

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

------🙙🕮🙛-------



**BÁO CÁO THỰC NGHIỆM**

Học phần: An toàn & Bảo mật thông tin

**Chủ đề**: **XÂY DỰNG CHƯƠNG TRÌNH MÃ HOÁ VÀ GIẢI MÃ AES (SỬ DỤNG NGÔN NGỮ C++, C#)**

.

Giáo viên hướng dẫn : TS. Phạm Văn Hiệp

Nhóm sinh viên thực hiện :

1. Trần Anh Đức Mã SV: 2021607565

2. Trần Thái Quyền Mã SV: 2022607309

3. Phạm Văn Thành Mã SV: 2021608254

4. Trương Thị Thuỷ Mã SV: 2021603283

Mã Lớp học phần: 20241IT6001002

Nhóm: 05

Hà Nội - Năm 2024

**MỤC LỤC**

**LỜI NÓI ĐẦU**…………………………………………………………………………

**CHƯƠNG 1: TỔNG QUAN**………………………………………………………….

1.1 Tổng quan về An toàn bảo mật thông tin…………………………………..

1.2 Các kiến thức cơ sở……………………………………………………….

1.3 Nội dung nghiên cứu……………………………………………………...

**CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU**………………………………………….

2.1 Nghiên cứu, tìm hiểu hệ mã hoá bí mật…………………………………….

2.2 Nghiên cứu tìm hiểu hệ mật mã AES……………………………………….

2.3 Thiết kế chương trình, cài đặt thuật toán…………………………………..

**CHƯƠNG 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM**…………………………..

3.1 Kiến thức kỹ năng đã học được trong quá trình thực hiện đề tài……………

3.2 Bài học kinh nghiệm………………………………………………………

3.3 Tính khả thi của chủ đề nghiên cứu, những thuận lợi, khó khăn trong quá trình thực hiện bài tập lớn………………………………………………………..

**TÀI LIỆU THAM KHẢO**……………………………………………………………..

# LỜI NÓI ĐẦU

Sự phát triển vượt bậc của công nghệ mạng dẫn đến vấn đề an toàn thông tin trong là rất quan trọng. Có nhiều phương pháp để trao đổi thông tin mật, trong đó phương pháp mã hóa thông tin được coi là xuất hiện sớm nhất, tuy nhiên phương pháp này làm cho người ta dễ phát hiện. Trong đề tài này sẽ sử dụng phương pháp mã hóa AES (advanced encryption standard) để mã hóa thông tin mật. Nội dung báo cáo gồm 3 chương chính sau:

**Chương 1: Tổng quan**

Giới thiệu tổng quan về mã hóa thông tin, phương pháp mã hóa AES.

**Chương 2: Kết quả nghiên cứu**

Thuật toán, sơ đồ thuật toán, ví dụ minh họa của mã hóa AES, đưa ra công cụ hướng dẫn cài đặt và chạy chương trình

**Chương 3: Phần kiến thức lĩnh hội và bài học kinh nghiệm.**

Tổng kết các phần trong bài tập lớn đã làm được, rút ra bài học kinh nghiệm

## CHƯƠNG 1: TỔNG QUAN

## Tổng quan về an toàn bảo mật thông tin

Khi nhu cầu trao đổi thông tin dữ liệu ngày càng lớn và đa dạng, các tiến bộ về điện tử - viễn thông và công nghệ thông tin không ngừng được phát triển ứng dụng để nâng cao chất lượng và lưu lượng truyền tin thì các quan niệm ý tưởng và biện pháp bảo vệ thông tin dữ liệu cũng được đổi mới. Bảo vệ an toàn thông tin dữ liệu là một chủ đề rộng, có liên quan đến nhiều lĩnh vực và trong thực tế có thể có rất nhiều phương pháp được thực hiện để bảo vệ an toàn thông tin dữ liệu.

Các phương pháp bảo vệ an toàn thông tin dữ liệu có thể được quy tụ vào ba nhóm sau:

- Bảo vệ an toàn thông tin bằng các biện pháp hành chính.

- Bảo vệ an toàn thông tin bằng các biện pháp kỹ thuật (phần cứng).

- Bảo vệ an toàn thông tin bằng các biện pháp thuật toán (phần mềm).

Ba nhóm trên có thể được ứng dụng riêng rẽ hoặc phối kết hợp. Môi trường khó bảo vệ an toàn thông tin nhất và cũng là môi trường đối phương dễ xân nhập nhất đó là môi trường mạng và truyền tin. Biện pháp hiệu quả nhất và kinh tế nhất hiện nay trên mạng truyền tin và mạng máy tính là biện pháp thuật toán. An toàn thông tin bao gồm các nội dung sau: - Tính bí mật: tính kín đáo riêng tư của thông tin - Tính xác thực của thông tin, bao gồm xác thực đối tác( bài toán nhận danh), xác thực thông tin trao đổi. - Tính trách nhiệm: đảm bảo người gửi thông tin không thể thoái thác trách nhiệm về thông tin mà mình đã gửi. Để đảm bảo an toàn thông tin dữ liệu trên đường truyền tin và trên mạng máy tính có hiệu quả thì điều trước tiên là phải lường trước hoặc dự đoán trước các khả năng không an toàn, khả năng xâm phạm, các sự cố rủi ro có thể xảy ra đối với thông tin dữ liệu được lưu trữ và trao đổi trên đường truyền tin cũng như 1 http://www.ebook.edu.vn trên mạng. Xác định càng chính xác các nguy cơ nói trên thì càng quyết định được tốt các giải pháp để giảm thiểu các thiệt hại. Có hai loại hành vi xâm phạm thông tin dữ liệu đó là: vi phạm chủ động và vi phạm thụ động. Vi phạm thụ động chỉ nhằm mục đích cuối cùng là nắm bắt được thông tin (đánh cắp thông tin). Việc làm đó có khi không biết được nội dung cụ thể nhưng có thể dò ra được người gửi, người nhận nhờ thông tin điều khiển giao thức chứa trong phần đầu các gói tin. Kẻ xâm nhập có thể kiểm tra được số lượng, độ dài và tần số trao đổi. Vì vậy vi pham thụ động không làm sai lệch hoặc hủy hoại nội dung thông tin dữ liệu được trao đổi. Vi phạm thụ động thường khó phát hiện nhưng có thể có những biện pháp ngăn chặn hiệu quả. Vi phạm chủ động là dạng vi phạm có thể làm thay đổi nội dung, xóa bỏ, làm trễ, sắp xếp lại thứ tự hoặc làm lặp lại gói tin tại thời điểm đó hoặc sau đó một thời gian. Vi phạm chủ động có thể thêm vào một số thông tin ngoại lai để làm sai lệch nội dung thông tin trao đổi. Vi phạm chủ động dễ phát hiện nhưng để ngăn chặn hiệu quả thì khó khăn hơn nhiều. Một thực tế là không có một biện pháp bảo vệ an toàn thông tin dữ liệu nào là an toàn tuyệt đối. Một hệ thống dù được bảo vệ chắc chắn đến đâu cũng không thể đảm bảo là an toàn tuyệt đối.

## Các kiến thức cơ sở

**1.2.1.Kiến thức về Toán học**

Mã hóa AES (Advanced Encryption Standard) dựa trên các khái niệm toán học cơ bản, bao gồm:

* Đại số tuyến tính: AES sử dụng ma trận trong quá trình mã hóa và giải mã. Việc hiểu biết về phép nhân ma trận và phép biến đổi tuyến tính là rất quan trọng.
* Số nguyên modulo: AES thực hiện nhiều phép toán trên các số nguyên trong không gian modulo 256. Kiến thức về phép toán modulo giúp hiểu rõ cách hoạt động của các phép toán trong AES.
* Hệ thống nhị phân: Dữ liệu trong AES được xử lý ở dạng nhị phân. Kiến thức về hệ nhị phân và cách chuyển đổi giữa các hệ số sẽ hữu ích khi làm việc với byte và bit.

**1.2.2.Kiến thức về Thuật toán**

AES là một thuật toán mã hóa khối, và để hiểu rõ cách thức hoạt động của nó, cần nắm vững các thuật toán và quy trình sau:

* Thuật toán mã hóa và giải mã: Nắm rõ quy trình mã hóa gồm các bước như SubBytes, ShiftRows, MixColumns, và AddRoundKey. Tương tự, quy trình giải mã cũng cần được hiểu tường tận.
* Khóa và quản lý khóa: Khóa AES có độ dài 128, 192, hoặc 256 bit. Kiến thức về cách tạo và quản lý khóa là rất quan trọng để đảm bảo tính bảo mật.
* An toàn thông tin: Hiểu biết về các khái niệm như bảo mật dữ liệu, tấn công mã hóa (ví dụ: tấn công brute-force) và các biện pháp phòng ngừa.

**1.2.3.Kiến thức về Ngôn ngữ lập trình**

Để triển khai chương trình mã hóa và giải mã AES, việc lựa chọn ngôn ngữ lập trình là rất quan trọng. Dưới đây là một số kiến thức cơ bản về C++ và C#:

* C++:
  + C++ cung cấp khả năng quản lý bộ nhớ tốt, cho phép lập trình viên tối ưu hóa hiệu suất xử lý.
  + Sử dụng các thư viện như OpenSSL, MFC có thể giúp triển khai AES một cách dễ dàng, hiệu quả và có giao diện thực hiện dễ nhìn hơn.
* C#:
  + C# có các thư viện tích hợp sẵn cho mã hóa, như System.Security.Cryptography, giúp đơn giản hóa quá trình triển khai.
  + C# hỗ trợ lập trình hướng đối tượng, cho phép tổ chức mã nguồn một cách gọn gàng và dễ bảo trì.

## Nội dung nghiên cứu

Trong bối cảnh công nghệ thông tin ngày càng phát triển, việc bảo mật thông tin trở thành một yêu cầu thiết yếu. Mã hóa dữ liệu không chỉ giúp bảo vệ thông tin nhạy cảm mà còn đảm bảo tính toàn vẹn và xác thực của dữ liệu. Do đó, nghiên cứu về các phương pháp mã hóa, đặc biệt là AES (Advanced Encryption Standard), là rất quan trọng.

**1.3.1. Lý do chọn đề tài**

Đề tài này được chọn vì các lý do sau:

* Tăng cường an ninh thông tin: Với sự gia tăng của các cuộc tấn công mạng và rò rỉ dữ liệu, việc áp dụng các phương pháp mã hóa như AES là cần thiết để bảo vệ thông tin.
* Khả năng ứng dụng rộng rãi: AES được sử dụng rộng rãi trong nhiều ứng dụng, từ truyền thông đến lưu trữ dữ liệu, làm cho nó trở thành một chủ đề nghiên cứu có giá trị.
* Cải thiện kỹ năng lập trình: Thực hiện mã hóa và giải mã AES bằng C++ và C# sẽ giúp sinh viên nâng cao kỹ năng lập trình và hiểu sâu hơn về các thuật toán mã hóa.

**1.3.2. Các nội dung nghiên cứu**

Nội dung nghiên cứu sẽ bao gồm các phần chính sau:

* Tổng quan về mã hóa: Giới thiệu về các phương pháp mã hóa, đặc biệt là mã hóa đối xứng và bất đối xứng.
* Chi tiết về AES: Phân tích cấu trúc và cách thức hoạt động của AES, bao gồm các bước mã hóa và giải mã, và các chế độ hoạt động của AES.
* Cài đặt thuật toán: Triển khai thuật toán AES bằng ngôn ngữ lập trình C++ và C#, bao gồm cách sử dụng thư viện và tối ưu hóa mã nguồn.
* Kiểm thử và đánh giá: Thực hiện các bài kiểm thử để đánh giá hiệu suất và tính an toàn của chương trình mã hóa và giải mã được phát triển.
* Ứng dụng thực tiễn: Nghiên cứu các ứng dụng thực tiễn của AES trong đời sống, chẳng hạn như trong ngân hàng, bảo mật mạng, và lưu trữ dữ liệu.

## CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU

* 1. **Nghiên cứu, tìm hiểu hệ mã hóa khóa bí mật**

Hệ mã hóa khóa bí mật, hay còn gọi là mã hóa đối xứng, là phương pháp mã hóa trong đó cùng một khóa được sử dụng để mã hóa và giải mã dữ liệu. Điều này có nghĩa là cả bên gửi và bên nhận phải biết và giữ bí mật về khóa này. Hệ mã hóa khóa bí mật thường được sử dụng trong nhiều ứng dụng bảo mật nhờ vào tính hiệu quả và khả năng bảo vệ dữ liệu.

**2.1.1 Đặc điểm của hệ mã hóa khóa bí mật**

* Khóa duy nhất: Hệ mã hóa này sử dụng một khóa duy nhất cho cả hai quá trình mã hóa và giải mã. Việc giữ bí mật cho khóa này là rất quan trọng để đảm bảo an toàn cho dữ liệu.
* Tốc độ nhanh: Mã hóa đối xứng thường nhanh hơn so với mã hóa bất đối xứng, do đó nó thường được sử dụng để mã hóa lượng lớn dữ liệu.
* Quản lý khóa: Việc quản lý và phân phối khóa là một thách thức lớn trong hệ thống mã hóa bí mật, vì nếu khóa bị lộ, toàn bộ dữ liệu có thể bị truy cập trái phép.

**2.1.2 Các thuật toán mã hóa khóa bí mật phổ biến**

Một số thuật toán mã hóa khóa bí mật nổi bật bao gồm:

* AES (Advanced Encryption Standard): Là một trong những tiêu chuẩn mã hóa phổ biến nhất hiện nay, AES sử dụng kích thước khóa 128, 192 hoặc 256 bit và rất an toàn trong việc bảo vệ dữ liệu.
* DES (Data Encryption Standard): Mặc dù đã được thay thế bởi AES, DES vẫn là một thuật toán quan trọng trong lịch sử mã hóa. DES sử dụng khóa 56 bit và đã cho thấy nhiều lỗ hổng bảo mật.
* 3DES (Triple DES): Là phiên bản nâng cao của DES, 3DES mã hóa dữ liệu ba lần bằng ba khóa khác nhau, tăng cường độ bảo mật so với DES.

**2.1.3 Ứng dụng của mã hóa khóa bí mật**

Mã hóa khóa bí mật được ứng dụng rộng rãi trong các lĩnh vực như:

* Bảo mật thông tin: Bảo vệ thông tin nhạy cảm trong các giao dịch trực tuyến, lưu trữ dữ liệu, và truyền tải thông tin.
* Bảo vệ mạng: Sử dụng trong các giao thức bảo mật như SSL/TLS để bảo vệ dữ liệu khi truyền qua mạng.
* Lưu trữ an toàn: Mã hóa dữ liệu trước khi lưu trữ để ngăn chặn truy cập trái phép.
* Nghiên cứu về hệ mã hóa khóa bí mật là rất quan trọng trong bối cảnh an ninh thông tin hiện nay. Việc hiểu rõ về các thuật toán như AES sẽ giúp chúng ta xây dựng được các ứng dụng mã hóa và giải mã hiệu quả, bảo vệ dữ liệu một cách an toàn.
  1. **Nghiên cứu, tìm hiểu về hệ mật mã AES**

### 2.2.1 Giới thiệu

AES (Advanced Encryption Standard) là một tiêu chuẩn mã hóa được phát triển và công nhận bởi NIST (Cơ quan tiêu chuẩn và công nghệ quốc gia Hoa Kỳ) vào năm 2001. Được thiết kế để thay thế DES (Data Encryption Standard), AES là một hệ mã hóa đối xứng, có nghĩa là cùng một khóa được sử dụng để mã hóa và giải mã dữ liệu.

**2.2.1.1 Đặc điểm chính của AES:**

* **Kích thước khóa**: AES hỗ trợ ba kích thước khóa khác nhau: 128, 192 và 256 bit. Kích thước khóa lớn hơn cung cấp mức độ bảo mật cao hơn.
* **Quy trình mã hóa**: AES thực hiện một chuỗi các bước xử lý trong nhiều vòng (10, 12 hoặc 14 vòng tùy thuộc vào kích thước khóa). Các bước chính bao gồm:
  + **Thay thế byte**: Thay thế từng byte trong khối dữ liệu bằng byte tương ứng từ một bảng tra cứu.
  + **Trộn hàng**: Xáo trộn các hàng trong khối dữ liệu để tăng cường sự phân tán.
  + **Trộn cột**: Kết hợp các cột để tạo ra sự rối loạn.
  + **Thêm vòng khóa**: Kết hợp khối dữ liệu với một phần của khóa để bảo mật.
* **An toàn và hiệu quả**: AES đã được phân tích rộng rãi và được coi là rất an toàn, với khả năng chống lại các cuộc tấn công mã hóa hiện đại.

**2.2.1.2 Ứng dụng**

AES được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

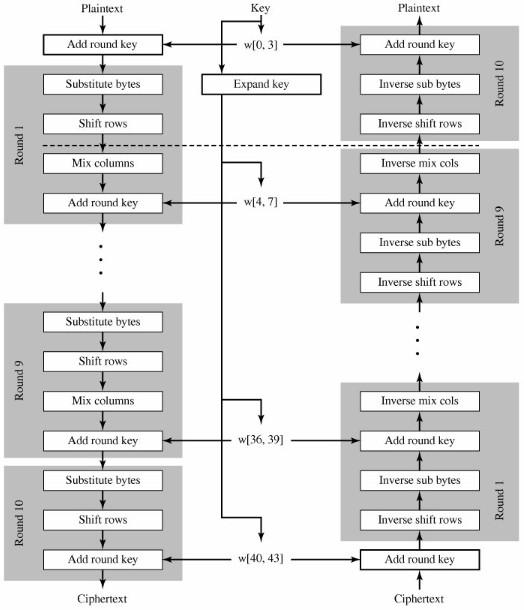
* Bảo mật dữ liệu cá nhân
* Bảo vệ thông tin trong giao dịch tài chính
* Mã hóa thông tin truyền thông qua Internet

Nhờ tính hiệu quả và an toàn, AES đã trở thành một trong những tiêu chuẩn mã hóa phổ biến nhất trên toàn cầu, được áp dụng trong nhiều công nghệ bảo mật hiện đại.

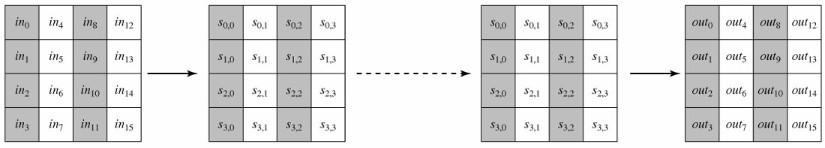
### 2.2.2 Mã hóa và giải mã

Cấu trúc tổng thể của mã hóa AES được mô tả trên *Hình 2.1*. Đầu vào cho thuật toán mã hóa và giải mã là một khối 128 bít, khối bít này được mô tả là một ma trận vuông, mỗi ô là 1 byte. Khối này được sao chép vào một mảng trạng thái, được sửa đổi ở mỗi giai đoạn mã hóa hoặc giải mã. Sau giai đoạn cuối cùng, mảng trạng thái này được sao chép vào một ma trận đầu ra. Các hoạt động này được mô tả trong *Hình 2.2*.

Tương tự, khóa 128 bit được mô tả như một ma trận vuông, mỗi phần tử là một byte. Khóa này sau đó được mở rộng thành một mảng các từ (word), mỗi từ là bốn byte và tổng chiều dài khóa là 44 từ cho khóa 128 bit như *Hình 2.3*. Lưu ý rằng thứ tự của các byte trong ma trận là theo cột. Vì vậy, bốn byte đầu tiên của bản rõ 128 bit đầu vào chiếm cột đầu tiên của ma trận, bốn byte thứ hai chiếm cột thứ hai, v.v. Tương tự, bốn byte đầu tiên của khóa mở rộng, tạo thành một từ, chiếm cột đầu tiên của ma trận w.



*Hình 2.1 Cấu trúc mã hóa và giải mã AES*

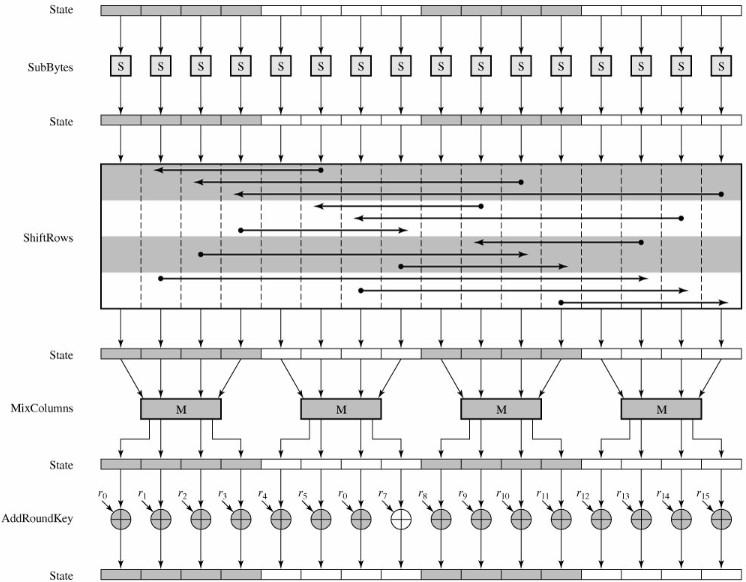


*Hình 2.2. Đầu vào, mảng trạng thái và đầu ra*



*Hình 2.3. Khóa và mở rộng khóa*

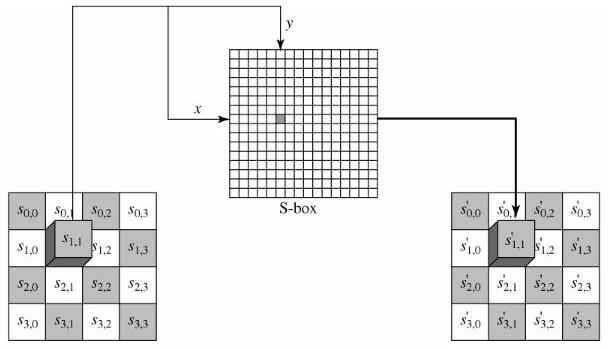
Cấu trúc của thuật toán AES tương đối đơn giản. Cả thuật toán mã hóa và giải mã đều bắt đầu giai đoạn AddRoundKey, tiếp theo là 9 vòng, mỗi vòng đầy đủ 4 giai đoạn: Thay thế các bytes (Substitute bytes) sử dụng hộp S để thực hiện việc thay thế từng byte của khối; dịch các dòng (ShiftRows) đơn giản là thực hiện hoán vị; trộn cột (MixColumns) là phép thay thế sử dụng các phép toán số học trên Z256; AddRoundKey đơn giản chỉ là phép XOR của khối hiện tại với một phần của khóa được mở rộng. Vòng cuối cùng chỉ có 3 giai đoạn. *Hình 2.4* minh họa một vòng mã hóa đầy đủ.



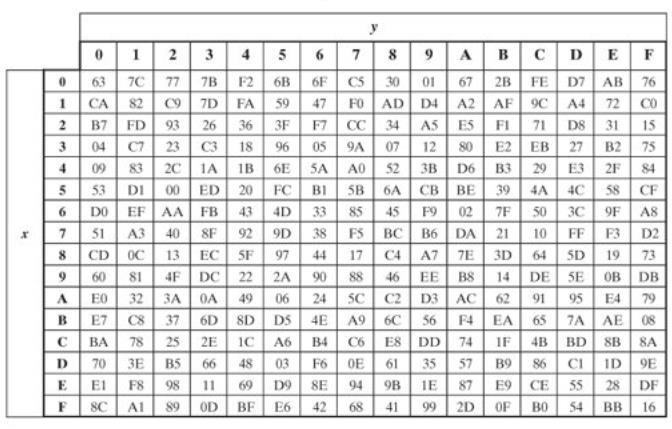
*Hình 2.4. Minh họa một vòng mã AES*

Tiếp teo ta đi xét chi tiết các giai đoạn trong một vòng mã hóa.

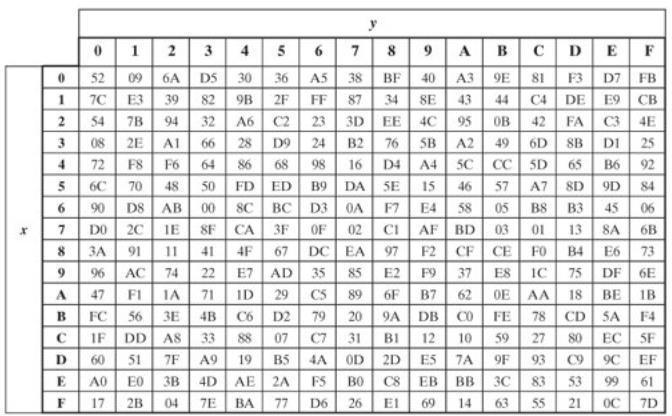
#### 2.2.2.1 Thay thế byte

*Thay thế byte đơn giản chỉ là tra cứu trong bảng 16 x 16, mỗi ô là 1 byte và được gọi là hộp S như bảng 2.1 và hộp S đảo như bảng 2.2. Minh họa việc tra cứu hộp S như Hình 2.5*

*Hình 2.5. Phép thay thế byte sử dụng hộp S*

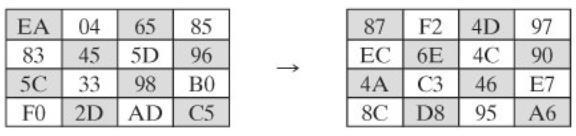


*Bảng 2.1 Hộp S*



*Bảng 2.2 Hộp S đảo (inverse S box)*

Ví dụ minh họa phép thay thế byte như *Hình 2.6*



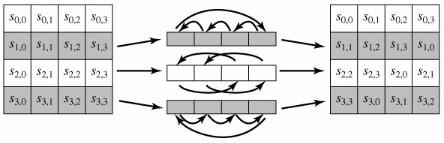
*Hình 2.6. Minh họa phép thay thế byte*

Để tìm byte thay thế của byte EA, ta tra dòng E và cột A trong hộp S thu được byte 87.

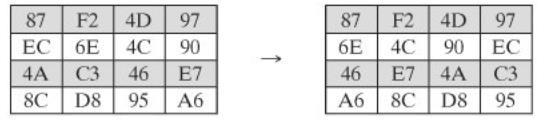
Như vậy, byte EA sẽ được thay thế bằng byte 87. Tương tự, byte 04 ta tra dòng 0 và cột 4 thu được F2.

Ta làm tương tự cho các byte còn lại sẽ thu được ma trận kết quả sau khi thực hiện phép thay thế.

#### 2.2.2.2 Dịch dòng (Shiftrows)

*Hình 2.7* minh họa phép dịch dòng. Dòng đầu tiên của ma trận trạng thái được giữ nguyên, dòng thứ hai quay trái 1 byte, dòng thứ 3 quay trái 2 byte và dòng cuối cùng quay trái 3 byte. *Hình 2.8* ví dụ minh họa phép dịch dòng.

*Hình 2.7. Minh họa phép dịch dòng*

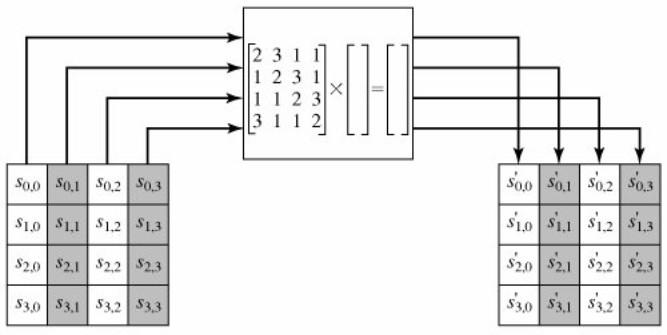


*Hình 2.8. Ví dụ minh họa phép dịch dòng*

Đối với thuật toán giải mã ta sử dụng phép dịch dòng ngược. Tức là, dòng đầu tiên của ma trận trạng thái giữ nguyên, dòng thứ 2 quay phải 1 byte, dòng thứ 3 quay phải 2 bytes và dòng cuối cùng quay phải 3 bytes.

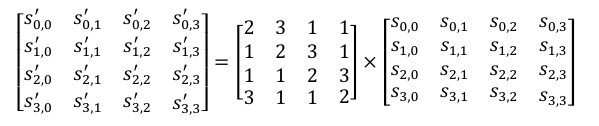
#### 2.2.2.3 Trộn cột

Phép trộn cột được thực hiện như minh họa trên *Hình 2.9*.



*Hình 2.9. Minh họa phép trộn cột*

Như vậy, kết quả của phép trộn cột sẽ được xác định theo công thức sau:



Áp dụng phép nhân hai ma trận ta thu được:

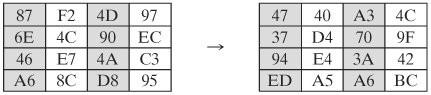
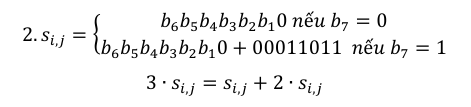
𝑠′ = (2 ∙ 𝑠0,𝑗) + (3 ∙ 𝑠1,𝑗) + 𝑠2,𝑗 + 𝑠3,𝑗

𝑠′ = 𝑠0,𝑗 + (2 ∙ 𝑠1,𝑗) + (3 ∙ 𝑠2,𝑗) + 𝑠3,𝑗

𝑠′ = 𝑠0,𝑗 + 𝑠1,𝑗 + (2 ∙ 𝑠2,𝑗) + (3 ∙ 𝑠3,𝑗)

𝑠′ = (3 ∙ 𝑠0,𝑗) + 𝑠1,𝑗 + 𝑠2,𝑗 + (3 ∙ 𝑠3,𝑗)

Trong đó, phép nhân (.) được thực hiện theo luật sau: Giả sử si,j được biểu diễn dưới dạng 8 bít b7b6b5b4b3b2b1b0 khi nhân với 2 sẽ được thực hiện theo công thức sau:

Phép cộng (+) trong các công thức trên là phép XOR bit

Ta diễn giải cách xác định phần tử đầu tiên trong ma trận sau khi thực hiện phép trộn cột. 

𝑠′= 2 ∙ (87) + 3. (6𝐸) + 46 + 𝐴6.

Chuyển các số từ hệ 16 sang hệ 2 thu được 87h = 10000111. Do bít b7 = 1 nên 2.(87)

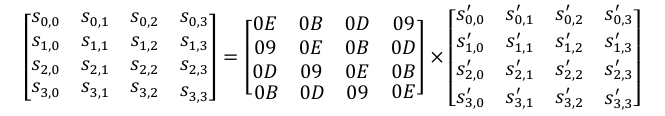
= 00001110 XOR 00011011 = 00010101, 6Eh = 01101110, 46h = 01000110, A6h = 10100110 và 3.(6E) = 6E + 2.(6E). Do bít b7 của 6E là 0 nên 2.(6E) = 11011100. Do đó,

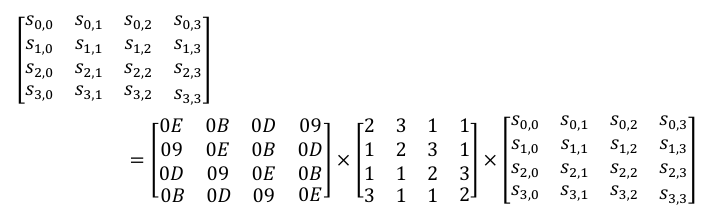
3.(6E) = 01101110 XOR 11011100 = 10110010.

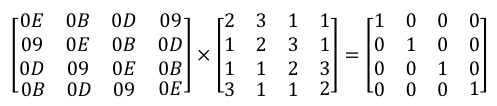
| 2.(87) | = | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3.(6E) | = | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 46 | = | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| A6 | = | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| XOR |  | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 = 47h |

Tính toán tương tự cho các phần tử còn lại ta thu được trạng thái sau khi thực hiện phép trộn cột.

Phép chuyển đổi đảo trộn cột (inverse mix column transform) trong thuật toán giải mã được thực hiện như sau:



Thay thế công thức của phép trộn cột thì ta thu được công thức sau:



Như vậy, thì công thức sau phải được thỏa mãn

Ta chứng minh phần tử đầu tiên thỏa mãn yêu cầu. Thật vậy, 2.(0E) + 0B + 0D + 3.(09) = 00011100 + 00001011 + 00001101 + 3.(09). Trong đó, 3.(09) = 09 + 2.(09) =

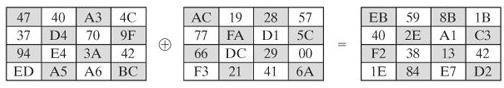
00001001 XOR 00010010 = 00011011. Cuối cùng ta thu được.

| 2.(0E) | = | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0B | = | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0D | = | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 3.(09) | = | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| XOR |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 = 1 |

Các phần tử còn lại người đọc tự chứng mình như là một bài tập.

#### 2.2.2.4 Cộng với khóa (add round key)

Phép cộng với khóa là thực hiện phép XOR bít của 128 bít của ma trận trạng thái và 128 bít của khóa tương ứng của vòng. *Hình 2.10* minh họa ví dụ thực hiện phép cộng khóa, ma trận đầu tiên là trạng thái và ma trận thứ 2 là khóa của vòng.

*Hình 2.10. Ví dụ minh họa phép cộn*g *khóa*

Mở rộng khóa

Thuật toán mở rộng khóa có đầu vào là 4 từ (16 bytes) khóa và tạo ra một mảng đầu ra 44 từ (176 bytes). Mã giả của thuật toán được mô tả như sau:

KeyExpansion (byte key[16], word w[44])

{

word temp for(i=0;i<4;i++)

w[i] = (key[4\*i], key[4\*i+1], key[4\*i+2], key[4\*i+3])

for(i=4,i<44;i++)

{

temp = w[i-1] if(i mod 4 = 0)

temp = SubWord(RotWord(temp)) XOR Rcon[i/4] w[i] = w[i-4] XOR temp

}

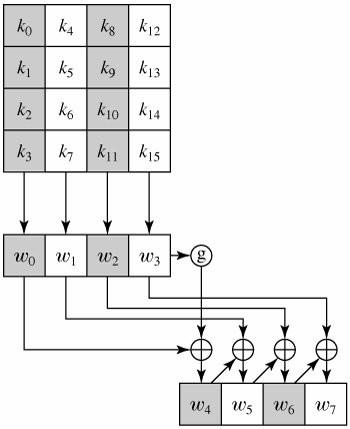
}

Trong đó, phép toán RotWord là thực hiện phép quay trái 1 byte, tức là đầu vào l từ có 4 byte [b0, b1, b2, b3] thì kết quả sau khi thực hiện phép quay trái 1 byte sẽ là [b1, b2, b3, b0]. Phép toán SubWord là phép thay thế byte sử dụng bảng S. Hằng số cho mỗi vòng khóa Rcon[j] = (RC[j], 0, 0, 0), với RC[1] = 1, RC[j] = 2.RC[j-1] và phép nhân (.) được thực hiện theo luật như trong thuật toán trộn cột. Giá trị của RC[j] được xác định như

*Bảng 2.3* ở hệ thập lục phân (hexadecimal).

| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RC[j] | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | 1B | 36 |

*Bảng 2.3. Giá trị của RC[j]*



*Hình 2.11. Minh họa cách xác định khóa của vòng 1*

Ví dụ minh họa cách xác định khóa cho vòng thứ 9 khi khóa tại vòng 8 là EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F tương ứng w[32] = [EA, D2, 73, 21], w[33] = [B5, 8D, BA, D2], w[34] = [31, 2B, F5, 60] và w[35] = [7F, 8D, 29, 2F]. Giá trị của khóa tại vòng 9 được xác định như bảng sau:

| Giá trị ở hệ thập phân | temp | Sau khi thực hiện phép RotWord | Sau khi thực hiện phép SubWord | Rcon(9) | Sau khi XOR với Rcon | w[i-4] | w[i] = temp XOR w[i-4] |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 36 | 7F8D292F | 8D292F7F | 5DA515D2 | 1B000000 | 46A515D2 | EAD27321 | AC7766F3 |
| 37 | AC7766F3 | AC7766F3 | AC7766F3 | 1B000000 | AC7766F3 | B58DBAD2 | 19FABC21 |
| 38 | 19FABC21 | 19FABC21 | 19FABC21 | 1B000000 | 19FABC21 | 312BF560 | 28B14941 |
| 39 | 28B14941 | 28B14941 | 28B14941 | 1B000000 | 28B14941 | 7F8D292F | 575C606E |

*Bảng 2.4. Ví dụ xác định khóa tại vòng 8*

Như vậy, khóa của vòng 9 sẽ là AC 77 66 F3 19 FA BC 21 28 B1 49 41 57 5C 60 6E.

* 1. **Thiết kế chương trình, cài đặt thuật toán**

**2.3.1 Thiết kế kịch bản chương trình**

Kịch bản chương trình cần xác định các chức năng chính và luồng xử lý của ứng dụng. Dưới đây là một ví dụ về kịch bản cho chương trình mã hóa và giải mã AES:

* Giao diện người dùng:
  + Hiển thị menu lựa chọn:
    1. Mã hóa dữ liệu
    2. Giải mã dữ liệu
    3. Thoát chương trình
* Chức năng Mã hóa:
  + Nhận đầu vào từ người dùng (chuỗi văn bản cần mã hóa và khóa)
  + Kiểm tra tính hợp lệ của khóa
  + Thực hiện mã hóa bằng thuật toán AES
  + Hiển thị kết quả mã hóa
* Chức năng Giải mã:
  + Nhận đầu vào từ người dùng (chuỗi văn bản đã mã hóa và khóa)
  + Kiểm tra tính hợp lệ của khóa
  + Thực hiện giải mã bằng thuật toán AES
  + Hiển thị kết quả giải mã
* Xử lý lỗi:
  + Thông báo lỗi nếu đầu vào không hợp lệ (khóa không đúng định dạng, văn bản không mã hóa, v.v.)

**2.3.2 Giới thiệu ngôn ngữ lập trình sử dụng để cài đặt thuật toán (C++, C#)**

**2.3.2.1 Giới thiệu ngôn ngữ lập trình C++**

C++ là một trong những ngôn ngữ lập trình phổ biến nhất và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ phát triển phần mềm đến hệ thống nhúng. Dưới đây là một số đặc điểm nổi bật của C++.

Lịch sử và Phát triển:

* Ra đời: C++ được phát triển bởi Bjarne Stroustrup vào đầu những năm 1980 như một phần mở rộng của ngôn ngữ C, với mục tiêu thêm tính năng lập trình hướng đối tượng.
* Phiên bản: C++ đã trải qua nhiều phiên bản, với các tiêu chuẩn quan trọng như C++98, C++11, C++14, C++17 và C++20, mỗi phiên bản đều giới thiệu nhiều tính năng mới.

Tính Năng:

* Hướng đối tượng: C++ hỗ trợ lập trình hướng đối tượng (OOP), cho phép lập trình viên tạo ra các lớp và đối tượng, giúp tổ chức mã nguồn hiệu quả.
* Kiểm soát tài nguyên: C++ cho phép lập trình viên kiểm soát tài nguyên hệ thống một cách chi tiết, bao gồm quản lý bộ nhớ thông qua con trỏ.
* Tính tương thích với C: C++ hoàn toàn tương thích với C, cho phép sử dụng các thư viện C hiện có và tích hợp mã C vào trong chương trình C++.

Hiệu Suất:

* Hiệu suất cao: C++ được biên dịch thành mã máy, giúp chương trình chạy nhanh và hiệu quả. Điều này làm cho nó trở thành lựa chọn hàng đầu cho các ứng dụng yêu cầu hiệu suất cao, như game, hệ điều hành, và các ứng dụng nhúng.

Thư Viện và Công Cụ:

* Thư viện phong phú: C++ có một số lượng lớn thư viện, bao gồm STL (Standard Template Library), Boost, và các thư viện khác hỗ trợ cho việc phát triển ứng dụng.
* Công cụ phát triển: Có nhiều IDE và công cụ phát triển cho C++, như Visual Studio, Code::Blocks, và CLion, giúp lập trình viên dễ dàng lập trình và gỡ lỗi.

Ứng Dụng:

C++ được sử dụng trong nhiều lĩnh vực, bao gồm:

* Phát triển phần mềm: Các ứng dụng desktop, phần mềm hệ thống.
* Game: Nhiều game nổi tiếng được phát triển bằng C++ vì hiệu suất và khả năng xử lý đồ họa.
* Hệ thống nhúng: C++ thường được sử dụng trong lập trình cho các thiết bị nhúng và hệ thống thời gian thực.
* C++ là một ngôn ngữ lập trình mạnh mẽ và linh hoạt, phù hợp cho nhiều loại ứng dụng khác nhau. Với khả năng kiểm soát tốt tài nguyên và hiệu suất cao, C++ tiếp tục là một lựa chọn hàng đầu cho các lập trình viên trên toàn thế giới.

**2.3.2.2 Giới thiệu ngôn ngữ lập trình C#**

C# là một ngôn ngữ lập trình hiện đại được phát triển bởi Microsoft vào đầu những năm 2000. Nó là một phần của nền tảng .NET và được thiết kế để đơn giản hóa việc phát triển ứng dụng trên Windows cũng như trên các nền tảng khác.

Lịch Sử và Phát Triển

* Ra đời: C# được giới thiệu lần đầu vào năm 2000 và nhanh chóng trở thành một trong những ngôn ngữ lập trình phổ biến nhất.
* Phiên bản: C# đã trải qua nhiều phiên bản, từ C# 1.0 đến C# 10.0, mỗi phiên bản đều mang lại nhiều tính năng mới và cải tiến.

Tính Năng

* Hướng đối tượng: C# hỗ trợ lập trình hướng đối tượng, cho phép tổ chức mã nguồn bằng cách sử dụng các lớp, đối tượng, kế thừa và đa hình.
* Đơn giản và dễ học: C# có cú pháp rõ ràng và dễ hiểu, giúp lập trình viên nhanh chóng làm quen và phát triển ứng dụng.
* Quản lý bộ nhớ tự động: C# sử dụng garbage collection để tự động quản lý bộ nhớ, giúp giảm thiểu rủi ro rò rỉ bộ nhớ.

Hiệu Suất

* Tích hợp với .NET: C# được tối ưu hóa cho nền tảng .NET, cho phép phát triển ứng dụng nhanh chóng và hiệu quả. Các ứng dụng C# có thể chạy trên nhiều hệ điều hành khác nhau nhờ vào .NET Core.
* Hiệu suất tốt: Mặc dù không nhanh bằng C hoặc C++, C# vẫn cung cấp hiệu suất tốt cho hầu hết các ứng dụng.

Thư Viện và Công Cụ

* Thư viện phong phú: C# đi kèm với một loạt thư viện và framework, bao gồm ASP.NET cho phát triển web, Windows Presentation Foundation (WPF) cho ứng dụng desktop, và Xamarin cho ứng dụng di động.
* Công cụ phát triển: Visual Studio là IDE phổ biến nhất cho C#, cung cấp nhiều công cụ hỗ trợ lập trình, gỡ lỗi và triển khai ứng dụng.

Ứng Dụng

C# được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

* Phát triển ứng dụng desktop: C# thường được sử dụng để phát triển các ứng dụng Windows với giao diện người dùng thân thiện.
* Phát triển web: ASP.NET cho phép xây dựng các ứng dụng web động và dịch vụ web.
* Phát triển game: C# là ngôn ngữ chính trong Unity, một trong những engine phát triển game phổ biến nhất hiện nay.
* Phát triển ứng dụng di động: Với Xamarin, lập trình viên có thể phát triển ứng dụng di động đa nền tảng bằng C#.
* C# là một ngôn ngữ lập trình mạnh mẽ và linh hoạt, phù hợp cho cả lập trình viên mới bắt đầu và những người có kinh nghiệm. Với sự hỗ trợ mạnh mẽ từ Microsoft và cộng đồng, C# tiếp tục là một lựa chọn hàng đầu cho việc phát triển ứng dụng trên nhiều nền tảng khác nhau.

**2.3.3 Cài đặt thuật toán, giao diện chương trình**

**2.3.1 Ngôn ngữ C++**

* **Các hàm nhân Galois:**

unsigned int Nhan2(unsigned int w)

{

unsigned int kq = w << 1;

if(kq > 256) kq = kq ^ 0x11b;

kq = kq & 0xFF;

return kq;

}

unsigned int Nhan3(unsigned int w)

{

unsigned int kq = w ^ Nhan2(w);

kq = kq & 0xFF;

return kq;

**}**

unsigned int Nhan9(unsigned int w)

{

unsigned int kq = (w << 3) ^ w;

if(kq > (256 << 2)) kq = kq ^ (0x11b << 2);

if(kq > (256 << 1)) kq = kq ^ (0x11b << 1);

if(kq > 256) kq = kq ^ 0x11b;

kq = kq & 0xFF;

return kq;

}

unsigned int NhanB(unsigned int w)

{

unsigned int kq = (w << 3) ^ (w << 1) ^ w;

if(kq > (256 << 2)) kq = kq ^ (0x11b << 2);

if(kq > (256 << 1)) kq = kq ^ (0x11b << 1);

if(kq > 256) kq = kq ^ 0x11b;

kq = kq & 0xFF;

return kq;

}

unsigned int NhanD(unsigned int w)

{

unsigned int kq = (w << 3) ^ (w << 2) ^ w;

if(kq > (256 << 2)) kq = kq ^ (0x11b << 2);

if(kq > (256 << 1)) kq = kq ^ (0x11b << 1);

if(kq > 256) kq = kq ^ 0x11b;

kq = kq & 0xFF;

return kq;

}

unsigned int NhanE(unsigned int w)

{

unsigned int kq = (w << 3) ^ (w << 2) ^ (w << 1);

if(kq > (256 << 2)) kq = kq ^ (0x11b << 2);

if(kq > (256 << 1)) kq = kq ^ (0x11b << 1);

if(kq > 256) kq = kq ^ 0x11b;

kq = kq & 0xFF;

return kq;

}

* **Hàm RotWords:**

unsigned int RotWord(unsigned int w)

{

//Dich vong trai 1 byte

unsigned int byte1 = (w >> 24) & 0xFF;

unsigned int byte234 = w & 0xFFFFFF;

unsigned int rot = (byte234 << 8) | byte1;

return rot;

}

* **Hàm KeyExpansion:**

unsigned int\* KeyExpansion(unsigned int Key[4])

{

unsigned int \*w = new unsigned int[44];

w[0] = Key[0];

w[1] = Key[1];

w[2] = Key[2];

w[3] = Key[3];

for(int i = 4; i < 44; i++)

{

if(i % 4 == 0) w[i] = G(w[i - 1], i/4) ^ w[i - 4];

else w[i] = w[i - 1] ^ w[i - 4];

}

return w;

}

* **Các hàm addRoundKey, subBytes, shiftRows, gMixColumns**

unsigned int\* AddRoundKey(unsigned int state[4], unsigned int \*K)

{

unsigned int \*kq = new unsigned int[4];

kq[0] = state[0] ^ K[0];

kq[1] = state[1] ^ K[1];

kq[2] = state[2] ^ K[2];

kq[3] = state[3] ^ K[3];

return kq;

}

unsigned int\* SubBytes(unsigned int state[4])

{

unsigned int \*kq = new unsigned int[4];

for (int i = 0; i < 4; i++)

kq[i] = SubWord(state[i]);

return kq;

}

unsigned int\* ShiftRows(unsigned int state[4])

{

unsigned int \*kq = new unsigned int[4];

for (int i = 0; i < 4; i++)

{

unsigned int byte1 = state[i] & 0xFF000000;

unsigned int byte2 = state[(i + 1) % 4] & 0xFF0000;

unsigned int byte3 = state[(i + 2) % 4] & 0xFF00;

unsigned int byte4 = state[(i + 3) % 4] & 0xFF;

kq[i] = byte1 | byte2 | byte3 | byte4;

}

return kq;

}

* **Hàm mã hóa AES**

unsigned int\* MahoaAES(unsigned int state[4], unsigned int key[4])

{

unsigned int \*w = KeyExpansion(key);

state = AddRoundKey(state,&w[0]);

for(int j = 1; j <= 9; j++)

{

state = SubBytes(state);

state = ShiftRows(state);

state = MixColumns(state);

state = AddRoundKey(state, &w[4\*j]);

}

//Vong thu 10

state = SubBytes(state);

state = ShiftRows(state);

state = AddRoundKey(state,&w[40]);

unsigned int \*kq = new unsigned int[4];

kq = state;

return kq;

}

* **Các hàm giải mã AES**

unsigned int\* InvShiftRows(unsigned int state[4])

{

unsigned int \*kq = new unsigned int[4];

for (int i = 0; i < 4; i++)

{

unsigned int byte1 = state[i] & 0xFF000000;

unsigned int byte2 = state[(i + 3) % 4] & 0xFF0000;

unsigned int byte3 = state[(i + 2) % 4] & 0xFF00;

unsigned int byte4 = state[(i + 1) % 4] & 0xFF;

kq[i] = byte1 | byte2 | byte3 | byte4;

}

return kq;

}

unsigned int\* InvSubBytes(unsigned int state[4])

{

unsigned int \*kq = new unsigned int[4];

for (int i = 0; i < 4; i++)

kq[i] = InvSubWord(state[i]);

return kq;

}

unsigned int InvNhanCot(unsigned int w)

{

unsigned int kq;

unsigned int byte1 = (w >> 24) & 0xFF;

unsigned int byte2 = (w >> 16) & 0xFF;

unsigned int byte3 = (w >> 8) & 0xFF;

unsigned int byte4 = w & 0xFF;

unsigned int kq1 = NhanE(byte1) ^ NhanB(byte2) ^ NhanD(byte3) ^ Nhan9(byte4);

unsigned int kq2 = Nhan9(byte1) ^ NhanE(byte2) ^ NhanB(byte3) ^ NhanD(byte4);

unsigned int kq3 = NhanD(byte1) ^ Nhan9(byte2) ^ NhanE(byte3) ^ NhanB(byte4);

unsigned int kq4 = NhanB(byte1) ^ NhanD(byte2) ^ Nhan9(byte3) ^ NhanE(byte4);

kq = (kq1 << 24) | (kq2 << 16) | (kq3 << 8) | kq4;

return kq;

}

unsigned int\* InvMixColumns(unsigned int state[4])

{

unsigned int \*kq = new unsigned int[4];

for (int i = 0; i < 4; i++)

kq[i] = InvNhanCot(state[i]);

return kq;

}

* **Hàm giải mã AES**

unsigned int\* GiaimaAES(unsigned int C[4], unsigned int key[4])

{

unsigned int \*w = KeyExpansion(key);

unsigned int \*state = AddRoundKey(C, &w[40]);

for(int j = 1; j <= 9; j++)

{

state = InvShiftRows(state);

state = InvSubBytes(state);

state = AddRoundKey(state, &w[40 - 4\*j]);

state = InvMixColumns(state);

}

//Vong thu 10

state = InvShiftRows(state);

state = InvSubBytes(state);

state = AddRoundKey(state, &w[0]);

return state;

}

* Các thành phần giao diện và bảng sự kiện

private:

void OnEncrypt(wxCommandEvent& event);

void OnDecrypt(wxCommandEvent& event);

wxTextCtrl\* m\_keyInputEncrypt;

wxTextCtrl\* m\_inputTextEncrypt;

wxTextCtrl\* m\_outputTextEncrypt;

wxButton\* m\_encryptButton;

wxTextCtrl\* m\_keyInputDecrypt;

wxTextCtrl\* m\_inputTextDecrypt;

wxTextCtrl\* m\_outputTextDecrypt;

wxButton\* m\_decryptButton;

wxDECLARE\_EVENT\_TABLE();

};

// Bảng sự kiện cho việc xử lý sự kiện

wxBEGIN\_EVENT\_TABLE(AESFrame, wxFrame)

EVT\_BUTTON(1001, AESFrame::OnEncrypt)

EVT\_BUTTON(1002, AESFrame::OnDecrypt)

wxEND\_EVENT\_TABLE()

* Hàm đọc dữ liệu từ file và điền dữ liệu vào textbox của hàm mã hóhóa

wxBoxSizer\* encryptSizer = new wxBoxSizer(wxVERTICAL);

wxStaticText\* keyLabelEncrypt = new wxStaticText(panel, wxID\_ANY, "Nhap\_khoa\_ma\_hoa\_32\_ky\_tu\_hex:");

encryptSizer->Add(keyLabelEncrypt, 0, wxALL, 5);

m\_keyInputEncrypt = new wxTextCtrl(panel, wxID\_ANY, "", wxDefaultPosition, wxSize(300, 30));

encryptSizer->Add(m\_keyInputEncrypt, 0, wxALL, 5);

wxStaticText\* inputLabelEncrypt = new wxStaticText(panel, wxID\_ANY, "Nhap\_van\_ban\_can\_ma\_hoa\_32\_ky\_tu\_hex:");

encryptSizer->Add(inputLabelEncrypt, 0, wxALL, 5);

m\_inputTextEncrypt = new wxTextCtrl(panel, wxID\_ANY, "", wxDefaultPosition, wxSize(300, 30));

encryptSizer->Add(m\_inputTextEncrypt, 0, wxALL, 5);

m\_encryptButton = new wxButton(panel, 1001, "Ma\_hoa", wxDefaultPosition, wxSize(100, 30));

encryptSizer->Add(m\_encryptButton, 0, wxALL, 5);

wxStaticText\* outputLabelEncrypt = new wxStaticText(panel, wxID\_ANY, "Ket\_qua\_sau\_khi\_ma\_hoa:");

encryptSizer->Add(outputLabelEncrypt, 0, wxALL, 5);

m\_outputTextEncrypt = new wxTextCtrl(panel, wxID\_ANY, "", wxDefaultPosition, wxSize(300, 120), wxTE\_MULTILINE | wxTE\_READONLY);

encryptSizer->Add(m\_outputTextEncrypt, 0, wxALL, 5);

* Hàm đọc dữ liệu từ file và điền dữ liệu vào textbox của hàm giải mã

wxBoxSizer\* decryptSizer = new wxBoxSizer(wxVERTICAL);

wxStaticText\* keyLabelDecrypt = new wxStaticText(panel, wxID\_ANY, "Nhap\_khoa\_giai\_ma\_32\_ky\_tu\_hex:");

decryptSizer->Add(keyLabelDecrypt, 0, wxALL, 5);

m\_keyInputDecrypt = new wxTextCtrl(panel, wxID\_ANY, "", wxDefaultPosition, wxSize(300, 30));

decryptSizer->Add(m\_keyInputDecrypt, 0, wxALL, 5);

wxStaticText\* inputLabelDecrypt = new wxStaticText(panel, wxID\_ANY, "Nhap\_van\_ban\_ma\_hoa\_32\_ky\_tu\_hex:");

decryptSizer->Add(inputLabelDecrypt, 0, wxALL, 5);

m\_inputTextDecrypt = new wxTextCtrl(panel, wxID\_ANY, "", wxDefaultPosition, wxSize(300, 30));

decryptSizer->Add(m\_inputTextDecrypt, 0, wxALL, 5);

m\_decryptButton = new wxButton(panel, 1002, "Giai\_ma", wxDefaultPosition, wxSize(100, 30));

decryptSizer->Add(m\_decryptButton, 0, wxALL, 5);

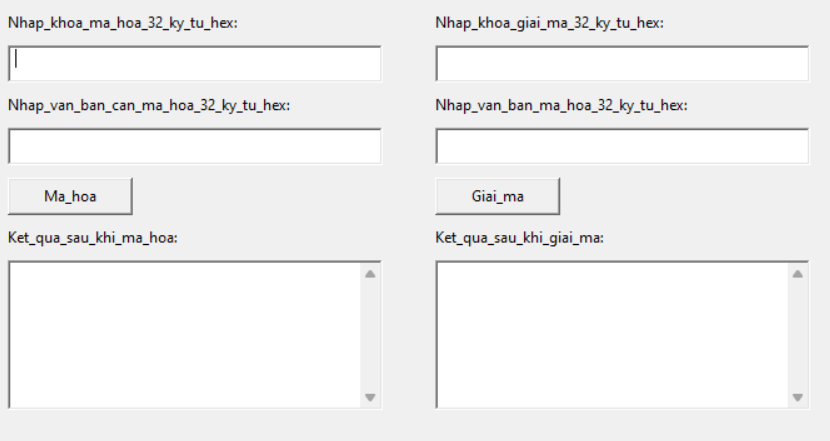
wxStaticText\* outputLabelDecrypt = new wxStaticText(panel, wxID\_ANY, "Ket\_qua\_sau\_khi\_giai\_ma:");

decryptSizer->Add(outputLabelDecrypt, 0, wxALL, 5);

m\_outputTextDecrypt = new wxTextCtrl(panel, wxID\_ANY, "", wxDefaultPosition, wxSize(300, 120), wxTE\_MULTILINE | wxTE\_READONLY);

decryptSizer->Add(m\_outputTextDecrypt, 0, wxALL, 5);

* **Giao diện**

****

**2.3.2 Ngôn ngữ C#**

* **Hàm sinh khóa ngẫu nhiên theo không gian khóa**

private string GenerateRandomKey(int keySizeInBits)

{

int keySizeInBytes = keySizeInBits / 8;

using (var rng = new RNGCryptoServiceProvider())

{

byte[] key = new byte[keySizeInBytes];

rng.GetBytes(key);

return Convert.ToBase64String(key);

}

}

* **Hàm đọc dữ liệu từ file và điền dữ liệu vào textbox**

private void ReadFile(string fileName, TextBox name)

{

using (StreamReader reader = new StreamReader(fileName))

{

StringBuilder res = new StringBuilder();

string line;

while ((line = reader.ReadLine()) != null)

{

res.Append(line);

}

if (String.IsNullOrEmpty(res.ToString()))

{

MessageBox.Show("File " + fileName + " không có dữ liệu", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

name.Text = res.ToString();

}

}

}

* Hàm ghi dữ liệu vào file

private void WriteFile(string fileName, string content)

{

string filePath = fileName;

using (StreamWriter writer = new StreamWriter(filePath))

{

writer.WriteLine(content);

}

MessageBox.Show("Ghi file thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

* **Hàm mã hóa dữ liệu AES và điền dữ liệu vào các textbox và xử lý ngoại lệ trong quá trình mã hóa**

private void AES\_Encrypt()

{

var inputText = mess = txtemess.Text.Trim();

var key = txtekey.Text.Trim();

if (String.IsNullOrWhiteSpace(inputText))

{

MessageBox.Show("Bản mã không được để trống", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (cbo.SelectedIndex == 0)

{

if (String.IsNullOrEmpty(key))

{

key = GenerateRandomKey(128);

txtekey.Text = key;

}

try

{

string EncryptText = Rijndael.Encrypt(inputText, key, KeySize.Aes128);

txtecipher.Text = EncryptText;

MessageBox.Show("Mã hóa dữ liệu thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch

{

MessageBox.Show("Có lỗi xảy ra trong quá trình mã hóa dữ liệu", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else if (cbo.SelectedIndex == 1)

{

if (String.IsNullOrEmpty(key))

{

key = GenerateRandomKey(192);

txtekey.Text = key;

}

try

{

string EncryptText = Rijndael.Encrypt(inputText, key, KeySize.Aes192);

txtecipher.Text = EncryptText;

MessageBox.Show("Mã hóa dữ liệu thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch

{

MessageBox.Show("Có lỗi xảy ra trong quá trình mã hóa dữ liệu", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else if (cbo.SelectedIndex == 2)

{

if (String.IsNullOrEmpty(key))

{

key = GenerateRandomKey(256);

txtekey.Text = key;

}

try

{

string EncryptText = Rijndael.Encrypt(inputText, key, KeySize.Aes256);

txtecipher.Text = EncryptText;

MessageBox.Show("Mã hóa dữ liệu thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch

{

MessageBox.Show("Có lỗi xảy ra trong quá trình mã hóa dữ liệu", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Có lỗi xảy ra trong quá trình chọn không gian khóa", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

* **Hàm giải mã dữ liệu AES và điền dữ liệu vào các textbox và xử lý ngoại lệ trong quá trình mã hóa**

private void AES\_Decrypt()

{

try

{

var inputText = txtdcipher.Text.Trim();

string key = txtdk.Text.Trim();

if (String.IsNullOrWhiteSpace(inputText))

{

MessageBox.Show("Bản mã không được để trống", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

else if (String.IsNullOrWhiteSpace(key))

{

MessageBox.Show("Khóa không được để trống", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string DecryptText;

if (cbo.SelectedIndex == 0)

{

DecryptText = Rijndael.Decrypt(inputText, key, KeySize.Aes128);

}else if(cbo.SelectedIndex == 1)

{

DecryptText = Rijndael.Decrypt(inputText, key, KeySize.Aes192);

}

else if(cbo.SelectedIndex == 2)

{

DecryptText = Rijndael.Decrypt(inputText, key, KeySize.Aes256);

}

else

{

MessageBox.Show("Có lỗi xảy ra trong quá trình chọn không gian khóa", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (DecryptText.Length > 0 && !DecryptText.Equals(txtemess.Text.Trim()))

{

MessageBox.Show("Bản mã đã bị sửa đổi", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

txtdmess.Text = DecryptText;

MessageBox.Show("Giải mã thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (CryptographicException ex)

{

if (ex.Message.Contains("Padding is invalid and cannot be removed"))

{

MessageBox.Show("Khóa đầu vào không hợp lệ hoặc đã bị sửa đổi", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Bản mã bị thay đổi", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception)

{

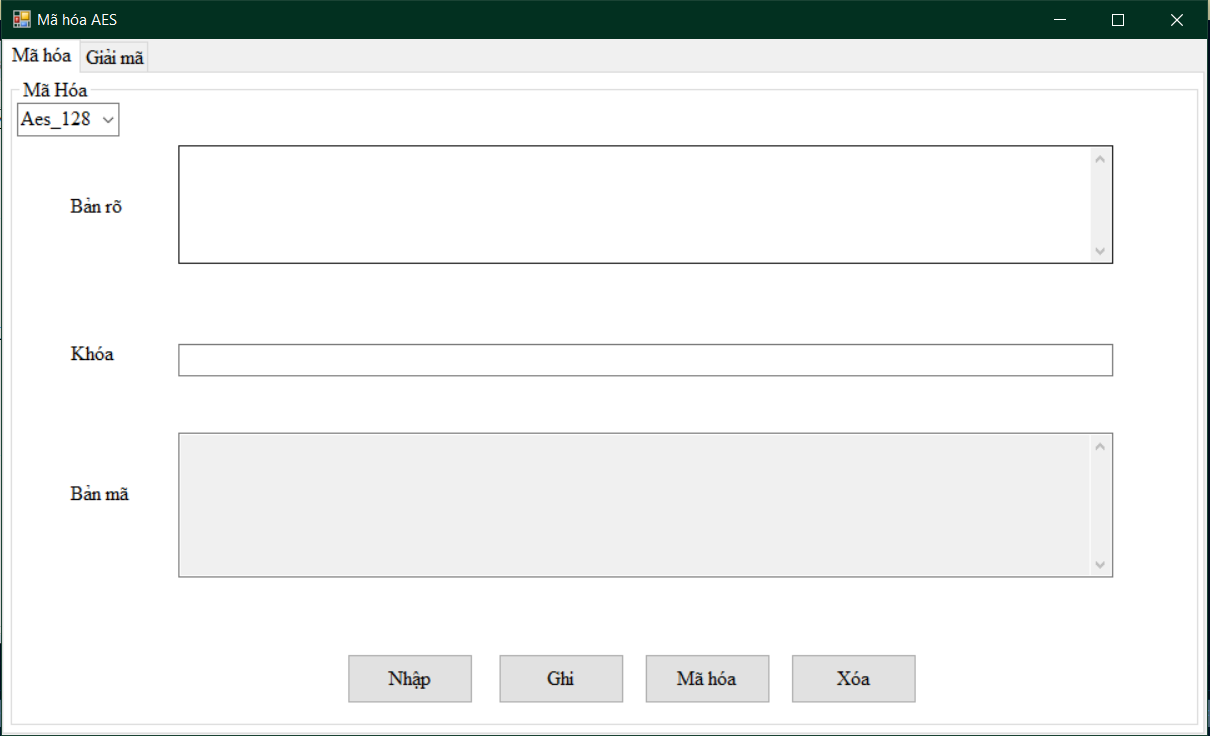
MessageBox.Show("Bản mã bị thay đổi", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

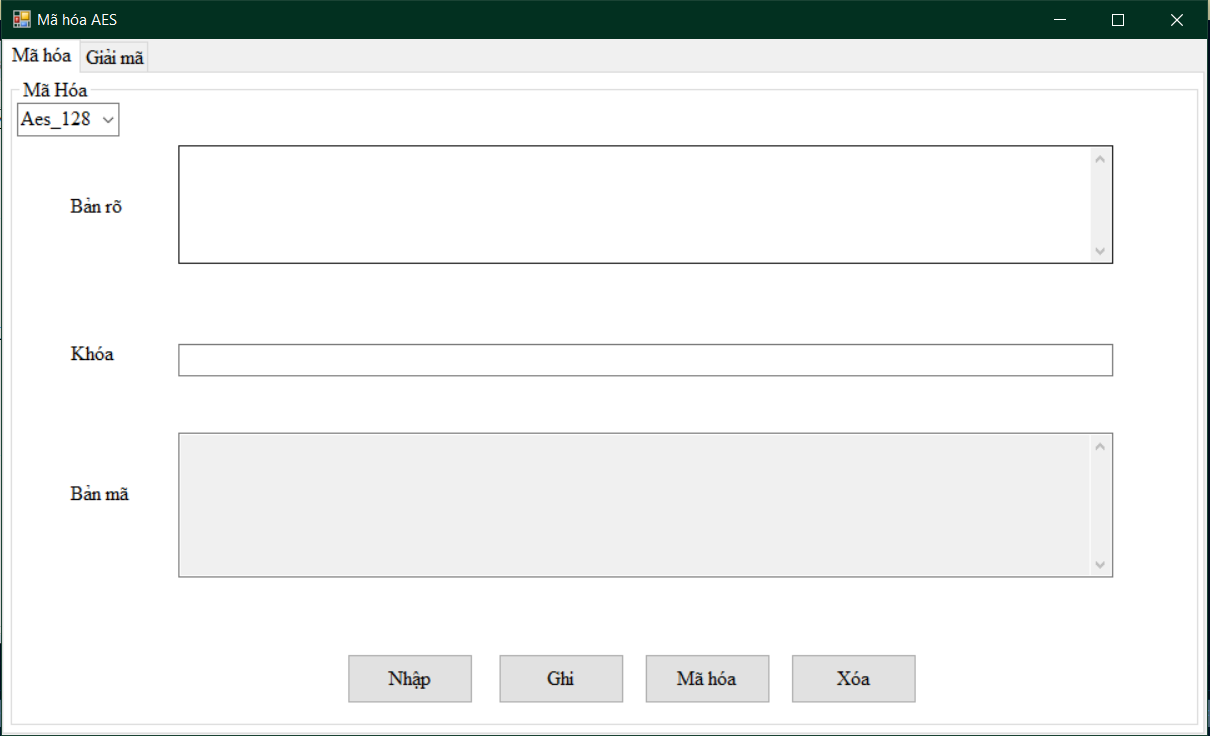
**Giao diện chương trình**

Giao diện khi khởi động chương trình



* Chương trình bao gồm 2 TabControl ứng với mỗi tab là một chức năng khác nhau:
  + Tab mã hóa thực hiện quá trình mã hóa dữ liệu
  + Tab mã hóa thực hiện quá trình giải mã dữ liệu

Giao diện tab Mã hóa



Trong tab mã hóa, người dùng có thể lựa chọn không gian khóa sẽ được sử dụng từ hộp thoại combobox dưới tiêu đề, có 3 sự lựa chọn: khóa 128bit, 192bit và 256bit

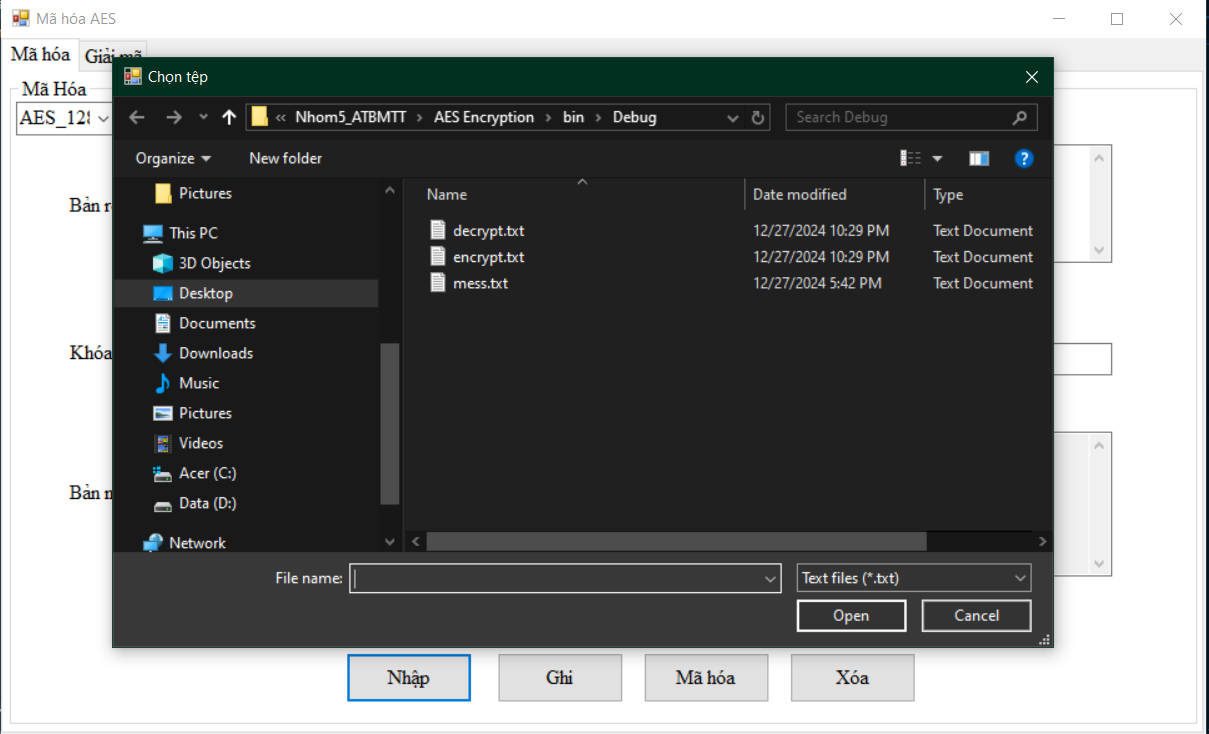
Hộp Textbox Bản rõ, cho phép nhập vào dữ liệu cần mã hóa, có thể nhập thủ công hoặc nhập dữ liệu từ file (sẽ được đề cập sau).

Hộp Textbox Khóa, cho phép nhập vào khóa theo độ dài theo giá trị được lựa chọn ở combobox, hoặc chương trình sẽ tự sinh khóa nếu người dùng nhập không đủ độ dài khóa theo giá trị combobox

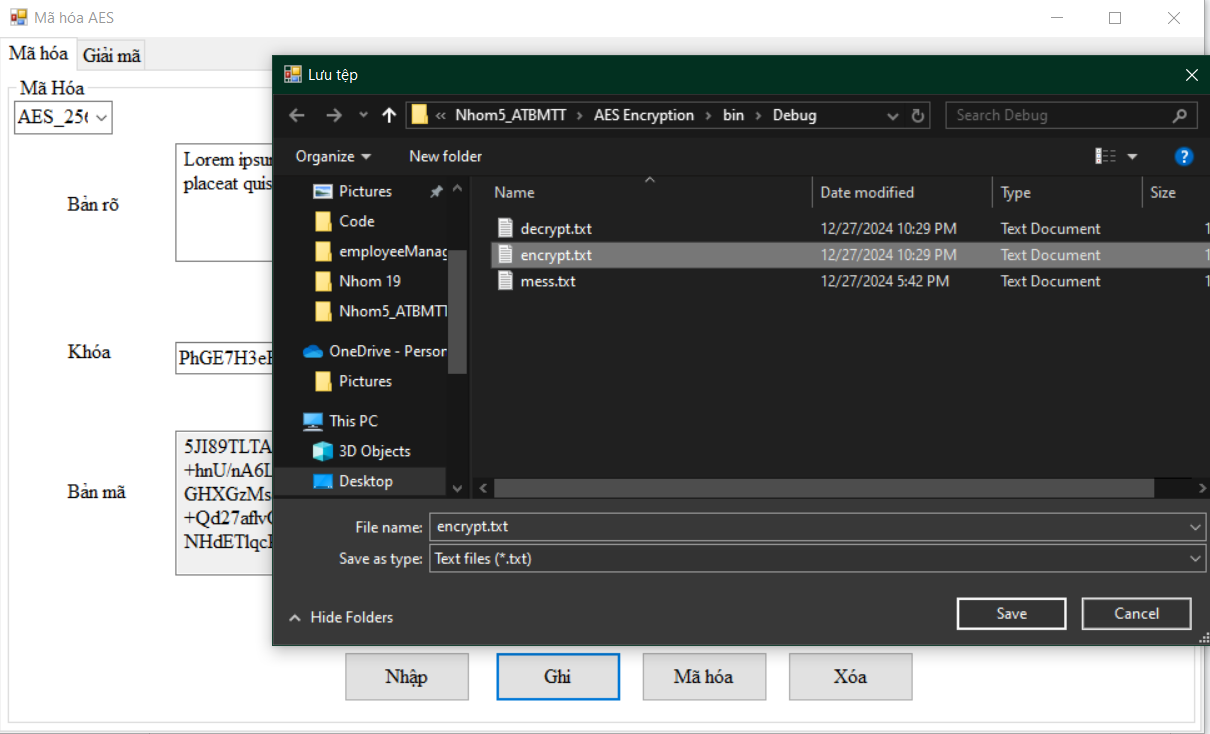
Hộp Textbox Bản mã sẽ chứa kết quả nội dung mã hóa, với dữ liệu sử dụng trong quá trình mã hóa từ hộp Textbox bản rõ và hộp Textbox Khóa

Button Nhập cho phép đọc dữ liệu cần được mã hóa từ một file txt và hiển thị lên Textbox Bản rõ

với giao diện dưới đây:



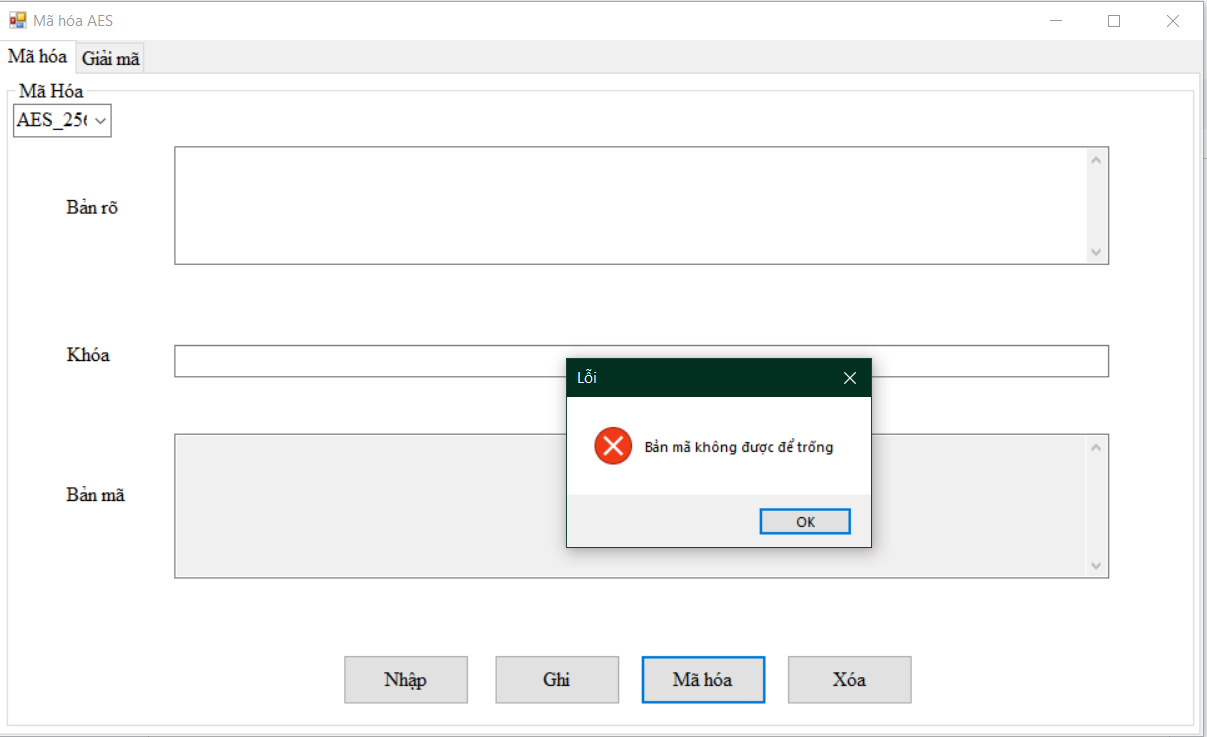
Button Ghi có chức năng ghi dữ liệu đã mã hóa từ hộp Textbox Bản mã và một file được lựa chọn để ghi, hộp thoại lựa chọn file có giao diện như dưới đây:



Button Mã hóa sẽ thực hiện mã hóa và trả về dữ liệu mã hóa và ghi dữ liệu mã hóa lên hộp Textbox Bản mã

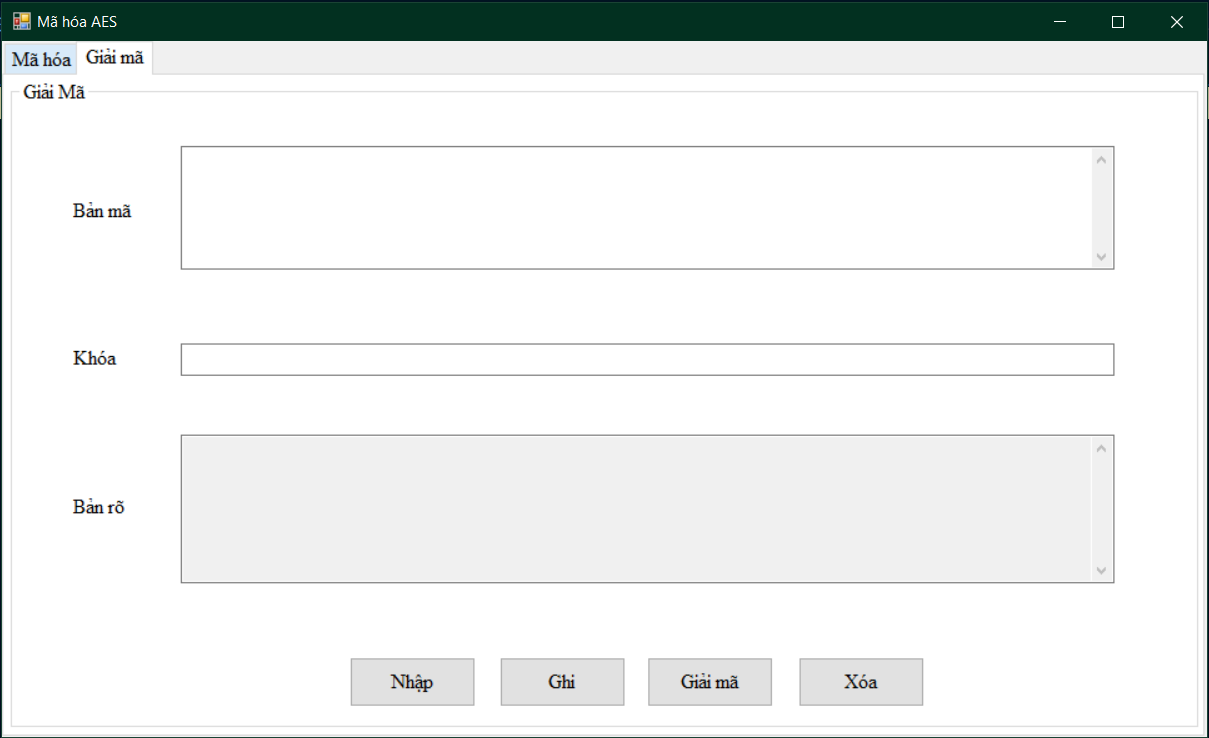
Button Xóa sẽ xóa dữ liệu ba hộp Textbox Bản rõ, Khóa, Bản mã, trả về trạng thái ban đầu khi chưa nhập dữ liệu

**Các ngoại lệ được gặp trong quá trình mã hóa dữ liệu AES**



Yêu cầu hộp Textbox Bản mã không được để trống hoặc dữ liệu nhập vào không được là khoảng trống trước khi mã hóa

Giao diện tab Giải mã



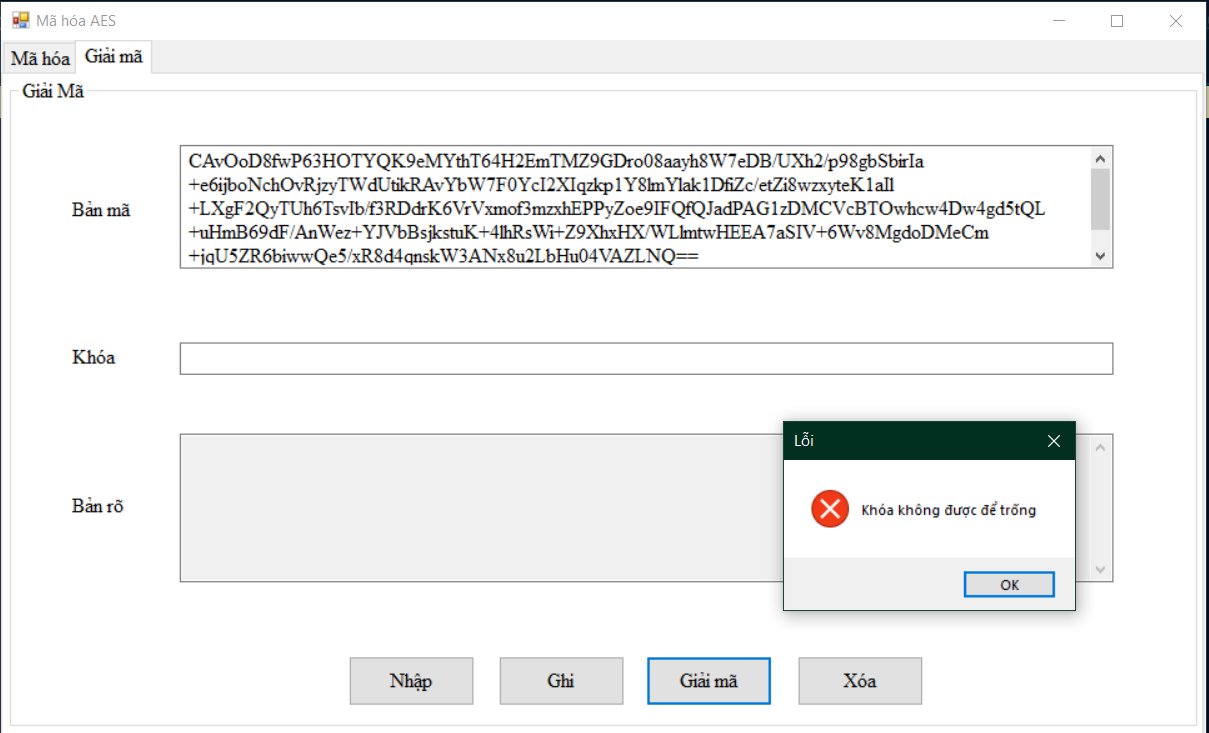
Hộp Textbox Bản mã, cho phép nhập vào dữ liệu cần được giải mã hóa, có thể nhập thủ công hoặc nhập dữ liệu từ file.

Hộp Textbox Khóa, cho phép nhập vào khóa đã được sử dụng để mã hóa bản mã của hộp Textbox phía trên

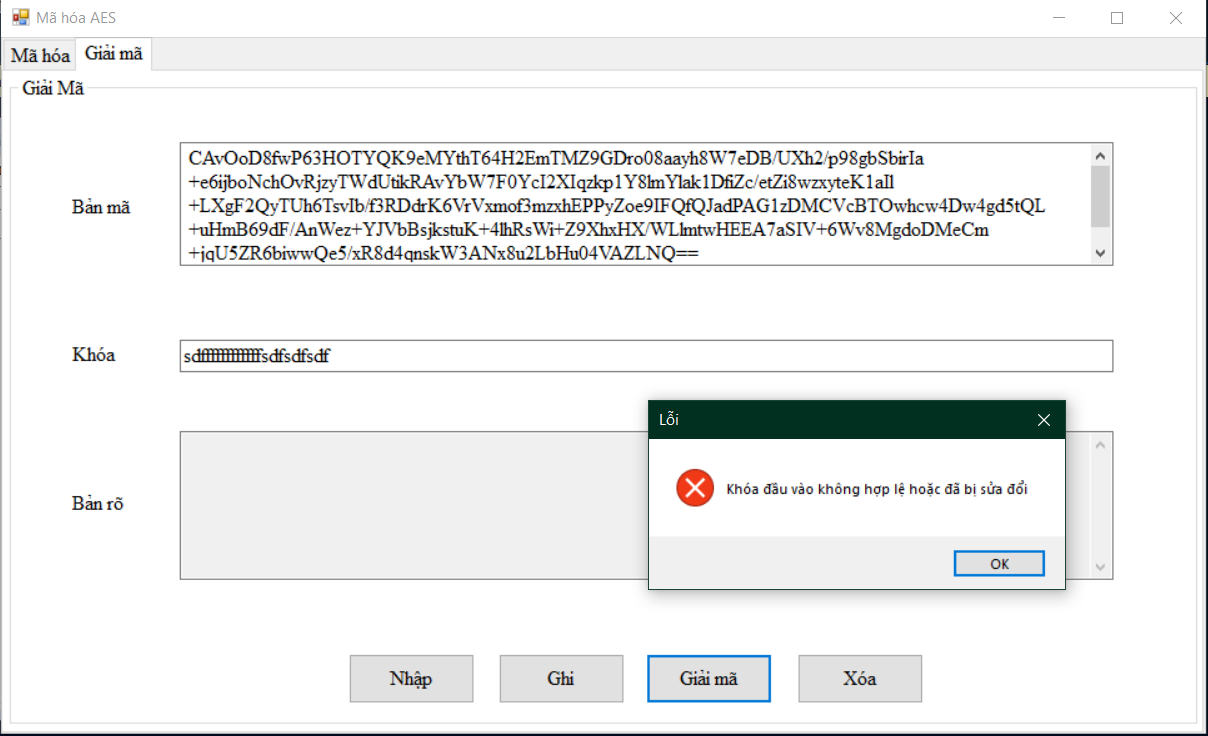
Hộp Textbox Bản rõ sẽ chứa kết quả nội dung giải mã hóa, với dữ liệu sử dụng trong quá trình giải mã từ hộp Textbox bản mã và hộp Textbox Khóa

Những Button còn lại trong tab Giải Mã có chức năng tương tự với tab Mã Hóa, với dữ liệu đọc từ file chứa dữ liệu mã hóa, và ghi dữ liệu giải mã.

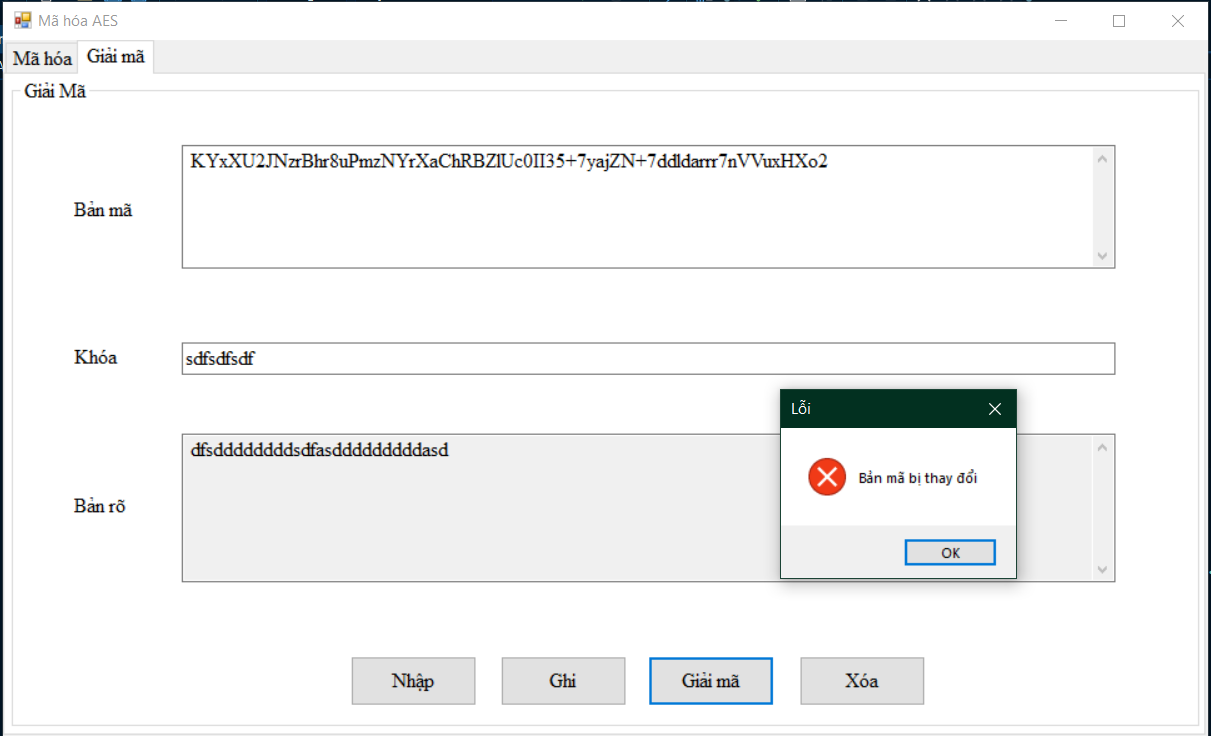
**Những ngoại lệ có thể gặp trong quá trình giải mã**

****

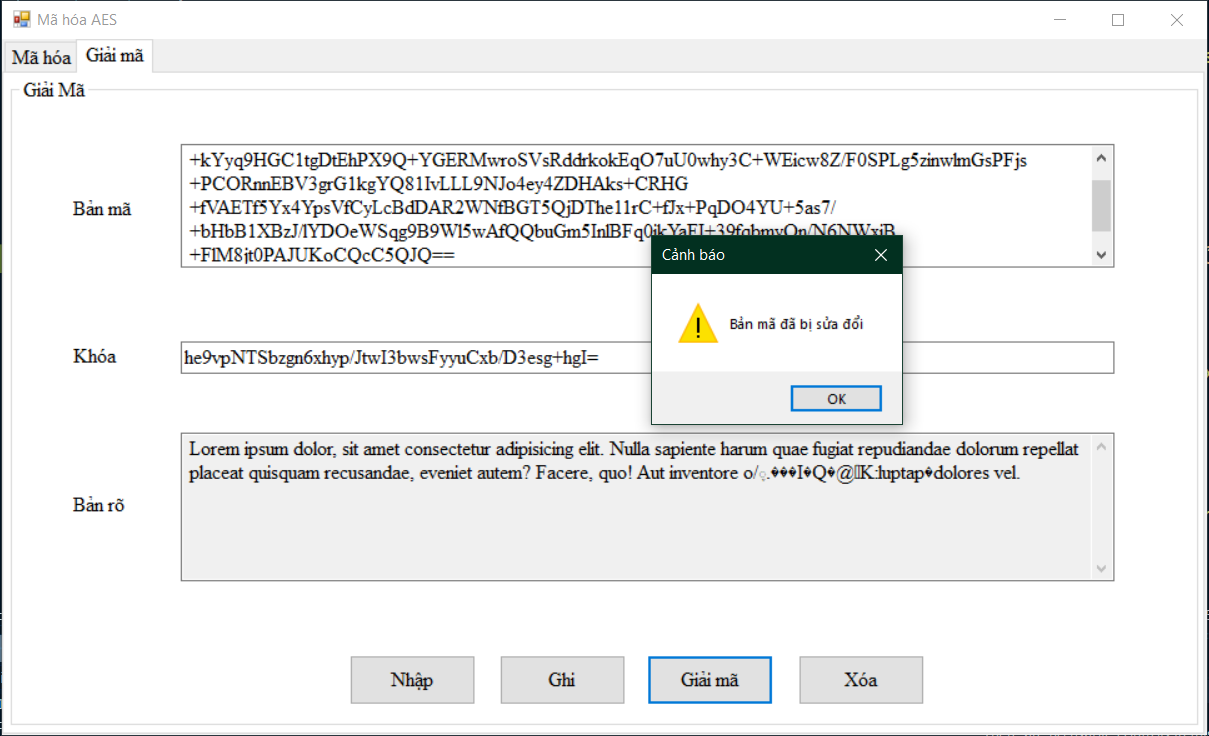
Ngoại lệ này sẽ được gặp nếu không nhập vào khóa, hoặc khóa là một khoảng trống

****

Ngoại lệ này sẽ được gặp khi khóa nhập vào bị thay đổi trước đó, khóa giải mã có thể thêm hoặc bớt bất kì kí tự nào từ giá trị của khóa gốc sau quá trình mã hóa trước đó,lỗi này sẽ xảy ra khi chương trình thực hiện mã hóa dữ liệu sử dụng một trong hai bản mã hoặc khóa có kích thước không phù hợp khi mã hóa



Ngoại lệ này sẽ được gặp nếu bản mã bị thay đổi bỏ bớt, thêm hoặc sửa bất kí tự từ bản mã hóa, lỗi này sẽ xảy ra khi chương trình thực hiện mã hóa dữ liệu sử dụng một trong hai bản mã hoặc khóa có kích thước không phù hợp khi mã hóa



Ngoại lệ này sẽ được gặp nếu bản mã đã bị sửa đổi 1 hay nhiều kí tự, quá trình giải mã vẫn sẽ thành công nhưng khi giải mã hóa kết quả bản mã không trùng khớp với bản rõ đã được mã hóa trước đó, chương trình sẽ xuất hiện hộp thoại cảnh báo

## CHƯƠNG 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

**3.1 Kiến thức kỹ năng đã học được trong quá trình thực hiện đề tài.**

Tinh thần hợp tác: Khuyến khích tinh thần làm việc nhóm, lắng nghe và tôn trọng ý kiến của  nhau. Hỗ trợ nhau trong công việc

Thống nhất mục tiêu : Đảm bảo rằng mọi thành viên đều đồng thuận với mục tiêu chung của bài tập và điều chỉnh, bàn luận khi cần thiết

Lập kế hoạch và  phân công công việc : Lập kế hoạch và phân công nhiệm vụ thích hợp cho các thành viên trong nhóm từ đó thực hiện hiệu quả , nhanh chóng, chính xác hơn

Quản lý thời gian : đặt ra thời hạn cho các công việc từ đó giảm thiểu tối đa các rủi ro ảnh hưởng đến những phần được giao và đảm bảo tiến độ công việc được hoàn thành đúng thời hạn

Khả năng giao tiếp và thích ứng :

* Tăng khả năng giao tiếp và thích ứng khi được vào các nhóm ngẫu nhiên và không quen biết nhau rèn luyện hiệu quả khi tăng khả năng hòa nhập và công bằng khi làm việc. Ngoài ra còn giúp đỡ nhau mỗi khi có thành viên thắc mắc.
* Thiết lập các kênh giao tiếp rõ ràng và thường xuyên đảm bảo mọi người có thể nắm rõ tiến độ và phân công đúng công việc của mình.

  Phát triển kỹ năng đọc hiểu và chọn lọc thông tin ở nhiều nguồn khác nhau : Chọn lọc các kiến thức và bài học được đăng trên mạng.

**3.2 Bài học kinh nghiệm**

Sau thời gian thực hiện đề tài xây dựng chương trình mã hóa và giải mã AES, nhóm chúng em đã hoàn thiện chương trình theo đúng tiến độ và tích lũy được các kỹ năng sơ bộ như:

* Hiểu rõ cấu trúc và hoạt động của thuật toán AES: Nắm bắt được quy trình mã hóa và giải mã của AES, cũng như các bước xử lý trong mỗi vòng.
* Phân tích bài toán và yêu cầu: Tìm hiểu kỹ lưỡng về bài toán đặt ra, xác định các chức năng cần có của chương trình, đặc biệt trong việc bảo mật thông tin và xác thực người dùng.
* Thiết kế chức năng chương trình: Thiết kế các chức năng của chương trình sao cho đáp ứng đầy đủ yêu cầu của bài toán, đảm bảo tính hiệu quả và bảo mật.
* Sử dụng thành thạo ngôn ngữ lập trình: Giúp các thành viên trong nhóm hiểu thêm và sử dụng thành thạo các ngôn ngữ như C++ và C#.
* Kỹ năng kiểm tra và sửa lỗi: Rèn luyện kỹ năng kiểm tra lỗi và sửa lỗi trong quá trình thực hiện, từ việc phát hiện lỗi đến việc khắc phục.
* Kỹ năng làm việc nhóm: Cải thiện khả năng làm việc nhóm hiệu quả, biết cách phân công công việc và hỗ trợ lẫn nhau trong quá trình phát triển.
* Kỹ năng trao đổi và phản biện: Học cách trao đổi ý kiến, đánh giá và phản biện trong nhóm, nâng cao khả năng giao tiếp và hợp tác.
* Kinh nghiệm tham gia dự án thực tế: Có thêm kinh nghiệm quý giá khi tham gia vào một dự án thực tế, giúp áp dụng kiến thức lý thuyết vào thực tiễn.
* Tích lũy kỹ năng lập trình: Tích lũy được các kỹ năng lập trình cần thiết cho bản thân, phục vụ cho các dự án tiếp theo trong tương lai.

**3.3 Đề xuất về tính khả thi của chủ đề nghiên cứu, những thuận lợi, khó khăn**

Tính khả thi: Chủ đề nghiên cứu về mã hóa và giải mã AES là rất khả thi và có ý nghĩa thực tiễn cao, đặc biệt trong bối cảnh bảo mật thông tin ngày càng trở nên quan trọng. Việc xây dựng chương trình mã hóa giúp hiểu rõ hơn về các nguyên lý bảo mật và ứng dụng của chúng trong thực tế.

Thuận lợi:

* Tài liệu phong phú: Có nhiều tài liệu và nguồn học liệu về AES và lập trình C++, C# giúp việc nghiên cứu và thực hiện đề tài dễ dàng hơn.
* Cộng đồng hỗ trợ: Cộng đồng lập trình viên rộng lớn giúp tôi dễ dàng tìm kiếm sự hỗ trợ và giải đáp thắc mắc khi gặp khó khăn.

Khó khăn:

* Độ phức tạp của thuật toán: AES có cấu trúc và quy trình xử lý phức tạp, yêu cầu kiến thức sâu về toán học và lập trình để triển khai thành công.
* Quản lý thời gian: Việc cân bằng giữa việc học, thực hiện dự án và các môn học khác đôi khi gây khó khăn trong việc đạt tiến độ.
* Còn hạn chế về kiến thức chuyên môn
* Kỹ năng làm bài tập nhóm còn hạn chế cần phải cải thiện
* Kỹ năng về lập trình còn hạn chế
* Kỹ năng trình bày bằng văn bản còn hạn chế

**TÀI LIỆU THAM KHẢO**

(Sắp xếp theo ngôn ngữ: Việt, Anh, Pháp, Đức, Nga…)

[1]. Nguyễn Xuân Dũng, *Bảo mật thông tin – Mô Hình và ứng dụng*, NXB thống kê, 2009.

[2]. Bùi Doãn Khanh, Nguyễn Đình Thúc, *Mã hóa thông tin – Lý thuyết và ứng dụng*, NXB Lao động xã hội, 2011.

[3]. William Stallings**,** *Cryptography and Network Security Principles and Practices***,** Fourth Edition, Prentice Hall, 2005.