BỘ GIÁO DỤC VÀ ĐÀO TẠO
TR- ỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------O0O-------

ISO 9001:2000

ISO 9001 : 2008

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TR- ỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------O0O-------

NGHIÊN CỨU VÀ XÂY DỰNG MỘT THUẬT TOÁN MÃ HÓA THÔNG ĐIỆP NHỜ KẾT HỢP GIỮA MẬT MÃ CHUYỂN VỊ VÀ MẬT MÃ VIGENERE

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TR- ỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------O0O-------

NGHIÊN CỨU VÀ XÂY DỰNG MỘT THUẬT TOÁN MÃ HÓA THÔNG ĐIỆP NHỜ KẾT HỢP GIỮA MẬT MÃ CHUYỂN VỊ VÀ MẬT MÃ VIGENERE

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Vũ Ngọc Anh
Giáo viên h□ớng dẫn: TS.Hồ Văn Canh
Mã số sinh viên: 1112101003

LỜI CẢM ƠN

Thời gian 4 năm học tập tại trường đại học đã sắp hết, đã đến lúc em cần tổng hợp và hoàn thiện các kiến thức mà em đã học được từ các thầy cô trong ngành Công nghệ thông tin Trường Đại học dân lập Hải Phòng.

Những tháng ngày vừa qua là những tháng ngày được sự tận tình chỉ bảo dạy dỗ của từng thầy từng cô,em đã cố gắng học tập và không ngừng tiếp thu những kiến thức quý báu đó. Để hôm nay đây em được ngồi làm đồ án tốt nghiệp với thành quả của quá trình 4 năm học tập.Trong thời gian làm tốt nghiệp này em đã nhận được sự tận tình của thầy cô giáo bộ môn, và quan trọng nhất là em được thầy Hồ Văn Canh – người trực tiếp truyền đạt những kiến thức quý báu giúp em hoàn thành tốt đồ án này. Em rất cảm ơn thầy đã tận tình chỉ bảo,dìu dắt, dành thời gian hướng dẫn cho em.

Những kiến thức em cần phải học hỏi cũng còn rất nhiều, vì vậy trình độ lý thuyết cũng như các kinh nghiệm thực tế còn hạn chế nên đồ án chắc sẽ còn những thiếu xót mà em chưa nhận ra.Vì vậy em kính mong thầy cô chỉ bảo đề em hoàn thiện hơn về đồ án và các kiến thức chuyên môn.

Em xin chân thành cảm ơn!

Hải Phòng, tháng 7 năm 2015

Sinh viên

Vũ Ngọc Anh

MỤC LỤC

PHẦN MỞ ĐẦU

Các hệ mật mã cổ điển chính là dạng của hệ mật mã khóa đối xứng.

Mã khóa đối xứng được dùng để chỉ các hệ mã mà trong đó, khi biết khóa lập mã ta có thể tìm được khóa giải mã một cách dễ dàng (vì vậy người ta thường coi chúng là một), đồng thời việc giải mã cũng đòi hỏi thời gian như việc lập mã. Các hệ mã thuộc loại này có thời gian lập mã và giải mã tương đối nhanh vì thế các hệ mã đối xứng thường được sử dụng để mã hóa những dữ liệu lớn. Nhưng các hệ mã đối xứng yêu cầu phải giữ bí mật hoàn toàn về khóa lập mã. Nếu đối phương biết khóa lập mã thì coi như thất bại.

Sau đây em xin giới thiệu đôi nét về việc cần thiết để mã hóa thông tin:

Hiện nay tin học đã được áp dụng vào hầu hết các lĩnh vực trong cuộc sống và có một ảnh hưởng rất lớn đối với sự tồn tại và phát triển của các ngành khoa học khác. Trong mọi hệ thống tin học, thông tin luôn là thành phần cơ bản nhất và quan trọng nhất. Chúng ta không ai mà không gặp phải những trường hợp khi máy tính bị mất hết những thông tin quan trọng do nhiều nguyên nhân khác nhau như bị virus, bị hư hỏng thiết bị, do không biết sử dụng, bị đánh cắp hay xoá thông tin… Nói chung vấn đề an toàn và bảo mật thông tin rất đa dạng và phụ thuộc vào nhiều yếu tố chủ quan và khách quan khác nhau như: con người, môi trường, công nghệ… Hiện nay có rất nhiều công cụ và phần mềm hỗ trợ an toàn cho hệ thống máy tính. Tuy nhiên vấn đề đánh giá và chọn lựa một hệ thống an toàn rất phức tạp và chỉ mang tính tương đối bởi vì một hệ thống được đánh giá là rất an toàn hôm nay có thể không còn an toàn nữa vào ngày mai. Nếu chúng ta thường xuyên theo dõi các thông tin bảo mật trên Internet, chúng ta có thể thấy thông tin về những lỗ hổng bảo mật của các hệ điều hành, các phần mềm bảo mật, các dịch vụ… Vì vậy an toàn và bảo mật thông tin là một trong những thành phần quan trọng nhất cần được quan tâm trong việc duy trì và phát triển của hệ thống.

Mật mã và vấn đề an toàn thông tin ?

Mật mã (Cipher) đã xuất hiện cách đây khoảng 4000 năm tại Ai cập. Khi mà các cuộc chiến tranh xẩy ra giữa các đế chế. Thông tin của bên A dưới dạng chữ cái (letter), chữ số (number) hay loại nào đó trước khi được gửi đi sẽ được mã hoá. Bên B nhận được thông tin mã hoá này thực hiện việc giải mã để hiểu được nội dung. Một người lấy được bản mã cũng khó có thể hiểu được nội dung của thông tin vì chỉ có A và B mới có cách giải mã. Thời kì này các thông tin được bảo mật bằng các phương pháp khác nhau, hay còn gọi là các hệ mật mã cổ điển. Các hệ mật mã sớm nhất được biết đến như mật mã Ceazar - mã dich chuyển (Shift Cipher), mã thế (Substitution Cipher)… Các hệ mật mã này được sử dụng trong một thời gian dài. Cho đến khi toán học phát triển. Các hệ mã mới được xây dựng trên các lý thuyết về toán học hiện đại. Một thế hệ mật mã được xây dựng dựa trên độ phức tạp tính toán, các hệ mật mã này được gọi là các hệ mã hiện đại. Các ứng dụng của các hệ mật mã ngày càng được áp dụng trong nhiều lĩnh vực xã hội. Giúp giải quết hàng loạt các vấn đề về an toàn thông tin trên các kênh thông tin không bảo mật.

Mật mã cung cấp một giải pháp nhằm mục đích thực hiện biến một thông tin cụ thể dễ hiểu thành một dạng khác (khó hiểu) có quan hệ chặt chẽ với thông tin gốc. Giờ đây ta gọi thông tin chưa mã hoá (tường minh) là "**bản rõ**", và thông tin sau khi được mã hoá là "**bản mã**". Vậy mật mã là gì ? Tại sao nó lại bảo vệ đươc bí mật thông tin ?

Cơ sở của nó là gì ?

Định nghĩa : Mật mã học là sự nghiên cứu các phương pháp toán học liên quan đến một số khía cạnh của thông tin như sự an toàn, sự toàn vẹn dữ liệu, sự xác nhận tồn tại và sự xác nhận tính nguyên bản của thông tin.

Sau đây em xin lần lượt giới thiệu 6 mật mã cổ điển :

CHƯƠNG I : CÁC HỆ MẬT MÃ CỔ ĐIỂN

1.1. Mở đầu :

Mong muốn được trao đổi thông tin một cách bí mật là một trong những đòi hỏi của con người xuất hiện từ rất sớm trong lịch sử. Vì thế lịch sử của việc trao đổi thông tin mật rất phong phú và bao gồm những phát minh độc đáo mang đầy tính giai thoại. Ngành học nghiên cứu cách thức che dầu thông tin đối với những đối tượng không mong muốn gọi là mật mã học (cryptography)

Mật mã (cipher) được dùng để bảo vệ bí mật của thông tin khi thông tin được truyền trên các kênh thông tin bảo mật như thư tín ,điện thoại,mạng truyền thông máy tính …

- Người A muốn gửi cho người B một văn bản bằng tiếng Việt (gọi là "bản rõ") , muốn được bảo mật thì A phải lập mật mã cho " bản rõ" đó gọi là "bản mã" và gửi bản mã này cho B. A và B có một khóa mật mã chung, vừa để A lập "bản mã" , vừa để B giải "bản mã" thành "bản rõ" . Một người khác không có khóa đó thì dù có lấy được "bản mã" từ kênh truyền tin cũng không thể biến thành "bản rõ" để hiểu được nội dung thông báo mà A gửi cho B.

- Các hệ mật mã cổ điển thực hiện việc bảo mật đó đều dùng một kháo chung cho việc lập mã và giải mã, các bản rõ và bản mã thường dùng cơ sở là bản chữ tự nhiên, cụ thể là ta sẽ dùng 26 chữ cái trong bản chữ cái tiếng Anh.

Để hiểu rõ hơn em sẽ dùng quan niệm toán học để mô tả hình thức hơn

Định nghĩa 1 :

Một hệ mật mã là một bộ năm (P , C , K , E , D) thỏa mãn các điều kiện sau đây:

- P là một tập hữu hạn các bản rõ.
- C là một tập hữu hạn các bản mã.
- K là một tập hữu hạn các khóa.

Với mỗi k $\in$ K , có một hàm lập mã $e_k \in E$,sao cho e_k : P -> C, và một hàm giải mã $d_k \in D$, d_k : C -> P sao cho $d_k(e_k(x)) = x$ với mọi x $\in$ P.

Trong thực tế , P và C thường là bảng chữ cái (hoặc tập các dãy chữ cái có độ dài cố định)

Nếu bản rõ là (một xâu chữ cái):

$x = x_1x_2x_3\ldots x_n$ ($x_i \in P$), và khoá là k $\in$ K thì bản mã sẽ là:

$y = y_1y_2y_3\ldots y_n$ ($y_i \in C$)

Trong đó $y_i = e_k(x_i)$ ($1 \leq i \leq n$). Nhận được bản mã y, biết khoá k, sẽ tìm được bản rõ x, vì $x_i = d_k(y_i)$

Sau đây thay cho bảng chữ cái A, B, C,…,X, Y, Z ta sẽ dùng các con số 0, 1, 2,…, 24, 25 và dùng các phép toán số học theo modulo 26 để diễn tả các phép biến đổi trên bảng chữ cái.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13

O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25

1.2. Mã dịch chuyển

Kí hiệu Z m là tập các số nguyên từ 0 đến (m-1), ký hiệu đó cũng dùng cho vành các số nguyên từ 0 đến (m-1) với các phép cộng và nhân với modulo m. Như vậy, bảng chữ cái tiếng Anh có thể xem là một vành Z_{26} với sự tương ứng kể trên.

Định nghĩa : Mã dịch chuyển (P , C , K , E , D)

P=C=K=Z_{26} với k K, định nghĩa

$e_k(x) = (x + k) \bmod 26;$

$d_k(y) = (y - k) \bmod 26;$

(trong đó x,y Z_{26})

Ví dụ : Dùng khóa k = 9 để mã hóa dòng thư :

"hentoithubay"

Dòng thư đó ta sẽ quy ra tương ứng với dòng số như ở bảng trên(trang 5):

B1: Ta lần lượt lắp từng kí tự bản rõ vào trong bảng sau

Kí tự	h	e	n	t	o	i	t	h	u	b	a	y
Số	7	4	13	19	14	8	19	7	20	1	0	24

B2 : Với khóa k = 9, ta sẽ lần lượt cộng các con số với 9 sau đó rút gọn mỗi tổng với modulo 26 (tức là qua phép mã hóa $e_{9)}$, ta có

Số	16	13	22	2	23	17	2	16	3	10	9	7
Kí tự	q	n	w	c	x	r	c	q	d	k	j	h

B3 : Lấy các kí tự lần lượt trong bảng đã qua mã hóa e_9,t sẽ được bản mã là :

"qnwcxrcqdkjh"

*) Giải mã: Khi B nhận được "qnwcxrcqdkjh" thì sẽ dùng d_9 để giải mã,cụ thể là :

B1: Lần lượt quy đổi từng kí tự bản mã ra số

Kí tự	q	n	w	c	x	r	c	q	d	k	j	h
Số	16	13	22	2	23	17	2	16	3	10	9	7

B2: Thực hiện phép trừ với 9,sau đó rút gọn hiệu với modulo 26, t sẽ có:

Số	7	4	13	19	14	8	19	7	20	1	0	24
Kí tự	h	e	n	t	o	i	t	h	u	b	a	y

B3: Lấy các ký tự vừa chuyển đổi sẽ thu được bản rõ mà A đã gửi "hentoithubay"

*) ***Nhận xét*** :

Cách đây 2000 năm mã dịch chuyển đã được Julius Ceasar sử dụng, với khoá k=3 mã đích chuyển được gọi là mã Ceasar.

Tập khoá phụ thuộc vào Z_m với m là số khoá có thể.

Trong tiếng Anh tập khoá chỉ có 26 khoá có thể, việc thám mã có thể được thực hiện bằng cách duyệt tuần tự 26 khoá đó

⇨ Rõ ràng mật mã dịch chuyển không an toàn vì người ra có thể tìm được khóa k bằng cách thử toàn bộ các khóa có thể cho đến khi nhận được thông báo có nghĩa .

1.3. Mã thay thế

Định nghĩa : Mã thay thế nghĩa là thay thế từng kí tự của bản rõ bằng các kí tự khác mà các ký tự này đều thuộc bảng chữ cái.Như vậy khóa của mã này chính là một hoán vị của bảng chữ cái.

Mã thay thế (P , C , K , E , D)

$P = C = Z_{26}$, $K = S(Z_{26}) – S(E)$ là tập các phép hoán vị các phần tử của E

Với mỗi л ∈ K, tức là một hoán vị trên Z_{26}, ta xác định :

$e_л(x) = л(x)$;

$d_л(y) = л^{-1}(y)$;

với x, y ∈ Z_{26}, $л^{-1}$ là nghịch đảo của л

Ví dụ : л được cho bởi hoán vị của các chữ cái thuộc Z_{26} :

E	a	b	c	d	e	f	g	h	i	j	k	l	m
S(E)	x	n	y	a	h	p	o	g	z	q	w	b	t

E	n	o	p	q	r	s	t	u	v	w	x	y	z
S(E)	s	f	l	r	c	v	m	u	e	k	j	d	i

Với bảng trên, ta có thể đối chiếu tương ứng từng kí tự trong bản rõ sau:

"hentoithubay"

Như h -> g , e ->h , n -> s , t -> m ….

Thành bản mã "ghsmfzmgunnxd"

*) Giải mã ta sẽ dùng khóa $л^{-1}$ làm ngược lại ,nghĩa là :

g -> h , h -> e , s ->n …

Ta sẽ thu được bản rõ : "hentoithubay"

*) ***Nhận xét*** :

Mã thay thế có tập hợp khoá khá lớn - bằng số các hoán vị trên bảng chữ cái, tức số các hoán vị trên Z_{26} (hay là 26!)

⇨ Việc duyệt toàn bộ các hoán vị để thám mã là rất khó, ngay cả đối với máy tính. Tuy nhiên, ta sẽ thấy có những phương pháp thám mã khác dễ dàng thực hiện, và do đó mã thay thế cũng không thể được xem là "an toàn".

1.4. Mã Apphin

Phép lập mã được cho bởi một hàm Apphin dạng:

$$e(x) = ax + b \bmod 26$$

trong đó a, b $\in Z_{26}$ (chú ý: nếu a = 1 ta có mã dịch chuyển)

Để có được phép giải mã tương ứng, tức để cho phương trình

$$ax + b = y \bmod 26$$

có nghiệm x duy nhất (với bất kỳ y $\in$ □ 26 cho trước), hay nói cách khác hàm Apphin phải là đơn ánh. Theo một định lý số học, điều kiện cần và đủ là a nguyên tố với 26, tức là (a, 26) = 1.

Ở đây (a, 26) ký hiệu cho ước số chung lớn nhất của a và 26.

Khi (a, 26) = 1 thì có số $a^{-1} \in Z_{26}$ sao cho $a.a^{-1} = a^{-1}.a = 1 \bmod 26$, và do đó, nếu:

$$y = ax + b \bmod 26$$
$$\leftrightarrow ax = y - b \bmod 26$$
$$\leftrightarrow a^{-1}.ax = a^{-1}.(y - b) \bmod 26$$
$$\leftrightarrow (a^{-1}.a)x = a^{-1}.(y - b) \bmod 26$$
$$\leftrightarrow x = a^{-1}.(y - b) \bmod 26$$

$$\rightarrow d(x) = a^{-1}.(y - b) \bmod 26$$

Định nghĩa : Mã Apphin((P , C , K , E , D)

P = C = Z_{26} ; K = { (a,b) $\in Z_{26}$ x Z_{26} : (a,26) = 1}

Với mỗi k = (a, b) $\in$ K ta có định nghĩa :

$e_k(x) = ax + b \bmod 26$

$d_k(y) = a^{-1} (y - b) \bmod 26$

Trong đó x,y $\in Z_{26}$

Để thử tính chất xem khóa có hợp lệ không, ta cần phải có những thuật toán thử (a, m) = 1 , và tính $a^{-1} \bmod m$ khi (a , m) = 1 .

Nhưng với m = 26 ta dễ thử rằng các số a sao cho (a ,26) = 1 là :

a	1	3	5	7	9	11	15	17	19	21	23	25
a^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

Ta lấy ví dụ : Với k =(5 , 6) và bản rõ :

"hentoithubay"

Bước 1 : ta quy đổi các kí tự bản rõ thành số theo quy ước (a -> 1, b ->2 ,c->3 …)

	h	e	n	t	o	i	t	h	u	b	a	y
x	7	4	13	19	14	8	19	7	20	1	0	24
y = 5x + 6 mod 26												
y	15	0	19	23	24	20	23	15	2	11	6	22
	p	a	t	x	y	u	x	p	c	l	g	w

Bước 2 : sau khi chuyển đổi các số qua công thức,a ánh xạ ngược lại từ số ra kí tự tương ứng ,và bản mã sẽ là :

"patxyuxpclgw

*) Giải mã : Khi B nhận được bản mã từ A,sẽ tiến hành giải mã:

K =(5,6) tức là a = 5, b = 6;

a = 5 => $a^{-1} = 21$ (như trong bảng đã nêu)

Giờ công thức giải mã sẽ là :

$d_k(y) = 21 (y – 6) \bmod 26$

Bước 1 : ta quy đổi các kí tự bản mã thành số theo quy ước (a -> 1,b ->2 ,c->3 …)

Ví dụ : x = 21 (15 -6) mod 26 = 7 ….

	p	a	t	x	y	u	x	p	c	l	g	w
y	15	0	19	23	24	20	23	15	2	11	6	22
x = 21(y – 6) mod 26												
x	7	4	13	19	14	8	19	7	20	1	0	24
	h	e	n	t	o	i	t	h	u	b	a	y

Bước 2 : Ta lấy các kí tự quy đổi sẽ thu được bản rõ :

"hentoithubay"

*) ***Nhận xét*** :

Với mã Apphin, số các khoá có thể có bằng (số các số ≤ 26 và nguyên tố với 26) × 26, tức là 12 × 26 = 312. Việc thử tất cả các khoá để thám mã trong trường hợp này tuy khá mất thì giờ nếu tính bằng tay, nhưng không khó khăn gì nếu dùng máy tính.

=> Do vậy, mã Apphin cũng không phải là mã an toàn.

1.5. Mã Vigenere

Mã lấy tên của Blaise de Vigenēre, sống vào thế kỷ 16. Khác với các mã trước, mã Vigenēre không thực hiện trên từng ký tự một,mà được thực hiện trên từng bộ m ký tự (m là số nguyên dương).

1.5.1. Định nghĩa: Mã Vigenere((P , C , K , E , D)

$P = C = K = Z_{26}^{m}$ Với mỗi $k = (k_1,k_2,\ldots,k_m) \in K$ ta có :

$e_k(x_1 , x_2,\ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots , x_m + k_m)$ modulo 26

$d_k(y_1 , y_2,\ldots, y_m) = (y_1 - k_1, y_2 - k_2,\ldots, y_m - k_m)$ modulo 26

1.5.2. Ví dụ : Cho Khóa k là từ CIPHER ,

⇨ Độ dài khóa là 6 (m =6) – và ta sẽ quy đổi khóa k theo quy tắc đổi kí tự sang số,nghĩa là k = (2,8,15,7,4,17)

⇨ Bản rõ : "hentoithubay"

Bước 1: Chuyển bản rõ và khóa sau khi quy đổi vào bảng sau :

	h	e	n	t	o	i	t	h	u	b	a	y
X	7	4	13	19	14	8	19	7	20	1	0	24
K	2	8	15	7	4	17	2	8	15	7	4	17
Y	9	12	2	0	18	25	21	15	9	8	4	15
	j	m	c	a	s	z	v	p	j	i	e	p

Bước 2 : Cộng lần lượt các số của bản rõ với khóa (lấy theo modulo 26) ta sẽ thu được cái chữ số bản mã

Bước 3: Lấy các kí tự đã chuyển đổi ngược,ta được bản mã là :

"jmcaszvpjiep"

*) Quy tắc giải mã :

Với bản mã : "jmcaszvpjiep"

Ta sẽ dùng hàm d_k lần lượt giải theo các bước :

Bước 1: Chuyển bản mã và khóa sau khi quy đổi vào bảng sau :

	j	m	c	a	s	z	v	p	j	i	e	p
y	9	12	2	0	18	25	21	15	9	8	4	15
k	2	8	15	7	4	17	2	8	15	7	4	17
x	7	4	13	19	14	8	19	7	20	1	0	24
	h	e	n	t	o	i	t	h	u	b	a	y

Bước 2 : Trừ lần lượt các số của bản rõ với khóa (lấy theo modulo 26) ta sẽ thu được cái chữ số bản rõ

Bước 3: Lấy các kí tự đã chuyển đổi ngược,ta được bản rõ là :

"hentoithubay"

) Nhận xét :

Mã Vigenēre với m = 1 sẽ trở thành mã Dịch chuyển.

Tập hợp các khoá trong mã Vigenēre mới $m \geq 1$ có tất cả là 26^m khoá có thể có. Với m = 6, số khoá đó là 308.915.776, duyệt toàn bộ chừng ấy khoá để thám mã bằng tính tay thì khó, nhưng với máy tính thì vẫn là điều dễ dàng.

1.6. Mã Hill

Mã này được đề xuất bởi Lester S.Hill năm 1929. Mã cũng được thực hiện trên từng bộ m ký tự, mỗi ký tự trong bản mã là một tổ hợp tuyến tính (trên vành Z_{26}) của m ký tự trong bản rõ. Như vậy, khoá sẽ được cho bởi một ma trận cấp m, tức là một phần tử của $Z^{m}_{26}{}^{\times m}$. Để phép biến đổi tuyến tính xác định bởi ma trận $k \in Z^{m}_{26}{}^{\times m}$ có phép nghịch đảo, ma trận k cũng phải có phần tử nghịch đảo $k\text{-}1 \in Z^{m}_{26}{}^{\times m}$. Điều kiện cần và đủ để ma trận k có ma trận nghịch đảo là định thức của nó - ký hiệu det(k),- nguyên tố với m.

Định nghĩa : Mã Hill((P , C , K , E , D)

Cho m là số nguyên dương.

$P = C = Z^{m}_{26}$

$K = \{ k \in Z^{m}_{26}{}^{\times m} : (\det(k), 26) = 1 \}$

với mỗi $k \in K$ định nghĩa:

$$e_k(x_1, x_2, \ldots, x_m) = (x_1, x_2, \ldots, x_m).k$$

$$d_k(y_1, y_2, \ldots, y_m) = (y_1, y_2, \ldots, y_m).k^{-1}$$

Ví dụ : Lấy m = 2 ; và $k = \begin{bmatrix} 11 & 3 \\ 8 & 7 \end{bmatrix}$

Với bộ 2 ký tự (x1, x2), ta có mã là (y1, y2) = (x1, x2). k được tính bởi

$y_1 = 11.x_1 + 3.x_2$

$y_2 = 8.x_1 + 7.x_2$

Giả sử ta có bản rõ: "tudo", tách thành từng bộ 2 ký tự, và viết dưới dạng số ta được 19 20 | 03 14 , lập bản mã theo quy tắc trên, ta được bản mã dưới dạng số là: 09 06 | 23 18, và dưới dạng chữ là

"fgxs" .

Để đơn giản cho việc tính toán, thông thường chọn ma trận vuông 2×2. Khi đó có thể tính ma trận nghịch đảo theo cách sau :

Giả sử : k = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ ta có ma trận nghịch đảo $k^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}$

Được tính : $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} \frac{d}{ad-bc} & \frac{-b}{ad-bc} \\ \frac{-c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}$

Một chú ý là để phép chia luôn thực hiện được trên tập Z_{26} thì nhất thiết định thức của k : det(k) = (ad – bc) phải có phần tử nghịch đảo trên Z_{26}, nghĩa là (ad – bc) phải là một trong các giá trị : 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, hoặc 25. Đây cũng là điều kiện để ma trận k tồn tại ma trận nghịch đảo.

Khi đó: $k^{-1}.k = 1$ là ma trận đơn vị (đường chéo chính bằng 1)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} \frac{d}{ad-bc} & \frac{-b}{ad-bc} \\ \frac{-c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix} * \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Định thức của $\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}^{-1} = \begin{bmatrix} 7 & -8 \\ -3 & 11 \end{bmatrix}$ mod 26

Đây là một ví dụ đặc biệt vì ma trận có định thức bằng 1, chúng ta sẽ xem xét một ví dụ tìm nghịch đảo của ma trận 2×2 khác.

Ví dụ :

Định thức của $\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}$ là $9\times 7 - 5 \times 4 = 43 \bmod 26 = 17$;

Khi đó $\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{7}{17} & \frac{-4}{17} \\ \frac{-5}{17} & \frac{9}{17} \end{bmatrix} \bmod 26$

Vì 17 mod 26 sẽ tương đương với nghịch đảo của 17 mod 26.Trong bảng nghịch đảo ta dễ thấy nghịch đảo của 17 trong Z_{26} là 23. Nên :

$$\begin{bmatrix} \frac{7}{17} & \frac{-4}{17} \\ \frac{-5}{17} & \frac{9}{17} \end{bmatrix} \bmod 26 = \begin{bmatrix} 7 \times 23 & -4 \times 23 \\ -5 \times 23 & 9 \times 23 \end{bmatrix} \bmod 26 = \begin{bmatrix} 161 & -92 \\ -115 & 207 \end{bmatrix}$$

$$\bmod 26 = \begin{bmatrix} 5 & 12 \\ 15 & 25 \end{bmatrix} \bmod 26 \Rightarrow \text{Kết quả } \begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}^{-1} = \begin{bmatrix} 5 & 12 \\ 15 & 25 \end{bmatrix}$$

Ta không có công thức để đánh giá số khoá k có thể có với m cho trước. Trong mục sau ta sẽ xét một phương pháp thám mã đối với mã Hill.

1.7. Mã chuyển vị

Khác với các mã trước, mã hoán vị không thay đổi các ký tự trong bản rõ mà chỉ thay đổi vị trí các ký tự trong từng bộ m các ký tự của bản rõ. Ta ký hiệu S_m là tập hợp tất cả các phép hoán vị của {1, 2,…, m}.

1.7.1. Định nghĩa

Cho m là số nguyên dương. $P = C = Z^m_{26}$, $K = S_m$ với mỗi $k = \pi \in S_m$, ta có:

$$e_k(x_1, x_2, \ldots, x_m) = (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(m)})$$

$$d_k(y_1, y_2, \ldots, y_m) = (y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \ldots, y_{\pi^{-1}(m)})$$

trong đó π^{-1} là hoán vị nghịch đảo của π

*) Mã hóa :

1.7.2. Ví dụ :

Giả sử m = 6 , và khóa k được cho bởi hoán vị của π

i	1	2	3	4	5	6
π	3	5	1	6	4	2

Nghĩa là : các thứ tự khóa sẽ bị xáo trộn theo 1 trật tự do người mã hóa đặt ra

Người gửi gửi bản rõ có nội dung :

"hentoithubay"

Bước 1 : tạo bảng để đẩy lần lượt các kí tự theo 1 trật tự:

Trong đây ta cần nắm được các thông tin : độ dài khóa(ở đây là 6) , độ dài bản rõ(ở đây là 12) -> thứ tự khóa sẽ được lặp lại từ 1 đến 6 cho đến đủ độ dài bản rõ thì thôi

	h	e	n	t	o	i	t	h	u	b	a	y
i	1	2	3	4	5	6	1	2	3	4	5	6

Bước 2 : ta bắt đầu chuyển đổi khóa theo quy tắc

i	1	2	3	4	5	6
π	3	5	1	6	4	2

Nghĩa là : 1 -> 3 , 2-> 5 , 3 -> 1, 4 -> 6, 5 -> 4 , 6->2;

Bước 3. Chuyển đổi tương ứng theo quy tắc khóa,ta được hoán vị của các kí tự (chữ h xuống thứ 3, chữ e xuống thứ 5 …)

π	3	5	1	6	4	2	3	5	1	6	4	2
	n	o	h	i	t	e	u	a	t	y	b	h

Giờ ta lấy lần lượt các kí tự trong bảng này sẽ được bản mã là:

"nohiteuatybh"

*) Giải mã:

Bước mã hóa sau không thể dùng khóa mà bên A đưa ra nữa,mà ta sẽ phải tìm khóa nghịch đảo của khóa bên A đưa ra,cụ thể là ta sẽ tìm phép hoán vị nghịch đảo của π - kí hiệu : π^{-1}

Trở lại với khóa π

i	1	2	3	4	5	6
π	3	5	1	6	4	2

Ta giờ sẽ sắp xếp lại khóa π theo thứ tự tăng dần từ 1 -> 6 ,khi sắp xếp lại thì chỉ số i sẽ được sắp lại tương ứng,lúc này chỉ số i đó sẽ chính là khóa π^{-1}

i	1	2	3	4	5	6
π^{-1}	3	6	1	5	2	4

Bước 1 : Bên B nhận được bản mã :

"nohiteuatybh"

Thứ tự các bước sẽ lần lượt là :

y	n	o	h	i	t	e	u	a	t	y	b	h
i	1	2	3	4	5	6	1	2	3	4	5	6
π^{-1}	3	6	1	5	2	4	3	6	1	5	2	4
x	h	e	n	t	o	i	t	h	u	b	a	y

Bước 2 : lấy các kí tự của hàng x ta sẽ thu được đúng bản rõ tương ứng

"hentoithubay"

CHƯƠNG 2 : Hệ mật mã Vigenege và hệ mật mã chuyển vị

2.1. Hệ mật mã Vigenere

2.1.1. Định nghĩa

- Nó được phát minh vào thế kỷ thứ 16 và được viết đầu tiên bởi nhà ngoại giao Pháp Blaise de Vigenère.
- Mã Vigenere((P , C , K , E , D)

  $P = C = K = Z_{26}^{m}$ Với mỗi $k = (k_1,k_2,\ldots,k_m) \in K$ ta có :

  $e_k(x_1 , x_2,\ldots, x_m) = (x_1 + k_1,x_2 + k_2, \ldots , x_m + k_m)$ modulo 26

  $d_k(y_1 , y_2,\ldots, y_m) = (y_1 - k_1,y_2 - k_2,\ldots,y_m - k_m)$ modulo 26

Giải thích :

C : bản rõ, thường kí hiệu là bản rõ $x = x_1,x_2,x_3\ldots x_n$;

D: bản mã,thường kí hiệu là $y = y_1,y_2,y_3,\ldots,y_n$;

K: khóa,thường kí hiệu $k= k_1,k_2,k_3,\ldots,k_m$;

$x_i \in \{a,b,c,\ldots z\}$, $y_i \in \{a,b,c,\ldots z\}$

2.1.2 Phương pháp mã hóa :

Có bản rõ và khóa ta sẽ biết được n : độ dài bản rõ , và m : độ dài khóa (m <=n)

Các bước :

Bước 1: Chuyển bản rõ và khóa từ chữ cái sang số { 0 , 1, 2, 3, 4,… 25}

Theo quy tắc đã nêu (a -> 0, b->1 ,c-> 2… z->25)

Bước 2 : Cộng lần lượt các số đúng theo thứ tự của bản rõ với số của khóa đã quy đổi :

Trường hợp 1 : nếu m = n, ta tiến hành cộng theo thứ tự bình thường từ trái sang phải.

Trường hợp 2 : nếu m < n , ta sẽ cần thêm khóa: m = m + (n-m)

	h	e	n	t	o	i	t	h	u	b	a	y
X	7	4	13	19	14	8	19	7	20	1	0	24
K	2	8	15	7	4	17	2	8	15	7	4	17
Y	9	12	2	0	18	25	21	15	9	8	4	15
	j	m	c	a	s	z	v	p	j	i	e	p

$$x1, x2, x3 \ldots xn$$
$$+$$
$$k1, k2, k3, \ldots, km$$
$$y1, y2, y3, \ldots, yn$$

Bước 3 : Chuyển đổi ngược lại từ số thành chữ cái để có được bản mã

Ví dụ : Cho bản rõ : "hentoithubay" và khóa k là : "cipher"

- ⇨ Độ dài khóa là 6 (m =6) – và ta sẽ quy đổi khóa k theo quy tắc đổi kí tự sang số,nghĩa là k = (2,8,15,7,4,17)
- ⇨ Trường hợp này là trường hợp 2: nghĩa là độ dài m = m + (n – m)
  m = 6 + (12 – 6) = 12
- ⇨ Bản mã là "jmcaszvpjiep"

2.1.3 Phương pháp giải mã :

Có bản rõ và khóa ta sẽ biết được n : độ dài bản mã , và m : độ dài khóa (m <=n),bước này sẽ làm ngược lại của pp mã hóa

Các bước :

Bước 1: Chuyển bản mã và khóa từ chữ cái sang số { 0 , 1, 2, 3, 4,… 25}

Theo quy tắc đã nêu (a -> 0, b->1 ,c-> 2… z->25)

Bước 2 : Trừ lần lượt các số đúng theo thứ tự của bản rõ với số của khóa đã quy đổi

$$\frac{y1, y2, y3, \dots, yn}{k1, k2, k3, \dots, km} - x1, x2, x3, \dots, xn$$

Bước 3 : Chuyển đổi ngược lại từ số thành chữ cái để có được bản rõ

Ví dụ: Bản mã ta vừa nhận được là : "jmcaszvpjiep" và đã biết khóa k = "cipher" giờ ta sẽ tiến hành giải mã.việc thực hiện được làm qua bảng sau :

	j	m	c	a	s	z	v	p	j	i	e	p
y	9	12	2	0	18	25	21	15	9	8	4	15
k	2	8	15	7	4	17	2	8	15	7	4	17
x	7	4	13	19	14	8	19	7	20	1	0	24
	h	e	n	t	o	i	t	h	u	b	a	y

2.1.4 Phân tích,đánh giá :

Độ an toàn của mật mã :

- Mã Vigenēre với m = 1 sẽ trở thành mã Dịch chuyển.
- Nếu độ dài khóa mà rất nhỏ so với độ dài bản rõ(m << n) thì có thể thám mã được. Tập hợp các khoá trong mã Vigenēre với m ≥ 1 có tất cả là 26^m khoá có thể có. Duyệt toàn bộ chừng ấy khoá để thám mã bằng tính tay thì khó, nhưng với máy tính thì vẫn là điều dễ dàng.Phương pháp thám mã cụ thể :

Khi người thám mã đã xác định được mã pháp mà Vinegere thì việc tiếp theo là tìm độ dài khóa(có thể dùng phép thử Kasiski)

Việc xác định độ dài khóa đúng sẽ giúp việc xác định bản rõ qua bảng ma trận phép thử,với số cột là độ dài khóa.

Giả sử với ví dụ trên: khóa (CIPHER) có độ dài là 6.

Bước 1: Kẻ thám mã sẽ lập bảng có số cột là 6,và lần lượt đẩy từng kí tự bản mã vào hàng theo thứ tự của ma trận.

Bước 2: Xác định tần suất của các kí tự xuất hiện trong bản mã theo thứ tự giảm dần

Bước 3 : Đối chiếu với tần số đặc trưng của ngôn ngữ tiếng Anh tự nhiên

Kí tự	Xác suất	Kí tự	Xác suất	Kí tự	Xác suất
A	.082	J	.002	S	.063
B	.015	K	.008	T	.091
C	.028	L	.040	U	.028
D	.043	M	.024	V	.010
E	.0127	N	.067	W	.023
F	.022	O	.075	X	.001
G	.020	P	.019	Y	.020
H	.061	Q	.001	Z	.001
I	.070	R	.060		

Từ bảng trên, Beker và Piper phân 26 chữ cái thành 5 nhóm như sau:

E: có xác suất khoảng 1,120

T, A, O, I, N, S, H, R : mỗi ký tự có xac suất khoảng 0,06 đến 0,09

D, L : mỗi ký tự có xác suất chừng 0,04

C, U, M, W, F, G, Y, P, B: mỗi ký tự có xác suất khoảng 0,015 đến 0,023

V, K, J, X, Q, Z mỗi ký tự có xác suất nhỏ hơn 0,01

Việc xem xét các dãy gồm 2 hoặc 3 ký tự liên tiếp (được gọi là bộ đôi - diagrams và bộ ba - Trigrams)cũng rất hữu ích. 30 bộ đôi thông dụng nhất (theo hứ tự giảm dần) là: TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI và OF. 12 bộ ba thông dụng nhất (theo thứ tự giảm dần) là: THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR và DTH.

Việc tính toán đối chiếu lần lượt các kí tự có tần suất cao ứng với các kí tự bản mã được lần lượt,sao cho khả năng tạo ra bản rõ có nghĩa hiểu được

- Nếu m = n thì mật mã là an toàn.Nhưng độ dài bản rõ càng dài thì độ dài khóa cũng càng dài => điều này gây khó khăn cho việc trao đổi khóa mã là rất lớn.

⇨ Do đó thay vì tăng độ dài khóa ta sẽ cải tiến nó bằng cách kết hợp với mật mã chuyển vị nhằm chống lại khả năng tấn công nhằm vào khóa.

⇨ Với việc kết hợp như vậy ta sẽ tạo được an toàn bởi qua hai lớp khóa , việc tìm ra sự kết hợp 2 mã pháp đã gây rất nhiều khó khăn với việc chỉ cần dựa vào một mật mã đã biết.

⇨ Kẻ thám mã lúc này để tìm được chính xác 2 khóa khác nhau cũng là điều không thể.

2.2 Hệ mật mã chuyển vị

2.2.1 Định nghĩa :

- Mã Chuyển vị((P , C , K , E , D)

  P = C = Z^m_{26}, K = S_m với mỗi k = $\pi \in S_m$, ta có

  $ek(x_1, x_2,\ldots, x_m) = (x_{\pi(1)}, x_{\pi(2)},\ldots, x_{\pi(m)})$

  $dk(y_1, y_2,\ldots, y_m) = (y_{\pi(1)} , y_{\pi(2)} ,..., y_{\pi(m)})$

  Giải thích :

  C : bản rõ, thường kí hiệu là bản rõ $x = x_1,x_2,x_3\ldots x_n$;

  D: bản mã,thường kí hiệu là $y = y_1,y_2,y_3,\ldots,y_n$;

  K: khóa,thường kí hiệu $k= k_1,k_2,k_3,\ldots,k_m$;

  $x_i \in \{a,b,c,\ldots z\}$, $y_i \in \{a,b,c,\ldots z\}$

2.2.2 Phương pháp mã hóa

Có bản rõ và khóa ta sẽ biết được n : độ dài bản rõ , và m : độ dài khóa (m <=n) . Các bước :

Bước 1 : Lập bảng ma trận có số cột là độ dài khóa k

Bước 2 : Lấy $\frac{n}{m}$

TH1 : Nếu n % m = 0 (n chia hết cho m) thì số dòng của bảng là $\frac{n}{m}$

TH2 : Nếu n % m != 0 thì số dòng của bảng là $\frac{n}{m}$ + 1

Bước 3 : Lập bảng ma trận theo số hàng và số cột ở bước 1 và bước 2

Chỉ số cột sẽ lần lượt là các số xuất hiện trong khóa

Chỉ số hàng sẽ được đánh số từ 0 đến $\frac{n}{m}$ (hoặc $\frac{n}{m} + 1$)

Bước 4 : Viết bản rõ vào bảng ma trận vừa lập theo thứ tự tự nhiên

Bước 5 : Nhặt các kí tự trong ma trận vừa lập theo cột từ trên xuống và từ cột bé nhất đến cột lớn nhất theo quy ước của khóa.Kết qua thu được đó chính là bản mã

Ví dụ : Cho Bản rõ "HENTOITHUBAY" có khóa k = "240531"

Ta có m = 6 , và khóa k được cho bởi hoán vị của π (nghĩa là : các thứ tự khóa sẽ bị xáo trộn theo 1 trật tự do người mã hóa đặt ra)

Ta dễ dàng xác định được số hàng của ma trận là 2 (n / m = 12 / 6) và số cột là 6 (m = 6)

Ta có bảng ma trận sau:

Chỉ số	2	4	0	5	3	1
1	H	E	N	T	O	I
2	T	H	U	B	A	Y

⇨ Nhặt các kí tự trong ma trận vừa lập theo cột từ trên xuống và từ cột bé nhất đến cột lớn nhất theo quy ước của khóa nghĩa là ta lấy từ cột có giá trị 0 -> 1 ->…->5

NU IY HT OA EH TB

⇨ Hợp lại ta có bản mã : "NUIYHTOAEHTB"

2.2.3 Phương pháp giải mã

Có bản rõ và khóa ta sẽ biết được n : độ dài bản mã , và m : độ dài khóa

(m <=n) . Các bước :

Bước 1: Lập bảng ma trận có số cột là độ dài khóa k

Bước 2 : Lấy $\frac{n}{m}$

TH1 : Nếu n % m = 0 (n chia hết cho m) thì số dòng của bảng là $\frac{n}{m}$

TH2 : Nếu n % m != 0 thì số dòng của bảng là $\frac{n}{m}$ + 1

Bước 3 : Lập bảng ma trận theo số hàng và số cột ở bước 1 và bước 2

Chỉ số cột sẽ lần lượt là các số xuất hiện trong khóa

Chỉ số hàng sẽ được đánh số từ 0 đến $\frac{n}{m}$ (hoặc $\frac{n}{m} + 1$)

Bước 4 : Viết các kí tự trong bản mã theo cột từ trên xuống và từ cột bé nhất đến cột lớn nhất theo quy ước của khóa.

Bước 5 : Viết bản mã vào bảng ma trận vừa lập theo thứ tự tự nhiên

Kết quả ta sẽ thu được bản rõ

Ví dụ : Ta sẽ lấy bản mã vừa thu được ở phần mã hóa

X = “NUIYHTOAEHTB” và khóa k=”240531” để giải mã

Ta dễ dàng xác định được số hàng của ma trận là 2 (n / m = 12 / 6) và số cột là 6 (m = 6)

Ta có bảng ma trận sau:

Lần lượt ta sẽ chọn cột có giá trị là 0(cột thứ 3) để điền kí tự bản mã theo thứ tự từ trên xuống -> NU được điền, tiếp đó sẽ là cột có giá trị 1(cột số 6)…đến hết cột có giá trị là 5

Chỉ số i	2	4	0	5	3	1
1	H	E	N	T	O	I
2	T	H	U	B	A	Y

⇨ Lấy các kí tự thứ tự tự nhiên của ma trận ta được bản rõ

"HENTOITHUBAY"

2.2.4 Phân tích , đánh giá

Thực chất mã chuyển vị là giữ các ký tự của bản rõ không thay đổi nhưng sẽ thay đổi vị trí của chúng bằng cách sắp xếp lại các ký tự này.

Điều này có nghĩa là tần số xuất hiện của 1 chữ cái trong bản rõ và trong bản mã là như nhau,không thay đổi tần suất

Với độ dài khóa là m , thì số khóa có thể có chính là m!

Với m = 26 ,nghĩa là số khóa có thể có là 26! (mã thay thế).

Việc thám mã mã chuyển vị khi kẻ thám mã biết được độ dài khóa sẽ dò tất cả số khóa có thể có => chỉ riêng mã chuyển vị là sẽ không an toàn

⇨ Giải pháp : Ta đem mã chuyển vị kết hợp với mật mã Vigenere, tức là bản mã thu được sau khi chuyển vị sẽ là bản rõ của mã vigenere,lúc này bản mã thực sự sẽ đi qua hai lần mã hóa

⇨ Với việc kết hợp như vậy ta sẽ tạo được an toàn bởi qua hai lớp khóa , việc tìm ra sự kết hợp 2 mã pháp đã gây rất nhiều khó khăn với việc chỉ cần dựa vào một mật mã đã biết.

⇨ Kẻ thám mã lúc này để tìm được chính xác 2 khóa khác nhau cũng là điều không thể.

Chương 3 : Đề xuất Hệ mật mã kết hợp giữa Vigenere và chuyển vị

3.1. Sự kết hợp hai mã chuyển vị và mã Vigenere

3.1.1. Lý thuyết :

Để thực hiện việc kết hợp này,bên A và bên B phải thống nhất được hai cặp khóa trước mới có thể mã hóa và giải mã.

Trong này em sẽ chọn thông điệp chỉ có các chữ số trong tập 26 chữ.Chuỗi này là các chữ cái liền nhau,không có khoảng trống

Lí do : Nếu chọn thông điệp có các khoảng trống thì sẽ tạo điều kiện cho kẻ thám mã,bởi kí tự khoảng trống(dấu cách) sẽ xuất hiện tần số lớn,đây chính là lỗ hổng để kẻ tấn công có thể đối chiếu thám mã dựa vào tần số xuất hiện và đoán nghĩa được bản rõ như thế nào

Trong trường hợp muốn gõ thông điệp bằng tiếng việt để gửi thì ta sẽ gõ theo kiểu gõ TELEX viết liền không để dấu cách.

3.1.2 Mã hóa

Bước 1 : Em sẽ dùng một thông điệp bản rõ và khóa xác định mã hóa theo Vigenere trước (thứ tự mã hóa theo 5 bước chương 2).

Bước 2 : Sau khi thu được bản mã lần 1, ta sẽ dùng bản mã 1 này và khóa chuyển vị tiến hành mã hóa tiếp lần nữa (thứ tự mã hóa theo 5 bước chương 3)

Bước 3 : Bảng ma trận cuối cùng thu được ta sẽ nhặt các chữ cái theo thứ tự tự nhiên ra sẽ được bản mã hoàn chỉnh do sự kết hợp của 2 mật mã tạo nên

3.1.3 Giải mã

Bên giải mã lúc này cũng đã đảm bảo là biết được đúng 2 khóa chuyển vị và Vigenere.thứ tự giải mã cũng theo 3 bước sau:

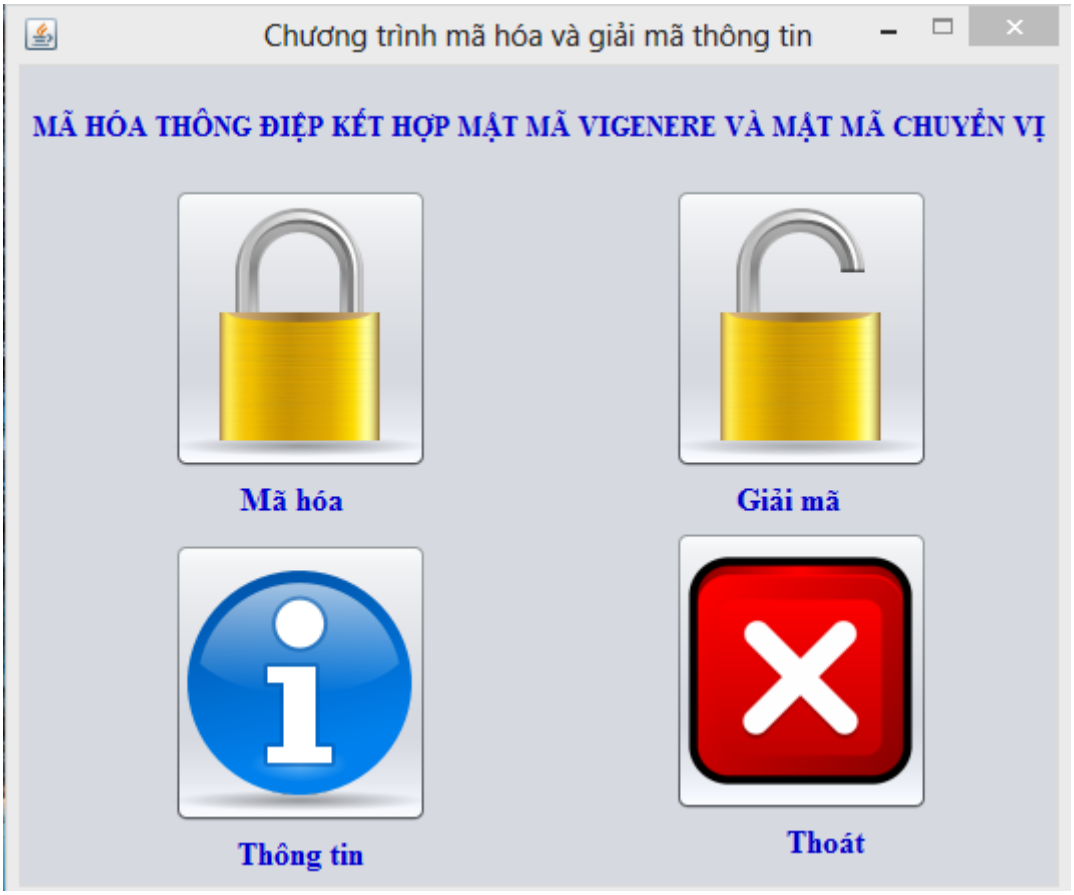
Bước 1 : Em sẽ dùng một thông điệp bản mã vừa được nhận tiến hành giải mã theo mật mã chuyển vị(thứ tự giải mã theo 5 bước chương 3).

Bước 2 : Sau khi thu được bản rõ lần 1, ta sẽ dùng bản rõ 1 này và khóa Vigenere tiến hành giải mã tiếp lần nữa (thứ tự giải mã theo 5 bước chương 2)

Bước 3 : Bảng ma trận cuối cùng thu được ta sẽ nhặt các chữ cái theo thứ tự tự nhiên ra sẽ được bản rõ hoàn chỉnh mà bên A đã gửi

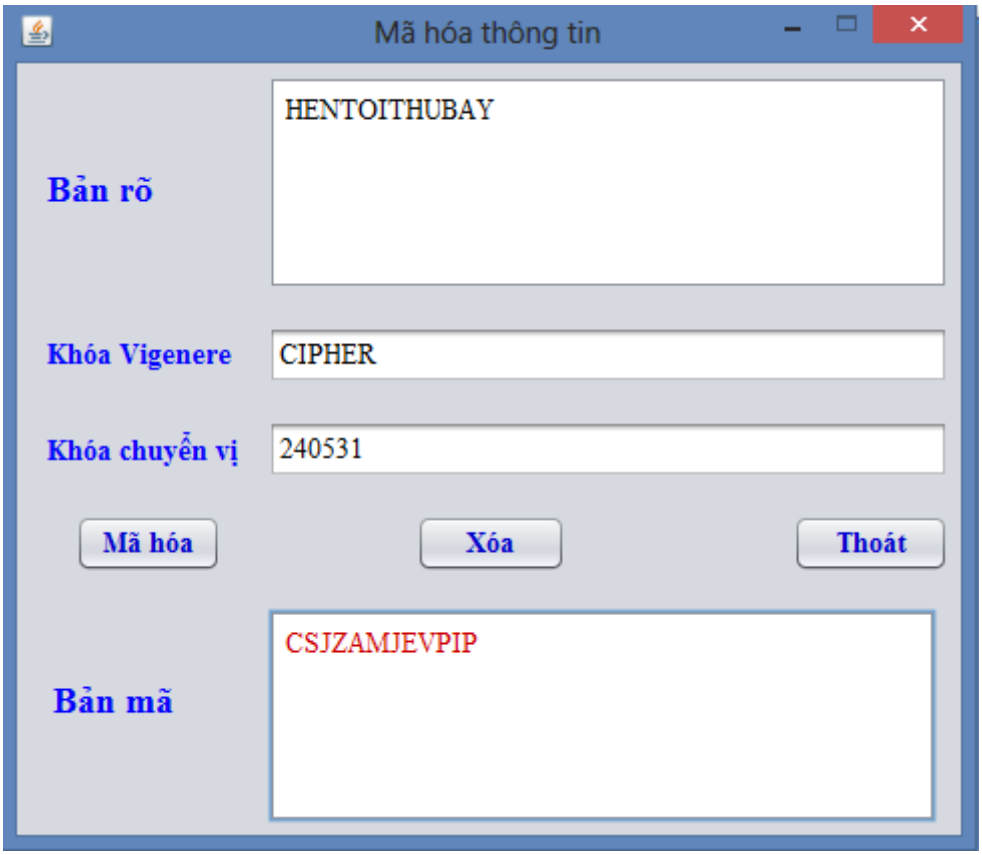
3.2 Chương trình Demo

Chương trình này em viết bằng ngôn ngữ lập trình Java,demo về mã hóa và giải mã dựa vào hai thuật toán trên.

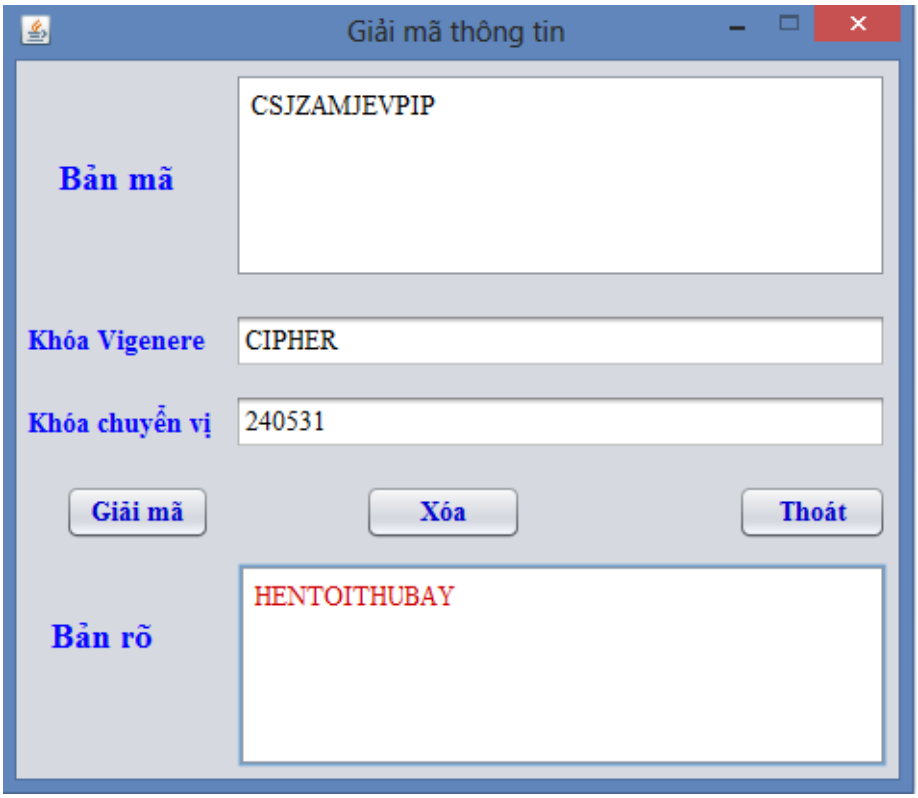


Chương trình có 4 menu:

Mã hóa : Mã hóa thông điệp



Giải mã : Giải mã thông điệp



Thông tin: Tên tác giả,tài liệu tham khảo,khóa đã dùng….

Thoát: Chức năng đóng chương trình

3.3. Mã nguồn

Em xây dựng chương trình gồm 4 form, 1 form giao diện chính và 3 form giao diện phụ thực hiện từng chức năng.

Mã nguồn form 1 : Cipher.java

```
package DO_AN_TOT_NGHIEP;

import javax.swing.JOptionPane;

public class Cipher extends javax.swing.JFrame {
```

```
    public Cipher() {
        initComponents();
    }
    private void initComponents() {
        Button_mahoa = new javax.swing.JButton();
        Button_giaima = new javax.swing.JButton();
        Button_info = new javax.swing.JButton();
        Button_exit = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Chương trình mã hóa và giải mã thông tin");
        setBounds(new java.awt.Rectangle(500, 200, 0, 0));
        setResizable(false);
        Button_mahoa.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        Button_mahoa.setForeground(new java.awt.Color(0, 0, 204));
        Button_mahoa.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/DO_AN_TOT_NGHIEP/lock.png"))); //
NOI18N
        Button_mahoa.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Button_mahoaActionPerformed(evt);
        }
    });

    Button_giaima.setFont(new java.awt.Font("Times New Roman", 1, 14));
Button_giaima.setForeground(new java.awt.Color(0, 0, 204));
    Button_giaima.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/DO_AN_TOT_NGHIEP/unlock.png")));
Button_giaima.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Button_giaimaActionPerformed(evt);
        }
    });

    Button_info.setFont(new java.awt.Font("Times New Roman", 1, 14));
    Button_info.setForeground(new java.awt.Color(0, 0, 204));
    Button_info.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/DO_AN_TOT_NGHIEP/about.png
Button_info.setMaximumSize(new java.awt.Dimension(209, 137));
    Button_info.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Button_infoActionPerformed(evt);
        }
```

```
        });
        Button_exit.setFont(new java.awt.Font("Times New Roman", 1, 14)); Button_exit.setForeground(new java.awt.Color(0, 0, 204));
        Button_exit.setIcon(new javax.swing.ImageIcon(getClass().getResource("/DO_AN_TOT_NGHIEP/exit.png"))); // NOI18N
        Button_exit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Button_exitActionPerformed(evt);
            }
        });
        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(0, 0, 204));
        jLabel2.setText("MÃ HÓA THÔNG ĐIỆP KẾT HỢP MẬT MÃ VIGENERE VÀ MẬT MÃ CHUYỂN VỊ");
        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 16)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(0, 0, 204));
        jLabel1.setText("Mã hóa");
        jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 16)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(0, 0, 204));
        jLabel3.setText("Giải mã");
        jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 16)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(0, 0, 204));
        jLabel4.setText("Thoát");
```

```
        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 16)); // NOI18N
        jLabel5.setForeground(new java.awt.Color(0, 0, 204));
        jLabel5.setText("Thông tin");
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
    private void Button_mahoaActionPerformed(java.awt.event.ActionEvent evt) {
        new Mahoa().setVisible(true);
    }
    private void Button_giaimaActionPerformed(java.awt.event.ActionEvent evt) {
        new Giaima().setVisible(true);
    }
    private void Button_exitActionPerformed(java.awt.event.ActionEvent evt) {
     int chon = JOptionPane.showConfirmDialog(this,"Bạn muốn thoát ??? ", "Thông báo ",
JOptionPane.YES_NO_OPTION);
        if(chon == JOptionPane.YES_OPTION)
        System.exit(0);
    }
    private void Button_infoActionPerformed(java.awt.event.ActionEvent evt) {
       new About().setVisible(true);
    }
    public static void main(String args[]) {
        try {
            for(javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
```

```
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(Cipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Cipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Cipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Cipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Cipher().setVisible(true);
```

```
            }
        });
    }
    private javax.swing.JButton Button_exit;
    private javax.swing.JButton Button_giaima;
    private javax.swing.JButton Button_info;
    private javax.swing.JButton Button_mahoa;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    // End of variables declaration
}
```

Mã nguồn Mahoa.java

```
package DO_AN_TOT_NGHIEP;
import javax.swing.JOptionPane;
public class Mahoa extends javax.swing.JFrame {
    public Mahoa() {
        initComponents();
    }
   private String chuoi="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    public String getChuoi() {
```

```
    return chuoi;
}
public void setChuoi(String chuoi) {
    this.chuoi = chuoi;
}
        //-----------------------------------------
        public int[] mang_chiso(String s){
                char[] s_s = s.toCharArray();
                int[] x = new int[s.length()];
                for (int i = 0; i < s.length(); i++) {
                        x[i] = getChuoi().indexOf(s_s[i]);
                }
                return x;
        }
        //-----------------------------------------
        public String chiso_chuoi(int[] a){
                String s = "";
                char[] chuyen_chuoi = getChuoi().toCharArray();
                for (int i = 0; i < a.length; i++) {
                        s += chuyen_chuoi[a[i]];
                }
                return s;
        }
    public String mahoa_vigenere(String banro,String khoa){
          banro = banro.toUpperCase();
          khoa = khoa.toUpperCase();
                String y = "";
```

```
            int[] x = new int[banro.length()];
            int[] k = new int[banro.length()];
            int[] kq = new int[banro.length()];
            x = mang_chiso(banro);
            k = mang_chiso(khoa);
        int i,j;
            for(i=0,j=0;i<banro.length();i++ ){
                kq[i] = (x[i] + k[j]) % getChuoi().length();
                j = ++j % k.length;
            }
            y = chiso_chuoi(kq);
            return y;
    }
//---------------------------------------------
public static String mahoa_chuyenvi(String banro,String khoa){
    banro = banro.toUpperCase();
    int sohang =0;
    char[] mang_banro = banro.toCharArray();
    // chuyen doi khoa tu String sang int[]
    int[] chuyen_khoa = chuyen_khoa(khoa);

    System.out.println(khoa.length());

    if(banro.length() % khoa.length() == 0){
        sohang = banro.length()/khoa.length();
    }else {
        sohang = banro.length()/khoa.length() + 1;
```

```
        }
        char[][] a = new char[sohang][khoa.length()];
        char[] mang_daydu = new char[sohang*khoa.length()];
        System.arraycopy(mang_banro, 0, mang_daydu, 0, mang_banro.length);
        for (int i = 0; i < sohang; i++) {
            for (int j = 0; j < khoa.length(); j++) {
                a[i][j] = mang_daydu[i * khoa.length() + j];
            }
        }
        //-------- ma hoa ma tran--------------------
        char[][] b = new char[sohang][khoa.length()];
        for (int i = 0; i < sohang; i++) {
            for (int j =0; j < khoa.length(); j++) {
                b[i][j] = a[i][chuyen_khoa[j]];
            }
        }
        //----------- in chuoi ma hoa----------------
        String st = "";
        for (int i = 0; i < sohang; i++) {
            for (int j =0; j < khoa.length(); j++) {
                st += b[i][j];
            }
        }
      //   st = st.trim();
        return st;
    }
    //-------------------------------------------------
```

```
    public static int[] chuyen_khoa(String khoa){
         int[] chuyen_khoa = new int[khoa.length()];
          // chuyen doi khoa tu String sang int[]
          for (int i = 0; i < khoa.length(); i++) {
              chuyen_khoa[i] = khoa.toCharArray()[i] - 48;
          }
          return chuyen_khoa;
    }
    //--------------------------------------------------
    public static int[] khoa_dao(int[] khoa){
       int[] a = new int[khoa.length];
       for (int i = 0; i < a.length; i++) {
          a[khoa[i]] = i;
       }
       return a;
    }
  private void initComponents() {
    jTextField1 = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    ta_banro = new javax.swing.JTextArea();
    jScrollPane2 = new javax.swing.JScrollPane();
    ta_banma = new javax.swing.JTextArea();
    tf_vigenere = new javax.swing.JTextField();
    tf_chuyenvi = new javax.swing.JTextField();
    bt_mahoa = new javax.swing.JButton();
    bt_xoa = new javax.swing.JButton();
```

```
bt_thoat = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();

jTextField1.setText("jTextField1");

jLabel3.setText("jLabel3");

setTitle("Mã hóa thông tin");
setBounds(new java.awt.Rectangle(500, 200, 0, 0));
setResizable(false);

ta_banro.setColumns(20);
ta_banro.setFont(new java.awt.Font("Times New Roman", 0, 14));
ta_banro.setRows(5);
jScrollPane1.setViewportView(ta_banro);

ta_banma.setColumns(20);
ta_banma.setFont(new java.awt.Font("Times New Roman", 0, 14));
ta_banma.setForeground(new java.awt.Color(204, 0, 0));
ta_banma.setRows(5);
jScrollPane2.setViewportView(ta_banma);

tf_vigenere.setFont(new java.awt.Font("Times New Roman", 0, 14));
```

```
tf_chuyenvi.setFont(new java.awt.Font("Times New Roman", 0, 14

bt_mahoa.setFont(new java.awt.Font("Times New Roman", 1, 14));
bt_mahoa.setForeground(new java.awt.Color(0, 0, 204));
bt_mahoa.setText("Mã hóa");
bt_mahoa.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bt_mahoaActionPerformed(evt);
    }
});

bt_xoa.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
bt_xoa.setForeground(new java.awt.Color(0, 0, 204));
bt_xoa.setText("Xóa");
bt_xoa.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bt_xoaActionPerformed(evt);
    }
});
bt_thoat.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
bt_thoat.setForeground(new java.awt.Color(0, 0, 204));
bt_thoat.setText("Thoát");
bt_thoat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bt_thoatActionPerformed(evt);
    }
});
```

```
jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel1.setForeground(new java.awt.Color(0, 0, 255));
jLabel1.setText("Khóa Vigenere");
jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
jLabel2.setForeground(new java.awt.Color(0, 0, 255));
jLabel2.setText("Bản mã");
jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
jLabel4.setForeground(new java.awt.Color(0, 0, 255));
jLabel4.setText("Bản rõ");
jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel5.setForeground(new java.awt.Color(0, 0, 255));
jLabel5.setText("Khóa chuyển vị");
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
   layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
   .addGroup(layout.createSequentialGroup()
      .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
         .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
               .addGroup(layout.createSequentialGroup()
                  .addGap(15, 15, 15)
                  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
                     .addComponent(jLabel1,   javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addGroup(layout.createSequentialGroup()

                    .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE,                    79,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addGap(0, 0, Short.MAX_VALUE)))

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED))

            .addGroup(layout.createSequentialGroup()

                .addGap(18, 18, 18)

                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
79, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(28, 28, 28)))

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)

            .addGroup(layout.createSequentialGroup()

                .addComponent(jScrollPane2)

                .addGap(6, 6, 6))

            .addComponent(jScrollPane1)

            .addComponent(tf_chuyenvi, javax.swing.GroupLayout.DEFAULT_SIZE,
340, Short.MAX_VALUE)

            .addComponent(tf_vigenere)))

        .addGroup(layout.createSequentialGroup()

            .addGap(29, 29, 29)

            .addComponent(bt_mahoa)

            .addGap(97, 97, 97)

            .addComponent(bt_xoa, javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
109, Short.MAX_VALUE)
                .addComponent(bt_thoat, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
107, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(52, 52, 52)
                        .addComponent(jLabel4)))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASE
LINE)
                    .addComponent(tf_vigenere, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel1))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING, false)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(1, 1, 1)
```

```
                .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

                .addComponent(tf_chuyenvi, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)

                    .addComponent(bt_mahoa)

                    .addComponent(bt_thoat))

                .addComponent(bt_xoa))

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  18,
Short.MAX_VALUE)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)

                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

                    .addComponent(jScrollPane2,  javax.swing.GroupLayout.PREFERRED_SIZE,
107, javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addContainerGap())

                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

                    .addComponent(jLabel2)

                    .addGap(56, 56, 56))))

        );

        pack();

    }// </editor-fold>

    private void bt_mahoaActionPerformed(java.awt.event.ActionEvent evt) {

        try {
```

```
            String banro = ta_banro.getText();
            String khoa1 = tf_vigenere.getText();
            String khoa2 = tf_chuyenvi.getText();
            String banma1 = mahoa_vigenere(banro, khoa1);
            String banma2 = mahoa_chuyenvi(banma1, khoa2);
            ta_banma.append(banma2);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Dữ liệu đầu vào không đúng", "Lỗi", JOptionPane.ERROR_MESSAGE);
        }
        catch(Exception e1){
            JOptionPane.showMessageDialog(null, "Dữ liệu bản rõ không hợp lệ", "Lỗi", JOptionPane.ERROR_MESSAGE);
        }
    }
    private void bt_xoaActionPerformed(java.awt.event.ActionEvent evt) {
        ta_banro.setText(null);
        ta_banma.setText(null);
        tf_chuyenvi.setText(null);
        tf_vigenere.setText(null);
  }
    private void bt_thoatActionPerformed(java.awt.event.ActionEvent evt) {
      int chon = JOptionPane.showConfirmDialog(this,"Bạn muốn thoát ??? ", "Thông báo ", JOptionPane.YES_NO_OPTION);
      if(chon == JOptionPane.YES_OPTION){
          dispose();
      }
    }
```

```
   public static void main(String args[]) {
      try {
         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
               javax.swing.UIManager.setLookAndFeel(info.getClassName());
               break;
            }
         }
      } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Mahoa.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
      } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Mahoa.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
      } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Mahoa.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
      } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Mahoa.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
      }
      java.awt.EventQueue.invokeLater(new Runnable() {
         public void run() {
            new Mahoa().setVisible(true);
         }
```

```
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton bt_mahoa;
    private javax.swing.JButton bt_thoat;
    private javax.swing.JButton bt_xoa;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextArea ta_banma;
    private javax.swing.JTextArea ta_banro;
    private javax.swing.JTextField tf_chuyenvi;
    private javax.swing.JTextField tf_vigenere;
    // End of variables declaration
}
```

Mã nguồn Giaima.java

```
package DO_AN_TOT_NGHIEP;
import javax.swing.JOptionPane;
public class Giaima extends javax.swing.JFrame {
    public Giaima() {
        initComponents();
```

```
}
private void initComponents() {
    tf_vigenere = new javax.swing.JTextField();
    tf_chuyenvi = new javax.swing.JTextField();
    bt_giaima = new javax.swing.JButton();
    bt_xoa = new javax.swing.JButton();
    bt_thoat = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    ta_banma = new javax.swing.JTextArea();
    jScrollPane2 = new javax.swing.JScrollPane();
    ta_banro = new javax.swing.JTextArea();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Giải mã thông tin");
    setBounds(new java.awt.Rectangle(500, 200, 0, 0));
    setResizable(false);
    tf_vigenere.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
    tf_chuyenvi.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
    bt_giaima.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
    bt_giaima.setForeground(new java.awt.Color(0, 0, 204));
    bt_giaima.setText("Giải mã");
    bt_giaima.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            bt_giaimaActionPerformed(evt);
        }
    });
    bt_xoa.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
    bt_xoa.setForeground(new java.awt.Color(0, 0, 204));
    bt_xoa.setText("Xóa");
    bt_xoa.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            bt_xoaActionPerformed(evt);
        }
    });
    bt_thoat.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
    bt_thoat.setForeground(new java.awt.Color(0, 0, 204));
    bt_thoat.setText("Thoát");
    bt_thoat.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            bt_thoatActionPerformed(evt);
        }
    });
    jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(0, 0, 255));
    jLabel1.setText("Khóa Vigenere");

    jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
    jLabel2.setForeground(new java.awt.Color(0, 0, 255));
    jLabel2.setText("Bản mã");
    ta_banma.setColumns(20);
```

```
ta_banma.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
ta_banma.setRows(5);
jScrollPane1.setViewportView(ta_banma);
ta_banro.setColumns(20);
ta_banro.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
ta_banro.setForeground(new java.awt.Color(204, 0, 0));
ta_banro.setRows(5);
jScrollPane2.setViewportView(ta_banro);
jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
jLabel4.setForeground(new java.awt.Color(0, 0, 255));
jLabel4.setText("Bản rõ");
jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel5.setForeground(new java.awt.Color(0, 0, 255));
jLabel5.setText("Khóa chuyển vị");
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 79, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 340, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(tf_vigenere, javax.swing.GroupLayout.PREFERRED_SIZE, 340, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(tf_chuyenvi, javax.swing.GroupLayout.DEFAULT_SIZE, 340, Short.MAX_VALUE))

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(bt_giaima)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(bt_xoa, javax.swing.GroupLayout.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```
                    .addComponent(bt_thoat, javax.swing.GroupLayout.PREFERRED_SIZE,
78, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE,
79, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(jScrollPane2)))))
            .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
107, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(50, 50, 50)
                        .addComponent(jLabel2)))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASE
LINE)
                    .addComponent(tf_vigenere, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel1))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 15,
Short.MAX_VALUE)
```

```
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING, false)
                .addGroup(layout.createSequentialGroup()
                    .addGap(1, 1, 1)
                    .addComponent(jLabel5,          javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addComponent(tf_chuyenvi,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(28, 28, 28)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(bt_giaima)
                    .addComponent(bt_thoat))
                .addComponent(bt_xoa, javax.swing.GroupLayout.Alignment.TRAILING))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(18, 18, 18)
                    .addComponent(jScrollPane2,  javax.swing.GroupLayout.PREFERRED_SIZE,
107, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(52, 52, 52)
                    .addComponent(jLabel4)))
            .addGap(57, 57, 57))
        );
        pack();
    }// </editor-fold>
```

```
private String chuoi="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    public String getChuoi() {
        return chuoi;
    }
    public void setChuoi(String chuoi) {
        this.chuoi = chuoi;
    }
    //-----------------------------------------
    public int[] mang_chiso(String s){
            char[] s_s = s.toCharArray();
            int[] x = new int[s.length()];
            for (int i = 0; i < s.length(); i++) {
                    x[i] = getChuoi().indexOf(s_s[i]);
            }
            return x;
    }
    //-----------------------------------------
    public String chiso_chuoi(int[] a){
            String s = "";
            char[] chuyen_chuoi = getChuoi().toCharArray();
            for (int i = 0; i < a.length; i++) {
                    s += chuyen_chuoi[a[i]];
            }
            return s;
    }
    //-----------------------------------------
    public String giaima_vigenere(String banma,String khoa){
```

```
        banma = banma.toUpperCase();
        khoa = khoa.toUpperCase();
        String x="";
        int[] y = new int[banma.length()];
        int[] k = new int[banma.length()];
        int[] kq = new int[banma.length()];
        y = mang_chiso(banma);
        k = mang_chiso(khoa);

        int i,j;
        for(i=0,j=0;i<banma.length();i++ ){
            kq[i] = (y[i] - k[j]) % getChuoi().length();
            if(kq[i] < 0){
                kq[i] = (y[i] + (getChuoi().length() - k[j])) % getChuoi().length();
            }
            j = ++j % k.length;
        }
        x= chiso_chuoi(kq);
        return x;
    }
    //-----------------------------------------------------
    public static String giaima_chuyenvi(String banma,String khoa){
        banma = banma.toUpperCase();
        int sohang =0;
        char[] mang_banma = banma.toCharArray();
        int[] chuyen_khoa = chuyen_khoa(khoa);
        int[] khoa_dao = khoa_dao(chuyen_khoa);
```

```
if(banma.length() % khoa.length() == 0){
	sohang = banma.length()/khoa.length();
}else {
	sohang = banma.length()/khoa.length() + 1;
}
char[][] a = new char[sohang][khoa.length()];
char[] mang_daydu = new char[sohang*khoa.length()];
System.arraycopy(mang_banma, 0, mang_daydu, 0, mang_banma.length);
for(int i1 = banma.length()  ; i1 < (banma.length() - sohang); i1++){
	mang_daydu[i1] = (Character) null;
}
for (int i = 0; i < sohang; i++) {
	for (int j = 0; j < khoa.length(); j++) {
		a[i][j] = mang_daydu[i * khoa.length() + j];
	}
}
//-------- ma hoa ma tran--------------------
char[][] b = new char[sohang][khoa.length()];
for (int i = 0; i < sohang; i++) {
	for (int j =0; j < khoa.length(); j++) {
		b[i][j] = a[i][khoa_dao[j]];
	}
}
//----------- in chuoi ma hoa-----------------
String st = "";
for (int i = 0; i < sohang; i++) {
```

```
                for (int j =0; j < khoa.length(); j++) {
                        st += b[i][j];
                }
        }
        return st;
}
//--------------------------------------------------
public static int[] chuyen_khoa(String khoa){
     int[] chuyen_khoa = new int[khoa.length()];
      // chuyen doi khoa tu String sang int[]
      for (int i = 0; i < khoa.length(); i++) {
         chuyen_khoa[i] = khoa.toCharArray()[i] - 48;
      }
      return chuyen_khoa;
}
//--------------------------------------------------
public static int[] khoa_dao(int[] khoa){
   int[] a = new int[khoa.length];
   for (int i = 0; i < a.length; i++) {
      a[khoa[i]] = i;
   }
   return a;
}
private void bt_giaimaActionPerformed(java.awt.event.ActionEvent evt) {
   try {
      String banma = ta_banma.getText();
      String khoa_vigener = tf_vigenere.getText();
```

```
            String khoa_chuyenvi = tf_chuyenvi.getText();
            String banro1 = giaima_chuyenvi(banma,khoa_chuyenvi);
            String banro2 = giaima_vigenere(banro1.trim(), khoa_vigener);
            ta_banro.append(banro2);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Dữ liệu đầu vào không đúng", "Lỗi", JOptionPane.ERROR_MESSAGE);
        }
        catch(Exception e1){
            JOptionPane.showMessageDialog(null, "Dữ liệu bản rõ không hợp lệ", "Lỗi", JOptionPane.ERROR_MESSAGE);
        }
    }
    private void bt_xoaActionPerformed(java.awt.event.ActionEvent evt) {
        ta_banma.setText(null);
        ta_banro.setText(null);
        tf_chuyenvi.setText(null);
        tf_vigenere.setText(null);
    }
    private void bt_thoatActionPerformed(java.awt.event.ActionEvent evt) {
        int chon = JOptionPane.showConfirmDialog(this,"Bạn muốn thoát ??? ", "Thông báo ", JOptionPane.YES_NO_OPTION);
        if(chon == JOptionPane.YES_OPTION)
        dispose();
    }
    public static void main(String args[]) {
        try {
```

```
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(Giaima.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Giaima.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Giaima.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Giaima.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Giaima().setVisible(true);
        }
    });
}
```

```
    // Variables declaration - do not modify
    private javax.swing.JButton bt_giaima;
    private javax.swing.JButton bt_thoat;
    private javax.swing.JButton bt_xoa;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextArea ta_banma;
    private javax.swing.JTextArea ta_banro;
    private javax.swing.JTextField tf_chuyenvi;
    private javax.swing.JTextField tf_vigenere;
    // End of variables declaration
}
```

3.4 Hướng phát triển

- Hướng phát triển : Từ xây dựng một hệ mật mã kết hợp giữa Vigenere và chuyển vị để mã hóa/giải mã một thông điệp ngắn, ta sẽ mở rộng ra thành mã hóa/giải mã một file lớn bất kì,với các file bất kì,nhằm che giấu đi thông tin trên một phương diện lớn nhằm mục đích đảm bảo an toàn thông tin.

Kết luận

Như vậy trên đây em đã trình bày xong đồ án tốt nghiệp với tên đề tài là :

"Nghiên cứu và xây dựng một thuật toán mã hóa thông điệp nhờ kết hợp giữa mật mã Vigenere và mật mã chuyển vị"

Với nội dung là đi sâu nghiên cứu các hệ mật truyền thống , trên cơ sở đó chọn hai Hệ mật khác nhau và kết hợp với nhau để tạo nên một Hệ mật mới, đồng thời thực hành mã hóa, phân tích, đánh giá thuật toán và kết luận.

Vậy với việc thay vì dùng một mật mã đơn giản cổ điển để mã hóa thông tin mà kẻ thám mã có khả năng tấn công,ta sẽ dùng các mật mã đơn giản đan xen kết hợp lại sẽ thành một hệ mã mới khiến kẻ tấn công không ngờ tới và rất khó có thể phát hiện sự kết hợp của những mật mã nào với nhau. Điều này hướng tới sự an toàn thông tin theo mong muốn của người mã hóa.

Danh mục tài liệu tham khảo

[1] Hồ Văn Canh,Nguyễn Viết Thế : Nhập môn Phân tích thông tin có bảo mật

[2] Bài giảng An Toàn và bảo mật thông tin- Trường Đại học dân lập Hải Phòng