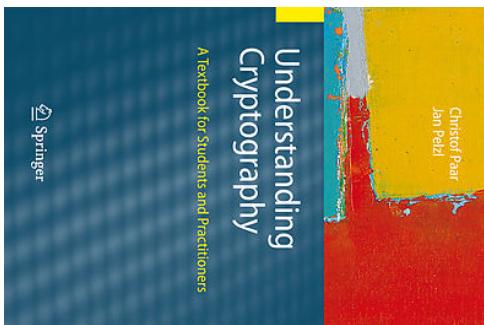
sotc.hust.edu.vn/ fb.com/groups/sotc

Không dùng LCG để sinh dòng bit cho khoá!

Mã dòng

<https://www.crypto-textbook.com>

Tài liệu



Câu hỏi: Làm thế nào sinh dãy s_i ?



2 / 34

4 / 34

Nội dung

1 Giới thiệu

Nhập môn An toàn Thông tin
Mã dòng dựa trên thanh ghi dịch

2 LFSR: Dạng tổng quát

3 Tấn công LFSR

4 Trivium: Một hệ mã dòng hiện đại

1 / 34

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1

$$s_{i+3} = s_{i+1} \oplus s_i$$

$$s_{i+3} = s_{i+1} \oplus s_i$$

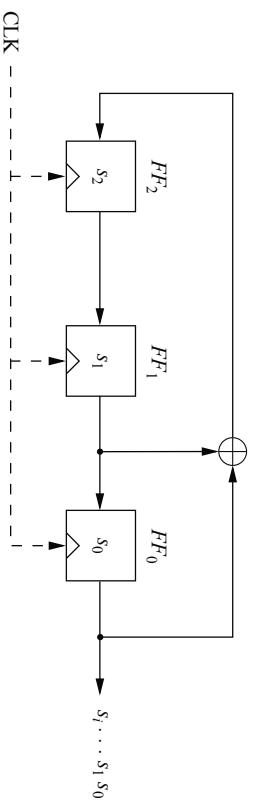


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

6 / 34

Thanh ghi dịch phản hồi tuyến tính

Linear Feedback Shift Register (LFSR)



Hình: LFSR bậc $m = 3$ với ba Flip-flops $FF_2, FF_1 \vee FF_0$

Công thức truy hồi:

$$s_{i+3} = s_{i+1} \oplus s_i.$$

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0

6 / 34

Ví dụ



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

5 / 34

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1

$$s_{i+3} = s_{i+1} \oplus s_i$$

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1

$$s_{i+3} = s_{i+1} \oplus s_i$$

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0

$$s_{i+3} = s_{i+1} \oplus s_i$$

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1

$$s_{i+3} = s_{i+1} \oplus s_i$$

Nội dung

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

1 Giới thiệu

2 LFSR: Dạng tổng quát

3 Tân công LFSR

4 Trivium: Một hệ mã dòng hiện đại

Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

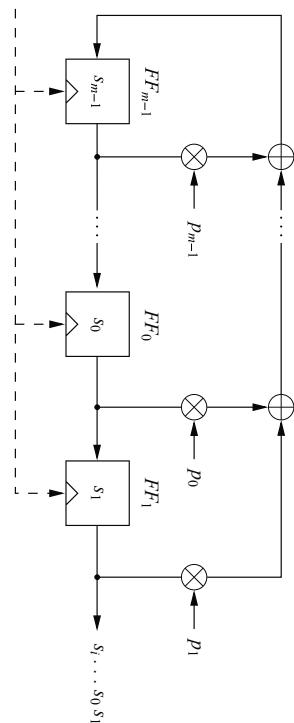
Ví dụ

clk	FF ₂	FF ₁	FF ₀ = s _i
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

Output: 0010111 0010111 0010111 ...

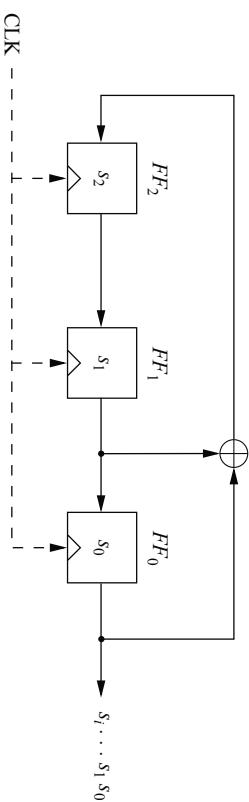
LFSR tổng quát

Bài tập



Hình: LFSR với hệ số phản hồi p_i và giá trị ban đầu s_{m-1}, \dots, s_0

Thanh ghi dịch phản hồi tuyến tính



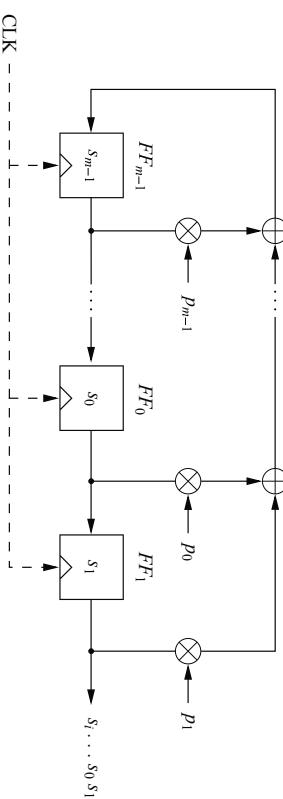
Công thức truy hồi:

$$s_{i+3} = 0 \cdot s_{i+2} + 1 \cdot s_{i+1} + 1 \cdot s_i \pmod{2}, \quad i = 0, 1, 2, \dots$$

- Xét LFSR với bậc $m = 4$ và hệ số phản hồi ($p_3 = 0, p_2 = 0, p_1 = 1, p_0 = 1$).
- Bắt đầu từ $s_3 = 0, s_2 = 1, s_1 = 0, s_0 = 0$ hãy tính 15 bit tiếp theo của dãy output.

Công thức truy hồi

$$s_{i+m} = \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \pmod{2}$$



Độ dài lớn nhất của dãy

LFSR và đà thực

- Xét dãy tạo bởi LFSR với công thức truy hồi:

$$s_{i+m} = \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \pmod{2}; \quad s_i, p_j \in \{0, 1\}; \quad i = 0, 1, 2, \dots$$

- Phụ thuộc vào m , dãy này lặp lại theo chu kỳ với độ dài khác nhau.

Định lý

Độ dài (chu kỳ) lớn nhất của dãy sinh bởi LFSR là $2^m - 1$.

Bài tập

Ví dụ

Xét LFSR với bậc $m = 4$ và hệ số phản hồi

- Xét LFSR với bậc $m = 4$ và hệ số phản hồi

$$(p_3 = 1, p_2 = 1, p_1 = 1, p_0 = 1).$$

Dãy output có chu kỳ $2^4 - 1 = 15$.

Ví dụ

Xét LFSR với bậc $m = 4$ và hệ số phản hồi

$$(p_3 = 1, p_2 = 1, p_1 = 1, p_0 = 1).$$

Dãy output có chu kỳ 5.

LFSR bậc m với hệ số phản hồi $(p_{m-1}, \dots, p_1, p_0)$ biểu diễn bởi đà thực

$$P(x) = x^m + p_{m-1}x^{m-1} + \dots + p_1x + p_0$$

Ví dụ
LFSR với bậc $m = 4$ và hệ số phản hồi

$$(p_3 = 0, p_2 = 0, p_1 = 1, p_0 = 1)$$

biểu diễn bởi đà thực

$$P(x) = x^4 + x + 1.$$

Nội dung

Đa thức nguyên thuỷ và LSFR

- Chỉ có LFSR xác định bởi đa thức nguyên thuỷ mới có **dãy output** với chu kỳ cực dài!

- Đa thức nguyên thuỷ là một trường hợp riêng của đa thức bất khả quy (giống số nguyên tố).

- **Ví dụ:** Đa thức

$$(0, 2, 5) \rightarrow 1 + x^2 + x^5$$

là đa thức nguyên thuỷ.

- 1 Giới thiệu

- 2 LFSR: Dạng tổng quát

- 3 Tần công LFSR

- 4 Trivium: Một hệ mã dòng hiện đại

Một số đa thức nguyên thuỷ

(0,1,2)	(0,1,3,4,24)	(0,1,46)	(0,1,5,7,68)	(0,2,3,5,90)	(0,3,4,5,112)
(0,1,3)	(0,3,25)	(0,5,47)	(0,2,5,6,69)	(0,1,5,8,91)	(0,2,3,5,113)
(0,1,4)	(0,1,3,4,26)	(0,2,3,5,48)	(0,1,3,5,70)	(0,2,5,6,92)	(0,2,3,5,114)
(0,2,5)	(0,1,2,5,27)	(0,4,5,6,49)	(0,1,3,5,71)	(0,2,93)	(0,5,7,8,115)
(0,1,6)	(0,1,28)	(0,2,3,4,50)	(0,3,9,10,72)	(0,1,5,6,94)	(0,1,2,4,116)
(0,1,7)	(0,2,29)	(0,1,3,6,51)	(0,2,3,4,73)	(0,11,95)	(0,1,2,5,117)
(0,1,3,4,8)	(0,1,30)	(0,3,52)	(0,1,2,6,74)	(0,6,9,10,96)	(0,2,5,6,118)
(0,1,9)	(0,3,31)	(0,1,2,6,53)	(0,1,3,6,75)	(0,6,97)	(0,8,119)
(0,3,10)	(0,2,3,7,32)	(0,3,6,8,54)	(0,2,4,5,76)	(0,3,4,7,98)	(0,1,3,4,120)
(0,2,11)	(0,1,3,6,33)	(0,1,2,6,55)	(0,2,5,6,77)	(0,1,3,6,99)	(0,1,5,8,121)
(0,3,12)	(0,1,3,4,34)	(0,2,4,7,56)	(0,1,2,7,78)	(0,2,5,6,100)	(0,1,2,6,122)
(0,1,3,4,13)	(0,2,35)	(0,4,57)	(0,2,3,4,79)	(0,1,6,7,101)	(0,2,123)
(0,5,14)	(0,2,4,5,36)	(0,1,5,6,58)	(0,2,4,9,80)	(0,3,5,6,102)	(0,3,7,124)
(0,1,15)	(0,1,4,6,37)	(0,2,4,7,59)	(0,4,81)	(0,9,103)	(0,5,6,7,125)
(0,1,3,5,16)	(0,1,5,6,38)	(0,1,60)	(0,4,6,9,82)	(0,1,3,4,104)	(0,2,4,7,126)
(0,3,17)	(0,4,39)	(0,1,2,5,61)	(0,2,4,7,83)	(0,4,105)	(0,1,127)
(0,3,18)	(0,3,4,5,40)	(0,3,5,6,62)	(0,5,84)	(0,1,5,6,106)	(0,1,2,7,128)
(0,1,2,5,19)	(0,3,41)	(0,1,63)	(0,1,2,8,85)	(0,4,7,9,107)	
(0,3,20)	(0,1,2,5,42)	(0,1,3,4,64)	(0,2,5,6,86)	(0,1,4,6,108)	
(0,2,21)	(0,3,4,6,43)	(0,1,3,4,65)	(0,1,5,7,87)	(0,2,4,5,109)	
(0,1,22)	(0,5,44)	(0,3,66)	(0,8,9,11,88)	(0,1,4,6,110)	
(0,5,23)	(0,1,3,4,45)	(0,1,2,5,67)			

Bước 1

Tính toán

$$\begin{aligned}y_i &= x_i + s_i \mod 2 \\s_i &= y_i + x_i \mod 2\end{aligned}$$

với $i = 0, 1, \dots, 2m - 1$

- Hệ phương trình tuyến tính m ẩn.
- **Dễ giải dùng phương pháp khử Gauss!**



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

21 / 34

Dựa trên giả sử rằng

Oscar có:

- Mọi bit bản mã y_i
- Bậc m
- Các bit bẩn rõ ($x_0, x_1, \dots, x_{2m-1}$)

- Mục đích: Lấy được dãy bit khoá

s_{2m}, s_{2m+1}, \dots

- **Câu hỏi:** Làm thế nào để tính:

p_0, p_1, \dots, p_{m-1} ?

Nhắc lại:

$$s_{i+m} = \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \mod 2$$

Bước 2: Tính p_i

$$\begin{aligned}i &= 0, & s_m &= p_m s_{m-1} + \dots + p_1 s_1 + p_0 s_0 & \mod 2 \\i &= 1, & s_{m+1} &= p_m s_m + \dots + p_1 s_2 + p_0 s_1 & \mod 2 \\&\vdots & & & \\i &= m-1, & s_{2m-1} &= p_{m-1} s_{2m-2} + \dots + p_1 s_m + p_0 s_{m-1} & \mod 2\end{aligned}$$

23 / 34

Bước 2



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

20 / 34



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

22 / 34

Bước 3

Giới thiệu Trivium

- Dùng cấu hình $(p_{m-1}, \dots, p_1, p_0)$ để xây dựng LFSR.
- Tính dãy bit khoá $s_0, s_1, \dots, s_{2m}, \dots$
- Giải mã $x_i = y_i + s_i \pmod{2}$.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

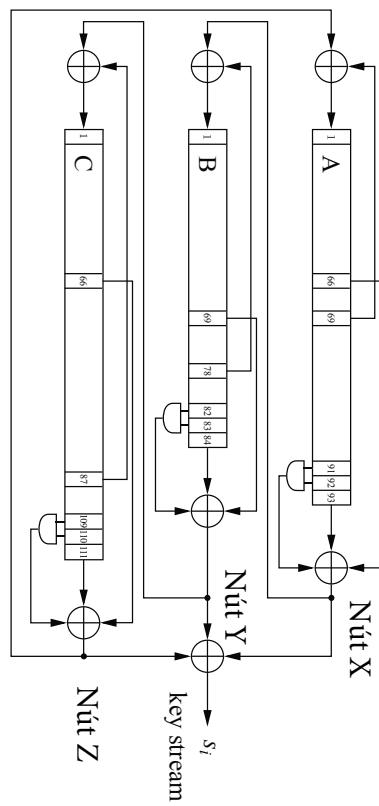
25 / 34

Hệ quả

Nếu kẻ tấn công có (ít nhất) $2m$ giá trị output của LFSR, anh ta có thể lấy được toàn bộ thông tin về cấu hình

p_0, p_1, \dots, p_{m-1} .

của LFSR.



Hình: Hệ mã dòng mới với kích thước khoá 80 bit. Dựa trên việc kết hợp ba thanh ghi dịch có phản hồi, và kết hợp với thành phần phi tuyến.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

27 / 34

Nội dung

- ➊ Giới thiệu
- ➋ LFSR: Dạng tổng quát
- ➌ Tấn công LFSR
- ➍ Trivium: Một hệ mã dòng hiện đại



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

24 / 34

Mã hoá với Trivium: Khởi tạo

- 80 bit IV được đưa vào 80 bit trái nhất của thanh ghi A. IV không cần giữ bí mật nhưng phải thay đổi sau cho mỗi phiên làm việc.
- **80 bit khóa** được đưa vào 80 bit trái nhất của thanh ghi B.
- Mọi bit thanh ghi khác được đặt bằng 0 ngoại trừ ba bit phải nhất của thanh ghi C:

$$c_{109} = c_{110} = c_{111} = 1.$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

29 / 34

Đặc tả Trivium

register length	feedback bit	feedforward bit	AND inputs
A	93	69	66 91,92
B	84	78	69 82,83
C	111	87	66 109,110

Mã hoá với Trivium: Pha khởi động

- Trong pha đầu tiên này, hệ mã được chạy $4 \times (93 + 84 + 111) = 1152$ lần, nhưng không tạo ra bit đầu ra nào.
- Pha này cần để tạo cho hệ mã đủ ngẫu nhiên.
- Nó đảm bảo dòng khoá phụ thuộc vào cả k và IV .

- Phép toán AND chính là phép nhân theo modun 2, và phương trình không còn là tuyến tính nữa.
- Feedforward paths liên quan đến phép toán AND là thành phần quan trọng cho tính an toàn của hệ.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

28 / 34



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

31 / 34

Mã hoá với Trivium



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

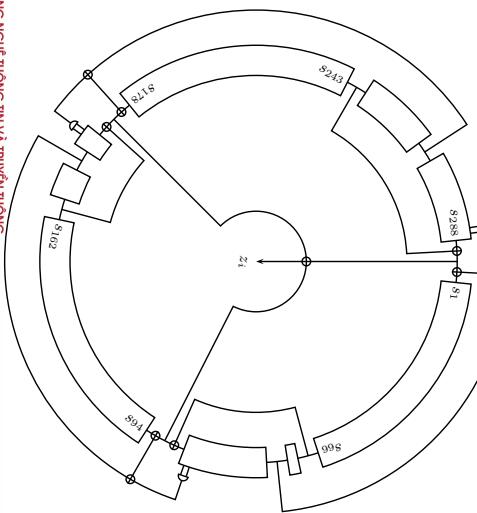
30 / 34

Bài tập lập trình

- ① Cài đặt hệ mã dòng Trivium.
- ② Mã hoá file với Trivium. Bạn có thể sinh IV ngẫu nhiên và đặt vào đầu bản mã.

Tham khảo đặc tả chi tiết Trivium

https://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf



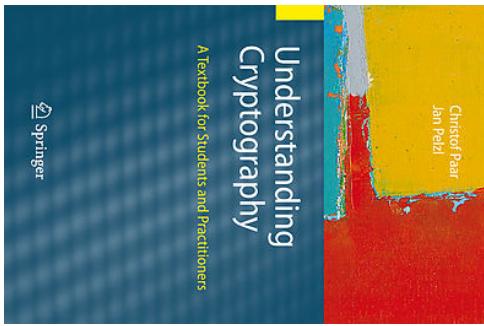
Cảm ơn!



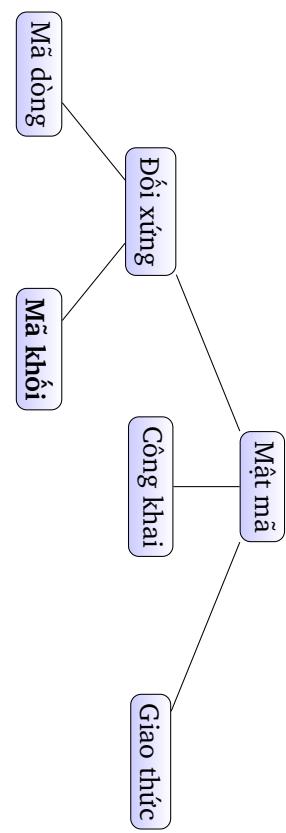
Mật mã

<https://www.crypto-textbook.com>

Tài liệu



2 / 39



4 / 39

Nội dung

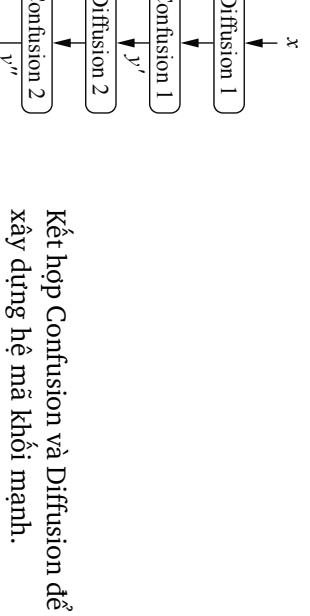
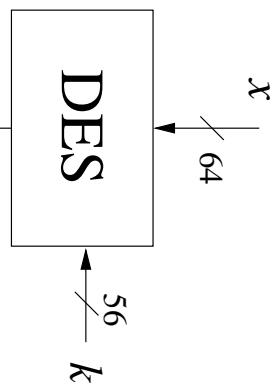
- ➊ Giới thiệu
- ➋ Tổng quan về DES
- ➌ Bên trong DES
- ➍ Mở rộng khóa
- ➎ Giải mã DES
- ➏ Tính an toàn của DES

Nhập môn An toàn Thông tin
Data Encryption Standard (DES) và một số biến thể



1 / 39

Nguyên lý xây dựng mã khối



- Hiện nay, DES không còn an toàn do kích thước khóa ngắn.
- Nhưng 3DES thì rất an toàn.

Lịch sử

- Đề xuất bởi IBM năm 1974 dựa trên hệ Lucifer.

Lucifer là hệ mật phat triển bởi Horst Feistel cuối những năm 1960. Lucifer có kích thước khối 64 bit và khóa 128 bit.

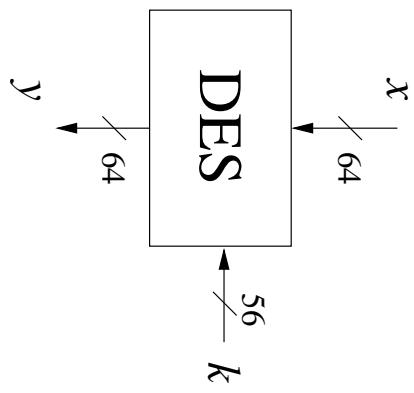
- National Security Agency (NSA) đã sửa đổi và đặt tên là **DES**.
- Sửa đổi này cho phép chống lại kiểu **thám mã vi phân**. Kiểu tấn công này chưa được biết đến trước năm 1990.
- Tuy nhiên, NSA lại sửa đổi kích thước khóa từ 128 bit xuống còn 56 bit!
⇒ Có thể tấn công vét cạn.
- Nhiều người giả thuyết rằng NSA có thể tìm kiếm khoá trong không gian 2^{56} .
- Năm 1977, công bố chuẩn mã hoá dữ liệu DES.

Nguyên lý xây dựng mã khối

theo Claude Shannon

- Làm hỗn loạn (Confusion)** là phép toán mã hoá nhằm che giấu liên hệ giữa khoá và bản mã.
- Khuêch tán (Diffusion)** là phép toán mã hoá làm cho việc sửa một bit ở bản rõ sẽ ảnh hưởng rộng đến nhiều bit của bản mã. Mục tiêu là giấu tính chất thống kê của bản rõ.

DES



- Khoá 56 bit

- Khối 64 bit.

Nội dung

1 Giới thiệu

2 Tổng quan về DES

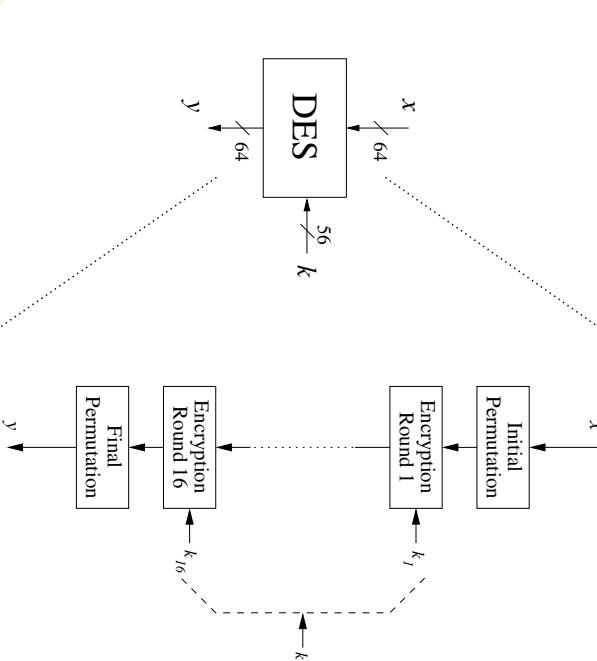
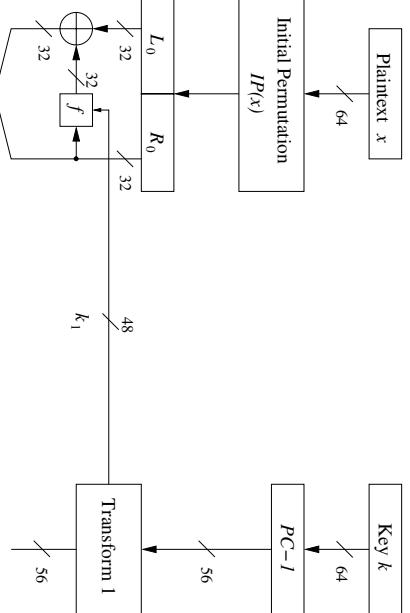
3 Bên trong DES

4 Mở rộng khóa

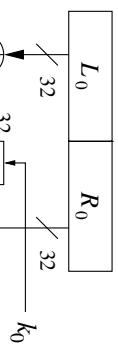
5 Giải mã DES

6 Tính an toàn của DES

Mạng Feistel: Vòng 1



Cấu trúc mạng Feistel

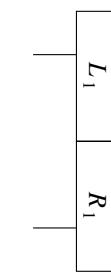


- Công thức tổng quát:

$$L_i = R_{i-1}$$

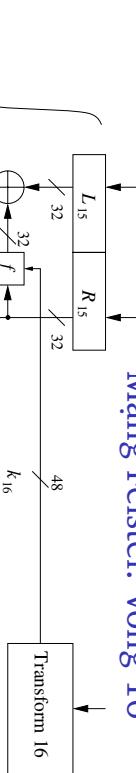
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

- Làm thế nào để tính ngược lại L_{i-1} và R_{i-1} ?



- Làm thế nào để tính ngược lại L_{i-1} và R_{i-1} ?

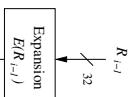
Mạng Feistel: Vòng 16



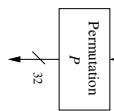
Nội dung

- ➊ Giới thiệu
- ➋ Tổng quan về DES
- ➌ Bên trong DES
- ➍ Mở rộng khóa
- ➎ Giải mã DES
- ➏ Tính an toàn của DES

Hàm f



- Mở rộng đầu vào $E(R_{i+1})$
- XOR với khoá vòng i
- Bảng thay thế S-box
- Hoán vị P

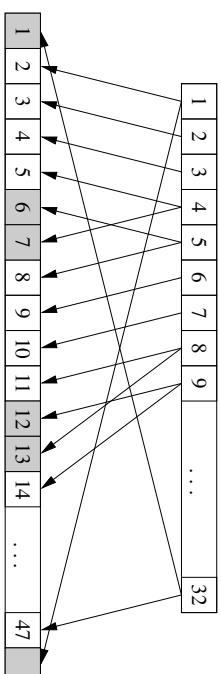


Hoán vị ban đầu IP và kết thúc IP^{-1}

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Hàm mở rộng đầu vào E



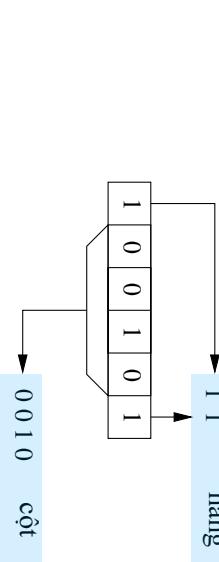
S-Box

- S-box là hàm $\{0,1\}^6 \rightarrow \{0,1\}^4$; 6 bit input và 4 bit output.
- Gồm 8 S-box được thiết kế phi tuyến

$S(a) \oplus S(b) \neq S(a \oplus b)$

để chống lại thám mã vi phân.

- Bảng S-box được giải mã theo cách đặc biệt:



Nội dung

Hoán vị P

P
16 7 20 21 29 12 28 17
1 15 23 26 5 18 31 10
2 8 24 14 32 27 3 9
19 13 30 6 22 11 4 25

Hình: Hoán vị P là phép khuếch tán, gây ảnh hưởng đến nhiều S box khác trong vòng tiếp theo.

Hoán vị P

1 Giới thiệu

2 Tổng quan về DES

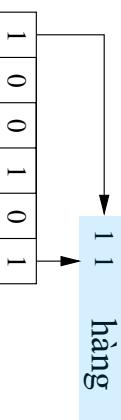
3 Bên trong DES

4 Mở rộng khóa

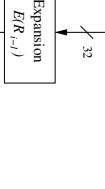
5 Giải mã DES

6 Tính an toàn của DES

S-box



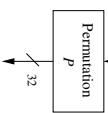
Hàm f



- Mở rộng đầu vào $E(R_{i+1})$

S ₁	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

S ₁	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13



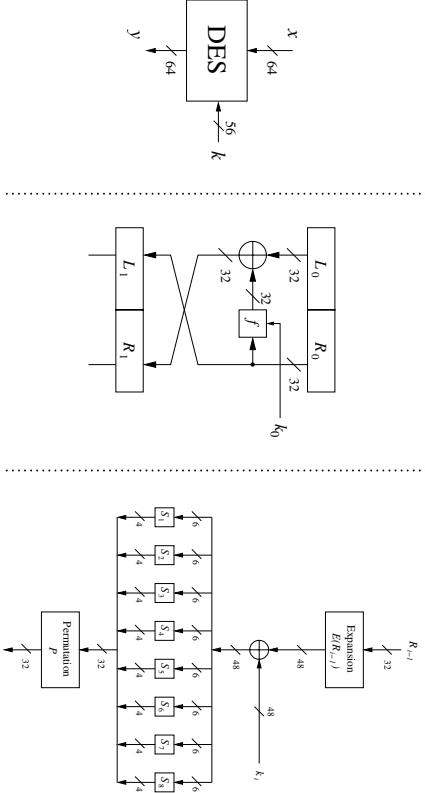
Mở rộng khoá

PC-1: Permutated Choice 1

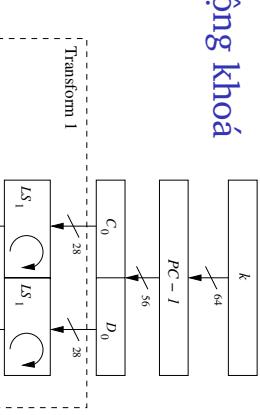
- Loại bỏ các bit 8, 16, 24, ..., 64 của khoá k kích thước 64 bit.
- Khoá thực sự của DES chỉ là $(64 - 8) = 56$ bit.

$PC - 1$
57 49 41 33 25 17 9 1
58 50 42 34 26 18 10 2
59 51 43 35 27 19 11 3
60 52 44 36 63 55 47 39
31 23 15 7 62 54 46 38
30 22 14 6 61 53 45 37
29 21 13 5 28 20 12 4

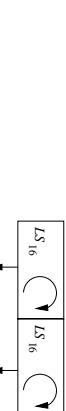
Nhắc lại: Thành phần của DES



Mở rộng khoá



- Câu hỏi: Làm thế nào để tính 16 khoá con k_1, \dots, k_{16} ?
- Mở rộng khoá chỉ gồm các phép toán đơn giản (hoán vị và xoay vòng trái) trên bit.



PC-2: Permuted Choice 2

- Loại bỏ 8 bit của $C_i \boxed{D_i}$;
- Số bit của khóa con k_i là $56 - 8 = 48$ bit

PC-2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32



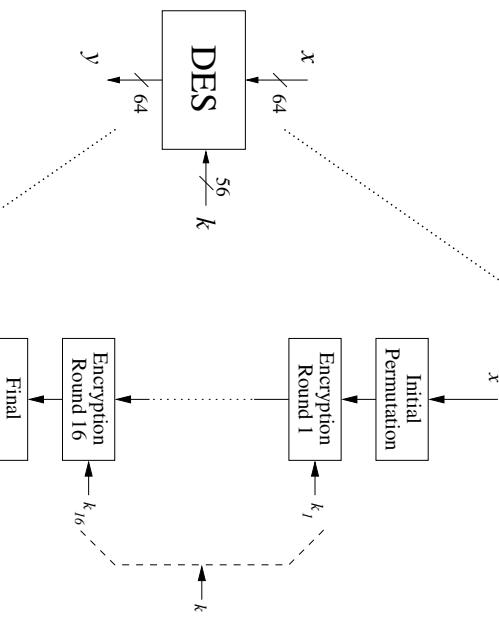
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

30 / 39



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

32 / 39



Nội dung

1 Giới thiệu

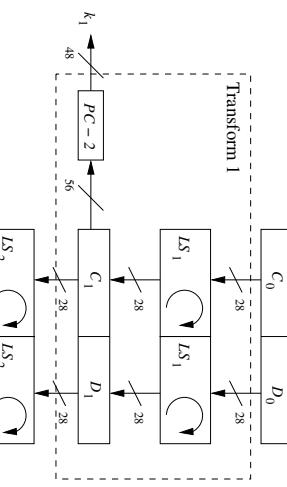
2 Tổng quan về DES

3 Bên trong DES

4 Mở rộng khóa

5 Giải mã DES

6 Tính an toàn của DES



$LS_i = \begin{cases} \text{Xoay vòng trái 1 vị trí nếu } i = 1, 2, 9, 16 \\ \text{Xoay vòng trái 2 vị trí trong trường hợp khác.} \end{cases}$

Chú ý: Tổng số bit được xoay vòng $4 \times 1 + 12 \times 2 = 28$, do đó

Nội dung

1 Giới thiệu

2 Tổng quan về DES

3 Bên trong DES

4 Mở rộng khóa

5 Giải mã DES

6 Tính an toàn của DES

Bổ đề

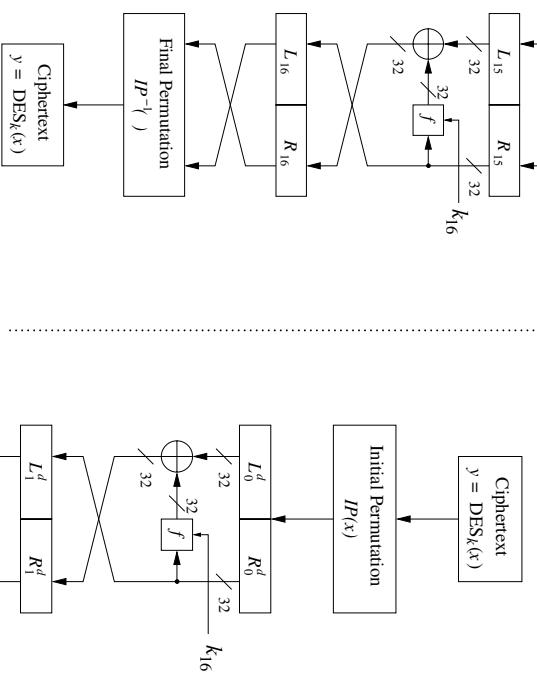
Giả sử DES là một hệ mã **ý tưởng** (2^{56} hàm khả nghịch ngẫu nhiên)
 $\pi_i : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$)

Vậy thì với mỗi cặp x, y có nhiều nhất **một** khóa k thỏa mãn

$$y = DES(k, x)$$

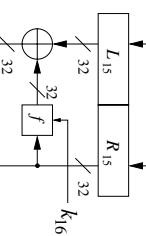
với xác suất $\geq 1 - 1/256 \approx 99.5\%$.

Giải mã mỗi vòng



Tán công vét can để tìm khóa của
mã khôi

mã khôi

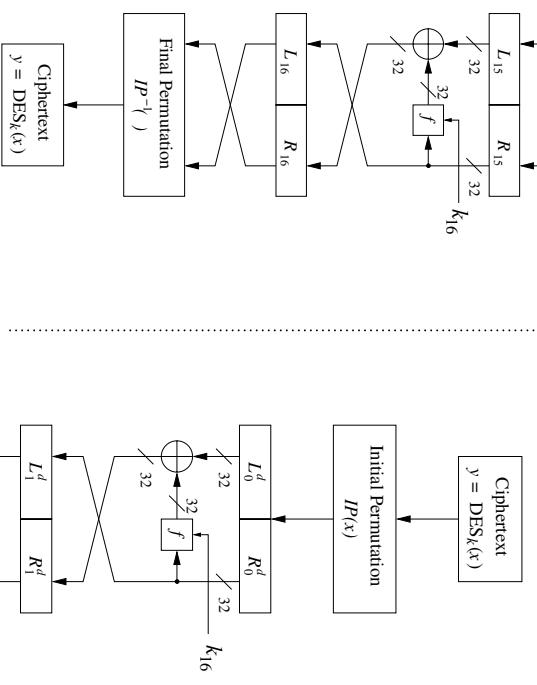


- Bài toán
- Cho một số cặp input/output

$$(x_i, y_i = \text{Enc}(k, x_i))$$

với $i = 1, 2, 3$.

- Hãy tìm khóa k .



Thử thách DES

Cho các cặp bản rõ và bản mã

msg = "The unknown messages is : xxxx . . ."
CT = $y_1 \quad y_2 \quad y_3 \quad y_4$

Hãy tìm khóa $k \in \{0, 1\}^{56}$ thỏa mãn $DES(k, x_i) = y_i$ với $i = 1, 2, 3$.

- 1997: DESCHALL project với internet search – **96 ngày**
- 1998: EFF dùng máy DeepCrack – **3 ngày** (250K \$)
- 1999: Kết hợp cả DeepCrack và internet search – **22 giờ**
- 2006: COPACOBANA (120 FPGA) – **7 ngày** (10K \$).

Không nên dùng mã khóa 56 bit khóa !!
128-bit khóa $\Rightarrow 2^{72}$ ngày

Tìm kiếm vét can để tìm khóa cho
mã khôi

- Với hai cặp DES:

$$(x_1, y_1 = DES(k, x_1)) \text{ và } (x_2, y_2 = DES(k, x_2))$$

xác suất để có k có duy nhất là $\approx 1 - 1/2^{71}$.

- Với AES-128: cho hai cặp input/output, xác suất có k duy nhất $\approx 1 - 1/2^{128}$
- Vậy hai cặp input/output là đủ thông tin để tìm kiếm vét can cho khóa.

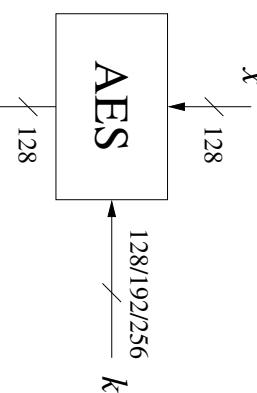
Cảm ơn!



Hệ AES

Định nghĩa (Trường)

Một trường F là một tập với các tính chất sau:



AES được xây dựng dựa trên một số phép toán trên trường hữu hạn.



2 / 51

$$a \times (b + c) = (a \times b) + (a \times c), \text{ với mọi } a, b, c \in F.$$

- Các phần tử của F ngoại trừ 0 tạo thành một nhánh với phép toán \times với phần tử đơn vị là 1.
- Các phần tử của F cùng với hai phép toán $+$ và \times thỏa mãn **luật phân phối**, tức là:



Nội dung

Nhập môn An Toàn Thông Tin The Advanced Encryption Standard (AES)

- ➊ Trường hữu hạn

- ➋ AES

- ➌ Giải mã AES

1 / 51



4 / 51

Điều kiện tồn tại trường hữu hạn

Số phần tử của trường F được gọi là **cấp** hay **lực lượng** của trường F .

Định lý

Tồn tại trường **cấp n** nếu $n = p^m$ với p là số nguyên tố và m là một số nguyên dương. Số p được gọi là **đặc số** của trường hữu hạn.

Ví dụ

- Tồn tại trường hữu hạn có 11 phần tử $GF(11)$.
- Tồn tại trường hữu hạn có 81 phần tử $GF(81)$.
- Tồn tại trường hữu hạn có 256 phần tử $GF(2^8)$ (cũng gọi là trường AES).
- Không tồn tại trường với 12 phần tử. Tại sao?

6 / 51

Trường mở rộng $GF(2^m)$

Các phần tử của $GF(2^m)$ là các đa thức:

$$a_{m-1}x^{m-1} + \dots + a_1x + a_0 = A(x) \in GF(2^m)$$

với $a_i \in GF(2) = \{0, 1\}$.

Ví dụ

Các phần tử của trường $GF(2^3) = GF(8)$ là các đa thức

$$A(x) = a_2x^2 + a_1x + a_0$$

$GF(2^3)$ có $2^3 = 8$ phần tử:

$$GF(2^3) = \{ \begin{array}{llll} 0, & 1, & x, & x+1 \\ x^2, & x^2+1, & x^2+x, & x^2+x+1 \end{array} \}$$

6 / 51

Trường mở rộng $GF(2^m)$

Các phần tử của $GF(2^m)$ là các đa thức:

$a_{m-1}x^{m-1} + \dots + a_1x + a_0 = A(x) \in GF(2^m)$

Ví dụ

Các phần tử của trường $GF(2^3) = GF(8)$ là các đa thức

$$A(x) = a_2x^2 + a_1x + a_0$$

$GF(2^3)$ có $2^3 = 8$ phần tử:

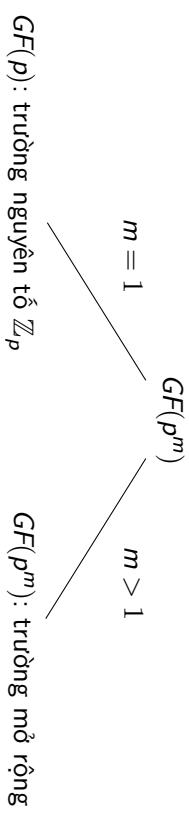
$$GF(2^3) = \{ \begin{array}{llll} 0, & 1, & x, & x+1 \\ x^2, & x^2+1, & x^2+x, & x^2+x+1 \end{array} \}$$

6 / 51

Kiểu trường hữu hạn

Ví dụ

- Tập số thực \mathbb{R} cùng với hai phép toán $+$ và \times tạo thành một trường.
- Tập \mathbb{Z}_p với hai phép toán $+$ và \times theo modun nguyên tố p là một trường. Trường này có hữu hạn phần tử.



Nhận xét

Hai trường hay được dùng trong mật mã là $GF(p)$ và $GF(2^m)$.

Cộng và trừ

Phép nhân trên trường mở rộng

Ta sẽ chia lấy dư cho một đa thức bất khả quy, là đa thức tương tự như số nguyên tố.

Ví dụ (Tính toán trên $GF(2^3)$)

Định nghĩa (Phép nhân trên trường mở rộng $GF(2^m)$)

Xét $A(x), B(x) \in GF(2^m)$ và xét

$$\begin{aligned} A(x) &= x^2 + x + 1 \\ B(x) &= x^2 + 1 \\ A(x) + B(x) &= x \\ P(x) &= \sum_{i=0}^m p_i x^i, \quad p_i \in GF(2) \end{aligned}$$

là một đa thức bất khả quy. Phép nhân của hai phần tử $A(x), B(x)$ có kết quả là

$$C(x) = A(x) \cdot B(x) \mod P(x).$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

10 / 51

Câu hỏi

Ví dụ (Tính toán trên $GF(2^3)$)

Làm thế nào để tính toán $(+, -, \times, /)$ trên $GF(2^m)$?

Tính toán như trên đa thức thông thường với các hệ số được tính trên $GF(p)$.

Nhân

Nhắc lại: với trường nguyên tố $GF(7) = \{0, 1, \dots, 6\}$

$$3 \cdot 4 = 12 = 5 \mod 7$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

9 / 51



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11 / 51

Đa thức bắt khả quy

Phân tử nghịch đảo trong mở rộng trường

- Không phải mọi đa thức đều bắt khả quy. Ví dụ,

$$x^4 + x^3 + x + 1 = (x^2 + x + 1)(x^2 + 1)$$

không phải là bắt khả quy.

- Trong AES, người ta sử dụng đa thức bắt khả quy

$$P(x) = x^8 + x^4 + x^3 + x + 1$$

Phép nhân trên trường mở rộng

Ví dụ (Tính toán trên $GF(2^3)$)

$$\begin{aligned}A(x) &= x^2 + x + 1 \\B(x) &= x^2 + 1 \\A(x) \times B(x) &= (x^2 + x + 1)(x^2 + 1) \\&= x^4 + x^3 + x + 1 \notin GF(2^3)\end{aligned}$$

Ta chọn đa thức bắt khả quy là

$$P(x) = x^3 + x + 1$$

Khi đó

$$A(x) \cdot B(x) = x^2 + x \mod P(x)$$

$$\text{vì } (x^4 + x^3 + x + 1)/(x^3 + x + 1) = x + 1; \text{ và dư } x^2 + x.$$

Trường hữu hạn trong SageMath

<https://cocalc.com>

- Trường hữu hạn $K = GF(2^3)$ với mo đun $P(x) = x^3 + x + 1$:

```
1 K.<x> = GF(2^3, name='x', modulus=x^3 + x + 1)
```

- Phép nhân trong SageMath:

```
1 A=x^2 + x + 1
2 B=x^2 + 1
3 C=A*B
4 print (C)
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	00	01	8D	F5	CB	52	7B	DB	E8	4F	29	C0	B0	E1	E5	C7	
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2	
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2	
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19	
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	B9	09	
5	ED	5C	05	CA	4C	24	87	F4	22	F0	51	EC	61	17			
6	16	5F	AF	D3	49	36	43	F4	47	91	DF	33	93	21	3B		
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82	
X	8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A	
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62	
B	0C	E0	1F	EF	11	75	78	71	A5	76	3D	BD	BC	86	57		
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6	
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B	
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3	
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C	

Bảng trên tính nghịch đảo trong $GF(2^8)$. Ví dụ, nghịch đảo của

$$x^7 + x^6 + x = (11000010)_2 = (C2)_{hex} = (XY)$$

được cho bởi ô tại dòng C, cột 2:

$$(2F)_{hex} = (00101111)_2 = x^5 + x^3 + x^2 + x + 1.$$

18 / 51

Sơ lược lịch sử

- 1997: Kêu gọi đề xuất Chuẩn mã hóa nâng cao AES bởi NIST
- 1998: Có 15 thuật toán đề xuất
- Tháng 8 năm 1999: Chọn 5 thuật toán vào vòng cuối
 - 1 **Mars** bởi IBM
 - 2 **RC6** bởi RSA Laboratories
 - 3 **Rijndael** bởi Joan Daemen và Vincent Rijmen
 - 4 **Serpent** bởi Ross Anderson, Eli Biham và Lars Knudsen
 - 5 **Twofish** bởi Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall và Neils Ferguson
- Tháng 10 năm 2000: **Rijndael** đã được chọn làm AES



Sơ lược lịch sử

20 / 51

Trường AES trong SageMath

```

1 sage: K.<x>=GF(2^8, name='x', modulus=x^8+x^4+x^3+x+1)
2 sage: (x^7+x^6+x)^-1
3 x^5 + x^3 + x^2 + x + 1
4 sage: (x^7+x^6+x)*(x^5+x^3+x^2+x+1)
5 1

```

Nội dung

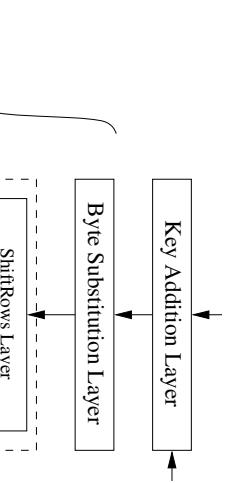
1 Trường hữu hạn

2 AES

3 Giải mã AES

Cấu trúc một vòng của AES

Byte Substitution layer



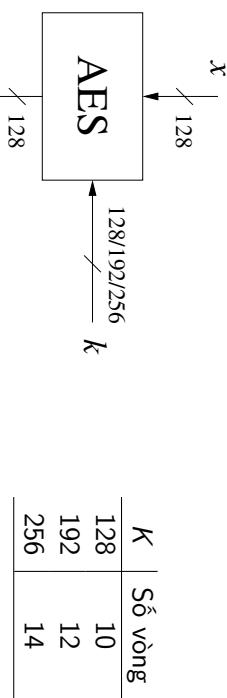
- Chú ý: Riêng vòng cuối cùng không có thao tác MixColumn.



VIEN CONG NGHE THONG TIN VÀ TRUYỀN THÔNG

22 / 51

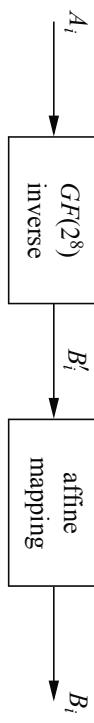
AES



K	Số vòng
128	10
192	12
256	14

y

1 vòng: 128 bit được tách thành 16 bytes
($16 \times 8 = 128$)



Hình: Dùng 16 Sbox giống nhau: $S(A_i) = B_i$

- Hỏi: Bảng Sbox được xây dựng thế nào?
- Trả lời: Coi $A_i \in GF(2^8)$ và tính nghịch đảo $A_i^{-1} = B'_i$; sau đó đưa qua một phép biến đổi tuyến tính.



VIEN CONG NGHE THONG TIN VÀ TRUYỀN THÔNG

24 / 51

Ví dụ

Diffusion Layer = Shift Rows và Mix Column

- Giả sử input

$$A_i = (11000010)_2 = (C2)_{hex} \in GF(2^8)$$

- Ta có

$$A_i^{-1} = B'_i = (2F) = ((00101111)_2 \in GF(2^8)$$

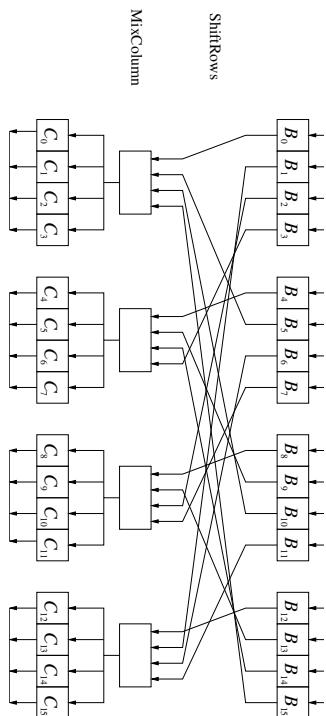
- Qua phép biến đổi tuyến tính ta được

$$B_i = (0010\ 0101)_2 = (25)_{hex}$$

Phép biến đổi tuyến tính $B'_i \rightarrow B_i$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \text{ mod } 2.$$

S-box: $S((C2)_{hex}) = S(C, 2) = (25)_{hex}$



Ví dụ

- Là một phép biến đổi tuyến tính biến đổi

$\text{MixColumn}(B) = C$

- Mỗi cột 4 byte được xem như một vector, và được nhân với một ma trận cố định trước.

- Phép cộng và phép nhân các hệ số được thực hiện trong $GF(2^8)$.

- Ma trận của phép biến đổi tuyến tính MixColumn là

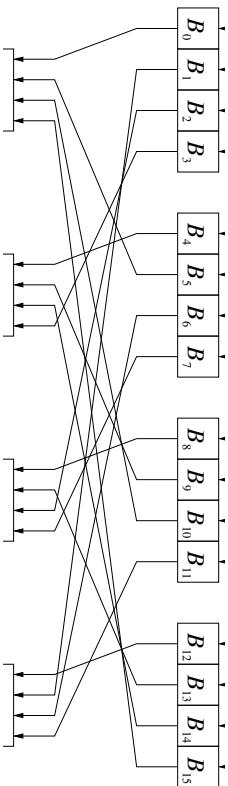
$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

30 / 51

Shift Rows



Mix Column

- Giả sử input của MixColumn là

$$B = (25, 25, \dots, 25).$$

- Do ma trận MixColumn, ta chỉ cần tính toán theo đa thức trong $GF(2^8)$ với 02 · 25 và 03 · 25:

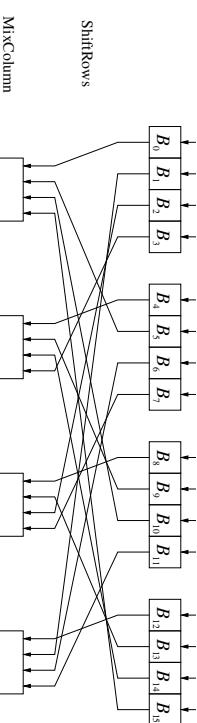
$$\begin{aligned} 02 \cdot 25 &= (x+1) \cdot (x^5 + x^2 + 1) \\ &= (x^6 + x^3 + x) + (x^5 + x^2 + 1) \\ &= x^6 + x^5 + x^3 + x^2 + x + 1 \end{aligned}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

32 / 51

Ví dụ



B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

→

B_0	B_4	B_8	B_{12}
B_5	B_9	B_{13}	B_1
B_{10}	B_{14}	B_2	B_6
B_{15}	B_3	B_7	B_{11}



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

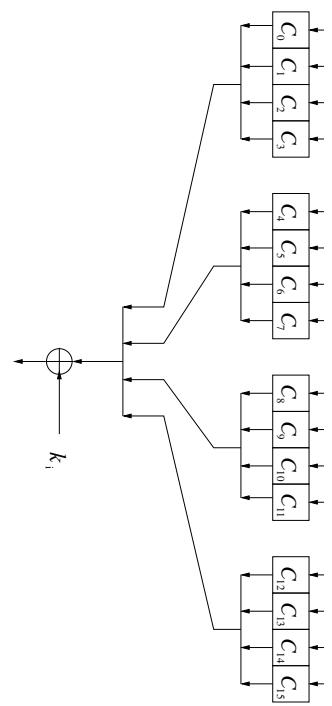
29 / 51

C_0	C_1	C_2	C_3
C_4	C_5	C_6	C_7
C_8	C_9	C_{10}	C_{11}
C_{12}	C_{13}	C_{14}	C_{15}

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix} \begin{pmatrix} B_1 \\ B_6 \\ B_2 \\ B_7 \end{pmatrix}$$

31 / 51

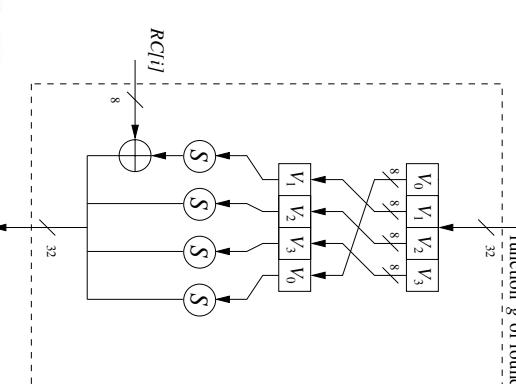
Key Addition Layer



- Input: Gồm 16-byte ma trận C và 16-byte khóa con k_i
- Output: $C \oplus k_i$
- Các khóa con được sinh trong thủ tục mở rộng khóa (Key schedule).

Hàm g ở vòng thứ i sử dụng $RC[i]$

$$\begin{aligned} RC[1] &= x^0 = (00000001)_2, \\ RC[2] &= x^1 = (00000010)_2, \\ RC[3] &= x^2 = (00000100)_2, \\ &\vdots \\ RC[10] &= x^9 = (00110110)_2. \end{aligned}$$



Key schedule cho AES với khóa 128 bit

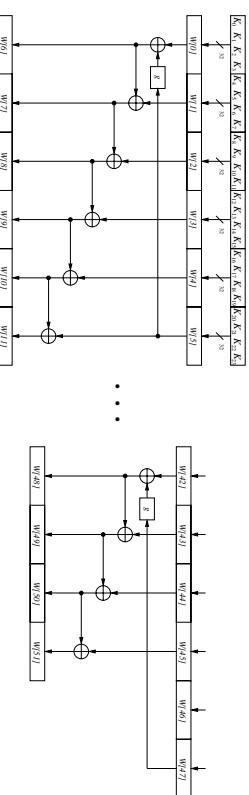


- Thực hiện phép cộng trong $GF(2^8)$ ta được kết quả của C :
- | | | | |
|-----------|---------|-------------|---------------|
| 01 · 25 = | $x^5 +$ | $x^2 +$ | 1 |
| 01 · 25 = | $x^5 +$ | $x^2 +$ | 1 |
| 02 · 25 = | $x^6 +$ | $x^3 +$ | x |
| $C_i =$ | $x^6 +$ | $x^5 + x^3$ | $x^2 + x + 1$ |
| | $x^5 +$ | $x^2 +$ | 1 |
- Vậy output của C là:
- $$C = (25, 25, \dots, 25).$$

AES-256: Key schedule vòng 1

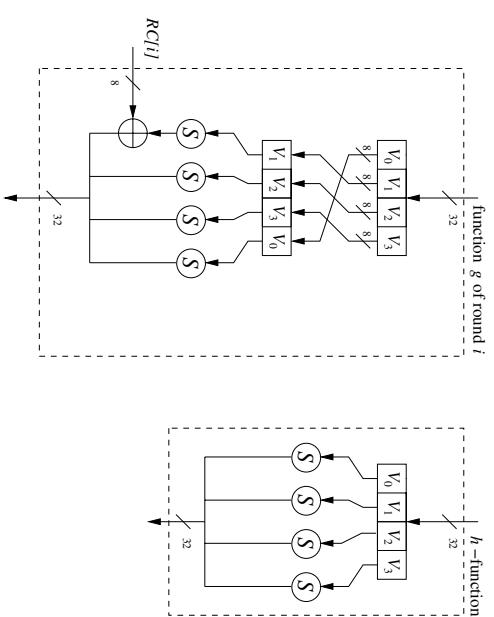


AES-192: Key schedule

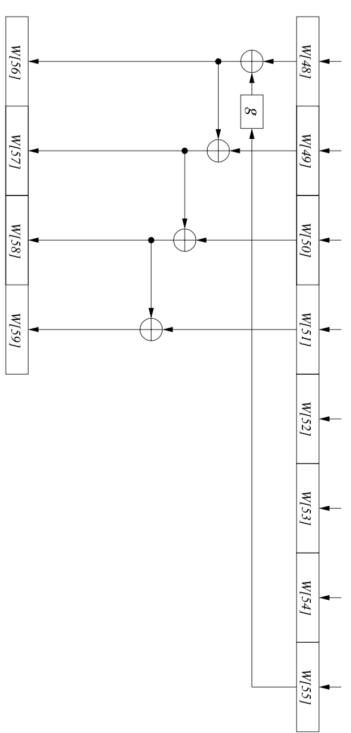


- AES với 192 bit có 12 vòng, vậy cần 13 Key Addition layer.
- Mỗi Key Addition Layer cần 128 bit khóa;
- Vây cần 52 khóa con $W[0], \dots, W[51]$ mỗi khóa 32 bit = 4 byte ($4 \times 13 = 52$).

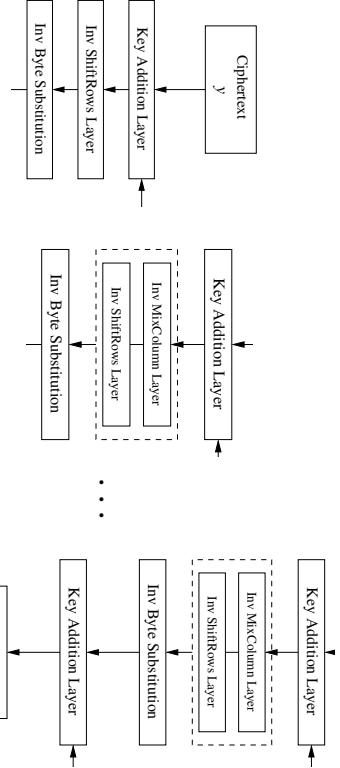
AES256: hàm g và h



AES-256: Key schedule vòng cuối



Sơ đồ giải mã



Hình: Vòng n

Hình: Vòng $n - 1$

Hình: Vòng 1

42 / 51

InvMixColumn: Hàm ngược của MixColumn

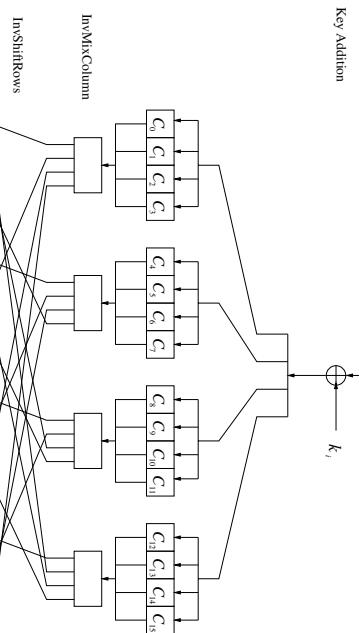
- Là phép biến đổi ngược của MixColumn
 $\text{InvMixColumn}(C) = B$
- Phép cộng và phép nhân các hệ số được thực hiện trong $GF(2^8)$;
- Mátrix của phép biến đổi tuyến tính InvMixColumn là

$$\begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix}$$

44 / 51

Nội dung

Key Addition



Hình: Mỗi vòng trong sơ đồ giải mã

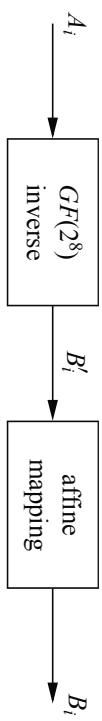
43 / 51

- Trường hữu hạn
- AES
- Giai mã AES

InvSubBytes: Hàm ngược của SubBytes

InvSubByte: $B'_i \rightarrow A_i$

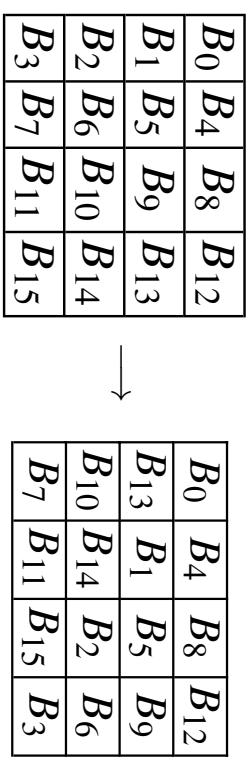
- Ta nhắc lại phép toán SubBytes:



- Để tính InvSubBytes, ta tính ngược lại:

$$B_i \rightarrow B'_i \rightarrow A_i$$

InvShiftRows: Hàm ngược của ShiftRows



InvSubBytes: Biến đổi tuyến tính ngược

$$B_i \rightarrow B'_i$$

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} \equiv \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \pmod{2},$$



Key Schedule của AES⁻¹

- Ở vòng thứ n của AES⁻¹, ta cần khóa con cuối cùng,
- Ở vòng thứ $n - 1$ của AES⁻¹, ta cần khóa con trước khóa con cuối, ...
- Tóm lại, ta cần tính các khóa con theo thứ tự ngược lại. Ví dụ, với AES⁻¹, thứ tự ta cần là

$$(W[40], W[41], W[42], W[43]) \rightarrow (W[0], W[1], W[2], W[3])$$

- Trên thực tế, ta sẽ tính trước toàn bộ 11 khóa con (nếu cho AES-128), 13 khóa con (nếu cho AES-192), hoặc 15 khóa con (nếu cho AES-256) và lưu lại.

Bảng tính InvSubBytes

x\y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5D	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	IE	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	ID	29	C5	89	6F	B7	62	0E	AA	18	BE	IB
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	IF	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D



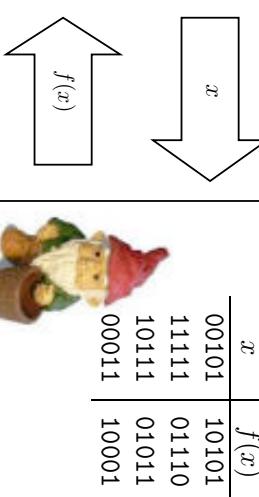
Hệ mã lý tưởng

Hoán vị ngẫu nhiên

Xét hệ mã khồi $E(k, x) = y$. Nếu giữ bí mật k , ta xác định hoán vị f như sau:

$$f(x) = E(k, x) = y$$

Nếu hệ mã E là **an toàn** thì f giống như một hoán vị ngẫu nhiên.



x	$f(x)$
00101	10101
11111	01110
10111	01011
00011	10001



Hệ mã lý tưởng

Xét hệ mã khồi $E(k, x) = y$. Nếu giữ bí mật k , ta xác định hoán vị f như sau:

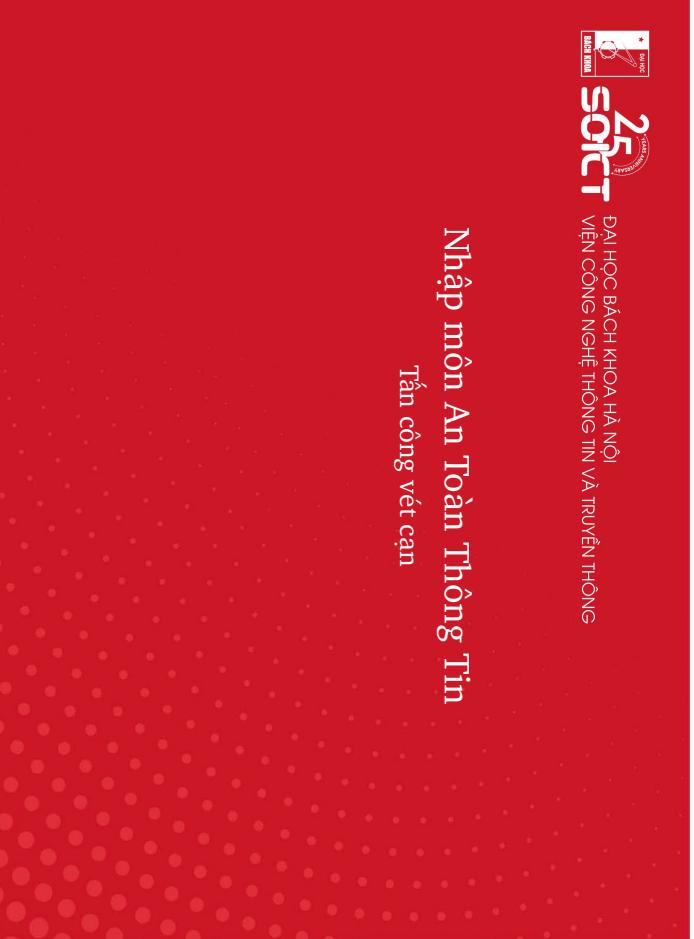
$$f(x) = E(k, x) = y$$

Nếu hệ mã E là **an toàn** thì f giống như một hoán vị ngẫu nhiên.

Cài đặt hoán vị ngẫu nhiên

Khi nhận truy vấn $x_i \in X$ từ kẻ tấn công \mathcal{A} :

```
if  $x_i == x_j$  với  $j < i$ 
then  $y_i = y_j$ 
else  $y_i \leftarrow_{\$} X \setminus \{y_1, \dots, y_{i-1}\}$ 
Gửi  $y_i$  cho  $\mathcal{A}$ .
```



Câu hỏi

Biết về ngữ cảnh

Rõ ràng A có thể biết c_1, \dots, c_q . Nhưng tại sao A lại biết m_1, \dots, m_q ?

- Do tiết lộ (về sau) của dữ liệu
- Do kiến thức có từ trước về ngữ cảnh.
- Bài toán càng chặt chẽ sẽ càng phù hợp cho nhiều tình huống thực tế!



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

5 / 23

Tán công vét cạn để tìm khóa của mã khôi

Kẻ tấn công A biết

$$E : K \times X \rightarrow Y$$

và k là khóa cần tìm.

Bài toán

- Cho một số cặp input/output ($m_i, c_i = E(k, m_i)$) với $i = 1, 2, \dots, q$.
- Hãy tìm khóa k .



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

7 / 23

Tiết lộ dữ liệu

- S và R chia sẻ khóa chung
- Vào ngày 10 tháng 1, S mã hóa thông điệp

$m =$ Hẹn gặp ngày mai lúc 5 giờ chiều

và gửi bản mã c đến cho R .

- Kẻ tấn công lấy được c
- Vào ngày 11 tháng 1, kẻ tấn công quan sát thấy có cuộc họp giữa S và R lúc 5 giờ chiều và do đó biết thông điệp m .



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4 / 23

6 / 23

Tìm kiếm vết cạn để tìm khóa cho mã khôi

Tấn công vết cạn để tìm khóa

```
tấn-công-vết-cạn ( $m_1, c_1$ )
for  $k = 1, 2, \dots, 2^{[K]}$ 
    if  $E(k, m_1) == c_1$  then return  $k$ 
```

Câu hỏi: Thuật toán trên liệu có giúp tìm khóa k đúng?

- Với hai cặp DES ($m_1, c_1 = DES(k, m_1)$) và ($m_2, c_2 = DES(k, m_2)$) xác suất để có k có duy nhất là $\approx 1 - 1/2^{71}$.
- Với AES-128: cho hai cặp input/output, xác suất có k duy nhất $\approx 1 - 1/2^{128}$
- Vậy hai cặp input/output là đủ thông tin để tìm kiếm vết cạn cho khóa.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

9 / 23

Kiểu tấn công

Cho $(m_1, c_1), \dots, (m_q, c_q)$ với $c_i = E(k, m_i)$.

Tấn công chọn thông điệp: Kẻ cần công A có thể lấy m_1, \dots, m_q , một cách thích nghi, tức là chọn m_i như một hàm theo $(m_1, c_1), \dots, (m_{i-1}, c_{i-1})$.

Alice
Kẻ tấn công

m_1

$c_1 = E(k, m_1)$

m_2

$c_2 = E(k, m_2)$

Bổ đề
Giả sử DES là một hệ mã lý tưởng

$$(2^{56} \text{ hoán vị ngẫu nhiên } \pi_i : \{0, 1\}^{64} \rightarrow \{0, 1\}^{64})$$

Khi đó, với mỗi cặp x, y có nhiều nhất **một** khóa k thỏa mãn

$$y = DES(k, x)$$

với xác suất $\geq 1 - 1/256 \approx 99.5\%$.

$$\begin{aligned} & \Pr[\exists k' \neq k \text{ thỏa mãn } c = DES(k, m) = DES(k', m)] \\ & \leq \sum_{k' \in \{0,1\}^{56}} \Pr[DES(k, m) = DES(k', m)] \\ & \leq 2^{56} \cdot \frac{1}{2^{64}} = \frac{1}{2^8}. \end{aligned}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

8 / 23



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11 / 23

Cải tiến DES chống tấn công vét cạn: Triple-DES

Tại sao không dùng double-DES?

Phương pháp 1: Triple-DES

- Xét $E : K \times X \rightarrow X$ là một hệ mã khối.
- Ta định nghĩa hệ mã khối

$$3E : K^3 \times X \rightarrow X$$

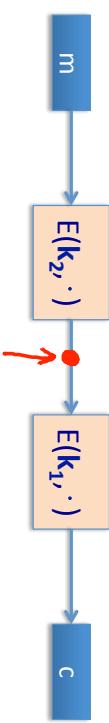
bởi:

$$3E((k_1, k_2, k_3), m) := E(k_3, D(k_2, E(k_3, m)))$$

Nhận xét

Nếu $k_1 = k_2 = k_3$ thì $3E = E$

13 / 23



Hình: $2E(k_1, k_2, m) := E(k_1, E(k_2, m))$

Ý tưởng tấn công double-DES: Tìm (k_1, k_2) thỏa mãn

$$E(k_1, E(k_2, m)) = c$$

tương đương với

$$E(k_2, m) = D(k_1, c).$$

15 / 23

Triple-DES: Một số nhận xét

Thử thách DES
Cho các cặp bản rõ và bản mã

```
msg = "The unknown messages is : XXXX ... "
CT =   c1   c2   c3   c4
```

Hãy tìm khóa $k \in \{0, 1\}^{56}$ thỏa mãn $DES(k, m_i) = c_i$ với $i = 1, 2, 3$.

- 1997: DESCHALL project với internet search – **96 ngày**
- 1998: EFF dùng máy DeepCrack – **3 ngày** (250K \$)
- 1999: Kết hợp cả DeepCrack và internet search – **22 giờ**
- 2006: COPACOBANA (120 FPGA) – **7 ngày** (10K \$).

Không nên dùng mã khóa 56 bit khóa !! (128-bit khóa $\Rightarrow 2^{72}$ ngày)

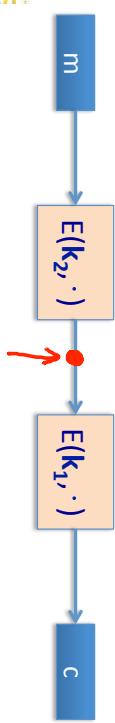
Thuật toán

- Xây dựng bảng

$k_0 = 00\dots 0$	$E(k_0, m)$
$k_1 = 00\dots 1$	$E(k_1, m)$
\dots	
$k_N = 11\dots 1$	$E(k_N, m)$

- Sắp xếp các phần tử của bảng theo cột thứ hai $E(k, m)$

- for $k \in \{0, 1\}^{56}$:
kiểm tra liệu $D(k, c)$ có nằm trong cột thứ hai của bảng
nếu có thì $E(k_i, m) = D(k, c) \Rightarrow (k_i, k) = (k_1, k_2)$.

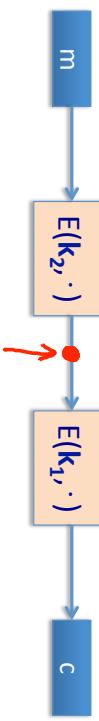


Chi phí tính toán

$$\text{Thời gian} = \overbrace{\text{Xây dựng bảng và sắp xếp}}^{56 \times 2^{56}} + \overbrace{\text{Tìm kiếm nhị phân trong bảng}}^{56 \times 2^{56}}$$

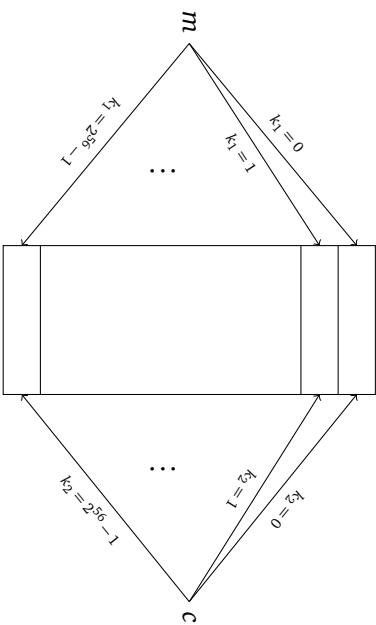
$$< 2^{63}$$

Không gian $\approx 2^{56}$



Ý tưởng tân công

T



Thuật toán

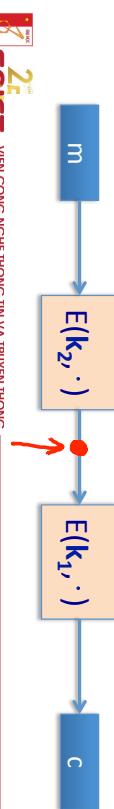
- Xây dựng bảng

$k_0 = 00\dots 0$	$E(k_0, m)$
$k_1 = 00\dots 1$	$E(k_1, m)$
\dots	
$k_N = 11\dots 1$	$E(k_N, m)$

Thuật toán

2^{56}

- Xây dựng bảng
- Sắp xếp các phần tử của bảng theo cột thứ hai $E(k, m)$
- for $k \in \{0, 1\}^{56}$:
kiểm tra liệu $D(k, c)$ có nằm trong cột thứ hai của bảng
nếu có thì $E(k_i, m) = D(k, c) \Rightarrow (k_i, k) = (k_1, k_2)$.



Thé nào là hệ mã khóa “tốt”?

Bài tập

Liệu cách xây dựng $E2((k_1, k_2), m)$ như dưới đây có hiệu quả hay không?

- $k_1 \oplus E(k_2, m)$
- $E(k_2, m \oplus k_1)$

Tính chất	Cần?	Đủ?
an toàn chống lại vết cạn khóa	Có	Không!
khó tìm m khi cho $c = E(k, m)$	Có	Không!
⋮		

Tà không thể nào định nghĩa hoặc hiểu tính an toàn nếu đưa ra một danh sách không xác định

Ta muốn một tính chất “chủ đạo” của hệ mã khóa để đảm bảo an toàn cho việc sử dụng mã khóa!

Phương pháp 2: DESX

Giới hạn của tính an toàn

Xét hệ mã

$$E : \{0,1\}^{128} \times \{0,1\}^{128} \rightarrow \{0,1\}^{128}$$

Định nghĩa EX bởi

$$EX((k_1, k_2, k_3), m) = k_1 \oplus E(k_2, m \oplus k_3)$$

- Với DESX: Độ dài khóa = $64 + 56 + 64 = 184$ bit

- Không thể tìm khóa nếu cho một số cặp bản rõ/bản mã.
- Nhưng hệ này là không an toàn!



Mã khóa lý tưởng

Ứng dụng của khoá dùng nhiều lần

- Trên thực tế, người ta xem AES hoặc 3DES như một hệ **mã**

khoá lý tưởng;

- Tức là, với mỗi khóa k , ánh xạ

$$F_k(x) = e(k, x)$$

là một hoán vị ngẫu nhiên độc lập từ $\{0, 1\}^{128}$ lên chính nó.

Mã hóa hệ thống file

- Mã hóa nhiều file dùng AES với cùng khóa

IPSec

- Nhiều gói tin cùng được mã hóa bằng AES với cùng một khóa



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

2/37



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4/37



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Nhập môn An Toàn Thông Tin

Sử dụng mã khóa

Các chế độ và mode sử dụng

Câu hỏi: Làm thế nào để mã hóa thông điệp với độ dài bất kỳ?
(dùng AES)

Trả lời: Dùng một trong các mode sau:

- “ECB” = “Electronic code book”
- “CTR” = “Counter mode”
- “CBC” = “Cipher Block Chaining”
- “OFB” = “Output Feedback”
- ...

Chế độ sử dụng: Khoá chỉ sử dụng một lần và **khoá dùng nhiều lần**.

1/37

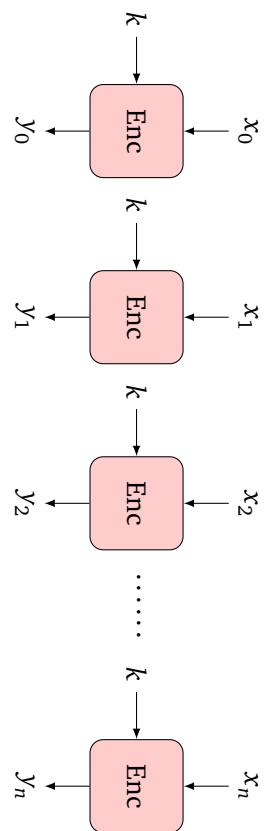


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

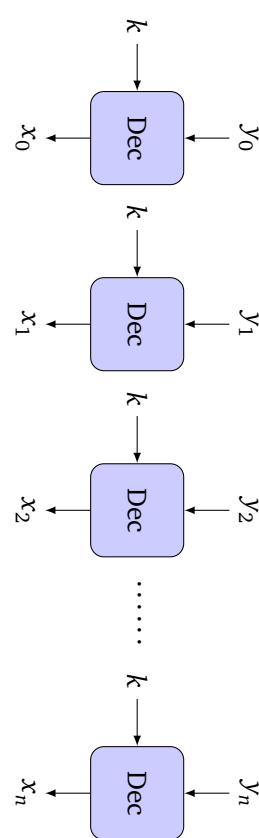
3/37

ECB (Electronic code book)

ECB: Làm thế nào để giải mã?

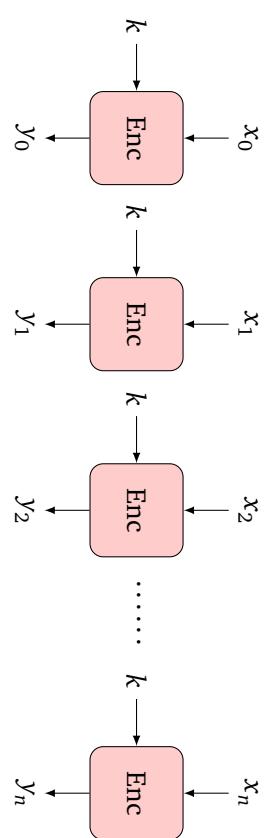


- Dữ liệu được chia thành các khối khói b bit, với $b =$ kích thước khối.
- Với dữ liệu không chia hết cho b bit: Thêm dãy “10..0” để đỡ dài thông điệp chia hết cho b .



Nội dung

① Electronic Codebook Mode (ECB)



② Cipher Block Chaining Mode (CBC)

- Mã dòng

- Tính an toàn

- Phép toán **padding** này cho có tính **khả nghịch**. Nó cho phép

ECB (Electronic code book)



Nội dung

Ví dụ: Chuyển tiền giữa hai ngân hàng dùng ECB

① Electronic Codebook Mode (ECB)

Block #	1	2	3	4	5
Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount \$	

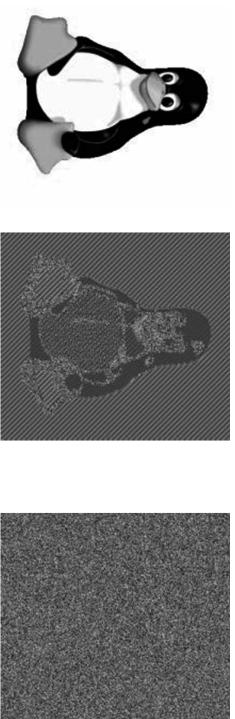
Hình: Giao thức trao đổi giữa các ngân hàng:

- ❶ **Giả sử:** Mỗi trường đều là n -bit (ví dụ 128 bit)
- ❷ **Giả sử:** Khoá k_{AB} để trao đổi thông tin giữa hai ngân hàng A và B là cố định.



9/37

ECB là không an toàn



Hình: Ảnh ở giữa là ECB mode, ảnh bên phải là mã hóa an toàn

- Vấn đề: Nếu $x_i = x_j$ thì $y_i = y_j$.

- ECB chỉ an toàn khi mã hóa dữ liệu ngẫu nhiên (ví dụ, các khóa).

② Cipher Block Chaining Mode (CBC)

③ Mã đóng

④ Tính an toàn



9/37

Oscar tân công

Block #	1	2	3	4	5
Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount \$	

- ❶ Oscar mở một tài khoản tại ngân hàng A và một tài khoản tại ngân hàng B ;
- ❷ Oscar chuyển nhiều lần 1\$ từ tài khoản của anh ta ở ngân hàng A sang tài khoản ở ngân hàng B ;
- ❸ Oscar bắt gói tin trên đường truyền và nhận được các bản mã giống nhau

$$B_1 \| B_2 \| B_3 \| B_4 \| B_5$$

và anh ta giữ lại bản mã B_4 .

- ❹ Trong tương lai, mỗi khi thấy lệnh chuyển tiền từ B_1 tới B_3 ,



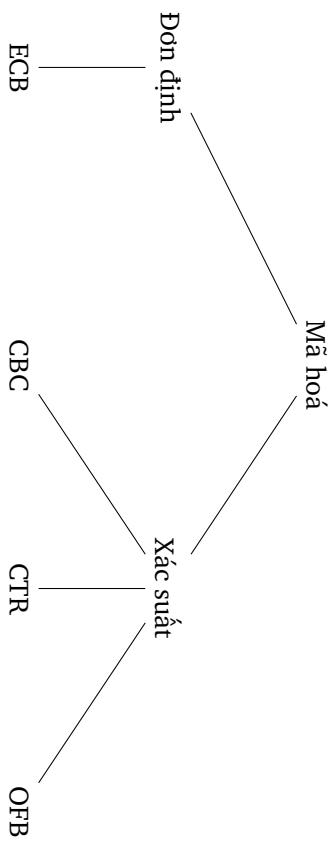
8/37

10/37

Mã hóa xác suất

Dạng mã hóa

- Mã hóa hai lần của cùng một thông điệp sẽ cho hai bản mã khác nhau
- Bản mã phải dài hơn bản rõ
- Nói một cách nôm na:
 $Kích thước bản mã = Kích thước bản rõ + "số bit ngẫu nhiên"$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

13 / 37

Bài toán

Bài tập

Hãy viết hàm giải mã D cho hàm mã hóa E được định nghĩa bởi:

$E(k, m)$:

$r = \text{random}()$

$c = \text{AES}(k, r) \oplus m$

$\text{return } (r, c)$

Ta cần giải quyết hai vấn đề:

- Làm cho hệ mã trở thành hệ mã xác suất;
- Ảnh hưởng của việc mã hóa trên mọi khối.



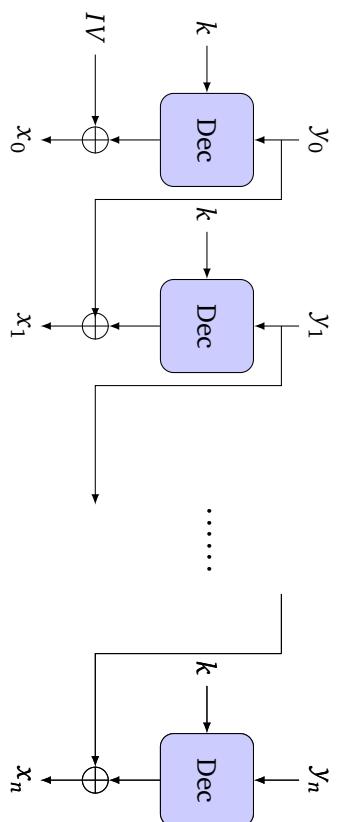
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

15 / 37

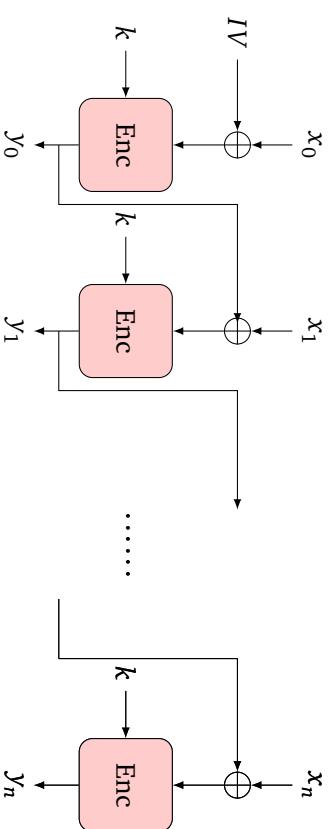
Sử dụng IV như thế nào?

CBC: Giải mã

- IV không cần giữ bí mật
 - Nhưng phải là “nonce” = “number used only once”
- Ví dụ
- ❶ Là ngẫu nhiên “thật”
 - ❷ Là bộ đếm “counter” (phải được lưu trữ bởi Alice)
 - ❸ $ID_A \parallel ID_B \parallel \text{time}$



CBC (Cipher Block Chaining mode)



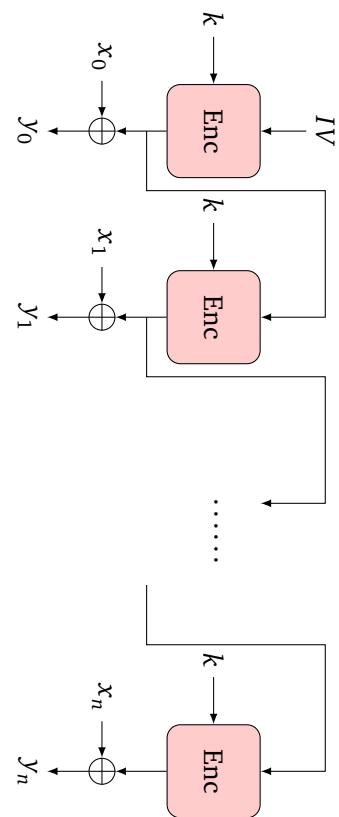
Thuật toán. Chọn IV (“initialization value”) một cách ngẫu nhiên, sau đó dùng y_i như “ IV ” cho M_{i+1} . Gửi IV cùng với bản mã

$IV \parallel y_0 \parallel y_1 \parallel \dots \parallel y_n$

Một kỹ thuật padding cho CBC

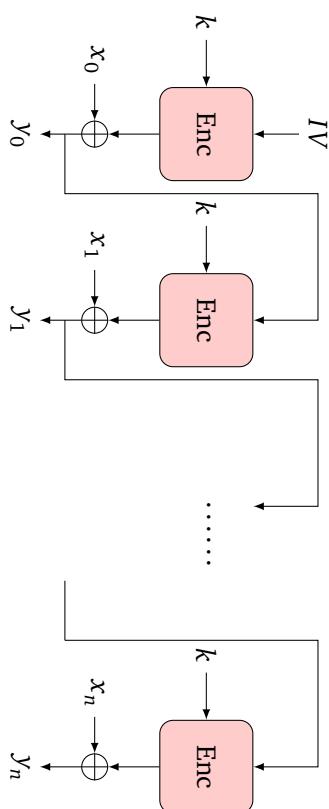
- Padding theo từng byte;
- Giá trị mỗi byte được thêm là số byte cần được thêm.
Ví dụ, nếu kích thước block là 8 và ta cần padding 4 byte:
... | DD DD DD DD 04 04 04 04 |
• nếu không cần padding, ta thêm một block giả.

Output Feedback Mode (OFB)



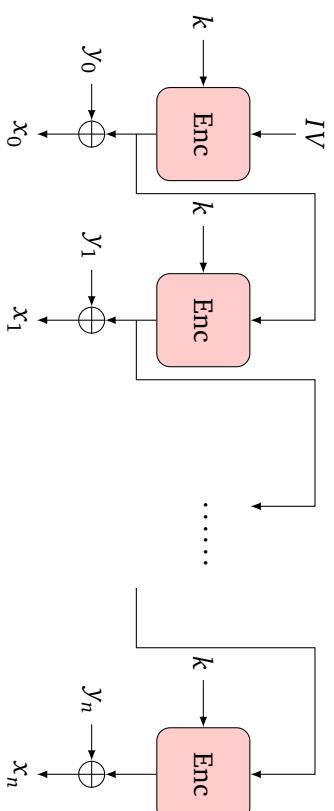
Thuật toán. Tương tự như CBC mode. Sử dụng IV ngẫu nhiên truyền cùng bản mã.
Nếu kích thước của bản rõ M không chia hết cho b , ta chỉ cần truyền bản mã rút gọn (không cần padding).

Cipher Feedback Mode (CFB)



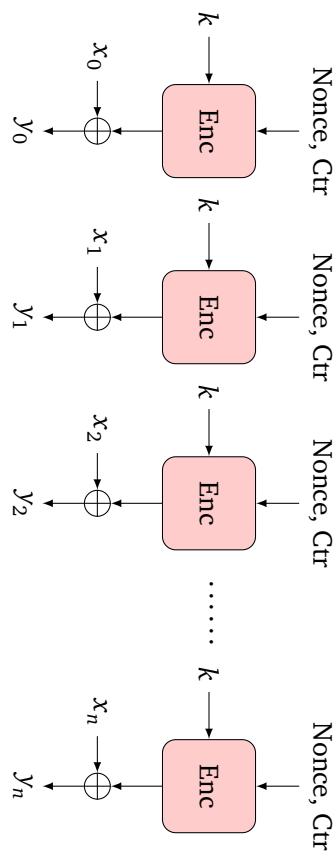
Nội dung

OFB: Giải mã



- ① Electronic Codebook Mode (ECB)
- ② Cipher Block Chaining Mode (CBC)
- ③ Mã dòng
- ④ Tính an toàn

Counter Mode (CTR)

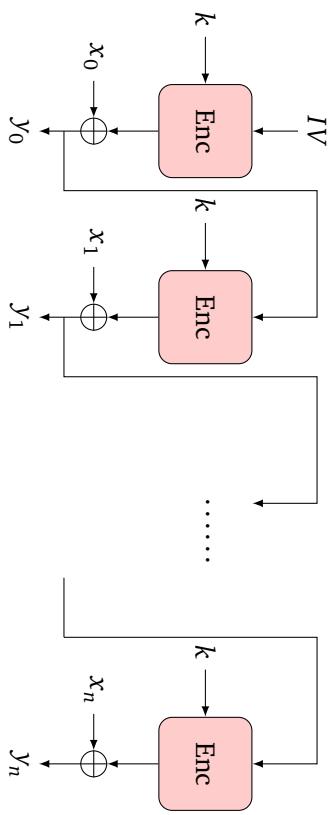


- Đảm bảo cặp Nonce||Ctr cặp không bao giờ lặp lại.
- Ctr được bắt đầu từ 0 cho mọi thông điệp.

25 / 37



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

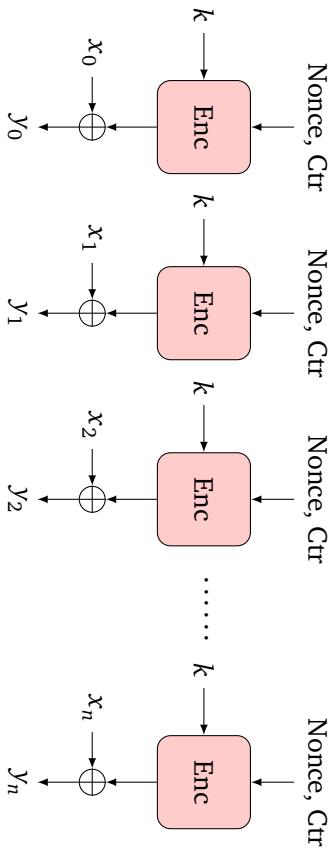


Bài tập
Hãy mô tả mạch giải mã CFB



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

26 / 37



Bài tập
Hãy mô tả mạch giải mã CTR

Bài tập
Xét thông điệp m gồm ℓ khối AES (ví dụ $\ell = 100$). Alice mã hóa m dùng CBC mode và truyền bản mã kết quả tới Bob. Do mang lỗi, khối bản mã số $\ell/2$ bị mất trong khi truyền. Mọi bản mã khác được truyền và nhận đúng. Khi Bob giải mã bản mã nhận được, bao nhiêu khối bản rõ sẽ bị mất?

Nội dung

Thử nghiệm IND-CCA

Xét K là khóa được chọn ngẫu nhiên. E_K là hàm mã hóa với khóa K . D_K là hàm giải mã.

1 Electronic Codebook Mode (ECB)

2 Cipher Block Chaining Mode (CBC)

3 Mã dòng

4 Tính an toàn

Pha I. ("Tìm kiếm")

- Kẻ tấn công có thể truy cập vào E_K, D_K như các hộp đen. (Có thể mã hóa/giải mã mọi thông điệp anh ta muốn)
- Kẻ tấn công đưa ra hai thông điệp M_0, M_1 cùng độ dài.

Pha II. ("Gọi ý")

- Ta bí mật chọn $d \leftarrow_s \{0, 1\}$ và tính $Y = E_K(M_d)$.

Nên sử dụng mode nào?

Bài tập

Xét thông điệp m bao gồm ℓ khối AES (ví dụ $\ell = 100$). Alice mã hóa m dùng randomized counter mode và truyền bản mã kết quả tới Bob. Do mạng lỗi, bản mã số $\ell/2$ bị mất trong khi truyền. Mọi khôi bản mã khác được truyền và nhận đúng. Khi Bob giải mã bản mã nhận được, bao nhiêu khôi bản rõ bị mất?

Mục đích. Nếu hệ mã khối là **không thể phân biệt** với hệ mã khối lý tưởng, thì mode sử dụng nên đảm bảo tính **không thể phân biệt** dựa trên **tấn công chọn bản mã**:

- Định nghĩa trò chơi với kẻ tấn công.
- Mode là IND-CCA an toàn nếu kẻ tấn công có thể thắng trong trò chơi với xác suất nhiều nhất chỉ là $1/2 + \epsilon$ với ϵ là nhỏ "không đáng kể".

Thử nghiệm IND-CCA

INC-CCA an toàn

Xét K là khóa được chọn ngẫu nhiên. E_K là hàm mã hóa với khóa K . D_K là hàm giải mã.

Pha I. (“Tìm kiếm”)

- Kẻ tấn công có thể truy cập vào E_K, D_K như các hộp đen. (Có thể mã hóa/giải mã mọi thông điệp anh ta muốn)
- Kẻ tấn công đưa ra hai thông điệp M_0, M_1 cùng độ dài.

Pha II. (“Gọi ý”)

- Tà bí mật chọn $d \leftarrow_s \{0, 1\}$ và tính $Y = E_K(M_d)$.
- Kẻ tấn công nhận Y , và có thể tiếp tục truy cập vào E_K và D_K (ngoại trừ trên Y).
- Kẻ tấn công tính toán và đưa ra d' là gọi ý cho d .

Thử nghiệm IND-CCA

Xét K là khóa được chọn ngẫu nhiên. E_K là hàm mã hóa với khóa K . D_K là hàm giải mã.

Pha I. (“Tìm kiếm”)

- Kẻ tấn công có thể truy cập vào E_K, D_K như các hộp đen. (Có thể mã hóa/giải mã mọi thông điệp anh ta muốn)
- Kẻ tấn công đưa ra hai thông điệp M_0, M_1 cùng độ dài.

Pha II. (“Gọi ý”)

- Tà bí mật chọn $d \leftarrow_s \{0, 1\}$ và tính $Y = E_K(M_d)$.
- Kẻ tấn công nhận Y , và có thể tiếp tục truy cập vào E_K và D_K (ngoại trừ trên Y).

IND-CCA an toàn

Hệ mã gọi là **an toàn chống lại tấn công CCA** (hay IND-CCA) nếu trong thử nghiệm IND-CCA, lợi thế của kẻ tấn công

$$\text{Adv} = |\Pr(d = d') - 1/2|$$

là nhỏ “không đáng kể”.

Sự kiện. Để là IND-CCA an toàn, phương pháp mã hóa phải ngẫu nhiên. !

Ngược lại, kẻ tấn công có thể mã hóa M_0 và M_1 rồi so sánh với y .

Các mode đã biết là không an toàn!

IND-CCA an toàn

Định lý. Các mode ECB, CTR, CBC, OFB **không** phải IND-CCA an toàn.

- **ECB.** Không ngẫu nhiên.

- **CTR.** Giá trị C_{tr} bắt đầu là ngẫu nhiên, nhưng nó được truyền dưới dạng bản rõ. Trong trường hợp này, kẻ tấn công có thể yêu cầu giải mã một khúc đầu của Y , và anh ta được khúc đầu của M_d .

- **CBC.** Tương tự CTR: IV ngẫu nhiên nhưng được truyền dưới dạng bản rõ. Kẻ tấn công có thể dùng kỹ thuật giải mã khúc đầu.

Các mode đã biết là không an toàn!

- **ECB.** Không ngẫu nhiên.

- **CTR.** Giá trị C_{tr} bắt đầu là ngẫu nhiên, nhưng nó được truyền dưới dạng bản rõ. Trong trường hợp này, kẻ tấn công có thể yêu cầu giải mã một khúc đầu của Y , và anh ta được khúc đầu của M_d .

Các mode đã biết là không an toàn!

- **ECB.** Không ngẫu nhiên.

- **CTR.** Giá trị C_{tr} bắt đầu là ngẫu nhiên, nhưng nó được truyền dưới dạng bản rõ. Trong trường hợp này, kẻ tấn công có thể yêu cầu giải mã một khúc đầu của Y , và anh ta được khúc đầu của M_d .

- **CBC.** Tương tự CTR: IV ngẫu nhiên nhưng được truyền dưới dạng bản rõ. Kẻ tấn công có thể dùng kỹ thuật giải mã khúc đầu.
- **OFB.** Tương tự. Kẻ tấn công có thể sử dụng kỹ thuật giải mã khúc đầu.

IND-CCA an toàn

Định lý: Các mode ECB, CTR, CBC, OFB **không** phải IND-CCA an toàn.

Chứng minh.

- Kẻ tấn công chọn $M_0 = 0^x$ và $M_1 = 1^x$ với x lớn.
- Khi đó $Y = E_K(M_d)$.
- Xét $Z = \text{nửa đầu của } Y$.
- Vì $Y \neq Z$ nên kẻ tấn công được phép yêu cầu tính $D_K(Z)$ trong Pha II.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

35 / 37

IND-CCA an toàn

Định lý: Các mode ECB, CTR, CBC, OFB **không** phải IND-CCA an toàn.

Chứng minh.

- Kẻ tấn công chọn $M_0 = 0^x$ và $M_1 = 1^x$ với x lớn.
- Khi đó $Y = E_K(M_d)$.
- Xét $Z = \text{nửa đầu của } Y$.
- Vì $Y \neq Z$ nên kẻ tấn công được phép yêu cầu tính $D_K(Z)$ trong Pha II.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

35 / 37

IND-CCA an toàn

Định lý: Các mode ECB, CTR, CBC, OFB **không** phải IND-CCA an toàn.

Chứng minh.

- Kẻ tấn công chọn $M_0 = 0^x$ và $M_1 = 1^x$ với x lớn.
- Khi đó $Y = E_K(M_d)$.
- Xét $Z = \text{nửa đầu của } Y$.
- Vì $Y \neq Z$ nên kẻ tấn công được phép yêu cầu tính $D_K(Z)$ trong Pha II.
- Vậy nó cho phép tính được một nửa đầu của M_d .
- Vậy kẻ tấn công luôn thắng.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

35 / 37

IND-CCA an toàn

Định lý: Các mode ECB, CTR, CBC, OFB **không** phải IND-CCA an toàn.

Chứng minh.

- Kẻ tấn công chọn $M_0 = 0^x$ và $M_1 = 1^x$ với x lớn.
- Khi đó $Y = E_K(M_d)$.
- Xét $Z = \text{nửa đầu của } Y$.
- Vì $Y \neq Z$ nên kẻ tấn công được phép yêu cầu tính $D_K(Z)$ trong Pha II.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

35 / 37



Có thể xây dựng IND-CCA an toàn?

Trả lời: Có.

yêu cầu khi xây dựng. Đưa ra bản mã Y của thông điệp M , kẻ tấn công không thể tạo được bản mã Z cho một thông điệp có liên quan.

Kỹ thuật. Sử dụng kết hợp tính bí mật và tính toàn vẹn thông điệp.

Hệ mã IND-CCA an toàn. Hệ mã có xác thực.

Ví dụ: Sơ đồ UFFE = “Unbalanced Feistel Encryption” của Desai (Crypto 2006).

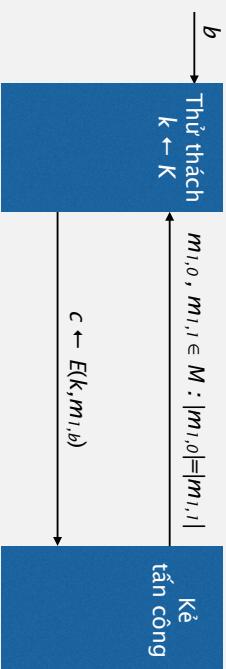
Ứng dụng

Mã hóa hệ thống file

- Mã hóa nhiều file dùng AES với cùng khóa

IPSec

- Nhiều gói tin cùng được mã hóa bằng AES với cùng một khóa



An toàn ngữ nghĩa cho khóa dùng nhiều lần

Xét $E = (E, D)$ là một hệ mà trên (K, M, \mathcal{O}) . Với $b = 0, 1$ ta định nghĩa $\text{EXP}(b)$ như sau:

2

4

An toàn ngữ nghĩa cho khóa dùng nhiều lần

Khóa được dùng nhiều lần

- Kẻ tấn công thấy nhiều bản rõ được mã hóa bởi cùng một khóa

Khả năng của kẻ tấn công

- Tấn công chọn bàn rõ = chosen-plaintext attack (CPA)
 - Có thể lấy được mã hóa của một số thông điệp mà anh ta muốn
 - Đây là mô hình thực tế

Mục đích của kẻ tấn công

- Phá được an toàn ngữ nghĩa

Nhập môn An toàn thông tin

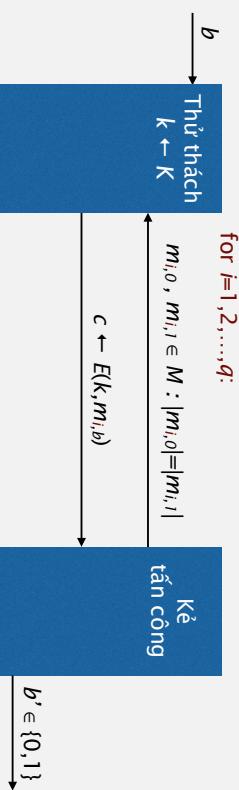
An toàn ngữ nghĩa cho khóa dùng nhiều lần



3

An toàn ngữ nghĩa cho khóa dùng nhiều lần

Xét $E = (E, D)$ là một hệ mã trên (K, M, \mathcal{O}) . Với $b=0,1$ ta định nghĩa $\text{EXP}(b)$ như sau:



Nếu kẻ tấn công muốn $c \leftarrow E(k, m)$ anh ta có thể gửi $m_{j,0} = m_{j,1} = m$

Định nghĩa. E là *an toàn ngữ nghĩa* dưới CPA nếu với mọi thuật toán “hiệu quả” A :

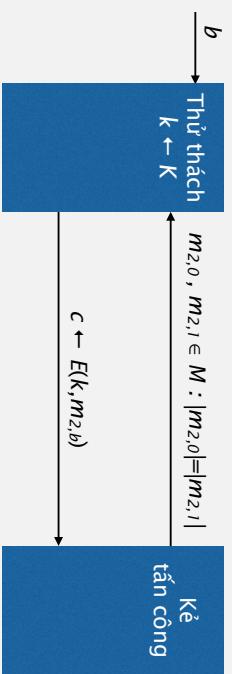
$$\text{Adv}_{\text{CPA}}[A, E] = |\Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1]|$$

là “không đáng kể”

6

An toàn ngữ nghĩa cho khóa dùng nhiều lần

Xét $E = (E, D)$ là một hệ mà trên (K, M, \mathcal{O}) . Với $b=0,1$ ta định nghĩa $\text{EXP}(b)$ như sau:



6

Mã hóa không an toàn dưới CPA

Giả sử $E(k, m)$ luôn cho cùng một bản mã cho thông điệp m . Vậy thì



8

Kẻ tấn công có thể kiểm tra được hai bản mã có phải là của cùng một bản rõ.

Có thẻ tấn công hệ mà khi không gian thông điệp M nhỏ.

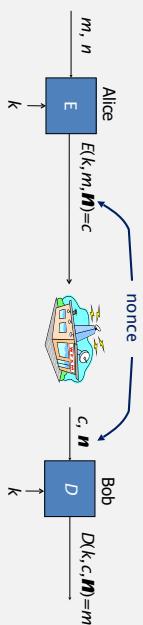
Giải pháp 2: Mã hóa dựa trên nonce

Câu hỏi

Xét PRF an toàn $F : K \times R \rightarrow M$. Ban đầu ta đặt $r = 0$.

Với $m \in M$ ta định nghĩa

$$E(k, m) = [r ++, \text{output } (r, F(k, r) \oplus m)]$$



Nonce n : một giá trị thay đổi theo các thông điệp

- Cặp (k, n) sẽ được dùng tối đa một lần

Phương pháp 1: nonce là một bộ đếm (ví dụ: đếm số gói tin)

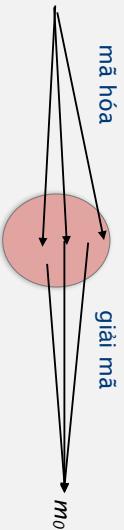
- được dùng khi bộ mã hóa giữ trạng thái thay đổi theo thông điệp
- nếu bộ giải mã có cùng trạng thái, không cần gửi nonce cùng với bản mã

Phương pháp 2: bộ mã hóa chọn nonce ngẫu nhiên

10

Giải pháp 1: Mã hóa xác suất

$E(k, m)$ là thuật toán ngẫu nhiên



- Mã hóa hai lần của cùng một thông điệp sẽ cho hai bản mã khác nhau

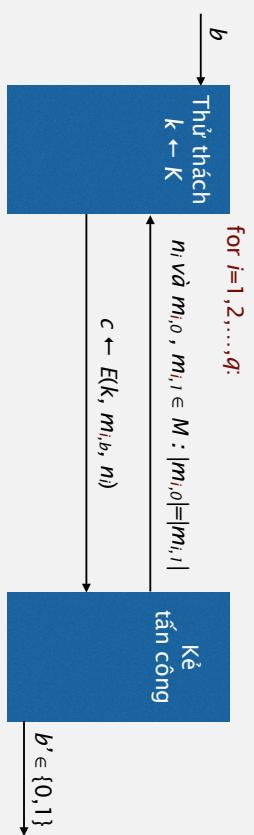
- Bản mã phải dài hơn bản rõ
- Nói một cách nôm na:

- Kích thước bản mã = Kích thước bản rõ + “số bit ngẫu nhiên”

Các nonce $\{n_1, \dots, n_q\}$ phải phân biệt

An toàn ngữ nghĩa cho các hệ mã dựa trên nonce

Hệ mã vẫn phải an toàn khi nonce được chọn bởi kẻ tấn công



Mã hóa hai lần của cùng một thông điệp sẽ cho hai bản mã khác nhau

- Bản mã phải dài hơn bản rõ

- Nói một cách nôm na:

- Kích thước bản mã = Kích thước bản rõ + “số bit ngẫu nhiên”

Định nghĩa. Hệ mã dựa trên nonce E là *an toàn ngữ nghĩa* dưới CPA nếu với mọi thuật toán “hiệu quả” A :

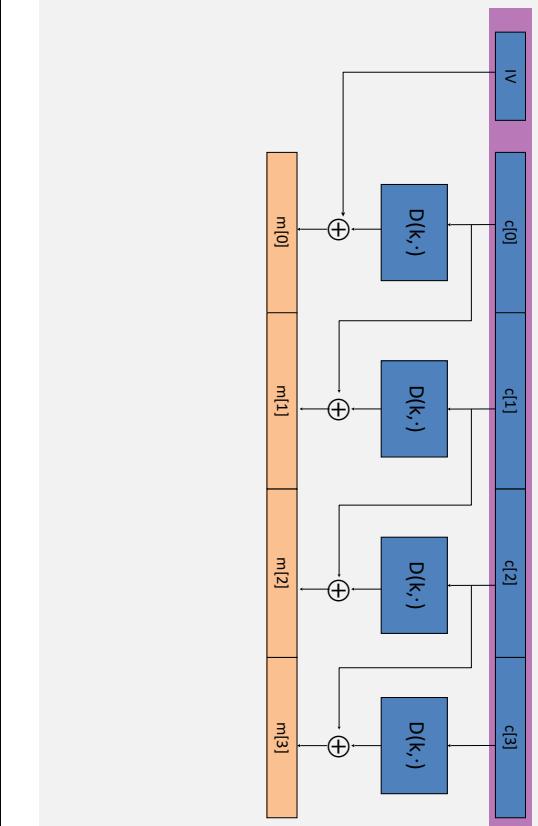
$$\text{Adv}_{\text{ncpa}}[A, E] = |\Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1]|$$

là “không đáng kể”

11

9

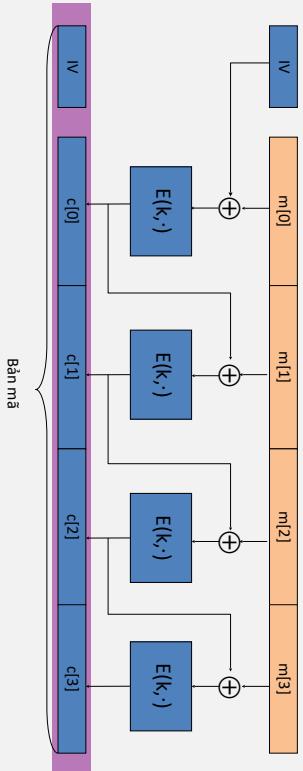
Mạch giải mã



14

Xây dựng 1: CBC với IV ngẫu nhiên

Xét (E, D) là một PRP. Chọn ngẫu nhiên $IV \in X$ và tính toán theo sơ đồ sau:

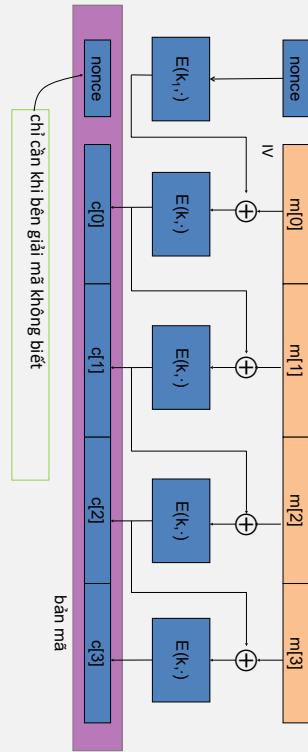


Bài tập. Hãy xây dựng mạch giải mã.

Xây dựng 1': CBC dựa trên nonce

CBC với nonce duy nhất : key = (k, k_1)

- Nonce duy nhất: cấp (key, n) dùng cho chỉ một thông điệp

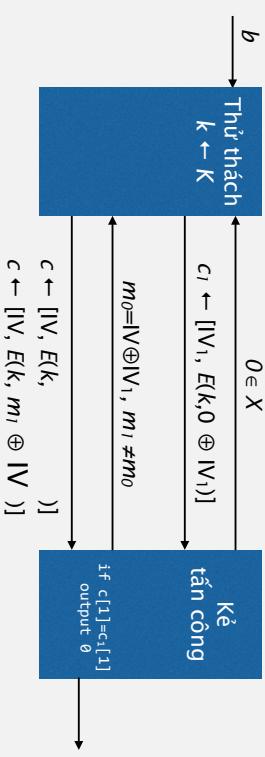


16

Chú ý: Tấn công CBC với IV ngẫu nhiên

Khi kẻ tấn công có thể dự đoán IV , vậy CBC không là CPA-an toàn !

Giả sử rằng biết $c \leftarrow E(k, m)$ ta có thể dự đoán IV cho thông điệp tiếp theo.

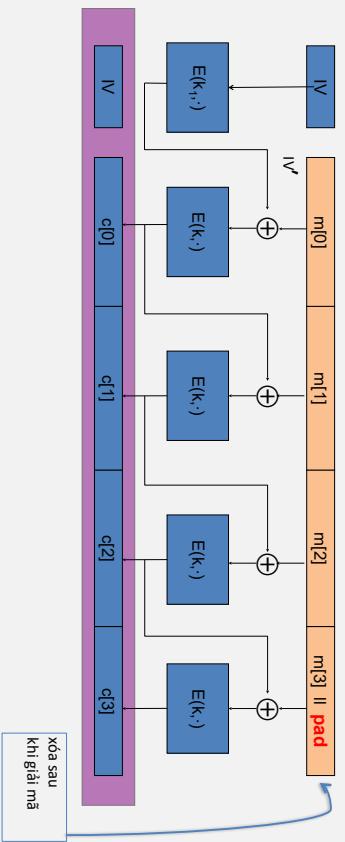


Lỗi trong SSL/TLS 1.0:

- IV cho bàn ghi thứ i là block cuối của bàn mã của bàn ghi thứ $i-1$.

15

Một kỹ thuật padding cho CBC



TLS: với $n > 0$, n byte pad là $\text{n n n} \dots \text{n}$

nếu không cần pad, ta thêm một block giả

18

Ví dụ về Crypto API



```
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);
```

Chú ý: Nếu **ivec** không lấy ngẫu nhiên thì ta phải mã hóa nó trước khi dùng.

Xây dựng 2: Rand CTR-mode

Xét một hệ mã khối an toàn $F : K \times \{0,1\}^n \rightarrow \{0,1\}^n$
 $E(k,m)$ được định nghĩa như sau: Chọn IV ngẫu nhiên và tính:



Chú ý: Khác với CBC-mode, CTR-mode cho phép song song hóa hiệu quả.

Bài tập: Xây dựng mạch giải mã.

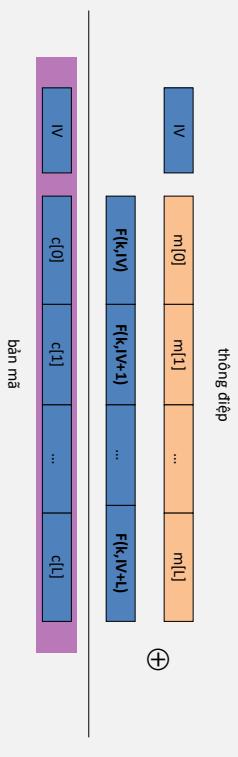
19

Sử dụng mã khóa

- PRP và PRF an toàn
- Định nghĩa an toàn cho khóa dùng nhiều lần

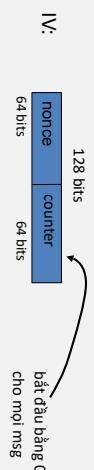
- CBC mode
- Rand CTR mode

Xây dựng 2: nonce CTR-mode



bản mã

Để đảm bảo rằng $F(k, x)$ không dùng khóa quá một lần, chọn IV như sau:



Nội dung

- Mã khối lý tưởng
- Chế độ ECB
- Mã hoá xác suất
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng



Nhập môn An Toàn Thông Tin

Các chế độ mã khối

- **Mã khối lý tưởng**
- Chế độ ECB
- Mã hoá xác suất
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng

Mã khối lý tưởng

- Trên thực tế, người ta xem AES hoặc 3DES như hệ mã khối lý tưởng $E(k, x)$.

- Tức là, với mỗi khóa k , ánh xạ
$$F_k(x) = E(k, x)$$
là một hoán vị ngẫu nhiên độc lập.



Nội dung

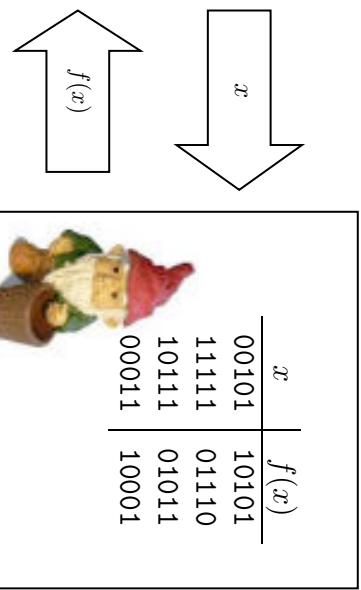
Các chế độ sử dụng

- **Câu hỏi:** Làm thế nào để mã hoá thông điệp với độ dài bất kỳ? (dùng AES hoặc 3DES)

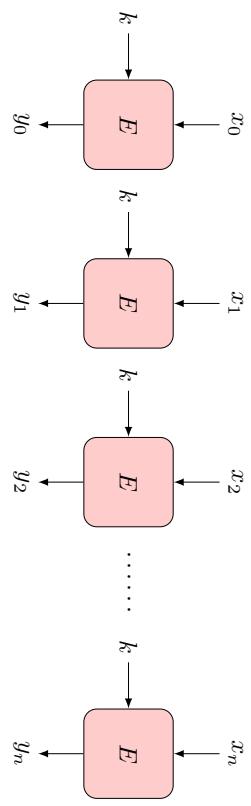
- **Trả lời:** Dùng một trong các chế độ sau:

- “ECB” = “Electronic code book”
- “CTR” = “Counter mode”
- “CBC” = “Cipher Block Chaining”
- “OFB” = “Output Feedback” • v.v.
- V.V.

Hoán vị ngẫu nhiên



ECB (Electronic code book)

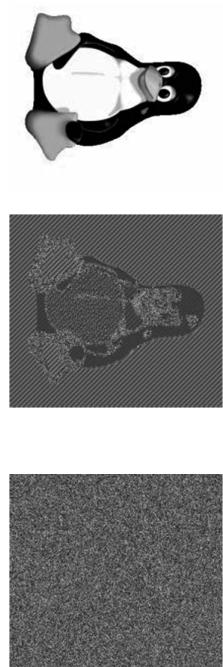


- Dữ liệu được chia thành các khối b bit, với $b =$ kích thước khối.
- Với dữ liệu không chia hết cho b bit: Thêm dãy “10..0” để dộ dài thông điệp chia hết cho b .
- Phép toán padding này cho có tính khả nghịch. Nó cho phép giải mã.

Nội dung

- Mã khối lý tưởng
- **Chế độ ECB**
- Mã hoá xác suất
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng

ECB không an toàn



Hình: Bên trái là Bản rõ. Ở giữa là chế độ ECB. Bên phải là Mã hóa an toàn

- **Vấn đề:** Nếu $x_i = x_j$ thì $y_i = y_j$
- ECB chỉ an toàn khi mã hoá dữ liệu ngẫu nhiên (Ví dụ, mã hoá các khoả).

Oscar tấn công

Block #	1	2	3	4	5
Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount \$	

1. Oscar mở một tài khoản tại ngân hàng A và một tài khoản tại ngân hàng B
2. Oscar chuyển nhiều lần 1\$ từ tài khoản của anh ta ở ngân hàng A sang tài khoản ở ngân hàng B
3. Oscar bắt gói tin trên đường truyền và nhận được các bản mã giống nhau

$$B_1 \parallel B_2 \parallel B_3 \parallel B_4 \parallel B_5$$

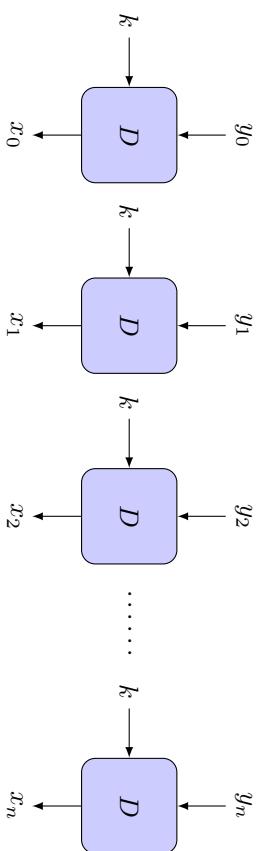
và anh ta giữ lại bản mã B_4

4. Trong tương lai, mỗi khi thấy lệnh chuyển tiền từ B_1 tới B_3 , thay block thứ 4 bởi B_4

ECB: giải mã

Ví dụ: Chuyển tiền giữa hai ngân hàng

Block #	1	2	3	4	5
Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount \$	



1. Giải sứ: kích thước mỗi trường là n-bit (ví dụ 128 bit)

2. Giải sứ: khóa k_{AB} để trao đổi thông tin giữa hai ngân hàng không thay đổi thường xuyên

Mã hóa xác suất

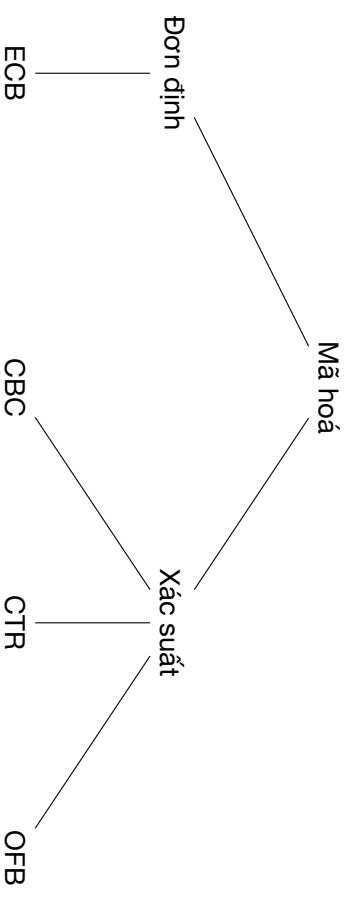
Dạng mã hóa

- Mã hóa hai lần của cùng một thông điệp sẽ cho hai bản mã khác nhau

- Bản mã phải dài hơn bản rõ

- Nói một cách nôm na:

Kích thước bản rõ + “dãy bit ngẫu nhiên”



Nội dung

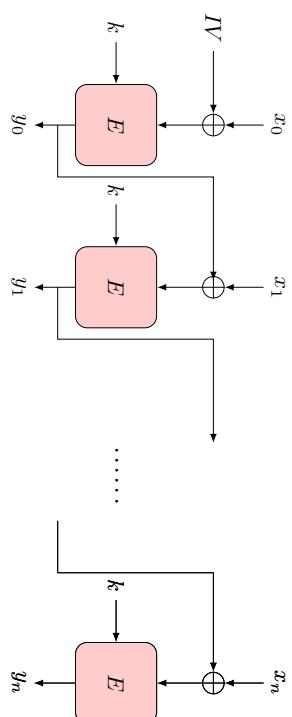
- Mã khối lý tưởng
- Chế độ ECB
- **Mã hóa xác suất**
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng

Bài tập

- Hãy viết hàm giải mã cho hàm mã hóa Enc được định nghĩa bởi

Enc(k , m):
 $r = \text{random}()$
 $c = \text{AES}(k, r) \oplus m$
return (r , c)

Chế độ CBC



Thuật toán. Chọn IV ("Initialization value") một cách ngẫu nhiên, sau đó dùng y_i như "IV" cho x_{i+1} . Gửi IV cùng với bản mã

$$IV \parallel y_0 \parallel y_1 \parallel \dots \parallel y_n$$

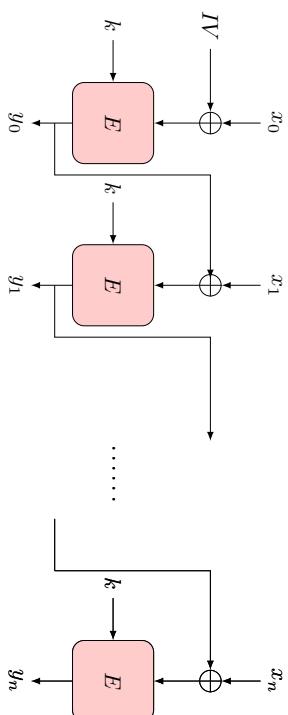
Nội dung

- Mã khối lý tưởng
- Chế độ ECB
- Mã hóa xác suất
- **Chế độ CBC**
 - Một số chế độ mã khối dựa trên mã dòng

Sử dụng IV như thế nào?

- IV không cần giữ bí mật
- Nhưng phải là "nonce" = "number used only once"
- **Ví dụ:** IV có thể là
 - ngẫu nhiên "thật"
 - bộ đếm "counter" (phải được lưu trữ bởi Alice)
 - ID_A || ID_B || time

CBC: công thức đại số



$$\bullet y_{-1} = IV \quad // Khởi tạo$$

$$\bullet y_i = E_k(y_{i-1} \oplus x_i) \text{ với } i = 0, 1, \dots$$

CBC: giải mã

Nội dung

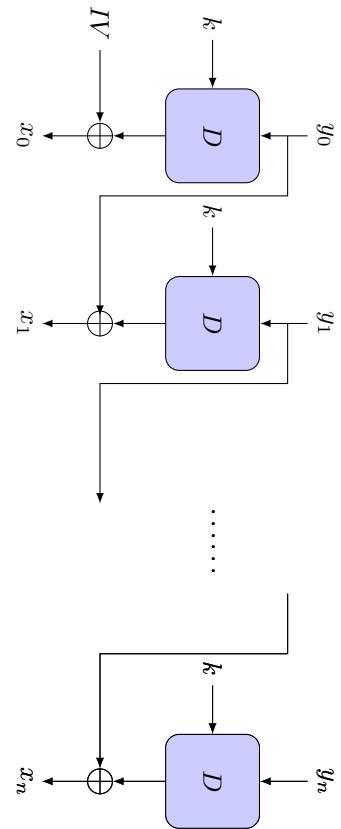
- Mã khối lý tưởng

- Chế độ ECB

- Mã hoá xác suất

- Chế độ CBC

- Một số chế độ mã khối dựa trên mã dòng



Padding cho CBC

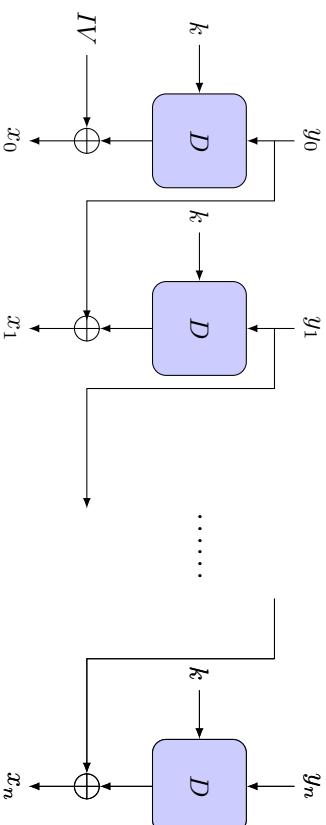
- Padding n byte, với $n > 0$,

n	n	n	n	\dots	n
-----	-----	-----	-----	---------	-----

- Nếu không cần pad, thêm một khối giả

- Khi giải mã, loại bỏ pad.

Bài tập

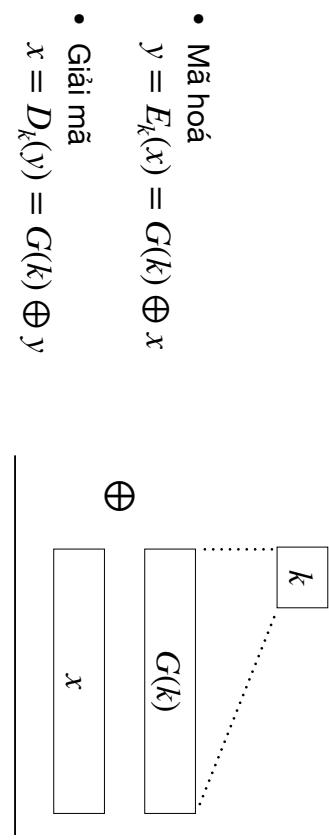


- Hãy viết công thức đại số cho mạch giải mã của chế độ CBC.

Mã dòng

Output Feedback (OFB)

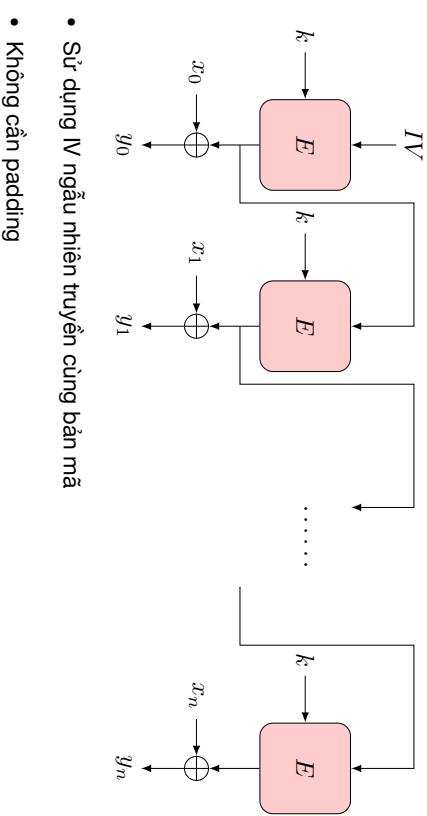
Chẽ độ



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Mã dòng

Mã dòng và mã khối

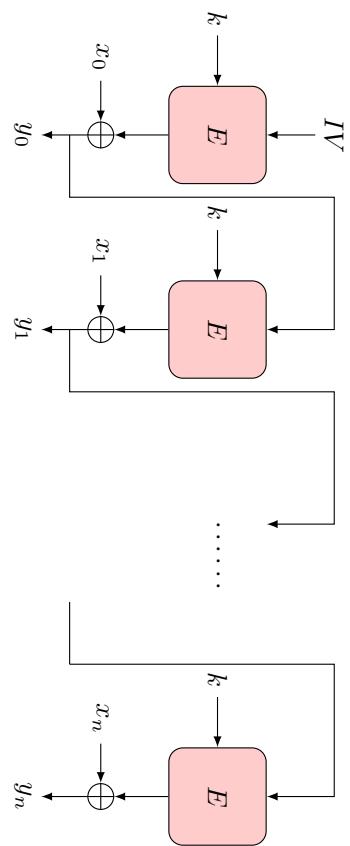


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

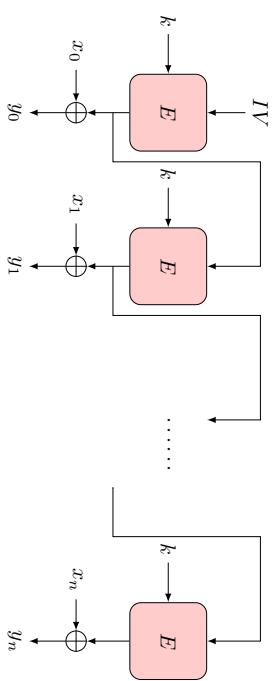
- Sử dụng một hàm sinh số giả ngẫu nhiên
 $G : \mathcal{X} \rightarrow \{0,1\}^n$,
là hàm đơn định từ không gian khoá đến dãy bit độ dài n
- Mã hoá
 $y = E_k(x) = G(k) \oplus x$
- Giải mã
 $x = D_k(y) = G(k) \oplus y$

Chẽ đố

Cipher Feedback (CFB)

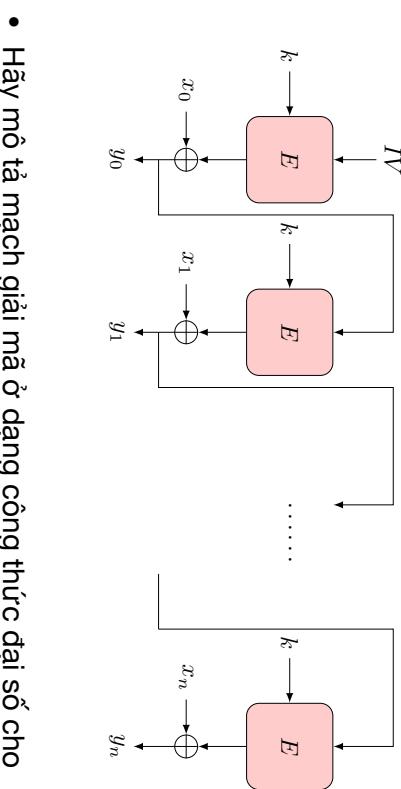


OFB: công thức đại số

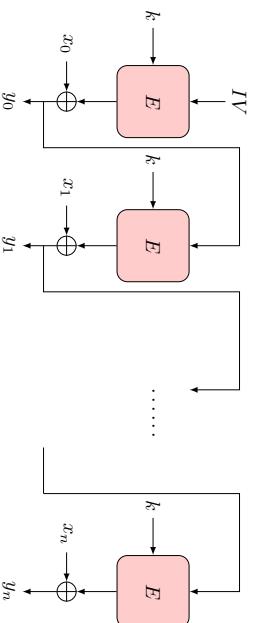


- $s_{-1} := IV$ // Khởi tạo
- $s_i := E_k(s_{i-1})$ // Khối bit giả ngẫu nhiên
- $y_i := s_i \oplus x_i$ với $i = 0, 1, 2, \dots$

Bài tập

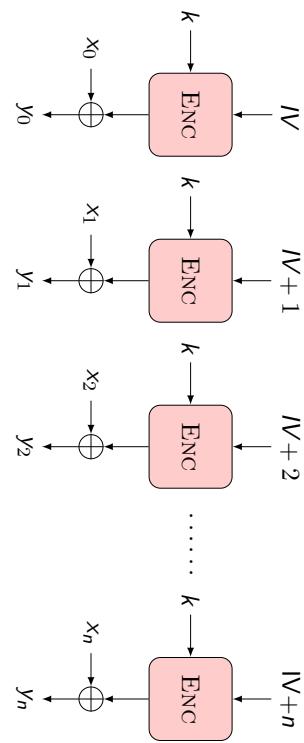


CFB: công thức đại số



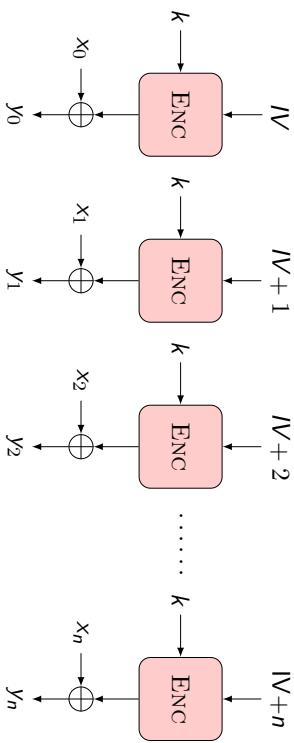
- $y_{-1} := IV$ // Khởi tạo
- $s_i := E_k(y_{i-1})$ // Khối bit giả ngẫu nhiên
- $y_i := s_i \oplus x_i$ với $i = 0, 1, 2, \dots$

Bài tập



- Hãy mô tả mạch giải mã cho chế độ CTR.

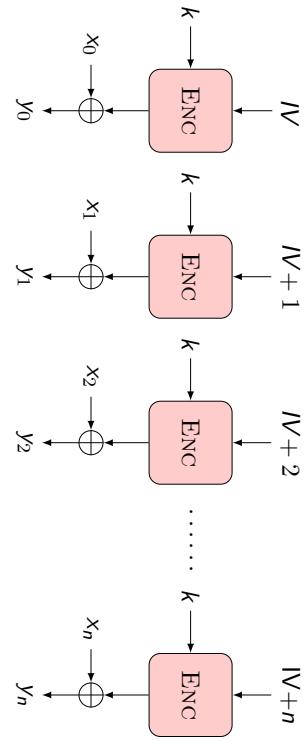
Chế độ Counter (CTR)



- Đảm bảo IV + Ctr không bao giờ lặp lại.

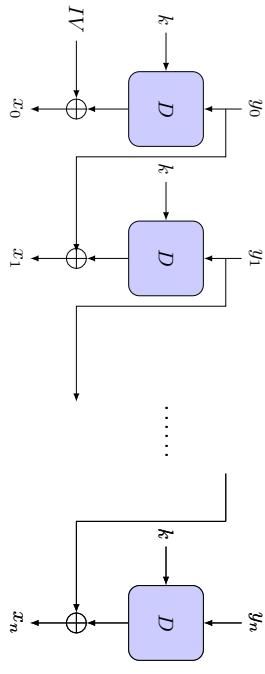
- Ctr được bắt đầu từ 0 cho mỗi thông điệp; và tăng ($Ctr = Ctr + 1$) sau mỗi khối của thông điệp.

Bài tập



- Xét thông điệp x gồm ℓ khối AES (ví dụ $\ell = 100$). Alice mã hóa x dùng chế độ CTR (với **None** ngẫu nhiên) và truyền bản mã kết quả tới Bob.
- Do mạng lõi, khối bản mã số $\ell/2$ bị mất trong khi truyền. Mọi bản mã khác được truyền và nhận đúng.
- Khi Bob giải mã bản mã nhận được, bao nhiêu khối bản rõ sẽ bị mất?

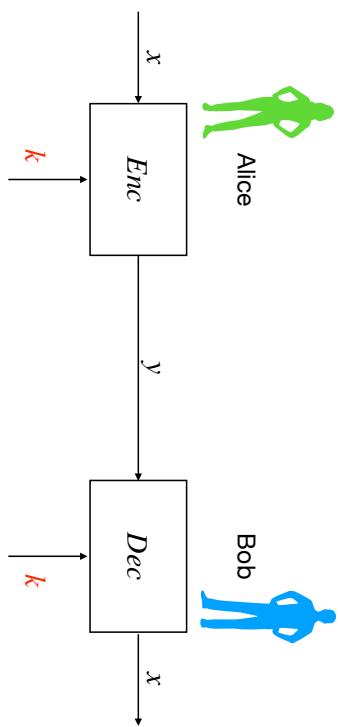
Bài tập



- Xét thông điệp x gồm ℓ khối AES (ví dụ $\ell = 100$). Alice mã hóa x dùng chế độ CBC và truyền bản mã kết quả tới Bob.
- Do mạng lõi, khối bản mã số $\ell/2$ bị mất trong khi truyền. Mọi bản mã khác được truyền và nhận đúng.
- Khi Bob giải mã bản mã nhận được, bao nhiêu khối bản rõ sẽ bị mất?

Mã hoá khoá đối xứng

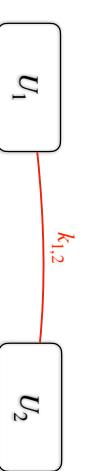
- Có n người dùng.
- Lưu trữ các cặp khoá phân biệt rất khó.



- Cùng khoá k cho cả việc mã hoá và giải mã;
- Hàm mã hoá và giải mã là tương tự (thậm chí trùng) nhau.

2/17

Vấn đề của mã khoá khoá đối xứng



Hình: $O(n)$ khoá cho mỗi người dùng

4/17

Mã hoá khoá đối xứng



Nhập môn An Toàn Thông Tin

Giới thiệu về Mật Mã Khoa Công Khai

- Thuật toán mã đối xứng AES hay 3DES rất an toàn, hiệu quả, và được dùng phổ biến; tuy nhiên
- Khoa đối xứng k phải được trao đổi an toàn!

1/17



3/17

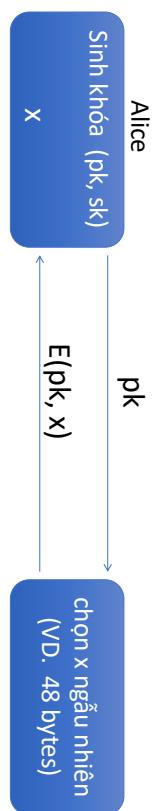
Mật mã khoá công khai

Whitfield Diffie, Martin Hellman, và Ralph Merkle năm 1976



6 / 17

Ứng dụng: Thiết lập khoá phiên

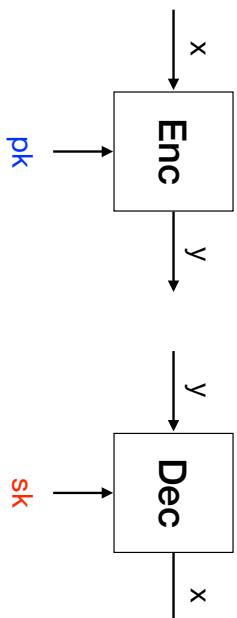


8 / 17

Vấn đề của mã hoá khoá đối xứng

- Alice và Bob có thể **lừa** nhau.
- Ví dụ: Alice có thể khẳng định rằng cô ấy chưa bao giờ đặt hàng TV trực tuyến từ Bob (anh ấy có thể đã bịa đặt hàng của cô ấy).
- Để ngăn chặn điều này: Tính “không chối bỏ” được.

Bob sinh cặp khóa $k = (\text{pk}, \text{sk})$ và đưa pk cho Alice.



Mật mã khoá công khai

Hàm cửa sập (Trapdoor functions - TDF)

Xây dựng mật mã khoá công khai từ TDF

Định nghĩa

Hàm cửa sập $X \rightarrow Y$ là bộ ba thuật toán hiệu quả (G, F, F^{-1})

- $G()$: thuật toán **ngẫu nhiên** output cặp khóa (pk, sk)
- $F(\text{pk}, \cdot)$: thuật toán **đơn định** định nghĩa một hàm $X \rightarrow Y$
- $F^{-1}(\text{sk}, \cdot)$: hàm từ $Y \rightarrow X$ tính nghịch đảo $F(\text{pk}, \cdot)$

Cụ thể: $\forall (\text{pk}, \text{sk})$ sinh bởi hàm $G()$, ta có

$$\forall x \in X : F^{-1}(\text{sk}, F(\text{pk}, x)) = x.$$

Mật mã khoá công khai

Định nghĩa

Một **hệ mật mã khoá công khai** là bộ ba thuật toán

$(G, \text{Enc}, \text{Dec})$

trong đó:

- $G()$: thuật toán **ngẫu nhiên** output cặp khóa (pk, sk)
- $\text{Enc}(\text{pk}, m)$: thuật toán **ngẫu nhiên** nhận $m \in M$ và output $c \in C$
- $\text{Dec}(sk, c)$: thuật toán **đơn định** nhận $c \in C$ và output $m \in M$ hoặc \perp

Tính đúng đắn: Với mọi (pk, sk) sinh bởi G :

$$\forall m \in M : \text{Dec}(\text{sk}, \text{Enc}(\text{pk}, m)) = m.$$

Bắt đầu từ

- (G, F, F^{-1}) là TDF an toàn;
- $(\text{Enc}_s, \text{Dec}_s)$ là hệ mã hoá đối xứng an toàn trên (K, M, C) ;
- $H : X \rightarrow Y$ là hàm băm.

Ta xây dựng hệ mật mã khoá công khai

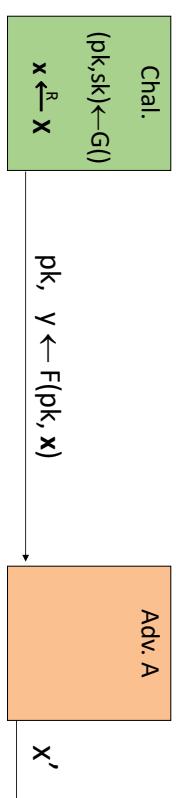
$(G, \text{Enc}, \text{Dec})$

với hàm sinh khoá chính là hàm G cho TDF;

Hàm cửa sập an toàn

(G, F, F^{-1}) là **an toàn** nếu $F(\text{pk}, \cdot)$ là hàm “một chiều”;

có thể tính xuôi, nhưng không thể tính nghịch đảo mà không có sk



Định nghĩa

(G, F, F^{-1}) là TDF **an toàn** nếu với mọi thuật toán hiệu quả A :

$$\Pr[x = x'] \text{ là “nhỏ không đáng kể”}$$

Sử dụng không đúng hàm của sập

Độ dài khoá và mức an toàn

Không mã hóa bằng cách áp dụng F để mã hóa bẩn rỗ:

$\text{Enc}(\text{pk}, m)$:
return $c = F(\text{pk}, m)$

$\text{Dec}(\text{sk}, c)$:
return $m = F^{-1}(\text{sk}, c)$

Vấn đề:

- Đây là hệ mã đơn định: không an toàn !
- Tồn tại nhiều cách tấn công



14/17

Xây dựng mật mã khoá công khai từ TDF

- (G, F, F^{-1}) là TDF an toàn;
- $(\text{Enc}_s, \text{Dec}_s)$ là hệ mã hoá đối xứng an toàn trên (K, M, C) ;
- $H : X \rightarrow Y$ là hàm băm.

$\text{Enc}(\text{pk}, m)$:
 $x \leftarrow_s X,$
 $y = F(\text{pk}, x)$
 $k = H(x),$
return (y, c)

$\text{Dec}(\text{sk}, (y, c))$:
 $x = F^{-1}(\text{sk}, y),$
 $k = H(x),$
return (y, c)

(Tính mũ a^x : **dễ**)

- Dương cong Elliptic**(EC) (ECDH, ECDSA, …): tổng quát hoá của bài toán Logarit rời rạc



16/17

Một số hàm “một chiều”

Các hệ mật khoá công khai dựa trên các **hàm một chiều**:
Tính f **dễ**, tính f^{-1} là **khó**!

- Phân tích thừa số nguyên tố** (RSA, …):
Cho hợp số n , tìm các thừa số nguyên tố của n
(nhân hai số nguyên tố: **dễ**)
- Logarit rời rạc** (Diffie Hellman, Elgamal, DSA, …):
Cho a, y , và m , tìm x thoả mãn
$$a^x = y \pmod{m}$$



13/17



Nội dung

Định nghĩa

- Ước chung của hai số nguyên a và b là số nguyên d thỏa mãn:

$$d \mid a \quad \text{và} \quad d \mid b.$$

1 Thuật toán Euclid

2 Thuật toán tính luỹ thừa

- $\gcd(12, 18) = 6$ vì $6 \mid 12$ và $6 \mid 18$ và không có số nào lớn hơn có tính chất này.

$$\gcd(748, 2014) = 44 \text{ vì}$$

các ước của $748 = \{1, 2, 4, 11, 17, 22, 34, 44, 68, 187, 374, 748\}$,
các ước của $2014 = \{1, 2, 4, 8, 11, 22, 23, 44, 46, 88, 92, 184, 253,$
 $506, 1012, 2024\}$.



Viện Công nghệ Thông tin và Truyền thông



Viện Công nghệ Thông tin và Truyền thông

3 / 34



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Định nghĩa

- Ước chung của hai số nguyên a và b là số nguyên d thỏa mãn:

$$d \mid a \quad \text{và} \quad d \mid b.$$

- Ta ký hiệu $\gcd(a, b)$ là ước chung **lớn nhất** của a và b .

Nhập môn An Toàn Thông Tin

Nhắc lại một số thuật toán trong lý thuyết số

1 / 34



Viện Công nghệ Thông tin và Truyền thông

3 / 34

Định lý (Thuật toán Euclid)

Xét a, b là hai số nguyên dương với $a \geq b$. Thuật toán sau đây tính $\gcd(a, b)$ sau một số hữu hạn bước.

- ❶ Đặt $r_0 = a$ và $r_1 = b$.
- ❷ Đặt $i = 1$.
- ❸ Chia r_{i-1} cho r_i , ta được

$$r_{i-1} = r_i \cdot q_i + r_{i+1} \quad \text{với} \quad 0 \leq r_{i+1} < r_i.$$

- ❹ Nếu $r_{i+1} = 0$, vậy thì

$$r_i = \gcd(a, b)$$

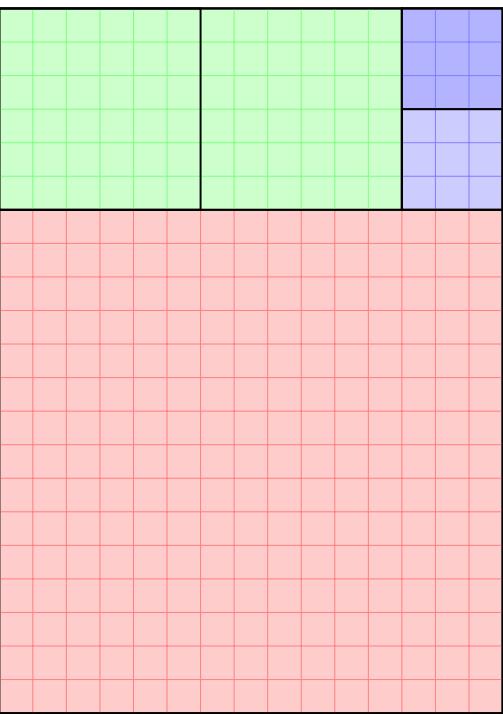
và thuật toán kết thúc.

- ❺ Ngược lại, $r_{i+1} > 0$, vậy thì đặt $i = i + 1$ và quay lại Bước 3.

Thuật toán Euclid (dạng đệ quy)

```
EUCLID(a, b)
if b == 0
    return a
else
    return EUCLID(b, a mod b)
```

$$\gcd(21, 15) = \gcd(15, 6) = \gcd(6, 3)$$



Định lý

Phép chia (Bước 3) của Thuật toán Euclid thực hiện nhiều nhất

$$\log_2(b) + 2 \quad \text{lần.}$$

Thuật toán Euclid mở rộng

Ví dụ

- Input : Cặp số nguyên dương (a, b)
- Output: Bộ ba (d, x, y) thỏa mãn

$$d = \gcd(a, b) = ax + by.$$

```
EXTENDED-EUCLID(a, b)
  if b == 0
    return (a, 1, 0)
  else
    (d', x', y') = EXTENDED-EUCLID(b, a mod b)
    (d, x, y) = (d', y', x' - ⌊a/b⌋y')
  return (d, x, y)
```



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

9/34

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Thuật toán Euclid mở rộng

- Thuật toán Euclid có thể mở rộng để tìm thêm một số thông tin.
- Cụ thể, chúng ta mở rộng thuật toán để tính thêm hệ số x, y thỏa mãn
$$d = \gcd(a, b) = ax + by.$$
- Các hệ số x, y có thể âm hoặc bằng 0. Các hệ số này sẽ có ích sau này khi tích phân tử nghịch đảo trong số học modun.

Tính đúng đắn của thuật toán

- Thuật toán tìm (d, x, y) thỏa mãn
$$d = \gcd(a, b) = ax + by$$
- Nếu $b = 0$, vậy thì
$$d = a = a \cdot 1 + b \cdot 0.$$
- Nếu $b \neq 0$, thuật toán EXTENDED-EUCLID sẽ tính (d', x', y') thỏa mãn

$$\begin{aligned} d' &= d = \gcd(b, a \bmod b) \\ &= bx' + (a \bmod b)y' \end{aligned}$$

- Vậy vậy thì

$$\begin{aligned} d &= b'x' + (a - b\lfloor a/b \rfloor)y' \\ &= ay' + b(x' - \lfloor a/b \rfloor y') \end{aligned}$$

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			
3	0	-	3	1	0
			3	0	-
			3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			
3	0	-			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			
3	0	-			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bô ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$

$$y = x' - \lfloor a/b \rfloor y'$$

11/34

Ví dụ

Bài tập
Hãy tính giá trị

$$(d, x, y) = \text{EXTENDED-EUCLID}(899, 493).$$

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bô ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$

$$y = x' - \lfloor a/b \rfloor y'$$

12/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bô ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$

$$y = x' - \lfloor a/b \rfloor y'$$

11/34

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bô ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$x = y'$$

$$y = x' - \lfloor a/b \rfloor y'$$

11/34

Tính nghịch đảo theo modun

Ví dụ: Tính $5^{-1} \bmod 12$

- Input : Số $n > 0$ và số $a \in \mathbb{Z}_n$ sao cho $\gcd(a, n) = 1$
- Output: Số b thoả mãn $a \cdot b = 1 \bmod n$.

MOD-INV (a, n)

(d, x, y) = EXTENDED-EUCLID (a, n)

$$b = x \bmod n$$

return b

$$\begin{array}{r} a & b & \lfloor a/b \rfloor & d & x & y \\ \hline 5 & 12 & 0 \\ 12 & 5 & 2 \end{array}$$

Tính nghịch đảo

Ví dụ: Tính $5^{-1} \bmod 12$

- Xét $n > 1$, nếu $\gcd(a, n) = 1$ thì ta có

$$\gcd(a, n) = 1 = ax + ny$$

- Vậy $ax = 1 \pmod{n}$. Tức là

$$x = a^{-1} \pmod{n}$$

$$\begin{array}{r} a & b & \lfloor a/b \rfloor & d & x & y \\ \hline 5 & 12 & 0 \end{array}$$

Ví dụ: Tính $5^{-1} \bmod 12$

Ví dụ: Tính $5^{-1} \bmod 12$

$$\begin{array}{r} a & b & \lfloor a/b \rfloor & d & x & y \\ \hline 5 & 12 & 0 & & & \\ 12 & 5 & 2 & & & \\ 5 & 2 & 2 & & & \\ 2 & 1 & 2 & & & \\ \hline 1 & 0 & - & 1 & 1 & 0 \end{array}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

15 / 34

Ví dụ: Tính $5^{-1} \bmod 12$

$$\begin{array}{r} a & b & \lfloor a/b \rfloor & d & x & y \\ \hline 5 & 12 & 0 & & & \\ 12 & 5 & 2 & & & \\ 5 & 2 & 2 & & & \\ 2 & 1 & 2 & & & \\ \hline 1 & 0 & - & 1 & 1 & 0 \end{array}$$

Ví dụ: Tính $5^{-1} \bmod 12$

$$\begin{array}{r|rrrrrr} a & b & \lfloor a/b \rfloor & d & x & y \\ \hline 5 & 12 & 0 & 1 & 5 & -2 \\ 12 & 5 & 2 & 1 & -2 & 5 \\ 5 & 2 & 2 & 1 & 1 & -2 \\ 2 & 1 & 2 & 1 & 0 & 1 \\ 1 & 0 & - & 1 & 1 & 0 \end{array}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

15 / 34

Ví dụ: Tính $5^{-1} \bmod 12$

$$\begin{array}{r|rrrrrr} a & b & \lfloor a/b \rfloor & d & x & y \\ \hline 5 & 12 & 0 & 1 & 5 & -2 \\ 12 & 5 & 2 & 1 & -2 & 5 \\ 5 & 2 & 2 & 1 & 1 & -2 \\ 2 & 1 & 2 & 1 & 0 & 1 \\ 1 & 0 & - & 1 & 1 & 0 \end{array}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

15 / 34

Ví dụ: Tính $5^{-1} \bmod 12$

$$\begin{array}{r|rrrrrr} a & b & \lfloor a/b \rfloor & d & x & y \\ \hline 5 & 12 & 0 & 1 & 5 & -2 \\ 12 & 5 & 2 & 1 & -2 & 5 \\ 5 & 2 & 2 & 1 & 1 & -2 \\ 2 & 1 & 2 & 1 & 0 & 1 \\ 1 & 0 & - & 1 & 1 & 0 \end{array}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

15 / 34

Tính lũy thừa nhanh

Thuật toán tính nhanh $a^b \pmod{n}$

Ví dụ

Gia sử ta muốn tính

$$3^{218} \pmod{1000}.$$

Đầu tiên, ta viết 218 ở dạng cơ số 2:

$$218 = 2 + 2^3 + 2^4 + 2^6 + 2^7.$$

Vậy thì 3^{218} trở thành

$$3^{218} = 3^{2+2^3+2^4+2^6+2^7} = 3^2 \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^6} \cdot 3^{2^7}.$$

Để ý rằng, để tính các mũ

$$3, 3^2, 3^{2^2}, 3^{2^3}, 3^{2^4}, \dots$$

17/34



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

19/34

Nội dung

Ví dụ (tiếp)

Ta lập bảng

i	0	1	2	3	4	5	6	7
$3^{2^i} \pmod{1000}$	3	9	81	561	721	841	281	961

rồi tính

$$3^{218} = 3^2 \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^6} \cdot 3^{2^7}$$

$$\equiv 9 \cdot 561 \cdot 721 \cdot 281 \cdot 961 \pmod{1000}$$

$$\equiv 489 \pmod{1000}.$$

② Thuật toán tính luỹ thừa

③ Nhóm vòng và phần tử sinh

Thuật toán tính nhanh $a^b \pmod{n}$

Modular-Exponentiation(a, b, n)

$c = 0$

$d = 1$

Biểu diễn $b = \langle b_k, b_{k-1}, \dots, b_0 \rangle_2$

for $i = k$ downto 0

$c = 2c$

$d = (d \cdot d) \pmod{n}$

if $b_i == 1$ then

$c = c + 1$

$d = (d \cdot a) \pmod{n}$

return d

- Giá trị của c bằng $\langle b_k, b_{k-1}, \dots, b_{l+1} \rangle_2$
- và $d = a^c \pmod{n}$.

Thuật toán đê quy tính $a^b \pmod{n}$

Modular-Exponentiation(a, b, n)

if $b == 0$ then return 1

if $b == 1$ then return a

$r = \text{Modular-Exponentiation}(a, b/2, n)$

$r = r * r$

if $b \bmod 2 == 1$ then $r = r * a$

return r

Thuật toán tính nhanh $a^b \pmod{n}$

Modular-Exponentiation(a, b, n)

$c = 0$

$d = 1$

Biểu diễn $b = \langle b_k, b_{k-1}, \dots, b_0 \rangle_2$

for $i = k$ downto 0

$c = 2c$

$d = (d \cdot d) \pmod{n}$

if $b_i == 1$ then

$c = c + 1$

$d = (d \cdot a) \pmod{n}$

return d

- Giá trị của c bằng $\langle b_k, b_{k-1}, \dots, b_{l+1} \rangle_2$

Ví dụ

Tính $7^{560} \pmod{561}$

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
d	7	49	157	526	160	241	298	166	67	1

- Kết quả tính $a^b \pmod{n}$ với

$$a = 7, \quad b = 560 = \langle 1000110000 \rangle_2, \quad \text{và } n = 561$$

- Kết quả cuối cùng bằng 1

Nội dung

Cấp của một phần tử

1 Thuật toán Euclid

2 Thuật toán tính luỹ thừa

3 Nhóm vòng và phần tử sinh

Xét G là một nhóm (hữu hạn) với phần tử đơn vị 1.

Định nghĩa

Cấp của phần tử $g \in G$, ký hiệu $o(g)$, là số nguyên $n \geq 1$ nhỏ nhất thoả mãn $g^n = 1$.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Nhóm con

Bài tập

Giả sử bạn biết $\varphi(n)$, hãy chỉ ra cách tính $a^{-1} \bmod n$ cho mọi $a \in \mathbb{Z}_n^*$ dùng thuật toán Modular-Exponentiation.

Gợi ý: Nhắc lại rằng $a^{\varphi(n)} = 1 \bmod n$.

Định nghĩa

Xét nhóm G và $S \subseteq G$. Khi đó S được gọi là **nhóm con** của G nếu S là một nhóm dưới phép toán của G .

Ví dụ

Xét $G = \mathbb{Z}_{11}^*$ và $S = \{1, 2, 3\}$. Khi đó S không phải là nhóm con vì

- $2 \cdot 3 \bmod 11 = 6 \notin S$, vi phạm tính chất đóng.
- $3^{-1} \bmod 11 = 4 \notin S$, vi phạm tính khả nghịch.

Tuy nhiên $\{1, 3, 4, 5, 9\}$ là một nhóm con. Bạn có thể kiểm tra!



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Nhóm con sinh bởi $g \in G$

Xét $G = \mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

Định nghĩa
Cho phần tử $g \in G$ có cấp n , ta đặt

$$\langle g \rangle = \{g^0, g^1, \dots, g^{n-1}\}.$$

Đây là một nhóm con của g và cấp của nó chính là $o(g) = n$.

Khi đó

$$\begin{aligned}\langle 2 \rangle &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\ \langle 5 \rangle &= \{1, 3, 4, 5, 9\}.\end{aligned}$$

Xác định cấp của phần tử

Xét $G = \mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

i	0	1	2	3	4	5	6	7	8	9	10
$2^i \bmod 11$	1	2	4	8	5	10	9	7	3	6	1
$5^i \bmod 11$	1	5	3	4	9	1	5	3	4	9	1

Cấp $o(a)$ của phần tử a là số $n \geq 1$ nhỏ nhất sao cho $a^n = 1$. Bởi vậy

- $o(2) = 10$
- $o(5) = 5$.

Cấp của nhóm con

Mệnh đề

Cấp $|S|$ của nhóm con $S \subseteq G$ luôn là ước của cấp $|G|$ của nhóm G .

Mệnh đề

Cấp $o(g)$ của g luôn là ước của $|G|$.

Ví dụ

Nếu $G = \mathbb{Z}_{11}^*$ thì

- $|G| = 10$
- $o(2) = 10$ là ước của 10
- $o(5) = 5$ là ước của 10

Nhóm con sinh bởi một phần tử

Logarit rời rạc

Phân tử sinh

Xét $G = \mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

i	0	1	2	3	4	5	6	7	8	9	10
$2^i \bmod 11$	1	2	4	8	5	10	9	7	3	6	1
$5^i \bmod 11$	1	5	3	4	9	1	5	3	4	9	1

Khi đó

$$\begin{aligned}\langle 2 \rangle &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\ \langle 5 \rangle &= \{1, 3, 4, 5, 9\}.\end{aligned}$$

- Liệu 2 có phải phân tử sinh?
- Liệu 5 có phải phân tử sinh?
- Nhóm \mathbb{Z}_{11}^* có phải nhóm vòng?

Xét $G = \mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Ta biết rằng 2 là một phân tử sinh.

a	1	2	3	4	5	6	7	8	9	10
$DLog_{\mathbb{Z}_{11}^*, 2}(a)$	0	1	8	2	4	9	7	3	6	5

Phân tử sinh

Nếu $G = \langle g \rangle$ là nhóm vòng thì với mọi phân tử $a \in G$ có duy nhất số mũ $i \in \{0, \dots, |G| - 1\}$ thoả mãn $g^i = a$. Ta gọi i là logarit rời rạc cơ sở g của a và ký hiệu

$$DLog_{G,g}(a)$$

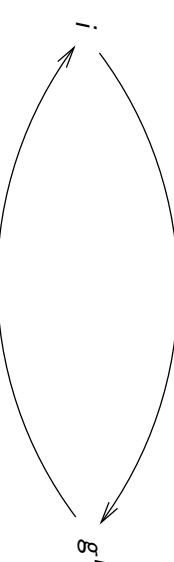
Logarit rời rạc là hàm ngược của hàm mũ.

Logarit rời rạc

G là phân tử sinh nếu và chỉ nếu $o(g) = G$.

$$\text{Exp}_G$$

- Định nghĩa**
 G là nhóm vòng nếu nó có phân tử sinh.



$$DLog_{G,g}$$



Nội dung

- Hàm cửa sập
- Hệ mật mã RSA

Hàm cửa sập (Trapdoor functions - TDF)

ĐN: hàm cửa sập $X \rightarrow Y$ là bộ ba thuật toán hiệu quả (G, F, F^{-1})

- $G()$: thuật toán *ngẫu nhiên* output cặp khóa (pk, sk)
- $F(pk, \cdot)$: thuật toán *đơn định* định nghĩa một hàm $X \rightarrow Y$
- $F^{-1}(sk, \cdot)$: hàm từ $Y \rightarrow X$ tính nghịch đảo $F(pk, \cdot)$

Cụ thể: $\forall (pk, sk)$ sinh bởi hàm G

$$\forall x \in X: F^{-1}(sk, F(pk, x)) = x$$

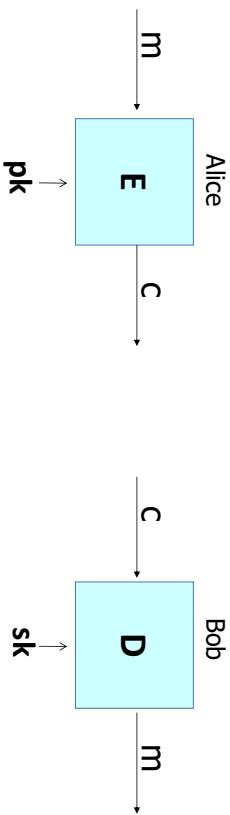


Nhập môn An toàn thông tin

Hệ mật mã RSA

Mật mã khóa công khai

Bob: sinh cặp khóa (pk, sk) và đưa pk cho Alice



Xây dựng hệ mật khóa công khai từ TDFs

- (G, F, F^{-1}) : TDF an toàn $X \rightarrow Y$
- (E_s, D_s) : hệ mật mã khóa đối xứng an toàn trên (K, M, C)
- $H: X \rightarrow K$: hàm băm

Tà xây dựng hệ mật khóa công khai (G, E, D) :

Sinh khóa G: giống như G cho TDF



Vấn đề:

- Đây là hệ mã đơn định: không an toàn !
- Tồn tại nhiều cách tấn công

$E(pk, m)$:
output $c \leftarrow F(pk, m)$

$D(sk, c)$:
output $F^{-1}(sk, c)$

Hàm của sập an toàn

(G, F, F^{-1}) là an toàn nếu $F(pk, \cdot)$ là hàm “một chiều” :
có thể tính xuôi, nhưng không thể tính nghịch đảo mà không có sk

Chál.

$(pk, sk) \leftarrow G()$

$x \xleftarrow{pk} X$

Ad.v. A

Hệ mật mã khóa công khai từ TDFs

- (G, F, F^{-1}) : TDF an toàn $X \rightarrow Y$
- (E_s, D_s) : hệ mật hóa đối xứng an toàn trên (K, M, C)
- $H: X \rightarrow K$: hàm băm

$E(pk, m)$:
 $x \xleftarrow{pk} X$, $y \leftarrow F(pk, x)$
 $k \leftarrow H(x)$, $c \leftarrow E_s(k, m)$
output (y, c)

$D(sk, (y, c))$:
 $x \leftarrow F^{-1}(sk, y)$,
 $k \leftarrow H(x)$, $m \leftarrow D_s(k, c)$
output m

ĐN: (G, F, F^{-1}) là TDF an toàn nếu với mọi thuật toán hiệu quả A:

$$Adv_{OW}[A, F] = \Pr[x = x'] < \text{"cực nhỏ"}$$



Sử dụng không đúng hàm Cửa sập (TDF)

Không mã hóa bằng cách áp dụng F để mã hóa bẩn rõ:



Nhắc lại: Số học modun hợp số

Xét $N = p \cdot q$ với p, q là các số nguyên tố

$$Z_N = \{0, 1, 2, \dots, N-1\} ; (Z_N)^* = \{\text{các phần tử khả nghịch trong } Z_N\}$$

Bổ đề: $x \in Z_N$ là khả nghịch $\Leftrightarrow \gcd(x, N) = 1$

- Số các phần tử của $(Z_N)^*$ là $\phi(N) = (p-1)(q-1) = N-p-q+1$

Định lý Euler: $\forall x \in (Z_N)^* : x^{\phi(N)} = 1$



Hoán vị cùa sập RSA

G(): chọn hai số nguyên tố $p, q \approx 1024$ bits.

Đặt $N=pq$.

chọn các số nguyên e, d thoả mãn $e \cdot d = 1 \pmod{\phi(N)}$
output $pk = (N, e)$, $sk = (N, d)$

$F(pk, x) : \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ $RSA(x) = x^e$ (in Z_N)

$F^{-1}(sk, y) = y^d ; y^d = RSA(x)^d = x^{ed} = x^{k\phi(N)+1} = (x^{\phi(N)})^k \cdot x = x$



Nội dung

- Hàm cùa sập
- Hệ mật mã RSA

Hoán vị cùa sập RSA
Ronald Rivest, Adi Shamir, và Leonard Adleman

Công bố: Scientific American, 8/1977.

Được sử dụng rộng rãi trong:

- SSL/TLS: chứng thư số và trao đổi khóa
- e-mail và hệ thống file an toàn
- ... và nhiều hệ thống khác



Hệ mật mã RSA

(chuẩn ISO)

(E_s, D_s) : hệ mật mã đối xứng an toàn.
 $H: Z_N \rightarrow K$ với K là không gian khóa của (E_s, D_s)

- $G()$: sinh tham số RSA: $pk = (N, e)$, $sk = (N, d)$
- $E(pk, m)$:
 - (1) chọn số ngẫu nhiên x thuộc Z_N
 - (2) $y \leftarrow RSA(x) = x^e$, $k \leftarrow H(x)$
 - (3) output $(y, E_s(k, m))$

• $D(sk, (y, c))$: output $D_s(H(RSA^{-1}(y)), c)$

 VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Giả sử RSA

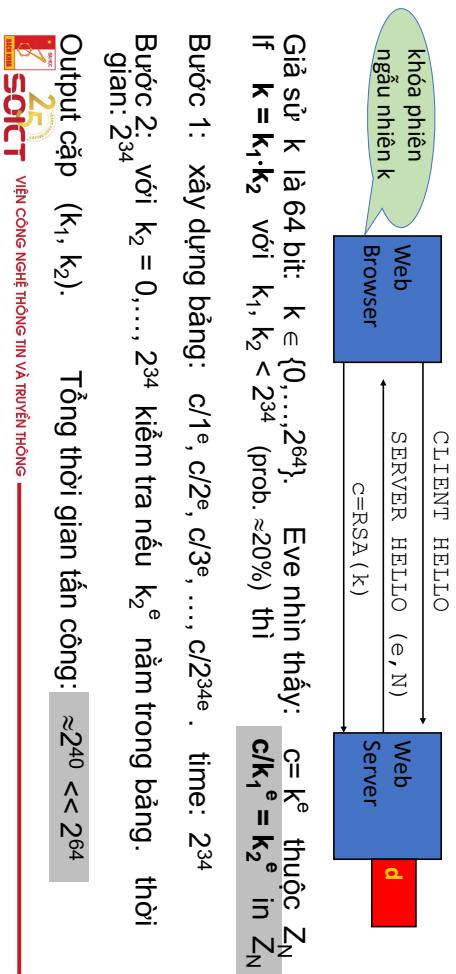
Giả sử RSA: RSA là hoán vị “một chiều”

với mọi k tến công hiệu quả A :

$$\Pr[A(N, e, y) = y^{1/e}] < \text{"cực nhỏ"}$$

ở đó $p, q \leftarrow^R$ số nguyên tố n -bit, $N \leftarrow pq$, $y \leftarrow^R Z_N^*$

Một tấn công đơn giản textbook RSA



Textbook RSA là không an toàn

Textbook RSA:

- khóa công khai: (N, e)
- khóa bí mật: (N, d)

Mã hóa: $c \leftarrow m^e$ (in Z_N)

Giải mã: $c^d \rightarrow m$

Hệ mật mã này không an toàn !

⇒ Mã hóa trực tiếp với hoán vị của sập RSA không phải là sơ đồ an toàn !

Độ dài khóa

Tính an toàn của hệ mật mã khóa công khai nên được so sánh với tính an toàn của hệ mật mã khóa đối xứng:

Khoa đối xứng	RSA	Kích thước Modulus N
80 bits	1024 bits	
128 bits	3072 bits	
256 bits (AES)	<u>15360</u> bits	

Bài tập (Tấn công RSA với modun nhỏ)

- Khoá công khai RSA của Bob có modun $N = 12191$ và số mũ $e = 37$.
- Alice gửi cho Bob bản mã $c = 587$.
- Không may, Bob đã chọn modun kích thước quá nhỏ.
- Bạn hãy giúp Oscar giải mã bằng cách phân tích thừa số nguyên tố của N và giải mã thông điệp của Alice.
- (Gợi ý. N có một thừa số nguyên tố nhỏ hơn 100.)

RSA với số mũ công khai nhỏ

Để tăng tốc việc mã hóa RSA, sử dụng số mũ e nhỏ: $c = m^e \pmod{N}$

Bài tập (Mã hóa với Textbook RSA)

Alice đưa cho Bob khoá công khai RSA của cô ấy:

modun $N = 2038667$ và số mũ $e = 103$.

- Giá trị nhỏ nhất: $e=3$ ($\gcd(e, \phi(N)) = 1$)
 - Giá trị nên dùng: $e=65537=2^{16}+1$
- Mã hóa: 17 phép nhân
- Tính bất đối xứng của RSA: mã hóa nhanh / giải mã chậm
- Hệ ElGamal (bài tiếp theo): thời gian gần nhau trong cả hai trường hợp

Nội dung

- **Bài toán Logarit rời rạc**
 - Giao thức trao đổi khoá Diffie-Hellman
 - Hệ mật mã ElGamal



Nhắc lại: Nhóm vòng

- Ký hiệu $\langle a \rangle = \{a^i \mid i \geq 0\}$ là nhóm con sinh bởi a .
- Nếu $\langle a \rangle = G$ thì a là một phần tử sinh của G .

- **Khẳng định:** $|\langle a \rangle| = \text{ord}(a)$.

- Định nghĩa: G là nhóm vòng nếu có g thoả mãn $\langle g \rangle = G$



Nhắc lại: Cấp của một phần tử trong nhóm

- Cấp của phần tử a , ký hiệu $\text{ord}(a)$, là số $u > 0$ nhỏ nhất thoả mãn

- **Định lý Lagrange:** Trong nhóm hữu hạn G với lực lượng t , ta có

- **Hệ quả:** Trong nhóm hữu hạn G với lực lượng t , ta có

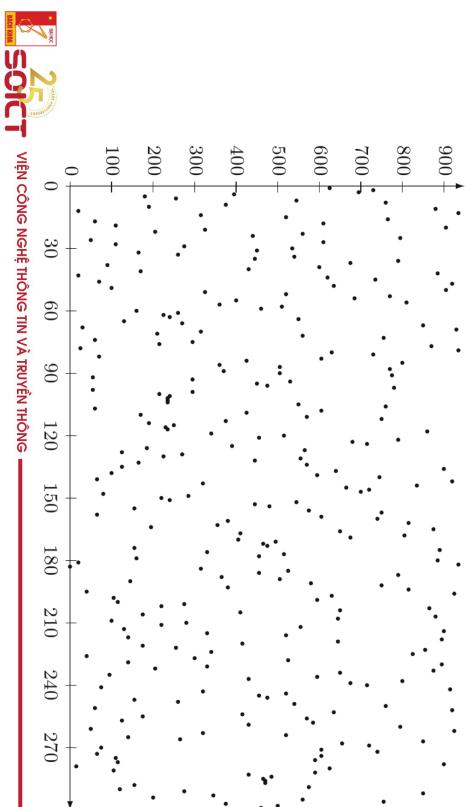
- Ký hiệu: $\langle a \rangle = \{a^i \mid i \geq 0\}$ là nhóm con sinh bởi a .

Nhập môn An toàn thông tin



Hệ mật mã dựa trên
Bài toán logarit rời rạc và Diffie-Hellman

Tính ngẫu nhiên của lũy thừa $627^x \pmod{941}$



Bài tập

Hãy tính các logarit rời rạc sau.

1. $\text{Dlog}_2(13)$ trong modun nguyên tố 23
2. $\text{Dlog}_{10}(22)$ trong modun nguyên tố $p = 47$.
3. $\text{Dlog}_{627}(608)$ trong modun nguyên tố $p = 941$.



Bài toán Logarit rời rạc

- **Khẳng định:** Nếu G là nhóm vòng cấp t và g là phần tử sinh, thì ánh xạ
$$x \leftrightarrow g^x$$
là 1-to-1 giữa $\{0, 1, \dots, t - 1\}$ và G .
- Hàm mũ
$$x \rightarrow g^x$$
- **Hàm logarit rời rạc** $g^x \rightarrow x$

Nội dung

- Bài toán Logarit rời rạc
- **Giao thíc trao đổi khoá Diffie-Hellman**
- Hệ mật mã ElGamal



Giao thíc Diffie-Hellman

Chọn một số nguyên tố lớn p (v.d. 600 chữ số)
Chọn một số nguyên g thuộc $\{1, \dots, p\}$

Alice
Chọn ngẫu nhiên a thuộc $\{1, \dots, p-1\}$
"Alice", $A \leftarrow g^a \pmod{p}$

Bob

$$B^a \pmod{p} = (g^b)^a = k_{AB} = g^{ab} \pmod{p} = (g^a)^b = A^b \pmod{p}$$



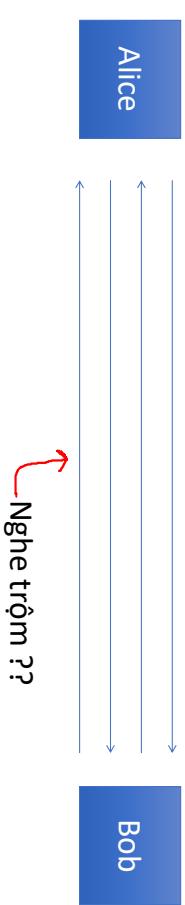
Tính Logarit rời rạc

- Xét số nguyên tố $p = 56509$, và ta có thể kiểm tra $g = 2$ là một phần tử sinh của Z_p .
- Làm thế nào để tính $\text{Dlog}_2(38679)$?
- Một phương pháp là tính $2^0, 2^1, 2^2, 2^3, \dots \pmod{56509}$ cho đến khi được lũy thừa bằng 38679.
- Bạn có thể kiểm tra rằng $2^{11235} \equiv 38679 \pmod{56509}$.



Trao đổi khoá không cần bên thứ ba

Mục đích: Alice và Bob muốn chia sẻ khoá bí mật, mà kẻ nghe trộm không biết



Tính an toàn

Kẻ nghe trộm nhìn thấy: $p, g, A=g^a \pmod{p},$ và $B=g^b \pmod{p}$

Liệu có thể tính $g^{ab} \pmod{p}$??

Tổng quát: $DH_g(g^a, g^b) = g^{ab} \pmod{p}$

Hàm DH theo modun p liệu có khó tính?



Hàm DH theo modun p

Giả sử p là số nguyên tố n dài bits long.

Thuật toán tốt nhất (GNFS): có thời gian ch $\exp(\tilde{O}(\sqrt[3]{n}))$

khoá bí mật	kích thước modun	Kích thước Elliptic Curve
80 bits	1024 bits	160 bits
128 bits	3072 bits	256 bits
256 bits (AES)	15360 bits	512 bits

Hệ quả: chuyển từ $(\text{mod } p)$ sang đường cong Elliptic



Bài tập

- Alice và Bob dùng số nguyên tố $p = 1373$ và cơ sở $g = 2$ để trao đổi khóa.
- Alice gửi Bob giá trị $A = 974.$
- Bob chọn số bí mật $b = 871.$
- Bob nên gửi cho Alice giá trị gì, và khóa bí mật họ chia sẻ là gì?
- Bạn có thể đoán được số bí mật a của Alice không?

Bài tập

Hãy tính hai giá trị sau trong $\mathbb{Z}_{13}^*.$

- $DH_7(10,5)$
- $DH_2(12,9)$

biết rằng

$$\begin{aligned}\langle 2 \rangle &= \{1, 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7\} \\ \langle 7 \rangle &= \{1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2\}\end{aligned}$$

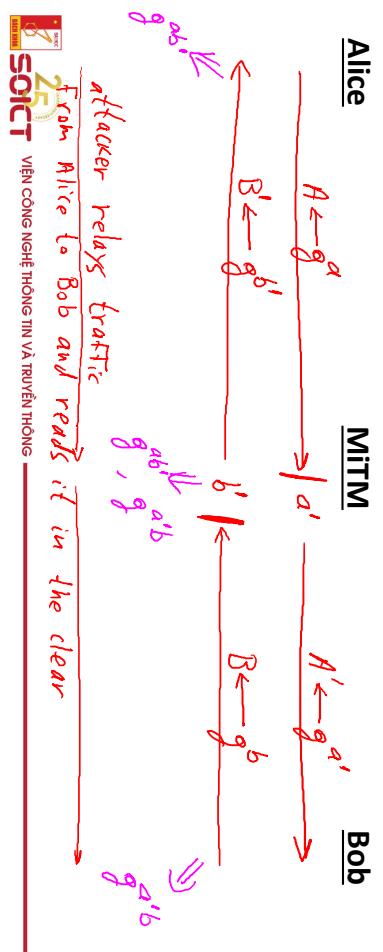
$$DH_g(g^a, g^b) = g^{ab} \pmod{p}$$



Không an toàn chống lại

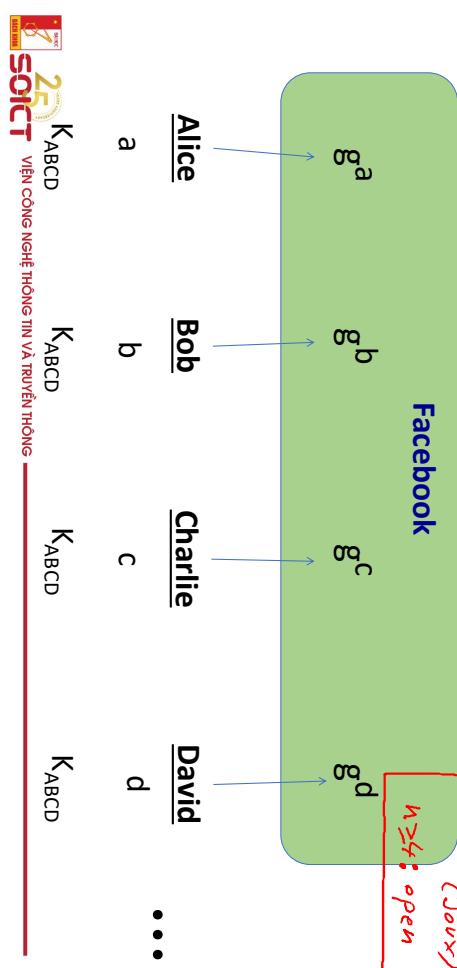
man-in-the-middle

Giao thức này không an toàn chống lại kẻ tấn công chủ động



Một bài toán mở

$n=2$: OH
 $h=3$: Known
 (Jaw)
 $h \geq 4$: open



Một cách nhìn khác về DH

www.google.com

The identity of this website has been verified by Thawte SGC C.A.

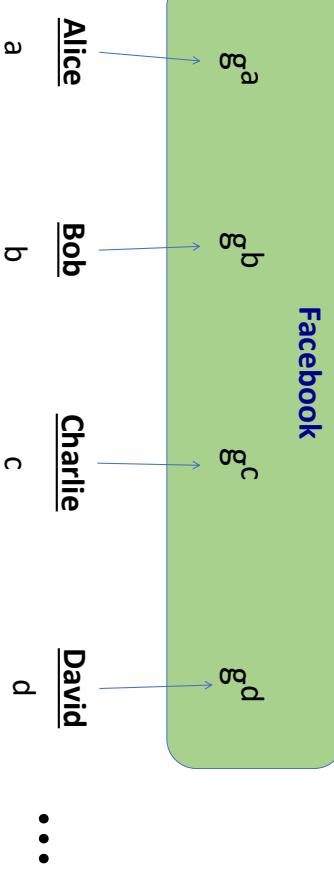
[Certificate Information](#)

Your connection to www.google.com is encrypted with 128-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using RC4 128, with SHA1 for message authentication and ECDHE_RSA as the key exchange mechanism.

Elliptic curve
Diffie-Hellman



Nhắc lại: Giao thức Diffie-Hellman (1977)

Xét nhóm vòng G (e.g. $G = (\mathbb{Z}_p)^*$) với cấp n

Lấy một phần tử sinh g thuộc G (i.e. $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)

Alice

Chọn ngẫu nhiên a in $\{1, \dots, n\}$

$$A = g^a$$

$$B = g^b$$

Bob

Chọn ngẫu nhiên b trong $\{1, \dots, n\}$

$$\begin{aligned} B^a &= (g^b)^a = \\ &\quad \text{[Alice]} \end{aligned}$$

$$\begin{aligned} k_{AB} &= g^{ab} = \\ &= (g^a)^b = A^b \end{aligned}$$



Một câu hỏi mở

- Nếu ta có thể giải bài toán Logarit rời rạc, vậy ta có thể giải bài toán Diffie-Hellman. Tại sao?
- Nhưng nếu ta có thể giải được bài toán Diffie-Hellman, vậy liệu ta có thể giải được bài toán logarit rời rạc không?

Nội dung

- Bài toán Logarit rời rạc
- Giao thức trao đổi khoá Diffie-Hellman
- Hệ mật mã ElGamal**



ElGamal: converting to pub-key enc. (1984)

Xét nhóm vòng G (e.g $G = (Z_p)^*$) với cấp n

Lấy một phần tử sinh g thuộc G (i.e. $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)

Alice

Chọn ngẫu nhiên a thuộc $\{1, \dots, n\}$

Coi a như khóa công khai

$$A = g^a$$

Để giải mã:
tính $g^{ab} = B^a$,
Dẫn ra k , và giải mã

$$ct = \left[\begin{array}{l} B = g^b \\ M = g^m \end{array} \right]$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Bob

Chọn ngẫu nhiên b in $\{1, \dots, n\}$

Coi b như khóa công khai

$$B = g^b$$

tính $g^{ab} = A^b$,
Dẫn xuất khoá đối xứng k ,
kết hợp với m để giải mã

$$M = g^m$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ElGamal: converting to pub-key enc. (1984)

Xét nhóm vòng G (e.g $G = (Z_p)^*$) với cấp n

Lấy một phần tử sinh g thuộc G (i.e. $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)

Alice

Chọn ngẫu nhiên a thuộc $\{1, \dots, n\}$

Coi A như khoá công khai

$$A = g^a$$

tính $g^{ab} = A^b$,
Dẫn xuất khoá đối xứng k ,
 $ct = [B = g^b, M = g^m]$

Bob

Tra xâу dung hệ mật khoá công khai (Gen, E, D):

- Sinh khoá Gen:

- Chọn ngẫu nhiên phần tử sinh g trong G và một số ngẫu nhiên a thuộc Z_n
- output $sk = a$, $pk = (g, h=g^a)$

Hệ mật ElGamal

- G: nhóm vòng cấp n

- (E_s, D_s) : mã đổi xứng an toàn trên (K, M, C)

- $H: G^2 \rightarrow K$ hàm băm

E(pk=(g,h), m):

$$\begin{aligned} b &\leftarrow Z_n, u \leftarrow g^b, v \leftarrow h^b \\ k &\leftarrow H(u, v), c \leftarrow E_s(k, m) \end{aligned}$$

output (u, c)

D(sk=a, (u,c)):

$$\begin{aligned} v &\leftarrow u^a \\ k &\leftarrow H(u, v), m \leftarrow D_s(k, c) \\ \text{output } m \end{aligned}$$



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Hiệu năng ElGamal

E(pk=(g,h), m):

$$b \leftarrow Z_n, u \leftarrow g^b, v \leftarrow h^b$$

D(sk=a, (u,c)):

$$v \leftarrow u^a$$

Mã hoá: 2 phép lấy mũ. (cơ sở cố định)

- Có thể tính trước $[g^{(2^i)}, h^{(2^i)} \text{ for } i=1, \dots, \log_2 n]$
- Tốc độ nhanh gấp 3x (hoặc hơn)

Decryption: 1 phép lấy mũ. (cơ sở thay đổi)

Nội dung

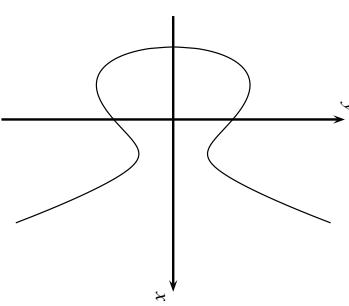
Đường cong Elliptic

1. Đường cong Elliptic (Elliptic Curve, EC)

2. Bài toán Logarit rời rạc trên EC

3. Giao thức trao đổi khóa Diffie-Hellman trên EC

Đường cong Elliptic trên K là tập mọi cặp $(x, y) \in K$ thoả mãn phương trình
 $y^2 = x^3 + a \cdot x + b$
cùng với một điểm vô cực O ,
trong đó
 $a, b \in K$
và thoả mãn $4 \cdot a^3 + 27 \cdot b^2 \neq 0$.



Vấn đề: Tìm hệ mật với tham số ngắn hơn

Hệ mật mã dựa trên đường cong Elliptic

Algorithm Family	Cryptosystems	80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

Phép toán nhóm trên EC

- Ký hiệu phép toán nhóm bởi ký hiệu cộng “+”.
- Cho hai điểm $P = (x_1, y_1)$ và $Q = (x_2, y_2)$
- Tính tọa độ của điểm thứ ba R thoả mãn:

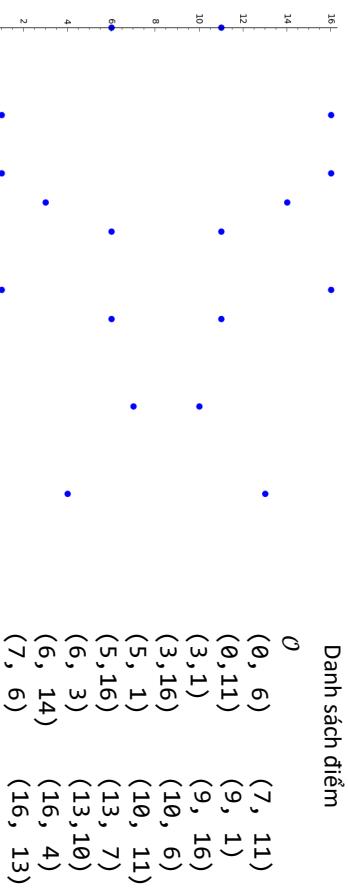
$$P + Q = R$$

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

- Phép cộng điểm $P + Q$:** Trường hợp $R = P + Q$ và $P \neq Q$
- Nhân đôi điểm $P + P$:** Trường hợp $P + Q$ nhưng $P = Q$.

Đường cong $y^2 = x^3 + 2x + 2$ trên \mathbb{Z}_{17}

Danh sách điểm



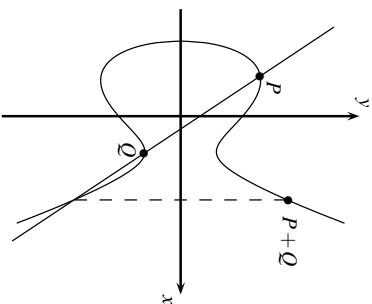
Phép toán cộng và nhân đôi các điểm

$$\begin{aligned} x_3 &= s^2 - x_1 - x_2 \mod p \\ y_3 &= s(x_1 - x_3) - y_1 \mod p \end{aligned}$$

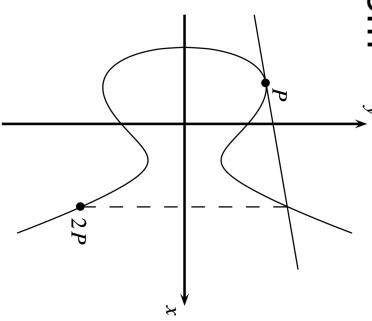
với

$$s = \begin{cases} (y_2 - y_1)/(x_2 - x_1) \mod p & \text{if } P \neq Q \\ (3x_1^2 + a)/(2y_1) \mod p & \text{if } P = Q \end{cases}$$

Phép toán nhóm



Cộng điểm $P + Q$



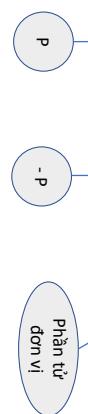
Nhân đôi $P + P = 2P$

Tính toán với SageMath

```
sage: E = EllipticCurve(GF(17), [2,2])
sage: E
Elliptic Curve defined by  $y^2 = x^3 + 2*x + 2$  over
Finite Field of size 17
sage: P = E(5,1)
sage: Q = P + P
sage: print Q
(6 : 3 : 1)
sage: E.is_on_curve(6,3)
True
```



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



```
sage: O = P + -P
sage: O
(0 : 1 : 0)
sage: 0 + 0 == 0
True
sage: P + O
(5 : 1 : 1)
sage: P + O == P
True
sage: O + P == P
True
```



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Ví dụ

Xét đường cong

$$E: \quad y^2 = x^3 + 2x + 2 \text{ mod } 17$$

Ta muốn nhân đôi điểm $P = (5,1)$.

$$\begin{aligned} 2P &= P + P = (5,1) + (5,1) = (x_3, y_3). \\ s &= (3x_1^2 + a)/(2y_1) = (2 \cdot 1)^{-1}(3 \cdot 5^2 + 2) = 2^{-1} \cdot 9 = 13 \text{ mod } 17 \\ x_3 &= s^2 - x_1 - x_2 = 13^2 - 5 - 5 = 6 \text{ mod } 17. \\ y_3 &= s(x_1 - x_3) - y_1 = 13(5 - 6) - 1 = -14 = 3 \text{ mod } 17 \\ 2P &= (5,1) + (5,1) = (6,3) \end{aligned}$$

Luật cộng đầy đủ cho EC

1. $\mathcal{O} + \mathcal{O} = \mathcal{O}$.
2. $\mathcal{O} + (x_2, y_2) = (x_2, y_2)$.
3. $(x_1, y_1) + \mathcal{O} = (x_1, y_1)$.
4. $(x_1, y_1) + (x_1, -y_1) = \mathcal{O}$.
5. cho $y_1 \neq 0$, $(x_1, y_1) + (x_1, y_1) = (s^2 - 2x_1, s(x_1 - x_3) - y_1)$ với $s = (3x_1^2 + a)/2y_1$.
6. cho $x_1 \neq x_2$, $(x_1, y_1) + (x_2, y_2) = (s^2 - x_1 - x_2, s(x_1 - x_3) - y_1)$ với $s = (y_2 - y_1)/(x_2 - x_1)$.

Kiểm tra các tính chất với SageMath

$$1. \mathcal{O} + \mathcal{O} = \mathcal{O}.$$

$$2. \mathcal{O} + (x_2, y_2) = (x_2, y_2).$$

$$3. (x_1, y_1) + \mathcal{O} = (x_1, y_1).$$

$$4. (x_1, y_1) + (x_1, -y_1) = \mathcal{O}.$$

$$5. \text{cho } y_1 \neq 0, (x_1, y_1) + (x_1, y_1) = (s^2 - 2x_1, s(x_1 - x_3) - y_1)$$

$$6. \text{cho } x_1 \neq x_2, (x_1, y_1) + (x_2, y_2) = (s^2 - x_1 - x_2, s(x_1 - x_3) - y_1)$$

Lợi ích của hệ toạ độ chiếu

- Tính toán phép “+” hiệu quả hơn do tránh được phép nghịch đảo trên trường hữu hạn
- Phép toán cơ bản kP trở nên dễ dàng

$$(x', y') = 2(x, y)$$

$$(X' : Y' : Z') = 2(X : Y : Z)$$

$$\begin{aligned} s &= \frac{3x^2+a}{2y} \\ x' &= s^2 - 2x \\ y' &= s(x - x') - y \end{aligned}$$



Viện Công nghệ thông tin và Truyền thông

Hệ toạ độ chiếu

- Điểm chiếu $(X : Y : Z)$, $Z \neq 0$ tương ứng với điểm $(X/Z, Y/Z)$.

- Phương trình chiếu của EC là $Y^2Z = X^3 + aXZ^2 + bZ^3$.

- Điểm tại vô cực O tương ứng với $(0:1:0)$, và phần tử nghịch đảo của $(X : Y : Z)$ là $(X : -Y : Z)$.

Tính toán với SageMath

```
sage: E = EllipticCurve(GF(17), [2, 2])
sage: E
Elliptic Curve defined by
y^2 = x^3 + 2*x + 2
over Finite Field of size 17
sage: for P in E:
....:     print P
....:
(0 : 1 : 0)
(0 : 6 : 1)
(0 : 11 : 1)
(3 : 1 : 1)
(3 : 16 : 1)
(10 : 11 : 1)
(13 : 7 : 1)
(13 : 10 : 1)
(16 : 4 : 1)
(16 : 13 : 1)
```



Viện Công nghệ thông tin và Truyền thông

Ví dụ trên SageMath

```
def point_doubling(x, y, z, a):
    x_ = 2*y*z*((3*x^2 + a*z^2)^2 - 8*y^2*x*z)
    y_ = (3*x^2 + a*z^2)*(12*y^2*x^2 - (3*x^2 + a*z^2)^2) - 8*y^4*z^2
    z_ = 8*y^3*z^3
    return (x_, y_, z_)
```

```
F = GF(17)
x, y, z, a = F(13), F(7), F(1), F(2)
print(point_doubling(x,y,z,a))
E = EllipticCurve(GF(17), [2, 2])
P = E(13, 7)
print(P+P)
```

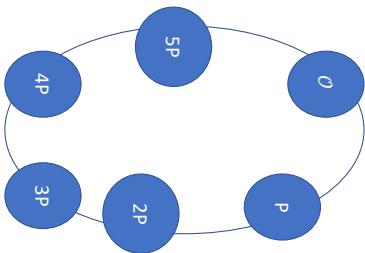
Nhóm con vòng (cyclic)

Định lý.

Các điểm trên đường cong Elliptic cùng với điểm \mathcal{O} có nhóm con vòng.

Dưới một số điều kiện các điểm trên EC lập thành một nhóm vòng.

$P = (5,1)$	$6P = (16,13)$	$11P = (13,10)$	$16P = (10,11)$
$2P = (6,3)$	$7P = (0,6)$	$12P = (0,11)$	$17P = (6,14)$
$3P = (10,6)$	$8P = (13,7)$	$13P = (16,4)$	$18P = (5,16)$
$4P = (3,1)$	$9P = (7,6)$	$14P = (9,1)$	$19P = \mathcal{O}$
$5P = (9,16)$	$10P = (7,11)$	$15P = (3,16)$	



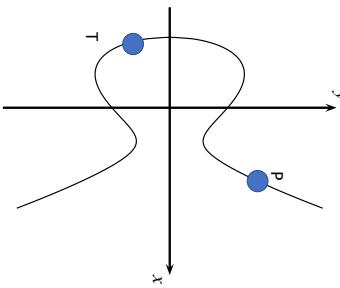
$$E: y^2 = x^3 + 2x + 2 \text{ mod } 17$$

Bài toán logarit rời rạc trên EC (ECDLP)

ĐN. Cho đường cong elliptic E . Ta xét một điểm P và điểm khác T .

Bài toán DL nhằm tìm số nguyên d thoả mãn

$$\underbrace{P + P + \cdots + P}_{d \text{ times}} = dP = T.$$



Tính $\log_P(Q)$ với $P = (5,1)$ và $Q = (10,11)$

$$P = (5,1)$$

$$2P = (6,3)$$

$$3P = (10,6)$$

$$4P = (3,1)$$

$$5P = (9,16)$$

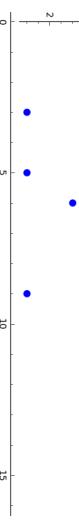
$$6P = (16,13)$$

$$7P = (0,6)$$

$$8P = (13,7)$$

$$9P = (7,6)$$

$$10P = (7,11)$$



$$E: y^2 = x^3 + 2x + 2 \text{ mod } 17$$

Số điểm của EC

Hass's Theorem:

Cho đường cong E modun p , số điểm trên đường cong ký hiệu bởi $\#E$ và bị chặn bởi:

$$p + 1 - \sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}$$

- $\#E \approx p$

- Nếu ta cần một đường cong với số điểm 2^{160} ta phải sử dụng số

Nội dung

- Đường cong Elliptic (Elliptic Curve, EC)
- Bài toán Logarit rác r牠 trên EC

3. Giao thức trao đổi kho\u00e1 Diffie-Hellman tr\u00e0n EC

B\u00e1i t\u00e1p

X\u00e9t đường cong

$$E: y^2 = x^3 + 2x + 2 \bmod 17$$

Ta đ\u00e1 tính c\u00e2c "m\u00fa" c\u00f3a P .

$P = (5, 1)$	$6P = (16, 13)$	$11P = (13, 10)$	$16P = (10, 11)$
$2P = (6, 3)$	$7P = (0, 6)$	$12P = (0, 11)$	$17P = (6, 14)$
$3P = (10, 6)$	$8P = (13, 7)$	$13P = (16, 4)$	$18P = (5, 16)$
$4P = (3, 1)$	$9P = (7, 6)$	$14P = (9, 1)$	$19P = 0$
$5P = (9, 16)$	$10P = (7, 11)$	$15P = (3, 16)$	

Với $P = (5, 1)$ và $T = (16, 4)$, h\u00e0y tìm s\u00f9 nguy\u00e4n d sao cho $P = T$.

T\u00ednh an toàn

M\u00f3i giao th\u00fc Ec d\u00e1ng tr\u00e0n t\u00f9nh kh\u00f4 gi\u00e1i c\u00f3a bài toán ECDLP

- Nếu EC đ\u00e1ng c\u00e2n c\u00e2n $\approx \sqrt{p}$ bước.
- ECDLP c\u00e2n $\approx \sqrt{p}$ bước.
- VD: $p \approx 2^{160}$
t\u00e1n c\u00f3ng c\u00e2n $\approx \sqrt{2^{160}} = 2^{80}$ bước

Pha 2: Trao đổi khoá

Alice
Chọn $a \in \{2, \dots, \#E - 1\}$

Bob
Chọn $b \in \{1, \dots, \#E - 1\}$

$$\begin{array}{l} A = aP \\ \downarrow \\ B = bP \end{array}$$


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

$$aB = a(bP) = k_{AB} = abP = bA = b(aP)$$

- Tính an toàn của giao thức trao đổi khoá Diffie Hellman
- kết tinh công nhìn thấy giá trị aP và bP
 - và phải tính giá trị $K_{ab} = abP$
 - Khó khăn của tính toán được dẫn từ hai bài toán được tin là khó

Bài toán quyết định (DDH):
• Cho (P, aP, bP, cP) , hãy kiểm tra liệu $ab == c$.

Bài toán tính toán Diffie Hellman (CDH):
• Cho (P, aP, bP) , hãy tính abP .


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Pha 1: Tham số miền cho ECDH

1. Chọn một số nguyên tố p và đường cong

$$E: y^2 = x^3 + ax + b \text{ mod } p$$

2. Chọn điểm $P = (x_p, y_p)$ trên đường cong

Phép nhân với hằng số

```
def scalarMult(n, P):
```

```
    if n == 0: return 0
```

```
    if n == 1: return P
```

```
    R = scalarMult(n//2, P)
```

```
    R=R+R
```

```
    if n % 2: R = R + P
```

```
return R
```

Trường hợp tối thiểu:
 $31P = 2(2(2(2P + P) + P) + P) + P$.

4 phép nhân đôi; 4 phép cộng.

Trường hợp trung bình:
 $35P = 2(2(2(2(2P))) + P) + P$.

5 phép nhân đôi; 2 phép cộng.

Thời gian CPU bị chẵn bởi

```
log2(n)
```

lần nhân đôi điểm


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Giả sử tính toán Diffie Hellman

Giả sử tính toán DH đúng trong E nếu: $P, aP, bP \not\Rightarrow abP$

với mọi thuật toán hiệu quả A:

$$\Pr[A(P, aP, bP) = abP] < \text{rất nhỏ}$$

với $P \leftarrow \{\text{phần tử sinh của } E\}, a, b \leftarrow Z_n$



Viện Công nghệ thông tin và Truyền thông

DLP \rightarrow DH

Quyết định Diffie Hellman (DDH):

- Cho (P, aP, bP, cP) , kiểm tra liệu $ab == c$



Tính toán Diffie Hellman (CDH):

- Cho (P, aP, bP) , hãy tính abP .

Nhiều người tin là "đúng"

Bài toán logarit rời rạc (DLP)

- Cho (P, aP) , hãy tính a

P256 trên SageMath

```
sage: p = 2^256 - 2^224 + 2^192 + 2^96 - 1
sage: is_prime(p)
True
sage: b = 0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e
sage: b
9559645253501865577261459496814587248748736692591040757213546036286
sage: P256 = EllipticCurve(GF(p), [-3, b])
sage: P256.order()
15792089210356248762697446949407573530139109078099854062854297063875752950436
sage: P = P256.random_element()
sage: P
(44003593087052944911338129277746441384567907740211507216344250174958576726 :
5520086247573023097393606373907129377872093705027768026391600457848619693219 : 1)
```



Viện Công nghệ thông tin và Truyền thông

Đường cong P256

Đường cong có dạng

$$y^2 = x^3 - 3x + b \pmod{p}$$

- Số nguyên tố $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$
- và b ở hexa là:

$$b := 5ac635d8 aa3a93e7 b3ebbd55 769886bc 651d06b0 \\ cc53b0f6 3bce3c3e$$

- Số nguyên tố gần bằng 2^{256} , số điểm gần bằng 2^{256} .
- Tính logarit rời rạc mất khoảng 2^{128} bước
- Tham số b trong P256 được chọn thế nào?
- $P256$ được dùng rộng rãi trong thực tế



Viện Công nghệ thông tin và Truyền thông



Viện Công nghệ thông tin và Truyền thông

Bài tập

- Xét đường cong
E: $y^2 = x^3 + 2x + 2 \text{ mod } 17$
- Và hai điểm P = (5,1) và Q = (10,6) trên E.
- Hãy tìm số nguyên d mà $1 \leq d \leq \#E$, thoả mãn: dQ = P?

1. d = 1

2. d = 13

3. d = 17

4. Không có số d như vậy.

Bài tập

- Xét đường cong

E: $y^2 = x^3 + 2x + 2 \text{ mod } 17$

- Và hai điểm P = (5,1) và Q = (10,6) trên E.

- Điểm R = P + Q là gì?

1. R = (15, 7)

2. R = (3, 1)

3. R = O

Một phần bức tranh mật mã

Mật mã khóa công khai

Thuật toán:

Tính bí mật	Tính toàn vẹn
Mã khóa đối xứng	Mã xác thực thông điệp (MAC)
Mã khóa công khai	Chữ ký điện tử

Ví dụ

- Giao thức trao đổi khóa Diffie-Hellman (DH)



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

2/7



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Thuật toán:

$$\begin{aligned} K &\leftarrow \underline{\text{Gen}}(1^\lambda) && \text{sinh khóa độ dài } \lambda \\ \underline{\underline{C}} &\leftarrow \underline{\underline{\text{Enc}}}(K, M) && \text{mã hóa thông điệp } M \text{ với khóa } K, \text{kết quả} \\ \underline{\underline{M}} &= \underline{\underline{\text{Dec}}}(\underline{\underline{K}}, \underline{\underline{C}}) && \text{là bản mã } C \\ &&& \text{giải mã } C \text{ dùng khóa } K \text{ để lấy được } M. \end{aligned}$$

Mật mã khóa đối xứng

Mật mã ứng dụng

Các thành phần mật mã cơ bản

- Sử dụng trong thực tế.
- Nếu chỉ cần tính bí mật: AES-128 với CBC mode hoặc CTR mode.
 - Nếu cần cả tính bí mật và xác thực: EAX, CCM, hoặc GCM

1/7



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

3/7

Mật mã khóa công khai

Mã xác thực thông điệp

Thuật toán:

$(SK, PK) \leftarrow Gen(1^\lambda)$ sinh cặp khóa (bí mật, công khai) độ dài λ
 $C \leftarrow Enc(PK, M)$ mã hóa thông điệp M với khóa công khai
 PK , kết quả là bản mã C
 $M = Dec(SK, C)$ giải mã C dùng khóa bí mật SK để được M .

Ví dụ.

- Giao thức trao đổi khóa Diffie-Hellman (DH)
- Hệ mật mã RSA
- Hệ mật mã dựa trên đường cong Elliptic (ECC)



Viện Công nghệ Thông tin và Truyền thông

4/7



Kiểm tra tag:

$$V(k, m, tag) \stackrel{?}{=} yes$$

Thuật toán:

$\frac{k \leftarrow Gen(1^\lambda)}{t \leftarrow S(k, m)}$ sinh khóa độ dài k
 $\underline{\underline{V(k, m, t)}}$ tạo chữ ký thông điệp m dùng khóa k
“yes” hoặc “no” cho biết chữ ký t có phải là
chữ ký hợp lệ của m hay không.

Ví dụ. HMAC.



Viện Công nghệ Thông tin và Truyền thông

6/7

Mật mã khóa công khai

Thuật toán:

$(SK, PK) \leftarrow Gen(1^\lambda)$ sinh cặp khóa (bí mật, công khai) độ dài λ
 $C \leftarrow Enc(PK, M)$ mã hóa thông điệp M với khóa công khai
 PK , kết quả là bản mã C
 $M = Dec(SK, C)$ giải mã C dùng khóa bí mật SK để được M .

Ví dụ.

- Giao thức trao đổi khóa Diffie-Hellman (DH)
- Hệ mật mã RSA



Kích thước khóa (theo bit)

Khuyến nghị của NIST

AES	DH & RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Thuật toán:



Viện Công nghệ Thông tin và Truyền thông

4/7

5/7

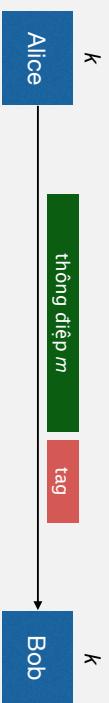
Chữ ký điện tử

Thuật toán:

$(sk, pk) \leftarrow Gen(1^\lambda)$ sinh khóa bí mật và công khai độ dài λ
 $\underline{t} \leftarrow S(\underline{sk}, m)$ tạo chữ ký thông điệp m dùng khóa bí mật
 \underline{sk}
 $\underline{V}(pk, m, \underline{t})$ “yes” hoặc “no” cho biết chữ ký t có phải là
chữ ký hợp lệ của m hay không.

Ví dụ. RSA, DSA, ECDSA

Toàn vẹn thông điệp: MAC (Message Authentication Code)



MÃ XÁC THỰC THÔNG ĐIỆP

- ▶ Toàn vẹn thông điệp
- ▶ MAC dựa trên PRF
- ▶ CBC-MAC và NMAC
- ▶ MAC padding

[https://iclass.coursera.org/
crypto-preview/class/index](https://iclass.coursera.org/crypto-preview/class/index)

MÃ XÁC THỰC THÔNG ĐIỆP

- ▶ Toàn vẹn thông điệp
- ▶ MAC dựa trên PRF
- ▶ CBC-MAC và NMAC
- ▶ MAC padding

[https://iclass.coursera.org/
crypto-preview/class/index](https://iclass.coursera.org/crypto-preview/class/index)



Toàn vẹn thông điệp

Mục đích

- Toàn vẹn, không cần bí mật

Ví dụ

- Bảo vệ các file công khai trên đĩa
- Bảo vệ các banner quảng cáo trên trang web

MAC an toàn

Khả năng của kẻ tấn công

- kẻ tấn công có thể lấy được các tag $t_i \leftarrow S(k, m_i)$ của m_1, m_2, \dots, m_q

Mục đích của kẻ tấn công: Giả mạo thông điệp

- đưa ra được một cặp thông điệp/tag (m, t) hợp lệ mới

$$(m, t) \notin \{(m_1, t_1), \dots, (m_q, t_q)\}$$

Có nghĩa rằng:

- kẻ tấn công không thể tạo ra một tag hợp lệ cho một thông điệp mới
- đưa ra (m, t) kẻ tấn công thậm chí không tạo được (m, t') với $t' \neq t$

Câu hỏi

Xét $I = (S, V)$ là một MAC.

Giả sử một kẻ tấn công có thể tìm được $m_0 \neq m_1$ sao cho

$$S(k, m_0) = S(k, m_1)$$

với $1/2$ số khóa k trong K .

Vậy MAC này có an toàn không?

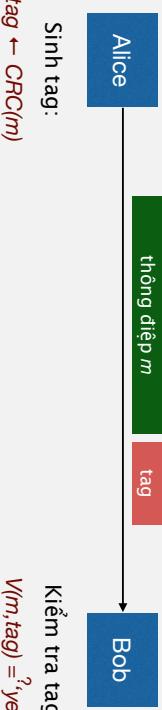
- Có, kẻ tấn công không thể sinh tag đúng cho m_0 hoặc m_1

- Không, MAC này có thể bị phá dùng tấn công chọn thông điệp

- Nó phụ thuộc vào thiết kế của MAC

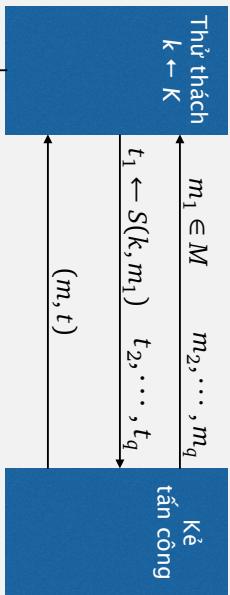
MAC an toàn

Toàn vẹn thông điệp cần một khóa bí mật



MAC an toàn

Cho MAC $I = (S, V)$ và một kẻ tấn công A. Ta định nghĩa một thử nghiệm MAC như sau:



- Kẻ tấn công có thể dễ dàng thay đổi thông điệp và tính lại CRC (Cyclic redundancy check).
- CRC được thiết kế để phát hiện lỗi xảy ra ngẫu nhiên chứ không chống được lỗi có chủ đích.

Định nghĩa. MAC $I = (S, V)$ là MAC an toàn nếu với mọi thuật toán “hiệu quả” A:

$$\text{Adv}_{\text{MAC}}[A, I] = \Pr[\text{Thử thách output} = 1]$$

là “không đáng kể”

Ví dụ: Bảo vệ hệ thống files

Giả sử tại thời điểm cài đặt hệ thống tính toán:



Sau đó hệ thống bị nhiễm virus, và các file bị sửa đổi.

Người dùng khởi động lại vào OS sạch và nhập mật khẩu

- Khi đó: MAC an toàn sẽ cho phép phát hiện các file bị sửa đổi

Hàm giả ngẫu nhiên

Hàm giả ngẫu nhiên (PRF) định nghĩa trên (K, X, Y) là hàm:

$$F: K \times X \rightarrow Y$$

thỏa mãn có thuật toán “hiệu quả” để tính $F(k, x)$

10

Câu hỏi

Xét MAC $I = (S, V)$ và giả sử $S(k, m)$ luôn output dãy 5 bit.

MAC này có an toàn không?

- Không, kẻ tấn công có thể gợi ý tag cho các thông điệp
- Nó phụ thuộc vào thiết kế chi tiết của MAC
- Có, kẻ tấn công không thể sinh tag hợp lệ cho bất kỳ thông điệp nào.

MÃ XÁC THỰC THÔNG ĐIỆP

Toàn vẹn thông điệp

MAC dựa trên PRF

CBC-MAC và NMAC

MAC padding

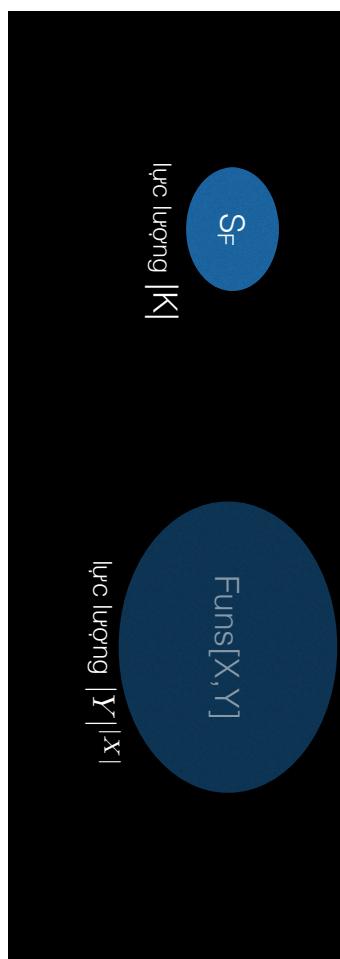
PMAC và Carter-Wegman MAC

Nhắc lại: MAC an toàn

PRF an toàn: trực giác

- MAC:**
- Thuật toán ký: $t \leftarrow S(k, m)$
 - Thuật toán kiểm tra: $V(k, m, t) = \text{'yes'}$ hoặc 'no'

Một PRF F là an toàn nếu ta không thể phân biệt được một hàm được lấy ngẫu nhiên từ $\text{Funs}[X, Y]$ hay lấy ngẫu nhiên từ S_F



PRF và hàm ngẫu nhiên

Ta ký hiệu

$$\text{Funs}[X, Y] := \{ \text{mọi hàm từ } X \text{ lên } Y \}$$

Câu hỏi: Lực lượng của $\text{Funs}[X, Y]$?

Cho trước một PRF $F: K \times X \rightarrow Y$ ta đặt

$$S_F := \{F(k,.) \text{ thỏa mãn } k \in K\}$$

Câu hỏi: Lực lượng của S_F ?

- ## PRF an toàn trong thực tế
- 3DES: $\{0,1\}^{168} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$
 - AES128: $\{0,1\}^{128} \times \{0,1\}^{128} \rightarrow \{0,1\}^{128}$

Câu hỏi

Giả sử $F : K \times X \rightarrow Y$ là một PRF an toàn với $Y = \{0,1\}^{10}$.

$MAC_{|F}$ có phải là hệ MAC an toàn ?

1. Có, MAC là an toàn vì PRF là an toàn.
2. Không, độ dài tag quá ngắn: người ta có thể gợi ý ngẫu nhiên tag cho thông điệp bất kỳ.
3. Phụ thuộc vào thiết kế chi tiết của hàm F .

18

Chặt bớt MAC dựa trên PRF

Bỏ dè dẽ. Giả sử $F : K \times X \rightarrow \{0,1\}^n$ là một PRF an toàn. Vậy thi

$$F_t(k, m) := F(k, m)[1 \dots t] \quad \text{với mọi } 1 \leq t \leq n$$

cũng là PRF an toàn.

Hệ quả. Nếu (S, V) là một MAC dựa trên PRF an toàn với output là tag độ dài n -bit, vậy thì MAC bị cắt chỉ lấy w bit cũng là an toàn khi $1/2^n$ là "không đáng kể". (Ví dụ, $w \geq 64$).

19

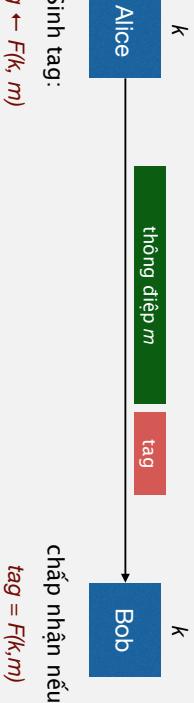
MAC an toàn từ PRF an toàn

Xét PRF $F : K \times X \rightarrow Y$ ta định nghĩa MAC

$$I_F = (S, V)$$

bởi

- $S(k, m) := F(k, m)$
- $V(k, m, t) := [\text{'yes' nếu } t = F(k, m); \text{'no' nếu ngược lại}]$



Sinh tag:

$$tag \leftarrow F(k, m)$$

Ví dụ: AES là một MAC với thông điệp độ dài 16 byte.

Câu hỏi: làm thế nào chuyển từ MAC nhỏ sang MAC lớn?

Trả lời: Có hai cách xây dựng được dùng trong thực tế.

- CBC-MAC (Ngân hàng - ANSI X9.9, X9.19, FIPS 186-3)
- HMAC (Giao thức cho Internet: SSL, IPsec, SSH,...)

Cả hai cách này đều chuyển từ một PRF nhỏ thành PRF-lớn.

Sinh tag:

$$tag = F(k, m)$$

17

20

MAC và PRF

Nhắc lại:

- PRF an toàn $F \Rightarrow$ MAC an toàn, khi $|Y|$ lớn.
- Cách xây dựng: $S(k, m) = F(k, m)$

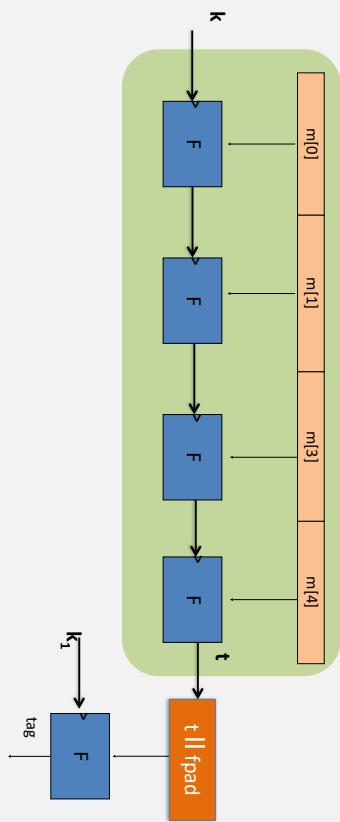
Mục đích của chúng ta:

- Từ PRF cho thông điệp ngắn (Ví dụ AES), tìm cách xây dựng PRF cho thông điệp dài tùy ý.

Xây dựng 2: NMAC

(nested MAC)

cascade



Xét $F: K \times X \rightarrow K$ là PRF.

Ta định nghĩa PRF:

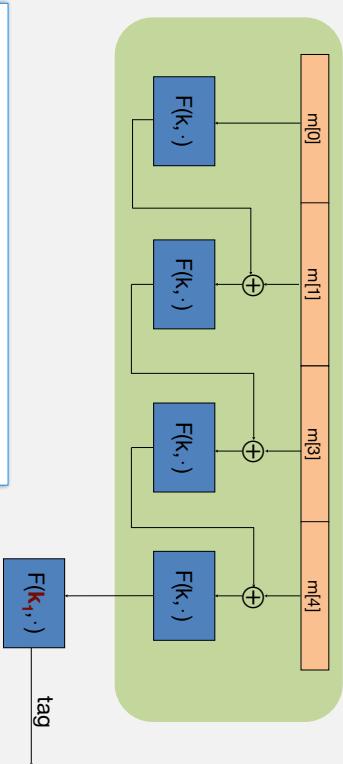
$$F_{\text{NMAC}} : K^2 \times X^{\leq L} \rightarrow K$$

24

Xây dựng 1: ECBC-MAC

(CBC-MAC được mã hóa)

raw CBC



Xét $F: K \times X \rightarrow X$ là PRF.

Ta định nghĩa PRF:

$$F_{\text{ECBC}} : K^2 \times X^{\leq L} \rightarrow X$$

23



<https://class.coursera.org/crypto-preview/class/index>

Tại sao trong bước cuối của ECBC-MAC phải mã hóa?

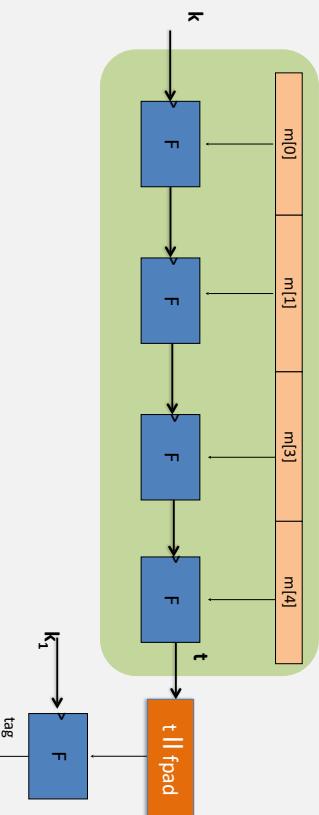
Giả sử ta định nghĩa MAC $I_{RAW} = (S, V)$ với

$$S(k, m) = \text{rawCBC}(k, m)$$

Vậy thì I_{RAW} có thể bị phá dẽ dàng dùng tấn công chọn 1 thông điệp.

Kẻ tấn công thực hiện:

- Chọn một thông điệp chỉ một khối $m \in X$.
- Truy vấn để lấy tag cho m . Anh ta được $t = F(k, m)$.
- Output $\textcolor{red}{t}$ như một MAC giả cho thông điệp gồm 2 khối $(m, t \oplus m)$.



$$\forall x, y, w: F_{\text{NMAC}}(k, x) = F_{\text{NMAC}}(k, y) \Rightarrow F_{\text{NMAC}}(k, x||w) = F_{\text{NMAC}}(k, y||w)$$

Thật vậy,

$$\text{rawCBC}(k, (m, t \oplus m)) = F(k, F(k, m) \oplus (t \oplus m)) = F(k, t \oplus (t \oplus m)) = t$$

26

Tại sao trong bước cuối của ECBC-MAC và NMAC phải mã hóa?

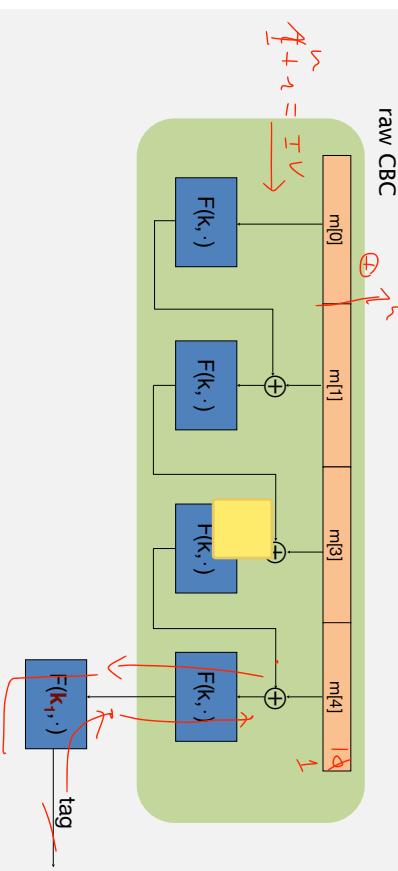
NMAC. Giả sử ta định nghĩa MAC $I = (S, V)$ với

$$S(k, m) = \text{cascade}(k, m)$$

- MAC này là an toàn.
- MAC này có thể bị giả mạo mà không cần truy vấn bất kỳ thông điệp nào.
- MAC này có thể bị giả mạo bằng cách truy vấn một thông điệp.
- MAC này có thể bị giả mạo chỉ bằng truy vấn hai thông điệp.

Tính chất mở rộng của ECBC-MAC

raw CBC



$$\forall x, y, w: F_{\text{ECBC}}(k, x) = F_{\text{ECBC}}(k, y) \Rightarrow F_{\text{ECBC}}(k, x||w) = F_{\text{ECBC}}(k, y||w)$$

25

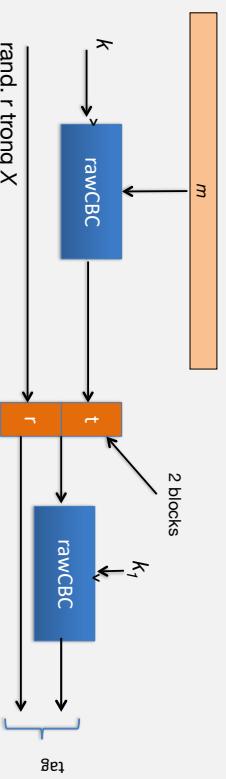
Tính chất mở rộng của NMAC (nested MAC)

cascade

$$S(k, m) = \text{cascade}(k, m)$$

28

Sơ đồ an toàn hơn: Xây dựng ngẫu nhiên RCBC



- PRF: $F: K \times X \rightarrow X$
- Kết quả: MAC với tag trong X^2

30

Tấn công MAC có tính chất mở rộng

Cho PRF $F_{BIG}: K \times X \rightarrow Y$ bê có tính chất mở rộng:

$$\forall x, y, w: F_{BIG}(k, x) = F_{BIG}(k, y) \Rightarrow F_{BIG}(k, x||w) = F_{BIG}(k, y||w)$$

MAC xây dựng từ PRF trên có thể bị tấn công theo thuật toán sau:

Bước 1: gửi $|Y|^{1/2}$ truy vấn ngẫu nhiên cho các thông điệp trong X .
đặt được (m_i, t_i) for $i = 1, \dots, |Y|^{1/2}$

Bước 2: tìm một xung đột $t_u = t_v$ for $u \neq v$ (với xác suất cao là tìm
được theo nghịch lý ngày sinh)

Bước 3: chọn một w và truy vấn để lấy $t := F_{BIG}(k, m_u||w)$

Bước 4: đưa ra cặp giả mạo $(m_u||w, t)$.

Thật vậy $t := F_{BIG}(k, m_u||w)$.



MÃ XÁC THỰC THÔNG ĐIỆP

- ▶ Toàn vẹn thông điệp
- ▶ MAC dựa trên PRF
- ▶ CBC-MAC và NMAC
- ▶ MAC padding

<https://iclass.coursera.org/crypto-preview/class/index>

So sánh

ECBC-MAC thường dùng là MAC dựa trên AES

- Mode mã hóa CCM (dùng trong 802.11i)
- Chuẩn NIST gọi là CMAC

NMAC thường không dùng với AES hoặc 3DES

- Lý do chính: phải đổi khóa AES trên mọi block \Rightarrow phải tính lại AES key expansion.
- Nhưng NMAC là cơ sở cho MAC được dùng phổ biến là HMAC.

29

31

Nếu kích thước thông điệp không là bội của block-size thì sao?

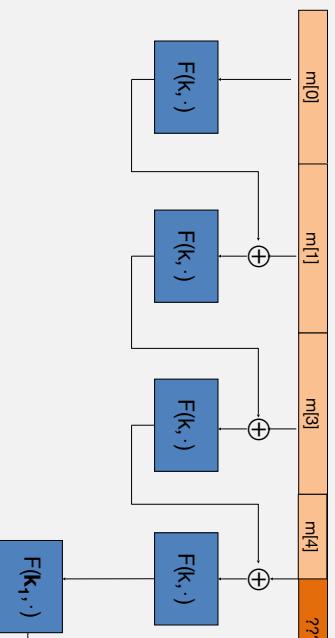
CBC MAC padding

Để an toàn, padding phải khả nghịch, tức là:

$$m_0 \neq m_1 \Rightarrow \text{pad}(m_0) \neq \text{pad}(m_1)$$

ISO. pad với "1000...00". Thêm block giả nếu cần.

- Số '1' chỉ ra vị trí bắt đầu của pad.

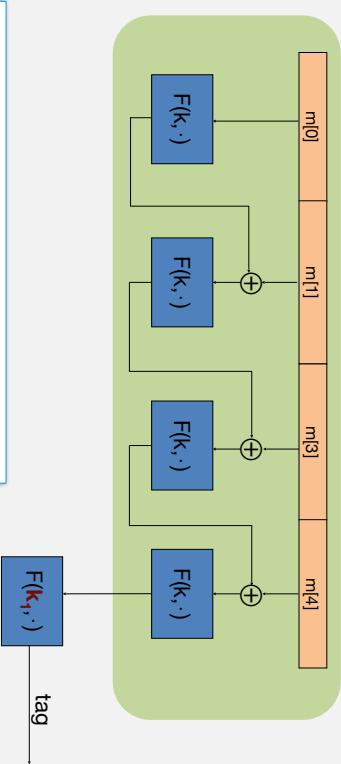


34

35

Nhắc lại: ECBC-MAC

raw CBC



Xét $F: K \times X \longrightarrow X$ là PRF.

Ta định nghĩa PRF:

$$F_{\text{ECBC}} : K^2 \times X^{\leq L} \rightarrow X$$

CBC MAC padding

Ý tưởng ngây thơ: pad m với dãy 0



Câu hỏi: MAC thu được có an toàn?

- Có, MAC này an toàn.
- Còn phụ thuộc vào thiết kế chi tiết của MAC.
- Không, nếu lấy được tag của m , kẻ tấn công cũng lấy được tag của $m||0$.

Xây dựng 4: HMAC (Hash-MAC)

Được dùng rộng rãi trên Internet.

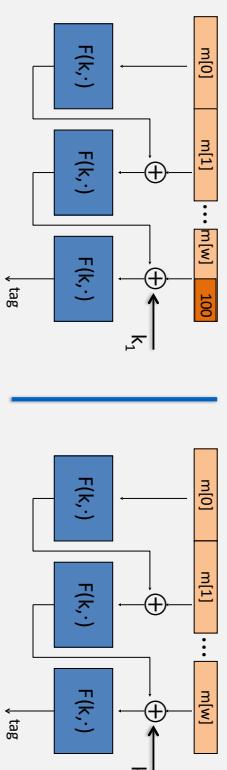
... nhưng, trước hết ta cần xem xét khái niệm hàn băm mật mã.

38

CMAC (Chuẩn NIST)

Là một biến thể của ECBC-MAC: với key = (k, k_1, k_2)

- Không cần bước mã hóa cuối (**tấn công mở rộng** bị chặn bằng cách **xor** với **khóa cuối**)
- Không cần block giả (nhập nhằng được loại bỏ bằng cách sử dụng khóa k_1 hoặc k_2)



Tài liệu đọc thêm và trình bày

- J. Black, P. Rogaway: CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. J. Cryptology 18(2): 111-131 (2005)
- K. Pietrzak: A Tight Bound for EMAC. ICALP (2) 2006: 168-179
- J. Black, P. Rogaway: A Block-Cipher Mode of Operation for Parallelizable Message Authentication. EUROCRYPT 2002: 384-397
- M. Bellare: New Proofs for NMAC and HMAC: Security Without Collision-Resistance. CRYPTO 2006: 602-619
- Y. Dodis, K. Pietrzak, P. Puniya: A New Mode of Operation for Block Ciphers and Length-Preserving MACs. EUROCRYPT 2008: 198-219

37

39

Tính kháng xung đột

Định nghĩa. Xét hàm băm $H: M \rightarrow T$ với $|M| >> |T|$. Một xung đột cho H là một cặp $m_0, m_1 \in M$ thỏa mãn :

$$H(m_0) = H(m_1) \quad \text{và} \quad m_0 \neq m_1$$

HÀM BĂM KHÁNG XUNG ĐỘT

Giới thiệu

- ▶ **Tần công dùng nghịch lý ngày sinh**
- ▶ **Sơ đồ Merkle-Damgard**
- ▶ **Xây dựng hàm nén**
- ▶ **HMAC: MAC dựa trên SHA256**
- ▶ **Timing Attack cho MAC**

<https://tchss.coursera.org/crypto-preview/class/index>

Ví dụ:

- SHA-256: Output là 256 bit.

4

Nhắc lại: Toàn vẹn thông điệp

MAC xây dựng dựa trên PRF:

- ECB-MAC, CMAC : Thường dùng với AES (Ví dụ, 802.11i)
- NMAC: làm cơ sở cho HMAC
- PMAC: một MAC song song

MAC ngẫu nhiên:

- ▶ **Giới thiệu**
- ▶ **Tần công dùng nghịch lý ngày sinh**
- ▶ **Sơ đồ Merkle-Damgard**
- ▶ **Xây dựng hàm nén**
- ▶ **HMAC: MAC dựa trên SHA256**
- ▶ **Timing Attack cho MAC**

<https://tchss.coursera.org/crypto-preview/class/index>

3

Xây dựng MAC từ hàm kháng xung đột

$$S_{big}(k, m) = S(k, H(m)) ; \quad V_{big}(k, m, t) = V(k, H(m), t)$$

Tính kháng xung đột là cần:

- Nếu kẻ tấn công có thể tìm được $m_0 \neq m_1$ sao cho $H(m_0) = H(m_1)$,
 - vậy thì MAC không còn an toàn trước tấn công chọn 1 bẩn rõ:
 - bước 1: kẻ tấn công truy vấn $t \leftarrow S(k, m_0)$
 - bước 2: output (m_1, t) là cặp thông điệp/tag giả mạo

6

Xây dựng MAC từ hàm kháng xung đột

Xây dựng. Xét $I = (S, V)$ là MAC cho thông điệp ngắn trên (K, M, T) (Ví dụ, AES). Xét hàm băm $H: M_{big} \rightarrow M$.

Ta định nghĩa $h_{big} = (S_{big}, V_{big})$ trên (K, M_{big}, T) như sau:

$$S_{big}(k, m) = S(k, H(m)) ; \quad V_{big}(k, m, t) = V(k, H(m), t)$$

Định lý. Nếu I là một MAC an toàn và H hàm kháng xung đột, vậy thì h_{big} là MAC an toàn.

Ví dụ:

$$S(k, m) = AES_{2-block-cbc}(k, SHA-256(m))$$
 là một MAC an toàn.



HÀM BĂM KHÁNG XUNG ĐỘT

<https://class.coursera.org/crypto-preview/course/index>

Timing Attack cho MAC



- ▶ Giới thiệu
- ▶ Tấn công dùng nghịch lý ngày sinh
- ▶ Số đồ Merkle-Damgard
- ▶ Xây dựng hàm nén
- ▶ HMAC: MAC dựa trên SHA256

Bảo vệ sự toàn vẹn của file dùng hàm băm kháng xung đột

Gói phần mềm:



Không gian chung
(chỉ đọc)

$$\underline{H(F_1)} \quad \underline{H(F_2)} \\ \dots \\ \underline{H(F_n)}$$

Toàn vẹn:

- Khi người dùng download file, chỉ ta có thể kiểm tra nội dung có khớp với mã băm
- H kháng xung đột \Rightarrow kẻ tấn công không thể sửa gói phần mềm mà không bị phát hiện
 - Không cần khóa (mỗi người đều có thể kiểm tra tính toàn vẹn), nhưng cần không gian lưu trữ công khai

5

Nghịch lý ngày sinh nhật

Thuật toán tân công hàm băm $H: M \rightarrow \{0,1\}^n$

Thuật toán:

1. Chọn $2^{n/2}$ thông điệp ngẫu nhiên trong $M: m_1, \dots, m_{2^{n/2}}$

(xác suất chúng phân biệt nhau là cao)

2. For $i = 1, \dots, 2^{n/2}$:

tính $t_i = H(m_i) \in \{0,1\}^n$.

3. Tìm xung đột ($t_i = t_j$). Nếu không thấy thì quay lại bước 1.

$$\begin{aligned}
 \text{Chứng minh. } \Pr[\exists i \neq j : r_i = r_j] &= 1 - \Pr[\forall i \neq j : r_i \neq r_j] \\
 &= 1 - \left(\frac{B-1}{B} \right) \left(\frac{B-2}{B} \right) \cdots \left(\frac{B-n+1}{B} \right) \\
 &= 1 - \prod_{i=1}^{n-1} \left(1 - \frac{i}{B} \right) \geq 1 - \prod_{i=1}^{n-1} e^{-i/B} \\
 &= 1 - e^{-1/B \sum_{i=1}^{n-1} i} \\
 &\geq 1 - e^{-n^2/(2B)} \\
 &\geq 1 - e^{-0.72} = 0.53
 \end{aligned}$$

10

Thuật toán tân công hàm băm

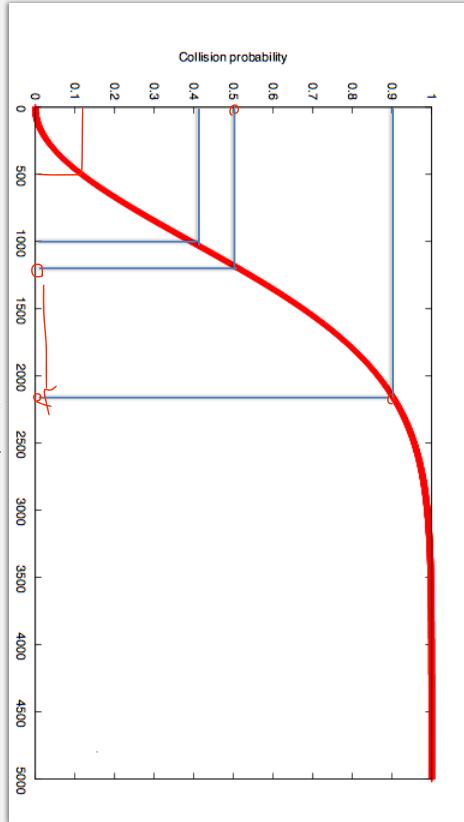
- Xét hàm băm $H: M \rightarrow \{0,1\}^n$ với $|M| \gg 2^n$
- Thuật toán sau cho phép tìm xung đột sau $O(2^{n/2})$ lần băm.

Thuật toán:

- Chọn $2^{n/2}$ thông điệp ngẫu nhiên trong $M: m_1, \dots, m_{2^{n/2}}$
(với xác suất chúng phân biệt là cao)
- For $i = 1, \dots, 2^{n/2}$:
tính $t_i = H(m_i) \in \{0,1\}^n$.
- Tìm xung đột ($t_i = t_j$). Nếu không thấy thì quay lại bước 1.

11

$B = 10^6$



12

Thuật toán lượng tử

Thuật toán cổ điển	Thuật toán lượng tử
Tấn công vét cạn hệ $E: K \times X \rightarrow X$	$O(K)$ $O(K ^{1/2})$
Tìm xung đột cho hàm băm $H: M \rightarrow T$	$O(T ^{1/2})$ $O(T ^{1/3})$

14

Một số hàm băm kháng xung đột

Crypto++5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

hàm	mã băm (số bit)	tốc độ (MB/gây)	thời gian tấn công
SHA-1	160	153	280
SHA-256	256	111	2128
SHA-512	512	99	2256
Whirlpool	512	57	2256

* thuật toán tốt nhất tìm xung đột cho SHA-1 cần 2^{51} lần tính mã băm.

13

Nhắc lại: Hàm băm kháng xung đột

Định nghĩa. Xét hàm băm $H: M \rightarrow T$ với $|M| >> |T|$. Một xung đột cho H là một cặp $m_0, m_1 \in M$ thỏa mãn :
 $H(m_0) = H(m_1)$ và $m_0 \neq m_1$

$$H(m_0) = H(m_1) \quad \text{và} \quad m_0 \neq m_1$$

Mục đích:

- Xây dựng hàm băm kháng xung đột
- Cho một hàm băm kháng xung đột cho thông điệp kích thước nhỏ,
- hãy xây dựng hàm băm cho thông điệp kích thước lớn.

Bước đầu tiên:

HÀM BĂM KHÁNG XUNG ĐỘT

- Giới thiệu
- Tấn công dùng nghịch lý ngày sinh
- Sơ đồ Merkle-Damgard
- Xây dựng hàm nén
- HMAC: MAC dựa trên SHA256
- Timing Attack cho MAC

<https://cse555.csail.mit.edu/crypto-preview/classes/index>

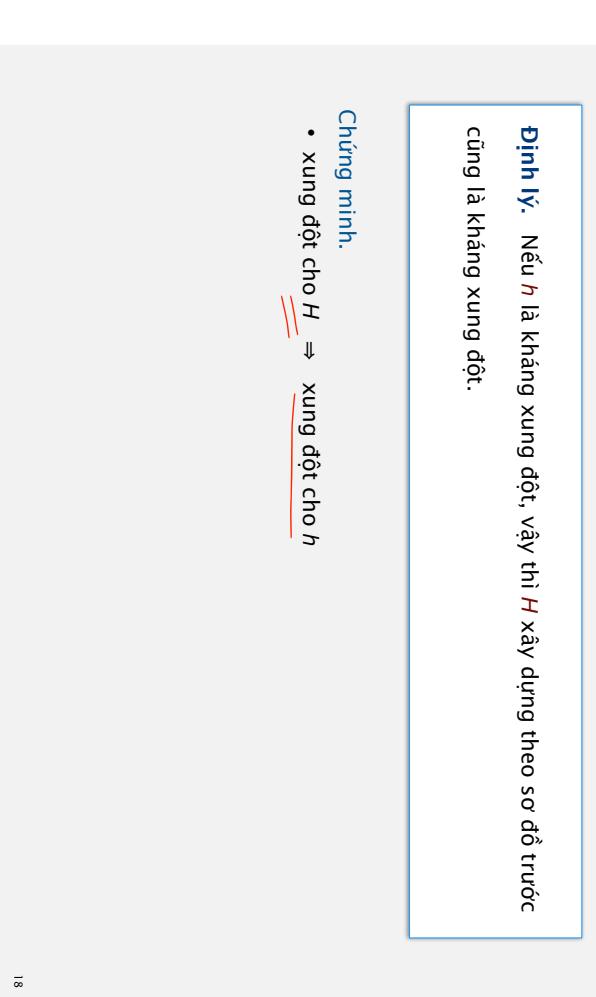
16

Tính kháng xung đột cho sơ đồ MD

Định lý. Nếu h là kháng xung đột, vậy thì H xây dựng theo sơ đồ trước cũng là kháng xung đột.

Chứng minh.

- xung đột cho $H \Rightarrow$ xung đột cho h



18

Vấn đề

Làm thế nào xây dựng được hàm băm kháng xung đột cho thông điệp kích thước nhỏ?

Cho trước hàm nén $h: T \times X \rightarrow T$

Xây dựng:

- Hàm băm $H: X^{\leq L} \rightarrow T$.

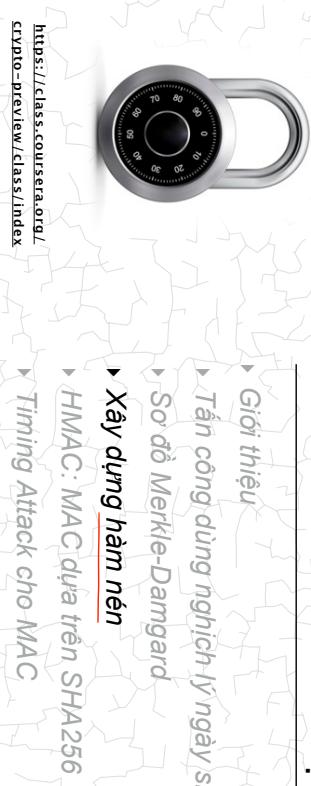
Padding:

1000...0 || msg len
64 bits

- Nếu không đủ không gian cho padding, vậy thì thêm block mới.

17

HÀM BĂM KHÁNG XUNG ĐỘT



<https://tutorials.csail.mit.edu/crypto-preview/course/index>

- Giới thiệu
- Tấn công dùng nghịch lý ngày sinh
- Sơ đồ Merkle-Damgård
- Xây dựng hàm nén
- HMAC: MAC dựa trên SHA256
- Timing Attack cho MAC

<https://tutorials.csail.mit.edu/crypto-preview/course/index>

19

Hãy chọn đáp án đúng

- Giả sử ta định nghĩa

$$h(H, m) = E(m, H)$$

- Vậy thì hàm $h(.,.)$ không kháng xung đột:
- để tìm xung đột (H, m) và (H', m') ta chọn ngẫu nhiên (H, m, m')

- và xây dựng H' như sau.

1. $H' = D(m', E(m, H))$

$$h(H', m) = E(m', D(m', E(m, H))) = E(m, H)$$

2. $H' = E(m', D(m, H))$

3. $H' = E(m', E(m, H))$

4. $H' = D(m', D(m, H))$

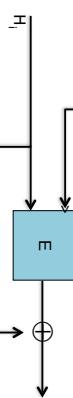
22

Hàm nén từ mã khối

Xây dựng. Xét hệ mã khối $\underline{E}: K \times \{0,1\}^n \rightarrow \{0,1\}^n$. Hàm nén Davies-Meyer

xây dựng bởi:

$$h(H, m) = E(m, H) \oplus H$$



Định lý. Giả sử E là một hệ mã lý tưởng (tập gồm $|K|$ hoán vị ngẫu nhiên).
Tìm một xung đột $h(H, m) = h(H', m')$ mất $O(2^{n/2})$ lần tính (E, D) .

Các cách xây dựng khác

- Để đơn giản, ta xét $E: \underline{\{0,1\}^n} \times \{0,1\}^n \rightarrow \{0,1\}^n$
- Miyaguchi-Preneel: $h(H, m) = E(m, H) \oplus H \oplus m$ (Whirlpool)
 $h(H, m) = E(H \oplus m, m) \oplus m$
 có 12 biến thể như vậy
- Các biến thể khác là không an toàn, ví dụ
 $h(H, m) = E(m, H \oplus m)$
 (Bài tập)

$$\begin{aligned} h(H', m) &= E(m', D(m', h(H, m) \text{ xor } m')) \text{ xor } m' \\ &= h(H, m) \text{ xor } m \text{ xor } m' \\ &\equiv h(H, m) \end{aligned}$$

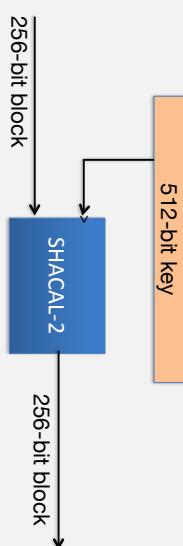
sinh ngẫu nhiên H, m, m' và tìm H' :
 $H' = D(m', h(H, m) \text{ xor } m')$

$$H' = D(m', E(m, H) \text{ xor } m' \text{ xor } m)$$

24

Case study: SHA-256

- Merkle-Damgård function
- Davies-Meyer compression function
- Block cipher: SHACAL-2



23

MAC từ hàm băm theo sơ đồ Merkle-Damgard

Xây dựng thử nghiệm:

- Xét $H: X^{sl} \rightarrow T$ là một hàm băm kháng xung đột theo sơ đồ Merkle-Damgard

Ta xây dựng

$$S(k, m) = H(k \| m)$$

- MAC này là không an toàn bởi vì:
 - Cho $H(k \| m)$ có thể tính $H(w \| k \| m \| PB)$ với mọi w .
 - Cho $H(k \| m)$ có thể tính $H(k \| m \| w)$ với mọi w .
 - Cho $H(k \| m)$ có thể tính $H(k \| m \| PB \| w)$ với mọi w .
 - Mọi người đều có thể tính $H(k \| m)$ với mọi m .

28

HÀM BĂM KHÁNG XUNG ĐỘT



<https://ihasc.net/cryptopreview/classic/index.html>

HMAC: MAC dựa trên SHA256

Timing Attack cho MAC

Hàm nén có thể chứng minh an toàn

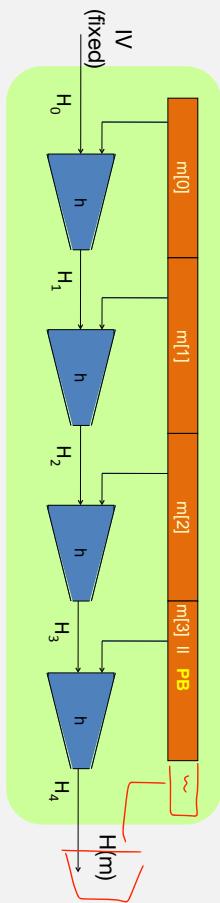
- Chọn một số nguyên tố ngẫu nhiên p kích thước 2000-bit và các số ngẫu nhiên $1 \leq u, v \leq p$.
- Với mỗi $m, h \in \{0, \dots, p-1\}$ ta định nghĩa

$$h(H, m) = u^H \cdot v^m \pmod{p}$$

Sự kiện. Tìm xung đột cho $h(\dots)$ là khó như giải bài toán “discrete-log” modun p .

Vấn đề: hàm nén này chậm.

Nhắc lại: Xây dựng theo sơ đồ Merkle-Damgard



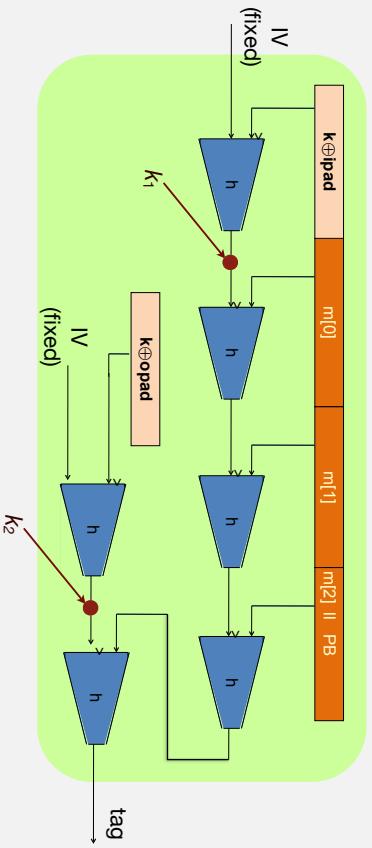
Định lý: h kháng xung đột $\Rightarrow H$ kháng xung đột.

Câu hỏi:

- Liệu chúng ta có thể sử dụng $H(\cdot)$ trực tiếp để xây dựng MAC?

25

Hình mô tả HMAC



Tương tự như NMAC PRF

- khác biệt chính: hai khóa k_1 và k_2 phụ thuộc nhau.

30

31

Phương pháp chuẩn: HMAC (Hash-MAC)

Được dùng rộng rãi trên Internet

Hàm băm H

- Ví dụ: SHA-256 ; output là 256 bits

Ta xây dựng MAC từ hàm băm:

$$\text{HMAC: } S(k, m) = H(\underbrace{k \oplus \text{opad}}_{k_1} \parallel H(\underbrace{k \oplus \text{ipad}}_{k_2} \parallel m))$$

Tính chất của HMAC

- Xây dựng từ cài đặt của SHA-256

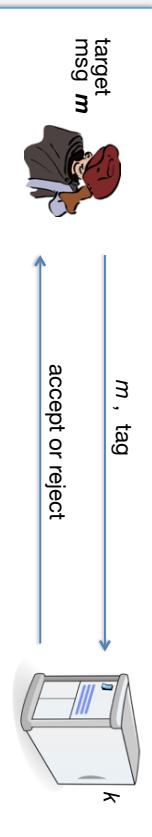
- HMAC được giả sử là một PRF an toàn
 - Có thể chứng minh với một số giả sử PRF về $h(.,.)$
 - Chặn về an toàn tương tự như NMAC:
Khi $\sigma^2/|T|$ là "không đáng kể".

- Trong TLS: có hỗ trợ HMAC-SHA1-96



29

Chú ý: Timing Attacks vào hàm kiểm tra



Timing attack:

- Để tính tag cho một thông điệp m ta thực hiện:

- Truy vấn server để lấy một tag ngẫu nhiên
- Lặp lại mọi khả năng của byte đầu tiên và gửi đến server, dùng khi hàm verification chạy nhanh hơn so với thời gian thực hiện bước 1
- Lặp lại với mọi byte trong tag cho đến khi tìm được tag.

m

3	5	3	*	*	*	*	*
---	---	---	---	---	---	---	---

34

Chú ý: Timing Attacks vào hàm kiểm tra

Ví dụ: Keyczar crypto library (Python)

[Đã đơn giản hóa]

```
def Verify(key, msg, sig_bytes):  
    return HMAC(key, msg) == sig_bytes
```

Vấn đề:

- Cài đặt của phép toán ' $=$ ' so sánh tuần tự từng byte
- Sẽ trả lại `false` ngay khi gặp byte khác nhau đầu tiên.

Chống Timing Attack #2

Phương pháp

- Đảm bảo rằng phép toán so sánh sẽ luôn thực hiện với thời gian bằng nhau trên mọi dữ liệu

```
def Verify(key, msg, sig_bytes):  
    mac = HMAC(key, msg)  
    return mac == HMAC(key, sig_bytes)
```

Đảm bảo

- Kẻ tấn công không biết giá trị nào đang được so sánh.

Chống Timing Attack #1

Phương pháp

- Đảm bảo rằng phép toán so sánh sẽ luôn thực hiện với thời gian bằng nhau trên mọi dữ liệu

```
def Verify(key, msg, sig_bytes):  
    result = 0  
    for x, y in zip(HMAC(key, msg), sig_bytes):  
        result |= ord(x) ^ ord(y)  
    return result == 0
```

Vấn đề:

- Không đảm bảo được việc trình biên dịch không sửa lại đoạn mã trong khi tối ưu

36

Đừng tự cài đặt crypto !

Bài học

Nội dung

Chữ ký phụ thuộc tài liệu

- **Chữ ký số là gì?**

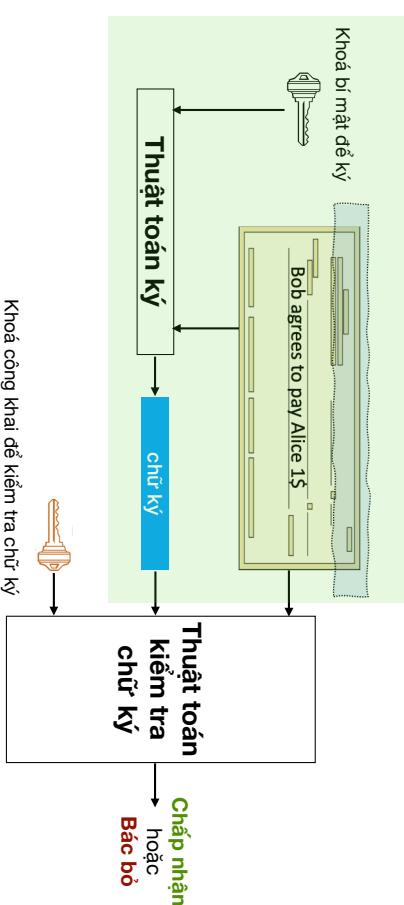
- Ứng dụng

- Sơ đồ chữ ký số RSA

- Sơ đồ chữ ký số ElGamal

- Chuẩn chữ ký số DSA

2

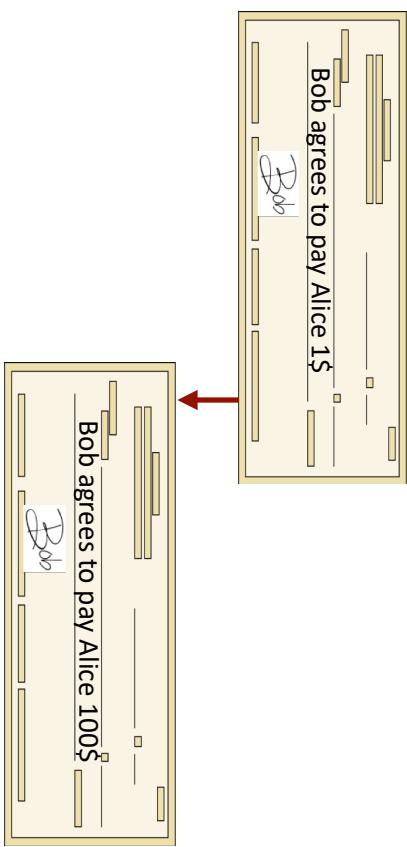


4

Chữ ký vật lý

Nhập môn An toàn Thông tin

Chữ ký số



3

Chữ ký số

Tính an toàn 1

Định nghĩa. Một sơ đồ chữ ký số bao gồm ba thuật toán

- $Gen()$ thuật toán ngẫu nhiên output ra cặp khoá (pk, sk)
- $S(sk, m \in M)$ output ra chữ ký σ
- $V(pk, m, \sigma)$ output 'chấp nhận' hoặc 'bá c bỏ'

6

Tấn công chọn bản rõ

- Kẻ tấn công có thể lấy được chữ ký chọn q thông điệp tùy chọn m_1, m_2, \dots, m_q

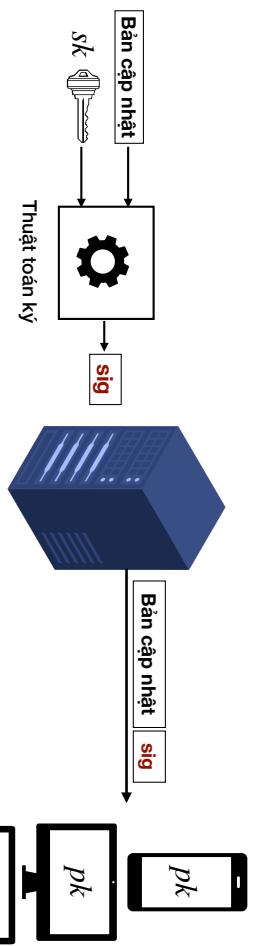
ký hiệu

$$\sigma_i = S(sk, m_i) \quad \text{với } i = 1, \dots, q$$

8

Ví dụ thực tế

Khách hàng



Kiểm tra sig ,
cài đặt nếu hợp lệ

Tính đúng đắn

- Với mọi cặp (pk, sk) sinh bởi thuật toán $Gen()$,
- và với mọi thông điệp $m \in M$, ta có

$$V(pk, m, S(sk, m)) = \text{'chấp nhận'}$$

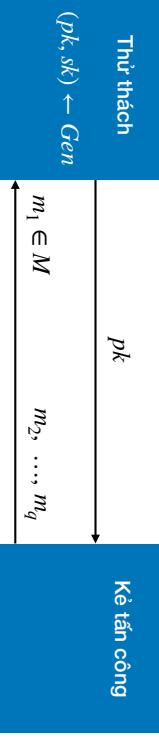
5

- Khả năng của kẻ tấn công là

7

Chữ ký an toàn

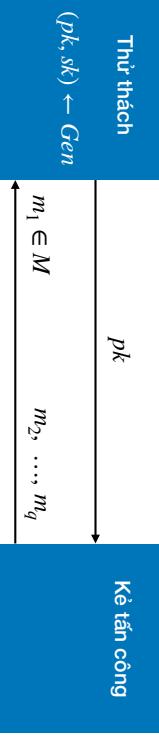
Nội dung



Kẻ tấn công **thắng** nếu $V(pk, m, \sigma) = \text{'chấp nhận'}$ và $m \notin \{m_1, \dots, m_q\}$

10

- Chữ ký số là gì?
- **Ứng dụng**
- Sơ đồ chữ ký số RSA
- Sơ đồ chữ ký số ElGamal
- Chuẩn chữ ký số DSA



Kẻ tấn công **thắng** nếu $V(pk, m, \sigma) = \text{'chấp nhận'}$ và $m \notin \{m_1, \dots, m_q\}$

11

Tính an toàn

- Mục đích của kẻ tấn công

Giả mạo thông điệp

- Đưa ra được cặp thông điệp/chữ ký hợp lệ (m, σ) mà

$$m \notin \{m_1, \dots, m_q\}$$

- Sơ đồ chữ ký là **an toàn** khi kẻ tấn công không tạo được chữ ký hợp lệ cho thông điệp mới.

Chữ ký an toàn

- Mục đích của kẻ tấn công

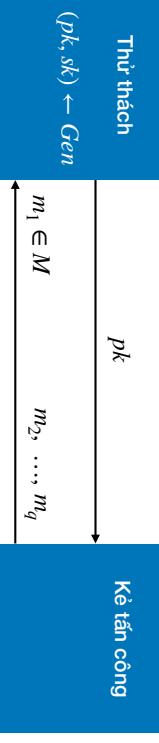
Giả mạo thông điệp

- Đưa ra được cặp thông điệp/chữ ký hợp lệ (m, σ) mà

$$m \notin \{m_1, \dots, m_q\}$$

- Sơ đồ chữ ký là **an toàn** khi kẻ tấn công không tạo được chữ ký hợp lệ cho thông điệp mới.

- Hệ chữ ký là **an toàn** nếu với **mọi** kẻ tấn công A , xác suất A thắng là “nhỏ không đáng kể”

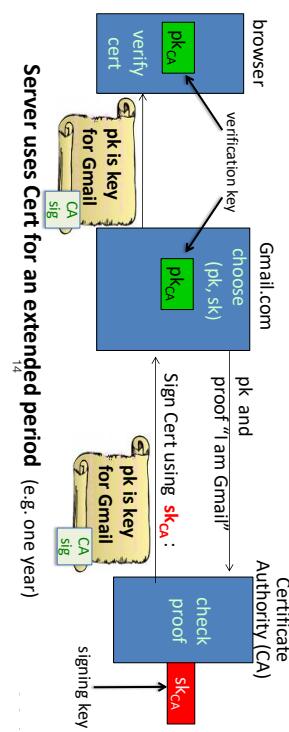


Kẻ tấn công **thắng** nếu $V(pk, m, \sigma) = \text{'chấp nhận'}$ và $m \notin \{m_1, \dots, m_q\}$

12

Chứng chỉ số

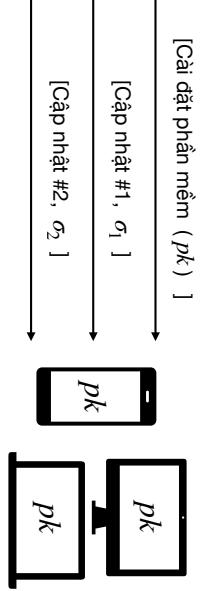
- Vấn đề:** trình duyệt cần khóa công khai của máy chủ để thiết lập khóa phiên
- Giải pháp:** máy chủ yêu cầu bên thứ ba tin cậy (CA) xác thực và ký lên khóa công khai pk



Ký trên phần mềm

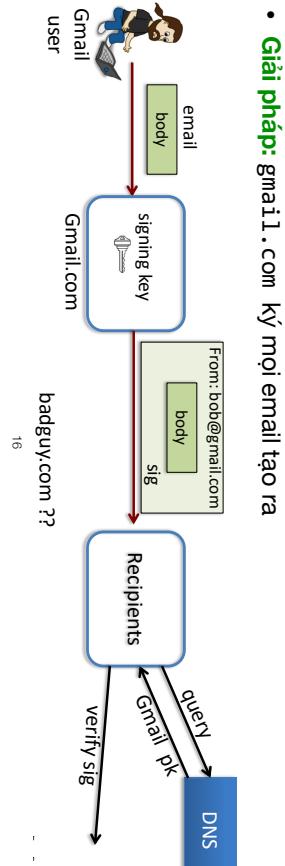
- Công ty bán phần mềm BK ký lên phần mềm

- Khách hàng có khóa công khai pk của BK. Họ sẽ cài đặt phần mềm nếu chữ ký là hợp lệ.



Ký email: DKIM (domain key identified mail)

- Vấn đề:** email giả khẳng định gửi từ someuser@gmaiil.com nhưng thực tế, mail gửi từ badguy.com
⇒ Làm gmail.com giống như một nguồn gửi email giả



Chứng chỉ số

The screenshot shows the following certificate information:

- Serial Number:** 581474488373390497
- Version:** 3
- Signature Algorithm:** SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)
- Parameters:** none
- Not Valid Before:** Wednesday, July 31, 2013 4:59:24 AM Pacific Daylight Time
- Not Valid After:** Thursday, July 31, 2014 4:59:24 AM Pacific Daylight Time
- Public Key Info:**
 - Algorithm: Elliptic Curve Public Key (1.2.840.10045.2.1)
 - Parameters: Elliptic Curve secp256r1 (1.2.840.10045.1.7)
 - Public Key: 65 bytes: 04 71 8C DD E0 0A C9 76 ...
 - Key Size: 256 bits
 - Key Usage: Encrypt, Verify, Derive
 - Signature: 256 bytes: 8A 38 F5 E7 F6 59 ...
- Details:**
 - Subject Name: mail.google.com
 - State/Province: California
 - Locality: Mountain View
 - Organization: Google Inc.
 - Common Name: mail.google.com
 - Issuer Name: Google Inc.
 - Organization: Google Inc.
 - Common Name: Google Internet Authority C2

Ba cách tiếp cận cho toàn vẹn thông điệp

Phương pháp tổng quát

1. **Hàm băm kháng xung đột:** cần không gian công khai chỉ đọc
2. **Mã xác thực thông điệp:** với mỗi khách hàng, người bán phải tính một MAC mới của phần mềm
⇒ phải quản lý một khoá bí mật dùng lâu dài (để sinh khoá MAC cho mỗi khách hàng)
3. **Chữ ký số:** người bán phải quản lý khoá bí mật dùng lâu dài
 - Chữ ký kèm với phần mềm
 - Phần mềm có thể download ở nơi không tin cậy

18

- Ý tưởng được Diffie & Hellman đưa ra năm 1976
- Xây dựng lược đồ chữ ký từ hệ mã khoá công khai đơn định (E, D)

$$\sigma = S(sk, m) = D(sk, m)$$

$$V(pk, m, \sigma) = \begin{cases} 1 & \text{nếu } E(pk, \sigma) = m \\ 0 & \text{ngược lại} \end{cases}$$

20

Khi nào sử dụng chữ ký số

Nếu **một** phía ký và **một** phía kiểm tra : **dùng MAC**

- Phải tương tác để có khoá chia sẻ
- Bên nhận có thể sửa đổi dữ liệu và ký lại nó trước khi chuyển dữ liệu tới bên thứ ba

Nếu **một** bên ký và **nhiều** bên kiểm tra: **dùng chữ ký số**

- Bên nhận không thể nào sửa dữ liệu nhận được trước khi chuyển dữ liệu tới bên thứ ba
- Không chối bỏ được

Nội dung

- Chữ ký số là gì?
- Ứng dụng

• **Sơ đồ chữ ký số RSA**

- Sơ đồ chữ ký số ElGamal
- Chuẩn chữ ký số DSA

17

19

Tấn công 1

Ví dụ: Sinh khoá

- Chọn $p = 3$ và $q = 11$
- $n = p \cdot q = 33$

- $\phi(n) = (3 - 1)(11 - 1) = 20$
- Chọn $e = 3$

- $d = e^{-1} = 7 \pmod{20}$
- Output ($pk = 3, sk = 7$)

22

- Có thể tạo ra chữ ký của thông điệp cụ thể
 - Ví dụ, dễ tính căn bậc e của thông điệp $m = 1$ hoặc
 - căn bậc ba của thông điệp $m = 8$

24

Ví dụ: Ký và kiểm tra

- Hàm sinh khoá $Gen()$:

- Chọn $n = pq$ (p, q nguyên tố ngẫu nhiên λ -bit)

- Chọn e, d thỏa mãn $ed = 1 \pmod{\phi(n)}$

- $pk = (n, e)$ và $sk = (n, d)$

• Hàm ký $S(sk, m) = m^d \pmod{n}$

• Hàm kiểm tra chữ ký $V(pk, m, \sigma) = 1 \Leftrightarrow \sigma^e = m \pmod{n}$

21

Kiểm tra chữ ký

$V(pk = e = 3, m = 4, \sigma = 16)$:

Tạo chữ ký
 $S(sk = d = 7, m = 4)$:

$$\begin{aligned} & \bullet \sigma = m^d \pmod{n} \\ & = 4^7 = 16 \pmod{33} \\ & \bullet \text{Do } m = m' \text{ nên 'chấp nhận'} \end{aligned}$$

23

Bài tập

Sơ đồ Băm và Ký

- $Gen(): [...]$
- $S(sk, m) = H(m)^d \mod n$
- $V(pk, m, \sigma) = 1 \Leftrightarrow \sigma^e = m \mod n$

Hãy chứng minh hệ chữ ký Textbook RSA

$$S(sk, m) = m^d \mod n$$

$$V(pk, m, \sigma) = 1 \Leftrightarrow \sigma^e = m \mod n$$

là không an toàn bằng cách chỉ ra rằng: từ chữ ký của thông điệp m ta có thể tạo ra chữ ký cho thông điệp m^2 .

- **Hỏi:** Có dễ tạo chữ ký cho thông điệp
- **Trả lời:** Tùy thuộc vào hàm băm H

26

Tấn công 2

Tấn công 3

- Có thể kết hợp hai chữ ký để thu được chữ ký thứ ba
 - Giả sử σ_1, σ_2 là chữ ký hợp lệ của thông điệp m_1, m_2
 - Khi đó $\sigma = [\sigma_1 \cdot \sigma_2 \mod n]$ là chữ ký hợp lệ của thông điệp $m = m_1 \cdot m_2$ bởi vì
$$(\sigma_1 \cdot \sigma_2)^e = \sigma_1^e \cdot \sigma_2^e = m_1 \cdot m_2$$

25

27

Sơ đồ Băm và Ký với RSA

- Nếu giả sử RSA là đúng, và H được mô hình như một hàm ngẫu nhiên (ánh xạ lên \mathbb{Z}_n^*), thì sơ đồ băm và ký với RSA là **an toàn**

• Trên thực tế, H được sửa đổi từ hàm băm mật mã quen thuộc

- Phải đảm bảo miền giá trị của H là đủ lớn!

- Một lựa chọn “tốt” cho H là:

$$H(m) = n \text{ byte đầu tiên của} \\ SHA256(1\|m)\parallel SHA256(2\|m)\parallel \dots \parallel SHA256(11\|m)$$

30

Trực giác cho tính an toàn

- Quay lại với các cách tấn công trước...

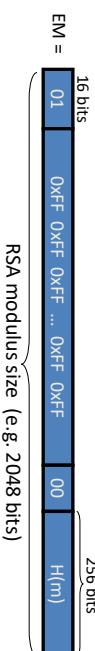
1. Không dễ tính căn bậc e của $H(1), \dots$

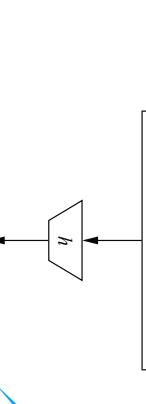
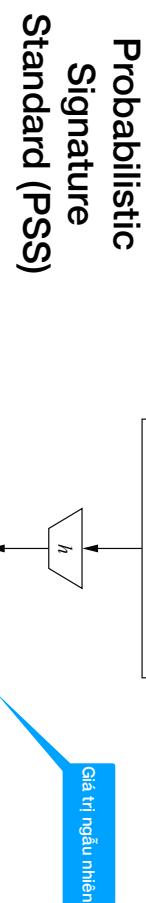
2. Chọn σ nhưng làm thế nào tìm được m để $H(m) = \sigma^e \pmod{n}$?

\Rightarrow Hàm H nên là hàm một chiều

3. $H(m_1) \cdot H(m_2) = \sigma_1^e \cdot \sigma_2^e = (\sigma_1 \cdot \sigma_2)^e \neq H(m_1 \cdot m_2)$

Chữ ký PKCS1 v1.5

- Hoán vị cửa sổ RSA : $pk = (n, e), sk = (n, d)$
 - $S(sk, m \in M)$:
- EM = 
- RSA modulus size (e.g. 2048 bits)
- output $\sigma = (EM)^d \pmod{n}$
- $V(pk, m \in M, \sigma)$: kiểm tra $\sigma^e \pmod{n}$ có dạng đúng như ở trên



29

Hệ chữ ký số ElGamal

Textbook ElGamal: Hàm ký

- $S(sk = d, m)$:

1. Chọn khóa tạm thời $k_E \in \{1, 2, \dots, p - 2\}$ thỏa mãn $\gcd(k_E, p - 1) = 1$ (để có $k_E^{-1} \bmod p - 1$)
2. Tính
$$r = g^{k_E} \bmod p,$$
$$s = \frac{(m - d \cdot r)}{k_E} \bmod p - 1$$

3. Output chữ ký $\sigma = (r, s)$

³⁶

- Chữ ký ElGamal rất khác so với hệ mật mã ElGamal.

³⁴

Nội dung

Textbook ElGamal: Sinh khóa

- Chữ ký số là gì?
- Ứng dụng
- Sơ đồ chữ ký số RSA
- **Sơ đồ chữ ký số ElGamal**
- Chuẩn chữ ký số DSA

³³

³⁵

Chú ý:

ElGamal: Tính đúng đắn

Ví dụ: Kiểm tra chữ ký

- $t = (g^d)^r \cdot r^s \pmod{p} = (g^d)^r \cdot (g^{k_E})^s \pmod{p}$
 $= g^{dr + s \cdot k_E} \pmod{p}$
- Do định lý Fermat nhỏ, điều kiện $g^m = t \pmod{p}$ tương đương với
 $m = (d \cdot r + s \cdot k_E) \pmod{p-1}$
- và tương đương với điều kiện

$$s = \frac{m - d \cdot r}{k_E} \pmod{p-1}$$



- Kiểm tra** $V(pk = g^d = 7, m = 26, \sigma = (3,26))$:
- $t = (g^d)^r \cdot r^s = 7^3 \cdot 3^{26} = 22 \pmod{29}$
- $g^m = 2^{26} = 22 \pmod{29}$
- Do $t = g^m \pmod{29}$ nên '**chấp nhận**'

38

40

Textbook ElGamal: Kiểm tra chữ ký

Ví dụ: Sinh khoá và chữ ký

Sinh khoá $Gen()$

- $V(pk = g^d, m, \sigma = (r, s))$:
- 1. Tính giá trị
 $t = (g^d)^r \cdot r^s \pmod{p}$
- 2. if $t = g^m \pmod{p}$ return '**chấp nhận**' else '**bác bỏ**'

Tạo chữ ký $S(sk = 12, m = 26)$:

- chọn $p = 29$
- chọn $g = 2$
- chọn $sk = d = 12$
 - $s = (m - d \cdot r) \cdot k_E^{-1} \pmod{p-1}$
 $= -10 \cdot 17 \pmod{28}$
 $= 26 \pmod{28}$
- Output chữ ký $(3, 26)$

37

39

Bài tập

- Biết rằng tham số và khoá công khai của Bob là:

- $p = 29, g = 2, g^d = 7$

- Giả sử Bob đã ký hai thông điệp với cùng khoá tạm k_E :

- $[m_1, (r, s_1)] = [26, (3, 26)]$

- $[m_2, (r, s_2)] = [13, (3, 1)]$

- Hãy tính khoá bí mật s_k của Bob.

42

Giả mạo chữ ký cho thông điệp “ngẫu nhiên”	
Tạo chữ ký và thông điệp	Kiểm tra chữ ký
<ul style="list-style-type: none">Chọn hai số i, j thoả mãn $\gcd(j, p - 1) = 1$Tính chữ ký $r = g^{i \cdot (g^d)^j} \mod p$ $s = -r \cdot j^{-1} \mod p - 1$Tính thông điệp $m = s \cdot i \mod p - 1$	<ul style="list-style-type: none">Tính $t = (g^d)^r \cdot r^s \mod p$bởi vì $t = g^m \mod p$ nên chữ ký là hợp lệ

- Kiểm tra chữ ký

- $t = (g^d)^r \cdot r^s \mod p$

- Tính $t = (g^d)^r \cdot r^s \mod p$

$$\begin{aligned} r &= g^{i \cdot (g^d)^j} \mod p \\ s &= -r \cdot j^{-1} \mod p - 1 \end{aligned}$$

44

Không an toàn khi sử dụng lại k_E

Nếu ta ký hai thông điệp m_1 và m_2 cùng sử dụng khoá tạm k_E , khi đó

$$s_1 = \frac{m_1 - dr}{k_E} \mod p - 1 \quad \text{và} \quad s_2 = \frac{m_2 - dr}{k_E} \mod p - 1$$

Kè tessel công Oscar sẽ tính được $k_E = \frac{m_1 - m_2}{s_1 - s_2} \mod p - 1$

và tính được khoá bí mật

$$d = \frac{m_1 - s_1 k_E}{r} \mod p - 1$$

41

Bài tập

- Với Textbook ElGamal, liệu bạn có thể tạo ra chữ ký hợp lệ của một thông điệp “ngẫu nhiên” (tương tự như với Textbook RSA)?

43

Nội dung

- Chữ ký số là gì?
- Ứng dụng
- Sơ đồ chữ ký số RSA
- Sơ đồ chữ ký số ElGamal

- Chuẩn chữ ký số DSA

46

DSA: Sinh khoá

- Sinh số nguyên tố p với $2^{1023} < p < 2^{1024}$
- Tìm một ước nguyên tố của q của $p - 1$ với $2^{159} < q < 2^{160}$
- Tìm một phần tử sinh g với cấp $\text{ord}(g) = q$; tức là g sinh nhóm con với q phần tử
- Chọn số ngẫu nhiên d với $0 < d < q$
- Tính $\beta = g^d$
- Output $pk = (p, q, g, \beta)$ và $sk = d$

48

Băm và ký với ElGamal

The Digital Signature Algorithm (DSA)

- Giống RSA, sơ đồ Băm và Ký không những tăng tính hiệu quả mà còn tăng độ an toàn.
- Nó giúp chống lại tấn công giả mạo chữ ký cho thông điệp “ngẫu nhiên”
- Trong sơ đồ này, phương trình ký trở thành

$$s = \frac{(H(m) - d \cdot r)}{k_E} \pmod{p-1}$$

- Chuẩn chữ ký số của Mỹ
- Được xuất bởi Viện tiêu chuẩn quốc gia (NIST)
- **Ưu điểm:**

- Độ dài chữ ký chỉ 320 bit
- Một số phương pháp tấn công sơ đồ chữ ký ElGamal không áp dụng được cho sơ đồ này

45

47

DSA: Kiểm tra chữ ký

- Tính $w = s^{-1} \bmod q$
- Tính $u_1 = w \cdot H(m) \bmod q$
- Tính $u_2 = w \cdot r \bmod q$
- Tính $v = (g^{u_1} \cdot (g^d)^{u_2} \bmod p) \bmod q$
- Hàm kiểm tra $V(pk, m, (r, s))$ như sau:
`if v = r mod q return 'chấp nhận' else 'báć bở'`

50

Sinh khoá

-
- | | | |
|--|----------------------------|---|
| • Chọn $p = 59$ | • Chọn $q = 29$ | • Chọn $r = (3^{10} \bmod 59) \bmod 29 = 20 \bmod 29$ |
| • Chọn $g = 3$ | • Khoá bí mật $sk = d = 7$ | • $s = (26 + 7 \cdot 20) \cdot 3 \bmod 29 = 5 \bmod 29$ |
| • Khoá công khai $pk = g^d = 4 \bmod 59$ | | |
-

Ký thông điệp $H(m) = 26$

- | | |
|---|--|
| • Chọn khoá tạm thời $k_E = 10$ | • $r = (3^{10} \bmod 59) \bmod 29 = 20 \bmod 29$ |
| • $s = (26 + 7 \cdot 20) \cdot 3 \bmod 29 = 5 \bmod 29$ | |

52

DSA: tham số và mức an toàn

p	q	Chữ ký	Kích thước mã băm	Mức an toàn
1024	160	320	160	80
2048	224	448	224	112
3072	256	512	256	128

49

Ví dụ: DSA sinh khoá và ký

Tính toán

Sinh số nguyên tố cho DSA

- **Bài toán:**
 1. Tìm số nguyên tố q với $2^{159} < q < 2^{160}$ dùng thuật toán Miller – Rabin
 2. **for** $i = 1$ to 4096

- Làm thế nào để tìm một nhóm vòng \mathbb{Z}_p^* kích thước 1024 bit, và
 - có nhóm con nguyên tố kích thước 2^{160}
- **Phương pháp:**
 - Sinh số nguyên tố q kích thước 160 bit và xây dựng số p từ nó
 - **if** p là số nguyên tố: **return** (p, q) // Dùng thuật toán Miller – Rabin
 - 3. Quay lại bước 1

54

56

Ví dụ: Kiểm tra chữ ký

Kiểm tra $V(pk = 4, H(m) = 26, (r = 20, s = 5))$:

- $w = 5^{-1} = 5 \pmod{29}$
- $u_1 = 6 \cdot 26 = 11 \pmod{29}$
- $u_2 = 6 \cdot 20 = 4 \pmod{29}$
- $v = (3^{11} \cdot 4^4 \pmod{59}) \pmod{29} = 20$
- Do $v = r \pmod{29}$ nên chữ ký là '**hợp lệ**'

53

Sinh số nguyên tố cho DSA

Output:

- hai số nguyên tố (p, q)
với $2^{1023} < p < 2^{1024}$ và $2^{159} < q < 2^{160}$
sao cho $p - 1$ là bội của q

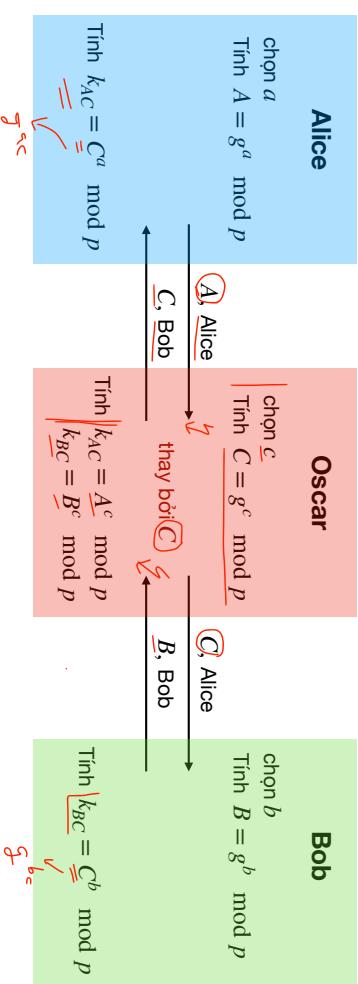
55



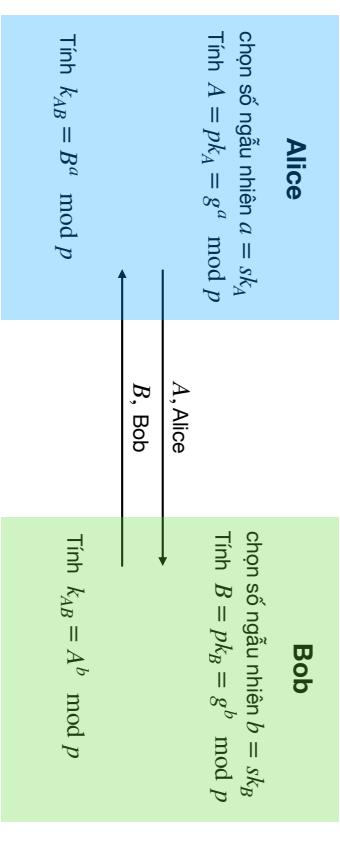
Nội dung

Man-in-the-Middle Attack

- **Chứng chỉ số**
- Cơ sở hạ tầng khoá công khai



Giao thức trao đổi khoá Diffie-Hellman



Sinh Chứng chỉ số với khoá người dùng cấp

Vấn đề

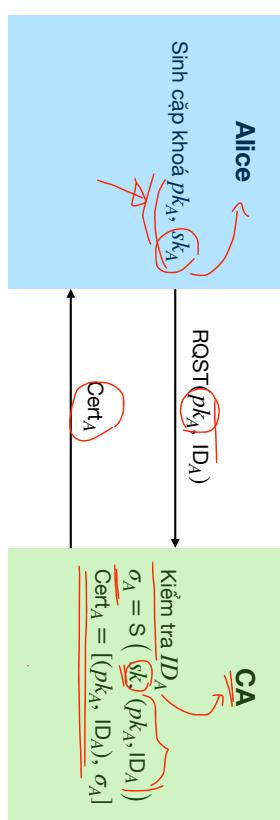
- Khoá của người dùng Alice:

$k_A = (pk_A, ID_A)$
với ID_A là thông tin định danh, ví dụ địa chỉ IP hoặc tên kèm ngày sinh; và
khoá công khai pk_A là một xâu nhị phân (độ dài 2048 bit).

- Khi Oscar thực hiện tấn công, anh ta phải thay đổi khoá thành:

$$k = (pk_O, ID_A)$$

- Giải pháp:** Phải xác thực cặp (pk_A, ID_A) là “hợp lệ”.



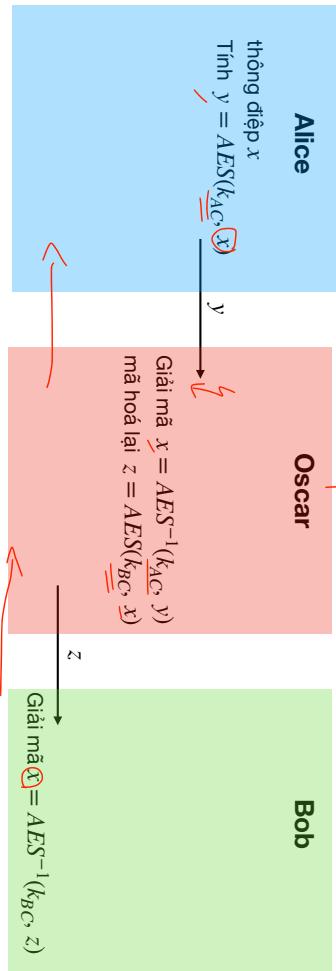
Sau khi Man-in-the-Middle Attack

Giải pháp

- Chứng chỉ số** cho người dùng Alice:

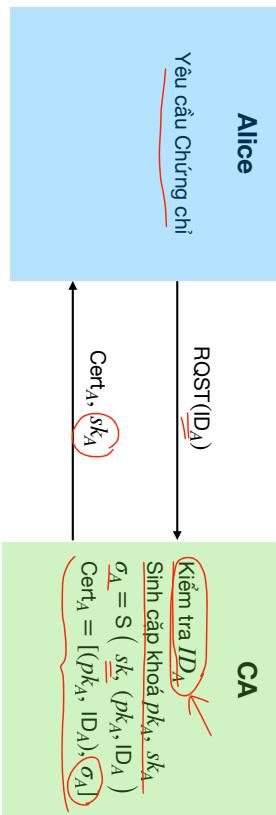
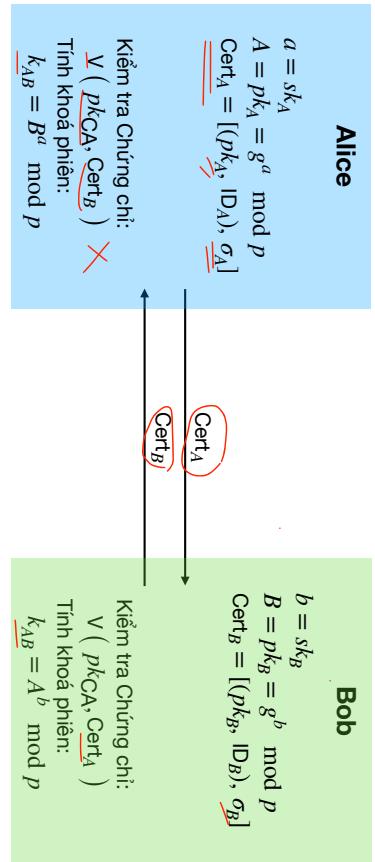
$Cert_A = [(pk_A, ID_A), \sigma]$
với $\sigma = S(sk, (pk_A, ID_A))$ là chữ ký số tạo bởi người có thẩm quyền
cấp Chứng chỉ số (CA, Certificate Authority)

- Chứng chỉ số gắn định danh của người dùng với khoá công khai của anh ta**



Trao đổi khoá Diffie-Hellman với Chứng chỉ số

Hệ tầng khoá công khai



Sinh Chứng chỉ và khoá

Nội dung

Chứng chỉ số X509 (tiếp)

Ví dụ: Chứng chỉ số X.509

Ví dụ: Chứng chỉ số X.509

Details	
Subject Name	mail.google.com
Common Name	
Issuer Name	
Country or Region	US
Organization	Google Trust Services LLC
Common Name	GTS CA 1C3
Serial Number	68 FE 3F 57 2A 7A 32 EA 0A 00 00 00 00 CE 57 C5
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.1)
Parameters	None
Not Valid Before	Monday, 3 May 2021 at 18:24:58 Indochina Time
Not Valid After	Monday, 26 July 2021 at 18:24:57 Indochina Time

Subject
Subject's Public Key:
- Algorithm - Parameters - Public Key

Signature
Signature : 256 bytes : 96 AF C4 29 E8 4E 26 A4 ...

- **Subject:** Thông tin về ID_A hoặc ID_B như trong ví dụ trước. Thường xác định thông tin như tên người trong tổ chức.
- **Subject's Public Key:** Khoa công khai được xác thực bởi Chứng chỉ. Gồm dây bit tương ứng với khóa công khai và thuật toán (ví dụ: Diffie-Hellman) và tham số thuật toán.
- **Signature:** Chữ ký trên mọi trường của Chứng chỉ.

Kiểm tra khoá công khai của CA

Alice

CA2

RQST(Cert_{CA2})

Cert_{CA2}

$V(pk_{CA1}, Cert_{CA2})$
 $\Rightarrow pk_{CA2}$ hợp lệ
 $V(pk_{CA2}, Cert_B)$
 $\Rightarrow pk_B$ hợp lệ

Dãy CA

Alice

Bob

pk_{CA1}
???

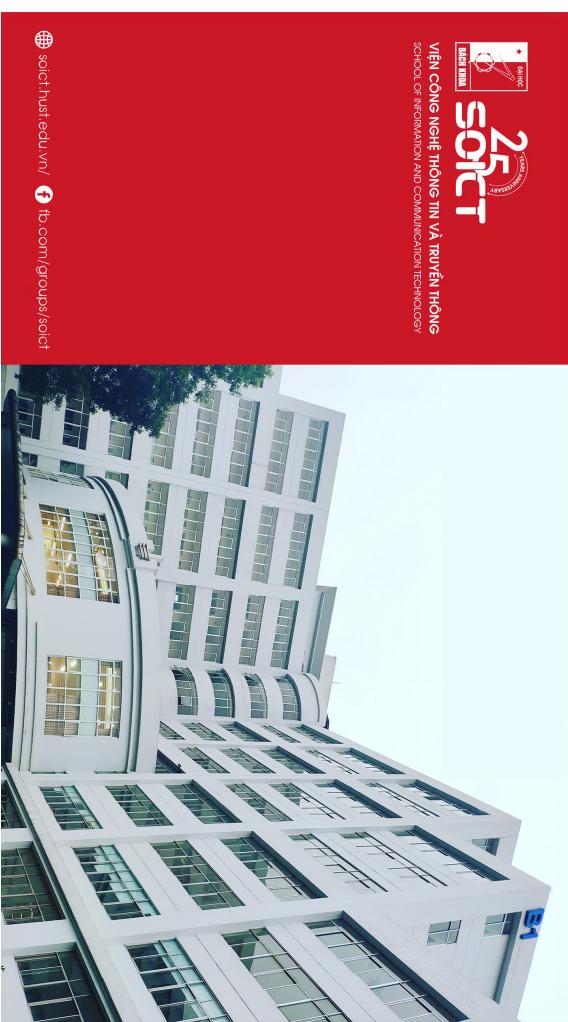
$Cert_B$

pk_{CA2}
 $Cert_B = [(pk_B, ID_B), \sigma_B(sk_{CA2})]$

- Chứng chỉ số của Alice được cấp bởi CA1

- CA1 cấp chứng chỉ số uỷ quyền cho CA2

- Chứng chỉ số của Bob được cấp bởi CA2



- J. Hoffstein, J. Pipher, J. H. Silverman, *An Introduction to Mathematical Cryptography*, Springer-Verlag – Undergraduate Texts in Mathematics, 2nd Ed., 2014.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to Algorithms*, Third Edition (3rd ed.). The MIT Press. 2009.
- H. H. Khoái, *Nhập môn số học thuật toán*

- $\mathbb{N} = \{1, 2, 3, \dots\}$
- $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$

Nội dung

1 Thuật toán Euclid

2 Số học đồng dư

3 Số nguyên tố và trường hữu hạn

Ngày 23 tháng 3 năm 2023

4 Lũy thừa và căn nguyên thủy trong trường hữu hạn

Định nghĩa

Xét $a, b \in \mathbb{Z}$. Ta nói
 b là ước của a , hay
 a chia hết cho b
nếu có một số nguyên c sao cho

$$a = bc.$$

Ta viết $b \mid a$ để chỉ a chia hết cho b . Nếu a không chia hết cho b thì ta viết $b \nmid a$.

Ví dụ

- $847 \mid 485331$ vì $485331 = 847 \cdot 573$.
- $355 \nmid 259943$ vì $259943 =$

5 / 57

Định nghĩa

Nhập môn số học thuật toán | Thuật toán Euclid

Định nghĩa

Xét $a, b \in \mathbb{Z}$. Ta nói
 b là ước của a , hay
 a chia hết cho b

nếu có một số nguyên c sao cho

$$a = bc.$$

Ta viết $b \mid a$ để chỉ a chia hết cho b . Nếu a không chia hết cho b thì ta viết $b \nmid a$.

7 / 57

Định nghĩa

Nhập môn số học thuật toán | Thuật toán Euclid

Mệnh đề

Xét $a, b, c \in \mathbb{Z}$.

- 1 Nếu $a \mid b$ và $b \mid c$, thì $a \mid c$.
- 2 Nếu $a \mid b$ và $b \mid a$, thì $a = \pm b$.
- 3 Nếu $a \mid b$ và $a \mid c$, thì $a \mid (b+c)$ và $a \mid (b-c)$.

5 / 57

Định nghĩa

Bài tập

Hãy chứng minh mệnh đề trước.

Định nghĩa

- Uớc chung của hai số nguyên a và b là số nguyên d thỏa mãn:

$$d \mid a \text{ và } d \mid b.$$

- Ta ký hiệu $\gcd(a, b)$ là ước chung **lớn nhất** của a và b .

Ví dụ

- $\gcd(12, 18) = 6$ vì $6 \mid 12$ và $6 \mid 18$ và không có số nào lớn hơn có tính chất này.
- $\gcd(748, 2014) = 44$ vì

các ước của $748 = \{1, 2, 4, 11, 17, 22, 34, 44, 68, 187, 374, 748\}$,
các ước của $2024 = \{1, 2, 4, 8, 11, 22, 23, 44, 46, 88, 92, 184, 253, 506, 1012, 2024\}$.

8 / 57

Định nghĩa (Chia lấy dư)

Xét a, b là các số nguyên dương. Ta nói a chia cho b có thương là q và phần dư là r nếu

$$a = b \cdot q + r \quad \text{với} \quad 0 \leq r < b.$$

Bài tập

Hãy chứng minh rằng các số q và r ở trên xác định duy nhất bởi a và b .

10 / 57

Định nghĩa

- Ước chung của hai số nguyên a và b là số nguyên d thỏa mãn:

$$d \mid a \text{ và } d \mid b.$$

- Ta ký hiệu $\gcd(a, b)$ là ước chung **lớn nhất** của a và b .

Ví dụ: Tính gcd(2024, 748)

$$2024 = 748 \cdot 2 + 528$$

$$748 = 528 \cdot 1 + 220$$

$$528 = 220 \cdot 2 + 88$$

$$220 = 88 \cdot 2 + 44$$

$$88 = 44 \cdot 2 + 0$$

$$\leftarrow \quad \text{gcd} = 44$$

12 / 57

Định lý

Phép chia (Bước 3) của Thuật toán Euclid thực hiện nhiều nhất

$$\log_2(b) + 2 \quad \text{lần.}$$

14 / 57

Thuật toán tính gcd(a, b)

Định lý (Thuật toán Euclid)

Xét a, b là hai số nguyên dương với $a \geq b$. Thuật toán sau đây tính $\text{gcd}(a, b)$ sau một số hữu hạn bước.

- 1 Đặt $r_0 = a$ và $r_1 = b$.
- 2 Đặt $i = 1$.

- 3 Chia r_{i-1} cho r_i , ta được

$$r_{i-1} = r_i \cdot q_i + r_{i+1} \quad \text{với} \quad 0 \leq r_{i+1} < r_i.$$

- 4 Nếu $r_{i+1} = 0$, vậy thì

$$r_i = \text{gcd}(a, b)$$

và thuật toán kết thúc.

- 5 *Ngược lại*, $r_{i+1} > 0$, vậy thì đặt $i = i + 1$ và quay lại Bước 3.

Thuật toán Euclid mở rộng

- Thuật toán Euclid có thể mở rộng để tìm thêm một số thông tin.
- Cụ thể, chúng ta mở rộng thuật toán để tính thêm hệ số x, y thỏa mãn
$$d = \gcd(a, b) = ax + by.$$
- Các hệ số x, y có thể âm hoặc bằng 0. Các hệ số này sẽ có ích sau này khi tích phân tử nghịch đảo trong số học modun.

16 / 57

Tính đúng đắn của thuật toán

- Thuật toán tìm (d, x, y) thỏa mãn
$$d' = \gcd(a, b) = ax + by$$
- Nếu $b = 0$, vậy thì
$$d = a = a \cdot 1 + b \cdot 0.$$
- Nếu $b \neq 0$, thuật toán EXTENDED-EUCLID sẽ tính (d', x', y') thỏa mãn
$$\begin{aligned} d' &= d = \gcd(b, a \bmod b) \\ &= b x' + (a \bmod b) y' \end{aligned}$$

- Và vây thì

$$\begin{aligned} d &= b' x' + (a - b \lfloor a/b \rfloor) y' \\ &= a y' + b(x' - \lfloor a/b \rfloor y') \end{aligned}$$

18 / 57

Thuật toán Euclid mở rộng

- Input :* Cặp số nguyên dương (a, b)
- Output:* Bộ ba (d, x, y) thỏa mãn

$$d = \gcd(a, b) = ax + by.$$

```

EXTENDED-EUCLID(a, b)
  if b == 0
    return (a, 1, 0)
  else
    return EUCLID(b, a mod b)
  
```

```

EXTENDED-EUCLID(a, b)
  if b == 0
    return (a, 1, 0)
  else
    (d', x', y') = EXTENDED-EUCLID(b, a mod b)
    (d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')
  return (d, x, y)
  
```

Định nghĩa

Xét số nguyên $m \geq 1$. Ta nói hai số nguyên a và b là đồng dư theo modun m nếu $a - b$ chia hết cho m , và viết

$$a \equiv b \pmod{m}$$

Số m được gọi là modun.

Đồng hồ có thể được viết theo như modun dùng modun $m = 12$:

$$6 + 9 = 15 \equiv 3 \pmod{12} \quad \text{và} \quad 2 - 3 = -1 \equiv 11 \pmod{12}$$

Ví dụ

Nhập môn số học thuật toán | Thuật toán Euclid

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1	3	-11	14
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

Nội dung

1 Thuật toán Euclid

2 Số học đồng dư

3 Số nguyên tố và trường hữu hạn

4 Lũy thừa và căn nguyên thủy trong trường hữu hạn

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo.
- Lời gợi ý thủ tục EXTENDED-EUCLID(99, 78) trả về $(3, -11, 4)$ thỏa mãn $\gcd(99, 78) = 3 = 99 \cdot (-11) + 78 \cdot 14$.

Mệnh đề

Xét số nguyên $m \geq 1$.

I Nếu $a_1 \equiv a_2 \pmod{m}$ và $b_1 \equiv b_2 \pmod{m}$, vậy thì

$$\begin{aligned} a_1 \pm b_1 &\equiv a_2 \pm b_2 \pmod{m}, \quad \text{và} \\ a_1 \cdot b_1 &\equiv a_2 \cdot b_2 \pmod{m}. \end{aligned}$$

24 / 57

Bài tập

- Lấy $m = 5$ và $a = 2$. Rõ ràng $\gcd(2, 5) = 1$, vậy thì tồn tại nghịch đảo của a theo modun 5. Hãy tìm a^{-1} .

25 / 57

Mệnh đề

Xét số nguyên $m \geq 1$.

I Nếu $a_1 \equiv a_2 \pmod{m}$ và $b_1 \equiv b_2 \pmod{m}$, vậy thì

$$\begin{aligned} a_1 \pm b_1 &\equiv a_2 \pm b_2 \pmod{m}, \quad \text{và} \\ a_1 \cdot b_1 &\equiv a_2 \cdot b_2 \pmod{m}. \end{aligned}$$

2 Xét số nguyên a . Vậy thì tồn tại số nguyên b thỏa mãn

$$a \cdot b \equiv 1 \pmod{m} \text{ nếu và chỉ nếu } \gcd(a, m) = 1.$$

Nếu tồn tại số b như vậy thì ta nói b là nghịch đảo của a theo modun m .

23 / 57

Định nghĩa

Ta viết

$$\mathbb{Z}/m\mathbb{Z} = \{0, 1, 2, \dots, m-1\}$$

và gọi $\mathbb{Z}/m\mathbb{Z}$ là **vành số nguyên modun m** .

Nhận xét

Khi chúng ta thực hiện phép cộng hoặc nhân trong $\mathbb{Z}/m\mathbb{Z}$ ta luôn chia hết quả cho m và lấy phần dư.

26 / 57

Định nghĩa

Ta biết rằng a có nghịch đảo modun m nếu và chỉ nếu $\gcd(a, m) = 1$. Các số khả nghịch gọi là **đơn vị**. Ta ký hiệu tập mọi đơn vị bởi

$$\begin{aligned} (\mathbb{Z}/m\mathbb{Z})^* &= \{a \in \mathbb{Z}/m\mathbb{Z} : \gcd(a, m) = 1\} \\ &= \{a \in \mathbb{Z}/m\mathbb{Z} : a \text{ có nghịch đảo theo modun } m\} \end{aligned}$$

Tập $(\mathbb{Z}/m\mathbb{Z})^*$ được gọi là **nhóm đơn vị theo modun m** .

28 / 57

Bài tập

- Lấy $m = 5$ và $a = 2$. Rõ ràng $\gcd(2, 5) = 1$, vậy thì tồn tại nghịch đảo của a theo modun 5. Hãy tìm a^{-1} .
- Tương tự $\gcd(4, 15) = 1$. Hãy tìm 4^{-1} theo modun 15.

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

	·	0	1	2	3	4
0	0	0	0	0	0	0
1	1	0	1	2	3	4
2	2	1	0	2	4	1
3	3	2	1	0	3	4
4	4	3	2	1	0	4

Bảng: Cộng và nhân theo modun 5

25 / 57

27 / 57

Tính lũy thừa nhanh

Ví dụ
Nhóm đơn vị theo modun 7 là

$$(\mathbb{Z}/7\mathbb{Z})^* = \{1, 2, 3, 4, 5, 6\}$$

vì các từ 1 đến 6 đều nguyên tố cùng nhau với 7. Bảng nhân của nhóm này được xác định như dưới đây.

.	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

30 / 57

Ví dụ
Giả sử ta muốn tính

$$3^{218} \pmod{1000}.$$

Đầu tiên, ta viết 218 ở dạng cơ số 2:

$$218 = 2 + 2^3 + 2^4 + 2^6 + 2^7.$$

Vậy thì 3^{218} trở thành

$$3^{218} = 3^{2+2^3+2^4+2^6+2^7} = 3^2 \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^6} \cdot 3^{2^7}.$$

Để ý rằng, dễ tính các mũ

$$3, 3^2, 3^{2^2}, 3^{2^3}, 3^{2^4}, \dots$$

32 / 57

Ví dụ

Nhóm đơn vị theo modun 24 là

$$(\mathbb{Z}/24\mathbb{Z})^* = \{1, 5, 7, 11, 13, 17, 19, 23\}.$$

Bảng nhân của nhóm này xác định như sau:

.	1	5	7	11	13	17	19	23
1	1	5	7	11	13	17	19	23
5	5	1	11	7	17	13	23	19
7	7	11	1	5	19	23	13	17
11	11	7	5	1	23	19	17	13
13	13	17	19	23	1	5	7	11
17	17	13	23	19	5	1	11	7
19	19	23	13	17	7	11	1	5
23	23	19	17	13	11	7	5	1

Định nghĩa

Phi hàm Euler là hàm $\phi(m)$ định nghĩa bởi luật

$$\begin{aligned}\phi(m) &= \#(\mathbb{Z}/m\mathbb{Z})^* \\ &= \#\{0 \leq a < m : \gcd(a, m) = 1\}.\end{aligned}$$

$$\begin{aligned}\phi(24) &= 8 \quad \text{và} \quad \phi(7) = 6.\end{aligned}$$

Thuật toán tính nhanh $a^b \pmod{n}$

Nội dung

MODULAR-EXPONENTIATION(a, b, n)

```

 $c = 0$ 
 $d = 1$ 
Biểu diễn  $b = \langle b_k, b_{k-1}, \dots, b_0 \rangle_2$ 
for  $i = k$  downto 0
   $c = 2c$ 
   $d = (d \cdot d) \pmod{n}$ 
  if  $b_i == 1$ 
     $c = c + 1$ 
   $d = (d \cdot a) \pmod{n}$ 
return  $d$ 

```

34 / 57

Nội dung

1 Thuật toán Euclid

2 Số học đồng dư

3 Số nguyên tố và trường hữu hạn

4 Lũy thừa và căn nguyên thủy trong trường hữu hạn

Ví dụ

Ví dụ (tiếp)

Ta lập bảng

i	0	1	2	3	4	5	6	7
$3^{2^i} \pmod{1000}$	3	9	81	561	721	841	281	961

rồi tính

$$\begin{aligned}
 3^{218} &= 3^2 \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^6} \cdot 3^{2^7} \\
 &\equiv 9 \cdot 561 \cdot 721 \cdot 281 \cdot 961 \pmod{1000} \\
 &\equiv 489 \pmod{1000}.
 \end{aligned}$$

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	1	1	0	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
d	7	49	157	526	160	241	298	166	67	1

■ Kết quả tính $a^b \pmod{n}$ với

$$a = 7, \quad b = 560 = \langle 1000110000 \rangle, \quad \text{và } n = 561$$

- Giá trị được chỉ ra sau mỗi bước lặp.
- Kết quả cuối cùng bằng 1

Định nghĩa

- **Số nguyên tố** là số nguyên lớn hơn 1, không chia hết cho số nguyên dương nào ngoài 1 và chính nó.
- Số nguyên lớn hơn 1 không phải số nguyên tố được gọi là **hợp số**.

Mệnh đề

Xét số nguyên tố p , và giả sử rằng tích ab của hai số a và b chia hết cho p . Vậy thì a hoặc b phải chia hết cho p .
Tổng quát hơn nữa

$$p \mid a_1 a_2 \dots a_n,$$

vậy thì ít nhất một trong các số a_i phải chia hết cho p .

37 / 57

100 số nguyên tố đầu tiên

39 / 57

Định nghĩa

- **Số nguyên tố** là số nguyên lớn hơn 1, không chia hết cho số nguyên dương nào ngoài 1 và chính nó.

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541

37 / 57

38 / 57

Định lý (Định lý cơ bản của số học)

Mỗi số nguyên $a \geq 2$ đều phân tích được thành tích các số nguyên tố

$$a = p_1^{e_1} \cdot p_2^{e_2} \cdot p_3^{e_3} \cdots p_r^{e_r}.$$

Hơn nữa phân tích này là duy nhất nếu các thừa số được viết với thứ tự không giảm.

Định nghĩa

- Định lý cơ bản của số học chỉ ra rằng trong phân tích thừa số nguyên tố của số nguyên dương a , mỗi số nguyên tố p xuất hiện với một số mũ nào đó.

- Ta ký hiệu số mũ này là $\text{ord}_p(a)$ và gọi nó là **cấp** (hoặc **số mũ**) của p trong a .
- Để cho tiện, ta kí hiệu $\text{ord}_p(1) = 0$ với mọi số nguyên tố p .

Bài tập
Hãy chứng minh mệnh đề trước.

Định lý (Định lý cơ bản của số học)

Mỗi số nguyên $a \geq 2$ đều phân tích được thành tích các số nguyên tố

$$a = p_1^{e_1} \cdot p_2^{e_2} \cdot p_3^{e_3} \cdots p_r^{e_r}.$$

Hơn nữa phân tích này là duy nhất nếu các thừa số được viết với thứ tự không giảm.

Định nghĩa

- Định lý cơ bản của số học chỉ ra rằng trong phân tích thừa số nguyên tố của số nguyên dương a , mỗi số nguyên tố p xuất hiện với một số mũ nào đó.

- Ta ký hiệu số mũ này là $\text{ord}_p(a)$ và gọi nó là **cấp** (hoặc **số mũ**) của p trong a .
- Để cho tiện, ta kí hiệu $\text{ord}_p(1) = 0$ với mọi số nguyên tố p .

Bài tập
Hãy chứng minh định lý trước.

Trường hữu hạn \mathbb{F}_p

Mệnh đề

Xét số nguyên tố p . Khi đó mọi phần tử a khác 0 của $\mathbb{Z}/p\mathbb{Z}$ đều có nghịch đảo, có nghĩa rằng, tồn tại b để

$$ab \equiv 1 \pmod{p}.$$

Ta ký hiệu giá trị b này bởi a^{-1} mod p , hoặc đơn giản là a^{-1} nếu p đã xác định.

Mệnh đề này chỉ ra rằng

$$(\mathbb{Z}/p\mathbb{Z})^* = \{1, 2, 3, 4, \dots, p-1\}.$$

$$a = b \text{ thay cho } a \equiv b \pmod{p}.$$

45 / 57

Mệnh đề

- Nếu p nguyên tố, vậy thì tập $\mathbb{Z}/p\mathbb{Z}$ với phép toán cộng, trừ, nhân và luật chia là một **trường**.

- Trường $\mathbb{Z}/p\mathbb{Z}$ chỉ có hữu hạn phần tử. Đây là trường hữu hạn và ta ký hiệu \mathbb{F}_p .
- Ta viết $(\mathbb{F}_p)^*$ cho nhóm $(\mathbb{Z}/p\mathbb{Z})^*$.
- Trong \mathbb{F}_p người ta thường ký hiệu

47 / 57

Ví dụ

Phân tích của 1728 là

$$1728 = 2^6 \cdot 3^3.$$

Vậy thì

$$\text{ord}_2(1726) = 6, \quad \text{ord}_3(1726) = 3,$$

và

$$\text{ord}_p(1728) = 0 \text{ với mọi số nguyên tố } p \geq 5.$$

Bài tập

Hãy chỉ ra thuật toán tính phần tử nghịch đảo a^{-1} của phần tử a trong nhóm $(\mathbb{Z}/p\mathbb{Z})^*$.

Ví dụ

$1^1 \equiv 1$	$1^2 \equiv 1$	$1^3 \equiv 1$	$1^4 \equiv 1$	$1^5 \equiv 1$	$1^6 \equiv 1$
$2^1 \equiv 2$	$2^2 \equiv 4$	$2^3 \equiv 1$	$2^4 \equiv 2$	$2^5 \equiv 4$	$2^6 \equiv 1$
$3^1 \equiv 3$	$3^2 \equiv 2$	$3^3 \equiv 6$	$3^4 \equiv 4$	$3^5 \equiv 5$	$3^6 \equiv 1$
$4^1 \equiv 4$	$4^2 \equiv 2$	$4^3 \equiv 1$	$4^4 \equiv 4$	$4^5 \equiv 2$	$4^6 \equiv 1$
$5^1 \equiv 5$	$5^2 \equiv 4$	$5^3 \equiv 6$	$5^4 \equiv 2$	$5^5 \equiv 3$	$5^6 \equiv 1$
$6^1 \equiv 6$	$6^2 \equiv 1$	$6^3 \equiv 6$	$6^4 \equiv 1$	$6^5 \equiv 6$	$6^6 \equiv 1$

Bảng: Các lũy thừa theo modun 7

Câu hỏi

Tại sao cột bên tay phải toàn nhận giá trị 1?

49 / 57

Nội dung

Ví dụ

Số $p = 15485863$ là số nguyên tố, vậy thì

$$2^{15485862} \equiv 1 \pmod{15485863}.$$

Vậy thì, không cần tính toán ta cũng biết rằng số $2^{15485862} - 1$ là bội số của 15485863.

51 / 57

1 Thuật toán Euclid

2 Số học đồng dư

3 Số nguyên tố và trường hữu hạn

Định lý (Định lý Fermat nhỏ)

Xét số nguyên tố p và xét số nguyên a . Khi đó

$$a^{p-1} \equiv \begin{cases} 1 & (\text{mod } p) \text{ nếu } p \nmid a, \\ 0 & (\text{mod } p) \text{ nếu } p \mid a. \end{cases}$$

4 Lũy thừa và căn nguyên thủy trong trường hữu hạn

Định nghĩa

Cấp của phần tử a theo modun p là số mũ $k > 0$ nhỏ nhất thỏa mãn

$$a^k \equiv 1 \pmod{p}.$$

Mệnh đề

Xét số nguyên tố p và xét số nguyên a không chia hết cho p . Giả sử $a^n \equiv 1 \pmod{p}$. Vậy thì n chia hết cho cấp của a theo modun p . Đặc biệt, $p - 1$ chia hết cho cấp của a .

$$\mathbb{F}_p^* = \{1, g, g^2, g^3, \dots, g^{p-2}\}.$$

Các phần tử có tính chất này được gọi là **căn nguyên thủy** của \mathbb{F}_p hoặc **phân tử sinh** của \mathbb{F}_p^* . Chúng là các phân tử của \mathbb{F}_p^* có cấp $p - 1$.

53 / 57

Nhận xét

Định lý Fermat nhỏ và thuật toán tính nhanh lũy thừa cho ta một phương pháp hợp lý để tính nghịch đảo theo modun p . Cụ thể

$$a^{-1} \equiv a^{p-2} \pmod{p}.$$

Thời gian tính toán của phương pháp này tương tự như dùng thuật toán Euclid mở rộng.

Bài tập

Hãy chứng minh mệnh đề trước.

Bài tập

- Hãy tìm một căn nguyên thủy của trường \mathbb{F}_{17} .
- Hãy liệt kê tất cả các căn nguyên thủy của \mathbb{F}_{17} .

57 / 57

Ví dụ

Trường \mathbb{F}_{11} có 2 là một căn nguyên thủy, bởi vì trong \mathbb{F}_{11} ,

$$\begin{array}{llllll} 2^0 & = 1 & 2^1 & = 2 & 2^2 & = 4 \\ 2^5 & = 10 & 2^6 & = 9 & 2^7 & = 7 \\ & & & & 2^8 & = 3 \\ & & & & 2^9 & = 6. \end{array}$$

nhưng 2 không phải căn nguyên thủy của \mathbb{F}_{17} , bởi vì trong \mathbb{F}_{17}

$$\begin{array}{llllll} 2^0 & = 1 & 2^1 & = 2 & 2^2 & = 4 \\ 2^5 & = 15 & 2^6 & = 13 & 2^7 & = 9 \\ & & & & 2^8 & = 1 \end{array}$$

Định nghĩa

- Ước chung của hai số nguyên a và b là số nguyên d thỏa mãn:

$$d \mid a \text{ và } d \mid b.$$

- Ta ký hiệu $\gcd(a, b)$ là ước chung **lớn nhất** của a và b .

1 Thuật toán Euclid**2 Thuật toán tính luỹ thừa**

- $\gcd(12, 18) = 6$ vì $6 \mid 12$ và $6 \mid 18$ và không có số nào lớn hơn có tính chất này.
- $\gcd(748, 2014) = 44$ vì các ước của $748 = \{1, 2, 4, 11, 17, 22, 34, 44, 68, 187, 374, 748\}$, các ước của $2024 = \{1, 2, 4, 8, 11, 22, 23, 44, 46, 88, 92, 184, 253, 506, 1012, 2024\}$.

Định nghĩa

- Ước chung của hai số nguyên a và b là số nguyên d thỏa mãn:

$$d \mid a \text{ và } d \mid b.$$

Nhắc lại một số thuật toán trong lý thuyết số

Trần Vĩnh Đức

HUST

Ngày 23 tháng 3 năm 2023

Định lý (Thuật toán Euclid)

Xét a, b là hai số nguyên dương với $a \geq b$. Thuật toán sau đây tính $\gcd(a, b)$ sau một số hữu hạn bước.

- 1 Đặt $r_0 = a$ và $r_1 = b$.
- 2 Đặt $i = 1$.
- 3 Chia r_{i-1} cho r_i , ta được

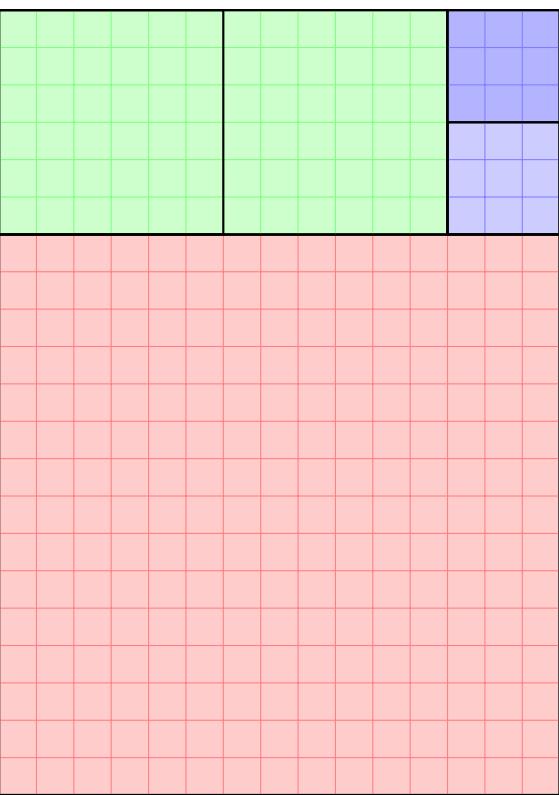
$$r_{i-1} = r_i \cdot q_i + r_{i+1} \quad \text{với } 0 \leq r_{i+1} < r_i.$$

- 4 Nếu $r_{i+1} = 0$, vậy thì

$$r_i = \gcd(a, b)$$

và thuật toán kết thúc.

- 5 Ngược lại, $r_{i+1} > 0$, vậy thì đặt $i = i + 1$ và quay lại Bước 3.



Định lý

Phép chia (Bước 3) của Thuật toán Euclid thực hiện nhiều nhất

$$\log_2(b) + 2 \quad \text{lần.}$$

Thuật toán Euclid mở rộng

- **Input :** Cặp số nguyên dương (a, b)
- **Output:** Bộ ba (d, x, y) thỏa mãn

$$d = \gcd(a, b) = ax + by.$$

EXTENDED-EUCLID(a, b)

```

if  $b == 0$ 
    return  $(a, 1, 0)$ 
else
     $(d', x', y') = \text{EXTENDED-EUCLID}(b, a \bmod b)$ 
     $(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$ 
    return  $(d, x, y)$ 

```

9 / 22

Ví dụ

$$\begin{array}{rccccccc} a & b & \lfloor a/b \rfloor & d & x & & y \\ 99 & 78 & 1 & & & & \end{array}$$

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned} x &= y' \\ y &= x' - \lfloor a/b \rfloor y' \end{aligned}$$

11 / 22

Nhắc lại một số thuật toán trong lý thuyết số | Thuật toán Euclid

Thuật toán Euclid mở rộng

Nhắc lại một số thuật toán trong lý thuyết số | Thuật toán Euclid

Tính đúng đắn của thuật toán

- Thuật toán tìm (d, x, y) thỏa mãn

$$d = \gcd(a, b) = ax + by$$

- Thuật toán Euclid có thể mở rộng để tìm thêm một số thông tin.
- Cụ thể, chúng ta mở rộng thuật toán để tính thêm hệ số x, y thỏa mãn

$$d = \gcd(a, b) = ax + by.$$

- Các hệ số x, y có thể âm hoặc bằng 0. Các hệ số này sẽ có ích sau này khi tích phân tử nghịch đảo trong số học modun.
- Vâng vậy thì

$$\begin{aligned} d' &= d = \gcd(b, a \bmod b) \\ &= b\lambda' + (a - b\lfloor a/b \rfloor)y' \end{aligned}$$

$$\begin{aligned} d &= b'\lambda' + (a - b\lfloor a/b \rfloor)y' \\ &= ay' + b(\lambda' - \lfloor a/b \rfloor)y' \end{aligned}$$

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			
3	0	—	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			
3	0	—	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			
3	0	—	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3			
21	15	1			
15	6	2			
6	3	2			
3	0	—	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1			
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Bài tập

Hãy tính giá trị

$$(d, x, y) = \text{EXTENDED-EUCLID}(899, 493).$$

12 / 22

Ví dụ

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1	3	-11	14
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0	-	3	1	0

- Mỗi dòng của bảng mô tả một mức độ quy: các giá trị đầu vào a và b , giá trị tính $\lfloor a/b \rfloor$, và giá trị trả về d, x, y .
- Bộ ba d, x, y được trả về trở thành bộ ba d', x', y' của mức tiếp theo từ công thức

$$\begin{aligned}x &= y' \\y &= x' - \lfloor a/b \rfloor y\end{aligned}$$

11 / 22

Ví dụ (tiếp)

Ta lập bảng

i	0	1	2	3	4	5	6	7
$3^{2^j} \pmod{1000}$	3	9	81	561	721	841	281	961

rồi tính

$$\begin{aligned} 3^{218} &= 3^2 \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^6} \cdot 3^{2^7} \\ &\equiv 9 \cdot 561 \cdot 721 \cdot 281 \cdot 961 \pmod{1000} \\ &\equiv 489 \pmod{1000}. \end{aligned}$$

Tính nghịch đảo

Nhắc lại một số thuật toán trong lý thuyết số | Thuật toán Euclid

Ví dụ (tiếp)

Ta lập bảng

i	0	1	2	3	4	5	6	7
$3^{2^j} \pmod{1000}$	3	9	81	561	721	841	281	961

rồi tính

$$\begin{aligned} 3^{218} &= 3^2 \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^6} \cdot 3^{2^7} \\ &\equiv 9 \cdot 561 \cdot 721 \cdot 281 \cdot 961 \pmod{1000} \\ &\equiv 489 \pmod{1000}. \end{aligned}$$

Tính lũy thừa nhanh

Nhắc lại một số thuật toán trong lý thuyết số | Thuật toán tính lũy thừa

Ví dụ

Giả sử ta muốn tính

$$3^{218} \pmod{1000}.$$

Đầu tiên, ta viết 218 ở dạng cơ số 2:

$$218 = 2 + 2^3 + 2^4 + 2^6 + 2^7.$$

Vậy thì 3^{218} trở thành

$$3^{218} = 3^{2+2^3+2^4+2^6+2^7} = 3^2 \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^6} \cdot 3^{2^7}.$$

Để ý rằng, để tính các mũ

$$3, 3^2, 3^{2^2}, 3^{2^3}, 3^{2^4}, \dots$$

Ví dụ

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
d	7	49	157	526	160	241	298	166	67	1

■ Kết quả tính $a^b \pmod{n}$ với

$$a = 7, \quad b = 560 = \langle 1000110000 \rangle, \quad \text{và } n = 561$$

- Giá trị được chỉ ra sau mỗi bước lặp.
- Kết quả cuối cùng bằng 1

18 / 22

Thuật toán tính nhanh $a^b \pmod{n}$

Nhắc lại một số thuật toán trong lý thuyết số | Thuật toán tính lũy thừa

MODULAR-EXPONENTIATION(a, b, n)

```

 $c = 0$ 
 $d = 1$ 
Biểu diễn  $b = \langle b_k, b_{k-1}, \dots, b_0 \rangle_2$ 
for  $i = k$  downto 0
   $c = 2c$ 
   $d = (d \cdot d) \pmod{n}$ 
  if  $b_i == 1$ 
     $c = c + 1$ 
   $d = (d \cdot a) \pmod{n}$ 
return  $d$ 

```

Định lý (Định lý Fermat nhỏ)

Xét số nguyên tố p và xét số nguyên a . Khi đó

$$a^{p-1} \equiv \begin{cases} 1 & (\text{mod } p) \text{ nếu } p \nmid a, \\ 0 & (\text{mod } p) \text{ nếu } p \mid a. \end{cases}$$

20 / 22

Thuật toán đệ quy tính $a^b \pmod{n}$

Nhắc lại một số thuật toán trong lý thuyết số | Thuật toán tính lũy thừa

```

MODULAR-EXPONENTIATION( $a, b, n$ )
  if  $b == 0$  then return 1
  if  $b == 1$  then return  $a$ 
   $r = \text{MODULAR-EXPONENTIATION}(a, b/2, n)$ 
   $r = r * r$ 
  if  $b \bmod 2 == 1$  then  $r = r * a$ 
return  $r$ 

```

Nhận xét

Định lý Fermat nhỏ và thuật toán tính nhanh lũy thừa cho ta một phương pháp hợp lý để tính nghịch đảo theo modun p . Cụ thể

$$a^{-1} \equiv a^{p-2} \pmod{p}.$$

Thời gian tính toán của phương pháp này tương tự như dùng thuật toán Euclid mở rộng.

Ví dụ

Số $p = 15485863$ là số nguyên tố, vậy thì

$$2^{15485862} \equiv 1 \pmod{15485863}.$$

Vậy thì, không cần tính toán ta cũng biết rằng số $2^{15485862} - 1$ là bội số của 15485863 .