Chương 6

VÀI ỨNG DỤNG VÀO LÍ THUYẾT MẬT MÃ

Cho đến khoảng cuối những năm 70, số học vẫn được xem là một trong những ngành lí thuyết thuần tuý nhất của toán học, vì hầu như không có ứng dụng thực tế. Quan niệm đó thay đổi hẳn sau khi số học được áp dụng để xây dựng những hệ mật mã khoá công khai. Các lí thuyết mới của số học, đặc biệt là số học thuật toán, tìm thấy những ứng dụng trực tiếp vào thực tiễn. Vì thế chúng tôi dành một chương trình bày những điểm cơ bản của lí thuyết mật mã, để qua đó, độc giả có thể thấy được vai trò quan trọng của những vấn đề xét đến trong lí thuyết số thuật toán, như vấn đề độ phức tạp của các thuật toán phân tích một số nguyên dương ra thừa số, hay vấn đề kiểm tra nguyên tố.

§1. Mã Ceasar.

Có thể nói, mật mã đã có từ thời cổ đại. Người ta cho rằng, người đầu tiên áp dụng mật mã một cách có hệ thống để đảm bảo bí mật thông tin quân sự là nhà quân sự thiên tài của La Mã cổ đại, Julius Ceasar. Sự phát triển của xã hội dẫn đến việc ngày nay mật mã không những chỉ được dùng trong bí mật quân sự và ngoại giao, mà còn dùng, và có thể chủ yếu là dùng trong bí mật kinh tế, thương mại. Vì thế xuất hiện những đòi hỏi mới đối với các hệ mật mã hiện đại, khác về nguyên tắc so với mật mã thường dùng trước đây. Khác với hoạt động quân sự hoặc ngoại giao, trong hoạt động kinh doanh, số lượng đơn vị phải cùng trao đổi thông tin thường là rất lớn. Thậm chí, những người chưa hề quen biết nhau cũng có nhu cầu trao đổi những thông tin mật với nhau. Bởi thế, những hệ thống mật mã xây dựng theo nguyên tắc cũ khó có thể thích hợp: trong các hệ mã đó, khi đã biết khoá lập mã, ta dễ dàng tìm ra khoá giải mã. Hiển nhiên, muốn gửi một thông báo mật cho một đối tượng nào đó, ta cần phải biết khoá lập mã của họ, vì thế, những người cùng dùng một hệ mã đều biết hết bí mật của nhau. Khi một bí mật có quá nhiều người biết thì không còn là bí mật nữa. Các hệ thống mật mã hiện đại, *mật mã khoá công khai*, khắc phục được những nhược điểm đó: mỗi người tham gia trong hệ thống chỉ cần giữ bí mật khoá giải mã của mình, trong khi khoá lập mã được thông báo công khai. Việc biết khoá lập mã không cho phép tìm ra khoá giả mã trong một thời gian chấp nhận được, ngay cả khi sử dụng những máy tính hiện đại nhất. Những mật mã khoá công khai tìm thấy đầu tiên là những mật mã dùng hàm số học.

Có một điều hết sức thú vị là, nói cho cùng, những hệ mật mã hiện đại cũng chỉ là sự cải tiến mật mã của Ceasar! Vì thế chúng tôi bắt đầu việc trình bày mật mã Ceasar.

Trước hết chúng ta cần thống nhất một số danh từ.

Văn bản tức là thông báo cần chuyển, được viết bằng ngôn ngữ thông thường. Ở đây, ta sẽ xem các văn bản đều được viết bằng tiếng Việt.

Việc chuyển thông báo đó thành dạng mật mã được gọi là *mã hóa*.

Bản đã mã hoá của văn bản được gọi là *văn bản mật*.

Giải mã tức là chuyển một văn bản mật thành văn bản ban đầu.

Ceasar chuyển thông báo mật bằng cách sau đây. Trước tiên, lập tương ứng mỗi chữ cái với một số. Nhờ bảng tương ứng đó, ta có thể chuyển một văn bản thành dạng chữ số. Sau đó ta cộng thêm 3 vào mỗi chữ số nhận được. Lại nhờ bảng tương ứng giữa chữ và số, ta biến bảng chữ số mới này về dạng chữ viết. Như vậy ta nhận được một văn bản mật cần chuyển đi. Đây là quá trình mã hoá.

Khi nhận được văn bản mật, ta giải mã bằng cách biến nó thành dạng chữ số nhờ bảng tương ứng giữa chữ và số, sau đó trừ đi 3 ở mỗi chữ số và lại chuyển nó về dạng chữ để lại có văn bản ban đầu.

Chú ý rằng khi phép cộng hoặc trừ đi 3 đưa ta vượt khỏi giới hạn của bảng tương ứng, ta thay số đó bằng thặng dư dương bé nhất modulo số các phần tử của bảng tương ứng giữa chữ và số.

Sau đây ta sẽ xét trên một ví dụ cụ thể.

Trước hết ta lập tương ứng các chữ cái với các số theo bảng sau:

a	ă	â	b	c	d	đ	e	ê	g	h
1	2	3	4	5	6	7	8	9	10	11

i	k	l	m	n	o	ô	ơ	p	q
12	13	14	15	16	17	18	19	20	21

r	s	t	u	ư	v	x	y
22	23	24	25	26	27	28	29

Bảng 1

Dĩ nhiên ta có thể thêm các số để chỉ dấu, nhưng để đơn giản , ở đây ta tạm thời viết các văn bản không dấu.

Như vậy mã Ceasar được thành lập theo công thức sau:

$$C \equiv P+3 (\text{mod } 29) \qquad (6.1)$$

trong đó P là chữ số trong văn bản, còn C là chữ số tương ứng trong văn bản mật. Chẳng hạn ta muốn mã hoá thông báo sau đây:

LY THUYÊT MÂT MA KHÔNG CO GI KHO

Trước hết nhằm nâng cao tính bảo mật, ta tách thông báo thành từng nhóm 5 chữ cái, để tránh việc một số từ của thông báo dễ bị phát hiện căn cứ vào số chữ cái. Như vậy thông báo cần mã hoá là:

LYTHU YÊTMÂ TMAKH ÔNGCO GIKHO

Nhờ bảng 1, ta chuyển thông báo thành dạng chữ số:

14 29 24 11 25 29 8 24 15 1 24 15 1 13 11 18 16 10 5 18 10 12 13 11 18

Ap dung công thức (6.1), bảng chữ số trên được chuyển thành:

17 3 27 14 28 3 11 27 18 3 27 18 4 16 14 21 19 13 8 21 13 15 16 14 21

Để có văn bản mật, ta chỉ cần chuyển lại thành dạng chữ cái theo Bảng 1:

OÂVLU ÂHVÔÂ VÔBNL QƠKEQ KMNLQ

Số 3 trong công thức (6.1) được gọi là khoá của mã Ceasar, vì nó được dùng để mã hoá cũng như giải mã.

Ta cũng có thể lập một hệ mật mã mới bằng cách thay số 3 trong công thức (6.1) bằng một số *k* tuỳ ý khác giữa 1 và 29:

$$C \equiv P+k \pmod{29} \qquad (6.2)$$

Trong trường hợp này, khoá của mã là *k*. Việc mã hoá và giải mã được tiến hành hoàn toàn tương tự như trên.

Ta có thể lập mã tổng quát hơn chút ít bằng cách thay công thức (6.2) bởi công thức sau đây:

$$C \equiv aP+b \pmod{29},$$

trong đó *a, b* là các số nguyên, và (a, 29)=1. Những mã như vậy được gọi là *mã biến đổi aphin*. Việc giải mã được tiến hành bằng cách giải phương trình đồng dư (6.2), khi đã biết *c, a, b*.

Phân tích sau đây cho thấy tính bảo mật của mã Ceasar là không cao. Khi bắt được một văn bản mật, người ta có thể dựa vào tần suất xuất hiện của các chữ cái để đoán ra khoá của mã. Chẳng hạn nếu chữ *a* nói chung xuất hiện nhiều nhất trong các văn bản thì chữ cái nào có mặt nhiều nhất trong văn bản mật có nhiều khả năng là chữ *a,* từ đó đoán ra khoá. Hơn nữa, chỉ có 29 cách khác nhau để chọn khoá cho loại mã nói trên, nên để dàng tìm ra khoá của mã, nhất là khi áp dụng máy tính. Đối với mã biến đổi aphin, chỉ cần dựa vào tần suất xuất hiện từ để tìm ra hai chữ cái tương ứng với 2 chữ nào đó trong văn bản mật, ta có thể tìm ra *a, b* bằng cách giải hệ hai phương trình đồng dư. Ngoài ra, việc giải những hệ mã biến đổi aphin cũng quá dễ dàng đối với máy tính.

Như vậy, với những yêu cầu về bảo mật cao hơn, người ta phải dùng những hệ mật mã phức tạp hơn. Sau đây là một vài hệ mã thường dùng, từ đơn giản đến phức tạp.

§2. Mã khối.

Mã khối xuất hiện nhằm chống lại việc sử dụng tần suất xuất hiện của các chữ cái trong văn bản để dò ra khoá giải mã. Khác với các hệ mã trình bày ở mục trên, ta không mã hoá từng chữ cái của văn bản, mà mã hoá từng khối chữ cái. Trước tiên ta xét trường hợp mã khối 2 chữ. Để dễ hiểu ta xét ví dụ sau đây.

Giả sử thông báo cần mã hoá là

KHÔNG CO ĐIÊU BI MÂT NAO GIU ĐƯƠC LÂU

Trước hết ta tách thông báo trên thành khối hai chữ:

KH ÔN GC OĐ ÊU BI MÂ TN AO GI UĐ ƯƠ CL ÂU

Sau đó các chữ cái được chuyển thành các chữ số tương ứng:

13 11 18 15 10 5 17 7 9 25 4 12 16 3 24 15 1 17 1012 26 19 5 14 3 25

Với mỗi mã khối hai chữ, ta chọn một ma trận cấp hai làm khoá của mã. Chẳng hạn ma trận

$$A=\begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix}$$

Khi đó các khối hai chữ số P_1P_2 trong văn bản được chuyển thành các khối hai chữ số C_1C_2 trong văn bản mật theo công thức sau đây:

$$C_1 \equiv 23P_1+11P_2 (\text{mod } 29)$$

$$C_2 \equiv 9P_1+12P_2 (\text{mod } 29) \qquad (6.3)$$

Như vậy thông báo trên đây đã được chuyển thành:

14 17 28 23 24 5 4 5 18 4 21 6 21 19 7 10 14 2 24 27 8 10 25 8

Trở lại các chữ cái tương ứng, ta được văn bản mật:

LO XS TC BC ÔB QD TD QƠ ĐG LI TV EG UE

Để giải mã, ta cần giải hệ phương trình đồng dư (6.3) để tìm P_1,P_2. Điều đó thực hiện được nhờ định lí sau đây:

Định lí 6.1. *Cho hệ phương trình đồng dư*

$$ax+by \equiv r(\text{mod } m)$$

$$cx+dy \equiv s(\text{mod } m)$$

Đặt $\Delta = ad-bc \ (\text{mod } m)$. *Khi đó, nếu* $(\Delta, m)=1$ *thì hệ phương trình đang xét tồn tại nghiệm duy nhất modulo m, cho bởi công thức sau:*

$$x \equiv \Delta^{-1}(dr-bs) \ (\text{mod } m),$$

$$y \equiv \Delta^{-1}(as-cr) \ (\text{mod } m),$$

trong đó Δ^{-1} *là nghịch đảo của* Δ *modulo m.*

Định lí trên đây được chứng minh hoàn toàn tương tự như trong đại số tuyến tính (chỉ cần thay điều kiện $(\Delta, m)=1$ bởi điều kiện $\Delta \neq 0$).

Trong ví dụ trên đây, $\Delta \equiv 3 (mod\ 29)$, $\Delta^{-1} \equiv 10 (mod\ 29)$. Như vậy, khi có khối C_1C_2 trong văn bản mật và đã biết mã khoá là ma trận A, ta tìm được khối chữ tương ứng trong văn bản là P_1P_2 theo công thức sau:

$$P_1=10(12C_1-11C_2) \equiv 4C_1+6C_2 (\text{mod } 29)$$

$$P_2=10(23C_2-9C_1) \equiv 26C_1+27C_2 (\text{mod } 29).$$

Tóm lại, việc mã hoá và giải mã được tiến hành nhờ các công thức:

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}; \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 26 & 27 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

Như vậy, việc sử dụng mã khối đã nâng cao rất nhiều tính bảo mật. Tuy vậy, khoá của mã vẫn có thể bị khám phá nhờ việc nghiên cứu tần suất xuất hiện của các khối chữ cái. Chẳng hạn nếu ta dùng mã khối hai chữ, thì có cả thảy $29^2=641$ khối trong tiếng Việt, và như vậy vẫn còn khả năng khám phá ra khoá của mã nhờ các máy tính hiện đại. Trong trường hợp ta sử dụng mã khối với những khối nhiều chữ cái, việc tìm ra khoá bằng tần suất các khối chữ trên thực tế là không sử dụng được: Chẳng hạn khi dùng mã khối 10 chữ cái, số khối chữ sẽ là 29^{10}, vượt quá khả năng thăm dò tần suất xuất hiện các khối chữ đó trong ngôn ngữ.

Các mã khối n chữ cái được lập và giải hoàn toàn tương tự như trên, trong đó các ma trận C,P là các ma trận n cột, A là ma trận vuông cấp n. Ma trận nghịch đảo của A tồn tại khi định thức của A nguyên tố cùng nhau với 29, và ma trận P sẽ được tính bằng quy tắc Kramer như trong đại số tuyến tính (chỉ cần thay dấu = bởi $\equiv (mod\ 29)$).

Sau đây ta trình bày một loại hệ mã mới, một mặt nó có tính bảo mật rất cao, Mặt khác là cơ sở cho những hệ mã hoàn toàn mới: các hệ mã khoá công khai.

§3. Mã mũ.

Hệ mã này được Pohlig và Hellman đưa ra năm 1978.

Giả sử p là một số nguyên tố lẻ, và giả sử khoá lập mã e là một số nguyên dương sao cho $(e,p-1)=1$. Cũng như trước đây, để mã hoá một thông báo, trước tiên ta chuyển các chữ cái thành dạng các chữ số tương ứng (thêm số 0 vào trước những số có một chữ số). Ta dùng bảng sau đây:

a	ă	â	b	c	d	đ	e	ê	g	h
01	02	03	04	05	06	07	08	09	10	11

i	k	l	m	n	o	ô	ơ	p	q
12	13	14	15	16	17	18	19	20	21

r	s	t	u	ư	v	x	y
22	23	24	25	26	27	28	29

Sau đó ta nhóm các số nhận được thành từng nhóm 2*m* chữ số theo nguyên tắc sau: 2*m* là số nguyên chẵn lớn nhất sao cho mọi số tương ứng với *m* chữ cái (xét như là một số nguyên có 2*m* chữ số) đều nhỏ hơn *p*. Để dễ hiểu, ta giả sử *p* là số nguyên tố trong khoảng $2929<p<292929$. Mỗi chữ cái được biểu diễn bằng một số không quá 29. Một số có *m* chữ cái sẽ được biểu diễn bằng một số không vượt quá *m* lần số 29 viết liên tiếp. Như vậy, để đảm bảo số đó luôn luôn nhỏ hơn *p, m* chỉ có thể là 1 hoặc 2. Ta lấy $m=2$.

Đối với một khối *P* trong văn bản (là một số 2*m* chữ số), ta lập khối *C* tương ứng trong văn bản mật theo công thức sau:

$$C \equiv P^e \pmod p,\ 0 \le P < p$$

Văn bản mật sẽ chứa những khối chữ số là các số nguyên nhỏ hơn *p*.

Ví dụ. Giả sử số nguyên tố sử dụng để tiến hành lập mã là $p=2939$ và khoá lập mã $e=31$, như vậy $(e,p-1)=(31,2938)=1$.

Ta cần mã hoá thông báo sau:

ĐI HA NÔI NGAY

Trong trường hợp này, $m=2$, và ta nhóm văn bản nhận được khi chuyển sang chữ số thành nhóm bốn chữ số:

0712 1101 1618 1216 1001 2928

Chú ý rằng, để khối cuối cùng đủ bốn chữ số, ta thêm chữ *X* trong văn bản, điều này không gây nhầm lẫn khi đọc thông báo (dĩ nhiên có thể thay chữ *X* bằng bất cứ chữ cái naò không gây hiểu nhầm).

Tiếp theo, ta chuyển các khối *P* trong văn bản thành các khối *C* trong văn bản mật theo công thức sau:

$$C \equiv P^{31} \pmod{2939},\ 0<C<2633$$

Chẳng hạn, để mã hoá khối đầu tiên, ta tính:

$$C \equiv 0712^{31} \pmod{2939}$$

Để tính được *C* một cách nhanh chóng, ta dùng thuật toán bình phương liên tiếp đã xét trong chương 5.

Trước tiên, ta viết 31 dưới dạng cơ số 2: $31=(11111)_2$. Tính toán đơn giản cho ta:

$$721^2 \equiv 1436,\ 712^4 \equiv 1857,\ 712^8 \equiv 1002,\ 712^{16} \equiv 1805 \pmod{2939}.$$

Từ biểu diễn của 31 dưới dạng cơ số 2, ta được:

$$712^{31} \equiv (712.1436.1857.1805) \equiv 898 \pmod{2939}.$$

Sau khi mã hoá toàn bộ văn bản, ta nhận được văn bản mật cần chuyển là:

898 2674 1003 746 1786 2614

Để giải mã một khối C trong văn bản mật, ta cần biết khoá giải mã d. Đó là số d thoả mãn $de \equiv 1 (mod\ p\text{-}1)$, có nghĩa là d là một nghịch đảo của e modulo p-1. Nghịch đảo đó tồn tại do giả thiết $(e,p\text{-}1)=1$. Để tìm lại được khối C trong văn bản, ta chỉ việc nâng khối C lên luỹ thừa d *modulo* p. Thật vậy,

$$C^d \equiv (P^e)^d \equiv P^{de} \equiv P^{k(p-1)+1} \equiv P (\text{mod } p)$$

trong đó $de=k(p\text{-}1)+1$ đối với số nguyên k nào đó, bởi vì $de \equiv 1 (mod\ p\text{-}1)$.

Ví dụ. Để giải mã một khối trong văn bản mật được mã hoá bằng cách sử dụng modulo p=2938 và khoá lập mã e=31, ta cần tìm số nghịch đảo của e=31 modulo p-1=2938. Thuật toán Euclid mở rộng giúp ta dễ dàng tìm được d. Thật vậy, theo các kí hiệu của thuật toán Euclid mở rộng, ta đặt: u=2938, v=31. Tính toán theo thuật toán đó, ta được kết quả sau đây:

q	u_1	u_2	u_3	v_1	v_2	v_3
-	1	0	2938	0	1	31
94	0	1	31	1	-94	24
1	1	-94	24	-1	95	7
3	-1	95	7	4	-379	3
2	4	-379	3	-9	853	1
3	-9	853	1	31	-2938	0

Như vậy, ta có: 31.853-9.2938=1, và d=853

Để giải mã khối C ta dùng công thức

$$P \equiv C^{853} (\text{mod } 2938).$$

Độ phức tạp của thuật toán lập mã và giải mã đối với mã mũ.

Với một khối P trong văn bản, ta mã hoá bằng cách tính $P^e (mod\ p)$, số các phép tính bit cần thiết là $O((log_2\ p)^3)$. Để giải mã trước hết ta phải tìm nghịch đảo d của e modulo p-1. Điều này thực hiện được với $O(log^3\ p)$ phép tính bit, và chỉ cần làm một lần. Tiếp theo đó, để tìm lại được khối P của văn bản từ khối C của văn bản mật, ta chỉ cần tính thặng dư nguyên dương bé nhất của C^d *modulo* p: số các phép tính bit đòi hỏi là $O((log_2 p)^3)$.

Như vậy, thuật toán lập mã và giải mã được thực hiện tương đối nhanh bằng máy tính.

Tuy nhiên ta sẽ chứng tỏ rằng, việc giải mã một văn bản mật được mã hoá bằng mũ nói chung không thể làm được nếu như không biết khoá *e*. Thật vậy, giả sử ta đã biết số nguyên tố *p* dùng làm modun khi lập mã, và hơn nữa, giả sử đã biết khối *C* nào đó trong văn bản mật tương ứng với khối *P* trong văn bản, tức là ta đã biết một đồng dư thức

$$C \equiv P^e \pmod{p}$$

Vấn đề còn lại là xác định *e* từ công thức trên. Số *e* thoả mãn điều kiện đó được gọi là l*ôgarít cơ số P của C modulo p*. Có nhiều thuật toán khác nhau để tìm lôgarit cơ số đã cho modulo một số nguyên tố. Thuật toán nhanh nhất được biết hiện nay đòi hỏi khoảng $exp(\sqrt{\log p \log\log p})$ phép tính bit. Để tìm logarit modulo một số nguyên tố có *n* chữ số thập phân, các thuật toán nhanh nhất cũng đòi hỏi số phép tính bit xấp xỉ số phép tính bit cần dùng khi phân tích một số nguyên *n* chữ số thành thừa số. Như vậy, nếu làm việc với các máy tính có tốc độ 1 triệu phép tính trong một giây, khi *p* có khoảng 100 chữ số thập phân, việc tìm logarit modulo *p* cần khoảng 74 năm, còn trong trường hợp *p* có khoảng 200 chữ số, thời gian cần thiết là 3,8 tỷ năm!

Cần phải lưu ý rằng, có những trường hợp việc tìm ra logarit modulo *p* được thực hiện bằng những thuật toán nhanh hơn rất nhiều. Chẳng hạn khi p-1 chỉ có những ước nguyên tố nhỏ, tồn tại những thuật toán đặc biệt cho phép tính logarit modulo *p* với $O(log^2 p)$ phép tính bit. Rõ ràng những số nguyên tố như vậy không thể dùng để lập mã. Trong trường hợp đó, ta có thể lấy *q* với $p=2q+1$, nếu số *q* cũng là số nguyên tố (khi đó *q*-1 không thể có các ước nguyên tố nhỏ).

Mã mũ và hệ thống có nhiều cá thể tham gia.

Một trong những ưu điểm của hệ mã mũ là trong một hệ thống có nhiều cá thể cùng tham gia trao đổi thông tin, từng cặp cá thể hoặc từng nhóm nhỏ cá thể vẫn có khả năng sử dụng khoá mật mã đang dùng để tạo những khoá mật mã chung, bí mật đối với các cá thể còn lại của hệ thống.

Giả sử *p* là một số nguyên tố lớn và *a* là một số nguyên, nguyên tố cùng nhau với *p*. Mỗi cá thể trong hệ thống chọn một số *k* nguyên tố cùng nhau với *p*-1 làm khoá cho mình. Khi hai cá thể với các khoá k_1, k_2 muốn lập một khoá chung để trao đổi thông tin, cá thể thứ nhất gửi cho cá thể thứ hai số nguyên y_1 tính theo công thức:

$$y_1 \equiv a^{k_1} \pmod{p},\ 0 < y_1 < p$$

Cá thể thứ hai sẽ tìm ra khoá chung *k* bằng cách tính

$$k \equiv y_1^{k_2} \equiv a^{k_1 k_2} \pmod{p},\ 0 < k < p$$

Tương tự cá thể thứ hai gửi cho cá thể thứ nhất số nguyên y_2

$$y_2 \equiv a^{k_2} \pmod{p},\ 0 < y_2 < p$$

và cá thể thứ nhất tìm ra khoá chung *k* theo công thức

$$k \equiv y_2^{k_2} \equiv a^{k_1 k_2} \pmod{p}$$

Ta lưu ý rằng, trong cách lập khoá chung trên đây, các cá thể thứ nhất và thứ hai không cần biết khoá mật mã của nhau, mà chỉ sử dụng khoá mật riêng của mình. Mặt khác, các cá thể còn lại của một hệ thống cũng không thể tìm ra khoá chung k đó trong một thời gian chấp nhận được, vì để làm việc đó, họ phải tính logarit modulo p.

Trên đây là cách lập khoá chung của hai cá thể. Hoàn toàn tương tự như vậy, từng nhóm các thể có thể lập khoá chung.

§4. Các hệ mật mã khoá công khai.

Trong tất cả các hệ mật mã trình bày trên đây, các khoá lập mã đều phải được giữ bí mật, vì nếu khoá lập mã bị lộ thì người ta có thể tìm ra khoá giải mã trong một thời gian tương đối ngắn. Như vậy nếu trong một hệ thống có nhiều cặp cá thể hoặc nhiều nhóm cá thể cần trao đổi thông tin mật với nhau, số khoá mật mã chung cần giữ bí mật là rất lớn, và như vậy, khó có thể bảo đảm được. Hệ mã mà chúng ta nghiên cứu dưới đây được lập theo một nguyên tắc hoàn toàn mới, trong đó việc biết khoá lập mã không cho phép tìm ra khoá giải mã trong một thời gian chấp nhận được. Vì thế, mỗi cá thể chỉ cần giữ bí mật khoá giải mã của riêng mình, trong khi khoá lập mã được thông báo công khai. Trong trường hợp một trong các cá thể bị lộ khoá giải mã của mình, bí mật của các cá thể còn lại không hề bị ảnh hưởng. Lí do của việc có thể xây dựng những hệ mã như vậy chính là điều ta đã nói đến khi xét các hệ mã mũ: độ phức tạp của thuật toán tìm logarit modulo p là quá lớn.

Trước hết, ta nói sơ qua về nguyên tắc của các hệ mã khoá công khai. Giả sử trong hệ thống đang xét có n cá thể cùng trao đổi các thông tin mật. Mỗi các thể chọn cho mình một khoá lập mã k và một công thức mã hoá $E(k)$, được thông báo công khai. Như vậy có n khoá lập mã công khai $k_1, k_2,...,k_n$. Khi cá thể thứ i muốn gửi thông báo cho cá thể thứ j, cũng như trước đây, mỗi chữ trong thông báo được chuyển thành số, nhóm thành từng khối với độ dài nào đó. Sau đó, mỗi khối P trong văn bản được mã hoá bằng khoá lập mã $E(k_j)$ của cá thể thứ j (đã thông báo công khai), và gửi đi dưới dạng $C=E(k_j)(P)$. Để giải mã thông báo này, cá thể thứ j chỉ cần dùng khoá giải mã (bí mật riêng cho mình) D_{k_j}

$$D_{k_j}(\mathrm{C}) = D_{k_j}\, E_{k_j}(P) = P,$$

bởi vì D_{k_j} và E_{k_j} là các khoá giải mã và lập mã của cùng cá thể thứ j. Các cá thể trong hệ thống, nếu nhận được văn bản mật, cũng không thể nào giải mã, vì việc biết khoá lập mã E_{k_j} không cho phép tìm ra khoá giải mã D_{k_j}.

Để cụ thể hoá nguyên tắc vừa trình bày, ta xét ví dụ trên hệ mã khoá công khai được tìm thấy đầu tiên năm 1978 bởi Rivest, Shamir và Adleman (xem[RSA]) (thường được gọi là hệ mã RSA).

Hệ RSA được xây dựng trên cơ sở mã mũ, trong đó khoá là cặp (e,n), gồm số mũ e và *modun* n. Số n được dùng ở đây là tích của hai số nguyên tố rất lớn nào đó, $n=pq$,

sao cho $(e,\phi(n))=1$, trong đó $\phi(n)$ là hàm Euler. Để mã hoá một thông báo, trước tiên ta chuyển các chữ cái thành các số tương ứng và nhóm thành các khối với độ dài lớn nhất có thể (tuỳ thuộc khả năng tính toán) với một số chẵn chữ số. Để mã hoá một khối P trong văn bản, ta lập khối C trong văn bản mật bằng công thức:

$$E(P) \equiv C \equiv P^e (\text{mod } n),\ 0<C<n.$$

Quá trình giải mã đòi hỏi phải biết được một nghịch đảo d của e modulo $\phi(n)$. Nghịch đảo này tồn tại theo điều kiện $(e,\ \phi(n))=1$.

Muốn giải mã một khối C trong văn bản mật, ta tính

$$D(C) \equiv C^d \equiv (P^e)^d \equiv P^{ed} \equiv P^{k\phi(n)+1} \equiv (P^{\phi(n)})^k P \equiv P(\text{mod } n).$$

trong đó $ed=k\phi(n)+1$ đối với số nguyên k nào đó, vì $ed \equiv 1(\text{mod } \phi(n))$, và do định lí Euler ta có: $P^{\phi(n)} \equiv 1(\text{mod } p)$, khi $(P,n)=1$ (chú ý rằng, xác suất để P và n không nguyên tố cùng nhau là hết sức nhỏ, xem bài tập 6.7). Cặp (d,n) như vậy được gọi là khoá giải mã.

Để minh hoạ, ta xét một ví dụ đơn giản, lấy $n=53.61=3233$ và $e=17$. Trong trường hợp đó ta có $(e,\ \phi(n))=(17,52.63)=1$. Giả sử ta cần mã hoá thông báo sau:

ĐA GƯI TIÊN

Trước tiên ta chuyển các chữ cái trong văn bản thành các số tương ứng và nhóm chúng thành từng khối 4 chữ số. Ta có:

0701 1026 1224 1209 1628

Ta mã hoá các khối nhờ công thức

$$C \equiv P^{17} (\text{mod } 3233)$$

Ta lại dùng phương pháp bình phương liên tiếp. Chẳng hạn, đối với khối đầu tiên, ta nhận được:

$$(701)^{17} \equiv 140(\text{mod } 3233)$$

Mã hoá toàn bộ văn bản, ta được văn bản mật sau đây:

140 721 1814 1819 361

Khi nhận được văn bản mật này, để giải mã, ta phải tìm một nghịch đảo d của e modulo $\phi(3233)$. Ta có $\phi(53.61)=52.60=3120$. Dùng thuật toán Euclid mở rộng, ta tính được $d=2753$. Như vậy, để giải mã khối C ta dùng công thức

$$P \equiv C^{2753} (\text{mod } 3233),\ 0 \le P \le 3233$$

Có thể thử lại:

$$C^{2753} \equiv (P^{17})^{2753} \equiv P(P^{3120})^{15} \equiv P(\text{mod } 3233)$$

ở đây ta dùng định lí Euler để nhận được $P^{\phi(3253)} \equiv P^{3120} \equiv 1(\text{mod } 3233)$, khi $(P,3233)=1$ (điều này đúng với mọi khối trong thông báo của chúng ta)

Bây giờ ta chỉ ra rằng, hệ mã RSA thoả mãn các nguyên tắc của hệ mã khoá công khai nói ở đầu tiết này. Trước tiên, ta chú ý rằng, mỗi cá thể phải chọn hai số nguyên tố lớn p và q, cỡ chừng 100 chữ số thập phân. Điều này có thể là trong ít phút nhờ một máy tính. Khi các số nguyên tố p và q đã được chọn, số mũ dùng để mã hoá e sẽ được lấy sao cho $(e, \phi(qp))=1$. Nói chung nên chọn e là số nguyên tố tuỳ ý lớn hơn q và p. Số e được chọn nhất thiết phải thoả mãn $2^e>n=pq$. Nếu điều kiện này không được thoả mãn, ta có $C=P^e<n$, và như vậy để tìm ra P, ta chỉ việc tính căn bậc e của C. Khi điều kiện $2^e>n$ được thoả mãn, mọi khối P khác 0 và 1 đều được mã hoá bằng cách nâng lên luỹ thừa và lấy đồng dư theo modulo n.

Ta cần phải chứng tỏ rằng, việc biết khoá lập mã (công khai) (e,n) không đẫn đến việc tìm được khoá giải mã (d,n).

Chú ý rằng, để tìm nghịch đảo d của e modulo ϕ (n), trước tiên phải tìm được ϕ (n). Việc tìm ϕ (n) không dễ hơn so với phân tích n, bởi vì, một khi biết ϕ (n) và n, ta sẽ phân tích được $n=pq$.

Thật vậy, ta có:

$$p+q=n-\phi(n)+1$$

$$p-q=\sqrt{(p+q)^2-4qp}=\sqrt{(p+q)^2-4n}$$

Từ các công thức đó tìm được q và p.

Nếu ta chọn các số p và q khoảng 100 chữ số thập phân, thì n sẽ có khoảng 200 chữ số thập phân. Để phân tích một số nguyên cỡ lớn như thế, với các thuật toán nhanh nhất hiện nay và với những máy tính hiện đại nhất, ta mất khoảng 3,8 tỷ năm!

Có một vài điều cần lưu ý khi chọn các số p và q để tránh rơi vào trường hợp tích pq bị phân tích nhanh nhờ những thuật toán đặc biệt: q và p cần chọn sao cho p-1 và q-1 chỉ có các thừa số nguyên tố lớn, $(p$-1,q-1$)$ phải nhỏ, q và p phải có số chữ số trong khai triển thập phân khác nhau không nhiều.

Có thể nảy ra câu hỏi: trong một hệ thống nhiều cá thể tham gia, các khoá lập mã đã lại được công khai, làm sao có thể tránh được trường hợp một cá thể này "mạo danh" một cá thể khác để gửi thông báo cho một cá thể thứ ba? Nói cách khác làm sao có thể "kí tên" dưới các thông báo mật? Vấn đề này được giải quyết đơn giản như sau: Giả sử "ông I" cần kí tên dưới thông báo gửi "ông J". Khi đó, trước tiên, ông I tính

$$S\equiv D_{k_i}(I)\equiv I^{d_i}\ (\mathrm{mod}\ n_i).$$

Chú ý rằng chỉ có ông I làm được việc này, vì trong công thức sử dụng khoá giải mã của ông I. Sau đó, I sẽ gửi cho J thông báo

$$C\equiv E_{k_j}(S)=S^{e_j}\ (\mathrm{mod}\ n_j),$$

trong đó (e_j,n_j) là khoá lập mã của J.

Khi nhận được, để giả mã, J trước tiên dùng khoá giải mã riêng của mình để nhận ra S:

$$D_{k_j}(C) \equiv D_{k_j}(E_{k_j}(S)) \equiv \mathrm{S}$$

Để xác minh *S* đích thực là chữ kí của *I, J* chỉ còn việc áp dụng vào *S* khoá lập mã công khai của *I:*

$$E_{k_i}(S) \equiv E_{k_i} D_{k_i}(I) \equiv I$$

Chú ý cách là như trên thích hợp khi $n_j > n_i$, vì khi đó ta luôn có $S < n_j$. Nếu ngược lại, *I* phải tách *S* thành từng khối có độ dài bé hơn n_j và mã hoá từng khối rồi mới chuyển.

Như vậy, một mặt *J* xác định được đó đúng là thông báo do *I* gửi đến, mặt khác *I* cũng không thể từ chối việc mình là chủ nhân của thông báo đó, vì ngoài *I* ra, không ai có khoá mã D_{k_i} để mạo "chữ kí" của *I*.

Trên đây là hệ mật mã khoá công khai xuất hiện đầu tiên. Từ đó đến nay, có nhiều hệ mật mã khoá công khai mới ra đời. Tuy vậy, nguyên tắc chung của cac hệ mã đó là sử dụng những "thuật toán một chiều", tức là những thuật toán cho phép tìm ra một đại lượng nào dó tương đối nhanh, nhưng việc tìm "nghịch đảo" (theo một nghĩa nào đó) của nó đòi hỏi thời gian quá lớn. Độc giả nào quan tâm đến vấn đề này có thể tìm đọc trong những tài liệu chuyên về lí thuyết mật mã. Trong chương tiếp theo, ta sẽ quay về với lí thuyết mật mã khoá công khai khi nghiên cứu các đường cong elliptic.

Cùng với sự phát triển của mật mã khoá công khai, có lẽ sẽ đến lúc bên cạnh địa chỉ và điện thoại của mỗi cơ quan, công ty, còn ghi thêm khoá lập mã của họ!

Bài tập và tính toán thực hành chương 6.

I. Bài tập

6.1. Biết rằng thông báo sau đây đã được mã hoá bằng mã Ceasar (với khoá k nào đó trong khoảng 1-29), hãy tìm khoá và giải mã:

SÔEMR IEIEH USSOT SLUƠI EIÔHE ITSAÂ UƠIEI ÔLUƠI ESÔYB SOSÔE MRDEI EIÔXÂ EIÔBS ORMCE SXSÔL GDESÔ MBSOĂ ÔMTMR

6.2. Dùng mã khối để mã hoá câu

CO CÔNG MAI SĂT CO NGAY NÊN KIM

với khoá ma trận là

$$\begin{pmatrix} 24 & 22 \\ 11 & 10 \end{pmatrix}$$

6.3. Giải mã câu sau đây, biết rằng nó được mã hoá bằng khối với ma trận

$$\begin{pmatrix} 8 & 4 \\ 17 & 11 \end{pmatrix}$$

OD OÂ XC ƠƠ EP YƠ NR EY

6.4. Có thể lập mã khối theo cách sau đây. Giả sử A, B là các ma trận vuông cấp hai. Quá trình mã hoá được thực hiện bởi công thức

$$C \equiv AP+B \pmod{29}.$$

Hãy viết công thức giả mã và cho một ví dụ cụ thể.

6.5. Mã hoá câu sau đây bằng mã mũ với p=3137, e=31:

ĐÊN NƠI AN TOAN

6.6. Hãy giải mã văn bản mật sau đây, nếu biết nó được mã hoá bằng mã mũ với p=3137, e=31:

0206 0248 1345 2200

6.7. Chứng minh rằng, khi lập mã RSA, nếu xảy ra trường hợp có một từ P nào đó trong thông báo không nguyên tố cùng nhau với khoá $n=pq$ đã chọn, và từ này bị phát hiện, thì nhân viên phân tích mã có thể phân tích được n ra thừa số nguyên tố, và do đó, tìm được khoá giải mã.

6.8. Chứng minh rằng, nếu các số q, p được chọn đủ lớn thì trường hợp "rủi ro" nói trong bài tập 6.7 xảy ra với xác suất rất nhỏ.

6.9. Dùng khoá với n=3233, e=17 để mã hoá câu

CHUC MƯNG NĂM MỚI

II. Thực hành tính toán trên máy

Để làm giảm nhẹ các thao tác trong việc lập mã và giải mã văn bản đối với mã khối và mã mũ chúng tôi chỉ ra cách dùng Maple để tính toán.

Để thống nhất ta gọi văn bản là thông báo cần chuyển được viết bằng ngôn ngữ thông thường không có dấu, P là chữ trong văn bản. Bản đã mã hoá của văn bản gọi là văn bản mật, C chữ số tương ứng trong văn bản mật. Giải mã tức là chuyển văn bản mật C thành văn bản ban đầu P.

Do trong Maple không có chế độ tiếng Việt, nên ta dùng kí hiệu aw, aa, dd, ee, oo, ow, uw thay cho các chữ ă, â, đ, ê, ô, ơ, ư tương ứng.

II. 1. Thực hành lập mã và giải mã khối

1. Lập mã: Đối với hệ mã khối và mã mũ, ta ứng các chữ trong văn bản với các số, chuyển các số đó thành hệ thống số khác thông qua khoá lập mã, sau đó lại dùng bảng tương ứng các số vừa tìm được ta được văn bản mật cần chuyển.

Giả sử ta cần mã hoá văn bản P bằng mã khối 2 chữ với khoá lập mã là $\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$.

Khoá lập mã được dùng ở đây là ma trận cấp 2×2, nên nếu số các chữ trong văn bản P là số chẵn thì việc mã hoá xảy ra bình thường nhưng nếu số các chữ trong văn bản P là số lẻ thì chữ cuối cùng của văn bản P sẽ không được mã hoá. Để khắc phục tình trạng đó trong trường hợp thứ 2 ta thêm vào cuối văn bản P một chữ mà không ảnh hưởng đến nội dung của văn bản (chẳng hạn chữ x). Ta thực hiện theo các bước sau đây:

Bước 1: Tính số các chữ trong văn bản P, ta thực hiện bằng dòng lệnh:

```
[>nops([P]);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên số các chữ cái trong văn bản P, nếu đó là số chẵn ta không thay đổi P, nếu đó là số lẻ ta thêm vào cuối văn bản P một chữ cái, ví dụ là chữ x.

Bước 2: Thiết lập tương ứng các chữ cái trong văn bản P (hoặc là P sau khi đã thêm chữ cái x) với các số thông qua dòng lệnh sau đây:

```
[>L:=
subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=11,
i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=22
,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[P]):
```

Sau dấu (:) ấn phím "Enter", vì ở đây ta không cần hiển thị kết quả của dòng lệnh này nên dùng dấu (:) thay cho dấu (;). Khi đó trên màn hình sẽ hiện lên dấu nhắc ([>) để thực hiện tiếp dòng lệnh thứ 3.

Bước 3: Thực hiện dòng lệnh:

```
[>N:=nops(L)/2:
```

(Lệnh `nops(L)` dùng để tính số phần tử của L)

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ xuất hiện dấu nhắc lệnh ([>), ta thực hiện tiếp bước 4.

Bước 4: Thực hiện dòng lệnh:

```
[> subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y}, [seq (msolve ({x-
a1*L[2*k-1]-a2*L[2*k],y-a3*L[2*k-1]-a4*L[2*k]},29),k=1..
N)]);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên các chữ tương ứng của văn bản mật. Ta viết lại theo thứ tự thích hợp sẽ được văn bản mật cần chuyển.

Như vậy, rất đơn giản, để mã hoá văn bản nào đó ta chỉ cần thay các chữ cái trong văn bản vào vị trí của P (với chú ý các chữ cái phải tách biệt nhau bởi dấu (,)) trong dòng lệnh thứ nhất, thứ hai và thay các số a_1a_2,a_3,a_4 của khoá lập mã vào dòng lệnh thứ tư. Để dễ hiểu ta theo dõi thí dụ sau đây:

Chú ý: Đối với hệ mã này ta cho tương ứng chữ *y* với số 0.

Thí dụ 1: Mã hoá câu CHUC BAN THANH CÔNG bằng mã khối với khoá lập mã là $\begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix}$.

Ta thực hiện như sau:

```
[>nops([c,h,u,c,b,a,n,t,h,a,n,h,c,oo,n,g]);
```

```
                              16
```

16 là một số chẵn do đó ta thực hiện tiếp dòng lệnh thứ hai mà không cần phải thêm chữ vào.

```
[>L:=
subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=11,
i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=22
,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[c,h,u,c,b,a,n,t,h,
a,n,h,c,oo,n,g]):

[> N:=nops(L)/2:

[>subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y},   [seq   (msolve   ({x-
23*L[2*k-1]-11*L[2*k],y-9*L[2*k-1]-
12*L[2*k]},29),k=1..N)]);


[{y = aa, x = b},{y = t, x = q},{y = ow, x = n},{y = uw,
x = s},{y = t, x = aa},{x = u, y = m},{y = y, x = s},{x
= l, y = aa}]
```

Vậy ta có văn bản mật tương ứng là BÂ QT NƠ SƯ ÂT UM SY LÂ

Thí dụ 2: Mã hoá câu LY THUYÊT MÂT MA KHÔNG CO GI KHO bằng mã khối với khoá lập mã là $\begin{pmatrix} 8 & 4 \\ 17 & 11 \end{pmatrix}$.

Ta thực hiện như sau:

```
[>nops([l,y,t,h,u,y,ee,t,m,aa,t,m,a,k,h,oo,n,g,c,o,g,i,k
,h,o]);
                                    25
```

25 là một số lẻ do đó ta phải thêm một chữ x vào trong văn bản P.

```
[>L:=subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,
h=11,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21
,r=22,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[l,y,t,h,u,y,e
e,t,m,aa,t,m,a,k,h,oo,n,g,c,o,g,i,k,h,o,x]):

[> N:=nops(L)/2:

[>subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=
h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22
=r,23=s,24=t,25=u,26=uw,27=v,28=x,0=y},[seq (msolve ({x-
8*L[2*k]-4*L[2*k-1],y-17*L[2*k]-11*L[2*k-
1]},29),k=1..N)]);


[{x = v, y = ee}, {x = g, y = n}, {x = k, y = l}, {x =
u, y = l},{x = uw, y = k}, {x = k, y = uw}, {x = q, y =
y}, {x = l, y = q},{y = v, x = x}, {x = h, y = u}, {x =
p, y = t}, {y = h, x = t},{x = aw, y = u}]
```

Vậy ta có văn bản mật tương ứng là VÊ GN KL UL ƯK KƯ QY LQ XV HU PT TH ĂU

2. Giải mã: Giả sử ta nhận được văn bản mật C, cần giải mã C với khoá ma trận A=$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ ta thực hiện trình tự từng bước như sau:

Bước 1: Lập tương ứng mỗi chữ cái trong văn bản mật với một số bằng dòng lệnh như quá trình lập mã.

```
[>L:=
subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=11,
i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=22
,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[C]):
```

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ xuất hiện dấu nhắc lệnh ([>), ta thực hiện tiếp bước 2.

Bước 2: Thực hiện dòng lệnh:

```
[>N:=nops(L)/2:
```

(Lệnh `nops(L)` dùng để tính số phần tử của L)

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ xuất hiện dấu nhắc lệnh ([>), ta thực hiện tiếp bước 3.

Bước 3: Thực hiện dòng lệnh:

```
[>subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y}, [seq (msolve ({L[2*k-
1]-a1*x-a2*y,L[2*k]-a3*x-a4*y},29),k=1..N)]);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên các chữ tương ứng của văn bản. Ta viết lại theo thứ tự thích hợp sẽ được văn bản ban đầu.

Như vậy, rất đơn giản, để giải mã văn bản mật C khi biết khoá ma trận A ta chỉ cần thay các chữ cái trong văn bản mật vào vị trí của C (với chú ý các chữ cái phải tách biệt nhau bởi dấu (,)) trong dòng lệnh thứ nhất và thay các số a_1a_2,a_3,a_4 của khoá lập mã vào dòng lệnh. Để dễ hiểu ta theo dõi thí dụ sau đây:

Thí dụ: Giải mã văn bản mật BÂ QT NƠ SƯ ÂT UM SY LÂ bằng mã khối với khoá lập mã là $\begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix}$.

Ta thực hiện như sau:

```
[>L:=subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,
h=11,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21
,r=22,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[b,aa,q,t,n,ow
,s,uw,aa,t,u,m,s,y,l,aa]):

[> N:=nops(L)/2:

[>subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y}, [seq (msolve ({L[2*k-
1]-23*x-11*y,L[2*k]-9*x-12*y},29),k=1..N)]);


[{y = h, x = c}, {y = c, x = u}, {y = a, x = b}, {y = t,
x = n},{x = h, y = a}, {y = h, x = n}, {y = oo, x = c},
{y = g, x = n}]
```

Như vậy ta có văn bản là CHUC BAN THANH CÔNG.

II. 2. Thực hành lập mã và giải mã mũ

1. Lập mã: Đối với hệ mã mũ, ta ứng các chữ trong văn bản với các số, chuyển các số đó thành hệ thống số khác thông qua khoá lập mã, sau đó lại dùng bảng tương ứng các số vừa tìm được ta được văn bản mật cần chuyển. Giả sử *p* là một số nguyên tố lẻ (để đảm bảo tính an toàn số *p* được chọn ở đây phải là số nguyên tố tương đối

lớn, chẳng hạn lớn hơn 2929), *e* là khoá lập mã (trong đó (e,p-1)=1). Để chuyển một văn bản cho đối tượng có khoá lập mã là (e,p) tiến hành lập mã theo các bước sau:

Bước 1: Tìm *m* (là số nguyên lớn nhất sao cho mọi số tương ứng với *m* chữ cái đều nhỏ hơn *p)*. Ta thực hiện dòng lệnh sau:

```
[>Lp:=length(p);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên kết quả.

Nếu kết quả là một số lẻ thì lấy *m* =(*Lp*-1)/2.

Nếu kết quả là một số chẵn xét p'=2929...29 trong đó số chữ số của p' bằng số chữ số của *p,* ta thực hiện tiếp dòng lệnh:

```
[>p-p';
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên kết quả. Nếu kết quả là một số dương lấy *m=Lp/2*. Nếu kết quả là một số âm lấy *m=(Lp-2)/2*.

Bước 2: Đặt tương ứng mỗi chữ trong văn bản *P* với một số, ta dùng dòng lệnh sau:

```
[>subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=1
1,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=
22,s=23,t=24,u=25,uw=26,v=27,x=28,y=29},[P]);
```

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả.

Bước 3: Chia các số tương ứng tìm được thành từng nhóm *m* số, trong đó các số 1,2,3,4,5,6,7,8,9 thay bởi 01,02,03,04,05,06,07,08,09. Nếu nhóm cuối cùng chưa đủ m số thì ta thêm vào các số 28 (tương ứng với chữ x). Công việc này ta thực hiện bằng tay vì nó đơn giản. Rồi tìm các chữ tương ứng trong văn bản mật. Ta thực hiện dòng lệnh:

```
[>L:=[nhóm thứ nhất, nhóm thứ hai,...]:seq (msolve (x-
L[k]&^e,p),k=1..nops(L));
```

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả.

Như vậy, rất đơn giản, để mã hoá văn bản nào đó ta chỉ cần thay các chữ cái trong văn bản vào vị trí của P (với chú ý các chữ cái phải tách biệt nhau bởi dấu (,)) trong dòng lệnh ở bước 2, và thay các số *p,e* của khoá lập mã vào các dòng lệnh ở bước 1, bước 3. Để dễ hiểu ta theo dõi thí dụ sau đây:

Thí dụ: Với khoá lập mã là p=2938,e=31. Mã hoá thông báo sau:

ĐI HA NÔI NGAY

Ta thực hiện như sau:

```
[>length(2839);
```

```
                                4
```

4 là một số chẵn do đó ta phải thực hiện tiếp lệnh thử với p'=2929

```
[> 2939-2929;
```

```
                                10
```

10 là một số dương do đó lấy m=4/2=2.

```
[>subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=1
1,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=
22,s=23,t=24,u=25,uw=26,v=27,x=28,y=29},[dd,i,h,a,n,oo,i
,n,g,a,y]);

              [7, 12, 11, 1, 16, 18, 12, 16, 10, 1, 29]

[>          L:=[0712,1101,1618,1216,1001,2928]:seq(msolve
(L[k]&^31-x,2939),k=1..nops(L));


{x = 898}, {x = 1853}, {x = 1003}, {x = 2156}, {x =
1786}, {x = 2614}
```

Vậy ta có văn bản mật là 898 1853 1003 2156 1786 2614.

2. Giải mã: Khi nhận được một văn bản mật C gửi cho mình, ta dùng khoá giải mã *(d,n)* của mình để tìm ra nội dung nhận được. Ta thực hiện các dòng lệnh như sau:

Bước 1: Tìm lại khối P' (tương ứng bằng các số) trong văn bản, ta dùng dòng lệnh:

```
[>L:=[C]: seq (msolve (x-L[k]&^d,p),k=1..nops(L));
```

Chú ý khi thay C vào trong dòng lệnh này thì các khối phải cách nhau bởi (,). Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên khối các số tương ứng của P'.

Bước 2: Để nhận được P ta tách mỗi khối của P' nhận được thành các nhóm có hai số rồi tương ứng mỗi nhóm với một chữ cái. Ta thực hiện như sau:

```
[>P:=[P']:subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,
10=g,11=h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p
,21=q,22=r,23=s,24=t,25=u,26=uw,27=v,28=x,29=y},[seq((P[
i] mod  29),i=1..nops(P))]);
```

Ta xét thí dụ sau:

Thí dụ: Hãy giải mã văn bản mật 898 1853 1003 2156 1786 2614 biết khoá giải mã là (853,2939).

Ta thực hiện như sau:

```
[>L:=[898,1853,1003,2156,1786,2614]:seq(msolve(x-
L[k]&^853,2939),k=1..nops(L));


{x = 712}, {x = 1101}, {x = 1618}, {x = 1216}, {x =
1001}, {x = 2928}


[>P:=
[07,12,11,1,16,18,12,16,10,1,29,28]:subs({1=a,2=aw,3=aa,
4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i,13=k,14=l,15=m,
```

```
16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=s,24=t,25=u,26=u
w,27=v,28=x,0=y},[seq((P[i] mod  29),i=1..nops(P))]);


                    [dd, i, h, a, n, oo, i, n, g, a, y, x]
```

Vậy văn bản nhận được là ĐI HA NÔI NGAY.

II. 3. Thực hành lập mã và giải mã RSA

Quá trình này được thực hiện tương tự đối với hệ mã mũ, ta chỉ cần thêm vào chữ kí của mình. Đối với hệ mã này, khoá lập mã là (e,n) trong đó *n* là tích của hai số nguyên tố lớn, khoá giải mã là (e,d). Ta xét thí dụ sau đây:

1. Lập mã:

Thí dụ : Linh có khoá lập mã là (19,221), Lan có khoá lập mã là (13,1457). Linh muốn gửi cho Lan lời nhắn sau: "Anh muốn gặp em, Linh". Anh ta thực hiện như sau:

Bước 1: Ứng mỗi chữ trong lời nhắn với một số, thực hiện bằng dòng lệnh:

```
[>subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=1
1,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=
22,s=23,t=24,u=25,uw=26,v=27,x=28,y=29},[a,n,h,m,u,oo,n,
g,aw,p,e,m,l,i,n,h]);
```

Sau khi ấn phím "Enter" ta nhận được kết quả:

```
    [1, 16, 11, 15, 25, 18, 16, 10, 2, 20, 8, 15, 14,
12, 16, 11]
```

Bước 2: Linh kí tên của mình, trong quá trình này Linh dùng đến khoá giải mã của mình là (91,221)

```
[>L:=[14,12,16,11]:seq(msolve(L[k]&^91-
x,221),k=1..nops(L));


                {x = 27}, {x = 142}, {x = 16}, {x = 80}
```

Bước 3: Thực hiện mã hoá các số nhận được, kể cả chữ kí của Linh

```
[>L:=[1,16,11,15,25,18,16,10,2,20,8,15,14,12,16,11,27,14
2,16,80]:seq(msolve (L[k]&^13-x,1457),k=1..nops(L));


{x = 1}, {x = 252}, {x = 207}, {x = 1360}, {x = 862}, {x
= 237},{x = 252}, {x = 226}, {x = 907}, {x = 1002}, {x =
1287}, {x = 1360}, {x = 679}, {x = 1040}, {x = 252}, {x
= 207},{x = 1207}, {x = 330}, {x = 252}, {x = 919}
```

Vậy Linh sẽ gửi cho Lan lời nhắn 1 252 207 1360 862 237 252 226 907 1002 1287 1360 679 1040 252 207 1207 330 252 919

2. Giải mã:

Thí dụ : Khi nhận được lời nhắn Lan sẽ giải mã theo các bước sau: (khoá giải mã của Lan là (637,1457):

```
[>L:=[1,252,207,1360,862,237,252,226,907,1002,1287,1360,
679,1040, 252, 207,1207,330,252,919]:seq(msolve
(L[k]&^637-x,1457),k=1..nops(L));


{x = 1}, {x = 16}, {x = 11}, {x = 15}, {x = 25}, {x =
18}, {x = 16},{x = 10}, {x = 2}, {x = 20}, {x = 8}, {x =
15}, {x = 14},{x = 12}, {x = 16}, {x = 11}, {x = 27}, {x
= 142}, {x = 16},{x = 80}


[>P:=[1,16,11,15,25,18,16,10,2,20,8,15,14,12,16,11,27,14
2,16,80]:subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,1
0=g,11=h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,
21=q,22=r,23=s,24=t,25=u,26=uw,27=v,28=x,0=y},[seq((P[i]
mod  29),i=1..nops(P))]);


   [a, n, h, m, u, oo, n, g, aw, p, e, m, l, i, n, h, v,
uw, n, r]
```

Lan giải ra được lời nhắn là "Anh muốn gặp em Linh v ư n r", như vậy người nhắn là Linh và 4 chữ cuối là chữ kí của Linh. Để kiểm tra xem có đúng thật hay không, Lan thực hiện tiếp các dòng lệnh:

```
[>L:=[27,142,16,80]:seq(msolve(L[k]&^19-
x,221),k=1..nops(L));


                {x = 14}, {x = 12}, {x = 16}, {x = 11}


[>P:=[14,12,16,11]:subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,
8=e,9=ee,10=g,11=h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,1
9=ow,20=p,21=q,22=r,23=s,24=t,25=u,26=uw,27=v,28=x,0=y},
[seq((P[i] mod  29),i=1..nops(P))]);


                                [l, i, n, h]
```

Vậy người gửi đúng là Linh.