ĐẠI NAM
UNIVERSITY

CHƯƠNG 2: MÃ KHÓA BÍ MẬT

Giảng viên: TS. Trần Quý Nam
(namtq@dainam.edu.vn)

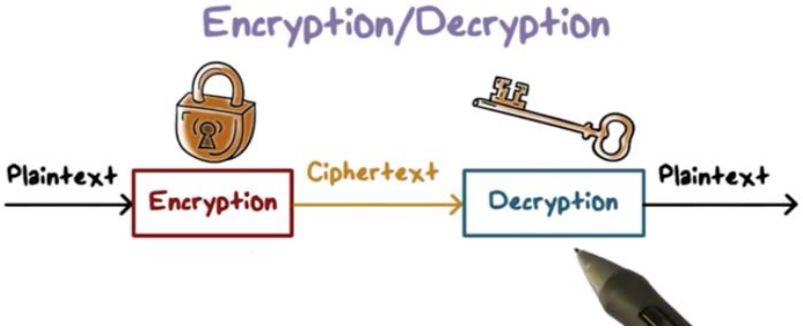
Giới thiệu

Mật mã học (cryptography)

- Ngành khoa học nghiên cứu các phương pháp toán học để mã hóa giữ mật thông tin. Bao gồm mã hóa và giải mã.

  • **Mã hóa:** biến đổi cách thức biểu diễn thông tin từ dạng bản rõ sang dạng bản mã => che giấu, giữ mật thông tin(lưu trữ, truyền thông tin đi)

  • **Giải mã:** ngược lại **Mã hóa** là biến bản mã thành bản rõ.



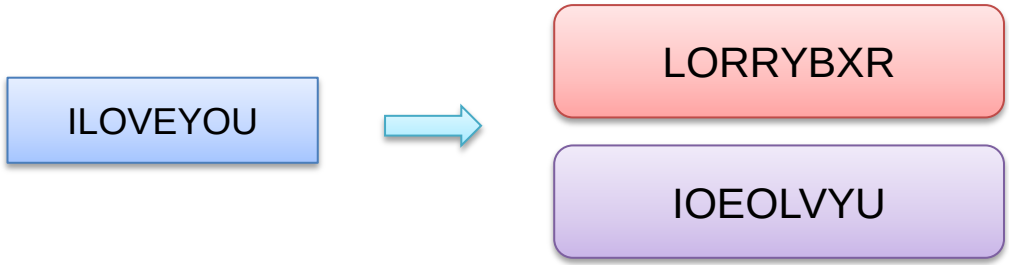
Giới thiệu

- ✓ Plaintext: Bản rõ (ta có thể đọc được)
- ✓ Ciphertext: Bản mã (chỉ người giải mã hiểu được)

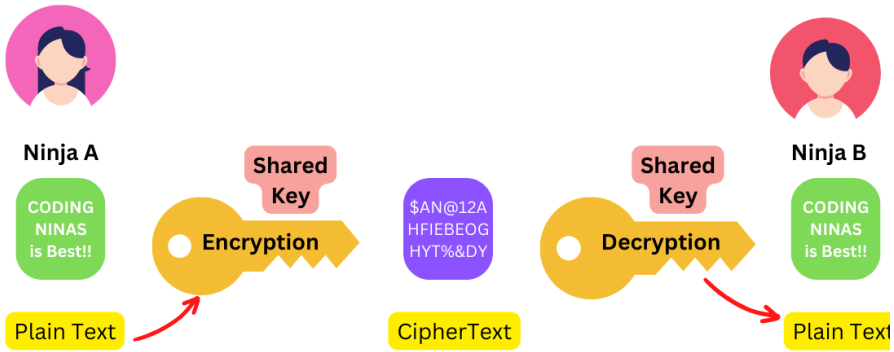
Giới thiệu

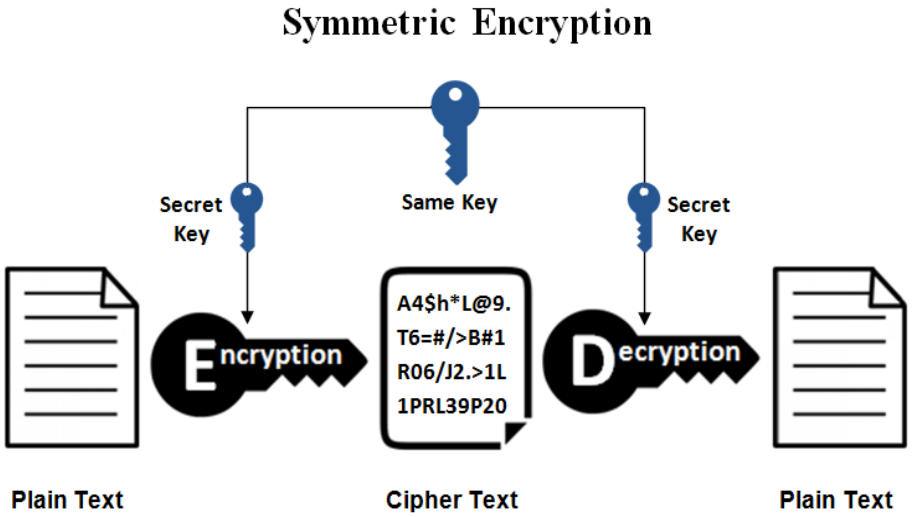
Mật mã học (cryptography)

- ✓ Bản rõ / thông điệp gốc (Plaintext - **P**) (mọi người có thể hiểu được)
- ✓ Bản mã / thông điệp mã hóa (Ciphertext - **C**) (chỉ người giải mã hiểu được)
- ✓ Mã hóa (Encryption - **E**)
- ✓ Giải mã (Decryption - **D**)



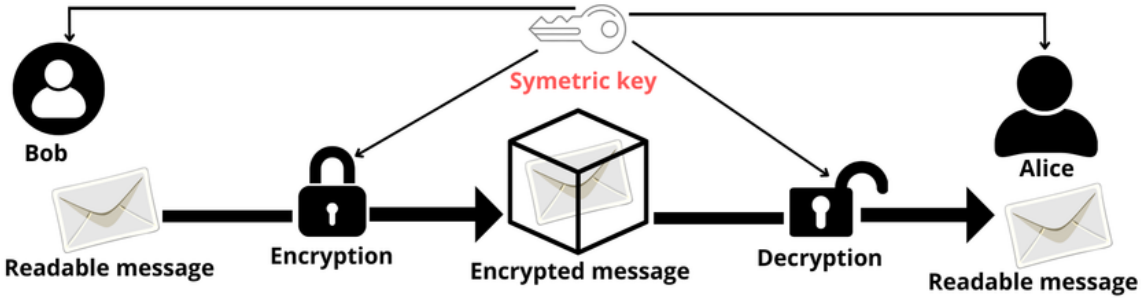
Giới thiệu


Ninja A
CODING NINAS is Best!!
Plain Text
Shared Key
Encryption
$AN@12A HFIEBEOG HYT%&DY
CipherText
Shared Key
Decryption
Ninja B
CODING NINAS is Best!!
Plain Text


Symmetric Encryption
Secret Key
Same Key
Secret Key
Encryption
A4$h*L@9.
T6=#/>B#1
R06/J2.>1L
1PRL39P20
Decryption
Plain Text
Cipher Text
Plain Text

Nội dung Chương 2: Mã hóa đối xứng

1. Kỹ thuật mã cổ điển
2. Mã khối DES
3. Mã khối AES



ĐẠI NAM
UNIVERSITY

Bài 3

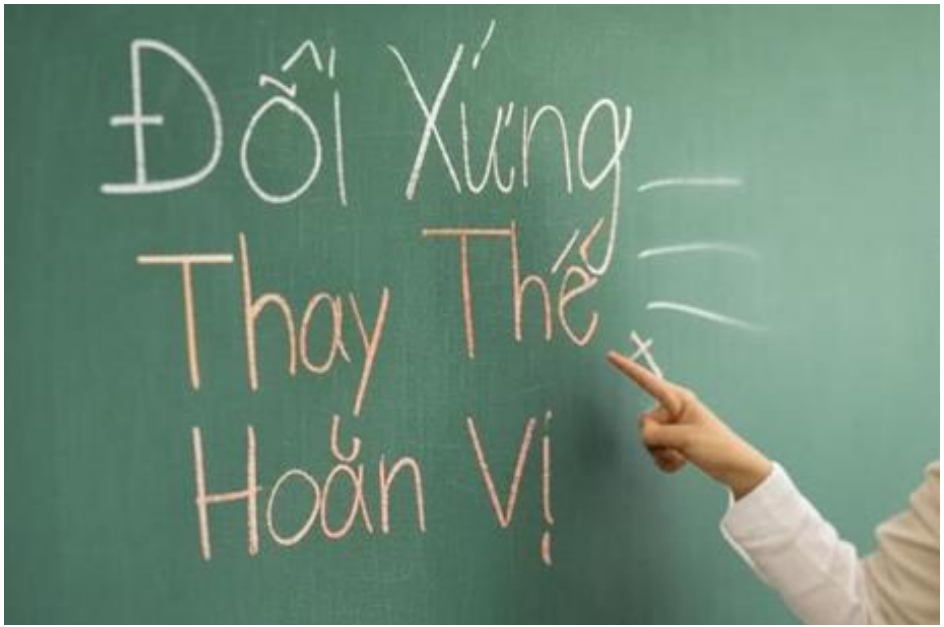
KỸ THUẬT MÃ CỔ ĐIỂN

Nội dung Bài 3

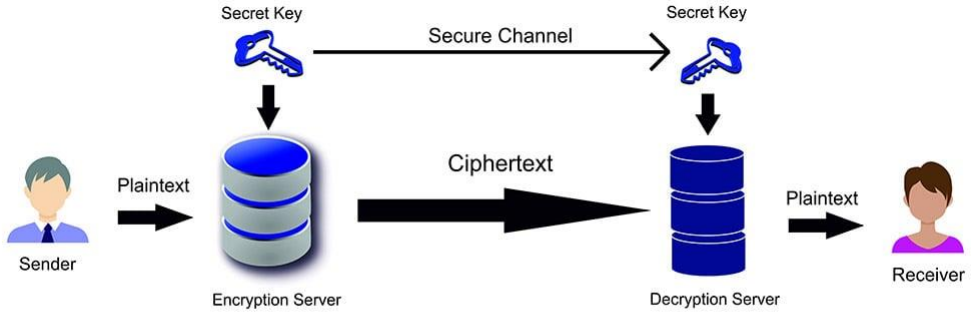
1. Mã đối xứng

2. Mã thay thế

3. Mã chuyển vị



Mã đối xứng



Symmetric Cryptography

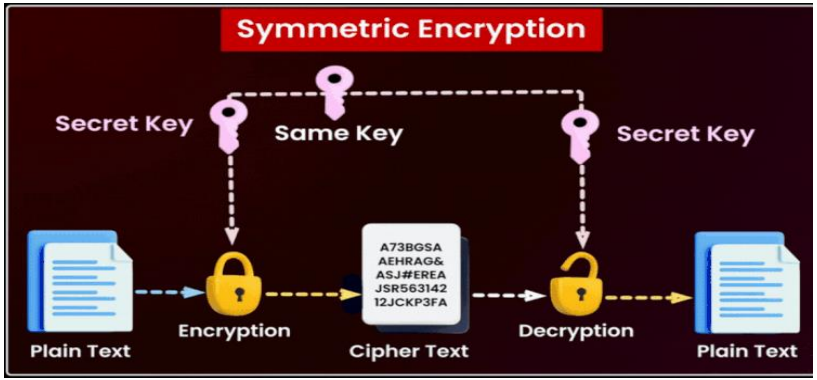
Những hệ mật được sử dụng chung 1 khóa trong quá trình mã hóa và mã hóa.

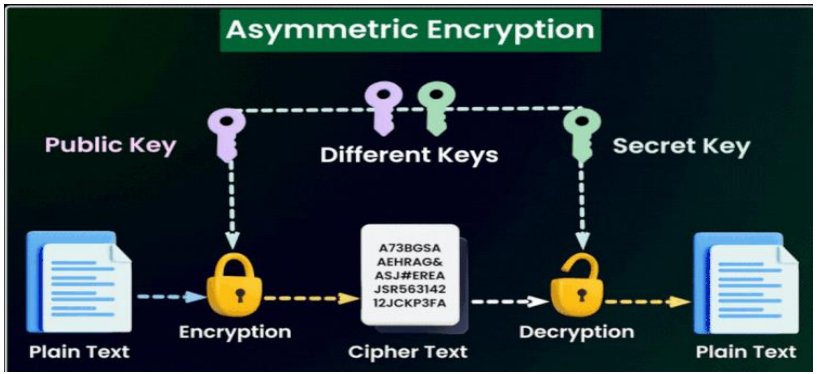
Điều gì đảm bảo thông tin không bị lộ?

=> Khóa phải được giữ bí mật tuyện đối

Mã đối xứng

- **Mã đối xứng** sử dụng **cùng một khóa** cho cả quá trình mã hóa và giải mã.
- Công thức mã hóa và giải mã đều sử dụng cùng một khóa **K (Secret key)**.





Mã đối xứng

- **Ưu điểm**:

  Tốc độ cao và hiệu quả, phù hợp cho việc mã hóa khối lượng lớn dữ liệu nhờ sử dụng cùng một khóa cho cả mã hóa và giải mã.

- **Nhược điểm**:

  Vấn đề quản lý và phân phối khóa an toàn, dễ bị phá vỡ nếu khóa bị lộ.

Mã đối xứng

Symmetric encryption, also referred to as **conventional encryption** or **single-key encryption**, was the only type of encryption in use prior to the development of public-key encryption in the 1970s. It remains by far the most widely used of the two types of encryption. Part Two examines a number of symmetric ciphers. In this chapter, we begin with a look at a general model for the symmetric encryption process; this will enable us to understand the context within which the algorithms are used. Next, we examine a variety of algorithms in use before the computer era. Finally, we look briefly at a different approach known as steganography. Chapters 4 and 6 introduce the two most widely used symmetric cipher: DES and AES.

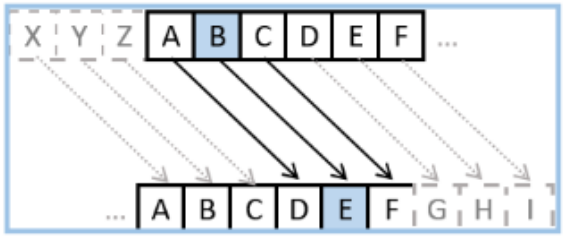
Before beginning, we define some terms. An original message is known as the **plaintext**, while the coded message is called the **ciphertext**. The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**; restoring the plaintext from the ciphertext is **deciphering** or **decryption**. The many schemes used for encryption constitute the area of study known as **cryptography**. Such a scheme is known as a **cryptographic system** or a **cipher**. Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**. Cryptanalysis is what the layperson calls "breaking the code." The areas of cryptography and cryptanalysis together are called **cryptology**.

3.1 SYMMETRIC CIPHER MODEL

MÃ THAY THẾ

Mã thay thế

- Mã thay thế là phương pháp mã hóa trong đó mỗi ký tự trong văn bản gốc được **thay thế** bằng một ký tự khác dựa trên một quy tắc cố định.
- Quy tắc này có thể là một bảng thay thế, một phép toán nào đó(Caesar)



SHIFT +3

This Caesar cipher has a shift of 3 characters.

The letter 'A' becomes a 'D'. The letter 'B' becomes 'E'.

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
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

Plaintext

Ciphertext

Mã thay thế

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Gán cho mỗi chữ cái một con số nguyên từ 0 đến 25:
- Với mỗi ký tự trong P thay bằng chữ mã hóa *C*, trong đó:

  $C = (P + k) \bmod 26$ (mod: phép chia lấy số dư)

- Và quá trình giải mã đơn giản là:

  $P = (C - k) \bmod 26$

- Số *k* được gọi là khóa.

Mã thay thế

Mã hóa "HELLO" bằng Caesar Cipher với khóa là 3:

- **Plaintext (P):** "HELLO"
- **Khóa bí mật (K):** 3 (Dịch chuyển 3 bước trong bảng chữ cái)
- **Mã hóa (E):** C = (P + K) mod 26

✓ Sau các bước mã hóa: E(K,P) => Ciphertext (C): "KHOOR"

✓ Giải mã (D) với cùng khóa K: D(K,C) => Plaintext (P) : "HELLO"

Mã thay thế

Quá trình mã hóa (E)

➢ *Các bước mã hóa :*

- 'H' (vị trí 7 trong bảng chữ cái) → 'K' (vị trí 10)
- 'E' (4) → 'H' (7)
- 'L' (11) → 'O' (14)
- 'L' (11) → 'O' (14)
- 'O' (14) → 'R' (17)

=> Ciphertext (C): "KHOOR"

Giải mã (D) với cùng khóa 3:

➢ *Để giải mã, ta sẽ trừ đi khóa từ mỗi ký tự của ciphertext* **P = (C - K) mod 26***:*

- 'K' (vị trí 10) → 'H' (vị trí 7)
- 'H' (7) → 'E' (4)
- 'O' (14) → 'L' (11)
- 'O' (14) → 'L' (11)
- 'R' (17) → 'O' (14)

=> Plaintext (P): "HELLO"

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
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

Mã thay thế

3.2 SUBSTITUTION TECHNIQUES

In this section and the next, we examine a sampling of what might be called classical encryption techniques. A study of these techniques enables us to illustrate the basic approaches to symmetric encryption used today and the types of cryptanalytic attacks that must be anticipated.

The two basic building blocks of all encryption techniques are substitution and transposition. We examine these in the next two sections. Finally, we discuss a system that combines both substitution and transposition.

A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.[1] If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

Mã thay thế

Với bản chữ cái Tiếng Việt (29 ký tự) :

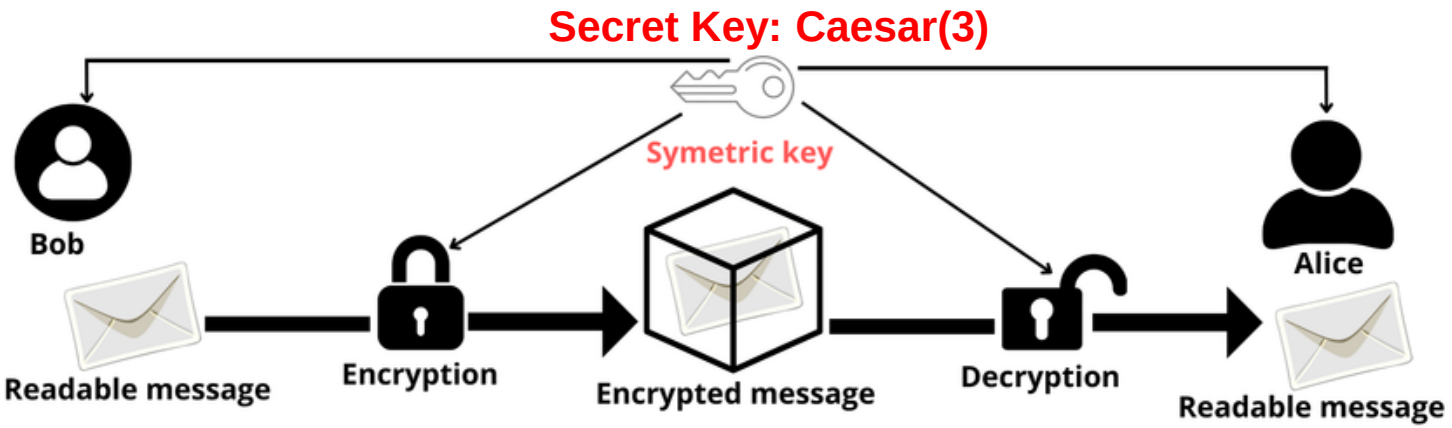
- Gán cho mỗi chữ cái một con số nguyên từ 0 đến 28:
- Phương pháp Ceasar biểu diễn tiếng Việt như sau: với mỗi chữ cái p thay bằng chữ mã hóa C, trong đó:

  C = (p + k) mod 29
- Và quá trình giải mã đơn giản là:

  p = (C – k) mod 29

A	Ă	Â	B	C	D	Đ	E	Ê	G	H	I	K	L	M	N	O	Ô	Ơ	P	Q	R	S	T	U	Ư	V	X	Y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Mã thay thế



Plaintext (P): "HELLO"

E(K,P)

Ciphertext (C): "KHOOR"

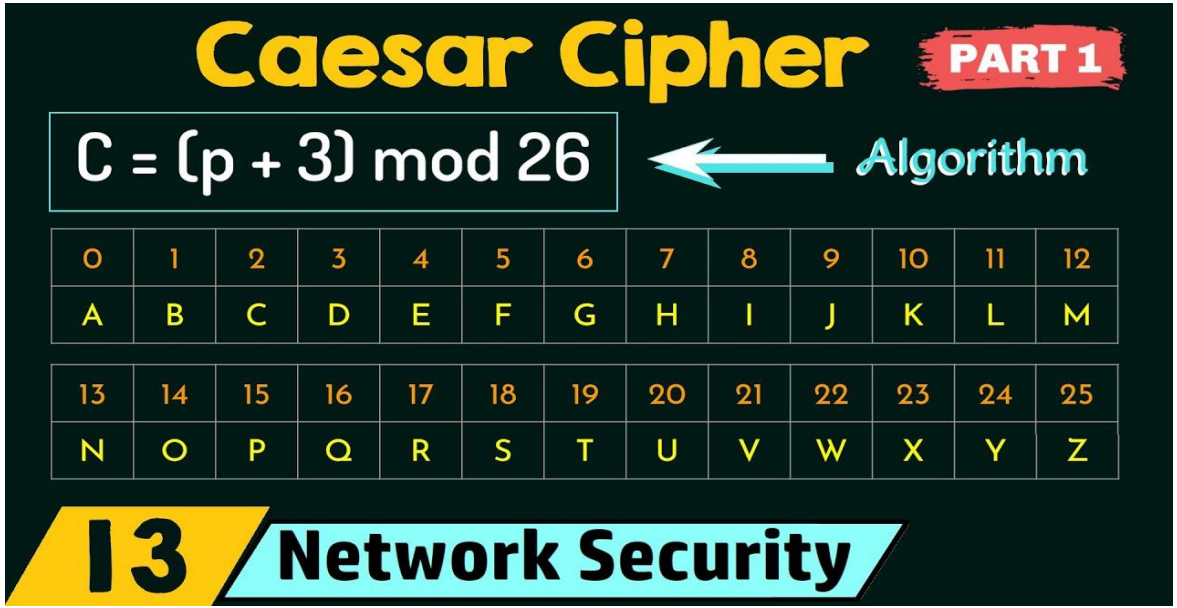
D(K,C)

Plaintext (P): "HELLO"

Mã thay thế

$$C = \mathrm{E}(3, p) = (p + 3) \bmod 26$$

$$C = \mathrm{E}(k, p) = (p + k) \bmod 26$$



Phá mã

Phá Mã Caesar bằng Brute Force

Phá mã

Phá Mã Caesar bằng Brute Force

- **Brute Force:** Phương pháp thử tất cả các khóa khả thi để tìm khóa đúng.
- Với mã Caesar cần thử bao nhiêu khóa?

  Chỉ cần thử 25 khóa (từ 1 đến 25, vì 26 không thay đổi thông điệp).

➢ **Cách hoạt động**

- Nhận thông điệp mã hóa ("LORYHBXR").
- Thử dịch ngược với từng khóa (1, 2, 3, ...).
- Kiểm tra kết quả để tìm thông điệp có ý nghĩa

(ví dụ: "ILOVEYOU" với khóa 3).

➢ **Ví dụ**

1) Khóa 1: "KNQXGAXQ"
2) Khóa 2: "JMPWFZWP"
3) Khóa 3: "ILOVEYOU" □

KEY	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	**meet**	**me**	**after**	**the**	**toga**	**party**
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrcp	rfc	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	ojbv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlq
12	dvvk	dv	rwkvi	kyv	kfxr	grikp
13	cuuj	cu	qvjuh	jxu	jewq	fqhjo
14	btti	bt	puitg	iwt	idvp	epgin
15	assh	as	othsf	hvs	hcuo	dofhm
16	zrrg	zr	nsgre	gur	gbtn	cnegl
17	yqqf	yq	mrfqd	ftq	fasm	bmdfk
18	xppe	xp	lqepc	esp	ezrl	alcej
19	wood	wo	kpdob	dro	dyqk	zkbdi
20	vnnc	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxc

Phá mã

- Trong 25 trường hợp, chỉ có trường hợp k=3 thì bản giải mã tương ứng là có ý nghĩa.
- Do đó đối thủ có thể chắc chắn rằng “meet me after the toga party” là bản rõ ban đầu.

Mã đối xứng

Cryptanalysis and Brute-Force Attack

Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:

- **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
- **Brute-force attack:** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

MÃ CHUYỂN VỊ

Mã hoán vị

- Mã chuyển vị là một phương pháp mã hóa trong đó các ký tự trong văn bản gốc được hoán đổi vị trí theo một quy tắc nhất định thay vì bị thay thế bằng các ký tự khác.

+ *Ưu điểm:* Mã chuyển vị đơn giản và dễ thực hiện, có thể cung cấp một lớp bảo mật cơ bản.

+ *Nhược điểm:* Dễ bị phá mã nếu kẻ tấn công biết hoặc có thể suy ra quy tắc hoán đổi. Trong hầu hết các trường hợp, mã chuyển vị không đủ mạnh để bảo vệ thông tin trước các phương pháp tấn công hiện đại.

Mã hoán vị

- Phương pháp này tổng quát hóa phương pháp Ceasar bằng cách dòng mã hóa không phải là một dịch chuyển *k* vị trí của các chữ cái A, B, C, … nữa mà là một *hoán vị* của 26 chữ cái này. Lúc này mỗi hoán vị được xem như là một khóa.
- Việc mã hóa được tiến hành bằng cách thay thế một chữ cái trong bản rõ thành một chữ cái trong bản mã, nên phương pháp này được gọi là phương pháp thay thế.

Plain: 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

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

Plaintext: ifwewishtoreplaceletters

Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

Mã hoán vị

- Số lượng hoán vị của 26 chữ cái là 26! =4x10^{26}(tương đương với số khóa).
- Vì 26! là một con số khá lớn → tấn công phá mã vét cạn khóa là bất khả thi (6400 thiên niên kỷ với tốc độ thử khóa là 109 khóa/giây).

→ phương pháp này được xem là một phương pháp mã hóa an toàn trong suốt 1000 năm sau công nguyên.

- Ví dụ:
- *Chữ ban đầu:* 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
- *Khóa :* Z P B Y J R S K F L X Q N W V D H M G U T O I A E C
- Như vậy bản rõ: meet me after the toga party
- được mã hóa thành: NJJU NJ ZRUJM UKJ UVSZ DZMUE

Mã hoán vị

Chữ ban đầu: 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

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

Như vậy bản rõ meet me after the toga party

được mã hóa thành: NJJU NJ ZRUJM UKJ UVSZ DZMUE

Mã chuyển vị

- Các ký tự của bản rõ (P) được sắp xếp lại theo một quy tắc nhất định để tạo thành bản mã (C), mà không thay đổi bản chất của các ký tự.
- **Nguyên tắc cơ bản**: Bản rõ được viết vào một cấu trúc (thường là ma trận hoặc bảng) theo một thứ tự nhất định.
  - Các ký tự được đọc ra theo một thứ tự khác để tạo bản mã.
  - Để giải mã, cần biết cấu trúc và thứ tự hoán vị ban đầu.

✓ **Mã hóa chuyển vị hàng rào từ "HELLO"**

P: "HELLO" → H . L . O . E . L . → **C: "HLOEL"**

Encryption

Mã chuyển vị

❖ **Chuyển vị hàng rào** (Rail Fence Cipher): Sắp xếp ký tự theo dạng zig-zag trên các "hàng rào" và đọc theo hàng.

Mã hóa chuyển vị hàng rào từ "HELLO"

➢ **Nguyên tắc**:

- Chọn số hàng (rails) để tạo "hàng rào" .
- Viết bản rõ theo dạng zig-zag qua các hàng.
- Đọc bản mã theo từng hàng từ trên xuống dưới.

H . L . O
. E . L .

- Hàng 1: HLO
- Hàng 2: EL

=> Bản mã(C): HLOEL

Mã chuyển vị

3.3 TRANSPOSITION TECHNIQUES

All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.

The simplest such cipher is the rail fence technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows. For example, to encipher the message "meet me after the toga party" with a rail fence of depth 2, we write the following:

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

The encrypted message is

```
MEMATRHTGPRYETEFETEOAAT
```

Mã chuyển vị

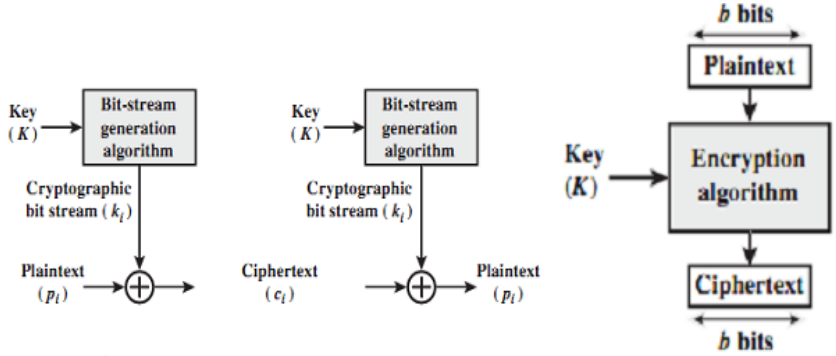
- Bản rõ được viết trên một hình chữ nhật và đọc theo cột. Thứ tự các cột trở thành khóa của giải thuật.
- Ví dụ: bản rõ “meet me at the toga party”
  - Key: 4 1 2 5 3 6
  - Bản rõ:

m	e	e	t	m	e
a	t	t	h	e	t
o	g	a	p	a	r
t	y	x	y	z	w

- bản mật **etgyetaxmeazmaotthpyetrw**
- Để tăng độ mật, có thể áp dụng hoán vị nhiều lần

Mã dòng và mã khối

- Mã dòng (stream cipher): mã hóa một dòng dữ liệu số một bit hoặc một byte tại một thời điểm.
- Mã khối (block cipher): mã hóa/giải mã một khối của bản rõ được xử lý như một tổng thể và dùng để tạo ra khối bản mã có độ dài bằng nhau. Thông thường kích cỡ khối là 64 hoặc 128 bit được sử dụng.

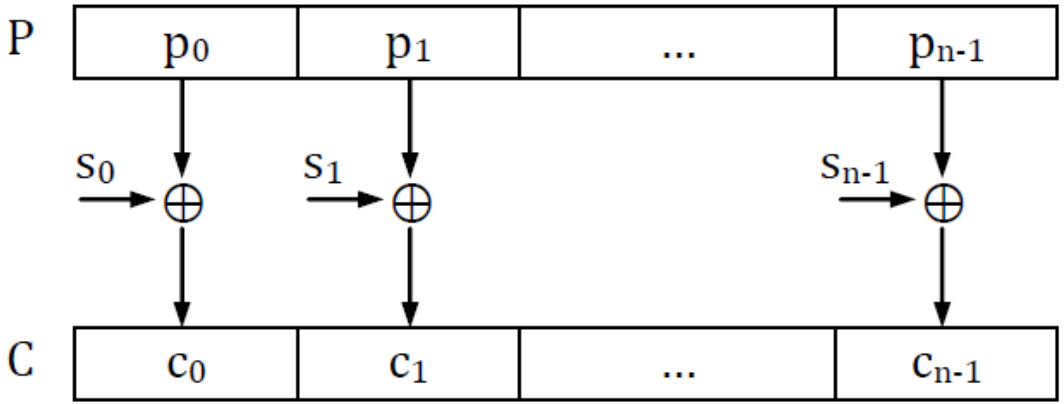


(a) Stream cipher using algorithmic bit-stream generator

(b) Block cipher

Mã dòng

- Đối với mã dòng, các số s_i được sinh ra phải đảm bảo một độ ngẫu nhiên nào đó (chu kỳ tuần hoàn dài).
- Điểm quan trọng nhất của các mã dòng là bộ sinh số ngẫu nhiên.



Hình 3-1. Mô hình mã dòng

Mã dòng

Mã dòng có các đặc tính sau:

- Kích thước một đơn vị mã hóa: gồm *k* bít. Bản rõ được chia thành các đơn vị mã hóa: $P \rightarrow p_0 p_1 p_2 \dots p_{n-1} \quad (p_i : k\ bít)$

- Một bộ sinh dãy số ngẫu nhiên: dùng một khóa *K* ban đầu để sinh ra các số ngẫu nhiên có kích thước bằng kích thước đơn vị mã hóa:

$$StreamCipher(K) \rightarrow S = s_0 s_1 s_2 \dots s_{n-1} \quad (s_i : k\ bít)$$

- Mỗi số ngẫu nhiên được XOR với đơn vị mã hóa của bản rõ để có được bản mã.

$$c_0 = p_0 \oplus s_0,\ c_1 = p_1 \oplus s_1 \dots \ ; C = c_0 c_1 c_2 \dots c_{n-1}$$

Mã dòng

- Ví dụ: mã hóa ký tự ‘A’ bởi Alice
- Ký tự ‘A’ trong bảng mã ASCII được tướng ứng với mã $65_{10}=1000001_2$ được mã hóa bởi hệ khóa $z_1,\ldots,z_7 = 0101101$
- Hàm mã hóa:

plaintext x_i:	1000001	= ‘A’	(ASCII symbol)
key stream z_i:	0101101		
ciphertext y_i:	1101100	= ‘l’	(ASCII symbol)

- Hàm giải mã:

ciphertext y_i:	1101100	= ‘l’	(ASCII symbol)
key stream z_i:	0101101		
plaintext x_i:	1000001	= ‘A’	(ASCII symbol)

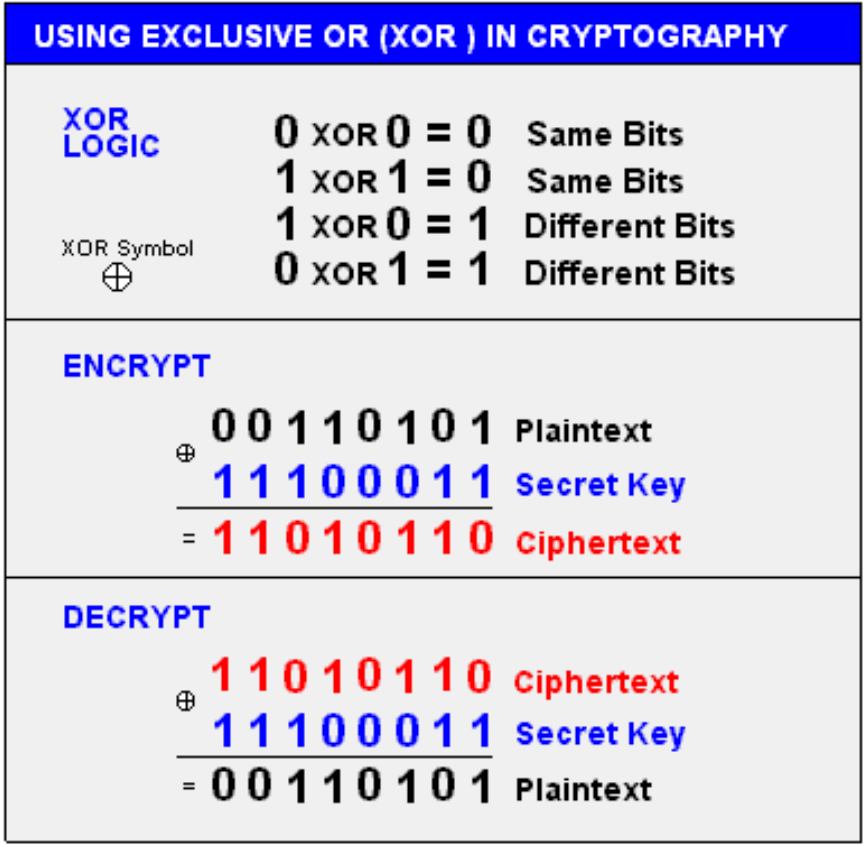
ENGLISH PRACTICE

A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher. In the ideal case, a one-time pad version of the Vernam cipher would be used (Figure 3.7), in which the keystream (k_i) is as long as the plaintext bit stream (p_i). If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream. However, the key-stream must be provided to both users in advance via some independent and secure channel. This introduces insurmountable logistical problems if the intended data traffic is very large.

Mã dòng

Đặc điểm của mã dòng:

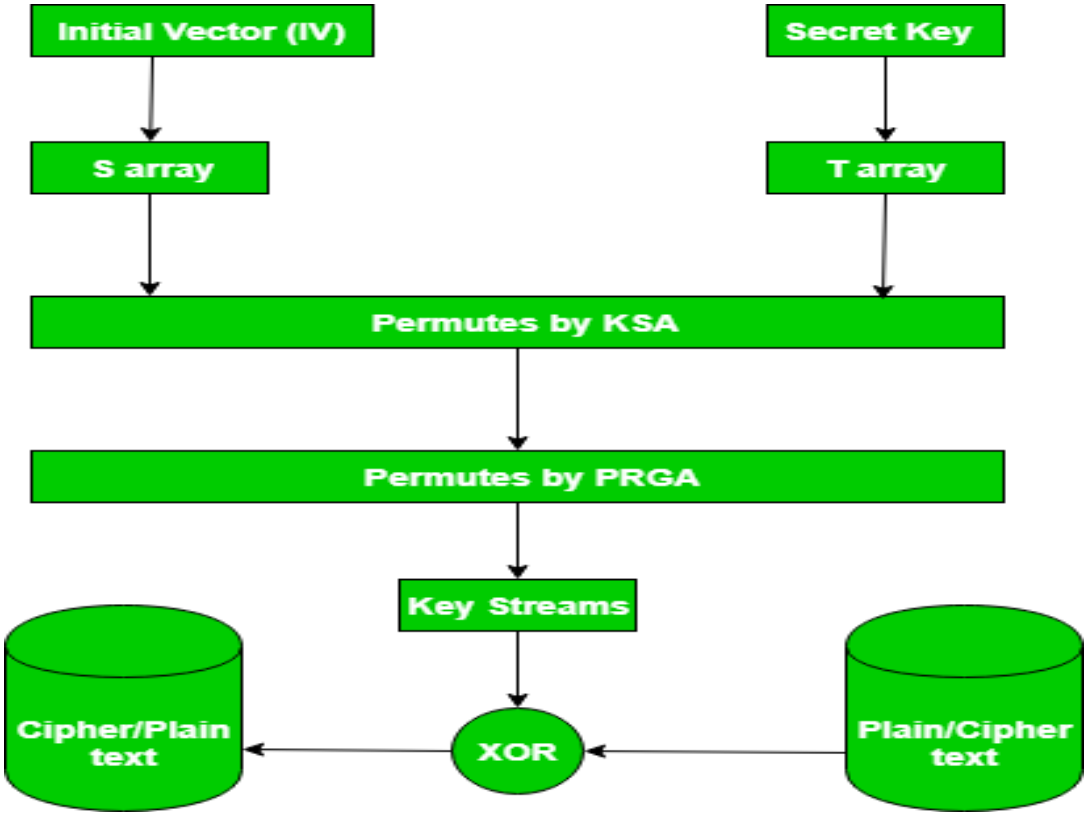
- Mã hóa bit hoặc byte
- Khóa dòng (Keystream)
- Phép toán XOR
- Hiệu suất cao
- Không cần bộ đệm



Mã dòng

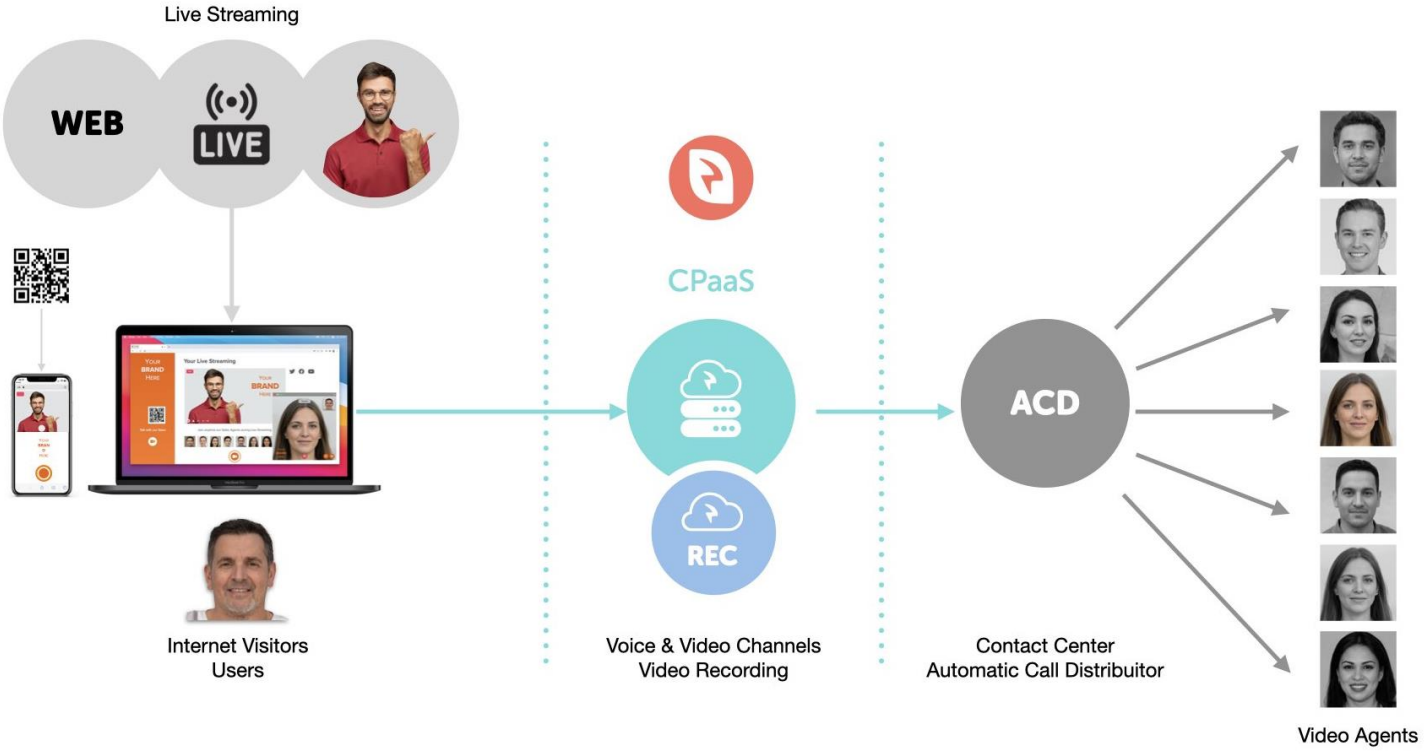
Thuật toán phổ biến:

- RC4
- Salsa20
- ChaCha

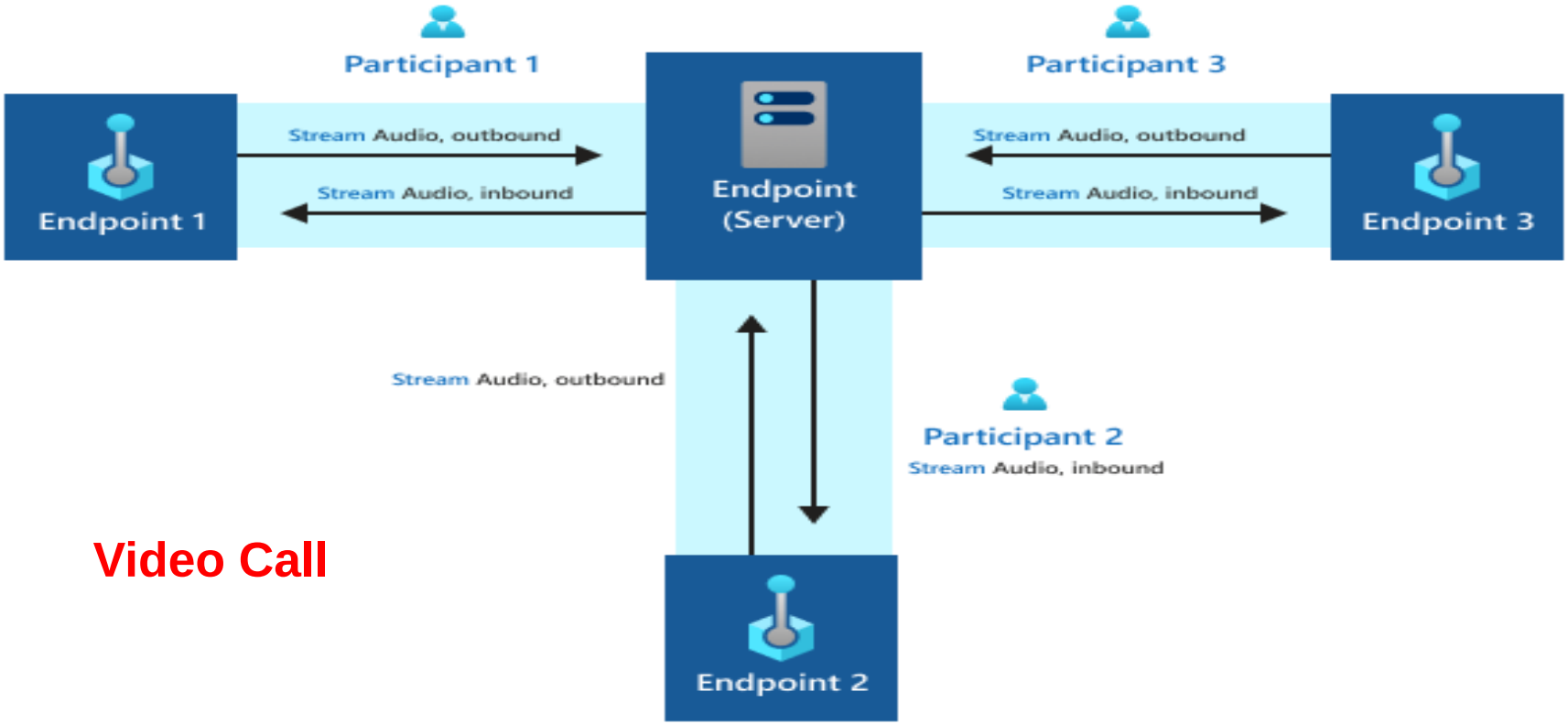


Ứng dụng mã dòng

Video Call



Ứng dụng mã dòng



Video Call

Mã khối

- Ví dụ: chúng ta sử dụng bản rõ là các chữ cái của một *ngôn ngữ* gồm có 8 chữ cái A, B, C, D, E, F, G, H trong đó mỗi chữ cái được biểu diễn bằng 3 bít.
- Nếu có bản rõ là 'head' thì biểu diễn nhị phân tương ứng là: 111100000011
- Giả sử dùng một khóa K gồm **khối 4 bít** 0101 để mã hóa bản rõ trên bằng phép XOR ⊕:

bản rõ:	1111 0000 0011	(head)
khóa:	0101 0101 0101	
bản mã:	1010 0101 0110	(FBCG)

Chữ cái	Nhị phân
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111

- Trong phép mã hóa trên, đơn vị mã hóa không phải là một chữ cái mà là một khối 4 bít. Để giải mã, lấy bản mã XOR một lần nữa với khóa thì có lại bản rõ ban đầu.

Mã khối

Phép toán XOR có một hạn chế là chỉ cần biết *một cặp khối* bản rõ và bản mã, người ta có thể dễ dàng suy ra được khóa và dùng khóa đó để giải các khối bản mã khác (known-plaintext attack). Xét lại ví dụ đầu chương:

bản rõ:	`1111 0000 0011`	`(head)`
khóa:	`0101 0101 0101`	
bản mã:	`1010 0101 0110`	`(FBCG)`

Nếu biết bản mã $c_0 = 1010$ có bản rõ tương ứng là $p_0 = 1111$, thì có thể dễ dàng suy ra khóa là 0101. Nói một cách tổng quát, nếu giữa bản rõ P và bản mã C có mối liên hệ toán học thì việc biết một số cặp bản rõ-bản mã giúp ta có thể tính được khóa K.

Mã khối

- Phòng tránh: làm cho plaintext và ciphertext không có mối liên hệ toán học
- Lập một bảng tra cứu ngẫu nhiên giữa bản rõ và bản mã
- Khóa là toàn bộ bảng

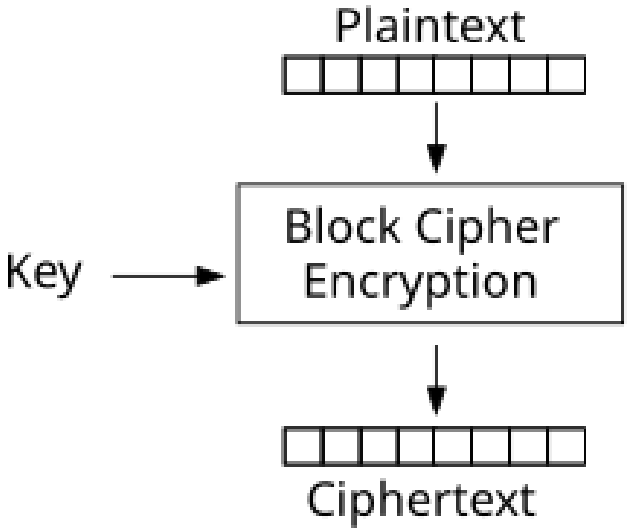
Bản rõ	Bản mã
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Mã khối

- Người gửi và người nhận phải biết toàn bộ bảng khóa để mã hóa và giải mã.
- Phá mã: nếu biết một số cặp bản rõ - bản mã thì cũng chỉ biết được một phần của bảng tra cứu trên ➔ không suy ra được bản rõ cho các bản mã còn lại.
- Muốn phá mã cần biết tất cả cặp bản rõ và bản mã. Nếu chọn kích thước khối là 64 bít thì số dòng của bảng khóa là 2^{64}, một con số rất lớn (và có khoảng $2^{64}!$ bảng khóa như vậy) ➔ Là mã khối an toàn lý tưởng.

Mã khối

- Kích thước khối lớn thì số dòng của bảng khóa cũng lớn ➔ Khó lưu trữ và trao đổi khóa giữa người gửi và người nhận.
- Bảng khóa có 2^{64} dòng mỗi dòng 64 bít do đó kích thước khóa sẽ là $64 \times 2^{64} = 2^{70} \approx 10^{21}$ bít.
- Do đó mã khối an toàn lý tưởng là không khả thi trong thực tế.

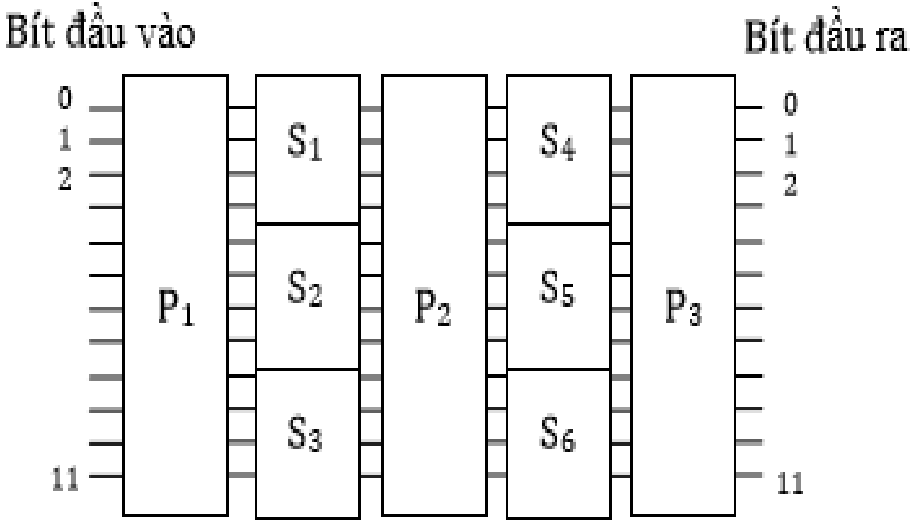


ENGLISH PRACTICE

A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64 or 128 bits is used. As with a stream cipher, the two users share a symmetric encryption key (Figure 4.1b). Using some of the modes of operation explained in Chapter 7, a block cipher can be used to achieve the same effect as a stream cipher.

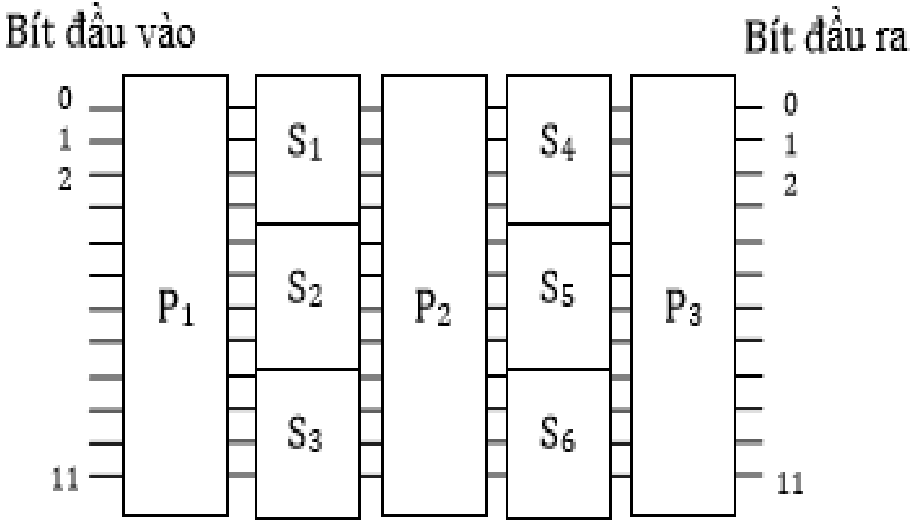
Mã khối

- Thực tế: khóa có kích thước ngắn giả lập một bảng tra cứu có độ an toàn xấp xỉ độ an toàn của mã khối lý tưởng.
- Các mã hóa đơn giản thường là phép thay thế (substitution, S-box) và hoán vị (Permutation, P-box).



Mã khối

- Thực tế: khóa có kích thước ngắn giả lập một bảng tra cứu có độ an toàn xấp xỉ độ an toàn của mã khối lý tưởng.
- Các mã là phép thay thế (substitution, S-box) và hoán vị (Permutation, P-box).



Mã khối

Đặc điểm của mã khối:

- **Kích thước khối cố định**: Mỗi khối dữ liệu mà mã khối xử lý có kích thước cố định, thường là 64 bit hoặc 128 bit. Nếu dữ liệu không đủ kích thước khối, nó sẽ được bổ sung thêm (padding).
- **Hoạt động theo từng khối:** Thay vì mã hóa từng bit, mã khối xử lý dữ liệu theo từng khối cùng một lúc.
- **Chế độ hoạt động**: Mã khối thường được sử dụng trong các chế độ hoạt động khác nhau như ECB (Electronic Codebook), CBC (Cipher Block Chaining), CFB (Cipher Feedback), OFB (Output Feedback), và CTR (Counter). Xem chi tiết ECB, CBC… tại https://viettelidc.com.vn/tin-tuc/tieu-chuan-ma-hoa-du-lieu-aes-la-gi-va-cac-che-do-hoat-dong-cua-aes-phan-2

Mã khối Feistel

- Đề xuất bởi Horst Feistel (1973)
- Mã khối Feistel là một cấu trúc thiết kế được sử dụng trong các thuật toán mã hóa khối (block cipher), là cơ sở cho DES (Data Encryption Standard) và một số biến thể của nó.

- Feistel chia plaintext thành các khối dữ liệu sau đó mã hóa trên từng khối, thực hiện thông qua n rounds, gồm các bước thay thế và chuyển vị trên dữ liệu đầu vào.

ENGLISH PRACTICE

What is a Feistel Cipher?

The Feistel cipher is a design model or structure used to build various symmetric block ciphers, such as DES. This design model can have invertible, non-invertible, and self-invertible components. Additionally, the Feistel block cipher uses the same encryption and decryption algorithms.

The Feistel structure is based on the Shannon structure proposed in 1945, demonstrating the confusion and diffusion implementation processes. Confusion produces a complex relationship between the ciphertext and encryption key, which is done by using a substitution algorithm. On the other hand, diffusion creates a complex relationship between plain text and cipher text by using a permutation algorithm.

The Feistel cipher proposed the structure that implements substitution and permutation alternately. Substitution replaces plain text elements with ciphertext. Permutation changes the order of the plain text elements rather than being replaced by another element as done with substitution.

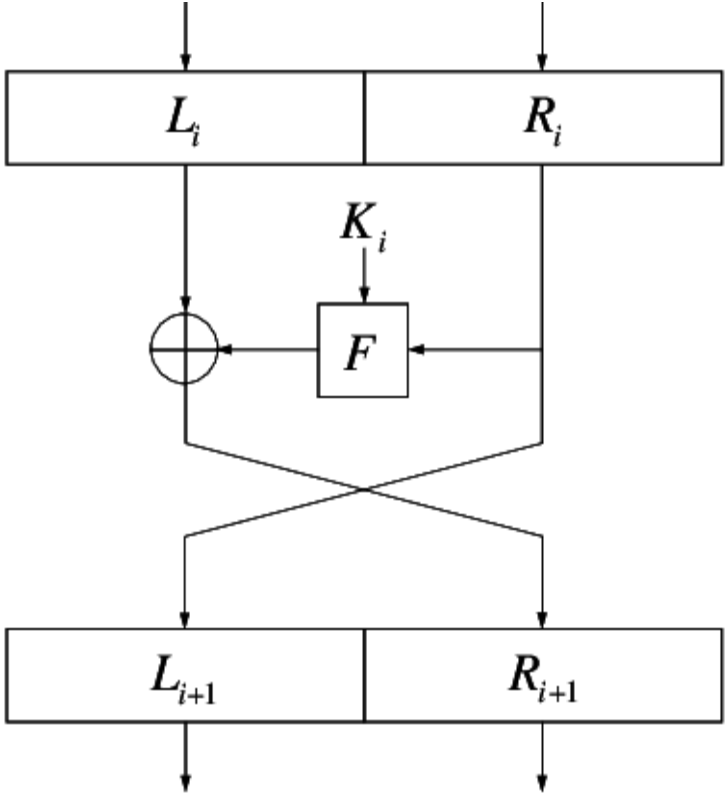
The Feistel cipher encryption process involves numerous rounds of processing plain text. Each round includes the substitution step and then the permutation step.

Mã khối Feistel

- Bản rõ sẽ được biến đổi qua một số vòng để cho ra bản mã cuối cùng
- Trong đó bản rõ P và các bản mã C_i được chia thành nửa trái và nửa phải:

$$P = (L_0, R_0)$$

$$C_i = (L_i, R_i)\ i = 1, 2, \ldots n$$



Mã khối Feistel

- Đầu tiên chia khối plaintext thành hai phần bằng nhau, nửa trái kí hiệu L_0, nửa phải kí hiệu R_0.
- Lặp qua n rounds (i từ 1 đến n) để mã hóa, các round sử dụng chung hàm mã hóa F nhưng khác sub-key K_i (được sinh từ bộ sinh khóa). Thực hiện các bước sau đây:

$$L_{i+1} = R_i$$

$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

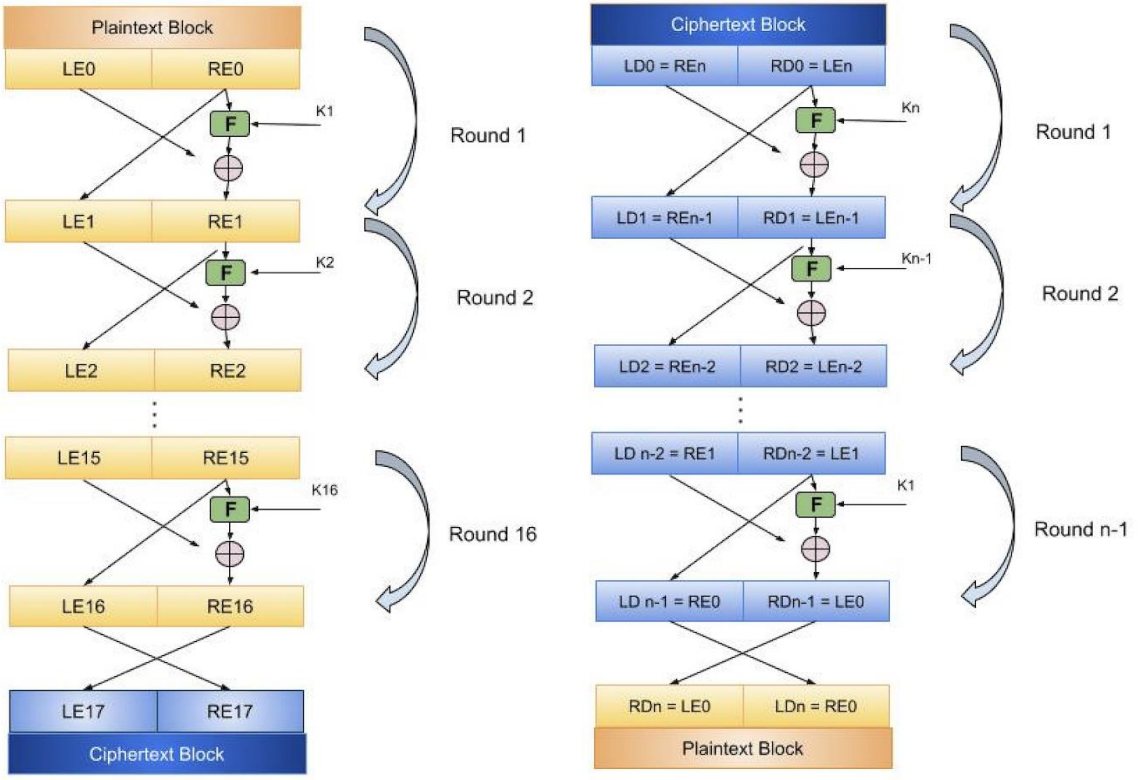
- Sau n rounds, thực hiện hoán vị L_n, R_n và gộp lại để thu giá trị mã hóa: R_nL_n.

THUẬT TOÁN MÃ KHỐI

Thuật toán Feistel Cipher:

1. Chia khối dữ liệu
2. Áp dụng hàm Feistel
3. Hoán đổi
4. Lặp lại

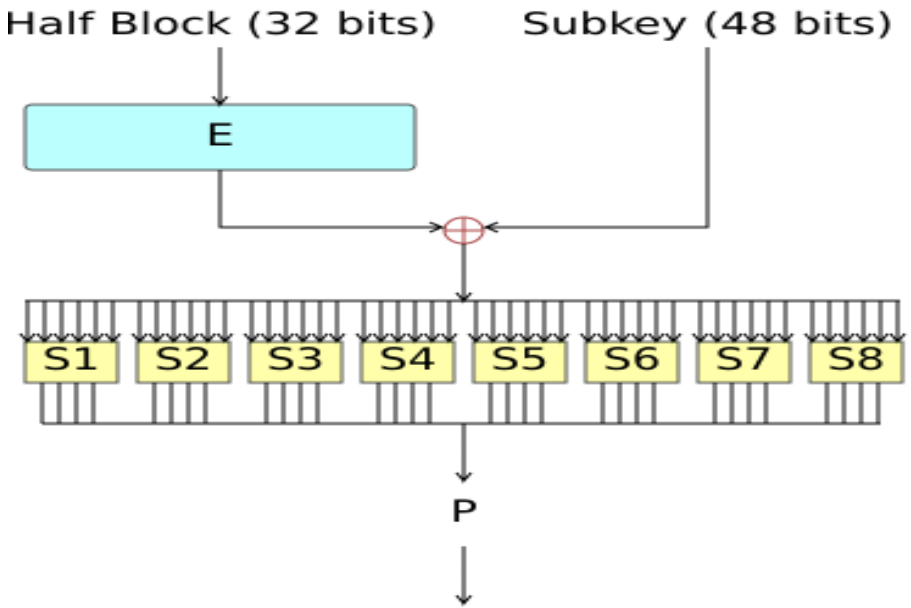
Ví dụ: DES



https://www.tutorialspoint.com/cryptography/cryptography_feistel_block_cipher.htm

Đặc điểm Feistel

- Hàm F đóng vai trò như là phép thay thế còn việc hoán đổi các nửa trái phải có vai trò hoán vị.
- Độ an toàn mã Feistel tùy thuộc vào:
  - Kích cỡ khối (Block size)
  - Kích cỡ của khóa (Key size)



Đặc điểm Feistel

- **Số dòng (Number of rounds):** Bản chất thuật toán mã Feistel là một dòng duy nhất là đã cung cấp tính an toàn nhưng nếu số vòng càng tăng thì tính an toàn càng cao. (thông thường 16 vòng).
- **Thuật toán phát sinh khóa con (Subkey generation algorithm):** Thuật toán càng phức tạp thì sẽ khó khăn hơn trong việc thám mã.
- **Hàm vòng F (Round function F):** Càng phức tạp thì đề kháng càng cao đối với thám mã.

Tổng kết

- Mã đối xứng
- Mã thay thế
- Mã chuyển vị
- Mã dòng
- Mã khối
- Mã Feistel

Thực hành

Thực hành bài Lab 3

Thực hành

Sinh viên cài đặt thực thi các chương trình sau dùng C++:

Bài 1. Mã hóa thay thế. Cài đặt chương trình C++ thực hiện mã hóa thay thế có các menu thực hiện các chức năng:

- Chọn mã hóa thay thế cho tiếng Việt và tiếng Anh
- Nhập khóa K bất kỳ
- Chọn mã hóa hoặc giải mã với xâu bất kỳ nhập vào từ bàn phím

Thực hành

Bài 6. Cài đặt thuật toán Feistel

Viết chương trình C++ để cài đặt thuật toán mã hóa khối Feistel theo các bước sau:

- Nhập một khối dữ liệu dạng chuỗi nhị phân.
- Chia thành hai nửa: L (left) và R (right).
- Thực hiện n vòng Feistel, mỗi vòng có:

$$L_{i+1} = R_i$$

$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

- Hàm F là một phép XOR
- Khóa vòng (subkey) được sinh từ một khóa chính
- Hiển thị kết quả của từng vòng.
- Giải mã dữ liệu bằng cách đảo ngược các bước.

ĐẠI NAM
UNIVERSITY

Thank You