TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ

AN TOÀN CÁC HỆ THỐNG THÔNG TIN

ThS. Nguyễn Văn Kha

dzokha1010@gmail.com

Chương 3
KỸ THUẬT MẬT MÃ

3.1. Khái niệm mật mã

3.2. Các phương pháp mã hoá cổ điển

3.3. Mã hoá đối xứng hiện đại

3.4. Mã hoá bất đối xứng hiện đại

3.5. So sánh mã hoá đối xứng và mã hoá bất đối xứng

3.1. Khái niệm mật mã
3.1.1. Mật mã học

Mật mã học (Cryptology) là môn khoa học về mã hoá, bao gồm mật mã (Cryptography) và phân tích mật mã (Cryptanalysis).

Mật mã xuất phát từ từ Hy Lạp “Kryptos” có nghĩa là “ẩn giấu” và "Graphein” có nghĩa là "viết," và liên quan đến việc tạo ra và sử dụng mã để bảo mật thông điệp.

Phân tích mật mã liên quan đến việc bẻ khóa hoặc phá vỡ các thông điệp đã được mã hóa để đưa chúng trở về dạng ban đầu chưa được mã hóa.

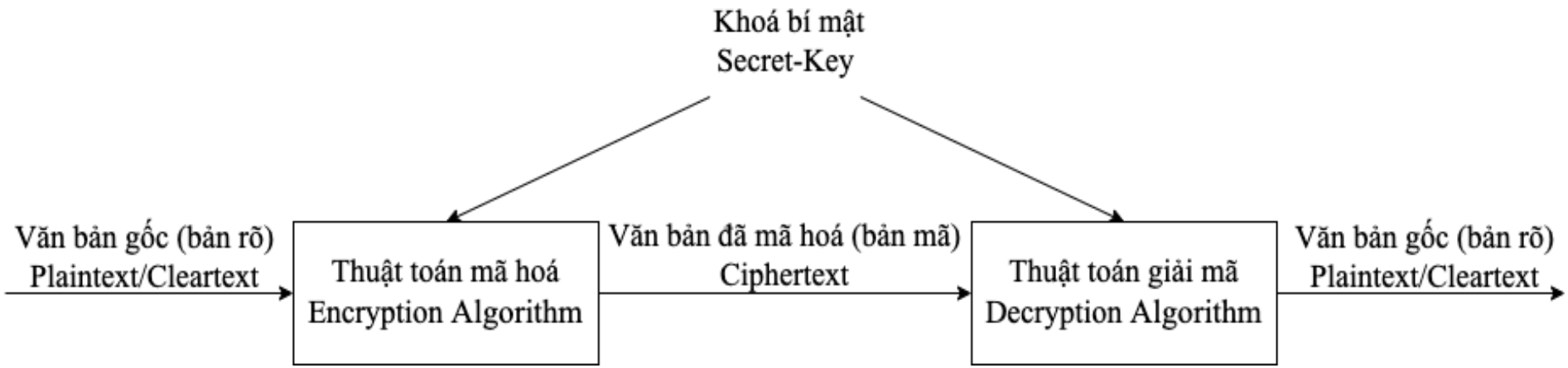
Mật mã sử dụng các thuật toán toán học mà mọi người thường biết. Mật mã không phải là kiến thức về thuật toán bảo vệ thông điệp mã hóa, mà là kiến thức về khóa, một chuỗi các ký tự hoặc bit được thêm vào thuật toán cùng với thông điệp gốc để tạo ra thông điệp mã hóa.

3.1. Khái niệm mật mã
3.1.2. Mật mã khoá bí mật

Mật mã khóa bí mật (**Secret-Key Cryptography**) là một mô hình mật mã trong đó cả quá trình mã hoá và giải mã đều sử dụng cùng một Khoá bí mật (Secret-Key), Khoá là một 'bí mật' mà những bên tham gia liên lạc mới biết.

Khóa được sử dụng để chuyển đổi thông điệp gốc (được gọi là bản rõ) thành dạng mã hóa mà bất kỳ ai không có Khóa đều không thể hiểu thông điệp. Quá trình này được gọi là mã hóa và thông điệp đã mã hóa (được gọi là bản mã). Khoá này cũng được sử dụng để chuyển đổi bản mã trở lại bản rõ; đây là quá trình giải mã. Trong mô hình này, các hàm mã hóa và giải mã là nghịch đảo của nhau.

3.1. Khái niệm mật mã
3.1.2. Mật mã khoá bí mật (tt)



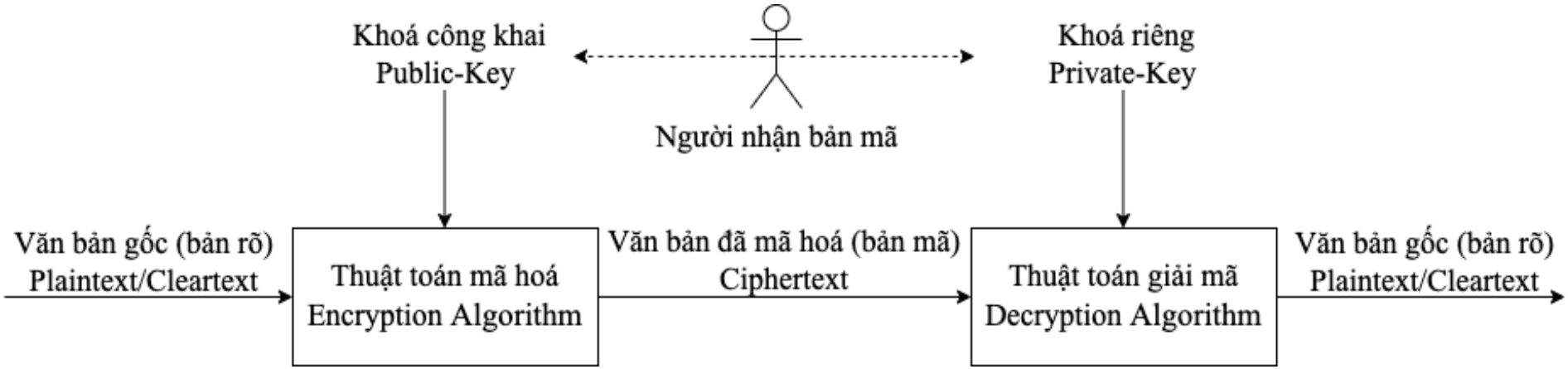
Sơ đồ mật mã khoá bí mật

3.1. Khái niệm mật mã
3.1.3. Mật mã khoá công khai

Mật mã khoá công khai (**Public-Key Cryptography**) là một mô hình mật mã trong đó quá trình mã hoá và giải mã phải sử dụng hai khoá riêng biệt. Khóa công khai sẽ được sử dụng để mã hóa bản rõ và khóa riêng sẽ cho phép giải mã bản mã.

Lưu ý rằng khóa công khai có thể được "mọi người" biết, trong khi khóa riêng chỉ được một người biết. Vì vậy, hệ thống mật mã khóa công khai sẽ cho phép bất kỳ ai mã hóa thông điệp với khoa công khai của người nhận và chỉ người nhận mới có khoá riêng để giải mã thông điệp đó.

3.1. Khái niệm mật mã
3.1.3. Mật mã khoá công khai (tt)



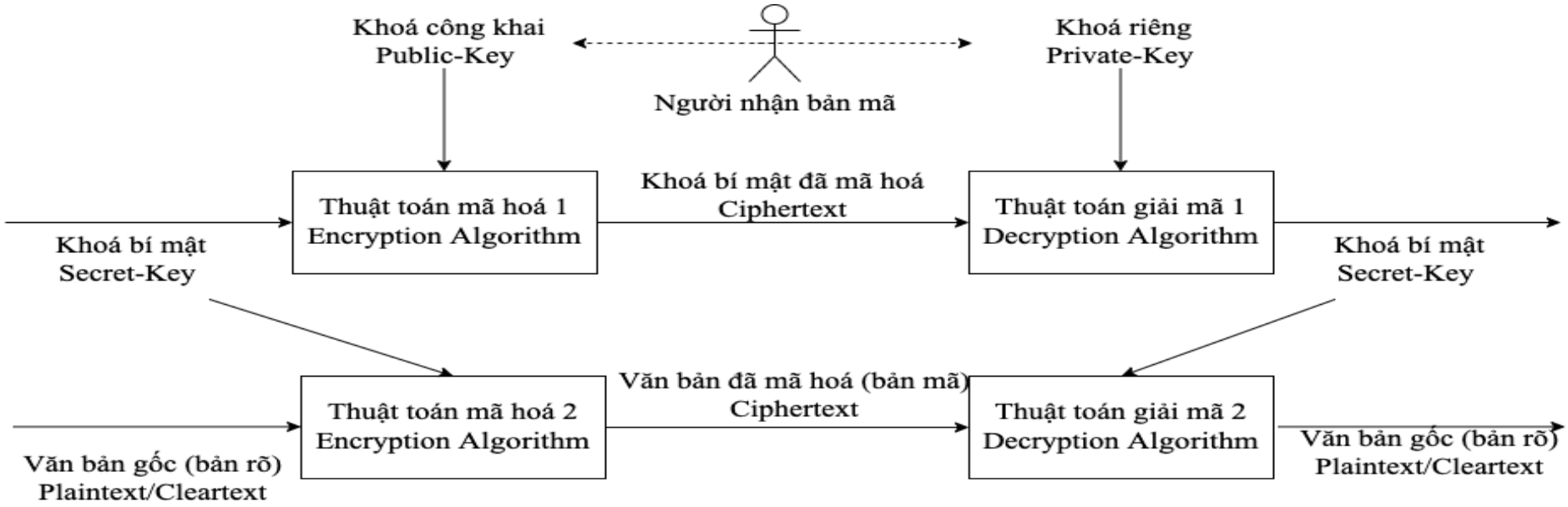
Sơ đồ mật mã khoá công khai

3.1. Khái niệm mật mã
3.1.4. Mật mã lai

Một trong những nhược điểm của hệ thống mật mã khóa công khai là chúng chậm hơn nhiều so với hệ thống mật mã khóa bí mật. Do đó, hệ thống mật mã khóa công khai chủ yếu được sử dụng để mã hóa lượng dữ liệu nhỏ. Tuy nhiên, có một cách hay để kết hợp mật mã khóa bí mật và công khai để đạt được lợi ích của cả hai. Kỹ thuật này được gọi là mật mã lai (**Hybrid Cryptography**).

Giả sử NG muốn mã hóa một thông điệp dài và gửi cho NN và cả hai chưa có khoá bí mật chung. NG có thể tạo ra một khóa bí mật ngẫu nhiên và mã hóa bản rõ, bằng cách sử dụng hệ thống mật mã khóa bí mật. Sau đó, NG mã hóa khóa bí mật này bằng khóa công khai của NN. NG gửi bản mã và khóa đã mã hóa cho NN. Đầu tiên, NN sử dụng khóa riêng của mình để giải mã khóa bí mật, sau đó anh ấy sử dụng khóa bí mật này để giải mã bản mã.

3.1. Khái niệm mật mã
3.1.4. Mật mã lai (tt)



Sơ đồ mật mã lai

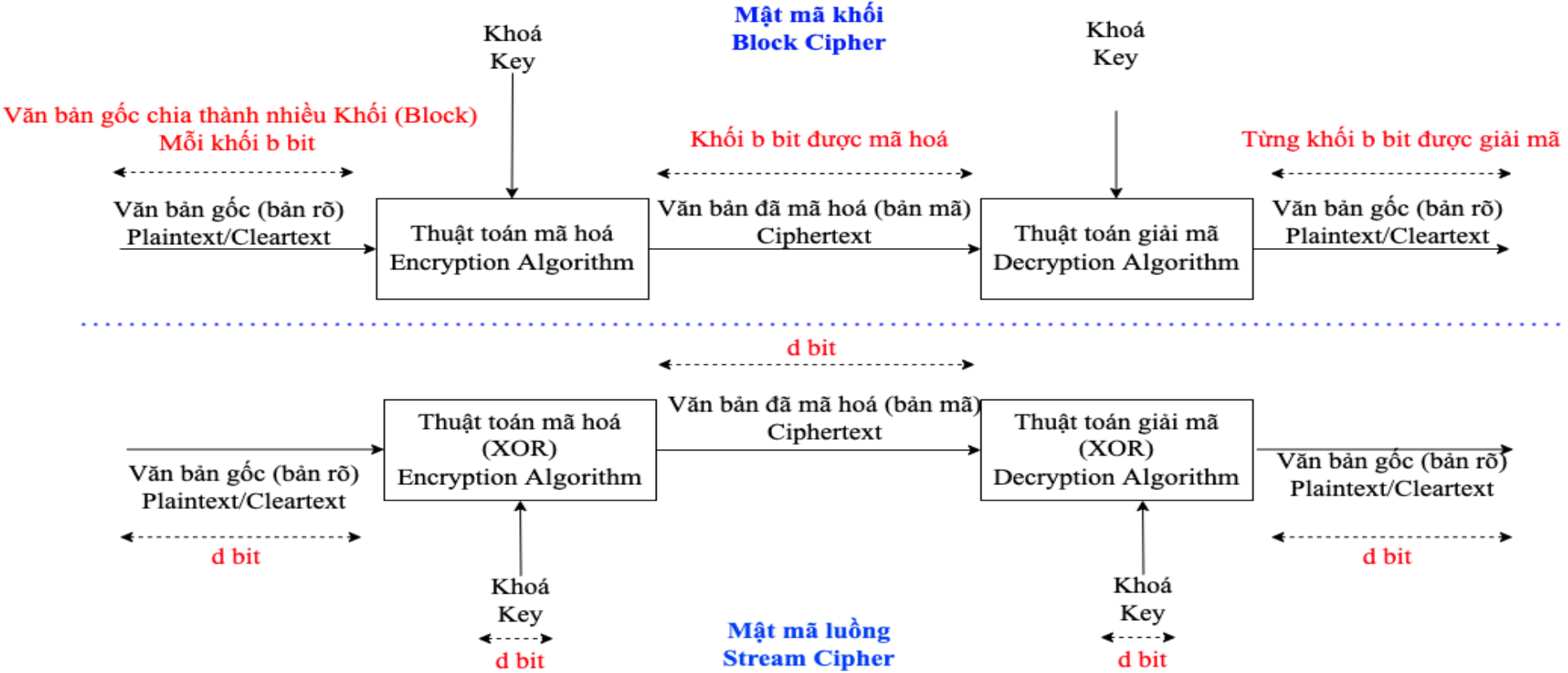
3.1. Khái niệm mật mã
3.1.5. Mật mã khối và Mật mã luồng

Mật mã khối **(Block Cipher)**, bản rõ được chia thành các khối có kích thước cố định gọi là Khối. Một Khối được chỉ định là một chuỗi bit có độ dài cố định (VD: 64 or 128 bit). Mật mã khối mã hóa (hoặc giải mã) từng Khối một.

Ngược lại, Mật mã luồng (**Stream Cipher**) trước tiên sử dụng khóa để xây dựng luồng khóa, là một chuỗi bit có độ dài chính xác bằng bản rõ (bản rõ là một chuỗi bit có độ dài tùy ý). Hoạt động mã hóa xây dựng bản mã dưới dạng phép toán XOR (Exclusive OR) của bản rõ và luồng khóa. Giải mã được thực hiện bằng cách tính toán phép toán XOR của bản mã và luồng khóa.

Hệ thống mật mã khóa công khai luôn là Mật mã khối, trong khi hệ thống mật mã khóa bí mật có thể là Mật mã khối hoặc Mật mã luồng.

3.1. Khái niệm mật mã
3.1.5. Mật mã khối và Mật mã luồng (tt)



Mô hình Mật mã khối và Mật mã luồng

3.2. Các phương pháp mã hoá cổ điển
3.2.1. Mật mã thay thế

Mật mã thay thế (Substitution Cipher) trao đổi một giá trị cho một giá trị khác. Đây là một phương pháp khá đơn giản, nhưng nó trở nên rất mạnh mẽ nếu kết hợp với các hoạt động khác. Phép thay thế dựa trên một bảng chữ cái duy nhất và do đó được gọi là phép thay thế đơn bảng chữ cái. Các mật mã thay thế tiên tiến hơn sử dụng hai hoặc nhiều bảng chữ cái và được gọi là phép thay thế đa bảng chữ cái.

- Mật mã Caesar
- Mật mã Vigenère

3.2. Các phương pháp mã hoá cổ điển
3.2.1. Mật mã thay thế
a) Mật mã Caesar

Mật mã Caesar, là mật mã thay thế đơn chữ cái được đặt theo tên của ông Julius Caesar: sử dụng phép dịch chuyển K vị trí sang phải để mã hóa các thông điệp

Ví dụ: K = 3

Bản rõ: TEXT

Bản mã: WHAW

3.2. Các phương pháp mã hoá cổ điển
3.2.1. Mật mã thay thế
b) Mật mã Vigenère

Mật mã Vigenère: sử dụng hình vuông Vigenère, gồm 26 chữ cái mật mã riêng biệt.

Ví dụ: Bản rõ: SECURITY

Bản mã: TGFYWOAG

Ví dụ, với từ khóa là ITALY

Bản rõ: SACK GAUL SPARE NO ONE

I	T	A	L	Y	I	T	A	L	Y	I	T	A	L	Y	I	T	A
S	A	C	K	G	A	U	L	S	P	A	R	E	N	O	O	N	E

Bản mã: ATCVEINLDNIKEYMWGE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

3.2. Các phương pháp mã hoá cổ điển
3.2.2. Mật mã hoán vị

Mật mã hoán vị (Transposition Cipher), giống như phương pháp thay thế, mật mã hoán vị chỉ đơn giản là sắp xếp lại các bit hoặc byte (ký tự) trong một khối để tạo ra bản mã.

Mẫu khoá: 8 → 3, 7 → 6, 6 → 2, 5 → 7, 4 → 5, 3 → 1, 2 → 8, 1 → 4

Bản rõ: 00100101011101011001010101010100

Vị trí bit: 87654321 87654321 87654321 87654321

Khối 8 bit dạng bản rõ: 00100101 | 01101011 | 10010101 | 01010100

Bản mã: 00001011 | 10111010 | 01001101 | 01100001

3.2. Các phương pháp mã hoá cổ điển
3.2.2. Mật mã hoán vị (tt)

Mẫu khoá: 8 → 3, 7 → 6, 6 → 2, 5 → 7, 4 → 5, 3 → 1, 2 → 8, 1 → 4

Bản rõ: SACK_GAUL_SPARE_NO_ONE

Vị trí chữ cái:	87654321 | 87654321 | 87654321
Bản rõ (khối 8 bit):	__ENO_ON | _ERAPS_L | UAG_KCAS
Bản mã:	ON_ON_E_ | _AEPL_RS | A_AKSUGC

3.2. Các phương pháp mã hoá cổ điển
3.2.3. Exclusive OR

Phép toán Exclusive OR được ký hiệu là XOR, là một hàm của đại số Boolean trong đó hai bit được so sánh và tạo ra kết quả nhị phân.

Ví dụ:

Bản rõ: CAT; Khoá K = V

Bản rõ nhị phân: 01000011 01000001 01010100

V lập 3 lần nhị phân: 01010110 01010110 01010110

Bản mã: 00010101 00010111 00000010

Bit 1	Bit 2	Kết quả
0	0	0
0	1	1
1	0	1
1	1	0

3.2. Các phương pháp mã hoá cổ điển
3.2.4. Mật mã Vernam

Mật mã Vernam (Vernam Cipher), sử dụng một tập hợp các ký tự ngẫu nhiên để mã hoá dữ liệu, tập ký tự này chỉ được dùng một lần duy nhất (còn gọi là One-Time Pad). Pad trong tên bắt nguồn từ thời kỳ mã hóa và giải mã thủ công khi các giá trị khóa được đóng trên một tờ giấy (Pad Paper).

Để thực hiện mã hóa mật mã Vernam, quá trình diễn ra như sau:

- Cộng giá trị của Pad và giá trị bản rõ: các ký tự bản rõ được chuyển đổi thành các giá trị số (theo bảng chữ cái, ví dụ: A = 1, B = 2,..., Z = 26). Sau đó, các giá trị này được cộng với giá trị số tương ứng từ khóa (Pad).

- Modulo 26: Nếu tổng của hai giá trị vượt quá 26, 26 sẽ được trừ khỏi tổng cho đến khi kết quả nằm trong phạm vi từ 1 đến 26, đảm bảo kết quả luôn nằm trong giá trị của bảng chữ cái.

- Chuyển đổi ngược lại thành chữ: Tổng giá trị số sẽ được chuyển ngược lại thành ký tự.

3.2. Các phương pháp mã hoá cổ điển
3.2.4. Mật mã Vernam (tt)

Bản rõ:	S	A	C	K	G	A	U	L	S	P	A	R	E	N	O	O	N	E
Giá trị bản rõ:	19	01	03	11	07	01	21	12	19	16	01	18	05	14	15	15	14	05
One-Time Pad:	F	P	Q	R	N	S	B	I	E	H	T	Z	L	A	C	D	G	J
Giá trị Pad:	06	16	17	18	14	19	02	09	05	08	20	26	12	01	03	04	07	10
Tổng bản rõ & Pad:	25	17	20	29	21	20	23	21	24	24	21	44	17	15	18	19	21	15
Sau khi trừ Modulo:				03								18						
Bản mã:	Y	Q	T	C	U	T	W	U	X	X	U	R	Q	O	R	S	U	O

3.2. Các phương pháp mã hoá cổ điển
3.2.5. Mật mã dựa trên cuốn sách
a) Mật mã sách

Mật mã sách (Book Cipher), bản mã bao gồm danh sách các mã biểu diễn số trang, số dòng và số từ của bản rõ và khoá là cuốn sách.

Ví dụ:

Khoá là cuốn sách "*Các hình thức tấn công mạng - Cyberspace*"

Thông điệp mã hoá có dạng: 89,19,8; 22,3,8…

3.2. Các phương pháp mã hoá cổ điển
3.2.5. Mật mã dựa trên cuốn sách
b) Mật mã khoá chạy

Mật mã khoá chạy (Running Key Cipher) tương tự như mật mã Vigenère, sử dụng một cuốn sách để truyền khóa cho một mật mã. Người gửi cung cấp một thông điệp được mã hóa với một chuỗi số ngắn chỉ ra số trang, dòng và từ trong một cuốn sách được xác định trước để sử dụng làm khóa hoặc khối chỉ báo.

Không giống như mật mã Vigenère, nếu khóa cần được mở rộng trong mật mã Running Key Cipher, bạn không lặp lại khóa. Thay vào đó, bạn tiếp tục sử dụng văn bản từ khối chỉ báo. Các bước thực hiện mã hoá cùng một phương pháp cơ bản như mật mã Vigenère.

3.2. Các phương pháp mã hoá cổ điển
3.2.5. Mật mã dựa trên cuốn sách
c) Mật mã mẫu

Mật mã mẫu (Template Cipher) hoặc mật mã trang đục lỗ, liên quan đến việc sử dụng một thông điệp ẩn trong một cuốn sách, lá thư hoặc thông điệp khác. Người nhận phải sử dụng một trang có số lỗ cụ thể được cắt sẵn và đặt lên trang sách hoặc lá thư để trích xuất thông điệp ẩn.

3.2. Các phương pháp mã hoá cổ điển
3.2.6. Hàm băm

Hàm băm (Hash Function) là thuật toán toán học được sử dụng để xác nhận tính toàn vẹn của thông điệp. Thay vì mã hóa dữ liệu như các phương pháp khác, hàm băm tạo ra một giá trị băm, còn gọi là chín chắc (Digest) hay chữ ký thông điệp, bằng cách chuyển đổi thông điệp có độ dài bất kỳ thành một giá trị có độ dài cố định duy nhất. Giá trị băm này giúp nhận diện và kiểm tra thông điệp có bị thay đổi không.

Các hàm băm hoạt động một chiều, nghĩa là từ cùng một thông điệp sẽ luôn cho ra cùng một giá trị băm. Nếu giá trị băm của hai thông điệp giống nhau, tức là thông điệp không bị thay đổi. Nếu giá trị băm khác nhau, thông điệp đã thay đổi.

Các thuật toán băm: SHA-1 có chín chắc 160 bit;

Tương tự: SHA-256, SHA-384, SHA-512

3.3. Mã hoá đối xứng

Mã hoá đối xứng là các phương pháp mã hóa yêu cầu cùng một khóa bí mật để mã hóa và giải mã thông điệp, còn được gọi là mật mã khóa bí mật. Các phương pháp mã hóa đối xứng sử dụng các phép toán, thuật toán tính toán nhanh để mã hóa và giải mã được thực hiện nhanh chóng, ngay cả bằng các máy tính nhỏ.

Một trong những thách thức là cả người gửi và người nhận đều phải có khóa bí mật. Nếu khóa bị rơi vào tay kẻ xấu, thông điệp có thể bị giải mã mà người gửi và người nhận không hề hay biết. Do đó, việc truyền khóa phải được thực hiện qua một kênh khác để tránh bị chặn, tức là không sử dụng cùng một kênh truyền tải với thông điệp mã hóa.

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES

Tiêu chuẩn DES (Data Encryption Standard) là một phương pháp mã hoá đối xứng được phát triển bởi NIST vào năm 1977. DES là một trường hợp của mật mã Feistel với l = 32, mật mã Feistel chia bản rõ thành một chuỗi các khối, mỗi khối dài 2l bit, Feistel sử dụng các phép toán cơ bản là XOR và phép thế trong mật mã.

DES là thuật toán mật mã khối, trong đó bản rõ được chia thành các khối 64 bit và sử dụng Khoá có độ dài 56 bit. Tuy nhiên, Khoá DES được biểu diễn dưới dạng chuỗi nhị phân 64 bit, trong đó bit thứ 8 (i = 1, 2, ⋯ , 8) là bit chẵn lẻ của bảy bit ngay trước nó. Bit chẵn lẻ được sử dụng để phát hiện lỗi. Giá trị của Khoá là chuỗi 48 bit.

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

Với M là bản rõ 64 bit và K là khoá 56 bit

Các bước mã hoá DES

B1: $IP(M) = L_0R_0$; $|L_0|=|R_0|=32$

B2: Với i =1,2…,16, tính:

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1},K_i)$

B3: $C = IP^{-1}(R_{16}L_{16})$

Các bước giải mã DES

B1: $IP(C) = L'_0R'_0$; $|L'_0|=|R'_0|=32$

B2: Với i =1,2…,16, tính:

$L'_i = R'_{i-1}$

$R'_i = L'_{i-1} \oplus F(R'_{i-1},K_{17-i})$

B3: $M = IP^{-1}(R'_{16}L'_{16})$

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

Sinh 16 khoá:

$X = x_1x_2...x_{56}$; $x_i = \{0,1\}$

Chuỗi Y 28 bit. $Ls_{z(i)}(Y)$ dịch chuyển Y theo vòng tròn sang trái z(i) lần

Viết $IP_{key}(K)$ thành U_0V_0, $|U_0|=|V_0|=28$ bit

Tìm khoá K_i, Với i = 1, 2, ⋯ , 16

$$P_{key}(X) = \begin{matrix} x_{14} & x_{17} & x_{11} & x_{24} & x_1 & x_5 & x_3 & x_{28} & x_{15} & x_6 & x_{21} & x_{10} \\ x_{23} & x_{19} & x_{12} & x_4 & x_{26} & x_8 & x_{16} & x_7 & x_{27} & x_{20} & x_{13} & x_2 \\ x_{41} & x_{52} & x_{31} & x_{37} & x_{47} & x_{55} & x_{30} & x_{40} & x_{51} & x_{45} & x_{33} & x_{48} \\ x_{44} & x_{49} & x_{39} & x_{56} & x_{34} & x_{53} & x_{46} & x_{42} & x_{50} & x_{36} & x_{29} & x_{32} \end{matrix}$$

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$z(i)$	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

$$U_i = LS_{z(i)}(U_{i-1}),$$
$$V_i = LS_{z(i)}(V_{i-1}),$$
$$K_i = P_{key}(U_iV_i).$$

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

Ví dụ:

U_0 = 10011010011101100010100110 10,

V_0 = 01100101100010011101011001 01.

$U_1 = LS_{z(1)}(U_0)$

= 00110100111011000101001101 10,

$V_1 = LS_{z(1)}(V_0)$

= 11001011000100111010110010 10.

$K_1 = P_{key}(U_1V_1)$

= Pkey(00110100111011000101001101011100101100010011101011001010)

= 10110011010110011000011000001111011011000100111 0.

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

Hoán vị ban đầu (IP)

Bản rõ: $M = m_1 m_2 \cdots m_{64}$. Mỗi số i biểu diễn bit m_i

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

- Chỉ số của hai hàng đầu tiên trong IP(M) bắt đầu từ 58, trong đó chỉ số tiếp theo bằng chỉ số hiện tại trừ 8 mod 66.
- Chỉ số của hai hàng cuối cùng trong IP(M) bắt đầu từ 57, trong đó chỉ số tiếp theo bằng chỉ số hiện tại trừ 8 mod 66.

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

Hàm thay thế F

Cho V = $v_1v_2 \cdots v_{32}$ là chuỗi nhị phân 32 bit.

$$P(V) = \begin{matrix} v_{16} & v_7 & v_{20} & v_{21} & v_{29} & v_{12} & v_{28} & v_{17} & v_1 & v_{15} & v_{23} & v_{26} & v_5 & v_{18} & v_{31} & v_{10} \\ v_2 & v_8 & v_{24} & v_{14} & v_{32} & v_{27} & v_3 & v_9 & v_{19} & v_{13} & v_{30} & v_6 & v_{22} & v_{11} & v_4 & v_{25} \end{matrix}$$

$$F(R_{i-1}, K_i) = P(S(EP(R_{i-1}) \oplus K_i)), i = 1, 2, \cdots, 16.$$

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

S-Box

Có 8 S-Box, mỗi S-Box là 1 ma trận 4 x 16,

mỗi hàng trong S-Box là một hoán vị của các số từ 0 đến 15.

Chúng ta dán nhãn các S-Box này là S1, S2, ⋯ , S8

Cho Y = $y_1y_2 \cdots y_{48}$; Y [i, j] (i<j) để biểu thị chuỗi con yi⋯ yj .
Chia Y thành tám khối 6 bit:

Y [6r-5,6r]=$b_1b_2b_3b_4b_5b_6$; với r = 1,2,...,8; i = b_1b_6; j = $b_2b_3b_4b_5$

$$S_r(Y\,[6r-5, 6r]) = s^{(r)}{}_{ij}\ .$$

Giá trị trong S-Box r tại hàng i + 1 cột j + 1

Ví dụ:

Nếu Y [7, 12] = 110010, then $S_2(110010) = s^{(2)}{}_{10,1001} = s^{(2)}{}_{2,9} = 8$.

$$S(Y) = S_1(Y[1,6])S_2(Y[7,12])...S_8(Y[43,48])$$

S_1 :	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2 :	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3 :	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4 :	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5 :	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6 :	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7 :	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8 :	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

Hoán vị mở rộng (EP)

EP (Expansion Permutation)

$U = u_1u_2...u_{32}$; với $u_i \in \{0, 1\}$

Các chỉ số của bốn cột ở giữa là 1, 2, ⋯ , 32; các chỉ số của cột đầu tiên bắt đầu từ 32, trong đó chỉ số tiếp theo là chỉ số hiện tại cộng với 4 mod 32; và các chỉ số của cột cuối cùng bắt đầu từ 5, trong đó chỉ số tiếp theo là chỉ số hiện tại cộng với 4 mod 32.

$$EP(U) = \begin{matrix} u_{32} & u_1 & u_2 & u_3 & u_4 & u_5 \\ u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ u_8 & u_9 & u_{10} & u_{11} & u_{12} & u_{13} \\ u_{12} & u_{13} & u_{14} & u_{15} & u_{16} & u_{17} \\ u_{16} & u_{17} & u_{18} & u_{19} & u_{20} & u_{21} \\ u_{20} & u_{21} & u_{22} & u_{23} & u_{24} & u_{25} \\ u_{24} & u_{25} & u_{26} & u_{27} & u_{28} & u_{29} \\ u_{28} & u_{29} & u_{30} & u_{31} & u_{32} & u_1 \end{matrix}$$

3.3. Mã hoá đối xứng
3.3.1. Mã hoá DES (tt)

Hoán vị nghịch đảo (IP^{-1})

Giả sử C = c1c2 ⋯ c64 (ci ∈ {0, 1}). Khi đó IP^{-1} (C) = σ(C)

IP ∘ IP^{-1} (M) = IP^{-1} ∘ IP(M) = M.

Ví dụ, hãy để C = IP(M). Vì IP thay đổi m_1 thành m_{58} và IP^{-1} thay đổi c_1 thành c_{40}, trong đó $c_1 = m_{58}$ và $c_{40} = m_1$, chúng ta biết rằng IP^{-1} ∘ IP thay đổi m_1 trở lại m_1.

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

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES

Tiêu chuẩn AES (Advanced Encryption Standard) được NIST công bố vào năm 2001. AES là một mật mã khối, đơn vị tính toán cơ bản trong AES là byte, thay vì bit như trong DES.

AES chia bản rõ thành các khối 128 bit và hỗ trợ độ dài khoá là 128 bit, 192 bit và 256 bit. Bất kể độ dài khoá nào được sử dụng, AES sẻ tạo và sử dụng khoá con 16 byte, còn được gọi là khoá vòng.

AES là một thuật toán mã hoá lặp với số lần lặp tuỳ theo độ dài khoá, 128 bit, 192 bit và 256 bit với số lần lặp lần lược là 10, 12 và 14 lần.

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

Mã hoá AES-128

$A_1 = ark(A_0,K_0)$,

$A_{i+1} = ark(mic(shr(sub(A_i))),K_i)$,

$i = 1, 2, \cdots, 9$,

$A_{11} = ark(shr(sub(A_{10})),K_{10})$.

Giải mã AES-128

$C_1 = ark(C_0,K_{10})$,

$C_{i+1} = mic^{-1}(ark(sub^{-1}(shr^{-1}(C_i)),K_{10-i}))$,

$i = 1, 2, \cdots, 9$,

$C_{11} = ark(sub^{-1}(shr^{-1}(C_{10})),K_0)$.

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

Khoá vòng AES-128

4-Word của khoá K là K[32i, 32i+31];

i = 0, 1, 2, 3; $|K_i|$=32 bit

AES mở rộng K thành mảng 44-Word W[0, 43].

$$
\begin{aligned}
&W[0] = K[0, 31],\\
&W[1] = K[32, 63],\\
&W[2] = K[64, 95],\\
&W[3] = K[96, 127],\\
&W[i] = \begin{cases} W[i-4] \oplus T(W[i-1], i/4), & \text{if } i \text{ is divisible by 4,}\\ W[i-4] \oplus W[i-1], & \text{otherwise,} \end{cases}\\
&\quad i = 4, \cdots, 43.
\end{aligned}
$$

Với $w=w_1w_2w_3w_4$ với w_i là 1 byte và b_i là bit. Hàm thay thế T

$$T(w, j) = [(S(w_2) \oplus m(j-1)]S(w_3)S(w_4)S(w_1),$$

$$\mathcal{M}(b_7b_6b_5b_4b_3b_2b_1b_0) = \begin{cases} b_6b_5b_4b_3b_2b_1b_00, & \text{if } b_7 = 0,\\ b_6b_5b_4b_3b_2b_1b_00 \oplus 00011011, & \text{if } b_7 = 1, \end{cases}$$

$$m(j) = \begin{cases} 00000001, & \text{if } j = 0,\\ 00000010, & \text{if } j = 1,\\ \mathcal{M}(m(j-1)), & \text{if } j > 1. \end{cases}$$

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

S-Box của AES

AES chỉ sử dụng một S-Box. Nó được sử dụng để tạo các khóa con và xác định hoạt động của các byte thay thế. AES S-Box là một ma trận 16 × 16 byte, được xác định trên cơ sở phép nhân của trường Galois GF(28). Không giống như các S-Box được sử dụng trong DES, AES S-Box là một hoán vị của tất cả 256 byte.

$$S = [s_{ij}]_{16\times16}$$

$$S^{-1} = [s_{ij}]_{16\times16}$$

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

S-Box của AES

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

	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	5c	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	1e	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	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	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

S-Box ngược của AES

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

Cho $w = b_0 \cdots b_7$ là một byte, trong đó mỗi b_i là một bit. $i = b_0b_1b_2b_3$ biểu thị biểu diễn nhị phân của chỉ số hàng và $j = b_4b_5b_6b_7$ biểu thị biểu diễn nhị phân của chỉ số cột của sij trong S-Box. Sau đó

$$S(w) = s_{ij}\,,\ S^{-1}(w) = s'_{ij}\ .$$

Nghĩa là, S(w) là phần tử trên giao điểm của hàng thứ (i+1) và cột thứ (j+1) trong S-Box S. Tương tự như vậy, S−1(w) là phần tử trên giao điểm của hàng thứ (i+1) và cột thứ (j+1) trong nghịch đảo S-Box S−1.

Ví dụ, giả sử w = b8, thì $S(w) = s_{b,8} = 6c$, và $S^{-1}(6c) = s'_{6,c} = b8$.

$$S(S^{-1}(w)) = w \text{ and } S^{-1}(S(w)) = w.$$

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

Add Round Keys

$$ark(\boldsymbol{A}, \boldsymbol{K}_i) = \boldsymbol{A} \oplus \boldsymbol{K}_i = \begin{bmatrix} a_{0,0} \oplus k_{0,0} & a_{0,1} \oplus k_{0,1} & a_{0,2} \oplus k_{0,2} & a_{0,3} \oplus k_{0,3} \\ a_{1,0} \oplus k_{1,0} & a_{1,1} \oplus k_{1,1} & a_{1,2} \oplus k_{1,2} & a_{1,3} \oplus k_{1,3} \\ a_{2,0} \oplus k_{2,0} & a_{2,1} \oplus k_{2,1} & a_{2,2} \oplus k_{2,2} & a_{2,3} \oplus k_{2,3} \\ a_{3,0} \oplus k_{3,0} & a_{3,1} \oplus k_{3,1} & a_{3,2} \oplus k_{3,2} & a_{3,3} \oplus k_{3,3} \end{bmatrix}.$$

$$\boldsymbol{K}_i = \begin{bmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \end{bmatrix},$$

$$\boldsymbol{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

Ki = W[4i, 4i + 3]

= W[4i]W[4i + 1]W[4i + 2]W[4i + 3],

i=0,…,10.

W[4i+j] = $k_{0,j}k_{1,j}k_{2,j}k_{3,j}$, j=0,1,2,3.

A = M (trạng thái hiên tại)

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

Substitute-Bytes

$$sub(\boldsymbol{A}) = [S(a_{ij})]_{4\times 4} = \begin{bmatrix} S(a_{0,0}) & S(a_{0,1}) & S(a_{0,2}) & S(a_{0,3}) \\ S(a_{1,0}) & S(a_{1,1}) & S(a_{1,2}) & S(a_{1,3}) \\ S(a_{2,0}) & S(a_{2,1}) & S(a_{2,2}) & S(a_{2,3}) \\ S(a_{3,0}) & S(a_{3,1}) & S(a_{3,2}) & S(a_{3,3}) \end{bmatrix},$$

$$sub^{-1}(\boldsymbol{A}) = [S^{-1}(a_{ij})]_{4\times 4},$$

$$sub(sub^{-1}(\boldsymbol{A})) = sub^{-1}(sub(\boldsymbol{A})) = \boldsymbol{A}.$$

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

Shift-Rows

$$shr(\boldsymbol{A}) = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,1} & a_{1,2} & a_{1,3} & a_{1,0} \\ a_{2,2} & a_{2,3} & a_{2,0} & a_{2,1} \\ a_{3,3} & a_{3,0} & a_{3,1} & a_{3,2} \end{bmatrix}, \quad shr^{-1}(\boldsymbol{A}) = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,3} & a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,2} & a_{2,3} & a_{2,0} & a_{2,1} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,0} \end{bmatrix}.$$

$$shr(shr^{-1}(\boldsymbol{A})) = shr^{-1}(shr(\boldsymbol{A})) = \boldsymbol{A}.$$

3.3. Mã hoá đối xứng
3.3.2. Mã hoá AES (tt)

Mix-Columns

$$mic(\boldsymbol{A}) = [a'_{ij}]_{4\times 4},$$

$$\begin{aligned}
a'_{0,j} &= \mathcal{M}(a_{0,j}) \oplus [\mathcal{M}(a_{1,j}) \oplus a_{1,j}] \oplus a_{2,j} \oplus a_{3,j},\\
a'_{1,j} &= a_{0,j} \oplus \mathcal{M}(a_{1,j}) \oplus [\mathcal{M}(a_{2,j}) \oplus a_{2,j}] \oplus a_{3,j},\\
a'_{2,j} &= a_{0,j} \oplus a_{1,j} \oplus \mathcal{M}(a_{2,j}) \oplus [\mathcal{M}(a_{3,j}) \oplus a_{3,j}],\\
a'_{3,j} &= [\mathcal{M}(a_{0,j}) \oplus a_{0,j}] \oplus a_{1,j} \oplus a_{2,j} \oplus \mathcal{M}(a_{3,j}).
\end{aligned}$$

$$mic^{-1}(\boldsymbol{A}) = [a''_{ij}]_{4\times 4},$$

$$\begin{aligned}
a''_{0,j} &= \mathcal{M}_1(a_{0,j}) \oplus \mathcal{M}_2(a_{1,j}) \oplus \mathcal{M}_3(a_{2,j}) \oplus \mathcal{M}_4(a_{3,j}),\\
a''_{1,j} &= \mathcal{M}_4(a_{0,j}) \oplus \mathcal{M}_1(a_{1,j}) \oplus \mathcal{M}_2(a_{2,j}) \oplus \mathcal{M}_3(a_{3,j}),\\
a''_{2,j} &= \mathcal{M}_3(a_{0,j}) \oplus \mathcal{M}_4(a_{1,j}) \oplus \mathcal{M}_1(a_{2,j}) \oplus \mathcal{M}_2(a_{3,j}),\\
a''_{3,j} &= \mathcal{M}_2(a_{0,j}) \oplus \mathcal{M}_3(a_{1,j}) \oplus \mathcal{M}_4(a_{2,j}) \oplus \mathcal{M}_1(a_{3,j}).
\end{aligned}$$

$$mic(mic^{-1}(\boldsymbol{A})) = mic^{-1}(mic(\boldsymbol{A})) = \boldsymbol{A}.$$

3.3. Mã hoá đối xứng
3.3.3. Ưu và nhược điểm

Ưu điểm

- Hiệu suất tính toán ít hơn.
- Nhanh hơn.
- Không tốn quá nhiều tài nguyên.
- Ngăn cản lan truyền sự tổn hại trên toàn hệ thống

Nhược điểm

- Gặp vấn đề trong quá trình trao đổi khóa.
- Có quá nhiều khóa để quản lý.
- Nguồn gốc và tính xác thực không được đảm bảo

3.4. Mã hoá bất đối xứng

Trong khi mã hóa đối xứng sử dụng một khóa duy nhất để mã hóa và giải mã một thông điệp, thì mã hóa bất đối xứng sử dụng hai khóa khác nhau nhưng có liên quan. Có thể sử dụng bất kỳ khóa nào trong hai khoá để mã hóa hoặc giải mã thông điệp.

Kỹ thuật này phát huy giá trị lớn nhất khi một khóa được giữ bí mật như khoá riêng, chỉ chủ sở hữu của cặp khóa biết. Khóa còn lại đóng vai trò là khóa công khai, được lưu trữ ở một vị trí công cộng mà bất kỳ ai cũng có thể sử dụng. Vì lý do này, mã hóa bất đối xứng thường được gọi là mật mã khóa công khai.

3.4. Mã hoá bất đối xứng
3.4.1. Thuật toán RSA

RSA (1977), tên tác giả Rivest, Shamir và Adleman.

+ Lựa chọn khóa chung và khóa riêng: 2 số nguyên tố p và q

Tính $n = p*q$ và $z = (p - 1)(q – 1)$.

Chọn e là số nguyên tố cùng nhau z: $gcd(e,z) = 1$, $1< e < z$

Tìm d: $e*d \bmod z = 1$.

Khoá công khai PK = {n, e}; Khoá riêng SK = {n,d}

+ Thuật toán mã hóa và giải mã

Ciphertext $c = m^e \bmod n$

Plaintext $m = c^d \bmod n$

3.4. Mã hoá bất đối xứng
3.4.1. Thuật toán RSA (tt)

RSA

+ 2 số nguyên tố p và q
Tính n = p*q và z = (p - 1)(q – 1)
Chọn e: gcd(e,z) = 1, 1< e < z
Tìm d: e*d mod z = 1.
Khoá công khai (n, e); Khoá riêng (n,d)

+ Thuật toán mã hóa và giải mã
Ciphertext c = m^e mod n
Plaintext m = c^d mod n

Ví dụ:
+ p = 3; q =5
=> n = 3*5 = 15; z = 8
=> e (3,5,7)
=> e = 3;
=> 3*d mod 8 = 1
Thử d = (3,5,...29,…)
=> ~~d = 3~~; d= 11
=> (n,e) = (15,3); (n,d) = (15,11)
Với m = 2
=> c = 2^3 mod 15 = 8
=> m = 8^11 mod 15 = 2

3.4. Mã hoá bất đối xứng
3.4.2. Thuật toán ElGamal

RSA (1984), trên cơ sở ý tưởng từ Diffie-Hellman.
Chọn một số nguyên tố lớn p và hai số nguyên ngẫu nhiên ε và a, cả hai đều nhỏ hơn p.

$$y = \varepsilon^a \pmod p$$

Bây giờ khóa công khai được lấy là {p,ε,y}, khoá mật là {p,a}.

3.4. Mã hoá bất đối xứng
3.4.2. Thuật toán ElGamal (tt)

Mã hoá

Chọn giá trị k (k<p); $K=y^k \bmod p$

Với bản rõ là m

$C_1 = \varepsilon^k \bmod p$; $C_2 = K*m \bmod p$

Cặp (C_1,C_2) được gửi đi

Giải mã

Bản mã là (C_2/C_1)

$m = C_2/C_1^a \bmod p$

$y = \varepsilon^a \pmod p$

PK = {p,ε,y}, SK = {p,a}.

VD: Với p = 11, ε = 3 và a = 6. Với bản rõ: m = 6, và chon random k = 7.Tính bản mã C?

3.4. Mã hoá bất đối xứng
3.4.3. Ưu và nhược điểm

Ưu điểm:

- Độ an toàn của mã hóa bất đối xứng cao.
- Cung cấp được tính chứng thực, toàn vẹn dữ liệu.
- Thuận tiện phân phối khóa.

Nhược điểm:

- Xử lý chậm hơn so với mã hóa đối xứng.
- Gặp khó khăn nếu mất khóa bí mật.
- Phức tạp trong vấn đề tìm số nguyên tố và ngẫu nhiên hợp lý.

3.5. So sánh mã hoá đối xứng và bất đối xứng

Các thuật toán mật mã thường nhóm thành hai loại là mã hóa đối xứng và bất đối xứng, nhưng trên thực tế, nhiều hệ thống mã hóa phổ biến hiện nay kết hợp cả hai phương pháp này.

Khi triển khai các mật mã, người dùng cần quyết định kích thước khóa, vì sức mạnh của hệ thống mật mã hóa phụ thuộc vào độ dài của khóa. Khóa càng dài, số lần đoán ngẫu nhiên cần để phá mã càng tăng lên, do đó làm tăng cường độ an toàn cho hệ thống.

Tính an toàn của hệ thống mật mã không không dựa trên việc giữ bí mật thuật toán mã hóa. Các thuật toán thường được công khai để cho phép các nhà nghiên cứu tìm ra điểm yếu. Yếu tố then chốt để bảo mật dữ liệu là giữ bí mật các thành phần của khóa, và đảm bảo khóa có độ dài đủ lớn.

3.5. So sánh mã hoá đối xứng và bất đối xứng

Mã hoá đối xứng	Mã hoá bất đối xứng
Sử dụng một khoá duy nhất cho cả mã hoá và giải mã	Có hai khoá, khoá công khai và khoá riêng, một khoá để mã hoá và khoá còn lại để giải mã
Kích thước bản mã bằng hoặc nhỏ hơn bản rõ	Kích thước bản mã bằng hoặc lớn hơn bản rõ
Quá trình mã hoá diễn ra rất nhanh	Quá trình mã hoá diễn ra chậm
Được sử dụng khi cần truyền lượng lớn dữ liệu	Được sử dụng khi cần truyền lượng nhỏ dữ liệu
Độ dài khoá sử dụng là 128 bit hoặc 256 bit	Độ dài khoá sử dụng là 2048 hoặc cao hơn
Hiệu quả vì được sử dụng để xử lý lượng lớn dữ liệu	Kém hiệu quả hơn vì chỉ có thể xử lý được một lượng dữ liệu nhỏ
Tính bảo mật thấp hơn vì chỉ sử dụng một khoá cho cả mã hoá và giải mã	Tính bảo mật cao hơn vì sử dụng hai khoá, một mã hoá và một để giải mã

Cảm ơn các bạn,

Chúc các bạn thành công!