BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

TÌM HIỂU CHUẨN MẬT MÃ DỮ LIỆU (DES) VÀ ỨNG DỤNG VÀO THI TUYỂN ĐẠI HỌC

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

TÌM HIỂU CHUẨN MẬT MÃ DỮ LIỆU (DES) VÀ ỨNG DỤNG VÀO THI TUYỂN ĐẠI HỌC

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Đỗ Thị Phương
Giáo viên hướng dẫn: TS. Hồ Văn Canh
Mã số sinh viên: 1351010046

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XA HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
-------o0o-------

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Đỗ Thị Phương Mã SV: 1351010046

Lớp: CT 1301 Ngành: Công nghệ Thông tin

Tên đề tài: **Tìm hiểu chuẩn mật mã dữ liệu (DES) và ứng dụng vào thi tuyển đại học.**

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung

1. Tìm hiểu mật mã DES.
2. Nghiên cứu bài toán chia sẻ bí mật của Lagrange.
3. Ứng dụng lược đồ chia sẻ bí mật của Lagrange để phân phối khóa.
4. Demo chương trình

b. Các yêu cầu cần giải quyết

1. Đọc tài liệu và hiểu được vấn đề đặt ra, nắm được các phương pháp mã dịch DES một cách thành thạo (cả tiếng việt và tiếng anh).
2. Hiểu được lược đồ chia sẻ bí mật Lagrange.
3. Đọc hiểu được một số tài liệu chuyên môn bằng tiếng Anh
4. Nắm vững một ngôn ngữ lập trình cơ bản (Vb, C#, C++) và giải được bài toán có tính ứng dụng vào thực tiễn.

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Hồ Thị Hương Thơm

Học hàm, học vị: Tiến Sĩ

Cơ quan công tác: Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn:

Người hướng dẫn thứ hai:

Họ và tên: …….

Học hàm, học vị: …….

Cơ quan công tác: ……

Nội dung hướng dẫn: ……..

……

……

……

……

……...

Đề tài tốt nghiệp được giao ngày tháng năm 2013

Yêu cầu phải hoàn thành trước ngày tháng năm 2013

Đã nhận nhiệm vụ: Đ.T.T.N	Đã nhận nhiệm vụ: Đ.T.T.N
Sinh viên	Cán bộ hướng dẫn Đ.T.T.N

TS. Hồ Thị Hương Thơm

Hải Phòng, ngày .....tháng.....năm 2013

HIỆU TRƯỞNG

GS.TS.NGƯT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....

....

.....

....

.....

....

.....

....

.....

.....

.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....

.....

.....

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn:

(Điểm ghi bằng số và chữ)

.....

.....

.....

Ngày.....tháng.....năm 2013

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP

1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế, ...)

2. Cho điểm của cán bộ phản biện

(*Điểm ghi bằng số và chữ*)

.....

.....

.....

Ngày.....tháng.....năm 2013

Cán bộ chấm phản biện

(Ký, ghi rõ họ tên)

MỤC LỤC

LỜI NÓI ĐẦU

Ngày nay, mạng máy tính ngày càng trở lên phổ biến. Mỗi quốc gia đều có mạng riêng với rất nhiều mạng mang tính bộ phận. trên phạm vi toàn cầu, người ta đã dùng mạng Internet một cách thông dụng. Nhiều dịch vụ điện tử như: thư điện tử, chuyển tiền, thương mại điện tử, chính phủ điện tử…đã được áp dụng rộng rãi.

Các ứng dụng trên mạng máy ngày càng trở lên phổ biến, thuận lợi và quan trọng thì yêu cầu về an toàn mạng, về an ninh dữ liệu càng trở lên cấp bách và cần thiết.

Trên thế giới có rất nhiều quốc gia, nhiều nhà khoa học ngiên cứu về vấn đề bảo mật, đưa ra nhiều thuật toán với mục đích thông tin truyền đi không bị lấy cắp hoặc nếu bị lấy cắp thì cũng không sử dụng được. Trong đề tài của em đưa ra một thuật toán đó là thuật toán DES(Data encryption standar) đây là thuật toán chuẩn của Mỹ, được Mỹ và nhiều nước trên thế giới sử dụng, thuật toán này đã được đưa vào sử dụng nhiều năm nhưng vẫn giữ được tính bảo mật của nó. Tuy nhiên với công nghệ phát triển như hiện nay thì thuật toán DES trở lên không được an toàn tuyệt đối nữa, người ta đã đưa ra thuật toán 3DES dựa trên nền tảng của thuật toán DES nhưng số bít được mã hóa tăng lên.

Mã hóa và các lược đồ chia sẻ bí mật có thể được ứng dụng trong rất nhiều lĩnh vực ví dụ: phát hành thẻ ATM trong ngân hang, đấu tầu từ xa, trong thi tuyển sinh, trong lĩnh vực quân sự… Trong đề tài của em đề cập tới một lĩnh vực đó là ứng dụng trong thi tuyển sinh đại học.

Vấn đề thi tuyển sinh đại học ở nước ta trở thành gánh nặng cho ngành Giáo Dục và các ban ngành khác liên quan. Nó làm tổn hại về kinh tế và công sức không chỉ đối với các ban ngành tham gia tổ chức kỳ thi mà ngay cả đối với các thí sinh dự thi, nhưng đó là điều bắt buộc phải được tổ chức hàng năm. Do vậy làm sao để giảm thiểu các khâu trong thi tuyển sinh mà vẫn đảm bảo tính công bằng và chính xác là điều cần thiết, theo tôi để làm được điều đó ta nên ứng dụng công nghệ thông tin vào việc thi tuyển sinh đại học, một trong những ứng dụng đó là ứng dụng lược đồ chia sẻ bí mật vì nó đảm bảo được tính bí mật và chính xác mà trong thi tuyển sinh hai điều đó là quan trọng nhất.

Phạm vi luận văn đề cập đến mật mã, thuật toán DES, lược đồ chia sẻ bí mật và ứng dụng của chúng trong thi tuyển sinh.

Luận văn gồm 4 chương:

Chương 1: Mật mã cổ điển: chương này giới thiệu về khái niệm và định nghĩa mật mã, một số mã cổ diển.

Chương 2: thuật toán DES: chương này nói về mã hóa và giải mã trong thuật toán DES, các vấn đề xung quanh DES.

Chương 3: Chia sẻ bí mật: chương này nói về khái niệm chia sẻ bí mật, phương thức chia sẻ và khôi phục khóa bí mật.

Chương 4: Ứng dụng thuật toán DES và lược đồ chia sẻ bí mật vào thi tuyển sinh: Chương này nói vè phần ứng dụng và mô phỏng lược đồ chia sẻ bí mật bằng ngôn ngữ C.

Để hoàn thành luận văn này, trước hết em xin chân thành cảm ơn TS Hồ Văn Canh - người đã trực tiếp hướng dẫn, cung cấp tài liệu và đóng góp nhiều ý kiến cho luận văn. Em cũng xin chân thành cảm ơn các thầy cô giáo, các cán bộ khoa công nghệ thông tin trường đại học Dân Lập Hải Phòng đã tận tình giảng dạy, giúp đỡ em trong suốt khóa học.

CHƯƠNG 1: MẬT MÃ CỔ ĐIỂN

1.1 KHÁI NIỆM VÀ ĐỊNH NGHĨA VỀ MẬT MÃ

1.1.1 Khái niệm

- Chức năng cơ bản của mật mã là tạo ra khả năng liên lạc trên một kênh không mật cho hai người sử dụng (tạm gọi là A và B) sao cho đối phương C không thể hiểu được thông tin truyền đi.
- Kênh liên lạc có thể là một đường dây điện thoại hoặc một mạng máy tính. Thông tin mà A muốn gửi cho B bản rõ có thể là một văn bản tiếng Anh, các dữ liệu bằng số hoặc bất cứ tài liệu nào có cấu trúc tùy ý.
- A sẽ mã hóa bản rõ bằng một khóa đã được xác định trước và gửi bản mã kết quả trên kênh. C có bản mã thu trộm được trên kênh xong không thể xác định nội dung của bản rõ, nhưng B (người biết khóa mã) có thể giải mã và thu được bản rõ.

Ta sẽ mô tả hình thức nội dung bằng cách dùng khái niệm toán học như sau:

1.1.2 Định nghĩa

Một hệ mật mã là một bộ 5 (P, C, K, E,D) thỏa mãn các điều kiện sau:

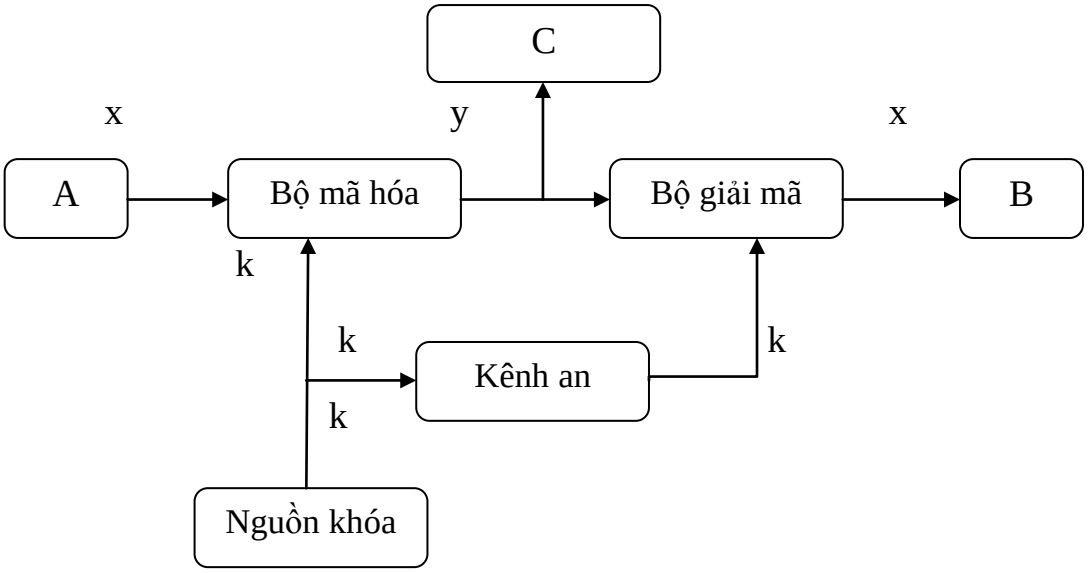
1. *P là một tập hữu hạn các bản rõ có thể*
2. *C là một tập hữu hạn các bản mã có thể*
3. *K (không gian khóa) là tập hữu hạn các khóa có thể.*
4. *Đối với mỗi k ϵ K có một quy tắc mã e_k: P→ C và một quy tắc giải mã tương ứng d_k ϵ D. Mỗi e_k: P→ C và d_k: C→ P là những hàm sao cho: $d_k(e_k(x)) = x$ với mọi bản rõ x ϵ P.*

Trong đó chúng ta cần lưu ý tính chất 4: Nội dung của nó là nếu một bản rõ x được mã hóa bằng e_k và bản mã nhận được sau đó được giải mã bằng d_k thì ta phải thu được bản rõ ban đầu x.

Giả sử ta có bản rõ cần truyền đi là: $x= x_1, x_2,\ldots,x_n$ với số nguyên $n\geq 1$ nào đó. Ở đây mỗi kí hiệu của mỗi bản rõ $x_i \epsilon P, 1\leq i \leq n$. Mỗi x_i sẽ được mã hóa bằng quy tắc mã e_k với khóa k xác định trước đó.

Bản mã thu được là: $y=y_1, y_2,\ldots,y_n$. Trong đó $y_k=e_k(x_i)$ i=1,2,…,n còn k ϵ K. Khi Bob nhận được $y_1, y_2,\ldots,y_n$ anh ta sẽ giải mã bằng hàm giải mã d_k và thu được bản rõ gốc $x= x_1, x_2,\ldots,x_n$.

Hình 1.1 là một ví dụ về một kênh liên lạc



Hình 1.1. Kênh liên lạc

Rõ ràng là trong trường hợp này hàm mã hóa phải là hàm đơn ánh(tức là ánh xạ 1-1), nếu không việc giải mã sẽ không thực hiện được một cách tường minh.

Ví dụ

$$y= e_k(x_1)= e_k(x_2)$$

trong đó $x_1 \neq x_2$, B sẽ không có cách nào đẻ biết liệu bản rõ là x_1 hay x_2.

1.2 MỘT SỐ MÃ HÓA ĐƠN GIẢN

- Mã dịch vòng
- Mã thay thế
- Mã Affine
- Mã Hill
- Mã chuyển vị

CHƯƠNG 2: CHUẨN MÃ DỮ LIỆU (DES)

Các hệ mật mã truyền thống ở chương 1 có một số ưu điểm là mã hóa và giải mã bằng thủ công được thực hiện rất dễ dàng, việc trao đổi khóa mã cũng rất đơn giản bằng thủ công hoặc bằng qui ước. Song với lượng thông tin quá phong phú như hiện nay và với mạng truyền thông như hiện nay thì mật mã thủ công vừa không đảm bảo bí mật. Mặt khác các thuật toán mã hóa thủ công đòi hỏi phải tuyệt đối giữ bí mật… Như vậy mật mã thủ công đòi hỏi bảo mật cả thuật toán mã hóa và cả khóa mã.

Sau những năm 70 của thế kỉ trước, các nhà toán học đã nghiên cứu và tạo ra nhiều phương thức mật mã với tốc độ mã hóa rất nhanh (hàng trục thậm chí hàng trăm kilo Byte trong một giây) và người ta chỉ cầm giữ bí mật khóa mã và mã hóa được mọi dữ liệu tùy ý. Đó là một bước tiến vĩ đại của kĩ thuật mật mã. Trong đó mã DES (Data Encryption Standard) là một điển hình của bước tiến này. Chương này em muốn mô phỏng mã hóa và giải mã của thuật toán DES.

2.1 Mô tả DES (Data Encryption Standard)

DES mã hóa một xâu bít x:

Bản rõ x độ dài 64 bít.

Khóa k độ dài 56 bit.

Bản mã y nhận được cũng là một xâu bit có độ dài 64 bit.

Thuật toán tiến hành theo 3 giai đoạn:

1. Với bản rõ cho trước x, một xâu bit x_0 sẽ được tạo ra bằng cách hoán vị các bit của x theo phép hoán vị cố định ban đầu IP.

Ta viết : $x_0=IP(X)=L_0R_0$, trong đó L_0 gồm 32 bit đầu và R_0 là 32 bit cuối.

2. Sau đó tính toán 16 lần lặp theo một hàm xác định. Ta sẽ tính được L_iR_i, $1 \leq i \leq 16$ theo qui tắc sau:

$L_i = R_{i-1}$

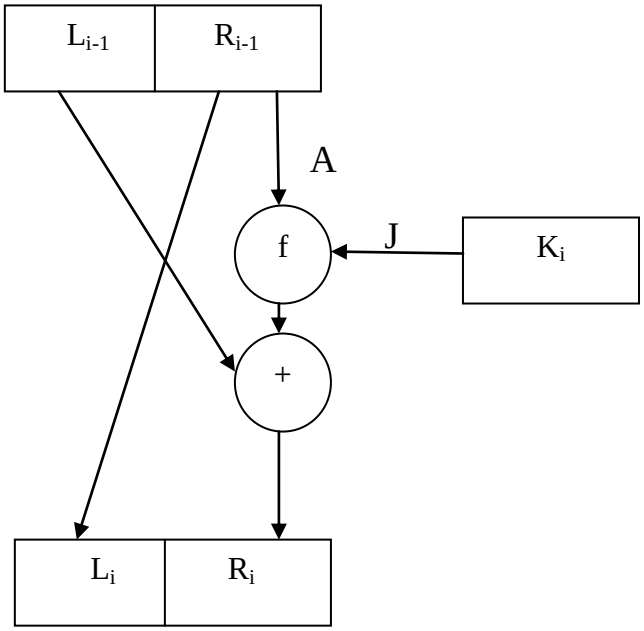
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

Trong đó $\oplus$ kí hiệu cộng theo modulo 2 của 2 xâu bit.

f là một hàm mà của R_{i-1}, k_i mô tả sau.

k_i là các xâu bit độ dài 48 bit được tính như hàm của khóa k. (Trên thực tế mỗi k_i là một phép chọn hoán vị bit trong k).

3. Áp dụng phép hoán vị ngược IP^{-1} cho xâu bit $R_{16}L_{16}$ ta thu được bản mã y.Tức là $y = IP^{-1}(R_{16} L_{16})$.



Hình 2.1. Một vòng (vòng thứ i) của DES.

2.2 Các bước thực hiện:

2.2.1 Cách tính biến x_0**:**

Hoán vị biến x theo phép hoán vi ban đầu IP(X)

$x_0 = IP(X) = L_0R_0$

Bảng 2.1.Bảng IP

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

Theo bảng 2.1 này có nghĩa là bit thứ 58 của x là bit đầu tiên của IP(x), bit thứ 50 của x là bit thứ 2 của IP(x), bit ở vị trí thứ 7 là bit cuối của IP(x).

2.2.2 Cách tính L_iR_i:

Để tính L_iR_i trước hết ta phải xác định hàm f

2.2.2.1 Các biến trong hàm f:

Có 2 biến vào

- Biến thứ nhất R_{i-1}là xâu bit độ dài 32
- Biến thứ hai J là xâu bít độ dài 48
- Đầu ra của f là một xâu bit độ dài 32 bít. Hàm f thực hiện qua các bước sau:

Bước 1: Xác định biến thứ nhất (biến A):

Xâu bit của R_{i-1} được mở rộng thành một xâu bit có độ dài 48 theo một hàm mở rộng cố định E.

Bảng 2.2. Bảng chọn E bit

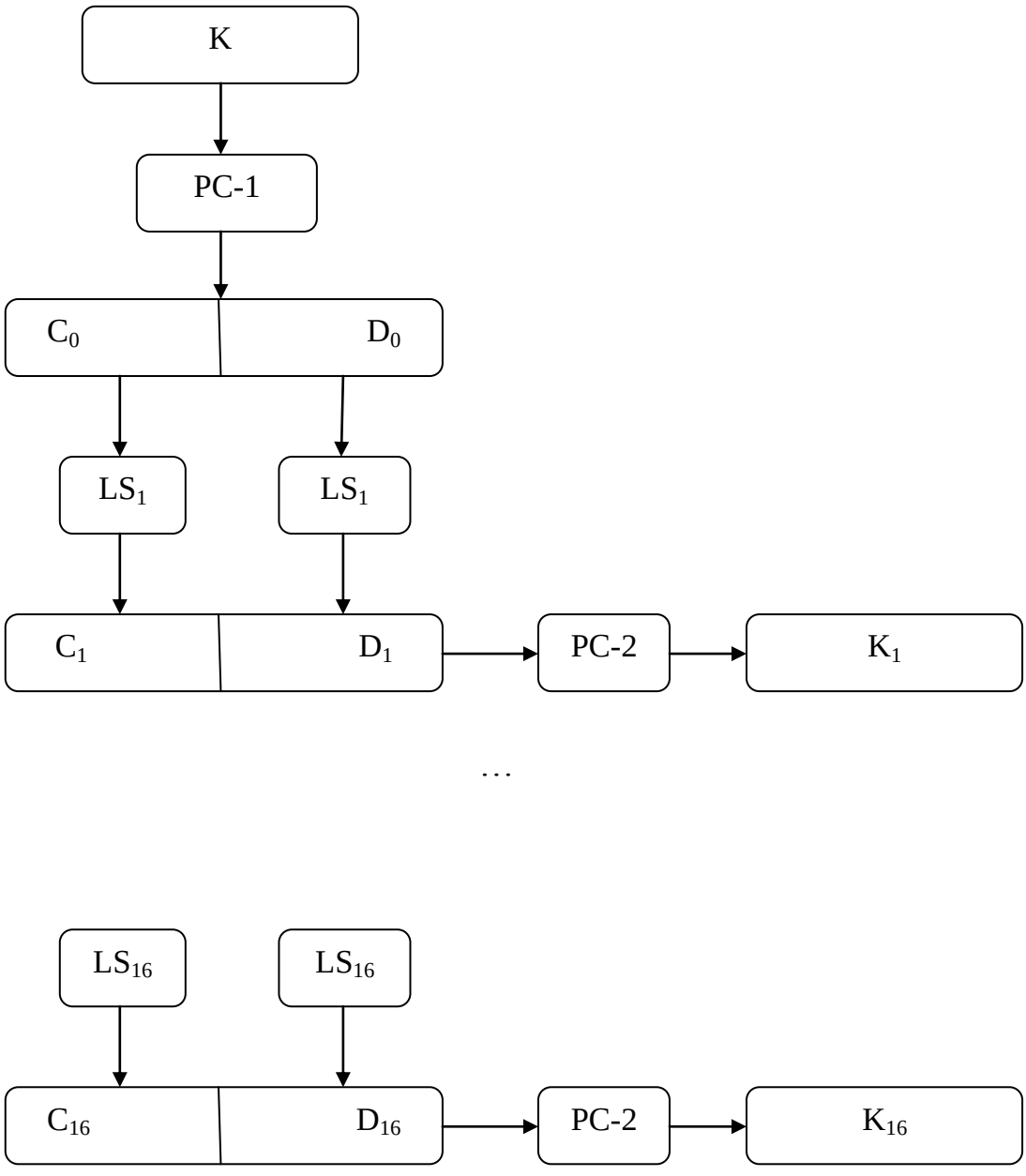
Bảng chọn E bit					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

$E(R_{i-1})$ gồm 32 bit của R_{i-1} (được hoán vị theo cách cố định) với 16 bit xuất hiện 2 lần.Theo bảng E bit ở vị trí thứ 32 là bit đầu tiên của $E(R_{i-1})$, ở vị trí thứ 1 là bit thứ 2 và bit cuối của $E(R_{i-1})$.

Bước 2: Xác định biến thứ hai (biến J)

Biến J thực chất là phép hoán vị và dịch vòng của xâu bit khóa k

Thuật toán tạo 16 khóa con k1, k2,…k_{16}



Hình 2.2. Sơ đồ tạo khóa k

Theo sơ đồ hình 2.2 trên việc xác định k_i được thực hiện như sau:

Với khóa k 64 bit cho trước hoán vị thực chất chỉ có 56 bit dùng để tạo k_i với i từ 1 đến 16.Tránh các bit ở vị trí 8,16,24,32,40,48,56,64.

Theo phép hoán vị cố định PC-1 ta viết:

PC-1(K) = C_0D_0. Trong đó C_0 là 28 bit đầu và D_0 là 28 bit cuối.

Mỗi phần sẽ được xử lí một cách độc lập.

$C_i = LS_i (C_i -1)$

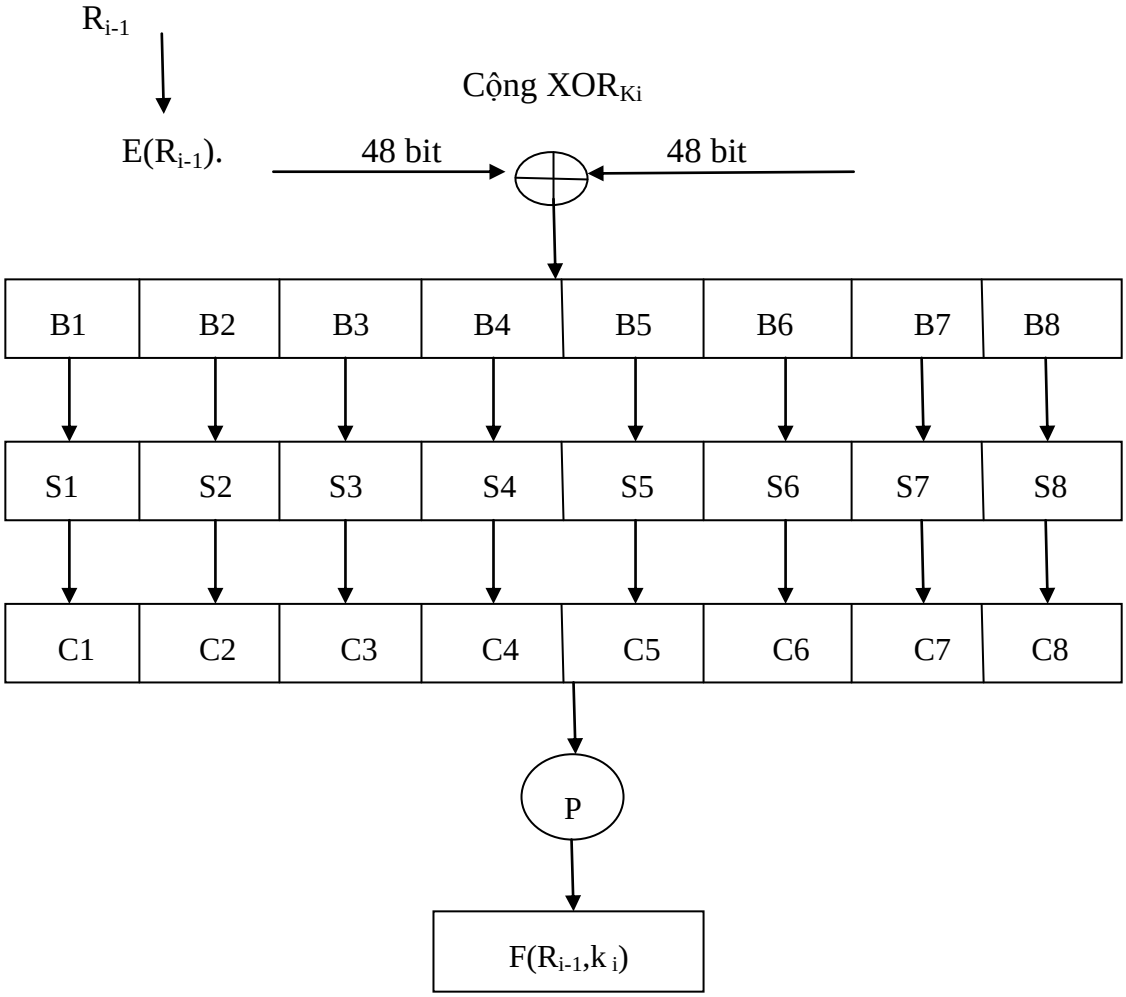
$D_i = LS_i (C_i -1)$ với $1 \leq i \leq 16$

- LS_i biểu diễn phép dịch bit vòng sang trái 1 hoặc 2 vị trí tùy thuộc vào i. Sang trái 1 bit nếu i= 1, 2, 9, 16 hoặc sang trái 2 bit nếu I thuộc các vị trí còn lại.

$k_i = PC-2 (C_i - D_i)$.

- PC-2 là hoán vị cố định sẽ hoán vị chuỗi $C_i - D_i$ 56 bit thành chuỗi 48 bit.

2.2.2.2 Cách tính hàm f:



Hình 2.3.Sơ đồ hoạt động của hàm f(R_{i-1},k_i):

Sau khi mở rộng R_{i-1} bởi hàm mở rộng E để chuyên R_{i-1} gồm 32 bit thành $E(R_{i-1})$ gồm 48 bit, $E(R_{i-1})$ cộng XOR với khóa k_i cho ra là:

$$E(R_{i-1}) + k_i = B = B_1B_2\ldots B_8$$

B gồm 48 bit được phân thành 8 khối (block) như nhau và mỗi block B_i, $i = \overline{1,8}$ có độ dài 6 bit.

Sau đó mỗi B_i được đưa vào hộp S_i, $i = \overline{1,8}$ và biến đổi để cho đầu ra là C_i gồm 4 bit ($i = \overline{1,8}$). Sự biến đổi này được thực hiện như sau:

Giả sử B_i gồm 6 bit là $B_i = b_{i1}b_{i2}\ldots b_{i6}$. Khi đó $b_{i1}b_{i6}$ được nhập thành cặp, $b_{i1}b_{i6}$ được chuyển thành số thập phân từ 1 đến 3. Ví dụ:

$b_{i1}b_{i6} = 00 \rightarrow 0$

$b_{i1}b_{i6} = 01 \rightarrow 1$

$b_{i1}b_{i6} = 10 \rightarrow 2$

$b_{i1}b_{i6} = 11 \rightarrow 3$

Còn $b_{i2}b_{i3}$ $b_{i4}b_{i5}$ của B_i được chuyển thành số thập phân từ 0 đến 15 như sau:

Bảng 2.3. Bảng chuyển đổi giá trị.

$b_{i2}b_{i3}$ $b_{i4}b_{i5}$	Số tự nhiên
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Số nguyên dương $r_i = b_{i1}b_{i6}$ và $s_i = b_{i2}b_{i3}\, b_{i4}b_{i5}$ chính là tọa độ (hoành tung) của ma trận S_i từ t_i chuyển thành $C_i = C_{i1}C_{i2}\, C_{i3}C_{i4}$ với $C_{ij} \epsilon \{0,1\}$ i= i =$\overline{1,8}$, j =$\overline{1,4}$.

Vậy đầu ra của hộp S là:

$C_1C_2 \ldots C_8$ mỗi C_i gồm 4 bit tọa thành khối $C = C_1C_{2\ldots}C8$ gồm 32 bit. 32 bit này được đưa vào ma trận chuyển vị P để cho đầu ra cũng là 32 bit. Đó là đầu ra của hàm $f(R_{i-1},K_i)$.

Ta có thể trình bày cụ thể cách tính hàm f như sau:

Với xâu bit có độ dài 6 bit (kí hiệu $B_i= b_1b_2\ldots b_6$), ta tính được $S_j(B_j)$ như sau: Hai bit b_ib_6 xác định biểu diễn nhị phân của hàng r của S_j ($0 \le r \le 3$) và 4 bit $b_2b_3\, b_4b_5$ xác định biểu diễn nhị phân của cột c của S_j ($0 \le c \le 15$). Khio đó $S_j(B_j)$ sẽ xác định phần tử $S_j(r,c)$; phần tử này viết dưới dạng nhị phân là một xâu bit có độ dài là 4.(Bởi vậy mỗi S_j có thể coi là một hàm mã mà đầu vào là một xâu bit có độ dài 2 và một xâu bit độ dài 4, còn đầu ra là một xâu bit cố độ dài 4). Bằng cách tương tự tính các $C_j= S_j(B_j)$, $1 \le j \le 8$.

Thật vậy mỗi chuỗi B là một chuỗi 6 bit trong đó bit đầu và bit cuối được dùng để xác định vị trí của hàng trong phạm vi từ 0 đến 3 (hoặc từ 00 đến 11) bốn bit giữa dùng để xác định vị trí của cột trong phạm vi từ 0 đến 15 (hoặc từ 0000 đến 1111). Sauk hi xác định được vị trí của hàng và cột ta đối chiếu trong bảng S được một số thập phân ,quy đổi số thập phân đó ra giá trị nhị phân ta được C_j.

Ví dụ: Bit đầu vào của B là B = 100110

Bit đầu và cuối là “10” cho ta thứ tự của hàng là 2, bốn bit giữa là “0011” cho ta thứ tự của cột là 4. Đối chiếu với hộp S_1 xuất hiện số 14, chuyển 14 sang nhị phân 14= 1110, Vậy S(B) = S(100110)= 1110.

Tám hộp S là:

S_1															
14	4	13	1	2	15	11	8	3	10	3	12	5	9	1	7
1	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	5	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	15	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	11	7	6	0	8	13

S_7															
4	11	12	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	5	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Sau khi thực hiện các phép đối chiếu hộp S ta được các giá trị S_iB_i

Tính hàm $f = P(S_1(B_1)) = S_2(B_2)…….. S_8(B_8)$ thực hiện theo phép hoán vị P

Phép hoán vị P có dạng:

Bảng 2.4. Bảng hoán vị P

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Theo bảng 2.4 thì bit thứ 16 và bit thứ 7 của xâu bit S_jB_j lần lượt là bít thứ nhất và bit thứ 2 của hàm f…

L_iR_i được xác định theo công thức sau:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} + f(R_{i-1}, k_i)$$

2.2.3 Xác định bản mã y**:**

Sau khi xác định được L_iR_i ta thu được $L_{16}R_{16}$, vậy bản mã y được xác định theo công thức:

$$y = IP^{-1}(R_{16}, L_{16})$$

Bảng 2.5. Bảng IP^{-1}

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

Ví dụ: Ta có bản rõ M= 0123456789ABCDEF

Khóa k=133457799BBCDFF1

Giải

Bước 1: Tìm x_0= IP(X)= L_0R_0

M=0000.0001. 0010.0011. 0100.0101. 0110.0111. 1000.1001. 1010.1011. 1100.1101. 1110.1111

16 32

L=0000.0001. 0010.0011. 0100.0101. 0110.0111

48 64

R= 1000.1001. 1010.1011. 1100.1101. 1110.1111

Thực hiện phép hoán vị IP

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

IP(M)= 1100.1100. 0000.0000. 11001100. 1111.1111. 1111.0000. 1010.1010. 1111.0000.1010.1010

L_0 = 1100.1100. 0000.0000. 11001100. 1111.1111

R_0 = 1111.0000. 1010.1010. 1111.0000.1010.1010

Bước 2: Xác định khóa k_i

k= 0001.0011. 0011.0100. 0101.0111. 0111.1001. 1001.1011. 1011.1100. 1101.1111. 1111.1001

Hoán vị khóa k theo phép hoán vị PC-1 ta thu được PC-1(k):

Bảng 2.5. Bảng PC-1

PC-1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC-1(k) = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

C_0 = 1111000 0110011 0010101 0101111

D_0 = 0101010 1011001 1001111 0001111

Tìm $C_iD_{i:}$

$$C_i = S_i(C_{i-1})$$

$$D_i = S_i(D_{i-1})$$

Trong đó: S_i là sự dịch chuyển C_{i-1}, D_{i-1} đi 1 sang trái với i= 1,2,9,16 và dịch 2 với các i còn lại.

C_1 = 1110000 1100110 0101010 1011111

D_1 = 1010101 0110011 0011110 0011110

C_2 = 1100001 1001100 1010101 0111111

D_2 = 0101010 1100110 0111100 0111101

C_3 = 0000110 0110010 1010101 1111111

D_3 = 0101011 0011001 1110001 1110101

C_4 = 0011001 1001010 1010111 1111100

D_4 = 0101100 1100111 1000111 1010101

C_5 = 1100110 0101010 1011111 1110000

D_5 = 0110011 0011110 0011110 1010101

C_6 = 0011001 0101010 1111111 1000011

D_6 = 1001100 1111000 1111010 1010101

C_7 = 1100101 0101011 1111110 0001100

D_7 = 0110011 1100011 1101010 1010110

C_8 = 0010101 0101111 1111000 0110011

D_8 = 1001111 0001111 0101010 1011001

C_9 = 0101010 1011111 1110000 1100110

D_9 = 0011110 0011110 1010101 0110011

C_{10} = 0101010 1111111 1000011 0011001

D_{10} = 1111000 1111010 1010101 1001100

C_{11} = 0101011 1111110 0001100 1100101

D_{11} = 1100011 1101010 1010110 0110011

C_{12} = 0101111 1111000 0110011 0010101

C_{13} = 0111111 1100001 1001100 1010101

D_{13} = 0111101 0101010 1100110 0111100

C_{14} = 1111111 0000110 0110010 1010101

D_{14} = 1110101 0101011 0011001 1110001

C_{15} = 1111100 0011001 1001010 1010111

D_{15} = 1010101 0101100 1100111 1000111

C_{16} = 1111000 0110011 0010101 0101111

D_{16} = 0101010 1011001 1001111 0001111

Cuối cùng các khóa con K_i thu được từ C_i ,D_i qua hoán vị PC-2 cho trước.

Bảng 2.6. Bảng PC-2

PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

k_1 = 000110 110000 001011 101111 111111 000111 000001 110010

k_2 = 011110 011010 111011 011001 110110 111100 100111 100101

k_3 = 010101 011111 110010 001010 010000 101100 111110 011001

k_4 = 011100 101010 110111 010110 110110 110011 010100 011101

k_5 = 011111 001110 110000 000111 111010 110101 001110 101000

K_6 = 011000 111010 010100 111110 010100 000111 101100 101111

k_7 = 111011 001000 010010 110111 111101 100001 100010 111100

k_8 = 111101 111000 101000 111010 110000 010011 101111 111011

k_9 = 111000 001101 101111 101011 111011 011110 011110 000001

k_{10} =101100 011111 001101 000111 101110 100100 011001 001111

k_{11}= 001000 010101 111111 010011 110111 101101 011110 000110

k_{12}= 011101 010111 000111 110101 100101 000110 011111 101001

k_{13}= 100101 111100 010111 010001 111110 101011 101001 000001

k_{14}= 010111 110100 001110 110111 111100 101110 011100 111010

k_{15}= 101111 110011 000110 001101 001111 010011 111100 001010

k_{16}= 110010 110011 110110 001011 000011 100001 011111 110101

Bước 3:Tìm hàm f(R_{i-1}, k_i)

Tính $S_1(B_1) = S_1(011000)$.

Bit đầu và bit cuối là "00" cho ta hàng 0

Bốn bit giữa "1100"=12 cho ta cột 12

Chiếu hàng 0 cột 12 vào bảng S_1 cho ta giá trị là 5= "0101"

Vậy $S_1(011000)$ = "0101"

Tính $S_2(B_2) = S_2(010001)$ thực hiện hoán vị $S_1(B_1)$ theo phép hoán vị S_1

Bit đầu và bit cuối là "01" cho ta hàng 1

Bốn bit giữa "1000"=8 cho ta cột 8

Chiếu hàng 1 cột 8 vào bảng S_2 cho ta giá trị là 12="1100"

Vậy $S_2(010001)$= "1100".

Tính tương tự ta có $S_1(B_1) = S_2(B_2) = S_3(B_3) = S_4(B_4) = S_5(B_5) = S_6(B_6) = S_7(B_7) = S_8(B_8)$ =0101 1100 1000 0010 1011 0101 1001 0111

Tính hàm f= P $S_1(B_1) = S_2(B_2)$ …… $S_8(B_8)$ thực hiện theo phép hoán vị P

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

f= P ($S_1(B_2) = S_2(B_2)$ …….. $S_8(B_8)$ = P (0101 1100 1000 0010 1011 0101 1001 0111) = 0010 0011 0100 1010 1010 1001 1011 1011

$R_1 = L_0 + f(R_0, k_1)$ = 1100 1100 0000 0000 1100 1100 1111 1111

+ 0010 0011 0100 1010 1010 1001 1011 1011

= 1110 1111 0100 1010 0110 0101 0100 0100

Tương tự ta tính : $R_2 = L_1 + f(R_1, k_2)$

……

$R_{16} = L_{15} + f(R_{15}, k_{16})$

Ta thực hiện 16 lần vòng lặp thu được $R_{16}L_{16}$

L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100

R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101

$R_{16}L_{16}$ = 0000 1010 0100 1100 1101 1001 1001 0101 0100 0011 0100 0010 0011 0010 0011 0100

Thực hiện phép hoán vị ngược $IP(IP^{-1})$

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

Ta được IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101. Chuyển IP^{-1} sang dạng hexa ta thu được bản mã : 85E823540F0AB405

2.3 Giải mã DES

Tương tự như mã hóa, để giải mã một chuỗi kí tự đã bị mã hóa ta cũng làm tương tự theo các bước trên, tuy nhiên hệ thống khóa lúc này đã được tạo theo chiều ngược lại.

2.3.1 Thuật toán

Mã hóa	Giải mã
- Cho bản rõ x - $x_0 = IP(x) = L_0 R_0$ - i = 1,2,3,….16 $L_i = R_{i-1}$ $R_i = L_{i-1} \wedge f(R_{i-1},K_i)$ - $y_0 = (R_{16} L_{16})$ - $y = IP^{-1}(y_0)$	- Cho bản mã y - $y_0 = IP(y) = R_{16} L_{16} = L'_0 R'_0$ - i = 1,2,3,….16 $L_i = R_{i-1}$' $R_i = L'_{i-1} \wedge f(R'_{i-1},K_{17-i})$ - $x_0 = (R'_{16} L'_{16})$ - $x = IP^{-1}(x_0)$

2.3.2 Chứng minh thuật toán

Có bản mã y

$y_0 = IP(y) = IP(IP^{-1}(R_{16} L_{16})) = R_{16} L_{16} = L'_0 R'_0$

Vậy $L'_0 = R_{16}$, $R'_0 = L_{16}$;

Với i=1

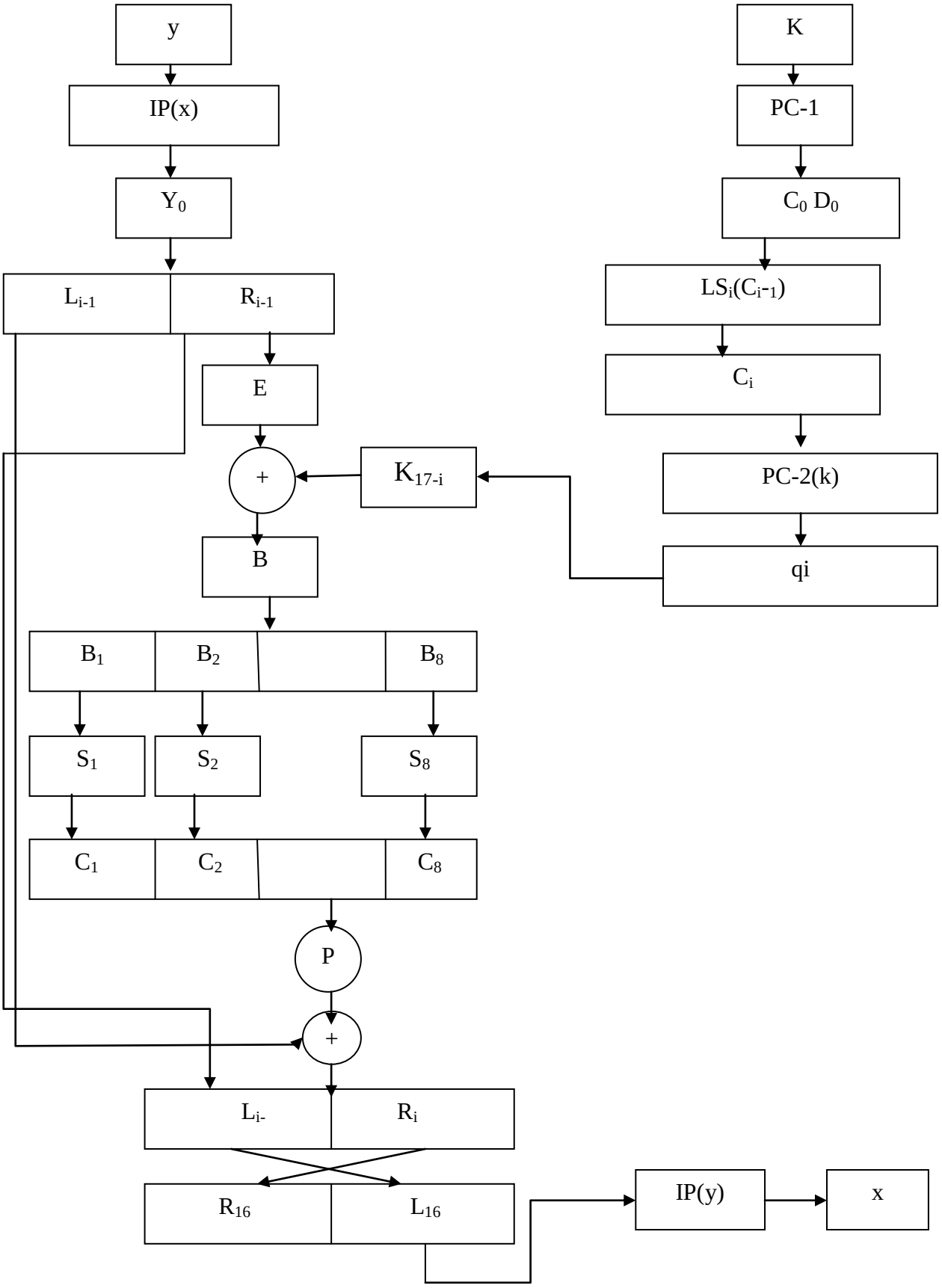
$L'_1 = R'_0 = L_{16} = R_{15}$;

$R'_1 = L'_0 \wedge f(R'_0,k_{16}) = R_{16} \wedge f(L_{16} k_{16})$

$= L_{15} \wedge f(R_{15},k_{16}) \wedge f(R_{15},k_{16}) = L_{15}$;

Tóm lại:

$$L'_1 = R_{15} ; \ R'_1 = L_{15};$$



Hình 2.4. Sơ đồ giải mã DES

Từ đó ta thấy , thuật toán giải mã chỉ khác thuật toán mã hóa ở chỗ tạo hệ thống khóa, nếu mã hóa tạo khóa từ k_1, k_2…, k_{16} thì giải mã tạo hệ thống khóa từ k_{16}, k_{15}…, k_1. Việc này được trình bày cụ thể trong hình 2.4 sơ đồ giải mã DES.

2.4 Các vấn đề xung quanh DES

2.4.1 Những ý kiến phản hồi

Khi DES được đề nghị như một tiêu chuẩn thì đã có những lời phê bình, sự phản hồi có liên quan đến các hộp S. Tất cả các sự tính toán trong DES, ngoại trừ các hộp S, đều là tuyến tính. Các hộp S, thành phần phi tuyến trong hệ thống mật mã là sống còn đối với sự an toàn của hệ thống. Tuy nhiên, tiêu chuẩn thiết kế của hộp S chưa được hiểu hết, một số người gợi ý rằng những hộp S này có chứa đựng những cửa sập còn ẩn dấu ở đâu đó. Những cửa sập này có thể cho phép cục an ninh quốc gia giải mã các thông điệp trong khi người dùng vẫn cho rằng hệ thống này an toàn. Tất nhiên khó có thể phản đối lại lời tuyên bồ này nhưng chưa có bằng chứng nào được đưa ra để chỉ rõ rằng trong thực tế các cửa sập tồn tai trong DES. Năm 1976, cục an ninh quốc gia Hoa kỳ tuyên bố rằng những thuộc tính của hộp S là tiêu chuẩn thiết kế:

- Mỗi một dòng của một hộp S là sự lặp lại của các số nguyên từ 0 đến 15.
- Không có hộp S nào có tuyến tính hay là hàm Affine của các đầu vào.
- Thay đổi một bit đầu vào của hộp S gây ra ít nhất 2 bit đầu rat hay đổi.
- Đối với bất kì một hộp S và bất kì đầu vào x, S(x) và S(x^001100) gây ra sự khác biệt ở ít nhất hai bit 9x ở dây là một chuỗi có độ dài 6.

Hai đặc tính khác của hộp S được thiết kế như là được điều khiển bởi các tiêu chuẩn thiết kế của cục An ninh Quốc gia.

- Đối cới bất kì hộp S nào, bất kì x đầu vào và đối với e, $f \in \{0,1\}$, $S(x) \neq S(x$ + 1 lef00)
- Đối với bất kì hộp S nào, nếu một bit đầu vào là cố định và ta xem giá trị của bit đầu ra cố định, số đầu vào làm cho đầu ra = 0 sẽ gần với số đầu vào làm cho đầu ra = 1(lưu ý rằng nếu ta cố định giá trị của bit 1 và bit 16 thì sẽ có 16 đầu vào làm cho mỗi bit đầu ra cụ thể = 0 và 16 đầu vào làm cho 1 bit đầu ra cụ thể = 1. Đối với lần thứ 2 thông qua các bit đầu vào thứ 5 thì điều này sẽ không đúng nhưng gần đúng. Cặn kẽ hơn với bất kì hộp S nào nếu như giá trị của bất cứ bit đầu vào nào là cố định thì số đầu vào mà nhờ đó bất cứ bit đầu ra cố định nào có giá trị là 0 hoặc 1

thì luôn luôn nằm giữa 13 và 19). Lời chỉ trích thích đáng nhất về DES là kích cỡ của không gian khóa 2^{56} là khóa nhỏ để trở lên an toàn. Những máy có mục đích đặc biệt khác nhau được đẹ trình cho tấn công bản rõ, mà nó thực hiện chủ yếu tìm kiếm khóa. Đó là đưa ra bản rõ x gồm 64bit và một bản tương đương y, thì mỗi một khóa có thể có lớn hơn một khóa k là $E_k(x) = y$ được tìm thấy.(Lưu ý rằng có thể lớn hơn một khóa k_0).

- Đầu năm 1977, Diffie và Hellman đề nghị rằng một người có thể xây dựng một chip VLSL có thể kiểm tra được 10^6 khóa một giây. Một máy với 10^6 có thể tìm kiếm toàn bộ không gian khóa trong một ngày. Họ dự tính rằng máy này được xây dựng với giá khoảng 20 USD.

- Tại hôi nghị CRYPTO'93 (phần kéo dài), Michael Wiener đưa ra một thiết kế rất chi tiết về một máy dò khóa. Máy này dựa trên một chip dò khóa được nối truyền liên tiếp, vì vậy 16 mã hóa được diễn ra đồng thời. Chip này có thể kiểm tra 5.10^7 khóa 1 giây và có thể được xây dựng sử dụng công nghệ hiện thời với giá 10,5 USD một chip. Một khung bao gồm 5760 chip có thể được xây dựng với giá 1 triệu USD nhưng giảm thời gian dò trung bình xuống còn 3,5 giờ.

2.4.2 DES trong thực tế

Ngay cả khi việc mô tả DES khá dài dòng thì DES được thực hiện rất hiệu quả trong cả phần cứng lẫn phần mền. Những tính toán số học duy nhất được thực hiện là phép XOR của các chuỗi bit. Việc mở rộng hàm E các hộp S, sự hoán vị IP và P, và việc tính toán k_1, k_2,…, k_{16} tất cả được thực hiện trong thời gian ngắn bởi bảng tìm kiếm trong phần mền hoặc cách nối dây cứng chúng vào môt mạch. Những thi hành phần cứng hiện thời có thể đạt tốc độ mã hóa cực nhanh, công ty thiết bị số thông báo tại CRYPTO'92 rằng họ vừa mới chế tạo được một chip với 50k Transistors có thể mã hóa với tốc độ 1GB/s sử dụng một đồng hồ tốc độ là 250MHz. Giá của chip này khoảng 300USD. Năm 1991 có 45 phần cứng và chương trình cài sẵn thi hành DES đã được ủy ban tiêu chuẩn quốc gia Mỹ phê chuẩn.

Một ứng dụng rất quan trọng của DES là ứng dụng vào việc giao dịch ngân hàng sử dụng các tiêu chuẩn được hiệp hội các ngân hàng Mỹ phát triển. DES được sử dụng đẻ mã hóa các con số, nhận dạng các nhân(PIN) và trao đổi tài khoản được máy thu ngân tự động thực hiện (ATM). DES cũng được clearing Hourse Interbank System (CHIPS) sử dụng đẻ trao đổi có liên quan đến trên $1,5.10^{12}$ USD/ tuần.

DES cũng được sử dụng rộng rãi trong các tổ chức chính phủ như: Bộ năng lượng, Bộ tư pháo và hệ thống phản chứng liên bang.

2.4.3 Một vài kết luận về mã DES

Có rất nhiều phương pháp mã hóa đẻ đảm bảo an toàn dữ liệu. Để đánh giá tính ưu việt một giải thuật mã hóa, nhười ta thường dựa vào các yếu tố: Tính bảo mật, độ phức tạp, tốc độ thực hiện giải thuật và vấn đề phân khóa trong môi trường nhiều người sử dụng.

Hiện nay phương pháp mã hóa DES được sử dụng rộng rãi nhất. Các chip chuyên dung DES được thiết kế nhằm tăng tốc độ xử lí của DES. Rất nhiều nhà toán học ,tin học đã bỏ nhiều công nghiên cứu trong nhiều năm nhằm tìm cách phá vỡ DES (tức là tìm ra cách giải mã trong khoảng thời gian ngắn hơn thời gian cần để thử lần lượt tất cả các khóa). Ngoại trừ việc tìm ra 4 khóa yếu và 12 khóa tương đối yếu cho tới nay chưa có một thông báo nào về việc tì ra cách phá vỡ phương pháp mã hóa này. Để phá vỡ DES bằng phương pháp " vét cạn" thử tất cả các khóa trong không gian khóa cần có một khoản tiền lớn và đòi hỏi một khoảng thời gian dài.

Nhược điểm của DES: nó là thuật tán mã hóa đối xứng. Khi phương pháp này mới được tìm ra ý tưởng thực hiện 50000 tỷ phép mã hóa cần thiết để vượt mặt DES bằng cách thử lần lượt các khóa có thể là điếu không thể làm được nhưng ngày nay với sự phát triển mạnh của phần cứng liệu độ dài 56 bit đã đủ chưa? Và các phép thay thế đã đủ phức tạp chưa ? để đạt được độ an toàn thông tin như mong muốn, đó là vấn đề người ta vẫn đang bàn luận.

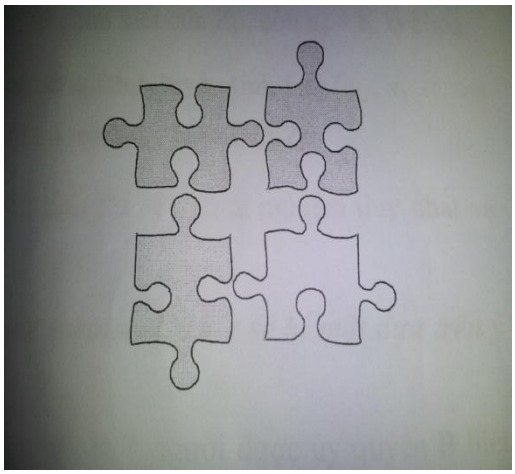
Tuy vậy, DES đã được phân tích kĩ lưỡng và công nhận là vững chắc. Các hạn chế của nó đã được hiểu rõ và có thể xem xét trong quá trình thiết kế và để tăng độ an toàn hơn, ngày nay các hệ thống ma hóa sử dụng DES mở rộng (3DES), được ứng dụng rộng dãi. Với DES mở rộng khóa có thể là 128 bit,…độ lớn khối có thể là 128 bit. Do vậy độ an toàn mở rộng của DES cao hơn rất nhiều.

CHƯƠNG 3. CÁC SƠ ĐỒ CHIA SẺ BÍ MẬT

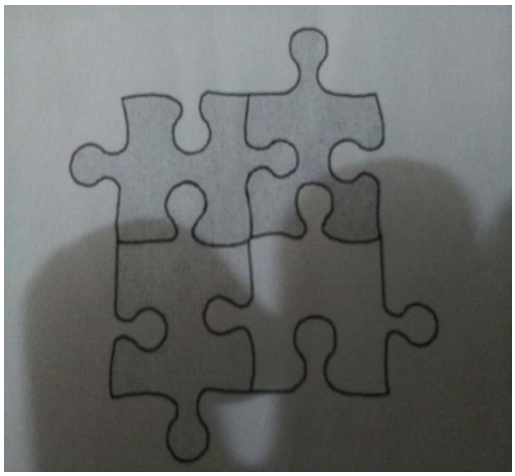
3.1 Khái niệm về chia sẻ bí mật

Thông tin cần giữ bí mật được chia thành nhiều mảnh và giao cho nhiều người, mỗi người giữ một mảnh. Thông tin này có thể được xem lại, khi mọi người giữ các mảnh nhất trí. Các mảnh khớp lại để được tin gốc.

- Thông tin cần giữ bí mật được chia thành nhiều mảnh và trao cho mỗi thành viên tham gia nắm giữ.



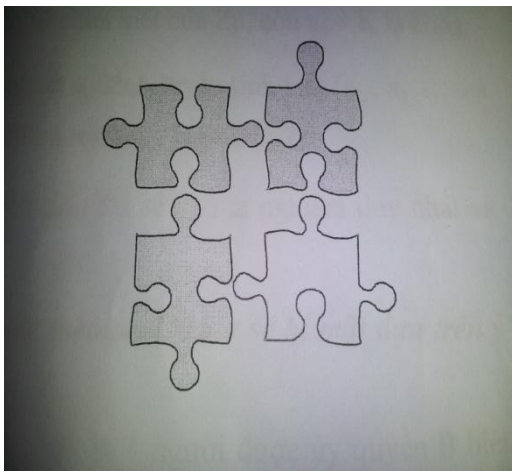
Thông tin bí mật



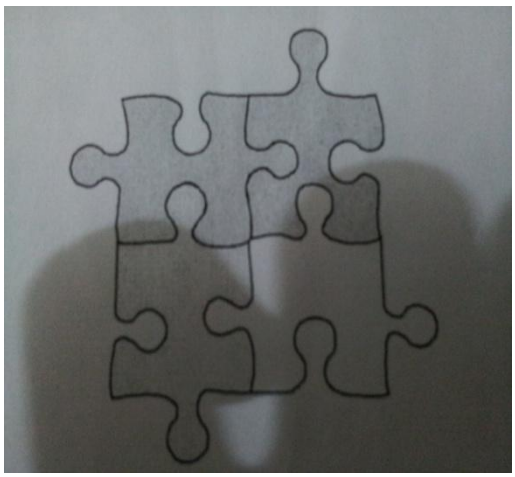
Các mảnh được chia

-

- Khi các mảnh được khớp lại sẽ cho ta thông tin gốc.



Các mảnh được chia



Thông tin bí mật

3.2 Sơ đồ chia sẻ bí mật

Bài toán: Trong một ngân hàng có một két phải mở hằng ngày. Ngân hàng sử dụng 3 thủ quỹ lâu năm nhưng họ không tin bất kì người nào. Bởi vậy họ cần thiết kế một hệ thống sao cho bất kì 2 thủ quỹ nào cũng không thể mở được két song riêng từng người một thì không thể mở được. Vấn đề này có thể được giải quyết được bằng lược đồ chia sẻ bí mật.

3.2.1 Khái niệm " sơ đồ chia sẻ bí mật":

Sơ đồ chia sẻ bí mật là một phương thức để chia sẻ bí mật ra nhiều phần sau đó phân phối cho một tập hợp những người tham gia sao cho các tập con trong số những người nay được chỉ thị, có khả năng khôi phục lại bí mật bằng cách kết hợp dữ liệu của họ.

Một sơ đồ chia sẻ bí mật là hoàn hảo, nếu bất kì một tập hợp những người tham gia mà không được chỉ định, sẽ không thu được thông tin về bí mật.

3.2.2 Định nghĩa:

Cho t, w là các số nguyên dương $t \leq w$. Một sơ đồ ngưỡng A(t,w) là một phương pháp phân chia khóa k cho một tập w thành viên (kí hiệu là P) sao cho t thành viên bất kì có thể tính được k nhưng không một nhóm (t-1) thành viên nào có thể làm được điều đó.

Giá trị k được chọn bởi một thành viên đặc biệt được gọi là ngươi phân phối (D). $D \notin P$.

D phân chia khóa k cho mỗi thành viên trong P bằng cách cho mỗi thành viên một thông tin cục bộ gọi là mảnh. Các mảnh được phân phát một cách bí mật để không thành viên nào biết được mảnh được trao cho mỗi thành viên khác. Một tập con các thành viên $B \subseteq P$ sẽ kết hợp các mảnh của họ để tính khóa k (cũng có thể trao các mảnh của mình cho một người đáng tin cậy để tính khóa hộ).

Nếu $|B| \geq t$ thì họ có khả năng tính được k.

Nếu $|B| \leq t$ thì không thể tính được k.

Gọi P là tập các giá trị được phân phối khóa K: $P = \{ p_i: 1 \leq i \leq w\}$

K là tập khóa: tập tất cả các khóa có thể.

S tập mảnh: tập tất cả các mảnh có thể.

Sau đây ta trình bày một sơ đồ ngưỡng được gọi là sơ đồ ngưỡng Shamir.

Giai đoạn khởi tạo:

1. D chọn w phần tử khác nhau và khác 0 trong Zp và kí hiệu chúng là: x_i , $1 \leq i \leq w$ ($w \geq p+1$).
   Với $1 \leq i \leq w$, D cho giá trị x_i cho p_i .Các giá trị x_i là công khai.

Phân phối mảnh:

2. Giả sử D muốn phân chia khóa $k \in Zp$. D sẽ chọn một cách bí mật (nhẫu nhiên và độc lập) t-1 phần tử Zp, a_i……...a_{i-1}
3. Với $1 \leq i \leq w$, D tính $y_i = a(x_i)$, trong đó
   $a(x) = k + \sum_{i=1}^{t-1} a^i x^i \bmod p$
4. Với $1 \leq i \leq w$, D sẽ trao mảnh y_i cho p_i

Hình 3.1 : Sơ đồ ngưỡng Shamir

Trong sơ đồ 3.1 D xây dựng một đa thức ngẫu nhiên a(x) có bậc tối đa là t-1. Trong đa thức này hằng số là khóa k. Mỗi thành viên p_i sẽ có một điểm (x_i, y_i). Ta xét một tập con B gồm t thành viên tạo lại khóa k bằng 2 phương pháp:

- Phép nội suy đa thức
- Công thức nội suy Lagrange

Tạo lại khóa k bằng phương pháp sử dụng phép nội suy đa thức:
Giả sử các thành viên p_i , muốn xác định khóa k.
Ta biết rằng:
$y_{ti} = a(x_i)$. trong đó $a(x_i)$ là một đa thức bí mật được D chọn. Vì a(x) có bậc lớn nhất là t-1 nên ta có thể viết như sau:

$a(x) = a_0 + a_1x + \ldots\ldots + a_{i-1}x^{i-1}$

Ta có hệ các phương trình tuyến tính (trong Zp) như sau:

$a_0 + a_1x_{i1} + a_2x \ldots\ldots\ldots + a_{t-1}x_{it1}^{t-1} = y_{i1}$

$a_0 + a_1x_{i2} + a_2x_{i2} \ldots\ldots\ldots + a_{t-1}x_{it}^{t-1} = y_{i2}$

……..

$a_0 + a_1x_{i1} + a_2x_{it} \ldots\ldots\ldots + a_{t-1}x_{itt}^{t-1} = y_{it}$

Trong đó hệ số a_0, a_1,…. a_{t-1} là các phần tử chưa biết của Z_p , còn $a_0 = K$ là khóa.
Vì $y_i = a(x_i)$ nên B có thể thu được t phương trình tuyến tính t ẩn (a_0, a1,….a_{t-1}), ở đây tất cả các phép tính số học đều được thực hiện trên Z_p.

Nếu các phương trình này độc lập tuyến tính thì sẽ cho ta nghệm duy nhất và thu được giá trị khóa a_0.

Sau đây chúng tôi trình bày một thủ tục (protocol) chia sẻ bí mật dựa trên ý tưởng của Languange:

Giá sử ta có n thực thể A_1 ,A_2,…..A_n và có một người ủy quyền B biết được toàn bộ khóa bí mật SϵN.

Người được ủy quyền B thực hiện các bước sau đây:

1. B chọn 1 số nguyên tố P đủ lớn sao cho: n << √p. Với SϵZ_p.
2. B, tiếp theo, chọn 2n-1 số một cách ngẫu nhiên:

   $a_1, a_2,\ldots, a_{n-1}$

   $v_0, v_1,\ldots, v_{n-1}$

   trong đó: $v_i \neq 0$, $v_i \neq v_j$, $i \neq j$
3. B xác định một đa thức với các hệ số $a_1, a_2,\ldots, a_{n-1}$ trên Z_p:

   $f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots\ldots\ldots + a_1x + s \pmod P$
4. Bây giờ B gửi cho A_j một cách công khai cặp $(v_j, f(v_j))$ với j = 0,1,2,….n-1 coi như là một mảnh riêng của A_j

Khôi phục bí mật S:

Tất cả n người A_1 ,A_2,…..A_n có thể hợp tác lại để khôi phục lại bí mật S bằng cách tính:

$$g(x) = \sum_{0\leq j\leq n} f(vj) \prod_{0\leq i\leq n} (vj - vi)^{-1}(x - vi) \text{ mob p.} \qquad \textbf{(3.1)}$$

Khi đó dễ dàng xác định được S = g(0)

Ta có định lí sau:

n thực thể kết hợp với nhau thì có thể khôi phục bí mật S một cách có hiệu quả đó là: S= g(0) = f(0)

Chứng minh:

Thật vậy, dễ thấy rằng g(x) là hàm nội suy Lagrange của hàm f(x) là một đa thức có cấp bé hơn n và g thỏa mãn điều kiện: $g(v_j) = f(v_j)$ với $0 \leq j \leq n$

Do đó, f-g là đa thức trên Z_p có cấp bé hơn n, nhưng nó lại có ít nhất n nghiệm khác nhau: là các số r thỏa mãn f(r)- g(r) = 0. Chứng tỏ rằng: f(a)= g(a) $\forall a \in Z_p$, đặc biệt f(0) = g(0) = S đó là điều cần chứng minh.

Sau đây tôi xin lấy một số ví dụ cụ thể:

Ví dụ 1: Co 3 người A_1 ,A_2,A_3 muốn chia sẻ bí mật S = 472

Cho p=1999 công khai.

A chọn $v_0 = 626$, $v_1 = 674$, $v_0 = 93$

$a_1 = 334$, $a_2 = 223$

tính $f(v_j) = a_2 v_j^2 + a_1 v_j + S \bmod p$

Áp dụng công thức trên với S = 472 ta có:

$f(v_0) = 1724$

$f(v_1) = 1925$

$f(v_2) = 1241$

A_1 có cặp $(v_0, f(v_0)) = (626, 1724)$

A_1 có cặp $(v_1, f(v_1)) = (674, 1925)$

A_3 có cặp $(v_2, f(v_2)) = (93, 1241)$

Ba người hợp lại sẽ xác định được S:

$S= g(0) = \sum_{0\le j\le 3} f(vj)bj \bmod p.$

Với $b_j = \prod_{0\le i\le n, i\ne j} vi(vi - vj)^{-1} \bmod p$ (3.2)

Áp dụng công thức (3.2) trên ta tính được:

$b_0= 1847$

$b_1 = 1847$

$b_2 = 1847$

$S= g(0) = \sum_{0\le j\le 3} f(vj)bj \bmod p = 472.$

Ví dụ 2:

Cho số p=342853815608923 (Đây là một số nguyên tố được lấy trong bảng các số nguyên tố từ cuốn " The Art of Programing" của Knut vol2).

n=3, ta có:

a_1= 53958111706386

a_2= 151595058245452

v_0=111350135012507

v_1=207244959855905

v_2=20545949133543

Giá sử bí mật là S= 151595058245452

Tính $f(v_0)$=109351520587519

$f(v_1)$=174675701531216

$f(v_2)$=117471713218253

Đặt $b_j = \prod_{0\le i\le n, i\ne j} vi(vi - vj)^{-1} \bmod p$

Ta có: $S= g(0) = \sum_{0\le j\le 3} f(vj)bj \bmod p$

b_0 = 266921901220910

b_1 = 129147516050688

b_2 = 289638215946249

Ta tính được S' = 151595058245452. S' trùng với khóa bí mật đã cho.

3.3 Cấu trúc truy nhập và sơ đồ chia sẻ bí mật

Trong phần trước ta mong muốn rằng t thành viên bất kì trong w thành viên có khả năng xác định được khóa. Tình huống tổng quát hơn là phải chỉ rõ một các chính xác các thành viên có khả năng xác định khóa và những tập con không có khả năng này.

Ký hiệu:

- P là tập gồm m thành viên được chia mảnh công khai x_i.
- Γ là một tập các tập con của P, các tập con trong Γ là các tập con các thành viên có khả năng tính khóa.

- Γ được gọi là một cấu trúc truy nhập
- Các tập con trong Γ được gọi là các tập con hợp thức.

Ví dụ:

Chìa khóa để mở két bạc là chìa khóa số được chia thành 3 mảnh khóa, có 3 thủ quỹ là P_1 , P_2 ,P_3. Mỗi thủ quỹ giữ một mảnh khóa. Chỉ có thủ quỹ P_1 và P_2 hoặc P_2 và P_3 khi khớp 2 mảnh khóa của họ với nhau thì sẽ nhận được chìa khóa gốc để mở két bạc.

Các tập con hợp thúc là các tập con có thể mở khóa: { P_1 ,P_2 },{ P_2 ,P_3}.

Vậy Γ là: { P_1 ,P_2 },{ P_2 ,P_3}.

3.3.1 Định nghĩa sơ đồ chia sẻ bí mật hoàn thiện

Một sơ đồ chia sẻ bí mật hoàn thiện thể hiện cấu trúc truy nhập Γ là phương pháp chia sẻ khóa K cho một tập w thành viên (được kí hiệu là P) thỏa mãn 2 tính chất sau:

1. Nếu một tập con hợp thức các thành viên B $\subset$ P góp chung các mảnh của họ thì họ có thể xác định được giá trị của K.
2. Nếu một tập con không hợp thức các thành viên B $\subset$ P góp chung các mảnh của họ thì họ không thể xác định được khóa k.

Ví dụ:

Trong sơ đồ Samir A(t, m) thể hiện cấu túc truy nhập sau:

$\Gamma = \{B \subset P : /B/\}$

Vậy sơ đồ Samir là sơ đồ chia sẻ bí mật hoàn thiện.

Chú ý: "Tập trên" của một "tập hợp thức" sẽ là "tập hợp thức"

Giả sử $B \in \Gamma$ và $B \subset C \subset P$, giả sử tập con C muốn K

Vì B là một tập con hợp thức nên nó có thể xác định được K.

Tập con C có thể xác định được khóa K bằng cách bỏ qua các mảnh (tin) của các thành viên trong B, C.

Tức là: Nếu $B \in \Gamma$ và $B \subseteq C \subseteq P$ thì $C \in \Gamma$.

3.3.2 Định nghĩa tập hợp thức tối thiểu

Nếu Γ là một cấu trúc truy nhập thì $B \in \Gamma$ được gọi là "tập hợp thức" tối thiểu nếu: $A \subseteq B$, $A \neq B$ thì $A \neq \Gamma$. Nói cách khác B là tập hợp thức nhỏ nhất trong Γ.

Tập các tập con hợp thức tối thiểu của Γ kí hiệu là Γ_0 và được gọi là cơ sở của Γ. Vì Γ chứa tất cả các tập con của P là tập trên của một tập con trong cơ sở Γ_0 nên Γ được xác định một cách duy nhất như một hàm của Γ_0.

Biểu diễn về mặt toán học ta có:

$\Gamma = \{C \subseteq P; B \subseteq C, B \in \Gamma_0\}$

3.4 Mạch đơn điệu

Một phương pháp đẹp và đơn giản về mặt khái niệm do Benaloh và Leichter đưa ra. Ý tưởng của nó là xây dựng một mạch đơn điệu "ghi nhận" cấu trúc truy nhập và sau đó xây dựng một sơ đồ chia sẻ bí mật trên cơ sở xây dựng mô tả về mạch. Ta gọi đó là cấu trúc mạch đơn điệu.

3.4.1 Định nghĩa mạch đơn điệu

Một mạch Bolean C với w đầu vào x_1 ,….. x_w (tương ứng với w thành viên P_1 ,……P_w) và một đầu ra y.

Mạch này gồm các cổng "hoặc" và các cổng "và" không có bất kì một cổng "phủ định" nào một mạch như vậy gọi là mạch đơn điệu.

Mạch được ghép có số đàu vào tùy ý nhưng chỉ có một đầu ra (tức là một cổng có thể có nhiều dây vào nhưng chỉ có một dây ra).

Xây dựng mạch đơn điệu:

Nếu Γ là một mạch đơn điệu các tập con của P thì dễ dàng xây dựng được một mạch đơn điệu C sao cho $\Gamma(C) = \Gamma$.

Giả sử Γ_0 là cơ sở của Γ khi đó ta xây dựng công thức Bolean dạng tuyển hội sau:

$$C = \bigvee B \in \Gamma_0 \, (\wedge P_i \in BP_i)$$

Ví dụ:

Nếu trong tập các thành viên { P_1 , P_2 , P_3 } có tập cơ sở:

Γ_0 ={{ P_1 ,P_2 },{ P_2 ,P_3}}.

Ta thu được công thức Bolean sau:

C = (P_1^P_2) ^ ($P_2 \vee P_3$)

3.4.2 Chia sẻ khóa bí mật dựa vào "mạch đơn điệu"

Thuật toán thực hiện phép gán một giá trị f(W) $\in$ K cho mỗi dây W trong mạch C.

- Đầu tiên, dây ra W_{out} của mạch sẽ được gán giá trị khóa K.
- Thuật toán sẽ được lặp lại một số lần cho đến khi mỗi dây có một giá trị gán vào nó.
- Cuối cùng, mỗi thành viên P_i sẽ được một danh sách các giá trị f(W) sao cho W là một dây vào của mạch có đầu vào x_i.

Thuật toán chia sẻ khóa K:

1. $F(W_{out}) = K$
2. While tồn tại một dây W sao cho f(W) không xác định DO

Begin

3. Tìm cổng G của C sao cho f (W_g) được xác định, W_g là dây ra của G nhưng f(W) không được xác định với bất kì dây nào của G.
4. If G là cổng "hoặc" Then $f(W) = f(W_g)$ với mỗi dây vào W của G

Else (G là cổng "và")

Cho các dây vào của G là W_1 ……..W_t

Chọn độc lập, ngẫu nhiên t-1 phần tử của Z_m và kí hiệu chúng là:

Y_{g1} ,…….Y_{gt-1}

End;

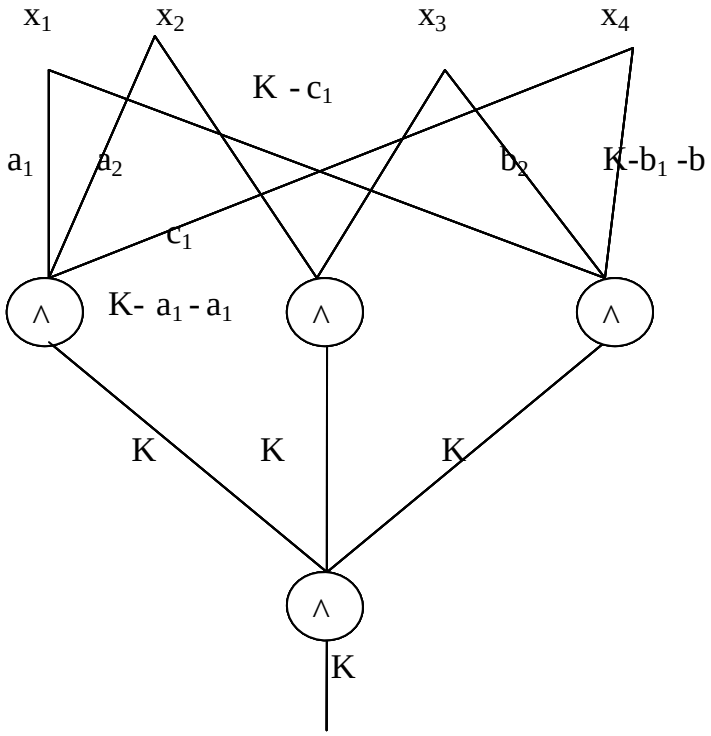
5. For $1 \leq i \leq m$ Do $f(W_i) = Y_g$

Ví dụ:

Giả sử K là khóa. Giá trị K sẻ được đưa tới mỗi một trong 3 đầu vào của cổng "hoặc" cuối cùng. Tiếp theo ta xét cổng "và" ứng với mệnh đề $P_1 \wedge P_2 \wedge P_4$. Ba dây vào sẽ được gán các gía trị tương ứng là: a_1, a_2, $k - a_1 - a_2$ ở đây tất cả các phép tính đều được thực hiện trên Z_m. Tương tự 3 dây vào tương ứng với $P_1 \wedge P_3 \wedge P_4$ sẽ được gán các giá trị b_1, b_2, $k - b_1 - b_2$. Cuối cùng 2 dây vào tương ứng với $P_2 \wedge P_3$ sẽ được gán các giá trị c_1, $k - c_1$. Chú ý rằng a_1, a_2, b_1, b_2 và c_1 đều là các biến

ngẫu nhiên độc lập trong Z_m. Nếu nhìn vào các mảnh mà 4 thành viên nhận được thì ta có:

1. P_1 nhân a_1, b_1
2. P_2 nhân a_2, c_1
3. P_3 nhân b_2, K-c_1
4. P_4 nhân K- a_1 - a_2, K- b_1 – b_2
5. Như vậy mỗi thành viên sẽ nhận 2 phân tử trong Z_m làm mảnh của mình.



Hình 3.2 Một mạch đơn điệu

Ta sẽ chứng tỏ rằng sơ đồ này là hoàn thiện.

Trước tiên ta kiểm tra thấy rằng mỗi tập con cơ sở có thể tính được k. Tập con hợp thức $\{P_1, P_2, P_4\}$ có thể tính :

$k = a_1 + a_2 + (k – a_1 – a_2)$ mob m

Tập con $\{P_1, P_3, P_4\}$ có thể tính:

$k= b_1 + b_2 + (k – b_1 – b_2)$ mob m

Cuối cùng tập con $\{P_2, P_3\}$ có thể tính:

$$K = c_1 + (k – c_1) \text{ mob } m$$

Như vậy, mọi tập con hợp thức đều có thể tính được k, do đó ta sẽ hướng sự chú ý tới các tập con không hợp thức. Chẳng hạn, nếu B_2 và B_2 là 2 tập con không hợp thức ($B_1 \subseteq B_2$) và B_2 không thể tính đươc k thì B_1 cũng không thể tính được k. Ta định nghĩa tập con $B \subseteq P$ là một tập con không hợp thức tối đa nếu $B_1 \in P$ đối với mọi $B_1 \subseteq B$, $B_1 \neq B$. Điều này dẫn đến kết luận là chỉ cần kiểm tra thấy không một tập con không hợp thức nào có thể xác định được một chút thông tin nào về khóa k là được. Ở đây các tập con không hợp thức tối đa là:

$\{P_1, P_2\}$, $\{P_1, P_3\}$, $\{P_1, P_4\}$, $\{P_2 , P_4\}$, $\{P_3, P_4\}$.

Trong mỗi trường hợp, dễ dàng thấy được k không thể tính được, hoặc do thiếu một mảnh thông tin ngâu nhiên cần thiết nào đó hoặc do tất cả các mảnh có từ một tập con ngẫu nhiên. Ví dụ tập con $\{P_1, P_2 \}$ chỉ có các giá trị ngẫu nhiên a_1, b_1, a_2, c_1. Một ví dụ khác, tập con $\{P_3, P_4 \}$ có mảnh b_2 , $k – c_1$, $k - a_1 - a_2$, $k - b_1 - b_2$. Vì các giá trị c_1 ,a_2,a_1 và b_1 là các giá trị ngẫu nhiên chưa biết nên k không thể tính được, trong mỗi trường hợp có thể mỗi tập con không hợp thức đều không có chút thông tin gì về giá trị của k.

CHƯƠNG 4. ỨNG DỤNG THUẬT TOÁN DES VÀ LƯỢC ĐỒ CHIA SẺ BÍ MẬT VÀO THI TUYỂN SINH

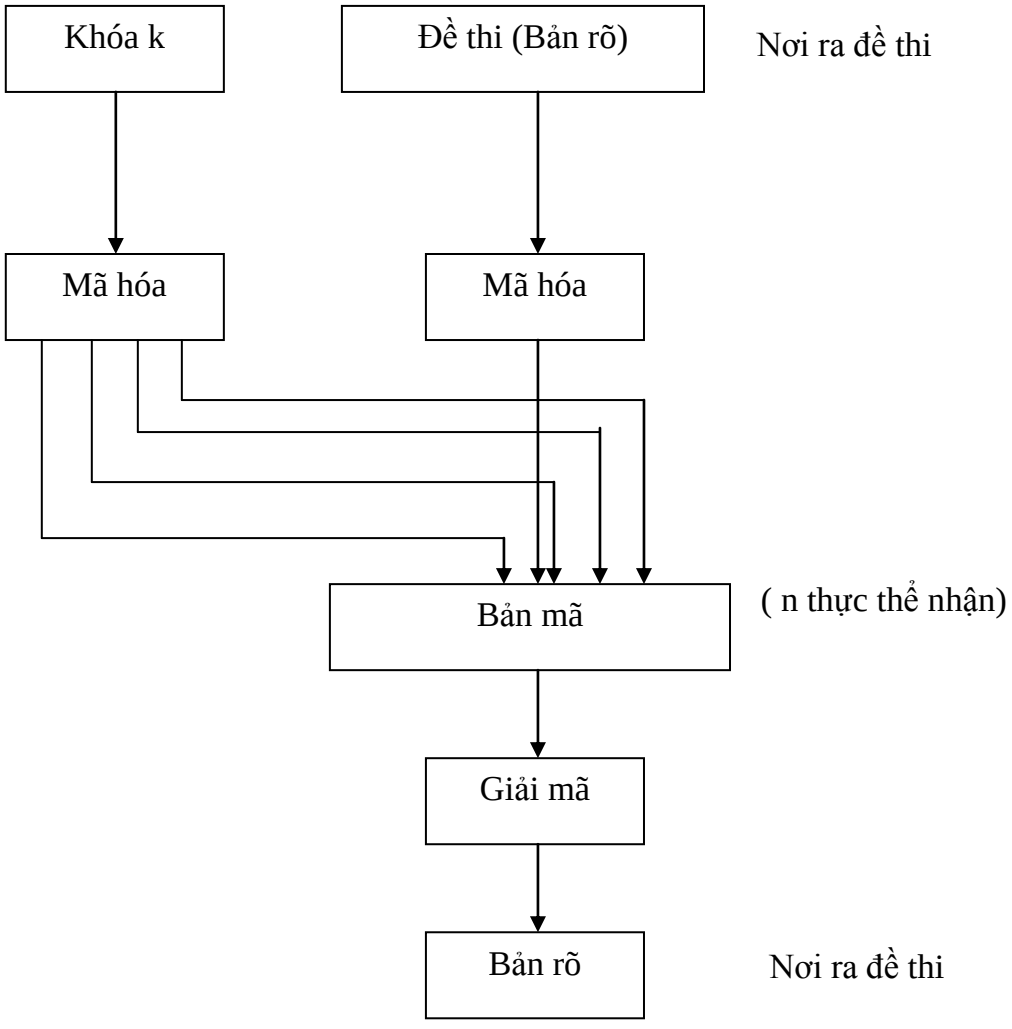
4.1 Các ứng dụng

Ta có thể áp dụng thuật toán DES và sơ đồ chia sẻ bí mật vào rất nhiều ứng dụng chẳng hạn trong đấu thầu từ xa, trong mã thẻ ATM, trong thi tuyển sinh…

Ở đây ta nghiên cứu một ứng dụng là trong thi tuyển sinh, vậy có một bài toán được đưa ra là: Trong một kì thi, nơi ra đề thi và nơi tổ chức thi ở cách xa nhau, ta phải thực hiện việc chuyển đề thi từ nơi ra đề tới nơi tổ chức thi trên mạng máy tính sao cho đảm bảo về tính bảo mật.

4.2 Quy trình thực hiện giải bài toán

4.2.1 Sơ đồ



Khóa DES gồm 56 bit, tương đương với một số nguyên gồm 20 chữ số thập phân. Con số bí mật nay không quá lớn đối với bài toán chia sẻ bí mật. Cho nên việc tính toán là rất hiệu quả.

4.2.2 Các bước thực hiện

Theo sơ đồ trên ta phải thực hiện theo các bước sau:

- Nơi ra đề thi:
  - Bản rõ (đề thi)
  - Mã hóa bản rõ
  - Tạo khóa k
  - Mã hóa khóa k
  - Gửi bản mã
- Nơi tổ chức thi:
  - Nhận bản mã và cặp (v_j, $f(v_j)$)
  - Giải bản mã (sau khi nhận đủ các cặp khác từ người ra đề thi để xác định được khóa K).

Mã hóa bản rõ (đề thi): Bộ giáo dục dùng bảng mã ASCII mở rộng để chuyển bản rõ từ dạng kí tự sang Hexa sau đó dùng thuật toán DES để mã hóa.

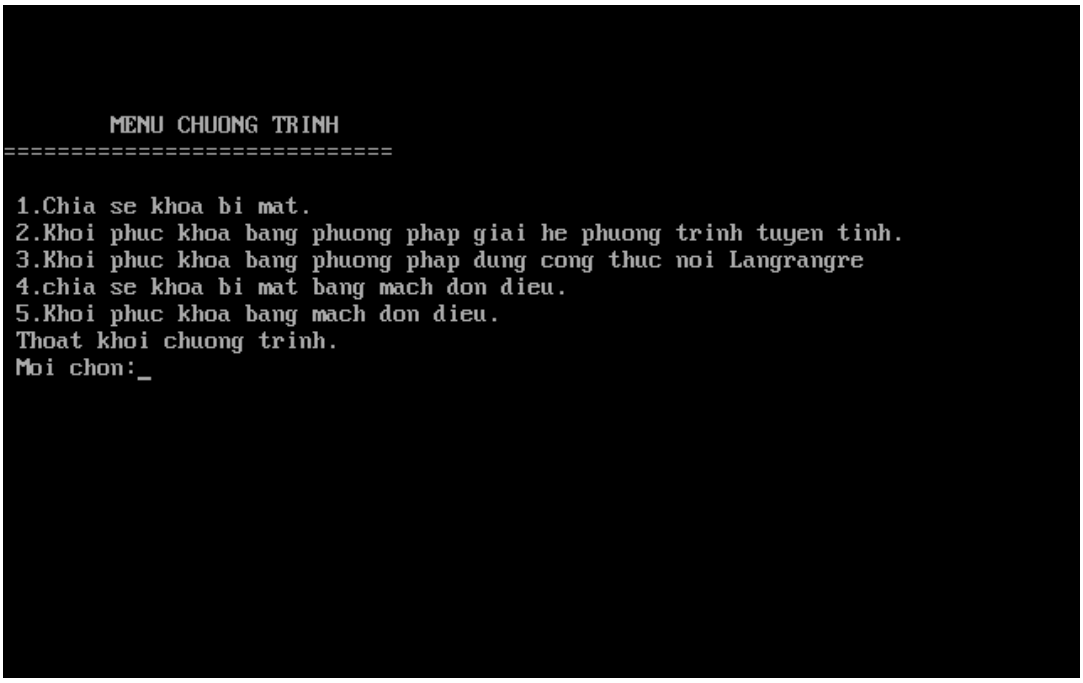
Tạo khóa k: Dùng dãy kí tự dạng chữ hoặc dạng số, nhóm 8 kí tự thành 1 nhóm sau đó dùng 56 bit để mã hóa.

Gửi bản tin: Dựa vào lược đồ chia sẻ bí mật chia khóa k thành 2 mảnh rời nhau k_1, k_2 : $k_1 + k_2 = k$. Sau đó gửi k_1 cho n thực thể (các địa chỉ thi). Quy định đến đúng giờ G vụ Đào tạo gửi nốt k_2 cho n thực thể đó trên cơ sở k_1, k_2. Tất cả các nơi đều mở được đề và trao cho học sinh hoặc gửi cho học sinh thông qua máy tính để làm (qua mail đồng thời).

Sau đây là chương trình mô phỏng "chia sẻ bí mật bằng ngôn ngữ C".

4.2.3 Mô phỏng lược đồ chia sẻ bí mật bằng ngôn ngữ C:

Chương trình gồm có 5 phần:

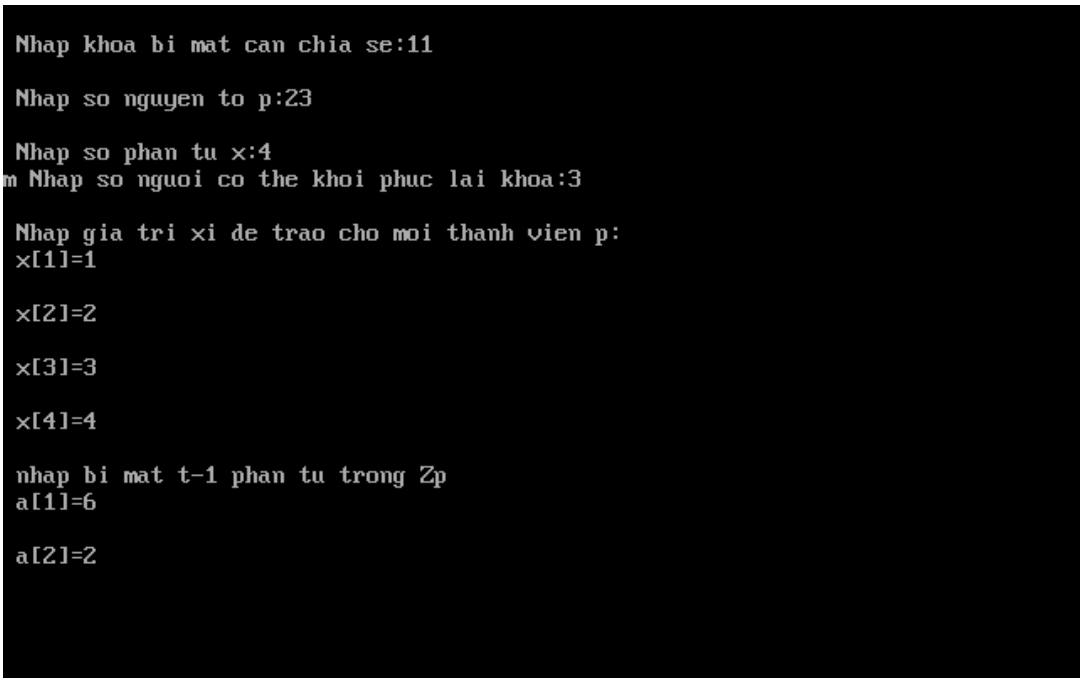


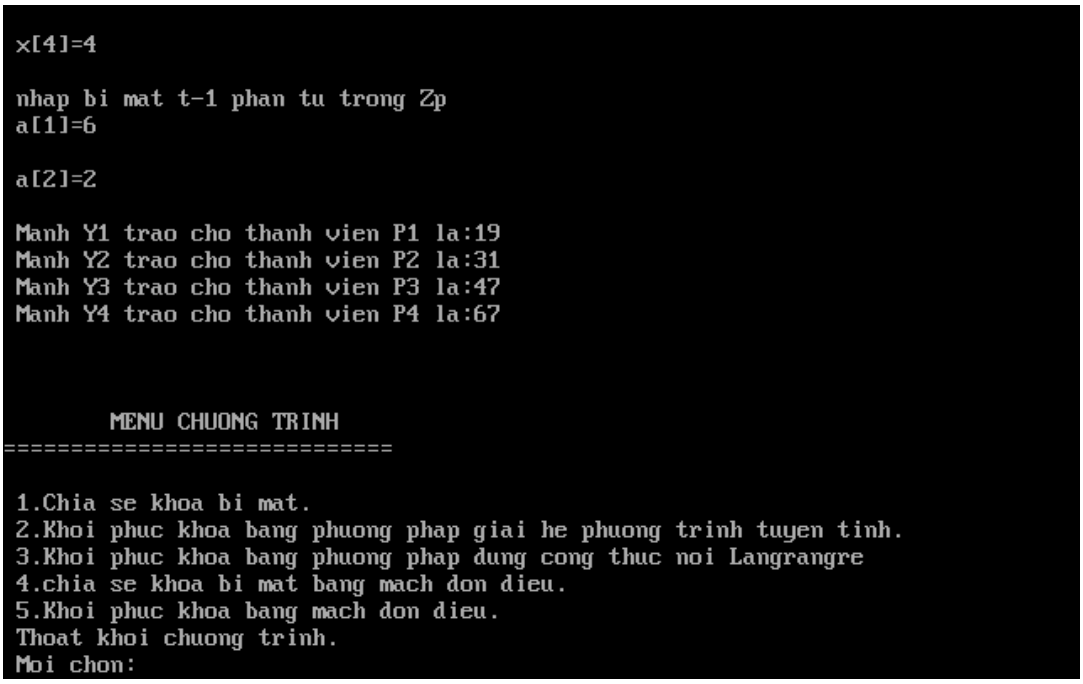
Hình 4.1 Giao diện chương trình

4.2.3.1 Chia sẻ khóa bí mật theo giao thức "chia sẻ bí mật" Shamir.

```
void giaothuc::chiakhoa()
{
int h;
cout<<"\n Nhap khoa bi mat can chia se:"; cin>>k;
cout<<"\n Nhap so nguyen to p:"; cin>>p;
cout<<"\n Nhap so phan tu x:"; cin>>m;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>t;
cout<<"\n Nhap gia tri xi de trao cho moi thanh vien P:";
for(i=1; i<=m; i++)
{
cout<<"\n x["<<i<<"]="; cin>>x[i];
}
cout<<"\n Nhap bi mat t-1 phan tu trong Zp";
for(j=1;j<t;j++)
```

```
{
cout<<"\n a["<<j<<"]="; cin>>a[j];
}
for(i=1;i<m;i++)
{ l=0;
for(j=1;j<t;j++)
h=pow(x[i],j);
l=l+a[j]*h;
}
y=k+l;
cout<<"\n Manh y"<<i<<" trao cho thanh vien P"<<i<<" la:"<<y;
}
}
```





Hình 4.2 và 4.3 Chia sẻ khóa bí mật theo giao thức Shamir

4.2.3.2 Khôi phục khóa bí mật bằng phương pháp giải hệ phương trình tuyến tính

```
void giaothuc::giaihekhoiphuckhoa()
{
int n,m,v,b;
float mx=0, g,e,c,gt,h;
```

```
float G[max] [max] [max], H[max] [max], A[max] [max], B[max] [max], M[max] [max];
cout<<"\n Nhap so nguyen to p:": cin>> p;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>n;
cout<<"\n Nhap gia tri xi da trao cho moi thanh vien P:";
for(j=0;j<n;j++)
{
cout<<"\n x["<<j<<"]="; cin>>x[j];
}
cout<<"\n Nhap cac mang khoa da trao cho moi thanh vien P:";
for(l=0;l<n;l++)
{
cout<<"\n Y"<<l<<"="; cin>> B[l][0];
}
for(i=0;i<n;i++)
{
for(j=0;j<n; j++)
{
A[i][j]=pow(x[i],j);
}
}
cout<<" He Phuong trinh tuyen tinh la:\n";
for(i=0; i<n; i++)
{
for(j=0; j<n; j++)
{
If(j<n-1)
{
```

```
cout<<”A[i][j]<<”a”<<j<<”+”;
}
If(j==n-1)
{
cout<<A[i][j]<<”a”<<j;
}
}
cout<<”=”<<B[i][j];
cout<<”\n;
}
cout<<”\n Vay nghiem cua he Phuong trinh la:”;
for(e=0; e<n; e++)
{
for(i=0; i<n; i++)
{
for(j=0; j<n; j++)
{
G[i][j]=A[i][j];
}
}
for(j=0; j<n; j++)
{
If(j==e-1)
{
for(i=0; i<n; i++)
{
G[i][j]= B[i][0];
```

```
}
}
}
i=0; t=0;
while(t<n-l)
{
If(G[i][j]==0)
{
for(j=1; j<n; j++)
If(G[i][j]>mx)
{
mx=G[i][j];
v=j;
}
for(j=0; j<n; j++)
{
G=G[i][j];
G[i][j]=-G[v][j];
G[v][j]=g;
}
}
If(G[i][i]!=0)
{
for(k=i+1; k<n; k++)
{ c=G[k][i]/ G[i][i];
h=c+(G[k][i]-c*G[i][i])/ G[i][i];
for(j=0; j<n; j++)
```

```
{
G[k][j]=G[k][j]*h;
}
}
i++;
}
t=t+1;
}
gt=1;
for(i=0; i<n; i++)
{
gt=gt+G[i][i];
}
If(e==0)
{
l=gt;
}
Else
{
b=gt/l;
If(b<p&& b>0)
{
cout<<"\n a"<<e<<"="<<b;
If(e==1)
{
cout<<"\n Khoa dc khoi phuc lai la:K= al="<<b;
}
```

```
}
}
}
}
```

```
Nhap so nguyen to p23

Nhap so nguoi co the khoi phuc lai khoa:3

Nhap gia tri xi da trao cho moi thanh vien P:
x[0]=1

x[1]=2

x[2]=3

nhap cac manh khoa da trao cho moi thanh vien P:
Y0=19

Y1=31

Y2=47_
```

```
Y1=31

Y2=47

He phuong trinh tuyen tinh la:
1a0+1a1+1a2=19
1a0+2a1+4a2=31
1a0+3a1+9a2=47

Vay he cua nghiem phuong trinh la:
a1=19
Khoa duoc khoi phuc lai la K=a1=19



       MENU CHUONG TRINH
============================

1.Chia se khoa bi mat.
2.Khoi phuc khoa bang phuong phap giai he phuong trinh tuyen tinh.
3.Khoi phuc khoa bang phuong phap dung cong thuc noi Langrangre
4.chia se khoa bi mat bang mach don dieu.
5.Khoi phuc khoa bang mach don dieu.
Thoat khoi chuong trinh.
Moi chon:
```

Hình 4.4 và 4.5 Khôi phục khóa bí mật bằng phương pháp giải hệ phương trình tuyến tính

4.2.3.3 Khôi phục khóa bí mật bằng phương pháp dùng công thức nội suy Lagrange

```
void giaothuc::congthuckhoiphuckhoa()
{
float n,m,b;
cout<<"\n Nhap so nguyen to p:"; cin>>p;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>t;
cout<<"\n Nhap gia tri xi da trao cho moi thanh vien P:";
for(j=1; j<=t; j++)
{
cout<<"\n x["<<j<<"]="; cin>>x[j];
}
cout<<"\n Nhap cac manh khoa da trao cho oi thanh vien P";
for(j=1; j<=t; j++)
{
cout<<"\n g["<<j<<"]="; cin>>g[j];
}
k=0;
for(l=1; l<=t; l++)
{
m=1;
for(j=1; j<=t; j++)
{ if(j!=1)
{ b=x[j] – x[l];
n=x[j]/b;
m=m*n;
}
}
```

```
k=k+g[l]*m;
}
cout<<"\n Vay khoa bi mat khoi phuc lai la:"<<k;
}
```

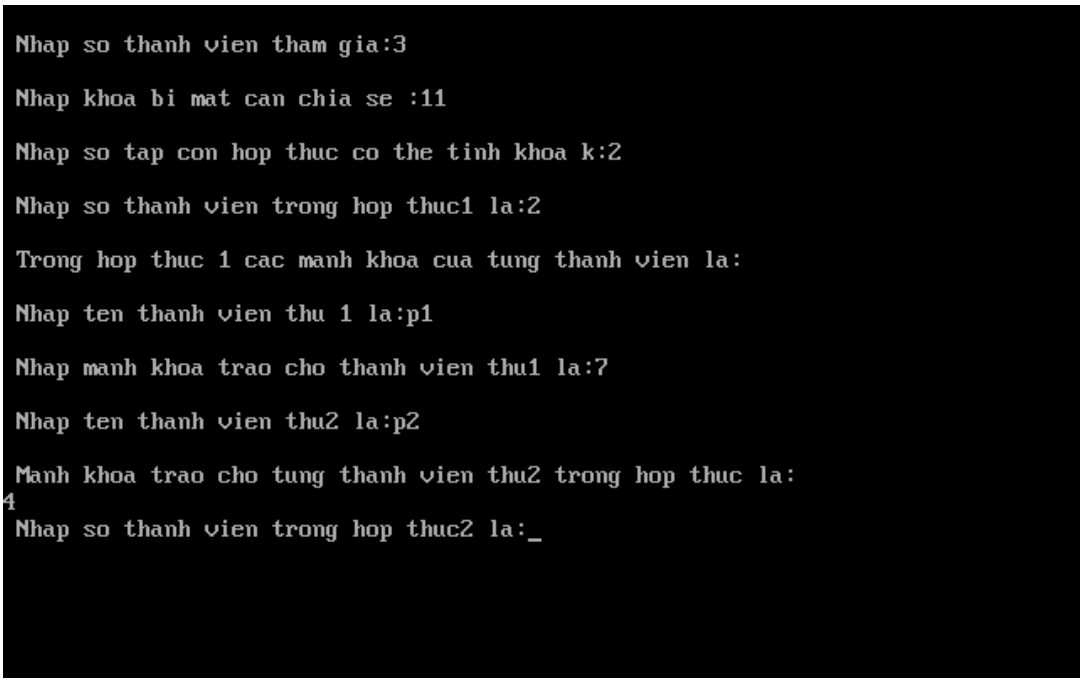
```
Nhap so nguyen to p23

Nhap so nguoi co the khoi phuc lai khoa:3

Nhap gia tri xi da trao cho moi thanh vien P:
x[0]=1

x[1]=2

x[2]=3

nhap cac manh khoa da trao cho moi thanh vien P:
Y0=19

Y1=31

Y2=47_
```

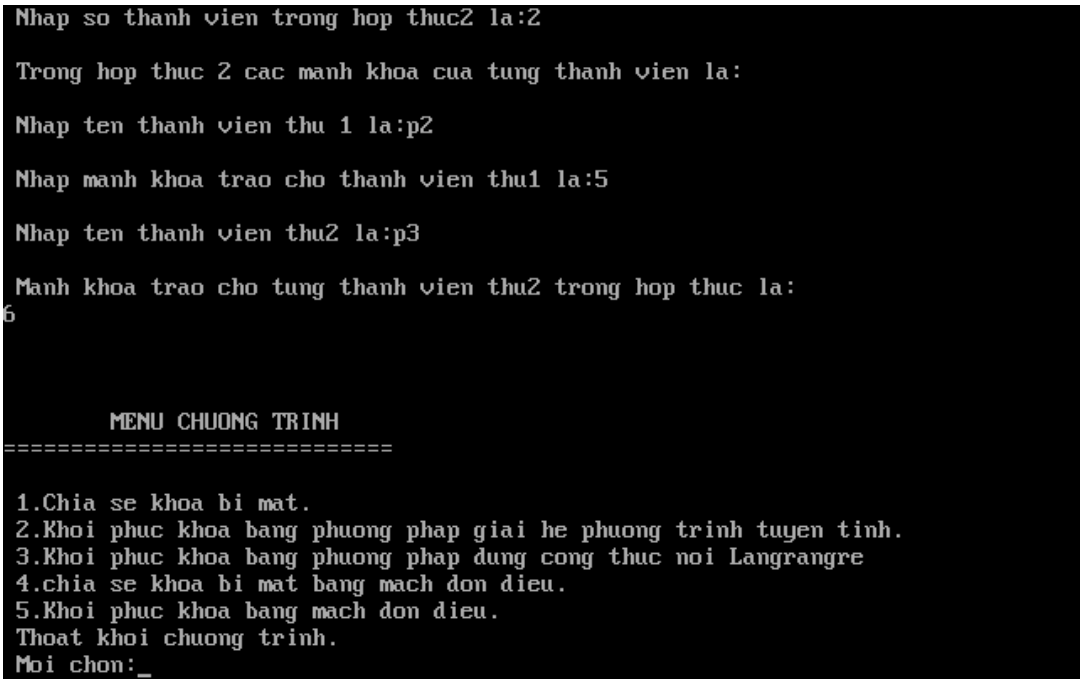
```
x[2]=2

x[3]=3

Nhap cac manh khoa da trao cho moi thanh vien P:
g[1]=19

g[2]=31

g[3]=47

Vay khoa bi mat khoi phuc lai la:11



       MENU CHUONG TRINH
============================

1.Chia se khoa bi mat.
2.Khoi phuc khoa bang phuong phap giai he phuong trinh tuyen tinh.
3.Khoi phuc khoa bang phuong phap dung cong thuc noi Langrangre
4.chia se khoa bi mat bang mach don dieu.
5.Khoi phuc khoa bang mach don dieu.
Thoat khoi chuong trinh.
Moi chon:_
```

Hình4. 6 và 4.7 Khôi phục khóa bí mật bằng phương pháp dùng công thức nội suy Lagrange

4.2.3.4 Chia sẻ khóa bí mật theo phương pháp bằng mạch đơn điệu

```
void giaothuc::machchiakhoa()
{
int n,m h[max],e;
char ten[32];
cout<<"\n Nhap so thanh vien tham gia:"; cin>>m;
cout<<"\n Nhap khoa bi mat can chia se:"; cin>>k;
cout<<"\n Nhap so tap con hop thuc co the tinh khoa k:"; cin>>n;
for(j=1; j<=n; j++)
{
cout<<"\n Nhap so thanh vien trong hop thuc"<<j<<" la:"; cin>>t;
cout<<"\n trong hop thuc"<<j<<" cac manh khoa cua tung thanh vien la:\n";
for(i=1; i<t; i++)
{
cout<<"\n Nhap so thanh vien thu"<<i<<" la:"; gets(ten);
cout<<"\n Nhap manh khoa trao cho tung thanh vien thu"<<i<<" la:"; cin>>h[i];
}
cout<<"\n Nhap ten thanh vien thu"<<t<<" la:"; gets(ten);
for(i=1; i<t; i++)
{
e=k – h[i];
}
cout<<"\n Manh khoa trao cho tung thanh vien thu"<<t<<" trong hop thuc la:\n"<<e;
}
}
```



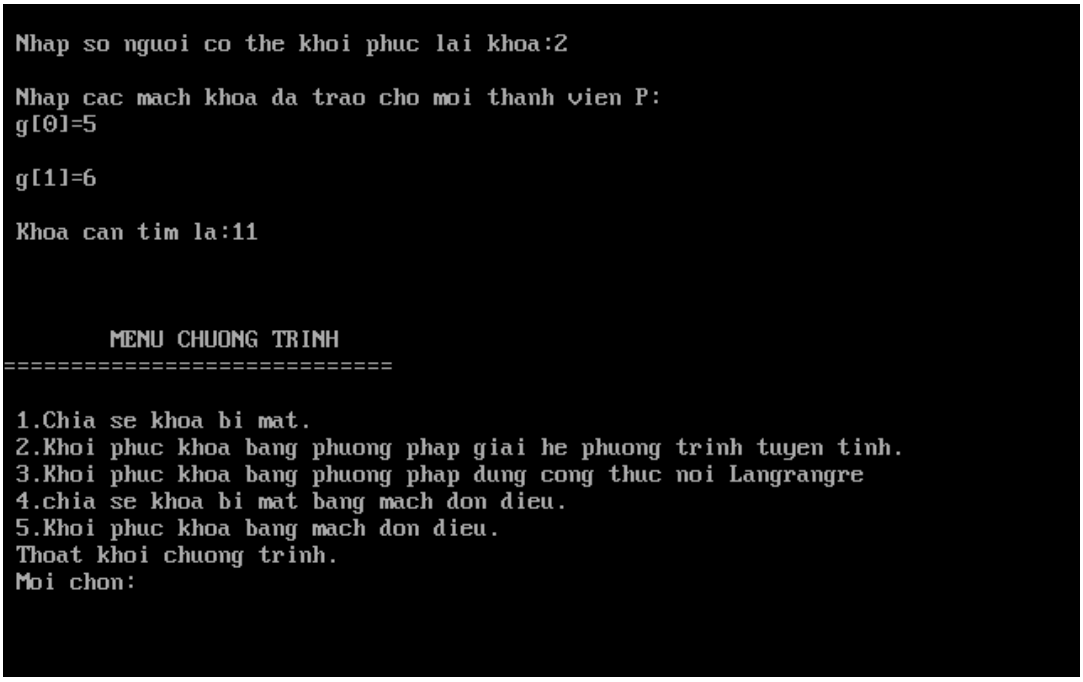


Hình 4.8 và 4.9 Chia sẻ bí mật bằng mạch đơn điệu

4.2.3.5 Khôi phục khóa bí mật theo phương pháp mạch đơn điệu

```
void giaothuc::machphuckhoa()
{
int h;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>t;
cout<<"\n Nhap cac manh khoa da trao cho moi thanh vien P:";
```

```
for(j=0; j<t; j++)
{
cout<<"\n g["<<j<<"]="; cin>>g[j];
}
h=0;
for(j=0; j<t; j++)
{
h=h+g[j];
}
cout<<"\n Khoa can tim la:"<<h;
}
```



Hình 4.10. Khôi phục khóa bí mật theo giao thức mạch đơn điệu

4.3 Mã nguồn mở của chương trình

Sau đây là mã nguồn của chương trình thử ngiệm:

```
#include<conio.h>
#include<stdio.h>
#include<math.h>
```

```
#include<iostream.h>
Const int max=30;
class giaothuc
{
Private:
int m, t, y, p;
float k;
int x[max], a[max], g[max];
int i, j, l;
Public:
void chiakhoa();
void giaihekhoiphuckhoa();
void congthuckhoiphuckhoa();
void machchiakhoa();
void machphuckhoa();
};
////////////////////////////////////////////////////////
void giaothuc::chiakhoa()
{
int h;
cout<<"\n Nhap khoa bi mat can chia se:"; cin>>k;
cout<<"\n Nhap so nguyen to p:"; cin>>p;
cout<<"\n Nhap so phan tu x:"; cin>>m;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>t;
cout<<"\n Nhap gia tri xi de trao cho moi thanh vien p:";
for(i=1; i<=m; i++)
{
```

```
cout<<"\n x["<<i<<"]="; cin>>x[i];
}
cout<<"\n Nhap bi mat t-1 pha tu trong Zp";
for(j=1; j<t; j++)
{
cout<<"\n a["<<j<<"]="; cin>>a[j];
}
for(i=1; i<=m; i++)
{
l=0;
for(j=1; j<t; j++)
{
h=pow(x[i],j);
l=l+a[j]*h;
}
y=k+l;
cout<<"\n Manh y"<<i<<" trao cho thanh vien P"<<i<<" la:"<<y;
}
}
////////////////////////////////////////////////////////////
void giaothuc::giaihekhoiphuckhoa()
{
int n, m, v, b;
float mx=0, ge, c, gt, h;
float G[max] [max], H[max] [max], A[max] [max], B[max] [max], M[max] [max];
cout<<"\n Nhap so nguyen to p:"; cin>>p;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>n;
```

```
cout<<"\n Nhap gia tri xi da trao cho moi thanh vien P:";
for(j=0; j<n; j++)
{
cout<<"\n x["<<j<<"]="; cin>>x[j];
}
cout<<"\n Nhap cac manh khoa da trao cho moi thanh vien P:";
for(l=0; l<n; n++)
{
cout<<"\n Y"<<l<<"="; cin>>B[l][0];
}
for(i=0; i<n; i++)
{
for(j=0; j<n; j++)
{
A[i][j]=pow(x [i],j);
}
}
cout<<"\n he Phuong trinh tuyen tinh la:\n";
for(i=0; i<n; i++)
{
for(j=0; j<n; j++)
{
If(j<n-1)
{
cout<<A[i][j]<<"a"<<j<<"+";
}
If(j==n-1)
```

```
{
cout<<A[i][j]<<”a”<<j;
}
}
cout<<”=”<<B[i][0];
cout<<”\n”;
}
cout<<”\n Vay nghiem cua he Phuong trinh la:”;
for(e=0; e<=n; e++)
{
for(i=0; i<n; i++)
{
for(j=0; j<n; j++)
{
G[i][j] = A[i][j];
}
}
for(j=0; j<n; j++)
{
If(j==e-1)
{
for(i=0; i<n; i++)
{
G[i][j] = B[i][j];
}
}
}
```

```
i=0; t=0;
while(t<n-1)
{
If(G[i][j]==0)
{
for(j=0; j<n; j++)
If(G[i][j]>mx)
{
mx= G[i][j];
v=j;
}
for(j=0; j<n; j++)
{
g=G[i][j];
G[i][j]= -G[v][j];
G[v][j]=g;
}
}
If(G[i][j]!=0)
{
for(k=i+1; k<n; k++)
{
c=G[k][i]/G[i][i];
h=c + (G[k][i]-c*G[i][i])/G[i][i];
for(j=0; j<n; j++)
{
G[k][j]= G[k][j]- G[i][j]*h;
```

```
}
}
i++;
}
t=t+1;
}
gt=1;
for(i=0; i<n; i++)
{
gt=gt*G[i][i];
}
If(e==0)
{
l=gt;
}
Else
{
b=gt/l;
if(b<p&& b>0)
{cout<<"\n a"<<e<<"="<<b;
if(e==1)
{
cout<<"\n Khoa duoc khoi phuc lai la k=al="<<b;
}
}
}
}
```

```
}
//////////////////////////////////////////////
void giaothuc::congthuckhoiphuckhoa()
{
float m, n, b;
cout<<"\n Nhap so nguyen to p:"; cin>>p;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>t;
cout<<"\n Nhap gia tri xi da trao cho moi thanh vien P:";
for(j=1; j<=t; j++)
{
cout<<"\n x["<<j<<"]="; cin>>x[j];
{
cout<<"\n Nhap cac manh khoa da trao cho moi thanh vien P:";
for(j=1; j<=t; j++)
{
cout<<"\n g["<<j<<"]="; cin>>g[j];
}
k=0;
for(l=1; l<=t; l++)
{ m=1;
for(j=1; j<=t; j++)
{ if(j!=1)
{ b=x[j]- x[l];
n= x[j]/b;
m=m*n;
}
}
```

```
k=k+g[l]*m;
}
cout<<"\n Vay khoa bi mat khoi phuc lai la:"<<k;
}
//////////////////////////////////////////////////////////////////
void giaothuc::machphuckhoa()
{
int h;
cout<<"\n Nhap so nguoi co the khoi phuc lai khoa:"; cin>>t;
cout<<"\n Nhap cac mach khoa da trao cho moi thanh vien P:";
for(j=0; j<t; j++)
{
cout<<"\n g["<<j<<"]="; cin>>g[j];
}
h=0;
for(j=0; j<t; j++)
{
h=h+g[j];
}
cout<<"\n Khoa can tim la:"<<h;
}
//////////////////////////////////////////////////////////////////////////
void giaothuc::machchiakhoa()
{
int n,m, h[max],e;
char ten[32];
cout<<"\n Nhap so thanh vien tham gia:"; cin>>m;
```

```
cout<<"\n Nhap khoa bi mat can chia se:"; cin>>k;
cout<<"\n Nhap so tap con hop thuc co the tinh khoa k:"; cin>>n;
for(j=1; j<=n; j++)
{
cout<<"\n Nhap so thanh vien trong hop thuc"<<j<<" la:"; cin>>t;
cout<<"\n Trong hop thuc"<<j<<" cac manh khoa cua tung thanh vien la"\n";
for(i=1; i<t; i++)
{
cout<<"\n Nhap ten thanh vien thu"<<i<<" la:"; gets(ten);
cout<<"\n Nhap manh khoa trao cho thanh vien thu"<<i<<" la:"; cin>>h[i];
}
cout<<"\n Nhap ten thanh vien thu"<<t<<" la:"; gets(ten);
for(i=1; i<t; i++)
{
e=k – h[i];
}
cout<<"\n Manh khoa trao cho tung thanh vien thu "<<t<<" trong hop thuc la:\n"<<e;
}
}
//////////////////CHUONG TRINH CHINH//////////////////////////////
void main()
{
Clrscr()
int chon;
giaothuc g;
do
{
```

```
cout<<"\n\n\n\n      MENU CHUONG TRINH"              ;
cout<<"\n ==========================================\n";
cout<<"\n 1.Chia se khoa bi mat";
cout<<"\n 2.Khoi phuc khoa bang Phuong phap giai he Phuong trinh tuyen tinh.:;
cout<<"\n 3. Khoi phuc khoa bang Phuong phap dung cong thuc noi suy Langrangre.";
cout<<"\n 4. Chia se khoa bi mat bang mach don dieu.";
cout<<"\n 5. Khoi phuc khoa bang mach don dieu.";
cout<<"\n 0. Thoat khoi chuong trinh.";
cout<<"\n Moi chon:"; cin>>chon;
switch(chon)
{
case 1: clrscr(); g.chiakhoa(); break;
case 2: clrscr(); g.giaihekhoiphuckhoa(); break;
case 3: clrscr(); g.congthuckhoiphuckhoa(); break;
case 4: clrscr(); g.machchiakhoa(); break;
case 5: clrscr(); g.machphuckhoa(); break;
}
}while(chon!=0);
getch();
}
```

KẾT LUẬN

Các ứng dụng trên mạng máy tính ngày càng trở lên phổ biến, thuận lợi và quan trọng thì yêu cầu về an toàn mạng, về an ninh dữ liệu càng trở lên cấp bách và cần thiết.

Thuật toán mã hóa được ứng dụng trong rất nhiều lĩnh vực như: xác thực người dùng, chữ kí số, mã hóa và xác thực dữ liệu…

Kết quả của luận văn của em gồm 2 phần chính:

1. **Tìm hiểu lí thuyết về mật mã**

Luận văn nghiên cứu lí thuyết về mật mã, thuật toán DES và lược đồ chia sẻ bí mật.

2. **Phần ứng dụng**

Luận văn đề cập đến vấn đè ứng dụng thuật toán DES và lược đồ chia sẻ bí mật vào thi tuyển sinh. Mô phỏng lược đồ chia sẻ bí mật bằng ngôn ngữ lập trình C.

Hạn chế luận văn đề cập tới phần lí thuyết nhiếu hơn những ứng dụng. Phần ứng dụng chưa được áp dụng trong thực tế do đó không thể tránh khỏi những thiếu sót, rất mong được sự góp ý của độc giả để luận văn của tôi được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

1. Phan Đình Diệu (2002). Lí thuyết mật mã và an toàn thông tin. NXB Đại học Quốc gia Hà Nội.
2. Lê Thị Sinh (2010) Nghiên cứu một số mô hình đảm bảo an ninh cơ sở dữ liệu và thử nghiệm ứng dụng, luaận ăn thạc sĩ Công nghệ thông tin, tr 28-25, Trường Đại học công nghệ - Đại học Quốc gia Hà Nội.
3. Dương Anh Đức (2008) Mã hóa và ứng dụng. Nhà xuất bản Đại học Quốc gia TPHCM.
4. Nguyễn Viết Kính (2007) Mã hóa. Bài giảng cho học viên cao học Trường Đại học Quốc gia Hà Nội.
5. Bảo mật thông tin, mô hình và ứng dụng, Nguyễn Xuân Dũng, 2007, Nhà xuất bản thông kê.
6. Douglas (1994) Mật mã lí thuyết và thực hành. Người dịch Nguyễn Bình.