**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**🙠**🕮**🙢**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**A close up of a sign

Description automatically generated**

**BÀI TẬP LỚN**

**AN TOÀN VÀ BẢO MẬT THÔNG TIN**

***ĐỀ TÀI***

**Ứng dụng thật toán DES và lược đồ chia sẻ bí mật vào thi tuyển sinh**

**- - - - - - - - - - 🙞🕮🙜- - - - - - - - - -**

|  |  |  |
| --- | --- | --- |
| **Giáo viên hướng dẫn** | **:** | **ThS. Trần Phương Nhung** |
| **Nhóm số** | **:** | **1** |
| **Mã lớp** | **:** | **20214IT6001001** |

**Hà Nội, 2022**

|  |
| --- |
|  |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------**  **A close up of a sign  Description automatically generated** |
| **BÀI TẬP LỚN**  **Môn: An toàn và bảo mật thông tin** |
|  |
| ***ĐỀ TÀI***  **Ứng dụng thật toán DES và lược đồ chia sẻ bí mật vào thi tuyển sinh** |
| **CBHD: *ThS. Trần Phương Nhung***  **Nhóm: 1** |
| **Thành viên nhóm:**   1. **Hoàng Minh An – 2020602174** 2. **Nguyễn Viết An – 2020603745** 3. **Đinh Đăng Duy Anh – 2021602842** 4. **Doãn Thị Lan Anh - 2019600624** 5. **Nguyễn Hoàng Anh – 2019600203** |
| **Hà Nội, 2022** |

**Mục lục**

[DANH MỤC HÌNH ẢNH 4](#_Toc113291102)

[DANH MỤC BẢNG 5](#_Toc113291103)

[Lời Nói Đầu 6](#_Toc113291104)

[Chương 1. Tổng quan 8](#_Toc113291105)

[1.1 Hệ mã hóa khối DES 8](#_Toc113291106)

[1.1.1. Lịch của DES 8](#_Toc113291107)

[1.1.2. Mật mã khối 9](#_Toc113291108)

[1.2 Kỹ thuật hệ mã hóa DES 10](#_Toc113291109)

[1.3 Ưu nhược điểm 13](#_Toc113291110)

[1.3.1 Ưu Điểm 13](#_Toc113291111)

[1.3.2 Các yếu điểm của DES 13](#_Toc113291112)

[1.4 Sơ đồ khối 17](#_Toc113291113)

[1.5 Thuật toán 18](#_Toc113291114)

[1.5.1 Quá Trình Mã Hóa 21](#_Toc113291115)

[1.5.2 Quá trình giải mã : 23](#_Toc113291116)

[1.5.3 Hàm F 23](#_Toc113291117)

[1.5.4 Quá trình tạo khóa con 24](#_Toc113291118)

[1.5.5 Hàm (ánh xạ ) mở rộng (E) 26](#_Toc113291119)

[1.5.6 Hộp S – Box 27](#_Toc113291120)

[1.5.7 Hộp P – Box 29](#_Toc113291121)

[1.6 Lập mã DES 30](#_Toc113291122)

[1.7 Kỹ thuật chia sẻ khóa (Secret Sharing) 34](#_Toc113291123)

[1.7.1 Khái niệm về chia sẻ bí mật 34](#_Toc113291124)

[1.7.2 Các sơ đồ chia sẻ bí mật 36](#_Toc113291125)

[1.8 Ứng dụng lược đồ chia sẻ bí mật của Lagarange để phân phối khóa. 40](#_Toc113291126)

[Chương 2. Kết Quả Nghiên Cứu 43](#_Toc113291127)

[2.1 Giới thiệu 43](#_Toc113291128)

[2.2 Nội dung thuật toán. 43](#_Toc113291129)

[2.2.1 Sơ đồ 43](#_Toc113291130)

[2.2.2 Các bước thực hiện 44](#_Toc113291131)

[2.3 Thiết kế, cài đặt chương trình demo thuật toán 46](#_Toc113291132)

[2.3.1 Giao diện chương trình demo 46](#_Toc113291133)

[2.4 Cài đặt và triển khai 50](#_Toc113291134)

[2.5 Thực Hiện Bài Toán 52](#_Toc113291135)

[2.5.1 Phân công công việc 52](#_Toc113291136)

[2.5.2 Hoàng Minh An 52](#_Toc113291137)

[2.5.3 Nguyễn Viết An 53](#_Toc113291138)

[2.5.4 Đinh Đăng Duy Anh 53](#_Toc113291139)

[2.5.5 Doãn Thị Lan Anh 54](#_Toc113291140)

[2.5.6 Nguyễn Hoàng Anh 54](#_Toc113291141)

[Chương 3. Phần Kiến Thức Lĩnh Hội Và Bài Học Kinh Nghiệm 55](#_Toc113291142)

[3.1 Nội dung đã thực hiện. 55](#_Toc113291143)

[3.2 Hướng phát triển. 56](#_Toc113291144)

[Tài Liệu Tham Khảo 57](#_Toc113291145)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 Biểu diễn dãy 64 bit x thành 2 thành phần L và R 9](#_Toc113290343)

[Hình 1.2 Quy trình phát sinh dãy Li Ri từ dãy Li-1 Ri-1 và khóa Ki 11](#_Toc113290344)

[Hình 1.3 Sơ đồ khối quá trình sinh khóa 16](#_Toc113290345)

[Hình 1.4 Sơ đồ khối quá trình sinh khóa 17](#_Toc113290346)

[Hình 1.5 Sơ đồ mã hóa DES 19](#_Toc113290347)

[Hình 1.6 Sơ đồ một vòng DES 20](#_Toc113290348)

[Hình 1.7 Sơ đồ hàm F 23](#_Toc113290349)

[Hình 1.8 Sơ đồ tạo khóa con 24](#_Toc113290350)

[Hình 1.9 DES và quá trình phân phối khóa 41](#_Toc113290351)

[Hình 2.1 Sơ đồ quy trình bảo mật đề thi gửi từ nơi ra đề đến nơi tổ chức thi 43](#_Toc113290352)

[Hình 2.2 Sơ đồ quy trình bảo mật đề thi gửi từ nơi ra đề đến nơi tổ chức thi 46](#_Toc113290353)

# DANH MỤC BẢNG

[Bảng 1.1 Các khóa yếu của DES 14](#_Toc113291092)

[Bảng 1.2 Các khóa nửa yếu của DES 14](#_Toc113291093)

[Bảng 1.3 Hoán vị IP 22](#_Toc113291094)

[Bảng 1.4 Hoán vị IP-1 22](#_Toc113291095)

[Bảng 1.5 Hoán vị PC-1 25](#_Toc113291096)

[Bảng 1.6 Bảng dịch bit tại các vòng lặp của DES 26](#_Toc113291097)

[Bảng 1.7 Hoán vị PC-2 26](#_Toc113291098)

[Bảng 1.8 Hàm mở rộng E 27](#_Toc113291099)

[Bảng 1.9 Hộp S-Box 29](#_Toc113291100)

[Bảng 1.10 Bảng hoán vị P 29](#_Toc113291101)

# Lời Nói Đầu

Với sự bùng nổ mạnh của công nghệ thông tin và sự phát triển của mạng Internet nên việc trao đổi thông tin trở nên dễ dàng hơn bao giờ hết. Tuy nhiên, phát sinh thêm một vấn đề ngày càng trở nên cấp bách và cần thiết về yêu cầu an toàn mạng, an ninh dữ liệu, bảo mật thông tin trong môi trường mạng cũng như trong thực tiễn.

Trên thế giới có nhiều quốc gia và nhà khoa học nghiên cứu vẫn đề bảo mật, đưa ra nhiều thuật toán giúp thông tin không bị đánh cắp hoặc nếu bị lấy cắp cũng không sử dụng được. Trong các giải pháp đó là an toàn thông tin bằng mật mã. Ở đề tài này nhóm em đề cập tới thuật toán mã hóa DES (Data Encryption Standard) từng được Liên bang Mỹ và nhiều quốc gia trên thế giới sử dụng. Tuy rằng DES hiện nay không còn được đánh giá cao về độ an toàn tuyệt đối, nhưng nó vẫn được ứng dụng trong nhiều lĩnh vực thực tiễn.

Bên cạnh mã hóa thông tin, lược đồ chia sẻ bí mật cũng được dùng để chia nhỏ thông tin trong quá trình truyền đi để đảm bảo an toàn dữ liệu. Sơ đồ chia sẻ bí mật thường được sử dụng để chia sẻ mật khẩu, khóa mã hóa trong đó có khóa mã hóa của DES.

Để ứng dụng 2 phương pháp trên vào thực tiễn, được sự hướng dẫn của cô Trần Phương Nhung, chúng em lựa chọn đề tài ***“Ứng dụng mã hóa bảo mật DES và lược đồ chia sẻ bí mật vào thi tuyển sinh”*** với mong muốn áp dụng kiến thức đã học, giải quyết bài toán bảo mật đề thi trong thi tuyển sinh.

Đề tài được hoàn thành bằng sự cộng tác của các thành viên nhóm cùng sự hướng dẫn của cô Trần Phương Nhung. Nội dung đề tài được hoàn thành dựa trên những lý thuyết đã học về Chuẩn dữ liệu DES cùng nhiều tài liệu tham khảo khác tuy nhiên không tránh khỏi thiếu xót mong nhận thêm phản ánh và góp ý từ phía giảng viên và quý bạn đọc.

# Tổng quan

## 1.1 Hệ mã hóa khối DES

### Lịch của DES

Vào những năm đầu thập kỷ 70, nhu cầu có một chuẩn chung về thuật toán mật mã đã trở nên rõ ràng. Các lý do chính là:

* Sự phát triển của công nghệ thông tin và của nhu cầu an toàn & bảo mật thông tin: sự ra đời của các mạng máy tính tiền thân của Internet đã cho phép khả năng hợp tác và liên lạc số hóa giữa nhiều công ty, tổ chức trong các dự án lớn của chính phủ Mỹ.
* Các thuật toán mã hóa cổ điển không thể đảm bảo được tính tin cậy đòi hỏi cao.
* Các thiết bị khác nhau đòi hỏi sự trao đổi thông tin mật mã thống nhất, chuẩn.

Một chuẩn chung cần thiết phải có với các thuộc tính như:

* Bảo mật ở mức cao.
* Thuật toán được đặc tả và hoàn toàn công khai, tức là tính bảo mật không được phép dựa trên những những đặc điểm của thuật toán mã hóa.
* Việc cài đặt phải dễ dàng để đem lại tính kinh tế.
* Phải mềm dẻo để áp dụng được cho muôn vàn nhu cầu ứng dụng.

Năm 1972, Viện tiêu chuẩn và công nghệ quốc gia Hoa kỳ (National Institute of Standards and Technology-NIST) đặt ra yêu cầu xây dựng một thuật toán mã hoá bảo mật thông tin với yêu cầu là dễ thực hiện, sử dụng được rộng rãi trong nhiều lĩnh vực và mức độ bảo mật cao. Năm 1974, IBM giới thiệu thuật toán Lucifer, thuật toán này đáp ứng hầu hết các yêu cầu của NIST. Sau một số sửa đổi, năm 1976, Lucifer được NIST công nhận là chuẩn quốc gia Hoa kỳ và được đổi tên thành Data Encryption Standard (DES).

DES là thuật toán mã hoá bảo mật được sử dụng rộng rãi nhất trên thế giới, tồn tại trong nhiều lĩnh vực như ngân hàng, thương mại, thông tin… và vẫn được tin dùng trong 2 thập kỷ sau đó, trước khi bị thay thế bởi AES.

### Mật mã khối

Các hệ mã hóa cổ điển đều có điểm đặc trưng là từng ký tự của bản rõ được mã hóa tách biệt. Điều này làm cho việc phá mã trở nên dễ dàng hơn. Chính vì vậy, trên thực tế người ta sử dụng một kiểu mật mã khác, trong đó từng khối ký tự của bản rõ được mã hóa cùng một lúc như là một đơn vị mã hóa đồng nhất. Trong kiểu mã hóa này, các tham số quan trọng là kích thước (độ dài) của mối khối và kích thước khóa.

*Điều kiện để mã hóa khối an toàn:*

* Kích thước khối phải đủ lớn để chống lại phương án tấn công bằng phương pháp thống kê. Tuy nhiên điều này sẽ dẫn đến thời gian mã hóa sẽ tăng lên.
* Không gian khóa, tức chiều dài khóa phải đủ lớn để chống lại phương án tấn công bằng vét cạn. Tuy nhiên khóa phải đủ ngắn để tạo khóa, phân phối và lưu trữ khóa được dễ dàng.

Khi thiết kế một hệ mã khối, phải đảm bảo hai yêu cầu sau:

* Sự hỗn loạn (confusion): sự phụ thuộc giữa bản rõ và bản mã phải thực sự phức tạp để gây khó khăn với việc tìm quy luật thám mã. Mối quan hệ này tốt nhất là phi tuyến.
* Sự khuếch tán (diffusion): Mỗi bit của bản rõ và khóa phải ảnh hưởng lên càng nhiều bit của bản mã càng tốt.

Trong khi sự hỗn loạn (confusion) được tạo ra bằng kỹ thuật thay thế thì sự khuếch tán (diffusion) được tạo ra bằng kỹ thuật hoán vị. Hệ mã hóa khối được xem xét trong tài liệu này đều thỏa mãn các yêu cầu đó.

Ngoài hệ mã hóa khối được trình bày trong tài liệu (DES) còn có rất nhiều các hệ ãm hóa khối khác đã phát triển qua thời gian (tại các quốc gia khác nhau và ứng dụng trong các lĩnh vực khác nhau), có thể kể ra như: Lucifer (1969), Madryga (1984), NewDES(1985), FEAL, REDOC, LOKI (1990), Khufu and Khafre (1990), RC2, RC4, IDEA (1990), MB, CA-1.1, Shipjack, GOST, CAST, ca, SAFER, 3-Way, Crab, SXAL8/MBAL, RC5, RC6, …

Đặc điểm chung của các hệ mã khối là quá trình mã hóa làm việc với khối dữ liệu (thường dạng xâu bit) có kích thước khác nhau (tối thiểu là 64 bit), khóa của các hệ mã cũng là một xâu bit có độ dài cố định (56 bit với DES, và các hệ mã khác là 128, 256 hoặc thậm chí là 512 bit). Tất cả các hệ mã này đều dựa trên lý thuyết của Shannon đưa ra năm 1949 và nếu mang mã hóa hai bản rõ giống nhau sẽ thu được cùng một bản mã. Hoạt động của các hệ mã khối thường được qua một số lần lặp, mỗi lần sẽ sử dụng một khóa con được sinh ra từ khóa chính.

## 1.2 Kỹ thuật hệ mã hóa DES

Trong phương pháp DES, kích thước khối là 64 bit. DES thực hiện mã hóa dữ liệu qua 16 vòng lặp mã hóa, mỗi vòng sử dụng một khóa chu kỳ 48 bit được tạo ra từ khóa ban đầu có độ dài 56 bit. DES sử dụng 8 bảng hằng số S-box để thao tác

Quá trình mã hóa của DES có thể tóm tắt như sau : Biểu diễn thông điệp nguồn x∈ P bằng dãy 64 bit. Khóa k có 56 bit. Thực hiện mã hóa theo 3 giai đoạn :

1. Tạo dãy 64 bit x0 bằng cách hoán vị x theo hoán vị IP (Initial Permutation)

Biểu diễn x0 =IP(x)=L0R0, L0 gồm 32 bit bên trái của x0 .R0 gồm 32 bit bên phải của x0.

|  |  |
| --- | --- |
| L0 | R0 |

X0

Hình 1.1 Biểu diễn dãy 64 bit x thành 2 thành phần L và R

2. Thực hiện 16 vòng lặp từ 64 bit thu được và 56 bit của khóa k (chỉ sử dụng 48 bit của khóa k trong mỗi vòng lặp). 64 bit kết quả thu được qua mỗi vòng lặp sẽ là đầu vào cho vòng lặp sau. Các cặp từ 32 bit Li, Ri (với 1 ≤ I ≤ 16 ) được xác định theo quy tắc sau:

Li=Ri-1

Ri=Li-1⊕ f(Ri-1, Ki)

Với ⊕ biểu diễn phép toán XOR trên hai dãy bit, K1, K2,...,K16 là các dãy 48 bit phát sinh từ khóa K cho trước ( Trên thực tế, mỗi khóa Ki được phát sinh bằng cách hoán vị các bit trong khóa K cho trước)

3. Áp dụng hoán vị ngược IP-1 đối với dãy bit R16L16, thu được từ y gồm 64 bit. Như vậy, y=IP-1 (R16L16)

Hàm f được sử dụng ở bước 2 là hàm số gồm 2 tham số: Tham số thứ nhất A là một dãy 32 bit , tham số thứ hai J là một dãy 48 bit. Kết quả của hàm f là một dãy 32 bit. Các bước xử lý của hàm f (A,J) như sau:

Tham số thứ nhất A (32 bit) được mở rộng thành dãy 48 bit được phát sinh từ A bằng cách hoán vị theo một thứ tự nhất định 32 bit của A, trong đó có 26 bit của A được lặp lại 2 lần trong E (A).

Hình 1.2 Quy trình phát sinh dãy Li Ri từ dãy Li-1 Ri-1 và khóa Ki

Li-1 Ri-1

Li  Ri

Ki

Thực hiện phép toán XOR cho hai dãy 48 bit E(A) và J, ta thu được một dãy 48 bit B. Biểu diễn B thành từng nhóm 6 bit như sau: B=B1 B2 B3 B4 B5 B6 B7 B8 Sử dụng 8 ma trận S1, S2, ..., S8 mỗi ma trận Si có kích thước 4x16 và mỗi dòng của ma trận nhận đủ 16 giá trị từ 0 đến 15. Xét dãy gồm 6 bit Bj= b1 b2 b3 b4 b5 b6, Sj(Bj) được xác định bằng giá trị của phần tử tại dòng r cột c của Sj, trong đó, chỉ số dòng r có biểu diễn nhị phân là b1 b6 , chỉ số cột c có biểu diễn nhị phân là b2 b3 b4 b5. Bằng cách này, ta xác định được các dãy 4 bit Cj=Sj(Bj), 1 ≤ j ≤ 8.

Tập hợp các dãy 4 bit Cj lại, ta có được dãy 32 bit C= C1 C 2 C3 C4 C5 C6 C7 C8. Dãy 32 bit thu được bằng cách hoán vị C theo một quy luật P nhất định chính là kết quả của hàm F(A,J).

Quá trình giải mã diễn ra tương tự nhưng với các khóa con ứng dụng vào các vòng theo thứ tự ngược lại

Có thể hình dung đơn giản là phần bên phải trong mỗi vòng (sau khi mở rộng input 32 bit thành 8 ký tự 6 bit – xâu 48 bit) sẽ thực hiện một tính toán thay thế phụ thuộc khóa trên mỗi ký tự trong xâu 48 bit, và sau đó sử dụng một phép chuyển bit cố định để phân bố lại các bit của các ký tự kết quả hình thành nên output 32 bit.

Các khóa con Ki (chứa 48 bit của K) được tính bằng cách sử dụng các bảng PC1 và PC2 (Permutation Choice 1 và 2). Trước tiên 8 bit ( K8, K16, …, K64) của K bị bỏ đi (áp dụng PC1). 56 bit còn lại được hoán vị và gán cho hai biến 28 bit C và D sẽ được quay 1 hoặc 2 bit, và các khóa con 48 bit Ki được chọn từ kết quả của việc ghép hai xâu với nhau.

Như vậy, ta có thể mô tả toàn bộ thuật toán sinh mã DES dưới dạng công thức như sau:

**Y = IP-1 • f16 • T • f15 • T • ... • f2 • T • f1 • IP(X)**

Trong đó :

- T mô tả phép hoán vị của các khối Li, RI (1 ≤ i ≤ 15).

- fi mô tả việc dùng hàm f với khóa Ki (1 ≤ i≤ 16)

## Ưu nhược điểm

### 1.3.1 Ưu Điểm

- Có tính bảo mật cao

- Công khai, dễ hiểu

- Nó có thể triển khai trên thiết bị điện tử có kích thước nhỏ

### Các yếu điểm của DES

#### Tính Bù

Nếu ta ký hiệu  là phần bù của u (ví dụ : 0100101 là phần bù của 1011010) thì des có tính chất sau

y = DES (x,k) → = DES (  ,)

Cho nên nếu ta biết mã y được mã hóa từ thông tin x với khóa K thì ta suy được bản mã  được mã hóa từ bản rõ  với khóa . Tính chất này là một yếu điểm của DES bởi vì qua đó đối phương có thể loại bỏ đi một số khóa phải thử khi tiến hành thử giải mã theo kiểu vét cạn

#### Khóa yếu

Khóa yếu là các khóa mà theo thuật toán sinh khóa con thì tất cả 16 khóa con đều như nhau : K1=K2=... =K16. Điều đó khiến cho việc mã hóa và giải mã đối với khóa yếu là giống hệt nhau

Bảng 1.1 Các khóa yếu của DES

|  |  |  |
| --- | --- | --- |
| **Khóa yếu (Hex)** | **C0** | **D0** |
| 0101 0101 0101 0101  FEFE FEFE FEFE FEFE  1F1F 1F1F 0E0E 0E0E  E0E0 E0E0 F1F1 F1F1 | {0}28  {1}28  {0}28  {1}28 | {0}28  {1}28  {1}28  {0}28 |

Đồng thời còn có 6 cặp khóa nửa yếu (semi-weak key) khác với thuộc tính như sau :

y= DES(x,k1) và y=DES(x,k2)

Nghĩa là với 2 khóa khác nhau nhưng mã hóa cùng một bản mã từ cùng một bản rõ :

Bảng 1.2 Các khóa nửa yếu của DES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C0 | D0 | Semi-weak key(Hex) | | C0 | D0 |
| {01}14  {01}14  {01}14  {01}14  {0}28  {1}28 | {01}14  {10}14  {0}28  {1}28  {01}14  {01}14 | 01FE 01FE 01FE 01FE  1FE0 1FE0 1FE0 1FE0  01E0 01E0 01F1 01F1  1FFE 1FFE 0EFE 0EFE  011F 011F 010E 010E  E0FE E0FE F1FE F1FE | FE01 FE01 FE01 FE01  E01F E01F E01F E01F  E001 E001 F101 F101  FE1F FE1F FE0E FE0E  1F01 1F01 0E01 0E01  FEE0 FEE0 FEF1 EF1 | {10}14  {10}14  {10}14  {10}14  {0}28  {1}28 | {10}14  {01}14  {0}28  {1}28  {10}14  {10}14 |

#### DES có cấu trúc đại số

Với 64 bit khối bản rõ có thể được ánh xạ lên tất cả các vị trí của khối 64 bit khối bản mã trong 264 cách. Trong thuật toán DES, với 56 bit khóa có thể cho chúng ta 256 (khoảng 1017 ) vị trí ánh xạ. Với việc đa mã hóa thì không gian ánh xạ còn lớn hơn. Tuy nhiên điều này chỉ đúng nếu việc mã hóa DES là không cấu trúc

Với DES có cấu trúc đại số thì việc đa mã hóa sẽ được xem ngang bằng với việc đơn mã hóa. Ví dụ như có hai khóa bất kỳ K1 và K2 thì sẽ luôn được khóa K3 như sau :

EK2(EK1(X))=EK3(X)

Nói một cách khác, việc mã hóa DES mang tính chất “nhóm”, đầu tiên mã hóa bản rõ bằng khóa K1 sau đó là khóa K2 sẽ giống với việc mã hóa ở khóa K3. Điều này thực sự quan trọng nếu sử dụng DES trong đa mã hóa. Nếu một “nhóm” được phát với cấu trúc hàm quá nhỏ thì tính an toàn sẽ giảm.

#### Không gian khóa K

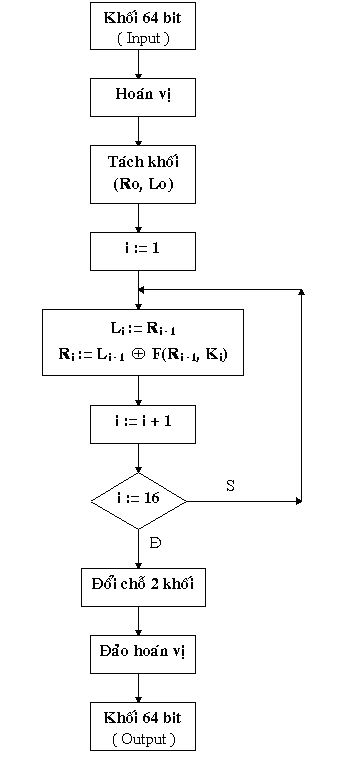
DES có 256 = 1017 khóa. Nếu chúng ta biết được một cặp “tin/mã” thì chúng ta có thể thử tất cả 1017 khả năng này để tìm ra khóa cho kết quả khớp nhất. Giả sử như một phép thử mất 10-6s, thì chúng sẽ mất 1011s, tức 7300 năm. Nhưng với các máy tính được chế tạo theo xử lý song song. Chẳng hạn với 107 con chip mã DES chạy song song thì bây giờ mỗi một con chipset chỉ phải chịu trách nhiệm tính toán với 1010 phép thử. Chipset mã DES ngày nay có thể xử lý tốc độ 4.5x107 bit/s tức có thể làm được hơn 105 phép mã DES trong một giây.

Vào năm 1976 và 1977, Dieffie và Hellman đã ước lượng rằng có thể chế tạo được một máy tính chuyên dụng để vét cạn không gian khóa DES trong ½ ngày với cái giá 20 triệu đô la. Năm 1984, chipset mã hóa DES với tốc độ mã hóa 256000 lần/giây. Năm 1987, đã tăng lên 512000 lần/giây. Vào năm 1993, Michael Wiener đã thiết kế một máy tính chuyên dụng với giá 1 triệu đô la sử dụng phương pháp vét cạn để giải mã DES trung bình trong vòng 3,5 giờ (và chậm nhất là 7 giờ).

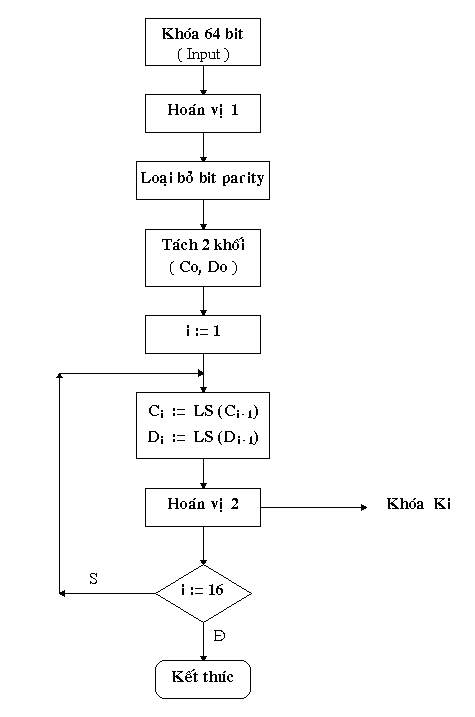
Đến năm 1990, hai nhà toán học người Do Thái – Biham và Shamir – đã phát minh ra phương pháp mã hóa vi sai (diferential cryptanalyis), đây là một kỹ thuật sử dụng những phỏng đoán khác nhau trong bản rõ để đưa ra những thông tin trong bản mã. Với phương pháp này, Biham và Shamir đã chứng minh rằng nó hiệu quả hơn cả phương pháp vét cạn.

Phá mã vi sai là thuật toán xem xét những cặp mã khóa khác nhau, đây là những cặp mã hóa mà bản mã của chúng là khác biệt. Người ta sẽ phân tích tiến trình biến đổi của những cặp mã này thông qua các vòng của DES khi chúng được mã hóa với cùng một khóa K. Sau đó sẽ chọn hai bản rõ khác nhau một cách ngẫu nhiên hợp lý nhất. Sử dụng sự khác nhau của kết quả bản mã và gán cho những khóa khác nhau một cách phù hợp nhất. Khi phân tích nhiều hơn những cặp bản mã, chúng ta sẽ tìm ra một khóa được xem là đúng nhất.

## 1.4 Sơ đồ khối



Hình 1.3 Sơ đồ khối quá trình sinh khóa



Hình 1.4 Sơ đồ khối quá trình sinh khóa

## 1.5 Thuật toán

DES là thuật toán mã hóa khối, nó xử lý từng khối thông tin của bản rõ có độ dài xác định là 64 bit. Trước khi đi vào 16 chu trình chính, khối dữ liệu cần bảo mật được “bẻ” ra từng khối 64 bit, và từng khối 64 bit này sẽ được lần lượt đưa vào 16 vòng mã hóa DES để thực hiện.

**Input:** bản rõ M = m1m2 … m64, là một khối 64 bit, khóa 64 bit K = k1k2 . . . k64 (bao gồm cả 8 bit chẵn lẻ, việc thêm bit chẵn lẻ sao cho các đoạn khóa 8 bit có số bit 1 là lẻ)

**Output:** bản mã 64 bit C = c1 c2 … c64

1. Sinh khóa con. Tính các khóa con theo thuật toán sinh khóa con
2. (L0,R0) ← IP (m1 m2 . . . m64) (sử dụng bản hoán vị IP để hoán vị các bit, kết quả nhận được chia thành 2 nửa là L0 = m58 m50 . . . m8, R0 = m57 m49 . . . m7)
3. Với i chạy từ i=1 đến 16 thực hiện:

Tính các Li và Ri theo công thức:

Li=Ri-1;

Ri=Li-1 ⊕ f(Ri-1) trong đó f ( Ri-1, Ki )=P ( S ( E ( Ri-1 ) ⊕ Ki ) );

Việc tính f ( Ri-1 ) = P ( S ( E ( Ri-1 ) ⊕ Ki ) ) được thực hiện như sau:

🟆 Mở rộng Ri-1 = r1r2 . .. r32 từ 32 bit thành 48 bit bằng cách sử dụng hoán vị mở rộng E.

T ← E ( Ri-1 ) . ( Vì thế T = r32 r1 r2 . . . r32 r1)

🟆 T’ ← T ⊕ Ki. Biểu diễn T’ như là các xâu gồm 8 ký tự 6 bit T’ = ( B1, . . . ,B8 )

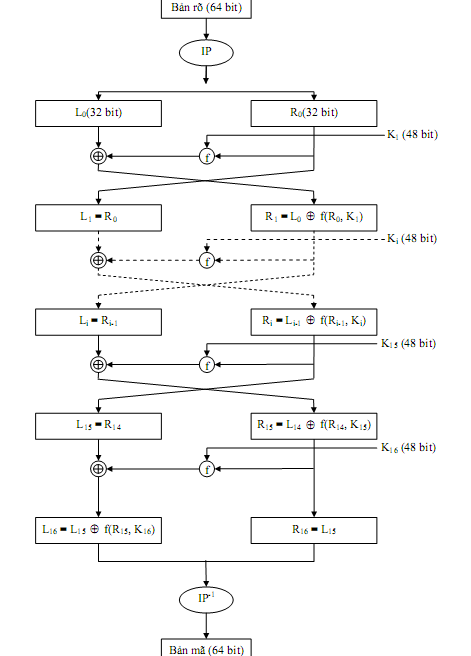
🟆 T’’ ← ( S1 ( B1 ) , S2 ( B2 ) , . . . , S8 ( B8 ) ). Trong đó Si ( Bi ) ánh xạ b1b2 . . . b6 thành các xâu 4 bit của phần tử thuộc hàng r và cột c của các bảng Si (S box) trong đó r = 2 \* b1 + b6 và c = b2 b3 b4 b5 là một số nhị phân từ 0 tới 15. Chẳng hạn S1 ( 011011) sẽ cho r = 1 và c = 3 và kết quả là 5 biểu diễn dưới dạng nhị phân là 0101.

🟆 T’’’ ← P ( T’’) trong đó P là hoán vị cố định để hoán vị 32 bit của

T’’ = t1 t2 . . . t32 sinh ra t16 t7 . . . t25

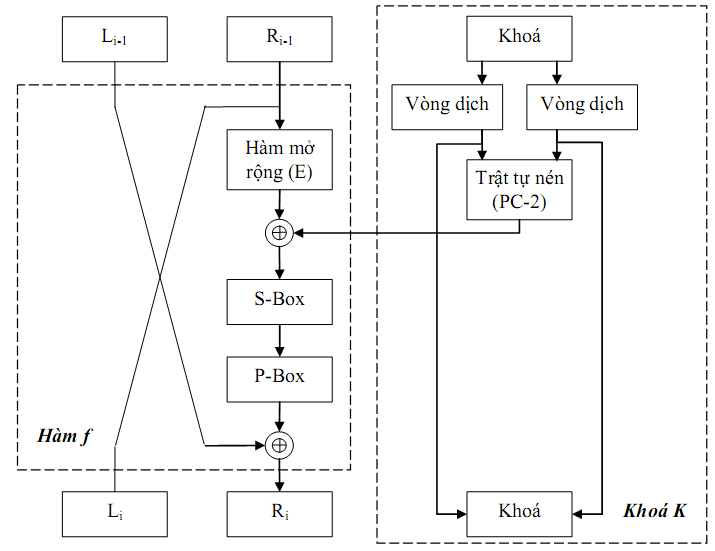
1. Khối từ b1 b2 . . . b64 ← ( R16, L16) ( đổi vị trí các khối cuối cùng L16, R16)

C ← IP-1 ( b1 b2 . . . b64) ( Biến đổi sử dụng IP-1, C = b40 b8 . . . b25).



Hình 1.5 Sơ đồ mã hóa DES

### 1.5.1 Quá Trình Mã Hóa

****

Hình 1.6 Sơ đồ một vòng DES

Chia làm 3 giai đoạn:

#### 1.5.1.1 Giai đoạn 1 :

Với bản rõ cho trước x, 1 xâu x' sẽ được tạo ra bằng cách hoán vị các bit của x theo hoán vị ban đầu IP:

x'=IP(x)=L0 R0

L0:32 bit đầu

R0:32 bit cuối

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Bộ chuyển vị IP*** | | | | | | | |
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 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 |

Bảng 1.3 Hoán vị IP

#### 1.5.1.2 Giai đoạn 2 :

Tính toán 16 lần lập theo 1 hàm xác định. Ta sẽ tính LiRi (1≤ i ≤ 16) theo quy tắc:

Li=Ri-1  
 Ri = Li-1⊕ f (Ri-1, Ki)

Với: ⊕ là toán tử Xor, k1,k2,k3.....k16 là xâu bit độ dài 48 bit được tính qua hàm khoá K (thực tế thì Ki là 1 phép hoán vị bit trong K)

#### 1.5.1.3 Giai Đoạn 3 :

Áp dụng hoán vị ngược IP-1 cho xâu bit R16 L16 ta thu được bản mã y:  
y = IP-1 (R16 L16)

+ Chú ý vị trí của R16 và L16.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Hoán vị nghịc đảo 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 |

Bảng 1.4 Hoán vị IP-1

### 1.5.2 Quá trình giải mã :

Do là 1 thuật toán đối xứng nên quá trình giải mã và mã hóa cũng gần giống nhau chỉ khác ở:

Li=Ri-1  
Ri = Li-1⊕ f (Ri-1, K16-i)

Khóa K của hàm F sẽ đi từ 16 🡪0

### 1.5.3 Hàm F

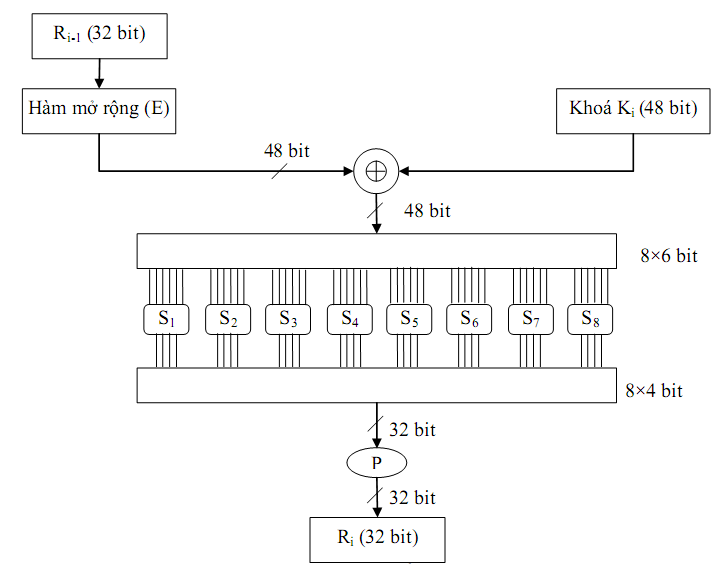
Đầu vào hàm f có 2 biến: biến 1: R là xâu bit có độ dài 32 bit, biến 2:K là xâu bit có độ dài 48 bit. Đầu ra của f là xâu bit có độ dài 32 bit.

- Biến thứ nhất Ri-1 được mở rộng thành một xâu bit có độ dài 48 bit theo một hàm mở rộng cố đinh E. Thực chất hàm mở rộng E ( Ri-1) là một hoán vị có lặp trong đó lặp lại 16 bit của Ri-1

- Tính E ( Ri-1 ) ⊕ Ki và viết kết quả thành 8 xâu 6 bit B1B2B3B4B5B6B7B8

- Đưa khối 8 bit Bi vào 8 bảng S1, S2, … .S8 ( được gọi là các hộp S-Box). Mỗi hộp S-Box là một bảng 4\*16 cố định có các cột từ 0 đến 15 và các hàng từ 0 đến 3. Với mỗi xâu 6 bit Bi = b1b2b3b4b5b6, ta tính được Si (B i) như sau: hai bit b1b6  xác định hàng r trong trong hộp Si, bốn bit b2b3b4b5 xác định cột c trong hộp S­i. Khi đó, Si (Bi) sẽ xác định phần tử Ci=Si ( r,c), phần tử này viết dưới dạng nhị phân 4 bit. Như vậy, 8 khối 6 bit Bi ( 1 ≤ i ≤ 8 ) sẽ cho ra 8 khối 4 bit Ci với ( 1 ≤ i ≤ 8 )

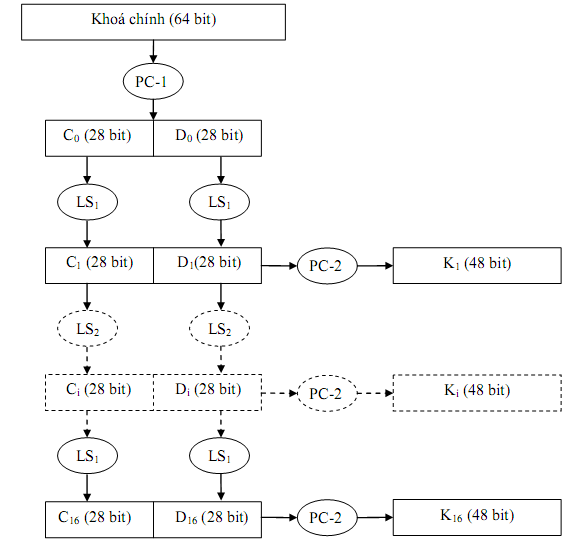
- Xâu bit C = C1C2C3C4C5C6C7C8 có độ dài 32 bit được hoán vị theo phép toán hoán vị P (hộp P-Box). Kết quả P(C) sẽ là kết quả của hàm f( Ri-1, K­i), và cũng chính Ri cho vòng sau



Hình 1.7 Sơ đồ hàm F

### 1.5.4 Quá trình tạo khóa con

- Mười sáu vòng lặp DES chạy cùng thuật toán như nhau nhưng với 16 khóa con khác nhau. Các khóa con đều được sinh ra từ khóa chính của DES bằng một thuật toán sinh khóa con.



Hình 1.8 Sơ đồ tạo khóa con

K là xâu có độ dài 64 bit, một bit trong 8 bit của byte sẽ được lấy ra dùng để kiểm tra phát hiện lỗi( thường thì các bit này ở vị trí 8, 16, 24, ...,64) tạo ra chuỗi 56 bit. Sau khi bỏ các bit kiểm tra ta sẽ hoán vị chuối 56 bit, 2 bước trên được thực hiện thông qua hoá vị ma trận PC-1.

**Bảng trật tự khóa 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 |

Bảng 1.5 Hoán vị PC-1

Ta chia PC-1 thành 2 phần:

C0: 28 bit đầu

D0: 28 bit cuối

Mỗi phần sẽ được xử lý 1 cách độc lập.

Ci=LSi(Ci-1)  
Di = LSi(Ci-1) với 1≤ i ≤ 16

+ LSi biểu diễn phép dịch bit vòng(cyclic shift) sang trái 1 hoặc 2 vị trí tuỳ thuộc vào i. Cyclic shift sang trái 1 bit nếu i=1, 2 , 9, 16 hoặc sang trái 2 bit nếu i thuộc các vị trí còn lại.

Ki=PC-2(CiDi).

+ Số bit dịch của các vòng:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Vòng lặp** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| **Số lần dịch trái** | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Bảng 1.6 Bảng dịch bit tại các vòng lặp của DES

+ PC-2 là hoán vị cố định sẽ hoán vị chuỗi CiDi 56 bit thành chuỗi 48 bit.

**Bảng trật tự nén 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 |

Bảng 1.7 Hoán vị PC-2

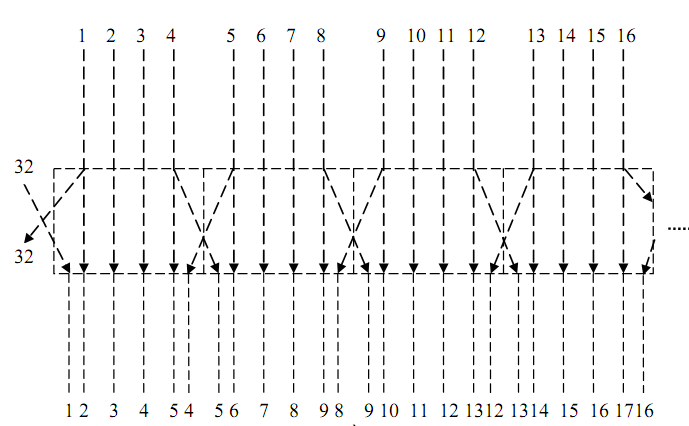
### 1.5.5 Hàm (ánh xạ ) mở rộng (E)

Hàm mở rộng (E) sẽ tăng độ dài từ Ri từ 32 bit lên 48 bit bằng cách thay đổi các thứ tự của các bit cũng như lặp lại các bit. Việc thực hiện này nhằm hai mục đích:

- Làm độ dài của Ri cùng cỡ với khóa K để thực hiện việc cộng modulo XOR.

- Cho kết quả dài hơn để có thể được nén trong suốt quá trình thay thế

Tuy nhiên, cả hai mục đích này nhằm một mục tiêu chính là bảo mật dữ liệu. Bằng cách cho phép 1 bit có thể chèn vào hai vị trí thay thế, sự phụ thuộc của các bit đầu ra với các bit đầu vào sẽ trải rộng ra. DES được thiết kế với điều kiện là mỗi bit của bản mã phụ thuộc vào mỗi bit của bản rõ và khóa.



Hỉnh 1.8: Sơ đồ của hàm mở rộng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Hàm mở rộng E** | | | | | |
| 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 |

Bảng 1.8 Hàm mở rộng E

Đôi khi nó được gọi là hàm E-Box, mỗi 4 bit của khối vào, bit thứ nhất và bit thứ tư tương ứng với 2 bit của đầu ra, trong khi bit thứ hai và ba tương ứng với 1 bit ở đầu ra

### 1.5.6 Hộp S – Box

- Mỗi hàng trong mỗi hộp S là hoán vị của các số nguyên từ 0 đến 15

- Không có hộp S nào là hàm Affine hay tuyến tính đối với các đầu vào của nó

- Sự thay đổi của một bit đầu vào sẽ dẫn đến sự thay đổi ít nhất hai bit đầu ra

- Đối với hộp S bất kỳ và với đầu vào x ( một xâu bit có độ dài bằng 6 bit) bất kỳ, thì S(x) và S (x ⊕ 001100) phải khác nhau ít nhất là 2 bit

Sau khi cộng modulo với khóa K, kết quả thu được chuỗi 48 bit chia làm 8 khối đưa vào 8 hộp S-Box. Mỗi hộp S-Box có 6 bit đầu vào và 4 bit đầu ra ( tổng bộ nhớ yêu cầu cho 8 hộp S-Box chuẩn DES là 256 bytes). Kết quả thu được là một chuỗi 32 bit tiếp tục vào hộp P-Box

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S1** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14  0  4  15 | | 4  15  1  12 | | 13  7  14  8 | | 1  4  8  2 | | 2  14  13  4 | | 15  2  6  9 | | 11  13  2  1 | | 8  1  11  7 | | 3  10  15  5 | | 10  6  12  11 | | 6  12  9  3 | | 12  11  7  14 | | 5  9  3  10 | | 9  5  10  0 | | 0  3  5  6 | 7  8  0  13 | |
| **S2** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15  3  0  13 | 1  13  14  8 | | 8  4  7  10 | | 14  7  11  1 | | 6  15  10  3 | | 11  2  4  15 | | 3  8  13  4 | | 4  14  1  2 | | 9  12  5  11 | | 7  0  8  6 | | 2  1  12  7 | | 13  10  6  12 | | 12  6  9  0 | | 0  9  3  5 | | 5  11  2  14 | | | 10  5  15  9 |
| **S3** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10  13  13  1 | 0  7  6  10 | | 9  0  4  13 | | 14  9  9  0 | | 6  3  8  6 | | 3  4  15  9 | | 15  6  3  8 | | 5  10  0  7 | | 1  2  11  4 | | 13  8  1  15 | | 12  5  2  14 | | 7  14  12  3 | | 11  12  5  11 | | 4  11  10  5 | | 2  15  14  2 | | | 8  1  7  12 |
| **S4** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7  13  10  3 | 13  8  6  15 | | 14  11  9  0 | | 3  5  0  6 | | 0  6  12  10 | | 6  15  11  1 | | 9  0  7  13 | | 10  3  13  8 | | 1  4  15  9 | | 2  7  1  4 | | 8  2  3  5 | | 5  12  14  11 | | 11  1  5  12 | | 12  10  2  7 | | 4  14  8  2 | | | 15  9  4  14 |
| **S5** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2  14  4  11 | 12  11  2  8 | | 4  2  1  12 | | 1  12  11  7 | | 7  4  10  0 | | 10  7  13  14 | | 11  13  7  2 | | 6  1  8  13 | | 8  5  15  6 | | 5  0  9  15 | | 3  15  12  0 | | 15  10  5  9 | | 13  3  6  10 | | 0  9  3  4 | | 14  8  0  5 | | | 9  6  14  3 |
| **S6** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12  10  9  4 | 1  15  14  3 | | 10  4  15  2 | | 15  2  5  12 | | 9  7  2  9 | | 2  12  8  5 | | 6  9  12  15 | | 8  5  3  10 | | 0  6  7  11 | | 13  1  0  14 | | 3  13  4  1 | | 4  14  10  7 | | 14  0  1  6 | | 7  11  13  0 | | 5  3  11  8 | | | 11  8  6  13 |
| **S7** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4  13  1  6 | 11  0  4  11 | | 2  11  11  13 | | 14  7  13  8 | | 15  4  12  1 | | 0  9  3  4 | | 8  1  7  10 | | 13  10  14  7 | | 3  14  10  9 | | 12  3  15  5 | | 9  5  6  0 | | 7  12  8  15 | | 5  2  0  14 | | 10  15  5  2 | | 6  8  9  3 | | | 1  6  2  12 |
| **S8** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13  1  7  2 | 2  15  11  1 | | 8  13  4  14 | | 4  8  1  7 | | 6  10  9  4 | | 15  3  12  10 | | 11  7  14  8 | | 1  4  2  13 | | 10  12  0  15 | | 9  5  6  12 | | 3  6  10  9 | | 14  11  13  0 | | 5  0  15  3 | | 0  14  3  5 | | 12  9  5  6 | | | 7  2  8  11 |

Bảng 1.9 Hộp S-Box

### 1.5.7 Hộp P – Box

Việc hoán vị này mang tính đơn ánh, nghĩa là một bit đầu vào sẽ cho một bit ở đầu ra, không bit nào được sử dụng 2 lần hay bị bỏ qua. Hộp P-Box thực chất chỉ là chức năng sắp xếp đơn thuần theo bảng sau:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 14 | 25 |

Bảng 1.10 Bảng hoán vị P

## 1.6 Lập mã DES

Đây là ví dụ về việc sử dụng DES. Giả sử ta mã hóa bản rõ sau trong dạng thập lục phân(Hexadecimal) *0123456789ABCDEF* sử dụng khóa thập lục phân *133457799BBCDFF1*

Khóa trong dạng nhị phân không có các bit kiểm tra sẽ là :

00010010011010010101101111001001101101111011011111111000.

Áp dụng IP , ta nhận được L0 và R0 (trong dạng nhị phân)

|  |  |  |
| --- | --- | --- |
| **L0**  **L1 = R0** | **=**  **=** | **11001100000000001100110011111111**  **11110000101010101111000010101010** |

16 vòng lặp mã được thể hiện như sau:

|  |  |  |
| --- | --- | --- |
| **E(R0)**  **K1**  **E(R0) ⊕ K1**  **S-box output**  **f(R0,K1)**  **L2 = R1** | **=**  **=**  **=**  **=**  **=**  **=** | **011110100001010101010101011110100001010101010101**  **000110110000001011101111111111000111000001110010**  **011000010001011110111010100001100110010100100111**  **01011100100000101011010110010111**  **00100011010010101010100110111011**  **11101111010010100110010101000100** |

|  |  |  |
| --- | --- | --- |
| **E(R1)**  **K2**  **E(R1) ⊕ K2**  **S-box output**  **f(R1, K2)**  **L3 = R2** | **=**  **=**  **=**  **=**  **=**  **=** | **011101011110101001010100001100001010101000001001**  **011110011010111011011001110110111100100111100101**  **000011000100010010001101111010110110001111101100**  **11111000110100000011101010101110**  **00111100101010111000011110100011**  **11001100000000010111011100001001** |

|  |  |  |
| --- | --- | --- |
| **E(R2)**  **K3**  **E(R2) ⊕ K3**  **S-box output**  **f(R2, K3)**  **L4 = R3** | **=**  **=**  **=**  **=**  **=**  **=** | **111001011000000000000010101110101110100001010011**  **010101011111110010001010010000101100111110011001**  **101100000111110010001000111110000010011111001010**  **00100111000100001110000101101111**  **01001101000101100110111010110000**  **10100010010111000000101111110100** |

|  |  |  |
| --- | --- | --- |
| **E(R3)**  **K4**  **E(R3) ⊕ K4**  **S-box output**  **f(R3, K4)**  **L5 = R4** | **=**  **=**  **=**  **=**  **=**  **=** | **010100000100001011111000000001010111111110101001**  **011100101010110111010110110110110011010100011101**  **001000101110111100101110110111100100101010110100**  **00100001111011011001111100111010**  **10111011001000110111011101001100**  **011101110** |

|  |  |  |
| --- | --- | --- |
| **E(R4)**  **K5**  **E(R4) ⊕ K5**  **S-box output**  **f(R4, K5)**  **L6 = R5** | **=**  **=**  **=**  **=**  **=**  **=** | **101110101110100100000100000000000000001000001010**  **011111001110110000000111111010110101001110101000**  **110001100000010100000011111010110101000110100010**  **01010000110010000011000111101011**  **00101000000100111010110111000011**  **10001010010011111010011000110111** |
| **E(R5)**  **K6**  **E(R5) ⊕ K6**  **S-box output**  **f(R5, K6)**  **L7 = R6** | **=**  **=**  **=**  **=**  **=**  **=** | **110001010100001001011111110100001100000110101111**  **011000111010010100111110010100000111101100101111**  **101001101110011101100001100000001011101010000000**  **01000001111100110100110000111101**  **10011110010001011100110100101100**  **11101001011001111100110101101001** |

|  |  |  |
| --- | --- | --- |
| **E(R6)**  **K7**  **E(R6) ⊕ K7**  **S-box output**  **f(R6, K7)**  **L8 = R7** | **=**  **=**  **=**  **=**  **=**  **=** | **111101010010101100001111111001011010101101010011**  **111011001000010010110111111101100001100010111100**  **000110011010111110111000000100111011001111101111**  **00010000011101010100000010101101**  **10001100000001010001110000100111**  **00000110010010101011101000010000** |

|  |  |  |
| --- | --- | --- |
| **E(R7)**  **K8**  **E(R7) ⊕ K8**  **S-box output**  **f(R7, K8)**  **L9 = R8** | **=**  **=**  **=**  **=**  **=**  **=** | **000000001100001001010101010111110100000010100000**  **111101111000101000111010110000010011101111111011**  **111101110100100001101111100111100111101101011011**  **01101100000110000111110010101110**  **00111100000011101000011011111001**  **11010101011010010100101110010000** |

|  |  |  |
| --- | --- | --- |
| **E(R8)**  **K9**  **E(R8) ⊕ K9**  **S-box output**  **f(R8, K9)**  **L10 = R9** | **=**  **=**  **=**  **=**  **=**  **=** | **011010101010101101010010101001010111110010100001**  **111000001101101111101011111011011110011110000001**  **100010100111000010111001010010001001101100100000**  **00010001000011000101011101110111**  **00100010001101100111110001101010**  **00100100011111001100011001111010** |

|  |  |  |
| --- | --- | --- |
| **E(R9)**  **K10**  **E(R9) ⊕ K10**  **S-box output**  **f(R9, K10)**  **L11 = R10** | **=**  **=**  **=**  **=**  **=**  **=** | **000100001000001111111001011000001100001111110100**  **101100011111001101000111101110100100011001001111**  **101000010111000010111110110110101000010110111011**  **11011010000001000101001001110101**  **01100010101111001001110000100010**  **10110111110101011101011110110010** |

|  |  |  |
| --- | --- | --- |
| **E(R10)**  **K11**  **E(R10) ⊕ K11**  **S-box output**  **f(R10, K11)**  **L12 = R11** | **=**  **=**  **=**  **=**  **=**  **=** | **010110101111111010101011111010101111110110100101**  **001000010101111111010011110111101101001110000110**  **011110111010000101111000001101000010111000100011**  **01110011000001011101000100000001**  **11100001000001001111101000000010**  **11000101011110000011110001111000** |

|  |  |  |
| --- | --- | --- |
| **E(R11)**  **K12**  **E(R11) ⊕ K12**  **S-box output**  **f(R11, K12)**  **L13 = R12** |  | **011000001010101111110000000111111000001111110001**  **011101010111000111110101100101000110011111101001**  **000101011101101000000101100010111110010000011000**  **01111011100010110010011000110101**  **11000010011010001100111111101010**  **01110101101111010001100001011000** |

|  |  |  |
| --- | --- | --- |
| **E(R12)**  **K13**  **E(R12)⊕ K13**  **S-box output**  **f(R12, K13)**  **L14 = R13** | **=**  **=**  **=**  **=**  **=**  **=** | **001110101011110111111010100011110000001011110000**  **100101111100010111010001111110101011101001000001**  **101011010111100000101011011101011011100010110001**  **10011010110100011000101101001111**  **11011101101110110010100100100010**  **00011000110000110001010101011010** |

|  |  |  |
| --- | --- | --- |
| **E(R13)**  **K14**  **E(R13)⊕ K14**  **S-box output**  **f(R13, K14)**  **L15 = R14** | **=**  **=**  **=**  **=**  **=**  **=** | **000011110001011000000110100010101010101011110100**  **010111110100001110110111111100101110011100111010**  **010100000101010110110001011110000100110111001110**  **01100100011110011001101011110001**  **10110111001100011000111001010101**  **11000010100011001001011000001101** |

|  |  |  |
| --- | --- | --- |
| **E(R14)**  **K15**  **E(R14)⊕ K15**  **S-box output**  **f(R14, K15)**  **L16 = R15** | **=**  **=**  **=**  **=**  **=**  **=** | **111000000101010001011001010010101100000001011011**  **101111111001000110001101001111010011111100001010**  **010111111100010111010100011101111111111101010001**  **10110010111010001000110100111100**  **01011011100000010010011101101110**  **01000011010000100011001000110100** |

|  |  |  |
| --- | --- | --- |
| **E(R15)**  **K16**  **E(R15)⊕ K16**  **S-box output**  **f(R15, K16)**  **R16** | **=**  **=**  **=**  **=**  **=**  **=** | **001000000110101000000100000110100100000110101000**  **110010110011110110001011000011100001011111110101**  **111010110101011110001111000101000101011001011101**  **10100111100000110010010000101001**  **11001000110000000100111110011000**  **00001010010011001101100110010101** |

Cuối cùng , áp dụng IP-1 cho ta nhận được bản mã trong dạng thập lục phân như sau :

*85E813540F0AB405*

## Kỹ thuật chia sẻ khóa (Secret Sharing)

### 1.7.1 Khái niệm về chia sẻ bí mật

Sơ đồ chia sẻ bí mật không phải là một lĩnh vực mới mẻ của an toàn bảo mật thông tin, nhưng hứa hẹn sẽ mang đến nhứng ứng dụng rộng khắp: khóa mã hóa , mã phóng tên lửa và số tài khoản ngân hàng, …

Sơ đồ chia sẻ bí mật chính là phương thức dùng đề chia một bí mật ra làm nhiều phần riêng biệt sau đó phân phối tới những người tham gia. Trong đó chỉ với số người được chỉ định trước mới có khả năng khôi phục bí mật bằng cách gộp những phần thông tin của họ, những người không được chỉ định sẽ không thu được bất kỳ thông tin gì về bí mật.

*Ý tưởng:* thông tin quan trọng cần bí mật, không nên trao cho một người nắm giữ, mà phải chia thông tin đó thành nhiều mảnh và trao cho mỗi người một hay một số mảnh. Thông tin gốc chỉ có thể được xem lại, khi mọi người giữ các mảnh thông tin đều nhất trí. Các mảnh thông tin được khớp lại để được thông gốc.

|  |  |
| --- | --- |
| - Thông tin cần giữ bí mật được chia thành nhiều mảnh: | |
| Thông tin bí mật | Các mảnh được chia sẻ |
| - Khi các mảnh được khớp lại sẽ cho ta thông tin ban đầu: | |
| Các mảnh cần để ghép lại thông tin | Thông tin bí mật được khôi phục |

*Yêu cầu:* để thực hiện công việc trên, phải sử dụng một sơ đồ gọi là Sơ đồ chia sẻ bí mật.

*Khái niệm chia sẻ bí mật:* Sơ đồ chia sẻ bí mật dùng để chia sẻ một thông tin cho m thành viên, sao cho chỉ những tập con hợp thức các thành viên mới có thể khôi phục lại thông tin bí mật, còn lại không ai có thể làm được điều đó.

*Ứng dụng:*

* Chia sẻ thông tin mật thành nhiều mảnh.
* Chia sẻ Passwod, khóa mật thành nhiều mảnh. Mỗi nơi, mỗi người hay mỗi máy tính cất giấu 1 mảnh.

*Các thành phần của sơ đồ chia sẻ bí mật:*

* Người phân phối bí mật (Dealer): Là người trực tiếp chia bí mật thành nhiều phần.
* Những người tham gia nhận dữ liệu từ Dealer (Participant), ký hiệu P.
* Nhóm có khả năng khôi phục bí mật (Acess structure): Là tập con của P trong đó có các tập con có khả năng khôi phục bí mật.

Ví dụ:Chìa khóa để mở két bạc là chìa khóa số được chủ két bạc D chia thành 3 mảnh khóa, có 3 thủ quỹ là P1, P2, P3. Mỗi thủ quỹ giữ một mảnh khóa. Chỉ có thủ quỹ P1 và P2 hoặc P2 và P3 hoặc P1 và P3 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: { P1, P2 }, { P2, P3}, {P1, P3}.

### Các sơ đồ chia sẻ bí mật

#### 1.7.2.1 Sơ đồ chia sẻ bí mật sơ khai

Một sơ đồ chia sẻ bí mật đảm bảo tính bảo mật là sơ đồ trong đó bất kỳ người nào có ít hơn t phần dữ liệu (là số lượng đủ để khôi phục bí mật) không có nhiều thông tin hơn một người không có dữ liệu.

Xem xét sơ đồ chia sẻ bí mật sơ khai trong đó cụm từ bí mật “password” được chia thành các phần “pa­­­\_\_\_”, “\_ss\_\_\_”, “\_\_\_wo\_” và “\_\_\_\_rd”. Một người không có một trong các phần bí mật đó chỉ biết mật khẩu có 8 chữ cái. Người đó sẽ phải đoán mật khẩu đó từ 268 = 208 tỷ khả năng có thể xảy ra. Một người có một phần trong số 4 phần của mật khẩu đó sẽ phải đoán 6 chữ cái tương đương với 266 = 308 tỷ khả năng. Hệ thống này không phải là một sơ đồ chia sẻ bí mật an toàn bởi vì một người tham gia có ít hơn t phần dữ liệu thu được một phần đáng kể thông tin về bí mật. Trong một sơ đồ chia sẻ bí mật an toàn, mặc dù một người tham gia chỉ thiếu một phần dữ liệu cũng có thể sẽ đối mặt với 268 = 208 tỷ khả năng.

#### 1.7.2.2 Sơ đồ chia sẻ bí mật Shamir

Ý tưởng về sơ đồ ngưỡng giới hạn của Shamir dựa trên tính chất: Hai điểm có thể định nghĩa một đường thẳng, 3 điểm định nghĩa được 1 parabol, 4 điểm định nghĩa được một hình lập phương, cứ như thế một cách tổng quát cần n+1 điểm để định nghĩa một đa thức bậc n.

***Bài toán***:

Cho t, m là các số nguyên dương, t ≤ m. Một sơ đồ ngưỡng A(t, m) 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.

*Ví dụ* : có m = 3 thủ quỹ giữ két bạc. Hãy xây dựng hệ thống sao cho bất kì t = 2 thủ quỹ nào cũng có thể mở được két bạc, nhưng từng người một riêng rẽ thì không thể. Đó là sơ đồ ngưỡng A(2,3).

Giá trị K được chọn bởi một thành viên đặc biệt được gọi là người phân phối (D), D∉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 các thành viên khác.

Một tập con các thành viên B 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| ≥ t thì họ có khả năng tính được K.

Nếu |B| < t thì hị không thể tính được K.

Gọi P là tập các giá trị được phân phối khóa K: P = {pi : 1 ≤ i ≤ m}

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

S tập mảnh: tập tất cả các mảnh có thể

* ***Khởi tạo:*** Chọn số nguyên tố p
  1. D chọn phần tử khác nhau, ≠ 0 trong Zp, (yêu cầu: , Tl: khác nhau, ≠ 0 trong Zp). D trao cho thành viên *Pi*. Giá trị là công khai.
* ***Phân phối mảnh khóa***

1. D chọn bí mật (ngẫu nhiên, độc lập) (*t – 1*) phần tử Zp là *a1, …, at* – 1.
2. Với *1 ≤ i ≤ m*, D tính: *yi = P(xi),*
3. Với *1 ≤ i ≤ m*, D sẽ trao mảnh *yi* cho *Pi*.

* ***Khôi phục khóa K từ t thành viên***

Giải hệ phương trình tuyến tính t ẩn, t phương trình

Vì *P(x) = K + a1 x1 + a2 x2 + …+ at-1 xt-1*

Các hệ số *K, a1, …, at-1* là các phần tử chưa biết của Zp, *a0 = K* là khóa.

Vì *yi j = P(xi j)*, nên ta có thể thu được t phương trình tuyến tính t ẩn *a0, a1, …, at-1*,

Nếu các phương trình độc lập tuyến tính thì sẽ có một nghiệm duy nhất và ta được giá trị khoá *a0 = K*.

#### 1.7.2.3 Chia sẻ bí mật dựa trên ý tưởng của Lagrange

Giả sử ta có n thực thể A1,A2,…,An và có 1 người được ủy quyền B biết được toàn bộ khóa bí mật S N.

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

1. B chọn một số nguyên tố P đủ lớn sao cho: với
2. B tiếp theo chọn (2n – 1) số một cách ngẫu nhiên:
3. B xác định một đa thức với các hệ số *a1, …, at-1* trên :
4. Bây giờ B gửi Aj (một cách công khai) cặp coi như mảnh riêng của Aj.

***Khôi phục bí mật S:***

Tất cả n người A1, …, An có thể hợp tác lại để khôi phục lại bí mật S bằng cách:

Khi đó dễ dàng xác định được *S = g(0) (x = 0)*

Ta có định lý như sau:

Nếu 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(vj)=f(vj) với 0 ≤ j < n.

Do đó, f – g là đa thức trên Zp có cấp bé hơn n, nhưng nó lại có ít nhất là n nghiệm khác nhau: là các số r thỏa mãn f(r) – g(r) = 0. Chứng tỏ rằng f(a) = g(a) với đặc biệt f(0) = g(0) = S. (ĐPCM)

*Ví dụ:*

*Ví dụ 1:* Có 3 người A1, A2, A3 muốn chia sẻ bí mật 472. Cho p = 1999 công khai. A chọn v1 = 626, v2 = 674, v3 = 93; a1 = 334, a2 = 223.

Tính

Áp dụng công thức trên ta có:

Ta lấy 3 cặp hợp lại sẽ xác định được S:

Áp dụng công thức ta được:

b0 = 1847

b1 = 793

b2 = 1359

*Ví dụ 2*:

Cho số nguyên tố p=342853815608923 (Đây là 1 số nguyên tố được lấy trong bảng các số nguyên tố từ cuốn “The Art of Programing” của Knut(Vol 2). Cho n=3, ta có: a1 =53958111706386; a2 =151595058245452;

v1 =111350135012507; v2 =207244959855905; v3 =20545949133543;

Giải sử bí mật là S = 151595058245452

Tính

Áp dụng công thức trên ta có:

Ta lấy 3 cặp hợp lại sẽ xác định được S:

Áp dụng công thức ta được:

b1 = 266921901220910

b2 = 129147516050688

b3 = 289638215946249

## 1.8 Ứng dụng lược đồ chia sẻ bí mật của Lagarange để phân phối khóa.

Như đã trình bày ở Chương 1: để bảo vệ thông tin của mình khi gửi cho người nhận người gửi thường chọn cách mã hóa thông tin bằng nhiều phương pháp khác nhau có thể phân loại ra thành Hệ mật mã khóa bia mật và Hệ mật mã khóa công khai. Trong cả 2 hệ mật mã này thì việc giữ bí mậy khóa mã hóa (cryptographic keys) là đặc biệt quan trọng. Khác với Hệ mã khóa công khai có thể dễ dàng chia sẻ khóa *publish key* giữa bên gửi và nhận thông tin thì Hệ mã khóa bí mật phải đặc biệt bảo mật khóa mã hóa vì nó quyết định cả quá trình mã hóa và giải mã.

Vậy làm thế nào để người gửi chia sẻ khóa mã hóa của hệ mã khóa bí mật cho người nhận có thể giải mã và đọc thông tin đã được gửi?

Phương pháp: Chúng ta sẽ sử dụng lược đồ chia sẻ bí mật để chia sẻ khóa thành nhiều mảnh để truyền đi trong các nút không dáng tin cậy. Khi nhận được các mảnh của chìa khóa người dùng sẽ ghép lại thành khóa hoàn chỉnh và có thể giải mã được thông tin. Vì mỗi mảnh khóa đều không thể suy ra khóa đã chia sẻ nên việc chia sẻ khóa sẽ là an toàn.

***Áp dụng bài toán: Phân phối khóa hệ mã DES***

Hệ mã DES với khóa 64 bit (trong đó chỉ có 56 bit được sử dụng để mã hóa và giải mã, 8 bit còn lại chỉ dùng cho việc kiểm tra) tương đương với 16 số hệ Hexa hay một số nguyên gồm 20 số thập phân. Con số bí mật này không quá lớn đối với bài toán chia sẻ bí mật nên việc tính toán là rất hiệu quả.

Để phân phối khóa của DES cho người nhận, ta sẽ lấy khóa gồm 20 số thập phân chia nhỏ thành chia nhỏ thành các cặp bằng sơ đồ chia sẻ bí mật của Lagrange. Sau đó, người nhận sau khi nhận được bản mã hóa và các mảnh sẽ khôi phục lại khóa để có thể giải mã.

Graphical user interface

Description automatically generated

Hình 1.9 DES và quá trình phân phối khóa

# Kết Quả Nghiên Cứu

## 2.1 Giới thiệu

Thuật toán mã hóa DES và sơ đồ chia sẻ bí mật được ứng dụng rất nhiều trong nhiều lĩnh vực khác nhau để đảm bảo an toàn thông tin khi trao đổi thông tin. Trong phần này ta nghiên cứu một ứng dụng trong việc chuyển đề thi tuyển sinh làm sao để giữ bí mật trước giờ phát đề cho thí sinh.

***- Bài toán:***

Trong một kỳ 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 trên mạng máy tính sao cho đảm bảo về tính bảo mật.

***- Ý tưởng giải quyết bài toán:***

Việc bảo mật đề thi là rất quan trọng, làm sao để vừa giao đề thi cho nơi tổ chức thi kịp thời mã vẫn không bị lộ đề thi trước giờ phát đề?

Để làm như vậy ta cần mã hóa đề thi (ở đây ta dùng thuật toán mã hóa DES với khóa mã hóa K) sau đó chuyển đề thi đã được mã hóa cho nơi tổ chức thi.

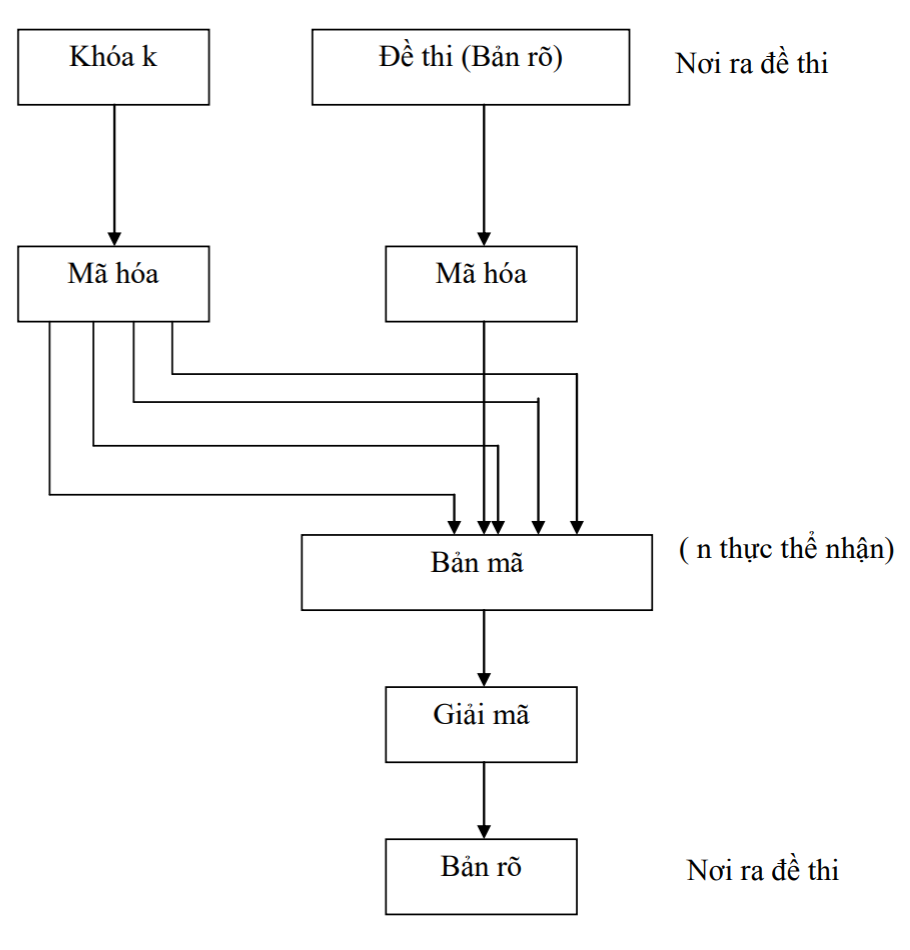
Về việc bảo mật khóa mã hóa: Dùng sơ đồ chia sẻ bí mật chia khóa K thành 2 khóa k1 và k2. Sau đó lần lượt gửi k1 và k2 cho nơi tổ chức. Nơi tổ chức sẽ dùng k1 và k2 để ghép lại khóa K sau đó tiến hành giải mã đề thi để được bản rõ ban đầu.

## Nội dung thuật toán.

### 2.2.1 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.

Ứng dụng lược đồ chia sẻ bí mật của Lagrange để phân phối khóa đã được mô tả ở chương trước.



Hình 2.1 Sơ đồ quy trình bảo mật đề thi gửi từ nơi ra đề đến nơi tổ chức thi

### 2.2.2 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 (vj, f(vj))
  + 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 k1, k2 : k1 + k2 = k. Sau đó gửi k1 cho n thực thể (các địa chỉ thi). Quy định đến đúng giờ G vụ Đào tạo gửi nốt k2 cho n thực thể đó trên cơ sở k1, k2. 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).

***Ví dụ:*** Tại nơi ra đề dùng khóa mã hóa là khóa K(64 bit) = ABCDEF0123456789HEX = 12379813738877118345DEC, lấy số nguyên tố p = 12764787846358441471 (theo https://primes.utm.edu/lists/small/small.html).

Sau khi dùng khóa K để mã hóa đề thi bằng thuật toán DES, Nơi ra đề sẽ thực hiện chia chìa khóa thành 2 phần (n=2) với:

v1 = 151595058245452 (tùy chọn)

v2 = 111350135012507 (tùy chọn)

a1 = 207244959855905 (tùy chọn)

- Quá trình tách khóa:

Tính

Áp dụng công thức trên ta có:

Sau đó cùng gửi đề thi đã mã hóa và *k1* cho các địa chỉ thi. Đến đúng giờ G vụ Đào tạo sẽ gửi nốt *k2* cho các điểm thi.

- Từ 2 cặp hợp lại sẽ xác định được K:

Áp dụng công thức ta được:

Từ đó ta tính được:

=> K = ABCDEF0123456789(HEX)

Khi đã có K, tai địa điểm thi sẽ có thể giải mã để mở được đề thi và tiến hành thi.

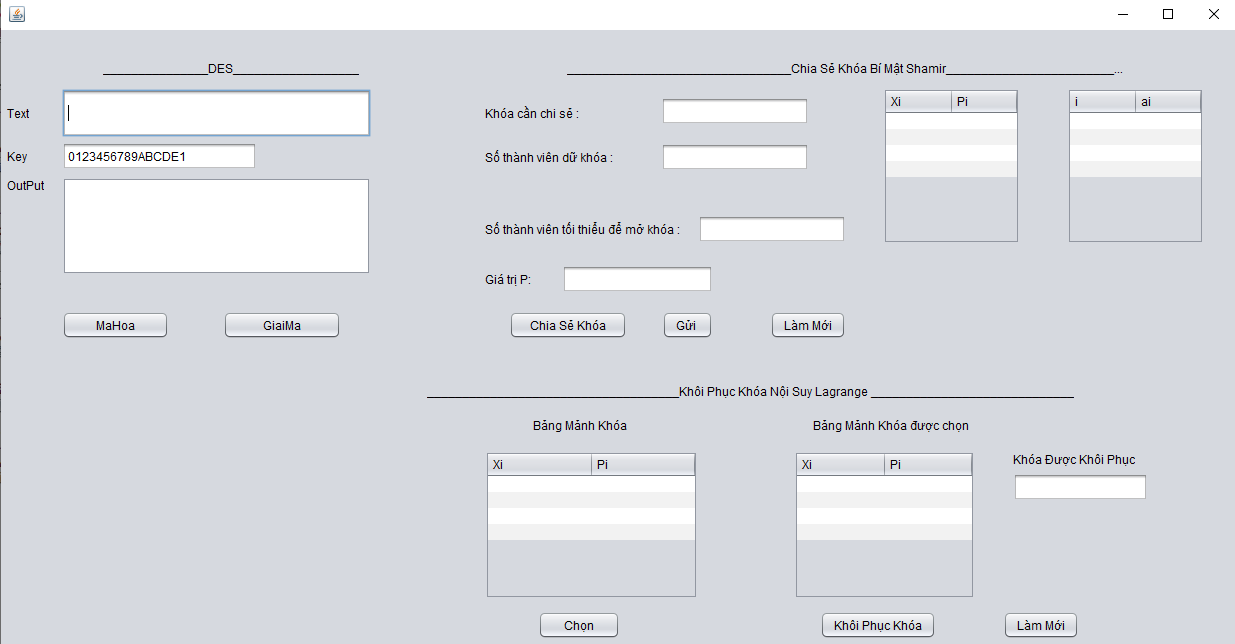
Như vậy, với thật toán DES và lược đồ chia sẻ bí mật có thể đảm bảo an toàn thông tin, không bị lộ đề thi, đảm bảo tính minh bạch của kỳ thi. Đây là một ứng dụng có hiệu quả cao trong thực tiễn.

## 2.3 Thiết kế, cài đặt chương trình demo thuật toán

### 2.3.1 Giao diện chương trình demo

Chương trình gồm các chức năng sau:

* Mã hóa văn bản
* Giải mã văn bản
* Chia sẻ khóa bí mật
* Khôi phục khóa bí mật



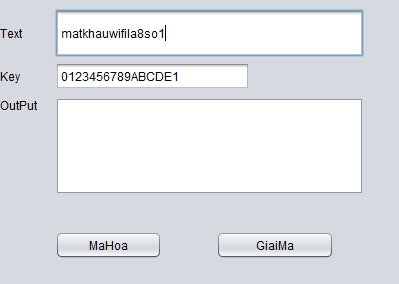
Hình 2.2 Sơ đồ quy trình bảo mật đề thi gửi từ nơi ra đề đến nơi tổ chức thi

***Kết quả cài đặt:***

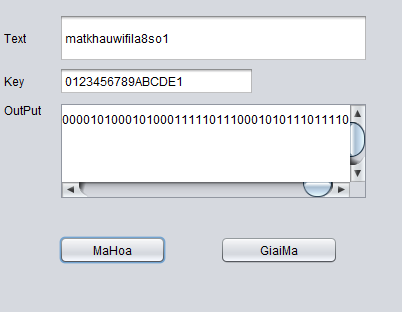
* *Ngôn ngữ cài đặt*: Java
* Môi trường cài đặt: JDK16
* Giao diện: Swing
* Các hệ thống con:
* Mã hóa và giải mã
  + Người sử dụng nhập văn bản cần mã hóa hoặc giải mã.
  + Người sử dụng chọn chức năng mã hóa (hoặc giải mã tương ứng).
  + Nếu người dùng không nhập khóa hoặc khóa sai thì chương trình thông báo nhập khóa.
* Phân phối khóa
  + Người dùng cần nhập đầy đủ các thông tin, các giá trị cần thiết cho việc chia sẻ khóa bí mật.

**Mã Hóa DES:**

* Nhập văn bản muốn mã hóa

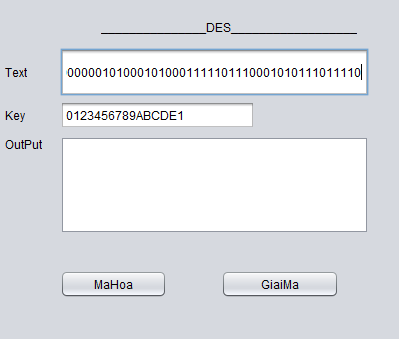
****

* Ấn vào mã hóa hóa ta được chuỗi mã hóa

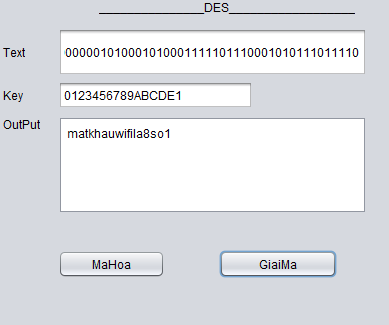
****

**Giải Mã DES:**

* Nhập chuỗi mã hóa

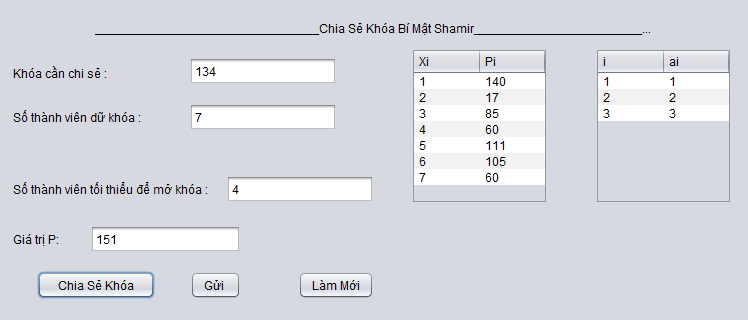
****

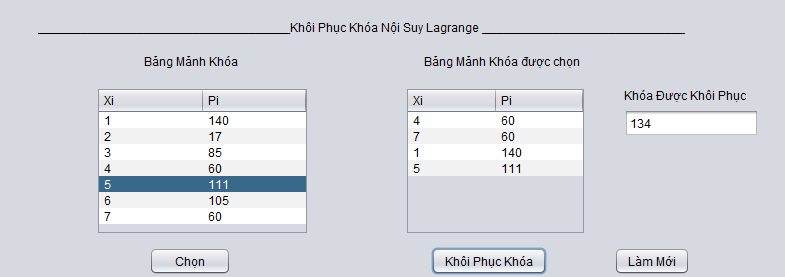
* Ấn giải mã

****

**Chia Sẻ Khóa**

* Nhập dữ liệu gồm khóa , số thành viên giữ khóa, số thành viên tối thiểu để ghép khóa, nhập số nguyên tố P(P>khóa, P> số thành viên giữu khóa, số thành viên giữ khóa >= số thành vien tối thiểu ghép khóa), ấn chia sẻ khóa.

****

* Ấn phim gửi , chọn các mảnh khóa (số mảnh khóa được chọn phải lớn hơn hoặc bằng số thành viên tối thiểu mở khóa) ấn Khôi phục   
  ****

## 2.4 Cài đặt và triển khai

* **Giới thiệu về NetBeans** **:**

NetBeans là một [môi trường phát triển tích hợp](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) cho [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). NetBeans cho phép các ứng dụng được phát triển từ một tập hợp các thành phần phần mềm  mô-đun được gọi là mô-đun.

NetBeans IDE là một “môi trường phát triển tích hợp“ (Integrated Development Environment) kiểu như Visual Studio của Microsoft và được xem là một một bộ ứng dụng “must-download” dành cho các nhà phát triển phần mềm.

NetBeans IDE hỗ trợ nhiều hệ điều hành khác nhau như Windows, Mac, Linux, và Solaris. NetBeans bao gồm một IDE mã nguồn mở và một nền tảng ứng dụng cho phép nhà phát triển nhanh chóng tạo nên các ứng dụng dành cho web, doanh nghiệp, desktop và thiết bị di động bằng các ngôn ngữ lập trình Java, C/C++, JavaScript, Ruby, Groovy, và PHP.

* **Hướng dẫn cài đặt :**
  + Cài đặt NetBeans: Vào trang web tải NetBeans IDE > Chọn **Download** ở phiên bản mới nhất > Chọn nền tảng > Chọn vào đường link sau dòng **We suggest the following mirror site for your download** > Chọn nơi lưu file và chọn **Save** > Nhấp chuột phải vào biểu tượng NetBeans IDE và chọn **Run as administrator** > Chọn ngôn ngữ lập trình muốn cài đặt > Chọn **Next** > Tích vào ô **I accept the terms in the license agreement** và chọn **Next** > Chọn nơi cài đặt > Chọn **Next** > Tích vào ô kiểm **Check for Update** > Chọn **Install** > Chọn **Finish**.
  + Cài đặt JDK : Truy cập vào trang dowloand JDK > Chọn **JDK download** > Chọn **phiên bản JDK phù hợp** với hệ điều hành sử dụng (**Windows**) > Tick vào **I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE** rồi nhấn **Download** > Mở tệp vừa tải xuống rồi nhấn **Run** > Nhấn vào **Next** để bắt đầu cài đặt > Chọn vị trí lưu thư mục rồi nhấn **Next** > Nhấn vào **Close** để hoàn thành.

## Thực Hiện Bài Toán

### Phân công công việc

|  |  |
| --- | --- |
| Tên sinh viên | Tên công việc |
| Hoàng Minh An | - Nghiên cứu bài toán chia sẻ bí mật của Lagrange.  - Ứng dụng lược đồ chia sẻ bí mật của Lagrange để phân phối khóa.  - Cài đặt và chạy chương trình với ngôn ngữ Python. |
| Nguyễn Viết An | - Nghiên cứu bài toán chia sẻ bí mật của Lagrange.  - Ứng dụng lược đồ chia sẻ bí mật của Lagrange để phân phối khóa.  - Cài đặt và chạy chương trình với ngôn ngữ JavaScript. |
| Đinh Đăng Duy Anh | - Giới thiệu, tìm hiểu về mật mã DES.  - Kết luận kiến thúc lĩnh hội và bài học kinh nghiệm.  - Cài đặt và chạy chương trình với ngôn ngữ C++. |
| Doãn Thị Lan Anh | - Giới thiệu, tìm hiểu về mật mã DES.  - Kết luận kiến thúc lĩnh hội và bài học kinh nghiệm.  - Cài đặt và chạy chương trình với ngôn ngữ C#. |
| Nguyễn Hoàng Anh | - Giới thiệu, tìm hiểu về mật mã DES.  - Kết luận kiến thúc lĩnh hội và bài học kinh nghiệm.  - Cài đặt và chạy chương trình với ngôn ngữ Java. |

### Hoàng Minh An

**Nội dung 1: Kỹ Thuật Chia sẻ khóa bí mật :**

Chia sẻ khóa bí mật là phương thức dùng đề chia một bí mật ra làm nhiều phần riêng biệt sau đó phân phối tới những người tham gia).

+ Sơ đồ chia sẻ bí mật của Shamir: Ý tưởng về sơ đồ ngưỡng giới hạn của Shamir dựa trên tính chất: Hai điểm có thể định nghĩa một đường thẳng, 3 điểm định nghĩa được 1 parabol, 4 điểm định nghĩa được một hình lập phương, cứ như thế một cách tổng quát cần n+1 điểm để định nghĩa một đa thức bậc n.

**Nội dung 2: Ứng dụng lược đồ chia sẻ bí mật của Lagrange :**

Phương pháp: Chúng ta sẽ sử dụng lược đồ chia sẻ bí mật để chia sẻ khóa thành nhiều mảnh để truyền đi trong các nút không dáng tin cậy. Khi nhận được các mảnh của chìa khóa người dùng sẽ ghép lại thành khóa hoàn chỉnh và có thể giải mã được thông tin. Vì mỗi mảnh khóa đều không thể suy ra khóa đã chia sẻ nên việc chia sẻ khóa sẽ là an toàn.

### Nguyễn Viết An

**Nội dung 1:**  Kỹ Thuật Chia sẻ khóa bí mật : Chia sẻ khóa bí mật là phương thức dùng đề chia một bí mật ra làm nhiều phần riêng biệt sau đó phân phối tới những người tham gia).

+ Chia sẻ bí mật dựa trên ý tưởng của Lagrange :

Giả sử ta có n thực thể A1,A2,…,An và có 1 người được ủy quyền B biết được toàn bộ khóa bí mật S N.

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

1. B chọn một số nguyên tố P đủ lớn sao cho: với
2. B tiếp theo chọn (2n – 1) số một cách ngẫu nhiên:
3. B xác định một đa thức với các hệ số *a1, …, at-1* trên :
4. Bây giờ B gửi Aj (một cách công khai) cặp coi như mảnh riêng của Aj.

**Nội dung 2: Ứng dụng lược đồ chia sẻ bí mật của Lagrange** : Áp dụng bài toán: Phân phối khóa hệ mã DES

### Đinh Đăng Duy Anh

**Nội dung 1: Tổng quát về mã hóa DES :**

DES (viết tắt của Data Encryption Standard, hay Tiêu chuẩn Mã hóa Dữ liệu) là một phương pháp [mật mã hóa](https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a) được [FIPS](https://vi.wikipedia.org/w/index.php?title=FIPS&action=edit&redlink=1) (Tiêu chuẩn Xử lý Thông tin Liên bang Hoa Kỳ) chọn làm chuẩn chính thức vào năm [1976](https://vi.wikipedia.org/wiki/1976).

**Nội dung 2: Thuật toán**

### Doãn Thị Lan Anh

**Nội dung 1: Ưu điểm , nhược điểm mã hóa DES**

**Nội dung 2: Kỹ Thuật hệ mã hóa DES:**

Trong phương pháp DES, kích thước khối là 64 bit. DES thực hiện mã hóa dữ liệu qua 16 vòng lặp mã hóa, mỗi vòng sử dụng một khóa chu kỳ 48 bit được tạo ra từ khóa ban đầu có độ dài 56 bit. DES sử dụng 8 bảng hằng số S-box để thao tác

**Nội dung 3: Thuật toán.**

### Nguyễn Hoàng Anh

**Nội dung 1:** DES (viết tắt của Data Encryption Standard, hay Tiêu chuẩn Mã hóa Dữ liệu) là một phương pháp [mật mã hóa](https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a" \o "Mã hóa) được [FIPS](https://vi.wikipedia.org/w/index.php?title=FIPS&action=edit&redlink=1) (Tiêu chuẩn Xử lý Thông tin Liên bang Hoa Kỳ) chọn làm chuẩn chính thức vào năm [1976](https://vi.wikipedia.org/wiki/1976).

**Nội dung 2:** **Sơ đồ khối.**

# Phần Kiến Thức Lĩnh Hội Và Bài Học Kinh Nghiệm

## Nội dung đã thực hiện.

Trong quá trình xây dựng chương trình mã hóa và giải mã DES và lược đồ chia se bí mật, các thành viên trong nhóm đã tổng hợp được rất nhiều kiến thức của môn An toàn bảo mật thông tin và một số môn học khác để áp dụng vào xây dựng một hệ thống thực tế. Thông qua quá trình làm việc nhóm các thành viên trong trong nhóm đã đoàn kết với nhau hơn, tinh thần làm việc tập thể được nâng cao hơn, rèn luyện kỹ năng làm việc nhóm. Nhưng do kiến thức và kinh nghiệm còn hạn chế nên nhóm rất mong nhận được sự đóng góp ý kiến của cô giáo và các bạn để hệ thống của nhóm xây dựng được hoàn thiện hơn. Nhóm xin chân thành cảm ơn giảng viên Trần Phương Nhung đã nhiệt tình hướng dẫn nhóm trong suốt quá trình thực hiện đề tài này, được sự nhiệt tình hướng dẫn của cô nhóm đã hoàn thành được đề tài và hiểu hơn về môn học.

- **Những bài học học kinh nghiệm sau khi hoàn thành đề tài nhóm:**

* Nắm rõ kỹ năng xác định vấn đề, kĩ năng phân tích vấn đề và sàng lọc ý kiến.
* Nhóm trưởng cần xác định vai trò của từng thành viên, kiểm soát công việc tối ưu nhất và đưa ra những quyết định đúng đắn.
* Xây dựng tiêu chí đánh giá công việc và thái độ làm việc của từng thành viên trong nhóm. Vừa đảm bảo tính khích lệ động viên, vừa đảm bảo tính ràng buộc về trách nhiệm đối với từng thành viên.
* Biết lắng nghe, tôn trọng ý kiến của từng thành viên.
* Đặt tinh thần trách nhiệm trong công việc thành ưu tiên hàng đầu.
* Phát huy sự gắn kết, tạo sự đồng thuận, phân chia công việc hiệu quả, đưa ra cơ chế giải quyết những mâu thuẫn phát sinh trong nhóm.

## Hướng phát triển.

Do thời gian nghiên cứu có hạn, nên chương trình mới chỉ mô phỏng được các thao tác: mã hóa , giải mã văn bản, chia sẻ khóa và ghép khóa mà chưa thiết kế một cách hoàn chỉnh .Hướng phát triển của đề tài là xây dựng chương trình để có thể mã hóa được cả tệp tin, file hình ảnh. Đồng thời xây dựng một hệ thống chứng thực khóa công khai cho các thành viên, nhằm tránh trường hợp bị người khác giả mạo khóa công khai của người nhận khi thực hiện trao đổi thông tin.

Cuối cùng, với những kết quả đạt được của đề tài nghiên cứu, tuy còn có những hạn chế, nhưng đã giúp chúng em có được khả năng nghiên cứu cơ bản về bảo mật và xác thực thông tin. Từ đó có thể xây dựng các ứng dụng về bảo mật và xác thực thông tin ở những cấp độ an toàn khác nhau.

# Tài Liệu Tham Khảo

1. Phan Đình Diệu (2002). Lý thuyết mật mã và an toàn thông tin.NXB Đại học quốc gia Hà Nội.
2. Giáo trình An toàn và bảo mật thông tin (2008), Đại học Nha Trang, Biên soạn Trần Minh Văn
3. Slide Bài giảng An toàn và Bảo mật thông tin, Ths. Trần Phương Nhung, Giảng viên Đại học Công nghiệp Hà Nội.
4. Secure Communicating Systems: Design, Analysis, and Implementation, Tác giả Michael Huth, Michael R. A. Huth. ([Link sách](https://books.google.com.vn/books?id=n9Mw5PW5BB4C&pg=PA154&lpg=PA154&dq=342853815608923&source=bl&ots=dcmwGNrUhP&sig=ACfU3U0TWGdLge4GDzdcx0eCfeRBMr5moQ&hl=vi&sa=X&ved=2ahUKEwjokPuInYXuAhUGrpQKHTftASAQ6AEwAHoECAEQAg#v=onepage&q=342853815608923&f=false))
5. <https://vi.wikipedia.org/wiki/DES_(m%C3%A3_h%C3%B3a)>
6. <https://en.wikipedia.org/wiki/Secret_sharing>
7. <https://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing>
8. <https://en.wikipedia.org/wiki/Lagrange_polynomial#Finite_fields>