|  |
| --- |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP VIỆT – HUNG**  **KHOA: CÔNG NGHỆ THÔNG TIN**      **ĐỒ ÁN HỌC PHẦN 1**  **ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH MÃ HÓA THÔNG TIN BẰNG NGÔN NGỮ C/C++**  **Giảng viên hướng dẫn: Nguyễn Đình Quyết**  **Họ tên sinh viên: Nguyễn Xuân Huân**  **Mã sinh viên: 2000842**  **Lớp: K4418-CNT2**  **HÀ NỘI, NĂM 2023** |

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP VIỆT-HUNG

**KHOA: CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN HỌC PHẦN 1**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH MÃ HÓA THÔNG TIN BẰNG NGÔN NGỮ C/C++**

**Giáo viên hướng dẫn: Nguyễn Đình Quyết**

**Họ tên sinh viên: Nguyễn Xuân Huân**

**Mã sinh viên : 2000842**

**Lớp:K4418-CNT2**

**HÀ NỘI, NĂM 2023**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP VIỆT-HUNG  KHOA CÔNG NGHỆ THÔNG TIN |  |
| **BỘ MÔN CÔNG NGHỆ THÔNG TIN** |

**ĐỒ ÁN HỌC PHẦN 1**

**Hệ đào tạo**: Cao đẳng, Đại học **Chuyên ngành**: Công nghệ thông tin

**Đề tài số:** QUYET104

**Tên đề tài: Xây dựng một chương trình mã hóa thông tin bằng ngôn ngữ lập trình C/C++.**

**Yêu cầu**

*1. Yêu cầu về kiến thức, kỹ năng:*

- Có kiến thức lý thuyết về mã hóa thông tin;

- Thành thạo ngôn ngữ lập trình C/C++;

*2. Yêu cầu về chương trình:*

- Dùng ngôn ngữ C/C++ để mô phỏng một thuật toán mã hóa thông tin.

*3. Nộp báo cáo:*

- Báo cáo bản cứng theo đúng mẫu của bộ môn;

- Mã nguồn chương trình và file mềm báo cáo.

*4. Đánh giá:*

- Mức đạt: Báo cáo rõ ràng, sáng sủa, đủ nội dung theo yêu cầu ở mục 2;

- Mức khá: Trình bày được một thuật toán mã hóa thông tin và trả lời tốt các câu

hỏi của hội đồng đánh giá;

- Mức tốt: Trình bày thêm được các thuật toán mã hóa khác.

*5. Các yêu cầu khác:*

- Sinh viên lập kế hoạch thực hiện gửi cho giảng viên hướng dẫn;

- Trong qua trình thực hiện mỗi tuần phải liên hệ với giảng viên ít nhất một lần

thông qua email hoặc gặp trực tiếp.

**Thời gian thực hiện:** Theo lịch của bộ môn

**Tài liệu tham khảo:** Giáo trình đồ họa máy tính; Nguồn Internet.

|  |  |
| --- | --- |
| **THÔNG QUA BỘ MÔN** | **GIẢNG VIÊN SOẠN**  **Nguyễn Đình Quyết** |

|  |  |
| --- | --- |
| **TRƯỜNG ĐHCN VIỆT – HUNG**  **KHOA CÔNG NGHỆ THÔNG TIN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  *Mẫu DA02*  *Mẫu DA02*  **Độc lập - Tự do – Hạnh phúc** |

**NHẬN XÉT ĐỒ ÁN HỌC PHẦN 1**

Chuyên ngành : Công Nghệ Thông Tin .

*(Nhận xét của GV hướng dẫn Nhận xét của GV phản biện )*

Họ tên sinh viên: **Nguyễn Xuân Huân**

Tên đề tài: **Xây dựng một chương trình mã hóa thông tin bằng ngôn ngữ lập trình C/C++**

Người nhận xét (họ tên, học hàm, học vị): **Ths. Nguyễn Đình Quyết**

Đơn vị công tác : **Trường Đại Học Công Nghiệp Việt - Hung**

**Ý KIẾN NHẬN XÉT**

1. Về nội dung & đánh giá thực hiện nhiệm vụ nghiên cứu của đề tài:

2. Về phương pháp nghiên cứu, độ tin cậy của các số liệu:

3. Về kết quả của đề tài:

4. Những thiếu sát và vấn đề cần làm rõ (nếu có):

5. Ý kiến kết luận (mức độ đáp ứng yêu cầu đối với ĐA/KL tốt nghiệp)

6. Câu hỏi người nhận xét dành cho học viên (nếu có):

|  |
| --- |
| 🖎 Đánh giá điểm: |

**Kết luận**: ⬜ Đồng ý / ⬜ Không đồng ý cho phép sinh viên được tham dự bảo vệ kết quả trước hội đồng.

*Ngày………tháng……..năm 2021****.***

**NGƯỜI NHẬN XÉT**

# NHẬT KÝ THỰC HIỆN ĐỀ TÀI

|  |  |  |
| --- | --- | --- |
| **Thời gian** | **Công việc** | **Ghi chú** |
| 25/12/2022 | Nhận đề tài từ giáo viên |  |
| 30/12/2022 | Tìm hiểu về mã hóa cổ điển Ceasar, Virgerne, Hill, Play Fair. |  |
| 10/01/2023 | Tìm hiểu về mã hóa khối DES và giải các bài tập về DES thủ công và trên excel |  |
| 04/02/2023 | Tìm hiểu về mã hóa AES và giải các bài tập về AES thủ công và trên excel |  |
| 21/02/2023 | Nghiên cứu về các phương pháp mã hóa đối xứng nâng cao như IDEA, Blowfish, RC5, Cast-128 và xây dựng chương trình mã hóa thông tin bằng ngôn ngữ lập trình C/C++ |  |
| 02/03/2023 | Viết báo cáo đồ án |  |
| 30/03/2023 | Hoàn thiện báo cáo |  |

# LỜI CẢM ƠN

Với sự phát triển nhảy vọt của công nghệ thông tin hiện nay, bảo mật dữ liệu ngày càng giữ vai trò quan trọng trong công nghệ thông tin. Chúng ta sử dụng nhiều thông tin trên Internet, Internet là một tập hợp máy tính nối kết với nhau, là một mạng máy tính toàn cầu mà bất kì ai cũng có thể kết nối bằng máy PC của họ. Với mạng Internet, có thể trao đổi thông tin trong mọi lĩnh vực văn hóa, xã hội, chính trị, kinh tế...

Trong thời đại ngày nay, thời đại truy cập và sử dụng dữ liệu là nhu cầu thường xuyên nhưng cùng với nó là sự xuất hiện các nguy cơ tấn công mạng của người sử dụng thông tin, truy cập thông tin hay thanh toán điện tử. Do đó việc mã hóa, che dấu thông tin để bảo mật thông tin là đặc biệt quan trọng.

Em xin được gửi lời cảm ơn đặc biệt đến thầy giáo **Nguyễn Đình Quyết** đã giúp đỡ và hướng nhiệt tình trong suốt quá trình thực hiện hoàn thành báo cáo và chương trình mô phỏng mã hóa thông tin bằng C/C++.

Tuy nhiên, do thời gian có hạn, kiến thức và kinh nghiệm còn hạn chế nên trong chương trình cũng như bài báo cáo này sẽ không thể tránh khỏi những thiếu sót nhất định. Những ý kiến nhận xét và góp ý của quý thầy cô và các bạn là cơ sở để em học hỏi và hoàn thiện thêm kiến thức tích lũy kinh nghiệm sau này. Em rất mong nhận được sự góp ý nhiệt tình từ quý thầy cô và các bạn !

Em xin chân thành cảm ơn!

**Sinh viên thực hiện**

**Nguyễn Xuân Huân**

MỤC LỤC

[NHẬT KÝ THỰC HIỆN ĐỀ TÀI 5](#_Toc131175059)

[LỜI CẢM ƠN 6](#_Toc131175060)

[MỞ ĐẦU 1](#_Toc131175061)

[CHƯƠNG I : TỔNG QUAN VỀ MẬT MÃ VÀ XÁC THỰC THÔNG TIN 3](#_Toc131175062)

[1.1 Tổng quan về mật mã 3](#_Toc131175063)

[1.1.1 Giới thiệu về mã hóa 3](#_Toc131175064)

[1.1.2 Các thành phần của một hệ thống mã hóa 3](#_Toc131175065)

[1.1.3 Các tiêu chí đặc trưng của một hệ thống mã hóa 4](#_Toc131175066)

[1.1.4 Tấn công một hệ thống mật mã 4](#_Toc131175067)

[1.2 Kỹ thuật mã hóa cổ điển,các kỹ thuật thay thế 5](#_Toc131175068)

[**1.2.1 Mật mã Caesar Cipher** 5](#_Toc131175069)

[**1.2.2 Mật mã Playfair** 6](#_Toc131175070)

[**1.2.3 Mật mã Hill** 6](#_Toc131175071)

[**1.2.4 Mật mã Vigenère** 6](#_Toc131175072)

[1.3 Áp dụng giải thuật Euclid mở rộng tìm số nghịch đảo trong vành \mathbb{Z}_m 7](#_Toc131175073)

[1.3.1 Số nghịch đảo trong vành \mathbb{Z}_m 7](#_Toc131175074)

[1.3.2 Giải thuật 8](#_Toc131175075)

[1.3.3 Các kỹ thuật chuyển vị 8](#_Toc131175076)

[1.3.4 Các kỹ thuật giấu tin trong một tin khác (Steganography) 9](#_Toc131175077)

[1.4 Mã hóa đối xứng 10](#_Toc131175078)

[1.4.1 Giới thiệu chung 10](#_Toc131175079)

[1.4.2 Cấu trúc mã khối cơ bản 10](#_Toc131175080)

[1.5 Thuật toán mã hóa AES 14](#_Toc131175081)

[1.5.1 Giới thiệu 14](#_Toc131175082)

[1.5.2 Mã hóa 15](#_Toc131175083)

[CHƯƠNG II: MÔ PHỎNG BÀI TOÁN MÃ HÓA DES 25](#_Toc131175084)

[2.1 Thuật toán mã hóa DES 25](#_Toc131175085)

[2.1.1 Giới thiệu 25](#_Toc131175086)

[2.1.2 Dịch chuyển số học 26](#_Toc131175087)

[2.1.3 Dịch chuyển luận lý 27](#_Toc131175088)

[2.1.4 Mã hóa 28](#_Toc131175089)

[2.2 Bài toán 36](#_Toc131175090)

[2.3 Mô phỏng mã hóa DES 36](#_Toc131175091)

[KẾT LUẬN 37](#_Toc131175092)

[TÀI LIỆU THAM KHẢO 38](#_Toc131175093)

# MỞ ĐẦU

**I. Đặt vấn đề**

Với sự bùng nổ của Công nghệ thông tin vào cuối thế kỷ XX đầu thế kỷ XXI, nhân loại đang bước vào một thời đại mới: Thời đại của nền kinh tế thông tin toàn cầu hóa. Mọi hoạt động xã hội, chính trị, kinh tế trong thời đại mới hiện nay xét cho cùng, thực chất đều là những hoạt động thu thập, xử lý, lưu trữ và trao đổi thông tin. Trong bối cảnh đó Bảo mật thông tin luôn là mối quan tâm hàng đầu trong mọi giao dịch xã hội, đặc biệt là giao dịch điện tử trên môi trường Internet, một môi trường mở, môi trường không được tin cậy.

Để bảo vệ bí mật cho thông tin của mình được gửi đi trong một môi trường “mở” tức là môi trường có thể có nhiều tác nhân tiếp cận ngoài hai đối tác trao đổi thông tin, người ta phải dùng mật mã tức là dùng những phương pháp biến đổi làm cho nguyên bản gốc của thông tin (plaintext) ở dạng thông thường ai cũng có thể hiểu được biến thành một dạng bí mật (ciphertext) mà chỉ có những người nắm được quy luật mới có thể biến đổi ngược lại thành dạng nguyên gốc ban đầu để đọc.. Để mã hóa dữ liệu chúng ta sử dụng các phương pháp mã hóa cổ điển như Ceasar, Vigenere, Hill, PlayFair….. hay mã hóa khối Des, Aes, Cast 128, RC5, Blowfish…..

Xây dựng một chương trình mô phỏng mã hóa thông tin bằng ngôn ngữ lập trình C/C++ là chương trình sẽ mô tả hoạt động mã hóa thông tin cơ bản DES sau khi có bản rõ, khóa thì chương trình xuất đầu ra là bản mã dựa trên bảng hoán vị IP, PC-1, PC-2, IP-1, S1, S2, E, P…….

**II. Mục tiêu nghiên cứu của đề tài**

Mục tiêu nghiên cứu chính của đề tài tập chung vào các kiến thức sau:

* Nghiên cứu về các phương pháp mã hóa và giải mã thông tin. Các kỹ thuật mã hóa dùng khóa đối xứng và kỹ thuật mã hóa dùng khóa bất đối xứng.
* Nghiên cứu các thành phần trong mã hóa thông tin như bản rõ (Plaintext) bản mã(Ciphertext) khóa(Key) và các giải thuật Euclid, giải thuật Euclid mở rộng.
* Tìm hiểu về các vòng và cơ chế của mã hóa khối DES, AES.

**III. Nhiệm vụ của đề tài**

* Nghiên cứu các phương pháp mã hóa thông tin và các thành phần trong mã hóa thông tin. Sử dụng các bảng trong mã hóa DES để giải các bài tập như bảng hoán vị IP, bảng hoán vị E,P, bảng thay thế S1….S8, bảng hoán vị khóa PC-1, PC-2…
* Sử dụng ngôn ngữ lập trình C/C++ để mô phỏng quy trình mã hóa thông tin DES.

**IV. Kết quả đạt được của đề tài**

* Sau quá trình nghiên cứu và tìm hiểu các quy trình mã hóa thông tin của mã hóa khối DES theo 16 vòng lặp, em đã giải các bài tập về mã hóa khối DES theo bảng IP, IP-1, S, E, P, PC-1, PC-2.
* Sử dụng ngôn ngữ lập trình C/C++ để xây dựng chương trình mô phỏng mã hóa DES.

**V. Bố cục của đề tài**

Đề tài gồm 2 chương như sau:

**Chương 1** : TỔNG QUAN VỀ MẬT MÃ VÀ XÁC THỰC THÔNG TIN

**Chương 2** : MÔ PHỎNG BÀI TOÁN MÃ HÓA DES

CHƯƠNG I :  
TỔNG QUAN VỀ MẬT MÃ VÀ XÁC THỰC THÔNG TIN

## Tổng quan về mật mã

### Giới thiệu về mã hóa

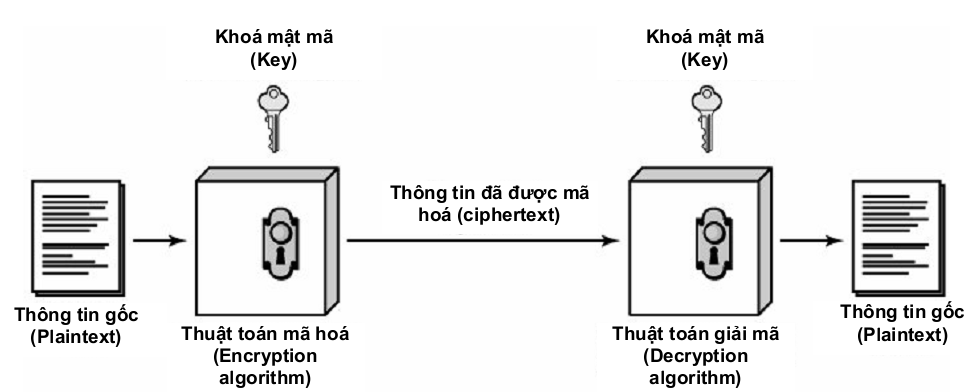
Mật mã (Encryption) là kỹ thuật cơ sở quan trọng trong bảo mật thông tin. Biến đổi thông tin gốc thành dạng thông tin bí mật mà chỉ có những thực thể tham gia xử lý thông tin một cách hợp lệ mới hiểu được.

* Quá trình mã hoá (encryption): Quá trình chuyển thông tin gốc thành thông tin mật theo một thuật toán nào đó
* Quá trình giải mã (decryption): Quá trình biến đổi thông tin mật về dạng thông tin gốc ban đầu

Kỹ thuật mã hoá được chia thành hai loại:

* Mã hoá dùng khoá đối xứng (symmetric key encryption)
* Mã hoá dùng khoá bất đối xứng (asymmetric key encryption).

### Các thành phần của một hệ thống mã hóa



* Plaintext: là thông tin gốc cần truyền đi giữa các hệ thống thông tin
* Encryption algorithm:thuật toán mã hóa
* Key: khóa mật mã, gọi tắt là khóa. Đây là thông tin cộng thêm mà thuật tóan mã hóa sử dụng để trộn với thông tin gốc tạo thành thông tin mật.
* Ciphertext:thông tin đã mã hóa (thông tin mật). Đây là kết quả của thuật toán mã hóa.
* Decryption algorithm:Thuật tóan giải mã. Đầu vào của thuật tóan này là thông tin đã mã hóa (ciphertext) cùng với khóa mật mã. Đầu ra của thuật toán là thông tin gốc (plaintext) ban đầu.

### Các tiêu chí đặc trưng của một hệ thống mã hóa

* Phương pháp mã (operation).
* Thay thế (substitution): thay thế 1 đơn vị mã (từ, ký tự, bit, …) bởi 1 đơn vị mã khác
* Chuyển vị (transposition): Thay đổi vị trí mã
* Số khóa sử dụng (number of keys)
* Nếu phía mã hóa và phía giải mã dung chung một khóa 🡪 (symmetric key) mã đối xứng còn gọi là mã một khóa (single-key), mã khóa bí mật (secret key) hoặc mã quy ước (conventional cryptosystem)
* Nếu phía mã hóa và phía giải mã dùng 2 khóa khác nhau 🡪 mã bất đối xứng (asymmetric key), mã hai khóa (two key) họăc mã khóa công khai (public key)
* Cách xử lý thông tin gốc (mode of cipher)
* Xử lý liên tục theo từng phần tử 🡪 hệ thống mã dòng (stream cipher).
* Xử lý theo từng khối 🡪 ta có hệ thống mã khối (block cipher).

### Tấn công một hệ thống mật mã

**Khái quát**

Tấn công (attack), bẻ khoá (crack) một hệ thống mật mã là quá trình thực hiện việc giải mã thông tin mật một cách trái phép

Để giải mã cần: thông tin mật (ciphertext), khóa (secret key) và thuật toán giải mã (decryption algorithm). Nếu không có thì thực hiện theo các phương pháp:

Phương pháp phân tích mã (cryptanalysis) 🡪 tìm ra khóa (thuật toán không được coi là thông tin mật)

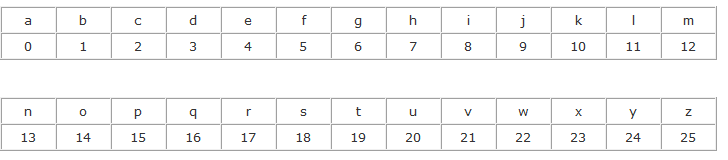
Phương pháp thử tuần tự (brute-force)

**Các kiểu tấn công mã**

* Ciphertext only (Encryption algorithm + Ciphertext): Kẻ tấn công E chỉ biết giải thuật, một số bản mã. E tiến hành phân tích mã với một số thông tin có thể thu thập được như ngôn ngữ mà bản rõ sử dụng.
* Known plaintext (Encryption algorithm + Ciphertext + plaintext-ciphertext pairs): Kẻ tấn công E đã biết giải thuật, một số cặp X-Y (bản rõ - bản mật tương ứng) nào đó. E tiến hành phân tích mã cho bản mã cần tấn công. Rõ ràng mô hình tấn công này làm mạnh hơn so với tấn công chỉ qua bản mã. E có thể dùng phép thử loại trừ để vét cạn không gian khóa (exshautive key search) và tìm ra khóa K sao cho Enc(K,X)=Y
* Chosen plaintext (Encryption algorithm + Ciphertext + chosen plaintext-ciphertext pairs)
* Chosen ciphertext (Encryption algorithm + Ciphertext + chosen ciphertext-plaintext pairs)
* Chosen text ((Encryption algorithm + Ciphertext + chosen ciphertext-plaintext pairs+chosen plaintext-ciphertext pairs)

## Kỹ thuật mã hóa cổ điển,các kỹ thuật thay thế

1. **Mật mã Caesar Cipher**

* Là mật mã đơn ký tự
* Thay thế mỗi ký tự bằng một ký tự khác cách nhau một khoảng cách nhất định trong bảng chữ cái
* 
* C = E(P, k) = (P + k) mod 26 (mã hóa)
* P = D(C, k) = (C - k) mod 26 (Giải mã, chú ý số âm)
* Plaintext = “hen gap o baly”, k =5 (Key)
* Ciphertext = “mjs lfv t gfqd”
* Mật mã Caesar Cipher cũng có thể được thực hiện dưới dạng sau:
* plaintext: meet me after the toga party
* ciphertext: phhw ph diwhu wkh wrjd sduwb
* Key: a b c d e f g h i j k l m n o pq r s t u v w x y z

=> d e f g h i j k l mn o p q r s t u vw x y z a b c

Nhận xét:

* Số khóa để thử rất ít 🡪 Có thể bị tấn công theo kiểu vét cạn (brute-force) bằng cách thử hết tất cả 25 khóa
* Tần số xuất hiện của các chữ cái trong các ngôn ngữ là cố định 🡪 Dựa vào tần suất xuất hiện ký tự để dò khóa
* Giải thuật mã hóa và giải mã được biết trước
* Ngôn ngữ của bản rõ được biết trước và dễ dàng nhận ra

Cải tiến:

* Sử dụng nhiều khóa (nhiều lần mã)
* Bản rõ có thể được nén lại (Huffman) để cho người đọc khó nhận ra ngôn ngữ sử dụng

1. **Mật mã Playfair**

* Xây dựng ma trận khóa

Cho trước một từ làm khoá, với điều kiện trong từ khoá đó không có chữ cái nào bị lặp. Ta lập ma trận Playfair cỡ 5 x 5 dựa trên từ khoá đã cho và gồm các chữ trên bảng chữ cái, được sắp xếp theo thứ tự như sau:

* Trước hết viết các chữ của từ khoá vào các hàng của ma trận bắt từ hàng thứ nhất.
* Nếu ma trận còn trống, viết các chữ khác trên bảng chữ cái chưa được sử dụng vào các ô còn lại. Có thể viết theo một trình tự qui ước trước, chẳng hạn từ đầu bảng chữ cái cho đến cuối.

Vì có 26 chữ cái tiếng Anh, nên thiếu một ô. Thông thường ta dồn hai chữ nào đó vào một ô chung, chẳng hạn I và J. Ví dụ

1. **Mật mã Hill**

Khóa của hệ mã là một ma trận vuông K kích thước MxM gồm các phần tử là các số nguyên thuộc bảng chữ cái ZN trong đó N là số phần tử của bảng chữ cái. K phải là một ma trận không suy biến trên ZN tức là tồn ma trận nghịch đảo của K trên ZN. Các ký tự của bảng chữ cái cũng được đánh số từ 0 tới N-1.

Để mã hóa người ta cũng chia bản rõ đó thành các xâu có độ dài M, chuyển các xâu này thành số thứ tự của các chữ cái trong bảng chữ cái dưới dạng một vectơ hàng M chiều và tiến hành mã hóa, giải mã theo công thức sau:

Mã hóa: C = P \* K

Giải mã: P = C \* K-1

1. **Mật mã Vigenère**

Hệ mã này được đặt theo tên của một nhà mật mã học người Pháp Blaise de Vigenère (1523-1596). Các thông điệp được tạo thành từ một bảng chữ cái ZN giống trong hệ mã Caesar, các chữ cái được đanh số từ 0 tới N-1 trong đó N là số phần tử của bảng chữ cái.

Không gian khóa *K* được xác định như sau:

Với mỗi số nguyên dương M, khóa có độ dài M là một xâu ký tự có độ dài M, K = k1k2…kM.

Để mã hóa một bản rõ P người ta chia P thành các đoạn độ dài M và chuyển thành số thứ tự tương ứng của chúng trong bảng chữ cái, chẳng hạn X = x1x2…xM. Khi đó việc mã hóa và giải mã được thực hiện như sau:

* EK(X) = (x1 + k1, x2 + k2, …, xM + kM) mod N
* DK(Y) = (y1 - k1, y2 - k2, …, yM - kM) mod N

**Ví dụ** :

Xét ZNlà bảng chữ cái tiếng Anh, ta có N = 26 giả sử khóa có độ dài 6 và K = “CIPHER”, bản rõ P = “THIS CRYPTOSYSTEM IS NOT SECURE”. Bỏ qua các dấu cách ta có :

*K = 2 8 15 7 4 17*

*P = 19 7 8 18 2 17 | 24 15 19 14 18 24 | 18 19 4 12 8 18 | 13 14 19 18 4 2 | 20 17 4.*

Quá trình mã hóa thực hiện như sau :

*P = 19 7 8 18 2 17 | 24 15 19 14 18 24 | 18 19 4 12 8 18 | 13 14 19 18 4 2 | 20 17 4*

*K = 2 8 15 7 4 17 | 2 8 15 7 4 17 | 2 8 15 7 4 17 | 2 8 15 7 4 17 | 2 8 15*

*C = 21 15 23 25 6 8 | 0 23 8 21 22 15 | 20 1 19 19 12 9 | 15 22 8 25 8 19 | 22 25 19*

*Mã là C = “VPXZGI AXIVWP UBTTMJ PWIZIT WZT”.*

Về thực chất hệ mã này là kết hợp của nhiều mã Caesar, trong hệ mã Caesar chúng ta thay thế từng ký tự đơn lẻ thì trong hệ mã Vigenere này thay thế từng bộ M ký tự liên tiếp.

## Áp dụng giải thuật Euclid mở rộng tìm số nghịch đảo trong vành \mathbb{Z}_m

### Số nghịch đảo trong vành \mathbb{Z}_m

Trong lý thuyết số, vành \mathbb{Z}_m được định nghĩa là vành thương của \mathbb{Z} với quan hệ đồng dư theo môđun m (là quan hệ tương đương) mà các phần tử của nó là các lớp đồng dư theo mođun m (m là số nguyên dương lớn hơn 1). Ta cũng có thể xét \mathbb{Z}_m chỉ với các đại diện của nó. Khi đó

\mathbb{Z}_m = \left \{ 0,1,...,m-1 \right \}

Phép cộng và nhân trong \mathbb{Z}m là phép toán thông thường được rút gọn theo mođun m:

a+b=(a+b) \quad mod \quad m  
a*b=(a*b) \quad mod \quad m

Phần tử a của \mathbb{Z}_m được gọi là khả nghịch trong \mathbb{Z}_m hay khả nghịch theo mođun m nếu tồn tại phần tử a' trong \mathbb{Z}_m sao cho a\*a'=1 trong \mathbb{Z}_m hay a*a' \equiv 1\pmod m. Khi đó a' được gọi là nghịch đảo modulo m của a. Trong lý thuyết số đã chứng minh rằng, số a là khả nghịch theo mođun m khi và chỉ khi ƯCLN của a và m bằng 1.  
 Khi đó tồn tại các số nguyên x, y sao cho

m \* x + a \* y = 1

Đẳng thức này lại chỉ ra x là nghịch đảo của a theo mođun m. Do đó có thể tìm được phần tử nghịch đảo của a theo mođun m nhờ thuật toán Euclid mở rộng khi chia m cho a.

### Giải thuật

Giải thuật sau chỉ thực hiện với các số nguyên m>a>0, biểu diễn bằng giã mã:

**Procedure Euclid\_Extended (a,m)**  
int, y0=0,y1:=1;  
  
**While** a>0 **do** {  
 r:= m mod a   
 if r=0 then Break   
 q:= m div a  
 y:= y0-y1\*q  
 m:=a  
 a:=r  
 y0:=y1   
 y1:=y  
 }  
 **If** a>1 **Then Return** "A không khả nghịch theo mođun m"   
 **else Return** " Nghịch dảo mođun m của a là y"

### Các kỹ thuật chuyển vị

Các ký tự trong bản rõ chỉ được thay đổi vị trí để tạo ra bản mã.

* Bản rõ: “GapNhauTaiCongTruong”
* Khóa: 4 3 1 2 5 6 7 (khóa là thứ tự các cột). m=7
* Mã hóa:
* Ghi vào theo dòng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | 3 | 1 | 2 | 5 | 6 | 7 |
| G | a | p | N | h | a | u |
| T | a | i | C | o | n | g |
| T | r | u | o | n | g |  |

* Hoán vị cột

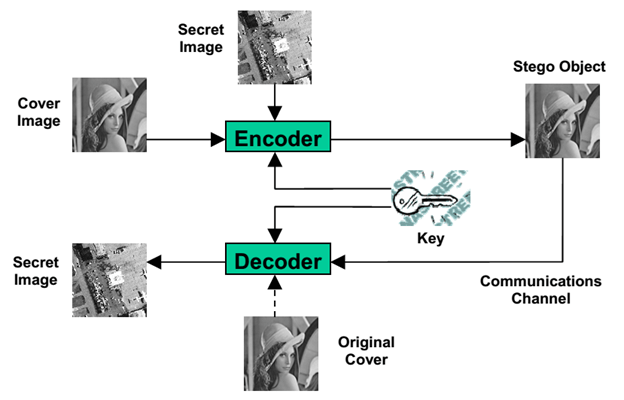
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| p | N | a | G | h | a | u |
| i | C | a | T | o | n | g |
| u | o | r | T | n | g |  |

* Đọc ra theo thứ tự cột
* Bản mã: piuNCoaarGTThonangug.
* Giải mã:
* Chia độ dài mã cho m=7 🡪 21 : 7 =3. Ghi theo cột với số ký tự là 3
* Hoán vị cột theo khóa
* Đọc ra theo hàng

### Các kỹ thuật giấu tin trong một tin khác (Steganography)

**Ví dụ :**

Giấu tin trong ảnh :

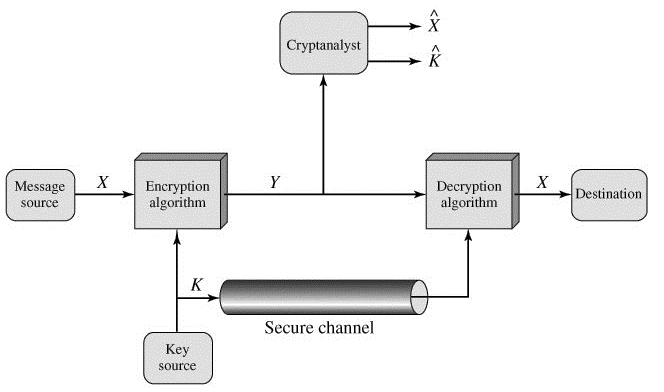


## Mã hóa đối xứng

### Giới thiệu chung

Kỹ thuật mật mã đối xứng sử dụng một khóa duy nhất cho cả quá trình mã hóa và giải mã thông tin. Bằng một cách an toàn nào đó, khóa chung này phải được trao đổi thống nhất giữa bên gửi và bên nhận, được giữ bí mật trong suốt thời gian sử dụng.

Còn được gọi là mật mã quy ước (conventional encryption) hoặc mật mã dùng khóa bí mật (secret key encryption) hoặc “Mã hóa truyền thống”.



**Secure chanel (Kênh an toàn)**

Được dùng để trao đổi khóa bí mật. Việc trao đổi có thể thực hiện bằng các phương pháp.

* A chọn ra một khóa bí mật và chuyển trực tiếp cho B (chuyển bằng phương tiện vật lý như ghi lên đĩa, nói trực tiếp, ghi ra giấy, …)
* Một thực thể thứ 3 chọn ra khóa bí mật và thông báo khóa này cho cả A và B (bằng phương tiện vật lý như trên)

Nếu A và B trước đó đã dùng một khóa nào đó để thông tin với nhau, thì một trong hai thực thể sẽ tiếp tục dùng khóa cũ để gởi thông báo về khóa mới cho thực thể kia

Nếu A và B có các kết nối an tòan đến một thực thể thứ 3 là C, thì C có thể gửi thông báo về khóa cho cả hai thực thể A và B thông qua kết nối an toàn này.

### Cấu trúc mã khối cơ bản

1. **Mã hóa khối**

Các hệ mã cổ điển đều có đặc điểm chung là từng ký tự của bản rõ được mã hoá tách biệt. Điều này làm cho việc phá mã trở nên dễ dàng hơn. Chính vì vậy, người ta dùng một kiểu mật mã khác, trong đó từng khối ký tự của bản rõ được mã hoá cùng một lúc như là một đơn vị mã hoá đồng nhất.

So với mã hóa luồng

* Mã hóa khối xử lý thông báo theo từng khối
* Mã hóa luồng xử lý thông báo 1 bit hoặc 1 byte mỗi lần

Giống như thay thế các ký tự rất lớn (≥ 64 bit)

* Bảng mã hóa gồm 2n đầu vào (n là độ dài khối)
* Mỗi khối đầu vào ứng với một khối mã hóa duy nhất: Tính thuận nghịch
* Độ dài khóa là n x 2n bit quá lớn

Xây dựng từ các khối nhỏ hơn

Hầu hết các hệ mã hóa khối đối xứng dựa trên cấu trúc hệ mã hóa Feistel

Điều kiện để mã hoá khối an toàn :

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

**Mạng S-P**

Mạng thay thế (S) - hoán vị (P) đề xuất bởi Claude Shannon vào năm 1949

Là cơ sở của các hệ mã hóa khối hiện đại

Dựa trên 2 phép mã hóa cổ điển

Phép thay thế : Hộp S(vào là 2, ra là 6)

0

1

2

3

4

5

6

7

Đầu vào

3 bit

0

1

0

0

1

2

3

4

5

6

7

1

1

0

Đầu ra

3 bit

* Phép hoán vị : Hộp P

Đầu vào

4 bit

1

1

0

1

1

0

1

1

1

1

0

1

1

0

1

1

Đầu ra

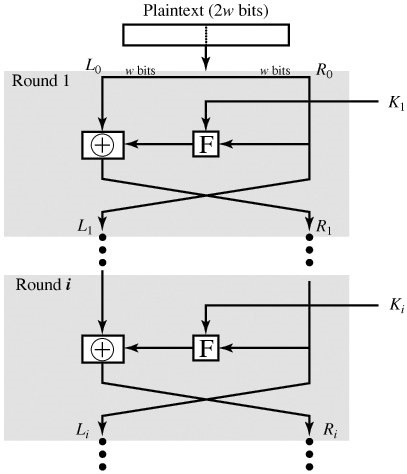
4 bit

Đan xen các chức năng

* Khuếch tán : Hộp P (kết hợp với hộp S): Phát tỏa cấu trúc thống kê của nguyên bản khắp bản mã
* Gây lẫn: Hộp S: Làm phức tạp hóa mối quan hệ giữa bản mã và khóa

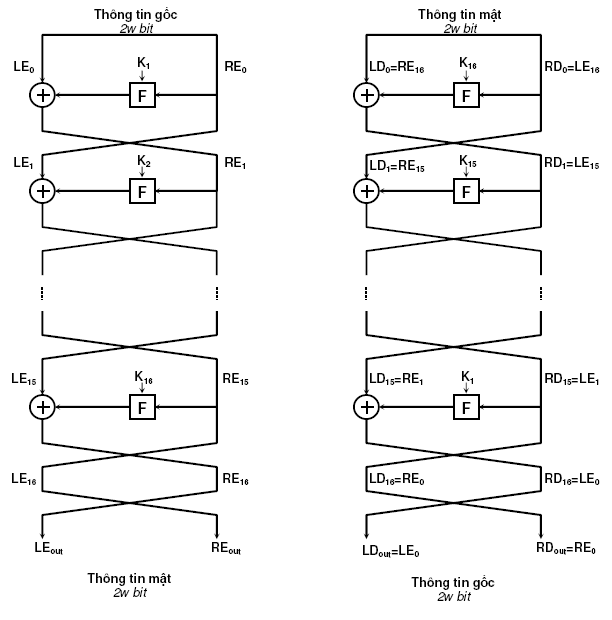
1. **Giới thiệu mật mã khối - Feistel**

* IBM được xây dựng năm 1973
* Làm cơ sở cho nhiều thuật toán mã hóa đối xứng hiện đại. VD mã hóa DES
* Dựa trên các thao tác thay thế (substitution) và hoán vị (permutation) nhiều lần trên dữ liệu gốc
* Claude Shannon đề xuất năm 1949. Mục đích: triệt tiêu tính thống kê của plaintext và key trong ciphertext.
* Diffusion (khuếch tán): plaintext và ciphertext
  + Chuyển vị, thực hiện nhiều vòng
* Confusion (xáo trộn): ciphertext và key
  + Thay thế
* Cấu trúc Feistel thỏa mãn cả 2 thuộc tính này.

1. **Mô tả**

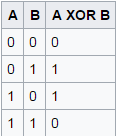
**Mã hóa**

* Thông tin gốc được chia thành từng khối có kích thước 2w bit, mỗi khối lại được chia thành 2 phần L và R
* Từng phần L và R được đưa qua n vòng xử lý giống nhau.
* Thao tác xử lý tại mỗi vòng: Hoán vị L và R, đưa R vào hàm F cùng với khóa Ki, XOR kết quả hàm F với L.
* Hoán vị lần cuối cùng
* Với mỗi thuật toán mã hóa, hàm F sẽ được xây dựng khác nhau.

Giả sử hệ thống dùng 16 vòng mã hóa, xét vòng cuối cùng của quá trình mã hóa:

* LE16 = RE15
* RE16 = LE15 ⊕ F(RE15, K16)

Trong đó: ⊕ là phép XOR



##### Giải mã

* Đưa thông tin mật vào khối giải mã:
* LD0 = LEout = RE16
* RD0 = REout = LE16
* Xét vòng thứ nhất của giải mã:
* LD1 = RD0
* RD1 = LD0 ⊕ F(RD0, K16)

## Thuật toán mã hóa AES

### Giới thiệu

AES được phát triển từ cuối những năm 90s để thay thế chuẩn mã hóa trước đó là Data Encryption Standard (DES) do IBM tạo ra đầu những năm 70s. AES được chính phủ Mỹ dùng trong năm 1977 nhưng sau đó có nhiều lỗ hổng dễ bị tấn công (brute force, phân tích mật mã khác biệt/tuyến tính) do dựa trên thuật toán 56 bit, nên không còn hữu ích nữa khi vi xử lý máy tính ngày càng mạnh hơn.

Vào năm 1998, DES trở thành 3DES hay còn gọi là Triple DES, dùng thuật toán DES để truyền thông điệp 3 lần liên tiếp với 3 khóa mã hóa khác nhau. 3DES khiến dữ liệu an toàn hơn trước kiểu tấn công brute force thời đó. Các thuật toán được đề xuất thay thế DES, bắt đầu quy trình 5 năm của chính phủ Mỹ. AES được hai nhà mật mã học là Vincent Rijmen và Joan Daemen đề xuất, sau được gọi là “đơn Rijindael”.

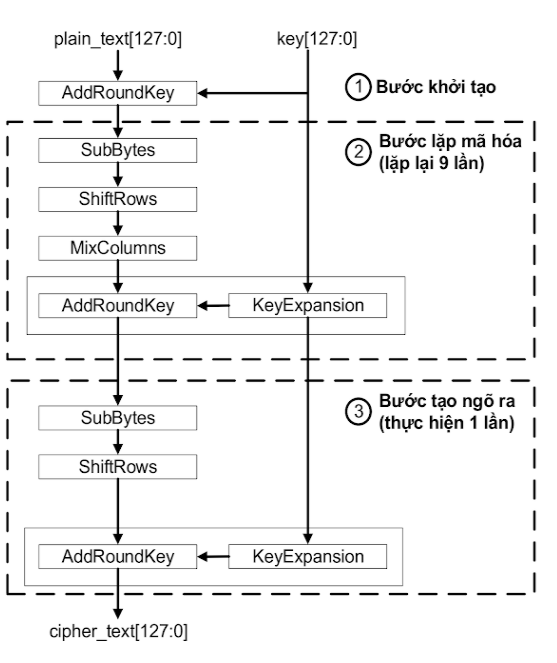
AES là chuẩn mở vì khi đó chuẩn thực sự cũng chưa được xác định. Trong quá trình thiết kế, nó cũng nhận bình luận, góp ý. Nó được Viện tiêu chuẩn và kỹ thuật quốc gia Hoa Kỳ phát triển với mục tiêu dễ dùng cho cả phần cứng và phần mềm. Một số thay đổi về khóa và khối được thực hiện để tăng tính an toàn. NSA cũng tham gia xem xét 15 bản đề xuất. Tới tháng 8/1999 chỉ còn 5 thuật toán (Rijndael, Serpent, RC6, Twofish và MARS). Các “ứng viên” được phân tích thêm về độ bảo mật, tính dễ sử dụng, bản quyền, tốc độ, độ chính xác khi mã hóa và giải mã.

AES là kiểu mã hóa khối, mỗi khối kích thước 128 bit. Khóa đối xứng với 3 kích thước là 128, 192 và 256 bit, trong đó 2 kích thước sau được chính phủ Mỹ dùng cho các tài liệu mật cấp cao, được gọi là “Top Secret”. Rijndael ban đầu được phép thêm khối và tăng độ dài khóa nhưng chuẩn sau này bị bỏ, giữ chuẩn kích thước như đã nói ở trên. AES là chuẩn mã hóa duy nhất được phát hành rộng rãi được NSA chấp thuận dùng để bảo vệ thông tin chính phủ ở mức cao cấp nhất.

### Mã hóa

Mã hóa AES được thực hiện thông qua 5 chức năng chính là AddRoundKey, SubBytes, ShiftRows, MixColumns và KeyExpansion. Năm chức năng này được sắp xếp để thực hiện ba bước cơ bản.

* Bước 1. Bước khởi tạo dữ liệu cần được mã hóa plain\_text kết hợp với key bằng chức năng AddRoundKey
* Bước 2. Bước lặp mã hóa: kết quả bước 1 được sử dụng để thực hiện tuần tự các chức năng SubBytes, ShiftRows, MixColumns và AddRoundKey. Bước này được lặp lại 9 lần.
* Bước 3. Bước tạo ngõ ra: Sau 9 lần lặp ở bước 2, kết quả được sử dụng để thực hiện tuần tự các chức năng SubBytes, ShiftRows và AddRoundKey để tạo ngõ ra cipher\_text

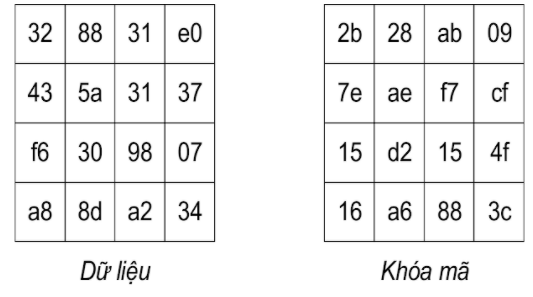


*Quá trình mã hóa AES*

Quá trình mã hóa AES-128 sẽ được giải thích trên một ví dụ cụ thể. Giả sử chuỗi dữ liệu cần mã hóa plain\_text và khóa mã keycó giá trị như sau:

* plain\_text[127:0] = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
* key[127:0] = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

Dữ liệu và khóa mã được sắp xếp dưới dạng ma trận với mỗi phần tử là một byte.



Trong quá trình mã hóa, ma trận dữ liệu ban đầu sẽ bị biến đổi bởi các chức năng AddRoundKey, SubBytes, ShiftRows hoặc MixColumns để tạo ra các dữ liệu trung gian gọi là ma trận trạng thái. Ma trận khóa mã sẽ bị biến đổi bởi chức năng KeyExpansion để tạo ra các khóa mã trung gian gọi là khóa vòng.

**Chức năng AddRoundKey**

Chức năng AddRoundKey thực hiện ở:

* Bước khởi tạo: XOR khóa mã với ma trận dữ liệu
* Bước lặp mã hóa và bước tạo ngõ ra: XOR khóa vòng (round key) với ma trận trạng thái.

|  |
| --- |
| [https://1.bp.blogspot.com/-vRjv-zOL9L8/XY9snv93zdI/AAAAAAAAJ-I/AR15FMTy9KomK5jB7HyBI85HuRTf3IsMACKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-vRjv-zOL9L8/XY9snv93zdI/AAAAAAAAJ-I/AR15FMTy9KomK5jB7HyBI85HuRTf3IsMACKgBGAsYHg/s1600/image.png) |
| *Chức năng AddRoundKey cho bước khởi tạo* |

Đối với bước lặp mã hóa và bước tạo ngõ ra, vị trí "khóa mã" là các "khóa vòng" còn dữ liệu là của lần tính trước đó.

**Chức năng SubBytes**

Chức năng SubBytes là thực hiện thay thế từng byte của ma trận trạng thái, ngõ ra của AddRoundKey, bằng một giá trị đã quy định trong chuẩn AES. Bảng quy định giá trị thay thế gọi là S-box.

|  |
| --- |
| [https://1.bp.blogspot.com/-ocF0DHnd76I/XY9tRJMGxvI/AAAAAAAAJ-Q/vHw8t8jmSFwaeXtyVDRI0ZbeAkggLFNnACKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-ocF0DHnd76I/XY9tRJMGxvI/AAAAAAAAJ-Q/vHw8t8jmSFwaeXtyVDRI0ZbeAkggLFNnACKgBGAsYHg/s1600/image.png) |
| *S-box của mã hóa AES* |

**Ví dụ:**

Byte cần thay thế là H08 thì dò ở hàng số 0 và cột số 8 trong bảng S-box sẽ được kết quả là 30.

|  |  |
| --- | --- |
| [https://1.bp.blogspot.com/-9NMeiih9bvI/XY9tbXh0eOI/AAAAAAAAJ-U/fFw0u8UXSU0P3jf15fKlTR9ozf09O8bpgCKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-9NMeiih9bvI/XY9tbXh0eOI/AAAAAAAAJ-U/fFw0u8UXSU0P3jf15fKlTR9ozf09O8bpgCKgBGAsYHg/s1600/image.png) |  |
| *Chức năng SubBytes* |  |

**Chức năng ShiftRows**

Chức năng ShiftRows thực hiện quay trái từng hàng của ma trận trạng thái, ngõ ra của SubBytes, theo byte với hệ số quay tăng dần từ 0 đến 3. Hàng đầu tiên có hệ số quay là 0 thì các byte được giữ nguyên vị trí. Hàng thứ hai có hệ số quay là 1 thì các byte được quay một byte. Hàng thứ ba quay hai byte và hàng thứ tư quay ba byte.

|  |
| --- |
| [https://1.bp.blogspot.com/-4T2A2aKJKwA/XY9t39J72EI/AAAAAAAAJ-c/JXiqzQCc0JEfXjvr5ZGSm6D6xeDUy5asACKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-4T2A2aKJKwA/XY9t39J72EI/AAAAAAAAJ-c/JXiqzQCc0JEfXjvr5ZGSm6D6xeDUy5asACKgBGAsYHg/s1600/image.png) |
| *Chức năng ShiftRows* |

**Chức năng MixColumns**

Chức năng MixColumns thực hiện nhân từng cột của ma trận trạng thái, ngõ ra của ShiftRows, với một ma trận chuyển đổi quy định bởi chuẩn AES.

|  |
| --- |
| [https://1.bp.blogspot.com/-uCXm3fz21fo/XY9udiS7hsI/AAAAAAAAJ-o/zgA6mMfukzkyBc_99phrWSY5ZYdHaIY2ACKgBGAsYHg/s320/image.png](https://1.bp.blogspot.com/-uCXm3fz21fo/XY9udiS7hsI/AAAAAAAAJ-o/zgA6mMfukzkyBc_99phrWSY5ZYdHaIY2ACKgBGAsYHg/s1600/image.png) |
| *Ma trận chuyển đổi sử dụng trong chức năng  MixColumns* |

Việc biến đổi một cột của ma trận trạng thái được thực hiện bởi hai phép toán là nhân (.) và XOR (+).

Biểu thức sau tạo ra phần tử *H04,* H là ký hiệu của số Hex, ở cột 1 trong hình minh họa "chức năng MixColumns".

***H04 =Hd4.H02 + Hbf.H03 + H5d.H01 + H30.H01=Hd4.H02 + (Hbf.H02 + Hbf.H01) + H5d.H01 + H30.H01***

|  |
| --- |
| [https://1.bp.blogspot.com/-ycQcABB_tr8/XY9vJ3bZqWI/AAAAAAAAJ-w/sOtFCU9yFnUg1NmORH82L4Ei8LbbwrbdACKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-ycQcABB_tr8/XY9vJ3bZqWI/AAAAAAAAJ-w/sOtFCU9yFnUg1NmORH82L4Ei8LbbwrbdACKgBGAsYHg/s1600/image.png) |
| *Chức năng MixColumns* |

Phép nhân với H01 thì giữ nguyên giá trị. Phép nhân với H02 tương đương với việc dịch trái một bit và XOR có điều kiện như sau:

* Nếu bit MSB của giá trị được dịch bằng 1 thì giá trị sau khi dịch được XOR với H1b
* Nếu bit MSB của giá trị được dịch bằng 0 thì giữ giá trị saukhi dịch.

|  |
| --- |
| [https://1.bp.blogspot.com/-MZ0e8mFKWA8/XY9wAkEm32I/AAAAAAAAJ_I/91bKUqN_2wgAG-ClgiXd3JFZs0_o1lJOwCKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-MZ0e8mFKWA8/XY9wAkEm32I/AAAAAAAAJ_I/91bKUqN_2wgAG-ClgiXd3JFZs0_o1lJOwCKgBGAsYHg/s1600/image.png) |
| *Chi tiết về cách tính MixColumns tạo ra phần tử H04 từ cột 1* |

**Chức năng KeyExpansion**

Chức năng KeyExpansion thực hiện tính toán khóa vòng cho bước lặp mã hóa và bước tạo ngõ ra. Kết quả của một lần thực thi KeyExpansion là một khóa vòng sử dụng cho chức năng AddRoundKey. Với mã hóa AES-128, số khóa vòng là 10 tương ứng với 9 lần AddRoundKey ở bước lặp mã hóa và 1 lần AddRoundKey ở bước tạo ngõ ra.  
Chức năng KeyExpansion được thực hiện thông qua 4 chức năng là RotWord, SubWord, AddRcon và AddW.

|  |
| --- |
| [https://1.bp.blogspot.com/-64RMlbYGIXI/XY9w680HrjI/AAAAAAAAJ_U/gmyV7LoGRoYrVbydvxmWGPTtjJ6u243cQCKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-64RMlbYGIXI/XY9w680HrjI/AAAAAAAAJ_U/gmyV7LoGRoYrVbydvxmWGPTtjJ6u243cQCKgBGAsYHg/s1600/image.png) |
| *Chức năng KeyExpansion* |

Mỗi khóa vòng có 128 bit được chia làm 4 word, mỗi word là 4 byte và ký hiệu là w[j] với j là số nguyên. Mã hóa AES-128 có 1 khóa mã và 10 khóa vòng nên tổng số từ là 44 và được đánh số từ 0 đến 43. Khóa mã có 4 từ là w[0], w[1], w[2] và w[3]. Khóa vòng 1 có 4 từ là w[4], w[5], w[6] và w[7]. Tương tự, khóa vòng 10 có 4 từ là w[40], w[41], w[42] và w[43].  
Từ w[j] tính theo công thức sau, với 3 < j < 44.

**w[j] = AddW[j - 4] = w[j - 1] + w[j - 4]**

**w[j = 4∗ n] = AddW[j - 4] = trans(w[j - 1])+ w[j - 4]**

Chú ý, khi tính các từ ở vị trí j là bội số của 4, như w[4], w[8],... và w[40], thì w[j-1] phải được biến đổi qua 3 chức năng RotWord, SubWord và AddRcon, gọi là trans(w[j-1]), trước khi XOR với w[j-4].

Khóa mã key ở mục 1 được sử dụng để minh họa việc tính toán khóa vòng. Khóa mã key[127:0] được chia làm 4 từ như biểu thức sau:

**w[0] = 2b7e1516 w[1] = 28aed2a6**

**w[2] = abf71588 w[3] = 09cf4f3c**

Việc tính toán khóa vòng 1 là thực hiện tính 4 từ w[4], w[5], w[6] và w[7]. Để tính khóa vòng 1, trans(w[3]) phải được tính trước thông qua 3 chức năng RotWord, SubWord và AddRcon.

**w[4] = AddW[0] = trans(w[3])+ w[0]w[5] = AddW[1] = w[4]+ w[1]w[6] = AddW[2] = w[5]+ w[2]w[7] = AddW[3] = w[6]+ w[3]**

Chức năng RotWord Chức năng RotWord thực hiện quay trái từ w[j] một byte.

|  |
| --- |
| [https://1.bp.blogspot.com/-w4Hwq9Uvi98/XY9yH9kuJPI/AAAAAAAAJ_k/Nf8kJJbu3sYt6AWlT0oXw7CnQWazN5X4wCKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-w4Hwq9Uvi98/XY9yH9kuJPI/AAAAAAAAJ_k/Nf8kJJbu3sYt6AWlT0oXw7CnQWazN5X4wCKgBGAsYHg/s1600/image.png) |
| *Thực thi RotWord cho từ w[3]* |

Chức năng SubWord thực hiện thay thế các phi tuyến từng byte của kết quả RotWord theo bảng S-box.

|  |
| --- |
| [https://1.bp.blogspot.com/-mZEVED6EO-o/XY9yU6ECE2I/AAAAAAAAJ_o/IJH7Gvk9Oss_u0LiOzArEMazYBZDMZY2wCKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-mZEVED6EO-o/XY9yU6ECE2I/AAAAAAAAJ_o/IJH7Gvk9Oss_u0LiOzArEMazYBZDMZY2wCKgBGAsYHg/s1600/image.png) |
| *Thực thi SubWord khi chuyển đổi từ w[3]* |

Chức năng AddRcon thực hiện XOR kết quả SubWord và giá trị Rcon[j/4] với j là bội số của 4. Số lượng giá trị Rcon[j/4] là 10 tương ứng với 10 lần tính khóa vòng. Chức năng AddRcon sẽ tạo ra kết quả cuối cùng của biến đổi trans(w[j-1]).  
**Rcon[j/4]**  **Giá trị  HEX**  **Vị trí sử dụng**  
Rcon[1]    01000000        sử dụng cho trans(w[3]) khi tính w[4]  
Rcon[2]    02000000        sử dụng cho trans(w[7]) khi tính w[8]  
Rcon[3]   04000000        sử dụng cho trans(w[11]) khi tính w[12]  
Rcon[4]    08000000        sử dụng cho trans(w[15]) khi tính w[16]  
Rcon[5]    10000000        sử dụng cho trans(w[19]) khi tính w[20]  
Rcon[6]    20000000        sử dụng cho trans(w[23]) khi tính w[24]  
Rcon[7]    40000000        sử dụng cho trans(w[27]) khi tính w[28]  
Rcon[8]    82000000        sử dụng cho trans(w[31]) khi tính w[32]  
Rcon[9]    1b000000        sử dụng cho trans(w[35]) khi tính w[36]  
Rcon[10]  36000000        sử dụng cho trans(w[39]) khi tính w[40]

|  |
| --- |
| [https://1.bp.blogspot.com/-yTCITLH547Q/XY9y6Zr_tAI/AAAAAAAAJ_w/Z8u5HKP_LXkDucO-cEP3RanAMf3G2H6XACKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-yTCITLH547Q/XY9y6Zr_tAI/AAAAAAAAJ_w/Z8u5HKP_LXkDucO-cEP3RanAMf3G2H6XACKgBGAsYHg/s1600/image.png) |
| *Thực thi AddRcon khi chuyển đổi từ w[3]* |

Chức năng AddW thực hiện XOR w[j-4] với w[j-1] hoặc trans(w[j-1]) như công thức 4.8 để tạo ra khóa vòng.

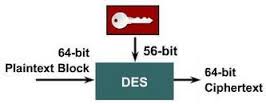
|  |
| --- |
| [https://1.bp.blogspot.com/-nqTfUnO1d4k/XY9zHys2T4I/AAAAAAAAJ_0/Fg-dC42Il1U_Y_k3E4rXvYZTtC77JWyBgCKgBGAsYHg/s640/image.png](https://1.bp.blogspot.com/-nqTfUnO1d4k/XY9zHys2T4I/AAAAAAAAJ_0/Fg-dC42Il1U_Y_k3E4rXvYZTtC77JWyBgCKgBGAsYHg/s1600/image.png) |
| *Thực thi AddW để tạo khóa vòng 1* |

# CHƯƠNG II: MÔ PHỎNG BÀI TOÁN MÃ HÓA DES

## Thuật toán mã hóa DES

### Giới thiệu

Data Encryption Standard (DES) được chuẩn hóa năm 1977 bởi Viện tiêu chuẩn và công nghệ quốc gia Hoa kỳ (NIST). DES được dùng trong nhiều ứng dụng bảo mật như Kerberos, SSL, IPSec, …



* Kích thước khối: 64 bit
* Chiều dài khoá: 64 bit, tuy nhiên trong quá trình thực hiện, chỉ có 56 bit được sử dụng. Tất cả các bit cuối cùng của byte (các bit 8, 16, 24, 32, 40, 48, 56 và 64) bị loại bỏ ngay từ vòng xử lý đầu tiên (Vị trí các bít này dùng chứa mã kiểm tra chẵn lẻ)

Plaintext

IP

L0

R0

L1=R0

R1=L0 ⊕ƒ(R0,K1)

L2=R1

R2=L1 ⊕ƒ(R1,K2)

L15=R14

R15=L14 ⊕ƒ(R14,K15)

R16=L15 ⊕ƒ(R15,K16)

L15=R14

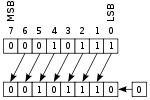
L16=R15

IP-1

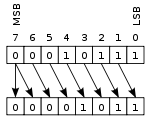
CipherText

* Số vòng mã: 16 vòng
* Thuật toán sinh khoá phụ: kết hợp phép dịch trái và hoán vị
* Hàm F: kết hợp các phép XOR, hoán vị và thay thế (S-box).

### Dịch chuyển số học

[](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Rotate_left_logically.svg)

Dịch chuyển số học trái

[](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Rotate_right_arithmetically.svg)

Dịch chuyển số học phải

Trong dịch chuyển số học, các bit được dịch chuyển ra khỏi đầu hoặc đuôi sẽ bị loại bỏ. Trong phép dịch chuyển số học về bên trái, các số 0 được dịch chuyển vào bên phải; trong phép dịch chuyển số học bên phải, [bit thể hiện dấu](https://vi.wikipedia.org/w/index.php?title=Bit_th%E1%BB%83_hi%E1%BB%87n_d%E1%BA%A5u&action=edit&redlink=1) được thêm vào bên trái, do đó dấu của số được giữ nguyên.

**Ví dụ :**  dưới đây sử dụng thanh ghi 8-bit:

00010111 (số thập phân +23) Dịch chuyển trái

= 00101110 (số thập phân +46)

10010111 (số thập phân -105) Dịch chuyển phải

= 11001011 (số thập phân -53)

Trường hợp đầu tiên, những số tận cùng bên trái được dịch chuyển khỏi [thanh ghi](https://vi.wikipedia.org/wiki/Thanh_ghi), một số 0 mới được thêm vào cuối bên phải của [thanh ghi](https://vi.wikipedia.org/wiki/Thanh_ghi). Trường hợp thứ hai, thành phần cuối bên phải đã được dịch chuyển ra khỏi, và số 1 được thêm vào bên trái, bảo toàn được dấu của số. Nhiều lần dịch chuyển có thể được rút ngắn lại còn một lần. Ví dụ:

00010111 (số thập phân +23) Dịch sang trái 2 lần.

= 01011100 (số thập phân +92)

Dịch chuyển số học bên trái *n* lần tương đương nhân với 2*n* (nếu giá trị đó không gây tràn bộ nhớ), trong khi đó thì phép dịch chuyển số học sang phải *n* lần của một giá trị bù 2 thì tương đương với việc chia cho 2*n* và làm tròn về phía âm vô cùng. Nếu số nhị phân được xem là bù 1, thì phép dịch chuyển sang phải tương tự sẽ cho kết quả bằng với việc chia số đó cho *2n* và làm tròn về phía 0.

### Dịch chuyển luận lý

|  |  |
| --- | --- |
| [150px-Rotate_left_logically](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Rotate_left_logically.svg)  Left logical shift | [150px-Rotate_right_logically](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Rotate_right_logically.svg)  Right logical shift |

Trong *dịch chuyển luận lý*, các số 0 sẽ được dịch chuyển vào để thay thế các bit bị loại bỏ. Do đó dịch chuyển luận lý và dịch chuyển số học bên trái là hoàn toàn giống nhau.

Tuy nhiên, dịch chuyển luận lý thêm giá trị 0 vào vị trí bit quan trọng nhất, thay vì sao chép bit mang dấu, điều này khá lý tưởng cho các số nhị phân không dấu, trong khi phép dịch chuyển số học sang phải thì lại lý tưởng cho các số nhị phân bù 2 có dấu.

### Mã hóa

IP

Thông tin gốc

*64 bít*

Vòng 1

*64 bít*

Vòng 16

*64 bít*

PC-1

Dịch trái

*56 bít*

Dịch trái

*56 bít*

PC-2

*56 bít*

*K1*

*48 bít*

PC-2

*56 bít*

*K16*

*48 bít*

32bit swap

IP-1

Thông tin mật

*64 bít*

*64 bít*

*64 bít*

Khóa bí mật

*64 bít*

*56 bít*

Mã khối

☞Vấn đề cần tìm hiểu:

* IP: Phép hoán vị khởi đầu, IP-1
* Hàm F
* Quy tắc sinh khóa phụ (K1 . . . K16)
* Phép dịch trái

**Phép hoán vị khởi đầu và phép hoán vị ngược**

Có chức năng làm thay đổi vị trí các bit trong khối thông tin gốc. Đây là phần thực hiện không có trong cấu trúc Feistel.

Ở phần cuối của quá trình mã hoá, phép hoán vị ngược sẽ trả lại các bit về vị trí ban đầu của nó. Phép hoán vị IP và IP-1 thực hiện dựa trên hai ma trận như sau, với các giá trị trong ma trận cho biết số thứ tự của bit trong khối.

64 phần tử trong khối được ánh xạ theo chỉ số trong ma trận và được đọc ra theo thứ tự hàng.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 0.1 Ma trận hoán vị IP

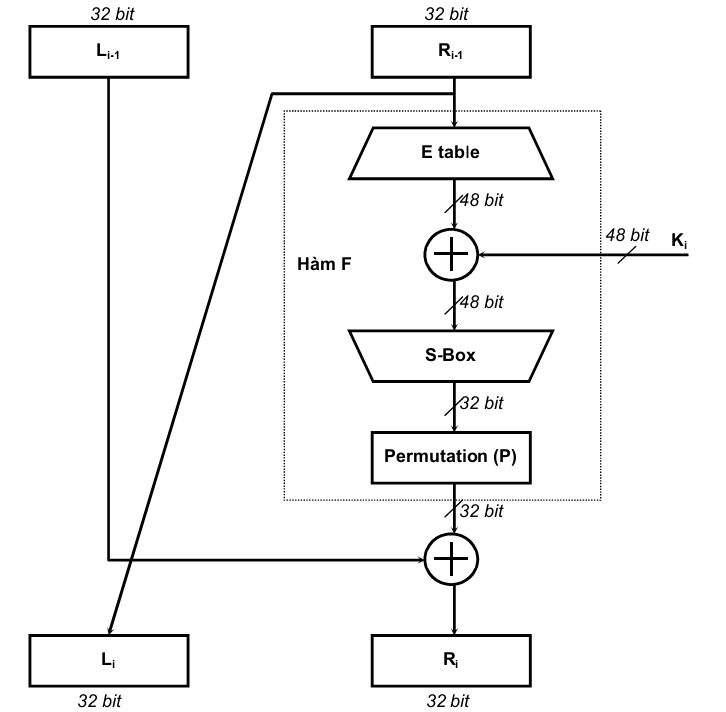
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 | 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 | 37 | 5 | 45 | 13 | 53 | 21 | 61 | 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 0.2 Ma trận hoán vị IP-1

**Hàm F**

Trộn giữa khoá phụ Ki với khối thông tin tại từng vòng. Hàm F trong DES gồm có thao tác:

* Hoán vị mở rộng (E table) chuyển từ 32 bit thành 48 bit,
* Hàm XOR cộng 48 bit vừa tạo ra với 48 bit của khoá phụ Ki,
* Khối thay thế S-Box chuyển 48 bit thành 32 bit,
* Khối hoán vị P



* E table (Expansion/Permutation), đôi khi được gọi là E-box thực hiện chức năng hoán vị các bit trong khối thông tin, đồng thời chuyển từ 32 bit thành 48 bit bằng cách sử dụng ma trận E table như sau (Sẽ có một số bit được lặp lại):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 0.3 Ma trận E table

* S-Box (Substitution Box) Thay thế chuỗi bit thành một chuỗi bit khác, đồng thời thực hiện thao tác ngược lại với E table là chuyển khối thông tin từ 48 bit thành 32 bit. Thực hiện thông qua các ma trận S-Box (S1 – S8)

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

Bảng 0.4 Hộp S1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Bảng 0.5 Hộp S2

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

Bảng 0.6 Hộp S3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 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 | 1 | 7 | 6 | 0 | 8 | 13 |

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

* 48 bit ngõ ra của phép XOR được chia thành 8 phần, mỗi phần 6 bit.
* Từng phần 6 bit được xử lý riêng biệt bằng một ma trận S-Box khác nhau (có 8 S-Box khác nhau).
* Tại mỗi S-Box, bit đầu và bit cuối của phần 6 bit thông tin được dùng để chọn 1 trong 4 hàng của ma trận, 4 bit còn lại được dùng để chọn 1 trong 16 giá trị của hàng tương ứng, giá trị được chọn sẽ chuyển thành 4 bit nhị phân.
* Ví dụ, xét ma trận S1, với chuỗi bit là 111011

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

* + bit đầu và bit cuối là 11=2+1=3, có giá trị thập phân là 2, do đó hàng được chọn là hàng số 3.
  + 4 bit còn lại là 1101=8+4+1=13 nhị phân, giá trị thập phân tương ứng là 6, do đó giá trị tại cột 6 được chọn.
  + Giá trị tại hàng 2 cột 6 trong ma trận S1 là 2, giá trị xuất ra là 0000.
* Ma trận hoán vị P (Permutation): (32 🡪 32). Còn được gọi là P-Box

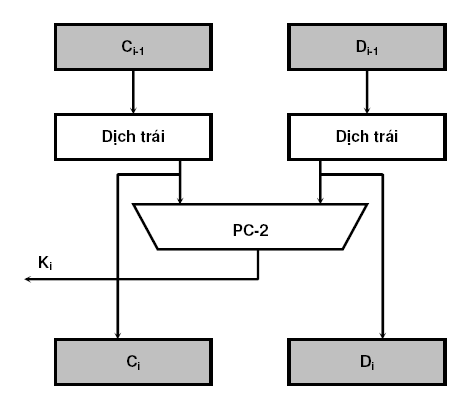
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

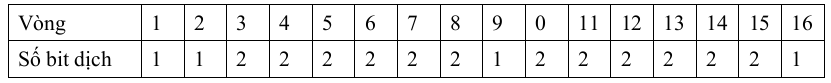
**Thuật toán sinh khóa phụ**

Khóa K (64 bit) đầu vào sẽ giảm xuống còn 56 bit bằng cách bỏ đi 8 bit (ở các vị trí chia hết cho 8), các bit này dùng để kiểm tra bit chẵn lẻ. Sau đó được hoán vị theo PC-1

Bảng trật tự khoá 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 |

* Ngõ ra của khối hoán vị PC-1 được chia thành 2 phần, mỗi phần 28 bit (C và D).
* Tại mỗi vòng mã hoá, hai phần này được dịch trái 1 hoặc 2 bit trước khi đi qua khối hoán vị PC-2 để thành 48 bit
* Số bit dịch trái tương ứng với mỗi vòng là khác nhau, thể hiện bởi bảng sau



* Ma trận hóan vị PC-2 (56 bít 🡪 48 bít)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

**Ví dụ**

Tìm đầu ra vòng thứ nhất (L1, R1) khi thực hiện giải thuật mã hóa DES với 1 khối dữa liệu của bản rõ là 21592FDCBA4390ED16 và khóa là 11223344AABBCCDD16

Chuyển khối dữ liệu sang dạng nhị phân

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 1 | 5 | 9 | 2 | F | D | C | B | A | 4 | 3 | 9 | 0 | E | D |
| 0010 | 0001 | 0101 | 1001 | 0010 | 1111 | 1101 | 1100 | 1011 | 1010 | 0100 | 0011 | 1001 | 0000 | 1110 | 1101 |

Chuyển khóa sang nhị phân

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | A | A | B | B | C | C | D | D |
| 0001 | 0001 | 0010 | 0010 | 0011 | 0011 | 0100 | 0100 | 1010 | 1010 | 1011 | 1011 | 1100 | 1100 | 1101 | 1101 |

Hoán vị bản rõ theo 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 |

Ta có:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

Kết quả sau khi qua IP:

1010101001011010100011001010011111011000100101011001111000110100

**Tách dữ liệu sau khi qua IP thành 2 phần**

L0=10101010010110101000110010100111

R0=11011000100101011001111000110100

* L1 = R0=11011000100101011001111000110100
* R1 = L0 ⊕ F(R0, K1)

Ta cần đi tìm K1, sau đó F(R0, K1), cuối cùng tính L0 ⊕ F(R0, K1)

**Tìm khóa phụ K1**

Hoán vị K (64 bít) theo 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 |

Ta có giá trị khóa K mới 56 bit

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

K (56 bit)=1111000110010000011011010100011011011001000111100000101

* Chia K làm 2 phần
* C0 =1111000011001000001101101010
* D0 =0011011011001000111100000101
* Dịch trái
* C0 =1110000110010000011011010101
* D0 =0110110110010001111000001010
* Khóa K sau khi dịch vòng trái:

11100001100100000110110101010110110110010001111000001010

* Hoán vị K theo PC-2

K1=000110110010011011000001001111110110100011100100

**Tính F(R0, K1)**

* Hoán vị R0 theo E-Table

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Ta có:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

011011110001010010101011110011111100000110101001

* Thực hiện XOR với K1

|  |  |
| --- | --- |
| ⊕ | 011011110001010010101011110011111100000110101001 |
| 000110110010011011000001001111110110100011100100 |
| 48 bít | 100010111100110110010101000011110101011010110010 |

* Cho qua S-Box ta được

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 100010 | 111100 | 110110 | 010101 | 000011 | 110101 | 011010 | 110010 |
| S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 |
| 4 | 2 | 12 | 2 | 11 | 1 | 10 | 6 |
| 0100 | 0010 | 1100 | 0010 | 1011 | 0001 | 1010 | 0110 |

Ta có: 01000010110000101011000110100110

* Hoán vị theo P-Box

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Ta được

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

F(R0, K1)=01100001010000111010010110100001

* Tính R1 = L0 ⊕ F(R0, K1)

|  |  |
| --- | --- |
| ⊕ | 10101010010110101000110010100111 |
| 01100001010000111010010110100001 |
| R1 | 00110100111001101101011011111001 |

**Nhận xét**

* DES dựa hòan tòan trên cấu trúc Feistel.
* Thích hợp với thực thi bằng phần cứng (XOR, dịch, hóan vị) → tốc độ mã cao.
* Chiều dài khóa 56 bit không an toàn với kỹ thuật dò khóa.
* Có thể bị tấn công bằng các phương pháp phân tích tuyến tính và vi phân.
* Cần cải tiến: Double DES, Triple DES

## Bài toán

* CHO INPUT M = 0123456789ABCDEF được mã hóa bằng DES với khóa

K = 133457799BBCDFF1 , Tìm bản mã C ?

* KẾT QUẢ BẢN MÃ C:= DD13E80A381EC0D6

## Mô phỏng mã hóa DES

* Text

  Description automatically generatedL0 , R0 , Bản giải mã và kết Quả:

# KẾT LUẬN

Sau thời gian nghiên cứu đồ án, em rút ra được một số điều:

Về kiến thức: Đồ án thúc đẩy sinh viên tìm tòi, học hỏi không ngừng những kiến thức xoay quanh ứng dụng mình đang tìm hiểu. Giới thiệu một số công nghệ mới và cách ứng dụng trong thực tế. Đào sâu nghiên cứu cái hay, cái hạn chế của một công nghệ để chúng ta có thể linh hoạt hơn khi sử dụng và lựa chọn công nghệ phù hợp với mình và hoàn cảnh. Còn đặt ra một vấn đề với ngành CNTT: nếu không học hỏi liên tục để cập nhật công nghệ mới sẽ mau chóng lạc hậu với thời đại ngay.

Về kỹ năng: Đồ án giúp sinh viên có trải nghiệm thực tế về kĩ năng sử dụng Powerpoint, Word; kĩ năng Quản Lý Thời Gian để kịp hạn nộp báo cáo. Nhưng đặc biệt hơn cả là kĩ năng Thuyết Trình, Phản Biện trước đám đông để bảo vệ cơ sở luận của mình.

Về thái độ: Đồ án rèn luyện cho sinh viên phẩm chất chăm chỉ, tinh thần tự giác, kiên nhẫn, … khi chỉnh sửa đúng chuẩn mẫu thiết kế báo cáo đã cho. Con người kỉ luật – Ý thức kỉ luật – Hành động kỉ luật. Đó cũng là những phẩm chất yêu cầu cấp thiết trong kỉ nguyên công nghiệp 4.0 và Big Data (dữ liệu lớn)

Do hiểu biết, tầm nhìn của bản thân còn hạn chế nên đồ án chỉ dừng lại ở mức thiết kế đơn giản. Đồng thời, cũng không tránh khỏi sai sót, nên em rất mong nhận được những ý kiến đóng góp sâu sắc từ thầy cô để đồ án ngày càng hoàn thiện hơn nữa.

Thật tuyệt vời khi thầy, cô đã tạo điều kiện tốt nhất cho em hoàn thành tốt đồ án này!

Em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

1. Đề cương bài giảng môn Bảo mật hệ thống thông tin, Khoa công nghệ thông tin, Trường Đại học công nghiệp Việt- Hung
2. Giáo trình bảo mật hệ thống thông tin, Đỗ Xuân Đức, Đại học giao thông vận tải
3. <https://www.youtube.com/watch?v=o-Y3IKVv-sM&t=0s>