

AN TOÀN & BẢO MẬT THÔNG TIN

Giảng viên: Ths Phạm Thanh Bình
Bộ môn Kỹ thuật máy tính & mạng
<http://dhthuyloi.blogspot.com>

Chương 2:

MẬT MÃ ĐỐI XỨNG *(tiếp)*

- ◆ Những vấn đề cơ bản của mật mã
- ◆ Các kỹ thuật mã hoá cổ điển
- ◆ Chuẩn mã hoá dữ liệu DES
- ◆ Chuẩn mã hoá cải tiến AES

Mã hoá đa kí tự

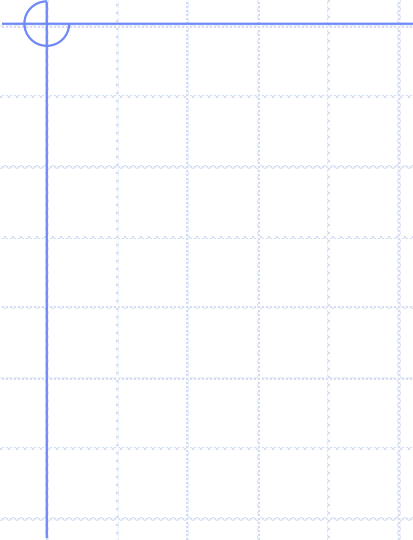
Ví dụ:

- ◆ Mật mã Playfair
- ◆ Mật mã Hill

Mật mã Playfair

- ◆ Được phát minh bởi nhà khoa học người Anh - Charles Wheatstone - năm 1854
- ◆ Được sử dụng rộng rãi trong quân đội Anh, Mỹ và đồng minh trong chiến tranh thế giới thứ II

- ◆ Mật mã Playfair sẽ thay thế từng cặp 2 kí tự trong bản rõ bởi 2 kí tự tương ứng trong ma trận khoá 5 x 5.
- ◆ Ví dụ, nếu chọn từ khoá là *monarchy* thì ma trận khoá sẽ như sau:



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

- ◆ Lần lượt viết từng kí tự của khóa vào ma trận, từ trái sang phải, từ trên xuống dưới, bỏ các kí tự trùng lặp
- ◆ Viết các ký tự còn lại trong bảng chữ cái vào ma trận theo thứ tự, I và J được coi như một ký tự

- ◆ Mỗi ký tự trong cặp plaintext sẽ được mã hoá bằng ký tự nằm cùng hàng với nó, nhưng cùng cột với ký tự kia
- ◆ Giả sử trong plaintext có cặp ký tự HS, nó sẽ được thay thế bởi cặp BP

- ◆ Nếu cặp ký tự plaintext rơi vào cùng một hàng của ma trận thì mỗi ký tự được thay thế bởi ký tự bên phải nó.
- ◆ Nếu ký tự plaintext rơi vào cột cuối cùng, thì ciphertext của nó là ký tự cùng hàng ở cột đầu tiên.
- ◆ Ví dụ, AR sẽ được mã hóa thành RM

- ◆ Nếu cặp ký tự plaintext rơi vào chung một cột của ma trận thì mỗi ký tự được thay thế bởi ký tự ngay sát dưới.
- ◆ Nếu ký tự plaintext rơi vào hàng cuối cùng, thì ciphertext của nó là ký tự cùng cột, ở hàng đầu tiên.
- ◆ Ví dụ, MU được mã hóa thành CM.

- ◆ Nếu hai kí tự trong plaintext giống nhau thì chúng sẽ được cách ly bằng một ký tự đại diện, chẳng hạn là **x**.
- ◆ Ví dụ, từ **balloon** sẽ được tách ra thành **ba
lx lo on**.

Bài tập: *(làm bằng tay)*

- ◆ Chọn một từ khoá bất kì rồi xây dựng ma trận khoá Playfair
- ◆ Chọn một plaintext bất kì, áp dụng ma trận khoá trên để tạo ra ciphertext.
- ◆ Thử giải mã ciphertext rồi so sánh với plaintext ban đầu.

Bài tập:

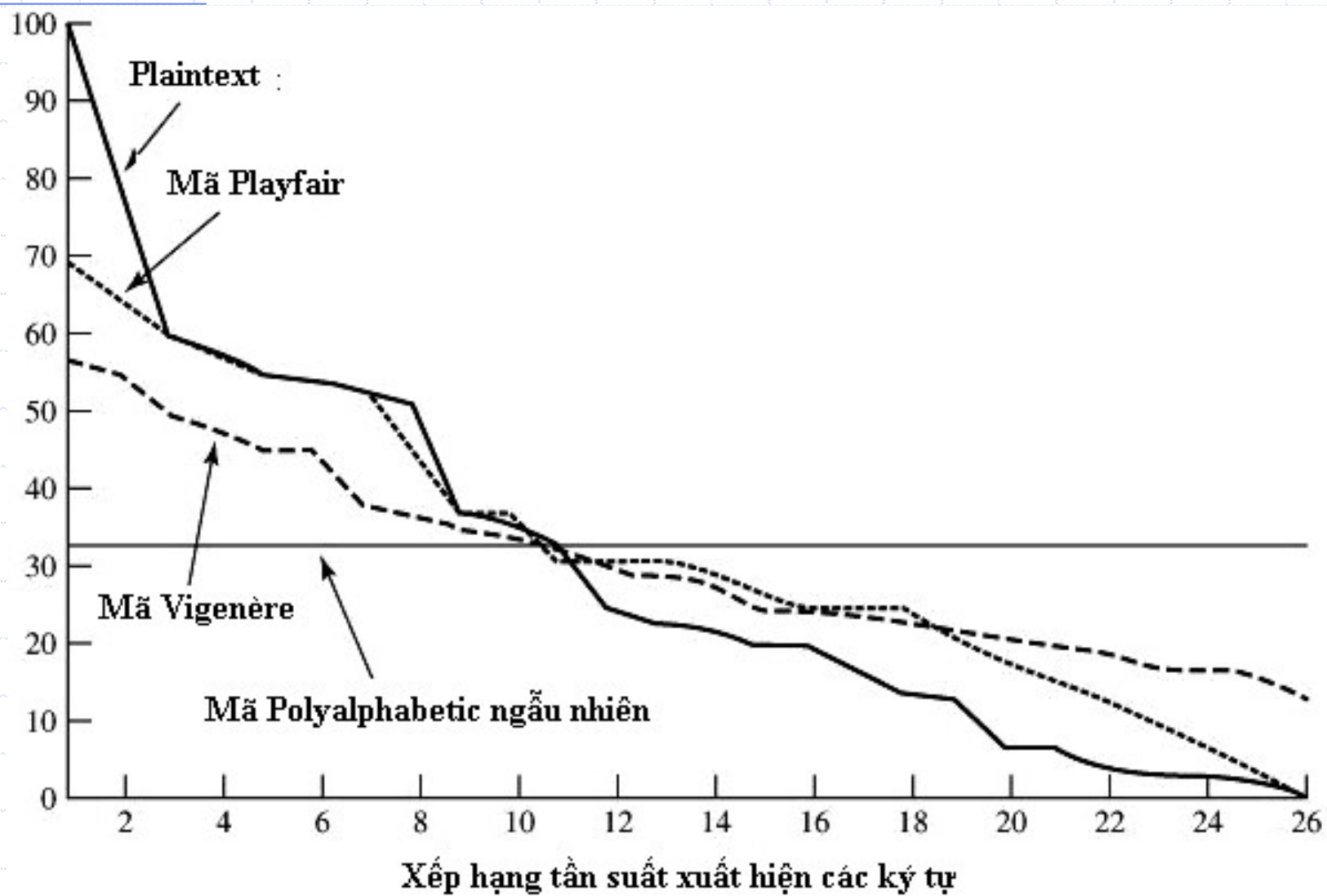
1. Nhập một từ khóa bất kì. Hãy loại bỏ các kí tự trùng lặp trong chuỗi.
2. Điền thêm vào chuỗi trên những kí tự còn lại trong bảng chữ cái
3. Đặt chuỗi thu được vào ma trận khóa 5x5

Bài tập

- ◆ Lập trình mã hoá và giải mã thông điệp theo thuật toán Playfair

Nhận xét

- ◆ Mật mã Playfair có không gian khoá lớn tương tự mật mã Monoalphabetic nên khó bẻ được khoá bằng phương pháp Brute - force
- ◆ Mật mã Playfair có khả năng che giấu một phần thông tin về tần suất xuất hiện các chữ cái, nhờ thực hiện mã hoá từng cặp hai kí tự



Mật mã Hill

- ◆ Được phát minh bởi nhà toán học Lester Hill vào năm 1929

- ◆ Mật mã Hill sẽ thay thế từng nhóm m kí tự trong plaintext bởi m kí tự ciphertext
- ◆ m kí tự ciphertext được xác định bởi hệ m phương trình tuyến tính sau:

$$C_1 = (k_{11}P_1 + k_{12}P_2 + \dots + k_{1m}P_m) \bmod 26$$

$$C_2 = (k_{21}P_1 + k_{22}P_2 + \dots + k_{2m}P_m) \bmod 26$$

$$C_3 = (k_{31}P_1 + k_{32}P_2 + \dots + k_{3m}P_m) \bmod 26$$

...

$$C_m = (k_{m1}P_1 + k_{m2}P_2 + \dots + k_{mm}P_m) \bmod 26$$

Biểu diễn dưới dạng ma trận (với $m = 3$)

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \bmod 26$$

hay $\mathbf{C} = \mathbf{KP} \bmod 26$

Trong đó:

- ◆ C là ma trận cột biểu diễn ciphertext
- ◆ P là ma trận cột biểu diễn plaintext
- ◆ K là ma trận khoá

khi đó $\mathbf{P} = \mathbf{K}^{-1}\mathbf{C} \bmod 26$

với:

- ◆ \mathbf{K}^{-1} là ma trận nghịch đảo của ma trận khoá \mathbf{K}
- ◆ tức là $\mathbf{K} \cdot \mathbf{K}^{-1} = \mathbf{K}^{-1} \cdot \mathbf{K} = \mathbf{I}$ (\mathbf{I} là ma trận đơn vị)

Ví dụ:

◆ Giả sử chọn ma trận khoá K như sau:

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

◆ Khi đó ma trận nghịch đảo sẽ là:

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

◆ Vì:

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Bài tập: (làm bằng tay)

- ◆ Với plaintext là "*paymoremoney*" và sử dụng với mật khóa K nói trên, hãy xác định ciphertext.
- ◆ Giải mã ciphertext thu được bằng cách nhân nó với K^{-1} , rồi so sánh kết quả với plaintext ban đầu

Nhận xét:

- ◆ Độ an toàn của mật mã Hill sẽ càng lớn khi sử dụng ma trận K càng lớn
- ◆ Mật mã Hill có khả năng che dấu hoàn toàn tần suất xuất hiện các kí tự đơn
- ◆ Mật mã Hill rất mạnh khi chống lại tấn công chỉ biết ciphertext, nhưng nó lại dễ dàng bị bẻ gãy với một tấn công biết plaintext, do có thể dễ dàng xác định ma trận K từ các cặp P-C đã biết.

Bài tập

- ◆ Lập trình mã hoá và giải mã thông điệp theo thuật toán Hill

Các mật mã Polyalphabetic

- ◆ Mật mã Monoalphabetic chỉ sử dụng một bảng mã (mỗi kí tự plain text được thay thế bởi một kí tự cố định), nên không giấu được tần suất xuất hiện các kí tự
- ◆ Còn mật mã Polyalphabetic lại sử dụng nhiều bảng mã khác nhau (mỗi kí tự plain text có thể được thay thế bởi nhiều kí tự khác nhau, dựa trên các khoá thay thế khác nhau)

Ví dụ: Mật mã Vigenère

- ◆ Mật mã Vigenère sẽ thay thế từng nhóm m kí tự trong plaintext bởi m kí tự ciphertext
- ◆ m kí tự ciphertext được xác định bởi hệ m phương trình sau:

$$C_1 = (P_1 + k_1) \bmod 26$$

$$C_2 = (P_2 + k_2) \bmod 26$$

$$C_3 = (P_3 + k_3) \bmod 26$$

...

$$C_m = (P_m + k_m) \bmod 26$$

Trong đó C_i là các kí tự ciphertext, P_i là các kí tự plaintext, k_i là các giá trị của khoá

Công thức giải mã như sau:

$$P_1 = (C_1 - k_1) \bmod 26$$

$$P_2 = (C_2 - k_2) \bmod 26$$

$$P_3 = (C_3 - k_3) \bmod 26$$

...

$$P_m = (C_m - k_m) \bmod 26$$

◆ Do k_1, k_2, \dots, k_m là các số nguyên thuộc Z_{26} , và các kí tự ‘A’, ‘B’, \dots , ‘Z’ cũng tương ứng với các số nguyên thuộc Z_{26} , nên có thể viết khoá K dưới dạng một dãy m chữ cái cho dễ nhớ

Ví dụ:

- ◆ Giả sử lấy $m=6$ và khoá K là CIPHER, plaintext là:

WEWILLMEETATMIDNIGHT

Hãy xác định ciphertext.

Giải:

- Do $m=6$, ta sẽ tách plaintext thành từng nhóm 6 kí tự:

WEWILL/MEETAT/MIDNIG/HT

Viết theo dạng số là:

22 4 22 8 11 11 / 12 4 4 19 0 19 / 12 8 3 13 8 6 / 7 19

- Từ khoá CIPHER tương ứng với:

$$K = (2, 8, 15, 7, 4, 17)$$

◆ Cộng từng nhóm 6 kí tự của plaintext với K ta có:

22	4	22	8	11	11/	12	4	4	19	0	19/	12	8	3	13	8	6/	7	19
2	8	15	7	4	17/	2	8	15	7	4	17/	2	8	15	7	4	17/	2	8
<hr/>																			
24	12	11	15	15	2 /	14	12	19	0	4	10/	14	16	18	20	12	23/	9	1

◆ Ciphertext tương ứng là:

YMLPPCOMTAEKOQSUMXJB

Bài tập: *(làm bằng tay)*

- ◆ Chọn một plaintext và một khoá K bất kì
- ◆ Áp dụng thuật toán Vigenère để xác định ciphertext

Gợi ý:

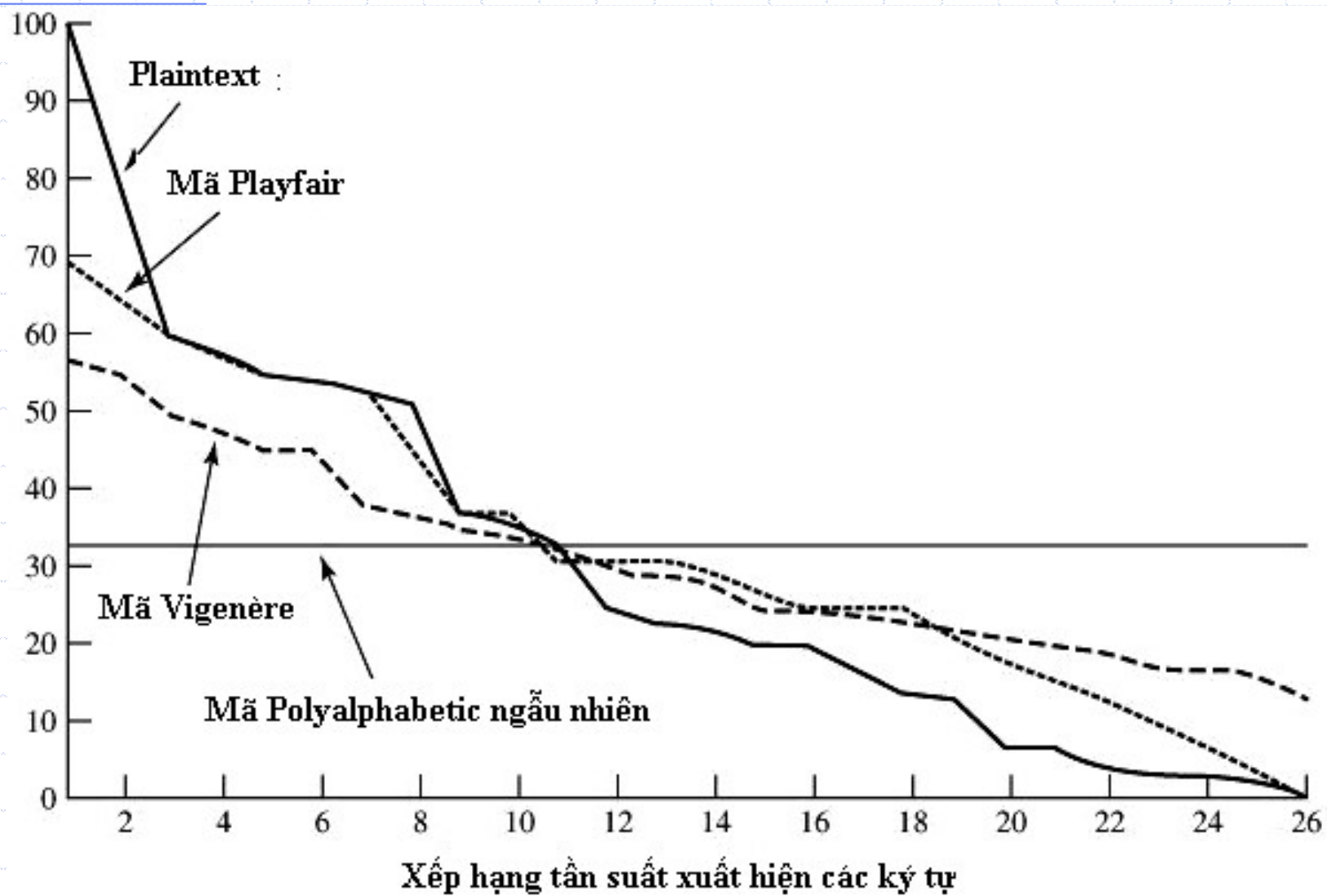
Có thể sử dụng cách tra bảng ở trang sau

Plaintext

	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

Nhận xét:

- ◆ Như vậy, nếu $k_1 = k_2 = \dots = k_m$ thì mật mã Vigenère sẽ trở thành mật mã Caesar tổng quát.
- ◆ Khi $k_1 \neq k_2 \neq \dots \neq k_m$: một kí tự plaintext có thể được thay thế bởi nhiều kí tự khác nhau (ứng với các k khác nhau), nhờ vậy có thể che giấu được tần suất xuất hiện các kí tự.



Bài tập:

- ◆ *Lập trình nhập chuỗi kí tự plaintext từ bàn phím, mã hoá chuỗi bằng thuật toán Vigenère với khoá K (là một chuỗi kí tự) nhập từ bàn phím. Hiện chuỗi mới ra màn hình.*
- ◆ *Lập trình giải mã để khôi phục lại chuỗi ban đầu.*

Nhận xét:

- ◆ Mật mã Vigenère vẫn không giấu được hoàn toàn tần suất xuất hiện các kí tự, và vẫn có thể bị phân tích
- ◆ Giải pháp khắc phục là sử dụng hệ thống biểu diễn thông tin không mang tính thống kê của ngôn ngữ - đó là hệ nhị phân

Mật mã Vernman

- ◆ Được phát minh bởi kỹ sư Gilbert Vernman của AT&T - năm 1918
- ◆ Hệ thống này làm việc trên dữ liệu nhị phân chứ không phải các ký tự.

- ◆ Plaintext được biểu diễn dưới dạng một chuỗi bit nhị phân
- ◆ Khóa K cũng được biểu diễn dưới dạng một chuỗi bit nhị phân (càng dài càng tốt, càng ngẫu nhiên càng tốt)
- ◆ Ciphertext được sinh ra bởi phép XOR giữa plaintext với khóa K

◆ Công thức mã hoá:

$$c_i = p_i \oplus k_i$$

◆ Công thức giải mã:

$$p_i = c_i \oplus k_i$$

Trong đó:

p_i = bit thứ i của plaintext

k_i = bit thứ i của mật khóa K

\oplus là phép XOR

Ví dụ 1:

◆ Mã hoá:

Plaintext = 0001 0110 ...

Khoá K = 0000 1101 ...

Ciphertext = 0001 1011 ...

◆ Giải mã:

Ciphertext = 0001 1011 ...

Khoá K = 0000 1101 ...

Plaintext = 0001 0110 ...

Ví dụ 2: (làm bằng tay)

- ◆ Giả sử plaintext là “HA NOI” (các kí tự trong chuỗi có thể được biểu diễn dưới dạng nhị phân theo bảng mã chuẩn ASCII), khóa K là một dãy nhị phân 8 bit như sau:

$$K = 10010011$$

- ◆ Hãy mã hóa chuỗi ban đầu bằng phương pháp Vernman.
- ◆ Giải mã chuỗi thu được rồi so sánh với chuỗi ban đầu.

Nhận xét:

- ◆ Với khoá K đủ dài và ngẫu nhiên, các thông tin mang tính thống kê của ngôn ngữ có thể được che giấu hoàn toàn
- ◆ Nếu K ngắn thì sẽ phải sử dụng K lặp đi lặp lại, làm giảm tính ngẫu nhiên, và có thể làm lộ một phần thông tin về thống kê tần suất.
- ◆ Tuy nhiên, việc sinh ra được một khoá K dài và thực sự ngẫu nhiên như vậy sẽ đòi hỏi nhiều công sức.
- ◆ Sự ra đời của mật mã hệ nhị phân là tiền đề cho sự ra đời của các mật mã hiện đại.

Bài tập 1:

- ◆ Nhập một chuỗi plaintext từ bàn phím
- ◆ Nhập một khóa K (dài 8 bit)
- ◆ Mã hóa chuỗi ban đầu bằng cách XOR các kí tự của nó với K
- ◆ Giải mã ciphertext thu được, rồi so sánh với chuỗi ban đầu.

Gợi ý: Một số phép toán thao tác với bit trong C

Phép toán	Trong C	Ví dụ
AND	&	a & b
OR		a b
XOR	^	a ^ b
NOT	~	~a
Dịch trái	<<	a << 4
Dịch phải	>>	a >> 4

Bài tập 2:

- ◆ Phân tích ưu nhược điểm của phương pháp mã hóa ở bài tập 1.
- ◆ Hãy đưa ra các giải pháp để khắc phục nhược điểm đó.

Kỹ thuật chuyển dịch - hoán vị

- ◆ Ngoài Kỹ thuật thay thế, các mật mã cổ điển còn sử dụng Kỹ thuật chuyển dịch - hoán vị
- ◆ Các kí tự của plaintext sẽ được hoán đổi vị trí cho nhau để tạo thành ciphertext

Ví dụ 1: Kỹ thuật rain-fence

- ◆ Plaintext được viết dịch xuống tuần tự theo các đường chéo rồi đọc trình tự theo các hàng.
- ◆ Giả sử plaintext là "*meet me after the toga party*", cách viết như sau:

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

◆ Ciphertext thu được là:
MEMATRHTGPRYETEFETEOAAT

Ví dụ 2:

- ◆ Viết plaintext trong một hình chữ nhật theo từng hàng, rồi đọc thông điệp theo từng cột, nhưng hoán đổi trật tự cột.
- ◆ Trật tự của các cột sẽ trở thành khóa cho thuật toán

◆ Giả sử plaintext là "*attack postponed until two am*", cách viết như sau:

Key: 4 3 1 2 5 6 7

Plaintext: a t t a c k p
 o s t p o n e
 d u n t i l t
 w o a m x y z

Ciphertext:

TTNAAPTMTSUOAODWCOIXKNLYPETZ

Nhận xét:

- ◆ Mật mã hoán vị thuần túy rất dễ nhận ra bởi nó giữ nguyên tần suất xuất hiện ký tự đơn (và làm thay đổi tần suất của các cặp, các bộ ký tự của plaintext)
- ◆ Để tăng độ phức tạp, người ta có thể tiến hành đổi chỗ nhiều lần, hoặc kết hợp với các thuật toán mã hoá khác.

Bài tập:

- ◆ Nhập một chuỗi plaintext
- ◆ Mã hóa chuỗi bằng kỹ thuật rain-fence, hiện ciphertext ra màn hình
- ◆ Giải mã ciphertext, so sánh kết quả với chuỗi ban đầu.

Bài tập:

- ◆ Nhập một chuỗi plaintext (dài không quá 25 kí tự)
- ◆ Sắp xếp các kí tự của chuỗi vào một ma trận 5×5 (lần lượt theo từng hàng)
- ◆ Đọc các phần tử của ma trận theo từng cột để tạo thành ciphertext
- ◆ Giải mã ciphertext, so sánh kết quả với chuỗi ban đầu.

Hết Phần 2_2