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 hóa và giải mã DES**

.

Giáo viên hướng dẫn : TS. Phạm Văn Hiệp

Nhóm sinh viên thực hiện :

1. Nguyễn Thị Hà Ánh Mã SV: 2021602112

2. Nguyễn Hoàng Hải Mã SV: 2021602114

3. Phan Thùy Linh Mã SV: 2021601685

4. Trần Đặng Phương Thảo Mã SV: 2021601876

5. Lại Thế Vinh Mã SV: 2022600268

Mã Lớp học phần: 20232IT6001005

Nhóm: 9

Hà Nội - Năm 2024

**MỤC LỤC**

[**LỜI MỞ ĐẦU 3**](#)

[CHƯƠNG 1: TỔNG QUAN 4](#_30j0zll)

[1.1. Lý do chọn đề tài 4](#_1fob9te)

[1.2. Tổng quan về An toàn bảo mật thông tin 5](#_3znysh7)

[1.2.1. Sự cần thiết của việc đảm bảo an toàn thông tin 5](#_f5zd8g7dgpfm)

[1.2.2. Khái niệm an toàn thông tin 6](#_z7iiluetccwp)

[1.2.3. Các mối đe doạ đối với một hệ thống và các biện pháp ngăn chặn 7](#_1v3eszlpp337)

[1.2.4. Các mức độ bảo vệ thông tin 8](#_c67ownalqw0a)

[1.3. Các kiến thức cơ sở 9](#_2et92p0)

[1.4. Nội dung nghiên cứu 10](#_2qzstqkqauwn)

[CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU 11](#_3dy6vkm)

[2.1. Nghiên cứu, tìm hiểu hệ mã hóa khóa bí mật 11](#_1t3h5sf)

[2.1.1. Giới thiệu 11](#_4d34og8)

[2.1.2. Khái niệm về mã hóa khóa bí mật 12](#_2s8eyo1)

[2.1.3. Các hệ mã khoá bí mật phổ biến 12](#_17dp8vu)

[2.1.4. Ứng dụng của khóa bí mật 13](#_3rdcrjn)

[2.2. Nghiên cứu, tìm hiểu về chuẩn mã dữ liệu DES 14](#_26in1rg)

[2.2.1. Giới thiệu 14](#_lnxbz9)

[2.2.2. Đặc điểm của thuật toán DES 16](#_35nkun2)

[2.2.3. Ưu điểm và nhược điểm của thuật toán DES 16](#_1ksv4uv)

[2.2.4. Thuật toán DES 18](#_44sinio)

[2.3. Thiết kế chương trình, cài đặt thuật toán 33](#_qsh70q)

[2.3.1. Thiết kế kịch bản chương trình 33](#_hlindpvkbi8i)

[2.3.2. Giới thiệu ngôn ngữ lập trình sử dụng để cài đặt thuật toán. 35](#_qc6rgi8ksjsj)

[2.3.3. Thiết kế, cài đặt chương trình 38](#_dqs46ws38y6s)

[CHƯƠNG 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 65](#_3as4poj)

[3.1. Kiến thức kỹ năng đã học được trong quá trình thực hiện đề tài. 65](#_1pxezwc)

[3.1.1. Kiến thức lý thuyết 65](#_49x2ik5)

[3.1.2. Kỹ năng lập trình 65](#_2p2csry)

[3.1.3. Kỹ năng nghiên cứu, phân tích và làm việc nhóm 65](#_147n2zr)

[3.2. Bài học kinh nghiệm 66](#_3o7alnk)

[3.3. Đề xuất về tính khả thi của chủ đề nghiên cứu, những thuận lợi, khó khăn 66](#_23ckvvd)

[**TÀI LIỆU THAM KHẢO 68**](#_s6l2yy8xaxam)

# LỜI MỞ ĐẦU

Trước đây khi công nghệ máy tính chưa phát triển, khi nói đến vấn đề an toàn bảo mật thông tin, chúng ta thường hay nghĩ đến các biện pháp nhằm đảm bảo cho thông tin được trao đổi hay cất giữ một cách an toàn và bí mật, chẳng hạn là các biện pháp như: Đóng dấu và ký niêm phong một bức thư để biết rằng lá thư có được chuyển nguyên vẹn đến người nhận hay không, dùng mật mã mã hóa thông điệp để chỉ có người gửi và người nhận hiểu được thông điệp, lưu giữ tài liệu trong các két sắt có khóa tại nơi được bảo vệ nghiêm ngặt. Ngày nay với sự phát triển của khoa học công nghệ, đặt biệt là sự phát triển của Internet, việc sử dụng máy tính và điện thoại cá nhân càng trở lên rộng rãi, dẫn đến càng nhiều thông tin được lưu trữ trên máy tính và gửi đi trên mạng Internet. Do đó nhu cầu về an toàn và bảo mật thông tin trên máy tính càng nhiều và việc sử dụng mật mã mã hóa càng được phổ biến.

Trong báo cáo này, em thực hiện **Xây dựng chương trình mã hóa và giải mã DES**. Báo cáo gồm 3 chương:

- Chương 1: Tổng quan

- Chương 2: Kết quả nghiên cứu

- Chương 3: Kết luận và bài học kinh nghiệm

Nhóm chúng em xin cám ơn thầy Phạm Văn Hiệp đã trực tiếp tận tình hướng dẫn cũng như cung cấp tài liệu thông tin khoa học cần thiết cho nhóm chúng em trong suốt quá trình thực hiện đề tài này.

# CHƯƠNG 1: TỔNG QUAN

## 1.1. Lý do chọn đề tài

Với sự phát triển của Internet như hiện nay, thông tin trên mạng mang nhiều nguy cơ bị tấn công và xâm nhập bất hợp pháp, việc bảo mật thông tin trên môi trường Internet trở nên vô cùng cấp thiết. Để vừa đảm bảo thông tin luôn được kết nối, vừa đảm bảo yếu tố bảo mật, phương pháp tốt nhất là mã hóa thông tin. Mã hóa giúp đảm bảo thông tin không thể bị truy cập bởi đối tượng thứ ba, loại trừ việc nghe trộm những thông tin “nhạy cảm” khi nó được truyền qua Internet. Khi dữ liệu được mã hóa, nó sẽ không thể bị đọc được bởi những người ngoài người gửi và người nhận, điều này đảm bảo tính bảo mật và toàn vẹn của dữ liệu.

Chính sự quan trọng của mã hóa thông tin đã thúc đẩy chúng em chọn đề tài này để tìm hiểu rõ hơn về các giải thuật mã hóa hiện nay và cơ chế mã hóa thông tin trên mạng. Trong số nhiều giải thuật mã hóa, chúng em quyết định tìm hiểu và thực hiện chương trình mã hóa và giải mã bằng thuật toán DES (Data Encryption Standard). DES là một trong những thuật toán mã hóa đối xứng cổ điển và đã từng được sử dụng rộng rãi trong nhiều ứng dụng bảo mật thông tin. Việc hiểu rõ và thực hiện DES giúp chúng em nắm vững nguyên lý hoạt động của một trong những thuật toán nền tảng trong lĩnh vực mật mã học.

DES là một thuật toán có cấu trúc rõ ràng và đã được nghiên cứu kỹ lưỡng, cung cấp nền tảng kiến thức quan trọng về mã hóa khối, hoán vị và thay thế, các yếu tố quan trọng trong nhiều thuật toán mã hóa hiện đại khác. Việc nghiên cứu DES sẽ giúp chúng em có cái nhìn sâu hơn về các nguyên lý cơ bản của mã hóa. Mặc dù DES đã bị thay thế bởi các thuật toán mạnh hơn như AES (Advanced Encryption Standard), việc thực hiện mã hóa và giải mã bằng DES vẫn đòi hỏi sự hiểu biết sâu sắc về các khái niệm như hoán vị, khóa phụ, và chuỗi Feistel, tạo cơ hội cho chúng em rèn luyện kỹ năng lập trình và tư duy thuật toán.

Hiểu rõ cách thức hoạt động của DES không chỉ giúp chúng em trong việc nghiên cứu học thuật mà còn có thể áp dụng trong các bài toán thực tế yêu cầu bảo mật dữ liệu. Mặc dù không còn được coi là an toàn cho nhiều ứng dụng hiện đại, DES vẫn có giá trị trong việc giáo dục và minh họa các nguyên tắc cơ bản của mã hóa dữ liệu. Bằng cách chọn đề tài "***Xây dựng chương trình mã hóa và giải mã DES***", chúng em mong muốn không chỉ hiểu rõ hơn về lý thuyết và cấu trúc của một thuật toán mã hóa cụ thể mà còn nắm bắt được tầm quan trọng của bảo mật thông tin trong kỷ nguyên số. Qua đó, chúng em hy vọng có thể áp dụng kiến thức này vào các nghiên cứu và dự án thực tế trong tương lai, đóng góp vào việc bảo mật thông tin trong môi trường mạng ngày càng phức tạp và tiềm ẩn nhiều rủi ro.

## 1.2. Tổng quan về An toàn bảo mật thông tin

### 1.2.1. Sự cần thiết của việc đảm bảo an toàn thông tin

Ngày nay sự xuất hiện của internet toàn cầu đã giúp cho việc trao đổi thông tin trở nên nhanh gọn, dễ dàng. Các phương thức chia sẻ dữ liệu qua mạng làm cho việc trao đổi, mua bán, chuyển tiền, … diễn ra mỗi ngày trên nền tảng số.

Tuy nhiên vấn đề mới lại phát sinh. Những thông tin đang nằm ở kho dữ liệu hay đang được truyền đi có thể bị trộm cắp, bị làm sai lệch, giả mạo. Điều này làm ảnh hưởng đến độ an toàn của thông tin nhạy cảm, tin mật, … có thể tác động lớn đến nhiều cá nhân, tổ chức, hay ác động đến an ninh quốc gia.

### 1.2.2. Khái niệm an toàn thông tin

1. Khái niệm

Định nghĩa của an toàn thông tin được nêu ra từ nhiều nguồn khác nhau, chúng ta có thể hiểu theo nhiều cách sau: *An toàn thông tin nghĩa là thông tin được bảo vệ, các hệ thống và dịch vụ có khả năng chống lại những sự can thiệp, lỗi và những tai họa không mong đợi, các thay đổi tác động đến độ an toàn của hệ thống là nhỏ nhất. Hệ thống không an toàn là hệ thống tồn tại những điểm: thông tin bị rò rỉ ra ngoài, thông tin bị thay đổi,* ...

Giá trị thực sự của thông tin chỉ đạt được khi thông tin được cung cấp chính xác và kịp thời, hệ thống phải hoạt động chuẩn xác thì mới có thể đưa ra những thông tin có giá trị cao. *Mục tiêu của an toàn bảo mật trong công nghệ thông tin là đưa ra một số tiêu chuẩn an toàn và áp dụng các tiêu chuẩn an toàn này vào chỗ thích hợp để giảm bớt và loại trừ những nguy hiểm có thể xảy ra.* Ngày nay với kỹ thuật truyền nhận và xử lý thông tin ngày càng phát triển và phức tạp nên hệ thống chỉ có thể đạt tới một mức độ an toàn nào đó và không có một hệ thống an toàn tuyệt đối.

1. Các yêu cầu an toàn bảo mật thông tin

Ngày nay, với sự phát triển rất nhanh của khoa học công nghệ, các biện pháp tấn công ngày càng tinh xảo hơn, độ an toàn của thông tin có thể bị đe dọa từ nhiều nơi, theo nhiều cách khác nhau, chúng ta cần phải đưa ra các chính sách đề phòng thích hợp. Các yêu cầu cần thiết của việc bảo vệ thông tin và tài nguyên:

* *Đảm bảo bí mật (Bảo mật):* thông tin không bị lộ đối với người không được phép.
* *Đảm bảo tính tin cậy (Confidentiality):* Thông tin và tài nguyên không thể bị truy cập trái phép bởi những người không có quyền hạn.
* *Đảm bảo tính toàn vẹn (Integrity):* Thông tin và tài nguyên không thể bị sửa đổi, bị thay thế bởi những người không có quyền hạn.
* *Đảm bảo tính sẵn sàng (Availability):* Thông tin và tài nguyên luôn sẵn sàng để đáp ứng sử dụng cho người có quyền hạn.
* *Đảm bảo tính không thể chối bỏ (Non-repudiation):* Thông tin và tài nguyên được xác nhận về mặt pháp luật của người cung cấp.

### 1.2.3. Các mối đe doạ đối với một hệ thống và các biện pháp ngăn chặn

*Phá hoại:* kẻ thù phá hỏng thiết bị phần cứng hoặc phần mềm hoạt động trên hệ thống.

*Sửa đổi:* Tài sản của hệ thống bị sửa đổi trái phép. Điều này thường làm cho hệ thống không làm đúng chức năng của nó. Chẳng hạn như thay đổi mật khẩu, quyền người dùng trong hệ thống làm họ không thể truy cập vào hệ thống để làm việc.

*Can thiệp:* Tài sản bị truy cập bởi những người không có thẩm quyền. Các truyền thông thực hiện trên hệ thống bị ngăn chặn, sửa đổi.

***Các đe dọa đối với một hệ thống thông tin có thể đến từ ba loại đối tượng như sau:***

* Các đối tượng từ ngay bên trong hệ thống (insider), đây là những người có quyền truy cập hợp pháp đối với hệ thống.
* Những đối tượng bên ngoài hệ thống (hacker, cracker), thường các đối tượng này tấn công qua những đường kết nối với hệ thống như Internet chẳng hạn.
* Các phần mềm (chẳng hạn nhƣ spyware, adware …) chạy trên hệ thống.

***Các biện pháp ngăn chặn:***

* *Điều khiển thông qua phần mềm:* dựa vào các cơ chế an toàn bảo mật của hệ thống nền (hệ điều hành), các thuật toán mật mã học.
* *Điều khiển thông qua phần cứng:* các cơ chế bảo mật, các thuật toán mật mã học được cứng hóa để sử dụng.
* *Điều khiển thông qua các chính sách của tổ chức:* ban hành các qui định của tổ chức nhằm đảm bảo tính an toàn bảo mật của hệ thống.

### 1.2.4. Các mức độ bảo vệ thông tin

*Quyền truy nhập:* Là lớp bảo vệ trong cùng nhằm kiểm soát các tài nguyên của mạng và quyền hạn trên tài nguyên đó.

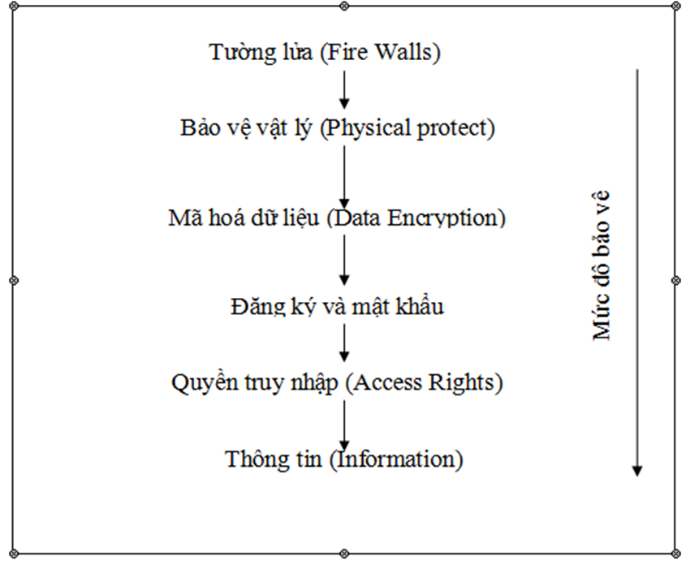
*Đăng ký tên /mật khẩu:* Thực ra đây cũng là kiểm soát quyền truy nhập, nhưng không phải truy nhập ở mức thông tin mà ở mức hệ thống.

*Mã hoá dữ liệu:* Dữ liệu bị biến đổi từ dạng nhận thức được sang dạng không nhận thức được theo một thuật toán nào đó và sẽ được biến đổi ngược lại ở trạm nhận (giải mã).

*Bảo vệ vật lý:* Ngăn cản các truy nhập vật lý vào hệ thống.

*Tường lửa:* Ngăn chặn thâm nhập trái phép và lọc bỏ các gói tin không muốn gửi hoặc nhận vì các lý do nào đó để bảo vệ một máy tính hoặc cả mạng nội bộ (intranet).

*Quản trị mạng:* Công tác quản trị mạng máy tính phải được thực hiện một cách khoa học. Toàn bộ hệ thống hoạt động bình thường trong giờ làm việc. Backup dữ liệu quan trọng theo định kỳ, có kế hoạch bảo trì định kỳ, bảo mật dữ liệu, phân quyền, …



*Ảnh 1.1: Các mức độ bảo vệ thông tin*

## 1.3. Các kiến thức cơ sở

**Thuật toán và định lý:**

* Nắm vững kiến thức về khái niệm, đặc điểm và ứng dụng của khóa bí mật
* Nắm được khái niệm, đặc điểm, ưu và nhược điểm của thuật toán DES
* Nắm được các phép toán module và phép toán XOR
  + Nắm được cấu trúc, quy trình tạo khóa, quy trình mã hóa và giải mã của thuật toán DES
  + Nắm được nguyên lý Feistel, quá trình hoán vị và thay thế

**Ngôn ngữ lập trình:**

* Ngôn ngữ C++: hiểu về cú pháp và cách sử dụng các cấu trúc dữ liệu cơ bản, lớp và đối tượng, thư viện chuẩn C++, và quy trình biên dịch và chạy chương trình.
* Ngôn ngữ C#: hiểu về cú pháp của C# và các khái niệm như lớp, đối tượng, và thuộc tính. Nắm vững các thư viện chuẩn của .NET Framework hoặc .NET Core và cách sử dụng chúng để phát triển ứng dụng.

**Kiến thức về sử dụng các công cụ lập trình**

* Đối với C++ : Visual studio 2022/ Dev C++
* Đối với C# : Visual studio 2022

## 1.4. Nội dung nghiên cứu

**Tìm hiểu về khóa bí mật**

* Khái niệm
* Các khóa bí mật phổ biến
* Ứng dụng

**Tìm hiểu về DES**

● Khái niệm

● Lĩnh vực hoạt động

● Một số ưu nhược điểm của thuật toán

**Thuật toán DES**

* Quá trình mã hóa
* Quá trình giải mã
* Quy trình tạo khóa
* Hàm Feistel, S-Box
* Ví dụ minh họa

**Áp dụng thực hiện xây dựng chương trình với các ngôn ngữ C++ và C**

# CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU

## 2.1. Nghiên cứu, tìm hiểu hệ mã hóa khóa bí mật

### 2.1.1. Giới thiệu

Trong thế giới kỹ thuật số ngày nay, bảo mật thông tin là vô cùng quan trọng. Một cách chính để đảm bảo an toàn cho dữ liệu là sử dụng các thuật toán mã hóa. Bài tiểu luận này tập trung vào hệ mã hóa khóa bí mật, hay còn gọi là mã hóa đối xứng. Chúng ta sẽ nghiên cứu về các thuật toán phổ biến, ứng dụng.

Hệ mã hóa khóa bí mật sử dụng cùng một khóa để mã hóa và giải mã dữ liệu. Các thuật toán như AES và DES là những ví dụ phổ biến. Ứng dụng của hệ này rất đa dạng, từ bảo mật thông tin truyền qua Internet đến bảo vệ dữ liệu lưu trữ và hệ thống thanh toán trực tuyến.

### 2.1.2. Khái niệm về mã hóa khóa bí mật

**Hệ mã hóa khóa bí mật** là một phương pháp mã hóa dữ liệu trong đó cùng một khóa được sử dụng cho cả quá trình mã hóa và giải mã. Điều này có nghĩa là người gửi và người nhận phải chia sẻ cùng một khóa bí mật và giữ nó an toàn để đảm bảo bảo mật thông tin.

* **Khóa bí mật (Secret Key)**: Chuỗi ký tự, số hoặc ký tự đặc biệt được giữ bí mật giữa các bên truyền thông.
* **Mã hóa (Encryption)**: Quá trình chuyển đổi dữ liệu từ dạng có thể đọc được (plaintext) sang dạng không thể đọc được (ciphertext) bằng khóa bí mật.
* **Giải mã (Decryption)**: Quá trình chuyển đổi dữ liệu từ dạng ciphertext trở lại dạng plaintext bằng cùng khóa bí mật.

### 2.1.3.  Các hệ mã khoá bí mật phổ biến

* **DES (Data Encryption Standard)**

DES là một thuật toán mã hóa dữ liệu đối xứng, được phát triển bởi Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST) vào những năm đầu của thập kỷ 1970 và được công bố vào năm 1977. DES sử dụng một khóa độc lập với chiều dài 56 bit và hoạt động trên các khối dữ liệu có kích thước 64 bit. Thuật toán DES bao gồm một loạt các vòng lặp, trong đó dữ liệu được hoán đổi, trộn lẫn và thay thế nhiều lần để tạo ra dữ liệu đã được mã hóa. DES đã trở thành một tiêu chuẩn quốc tế và được sử dụng rộng rãi cho đến khi bị thay thế bởi AES do các vấn đề về an toàn khóa ngắn.

* **AES (Advanced Encryption Standard)**

AES là một thuật toán mã hóa dữ liệu đối xứng hiện đại, được chọn lựa để thay thế DES bởi NIST vào năm 2001. AES sử dụng một phương pháp tiếp cận khác với DES và được thiết kế để cung cấp một mức độ bảo mật cao hơn. AES sử dụng các khóa với các chiều dài khác nhau, bao gồm 128-bit, 192-bit và 256-bit, và hoạt động trên các khối dữ liệu có kích thước 128 bit. Thuật toán AES sử dụng một số lượng vòng lặp khác nhau dựa trên chiều dài khóa và có các bước thay thế và hoán đổi dữ liệu cực kỳ phức tạp. AES đã trở thành một tiêu chuẩn quốc tế trong bảo mật thông tin và được sử dụng rộng rãi trong nhiều ứng dụng từ cơ sở dữ liệu đến truyền thông trên Internet.

### 2.1.4. Ứng dụng của khóa bí mật

* Một số ứng dụng tiêu biểu của khóa bí mật bao gồm:
* Bảo mật dữ liệu: Sử dụng khoá bí mật để mã hóa dữ liệu giúp bảo vệ thông tin cá nhân, doanh nghiệp và quốc gia khỏi sự truy cập trái phép.
* Giao tiếp an toàn: Khoá bí mật được sử dụng trong quá trình mã hóa và giải mã thông điệp, email, tin nhắn, đảm bảo rằng thông tin gửi đi chỉ được đọc bởi người nhận mong muốn.
* Xác thực và ủy quyền: Hệ thống khoá bí mật cũng được sử dụng để xác thực danh tính người dùng và quyền truy cập vào hệ thống, ứng dụng, dịch vụ.
* Tạo chữ ký điện tử: Khoá bí mật được sử dụng để tạo ra chữ ký điện tử, đảm bảo tính xác thực của các tài liệu, giao dịch điện tử.
* Bảo mật mạng: Trong mạng máy tính, khoá bí mật được sử dụng để bảo vệ dữ liệu trong quá trình truyền tải trên mạng, chẳng hạn qua giao thức SSL/TLS.
* Mặc dù có nhiều ưu điểm, mã hóa khóa bí mật cũng có một số hạn chế cần lưu ý:
* Vấn đề chia sẻ khóa: Việc chia sẻ khóa bí mật một cách an toàn giữa các bên tham gia giao tiếp có thể gặp nhiều khó khăn. Nếu khóa bí mật bị lộ, kẻ tấn công có thể dễ dàng giải mã thông tin.
* Quản lý khóa: Việc quản lý một lượng lớn khóa bí mật cho nhiều người dùng hoặc thiết bị có thể trở nên phức tạp và tốn kém.
* Rủi ro mất khóa: Nếu khóa bí mật bị mất hoặc hỏng, dữ liệu được mã hóa sẽ không thể truy cập được.
* Khả năng chống tấn công: Một số thuật toán mã hóa khóa bí mật có thể bị tấn công bẻ khóa bằng các phương pháp toán học tiên tiến hoặc sử dụng máy tính có hiệu suất cao.

## 2.2. Nghiên cứu, tìm hiểu về chuẩn mã dữ liệu DES

### 2.2.1. Giới thiệu

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

Khởi nguyên của thuật toán đã có từ đầu [thập niên 1970](https://vi.wikipedia.org/wiki/Th%E1%BA%ADp_ni%C3%AAn_1970). Vào năm [1972](https://vi.wikipedia.org/wiki/1972), sau khi tiến hành nghiên cứu về nhu cầu [an toàn máy tính](https://vi.wikipedia.org/w/index.php?title=An_to%C3%A0n_m%C3%A1y_t%C3%ADnh&action=edit&redlink=1) của chính phủ Hoa Kỳ, [Cục Tiêu chuẩn Liên bang Hoa Kỳ](https://vi.wikipedia.org/w/index.php?title=C%E1%BB%A5c_Ti%C3%AAu_chu%E1%BA%A9n_Li%C3%AAn_bang_Hoa_K%E1%BB%B3&action=edit&redlink=1) (National Bureau of Standard - NBS), hiện nay đã đổi tên thành [Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ](https://vi.wikipedia.org/w/index.php?title=Vi%E1%BB%87n_Ti%C3%AAu_chu%E1%BA%A9n_v%C3%A0_C%C3%B4ng_ngh%E1%BB%87_Qu%E1%BB%91c_gia_Hoa_K%E1%BB%B3&action=edit&redlink=1) (National Institute of Standards and Technology - NIST), đã nhận ra nhu cầu về một tiêu chuẩn của chính phủ dùng để [mật mã hóa](https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a) các thông tin mật/nhạy cảm. Các lý do chính là:

-      Sự phát triển của công nghệ thông tin và của nhu cầu an toàn & bảo mật thông tin: sự ra đời của các mạng máy tính tiền thân của Internet đã cho phép khả năng hợp tác và liên lạc số hóa giữa nhiều công ty, tổ chức trong các dự án lớn của chính phủ Mỹ.

-      Các thuật toán mã hóa cổ điển không thể đảm bảo được tính tin cậy đòi hỏi cao.

-      Các thiết bị khác nhau đòi hỏi sự trao đổi thông tin mật mã thống nhất, chuẩn.

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

-      Bảo mật ở mức cao.

-      Thuật toán được đặc tả và hoàn toàn công khai, tức là tính bảo mật không được phép dựa trên những những đặc điểm của thuật toán mã hóa.

-      Việc cài đặt phải dễ dàng để đem lại tính kinh tế.

-      Phải mềm dẻo để áp dụng được cho muôn vàn nhu cầu ứng dụng.

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

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

### 2.2.2. Đặc điểm của thuật toán DES

-      DES là thuật toán mã hóa khối, độ dài mỗi khối là 64 bit.

-      Khóa dùng trong DES có độ dài toàn bộ là 64 bit. Tuy nhiên chỉ có 56 bit thực sự được sử dụng và 8 bit còn lại dùng cho việc kiểm tra.

-      DES xuất ra bản mã 64 bit.

-      Thuật toán DES được thực hiện 16 vòng.

-      DES là hệ mật mã đối xứng vì vậy quá trình mã hóa và giải mã sử dụng cùng một khóa.

-      DES được thiết kế chạy trên phần cứng.

### 2.2.3. Ưu điểm và nhược điểm của thuật toán DES

**Ưu điểm:**

- DES cung cấp một mức độ bảo mật đáng tin cậy trong nhiều trường hợp. Dù đã có các kỹ thuật tấn công hiện đại như brute-force attacks và differential cryptanalysis, DES vẫn được sử dụng trong một số ứng dụng nhỏ và cần bảo mật thấp.

**-** DES dễ hiểu và triển khai, với một số loại phần cứng giới hạn. Điều này làm cho nó phổ biến trong các hệ thống có tài nguyên hạn chế.

**-** DES từng là một tiêu chuẩn quốc tế được chấp nhận rộng rãi cho việc mã hóa dữ liệu. Mặc dù đã bị thay thế bởi AES (Advanced Encryption Standard) với hiệu suất và bảo mật cao hơn, DES vẫn có giá trị trong một số trường hợp kỹ thuật cổ điển hoặc tương thích ngược.

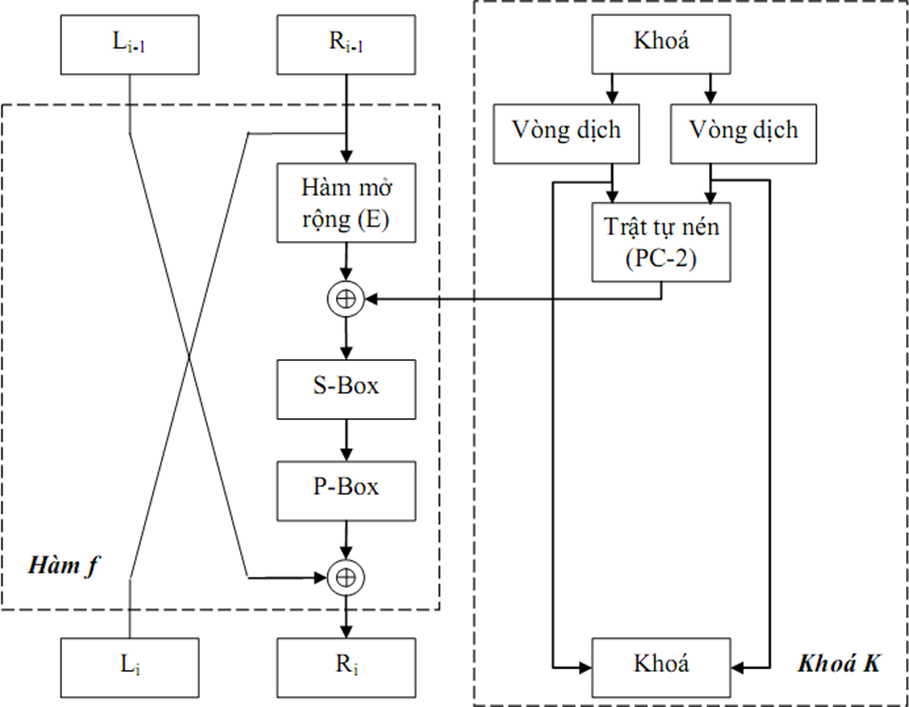
**Nhược điểm:**

Hiện nay DES được xem là không đủ an toàn cho nhiều ứng dụng. Nguyên nhân chủ yếu là độ dài 56 bit của khóa là quá nhỏ. Khóa DES đã từng bị phá trong vòng chưa đầy 24 giờ. Đã có rất nhiều kết quả phân tích cho thấy những điểm yếu về mặt lý thuyết của mã hóa có thể dẫn đến phá khóa, tuy chúng không khả thi trong thực tiễn. Thuật toán được tin tưởng là an toàn trong thực tiễn có dạng Triple DES (thực hiện DES ba lần), mặc dù | trên lý thuyết phương pháp này vẫn có thể bị phá. Gần đây DES đã được thay thế bằng AES (Advanced Encryption Standard, hay Tiêu chuẩn Mã hóa Tiên tiến).

Mặc dù đã có nhiều nghiên cứu về phá mã DES hơn bất kỳ phương pháp mã hóa khối nào khác nhưng phương pháp phá mã thực tế nhất hiện nay vẫn là tấn công kiều duyệt toàn bộ. Nhiều đặc tính mật mã hóa của DES đã được xác định và từ đó ba phương pháp phá mã khác được xác định với mức độ phức tạp nhỏ hơn tấn công duyệt toàn bộ. Tuy nhiên các phương pháp này đòi hỏi một số lượng bản rõ quá lớn (để tấn công lựa chọn bàn rõ) nên hầu như không thể thực hiện được trong thực tế.

*“NSA (National Standards Association) coi DES là một trong những sai lầm lớn nhất. Nếu họ biết trước rằng chi tiết của thuật toán sẽ được công bố để mọi người có thể viết chương trình phần mềm, họ sẽ không bao giờ đồng ý.”*

### 2.2.4. Thuật toán DES

****

*Hình 1. Sơ đồ một vòng DES*

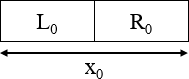
#### 2.2.4.1. Quá trình mã hóa

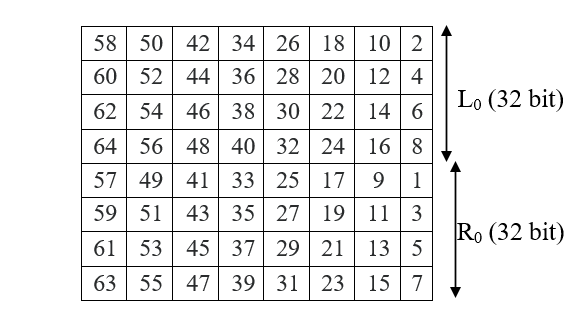
Quá trình mã hóa gồm 3 giai đoạn.

**a) Giai đoạn 1:**

Bản rõ x (64 bit) được hoán vị khởi tạo IP (Initial Permutation) tạo nên xâu bit x0. Chia x0 thành 2 khối L0 gồm 32 bit đầu tiên và R0 gồm 32 bit cuối.

**x0 = IP(x) = L0R0**



*Hình 5. Chia xâu x0 thành 2 khối L0 và R0*

*Bảng 1: Hoán vị IP.*

Hoán vị IP nhằm đổi chỗ khối dữ liệu vào, thay đổi vị trí của các bit trong khối dữ liệu vào. Ví dụ: hoán vị IP chuyển bit 1 thành bit 58, bit 2 thành bit 50, bit 3 thành bit 42…

**b) Giai đoạn 2:**

Từ L0 và R0 sẽ lặp 16 vòng, tại mỗi vòng tính :

**Li = Ri-1**

***Ri = Li-1***⊕ ***f(Ri-1,Ki),* với** 𝐢 **=**

Trong đó:

⊕ là phép XOR của 2 xâu bit.

Ki là các xâu có độ dài 48 bit được tính như là các hàm của khóa K. K1 đến K16 lập nên một lịch khóa.

**c) Giai đoạn 3:**

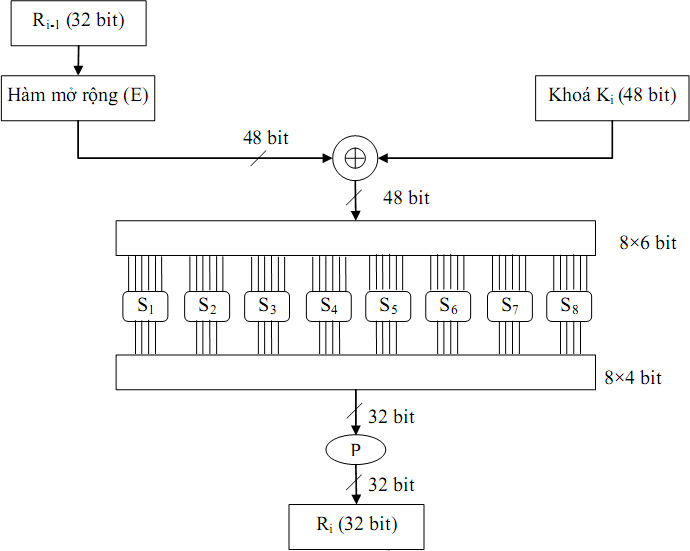
Áp dụng hoán vị ngược IP-1 cho xâu bit R16L16 để thu dược bản mã y (64 bit).

**y = IP-1(R16L16)**

| 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 2: Hoán vị IP-1.*

#### 2.2.4.2. Hàm F



*Hình 6. Sơ đồ hàm F.*

Hàm F lấy đối số đầu tiên là xâu nhập Ri-1 (32 bit), đối số thứ hai là Ki (48 bit) và tạo ra xâu xuất có độ dài 32 bit.

Hàm F được thực hiện qua các bước:

+) *Bước 1:* Đới số đầu tiên Ri-1 sẽ được “mở rộng” thành xâu có độ dài 48 bit thông qua hàm mở rộng E cố định.

Hàm mở rộng E: E(Ri-1) bao gồm 32 bit từ Ri-1, được hoán vị theo một cách thức xác định với 16 bit được tạo ra 2 lần.

| 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 3: Hàm mở rộng E.*

+) *Bước 2:* Tính E(Ri-1) ⊕ Ki thu được kết quả là một khối 48 bit. Khối này sẽ được chia thành 8 khối mỗi khối 6 bit. B = B1B2B3B4B5B6B7B8

+) *Bước 3:* Cho các khối Bi đi qua hộp Si sẽ biến một khối 6 bit thành một khối Ci 4 bit.

+) *Bước 4:* Xâu bit C = C1C2C3C4C5C6C7C8 có độ dài 32 bit được hoán vị tương ứng với hoán vị cố đinh P. Kết quả là P(C) = f(Ri-1, K1).

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

*Bảng 4: Hoán vị cố định P.*

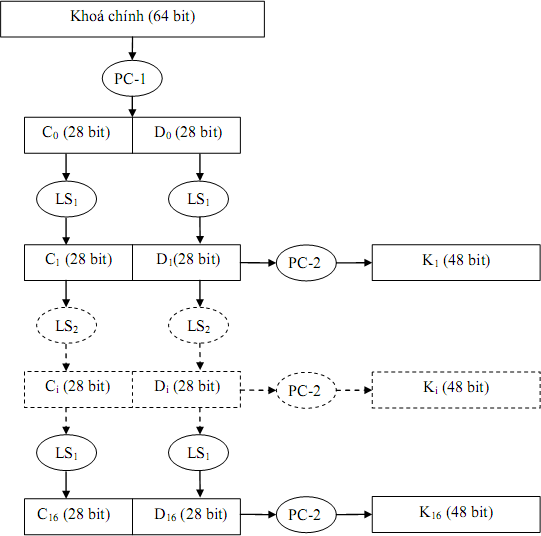
#### 2.2.4.3. Hộp S

* Dữ liệu vào 6 bit cho ra kết quả 4 bit. Dữ liệu ra là giao của hàng và cột.
* Mỗi hộp S-box là một bảng gồm 4 hàng 16 cột được đánh số từ 0. Như vậy mỗi hộp S có hàng 0, 1, 2, 3; cột 0, 1, ... ,15. Mỗi phần tử của hộp là một số 4 bit. Sáu bit vào hộp S sẽ xác định số hàng và số cột để tìm kết quả ra.
* Mỗi khối Bi có 6 bit kí hiệu là b1, b2, b3, b4, b5 và b6. Bit b1 và b6 được kết hợp thành một số 2 bit, nhận giá trị từ 0 đến 3, tương ứng với một hàng trong bảng S. Bốn bit ở giữa từ b2 tới b5 được kết hợp thành một số 4 bit, nhận giá trị từ 0 đến 15, tương ứng với một cột trong bảng S.

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

*Bảng 5: S-box.*

#### 2.2.4.4. Tạo khóa K



*Hình 7. Sơ đồ tạo khóa con.*

Khóa K là một xâu có độ dài 64 bit trong đó 56 bit dùng làm khóa và 8 bit dùng để kiểm tra sự bằng nhau để phát hiện lỗi. Các bit ở các vị trí 8, 16, 24, 32, 40, 48, 56, 64 được xác định là các bit kiểm tra.

Quá trình tạo khóa con:

* Cho khóa K 64 bit, loại bỏ các bit kiểm tra tại các vị trí 8, 16, 24, 32, 40, 48, 56, 64. Thực hiện hoán vị các bit còn lại của K tương ứng với hoán vị cố định PC-1.

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

*Bảng 6: Hoán vị cố định PC-1.*

* 56 bit kết quả được chia thành 2 khối C0 , D0 mỗi khối 28 bit trong dó C0 gồm 28 bit đầu tiên và D0 gồm 28 bit còn lại.

**PC-1(K) = C0D0**

* Tại mỗi vòng lặp, Ci và Di sẽ có được từ phép dịch trái các bit của Ci-1 và Di-1. Các vòng 1, 2, 9, 16 dịch trái 1 bit, các vòng còn lại dịch trái 2 bit.

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

*Bảng 7: Bảng dịch bit tại các vòng lặp của DES.*

* Sau khi dịch trái, cho xâu CiDi (56 bit) qua hoán vị cố định PC-2 thu được khóa Ki (48 bit).

**Ki = PC-2(CiDi)**

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

*Bảng 8: Hoán vị cố định PC-2.*

#### 2.2.4.5. Quá trình giải mã

Việc giải mã dùng cùng một thuật toán như việc mã hoá.

Để giải mã dữ liệu đã được mã hoá, quá trình giống như mã hoá được lặp lại nhưng các chìa khoá phụ được dùng theo thứ tự ngược lại từ K16 đến K1, nghĩa là trong bước 2 của quá trình mã hoá dữ liệu đầu vào ở trên Ri-1 sẽ được XOR với K17-i chứ không phải với Ki.

#### 2.2.4.6. Bài toán mã hóa và giải mã sử dụng thuật toán DES

Giả sử ta mã hóa bản rõ sau trong dạng thập lục phân **0123456789ABCDEF** và sử dụng khóa thập lục phân **133457799BBCDFF1.**

Khóa trong dạng nhị phân không có các bit kiểm tra sẽ là: 00010010011010010101101111001001101101111011011111111000

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

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

16 vòng lặp mã hóa được thể hiện như sau:

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

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

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

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

| **E(R4) K5**  **E(R4)** ⊕ **K5**  **S-box output f(R4, K5)**  **L6 = R5** | **=**  **=**  **=**  **=**  **=**  **=** | **101110101110100100000100000000000000001000001010**  **011111001110110000000111111010110101001110101000**  **110001100000010100000011111010110101000110100010**  **01010000110010000011000111101011**  **00101000000100111010110111000011**  **10001010010011111010011000110111** |
| --- | --- | --- |

| **E(R5) K6**  **E(R5)** ⊕ **K6**  **S-box output f(R5, K6)**  **L7 = R6** | **=**  **=**  **=**  **=**  **=**  **=** | **110001010100001001011111110100001100000110101111**  **011000111010010100111110010100000111101100101111**  **101001101110011101100001100000001011101010000000**  **01000001111100110100110000111101**  **10011110010001011100110100101100**  **11101001011001111100110101101001** |
| --- | --- | --- |

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

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

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

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

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

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

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

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

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

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

Cuối cùng, áp dụng hoán vị IP-1 cho R16L16 ta nhận được bản mã trong dạng thập lục phân như sau:

**Y = 85E813540F0AB405**

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

### 2.3.1. Thiết kế kịch bản chương trình

#### 2.1.1.1. Chuẩn bị dữ liệu

Thư viện: Cần import thư viện mã hóa phù hợp với ngôn ngữ lập trình đang sử dụng.

Dữ liệu: Chuẩn bị dữ liệu cần mã hóa hoặc giải mã dưới dạng chuỗi hoặc mảng byte. Đảm bảo dữ liệu nhập vào có độ dài phù hợp với chuẩn DES (64 bit cho mỗi khối liệu). Yêu cầu người dùng nhập dữ liệu cần mã hóa/giải mã (có thể là văn bản, chuỗi byte,...)

Khóa: Chuẩn bị khóa mã hóa 56 bit dưới dạng chuỗi hoặc mảng byte. Có thể cho phép người dùng nhập khóa dưới dạng nhị phân hoặc mã hóa theo định dạng khác (ví dụ: mã hóa Base64).

#### 2.1.1.2. Lựa chọn thao tác

Hiển thị menu cho phép người dùng lựa chọn thao tác: mã hóa hoặc giải mã. Ghi nhận lựa chọn của người dùng.

#### 2.1.1.3. Mã hoá

Chia nhỏ dữ liệu: Chia nhỏ dữ liệu thành các khối 64 bit.

Tạo khóa con: Sử dụng thuật toán DES để tạo 16 khóa con 48 bit từ khóa chính 56 bit.

Vòng lặp mã hóa:

* Mở rộng khối dữ liệu: Mở rộng khối dữ liệu 64 bit thành 64 bit bằng cách lặp lại mỗi bit.
* Chia thành hai nửa: Chia khối dữ liệu mở rộng thành hai nửa 32 bit trái (L) và phải (R).
* Vòng lặp 16 vòng:

o Chuyển đổi R: Sử dụng hàm F và khóa con tương ứng để biến đổi nửa khối R.

o Gộp và hoán đổi: Gộp hai nửa L và R đã biến đổi, sau đó hoán đổi vị trí của L và R.

* Kết hợp: Kết hợp hai nửa L và R sau vòng lặp 16 thành khối dữ liệu mã hóa 64 bit.

Lặp lại: Lặp lại các bước 2 và 3 cho đến khi mã hóa tất cả các khối dữ liệu.

#### 2.1.1.4. Giải mã

Tạo khóa con đảo ngược: Sử dụng thuật toán DES để tạo 16 khóa con 48 bit đảo ngược từ khóa chính 56 bit.

Vòng lặp giải mã:

* Chia thành hai nửa: Chia khối dữ liệu mã hóa 64 bit thành hai nửa 32 bit trái (L) và phải (R).
* Vòng lặp 16 vòng:

o Chuyển đổi R: Sử dụng hàm F và khóa con đảo ngược tương ứng để biến đổi nửa khối R.

o Gộp và hoán đổi: Gộp hai nửa L và R đã biến đổi, sau đó hoán đổi vị trí của L và R.

* Kết hợp: Kết hợp hai nửa L và R sau vòng lặp 16 thành khối dữ liệu giải mã 64 bit.

Lặp lại: Lặp lại các bước 1 và 2 cho đến khi giải mã tất cả các khối dữ liệu.

Gộp dữ liệu: Gộp các khối dữ liệu giải mã thành chuỗi hoặc mảng byte ban đầu.

#### 2.1.1.5. Hiển thị kết quả

Hiển thị văn bản mã hóa hoặc giải mã trên màn hình.

Cung cấp thông tin bổ sung như thời gian thực hiện, trạng thái thành công/thất bại,...

### 2.3.2. Giới thiệu ngôn ngữ lập trình sử dụng để cài đặt thuật toán.

#### 2.3.2.1. Ngôn ngữ lập trình C#

* Tổng quan

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.

C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), . . . trở nên rất dễ dàng.

* Đặc điểm:
  + Hướng Đối Tượng (Object-Oriented Programming - OOP): C# là ngôn ngữ lập trình hướng đối tượng hoàn toàn, hỗ trợ các khái niệm như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism), và đóng gói (encapsulation).
  + Tích Hợp Chặt Chẽ Với .NET Framework: C# được thiết kế để chạy trên .NET Framework, cung cấp một thư viện phong phú và các dịch vụ chạy thời gian (runtime) mạnh mẽ, bao gồm quản lý bộ nhớ, bảo mật, và xử lý lỗi.
  + Quản Lý Bộ Nhớ Tự Động (Automatic Memory Management): C# sử dụng cơ chế Garbage Collection để tự động quản lý và giải phóng bộ nhớ, giúp giảm bớt các vấn đề liên quan đến quản lý bộ nhớ thủ công.
  + Hỗ Trợ Đa Luồng (Multithreading Support): C# cung cấp các công cụ và thư viện để phát triển các ứng dụng đa luồng một cách dễ dàng và hiệu quả.
  + Tính An Toàn và Bảo Mật (Type Safety and Security): C# có cơ chế kiểm tra kiểu dữ liệu nghiêm ngặt và cung cấp nhiều công cụ bảo mật để đảm bảo an toàn cho ứng dụng.
* Ứng dụng của C#:
  + Phát Triển Web: ASP.NET là một framework mạnh mẽ cho phép xây dựng các ứng dụng web và dịch vụ web.
  + Phát Triển Ứng Dụng Di Động: Xamarin cho phép phát triển các ứng dụng di động đa nền tảng sử dụng C# và .NET.
  + Phát Triển Ứng Dụng Desktop: Windows Forms và WPF (Windows Presentation Foundation) là các công cụ phổ biến để phát triển ứng dụng giao diện người dùng trên desktop.
  + Phát Triển Game: Unity, một trong những engine phát triển game phổ biến nhất, sử dụng C# làm ngôn ngữ kịch bản chính.

#### 2.3.2.2. Ngôn ngữ lập trình C++

* Tổng quan

C++ là một ngôn ngữ lập trình đa mục đích, hướng đối tượng, được phát triển bởi Bjarne Stroustrup vào đầu những năm 1980. Nó là sự mở rộng của ngôn ngữ C, và tên gọi C++ xuất phát từ việc sử dụng toán tử "++" trong C để biểu thị sự gia tăng, ý chỉ sự cải tiến và mở rộng của C.

* Đặc điểm nổi bật

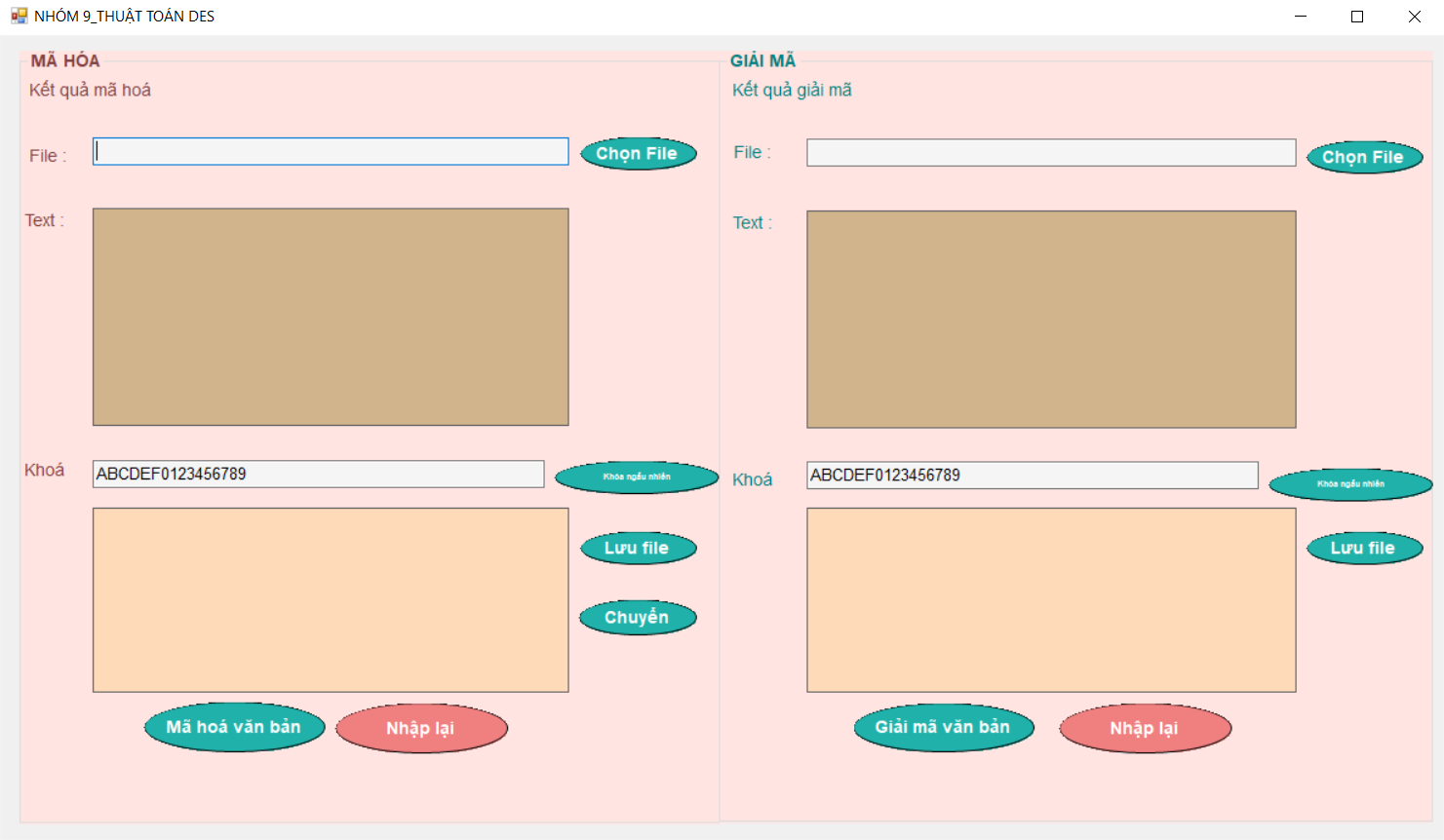
1. Hướng đối tượng (Object-Oriented Programming - OOP):
   * C++ hỗ trợ đầy đủ các khái niệm của lập trình hướng đối tượng như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism), và tính đóng gói (encapsulation).
2. Hiệu suất cao:
   * C++ cho phép lập trình viên quản lý tài nguyên hệ thống và bộ nhớ một cách chi tiết, giúp tối ưu hóa hiệu suất của chương trình.
3. Đa năng:
   * C++ có thể sử dụng để phát triển nhiều loại ứng dụng khác nhau, từ hệ thống nhúng, ứng dụng desktop, game, đến các phần mềm hệ thống.
4. Thư viện phong phú:
   * C++ có một hệ thống thư viện tiêu chuẩn (Standard Library) mạnh mẽ và phong phú, bao gồm STL (Standard Template Library) với các cấu trúc dữ liệu và thuật toán hữu ích.

* Ưu điểm
  + Tính linh hoạt: C++ có thể sử dụng để viết các phần mềm cấp thấp gần với phần cứng và các phần mềm ứng dụng cao cấp.
  + Hiệu suất: Khả năng quản lý bộ nhớ thủ công cho phép tối ưu hóa hiệu suất, đặc biệt quan trọng trong các ứng dụng yêu cầu tài nguyên lớn.
  + Khả năng mở rộng: Các tính năng của C++ cho phép xây dựng các hệ thống phần mềm lớn và phức tạp.
* Nhược điểm
  + Độ phức tạp: C++ có cú pháp phức tạp và yêu cầu kiến thức sâu về quản lý bộ nhớ, điều này có thể gây khó khăn cho người mới bắt đầu.
  + Lỗi runtime: Quản lý bộ nhớ thủ công có thể dẫn đến các lỗi như rò rỉ bộ nhớ (memory leak) hoặc lỗi truy cập bộ nhớ (segmentation fault).
* Ứng Dụng
  + Phần mềm hệ thống: Hệ điều hành, trình biên dịch, và các phần mềm hệ thống khác.
  + Game: Các engine game nổi tiếng như Unreal Engine và Unity sử dụng C++ để tối ưu hóa hiệu suất.
  + Ứng dụng tài chính: Nhiều hệ thống giao dịch tài chính yêu cầu hiệu suất cao được viết bằng C++.
  + Ứng dụng khoa học và kỹ thuật: C++ được sử dụng để phát triển các phần mềm mô phỏng và tính toán khoa học

### 2.3.3. Thiết kế, cài đặt chương trình

#### 2.3.3.1. Chương trình C#

* Chương trình với ngôn ngữ C#



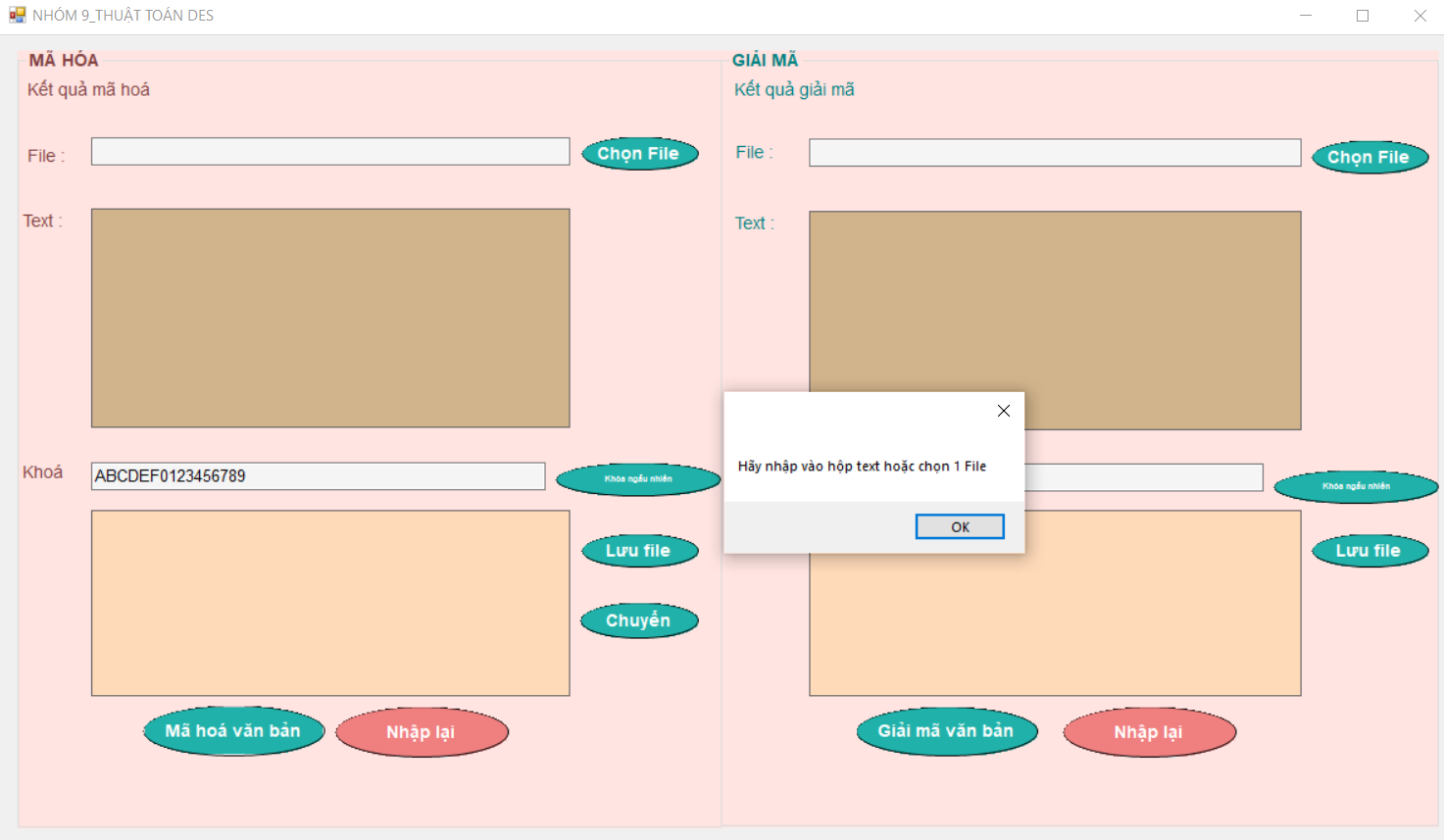
*Hình 2.3.3.1. Giao diện demo mã hóa và giải mã DES* ngôn ngữ C#

**Mã hóa**

***Bước 1: Chọn file văn bản cần mã hóa hoặc nhập vào ô “Text”***

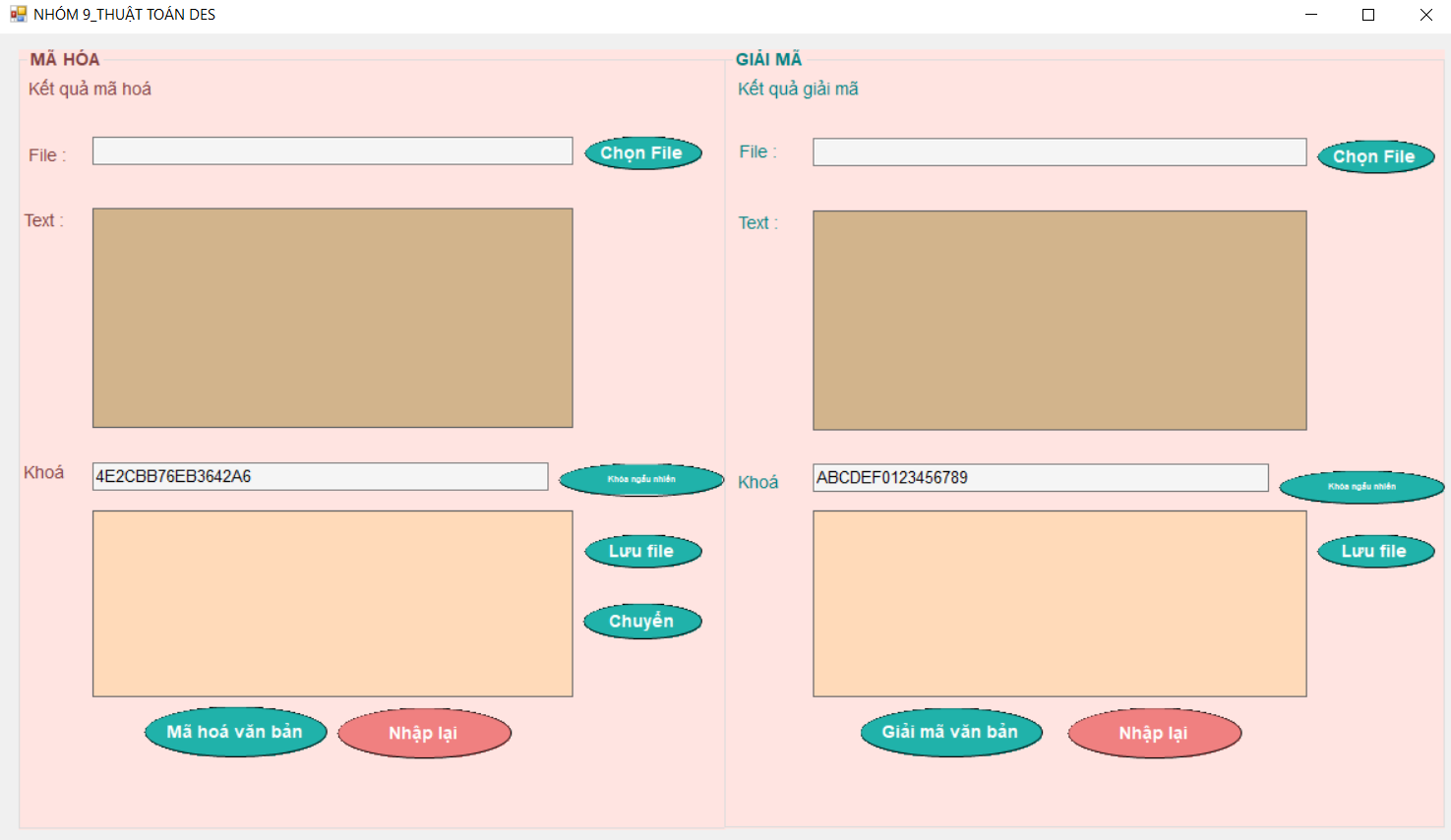
Ở đây thực hiện bắt lỗi nhập liệu :

- Lỗi chưa chọn file hoặc lỗi chưa nhập vào ô “Text”

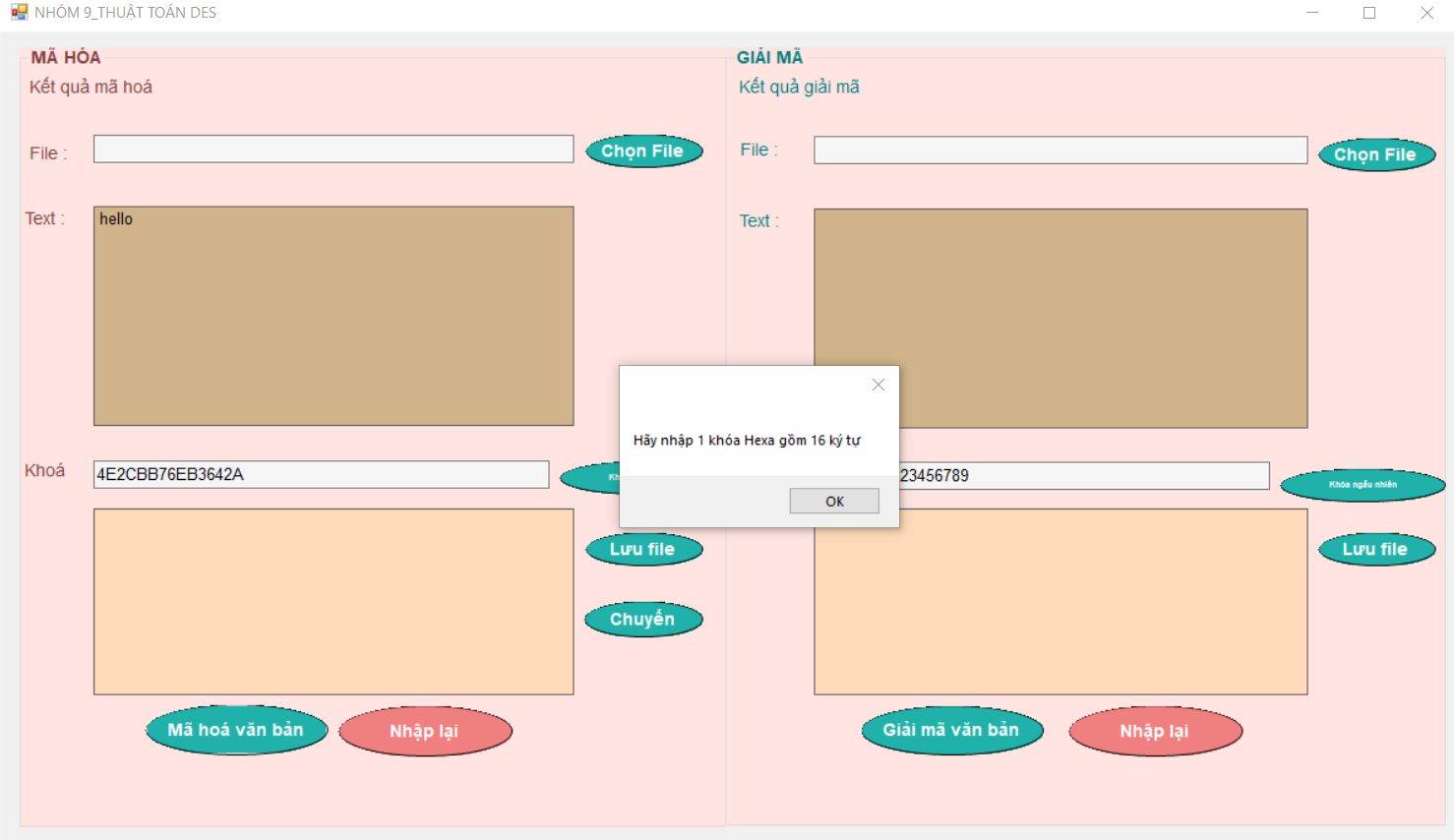


***Bước 2: Tạo khóa ngẫu nhiên***

Khi người dùng kích vào nút “Khóa ngẫu nhiên”, chương trình sẽ tự động sinh ra khóa ngẫu nhiên nào đó.

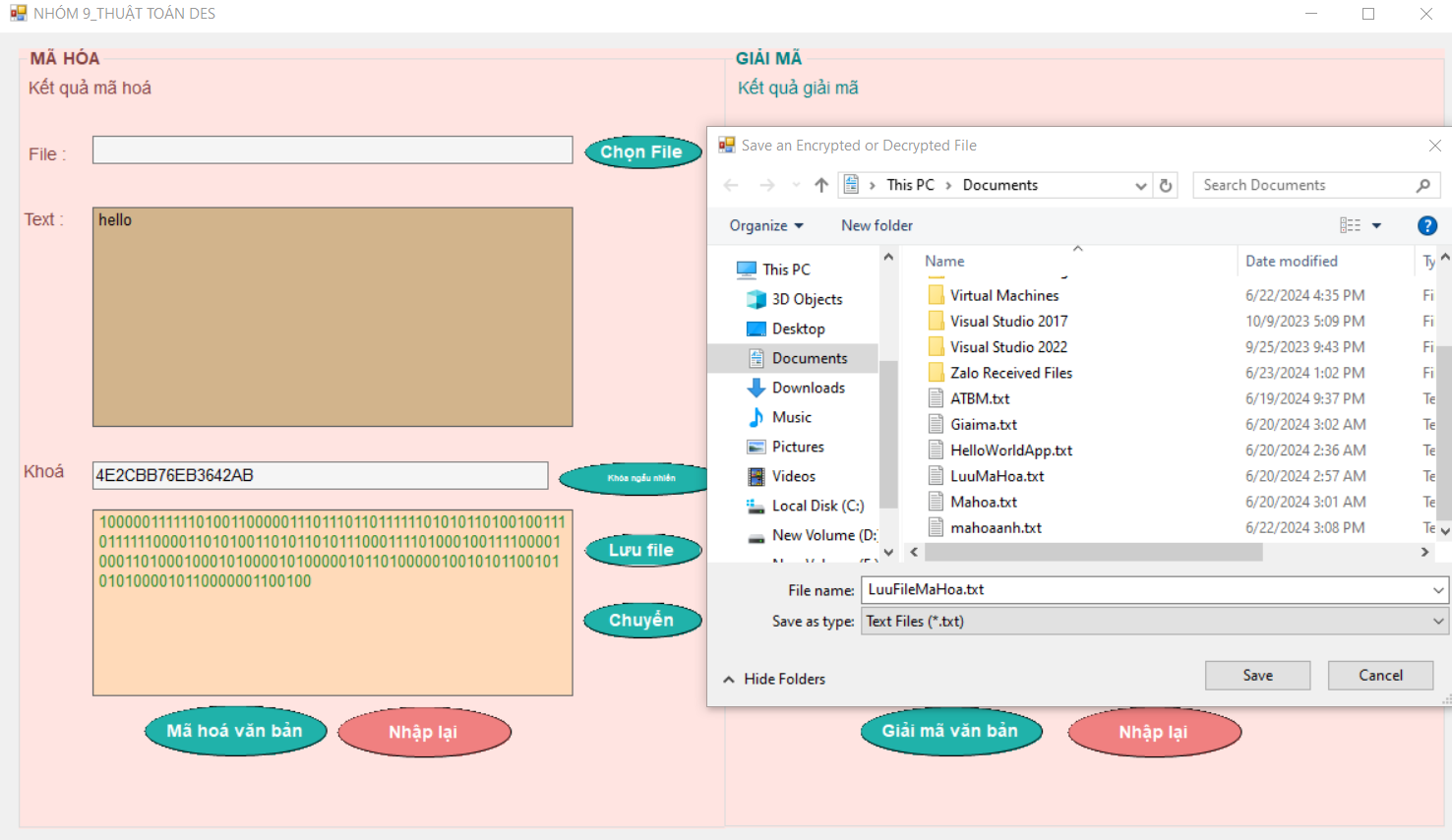


- Lỗi khi nhập thiếu khóa Hexa gồm 16 kí tự:

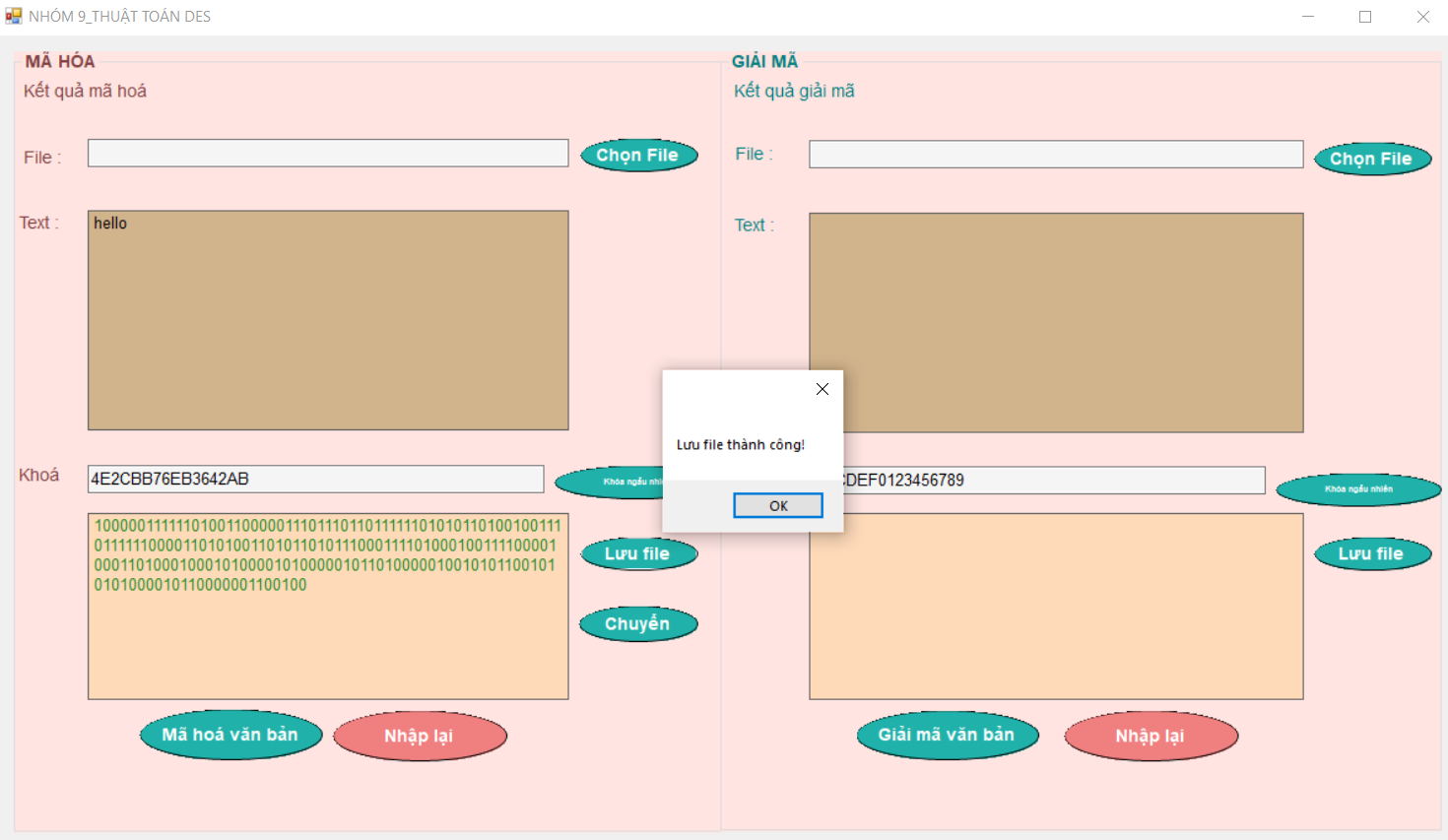


***Bước 3: Lưu file khi thực hiện mã hóa thành công***

Người dùng kích vào nút “Lưu file”, màn hình sẽ hiện lên màn hình yêu cầu người dùng lưu file mã hóa theo tên file

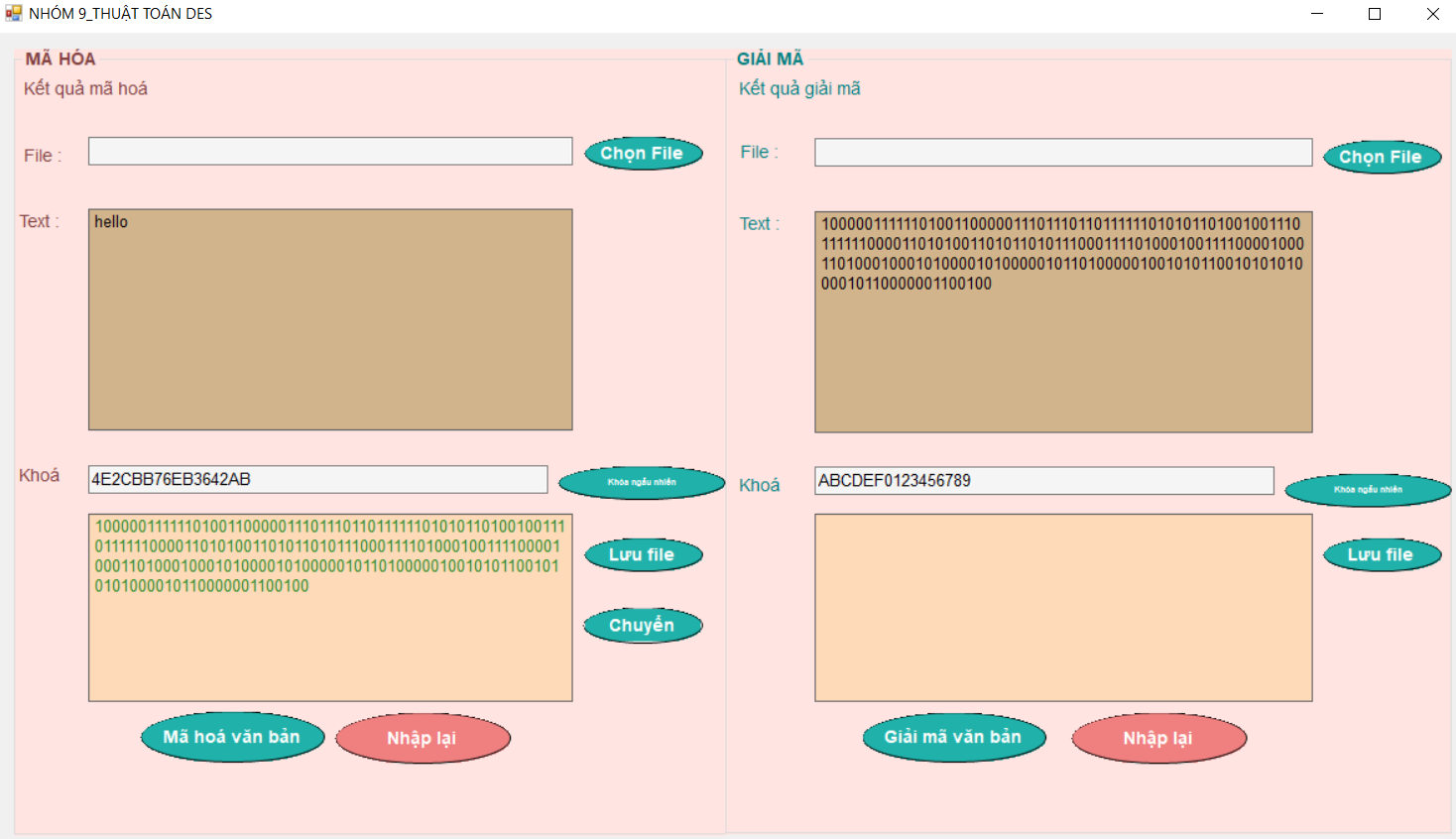


- Kết quả khi lưu file thành công:



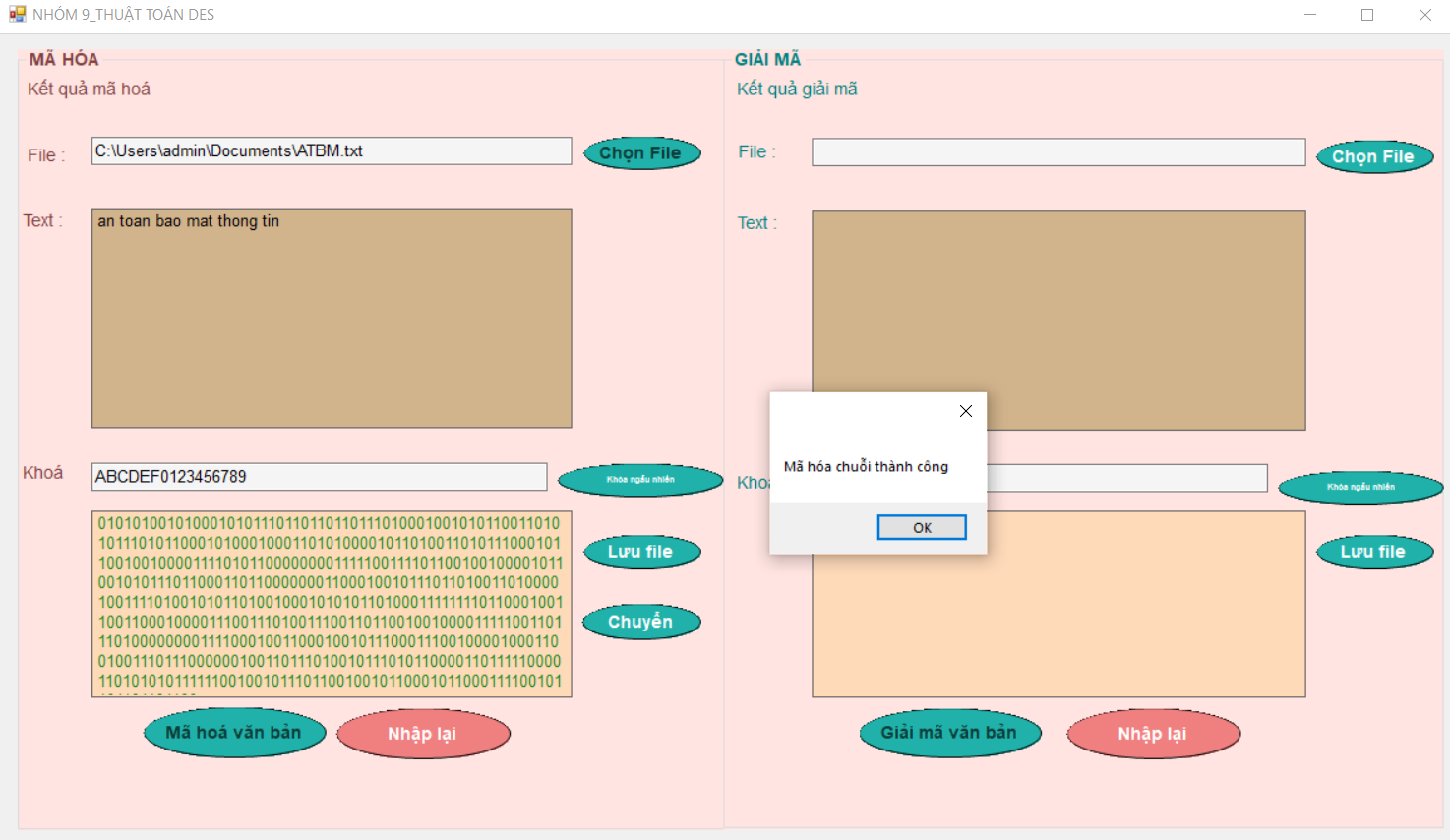
***Bước 4: Chuyển kết quả mã hóa sang ô text của giải mã***

- Khi người dùng kích vào nút chuyển, chương trình sẽ chuyển kết quả mã hóa sang ô text của giải mã



***Bước 5: Mã hóa chuỗi thành công***

Khi thực hiện mã hóa văn bản thành công, chương trình sẽ hiển thị nội dung của file vào ô text, hiển thị khóa Hexa gồm 16 ký tự và kết quả mã hóa sẽ hiển thị với thông báo “Mã hóa chuỗi thành công” như sau:

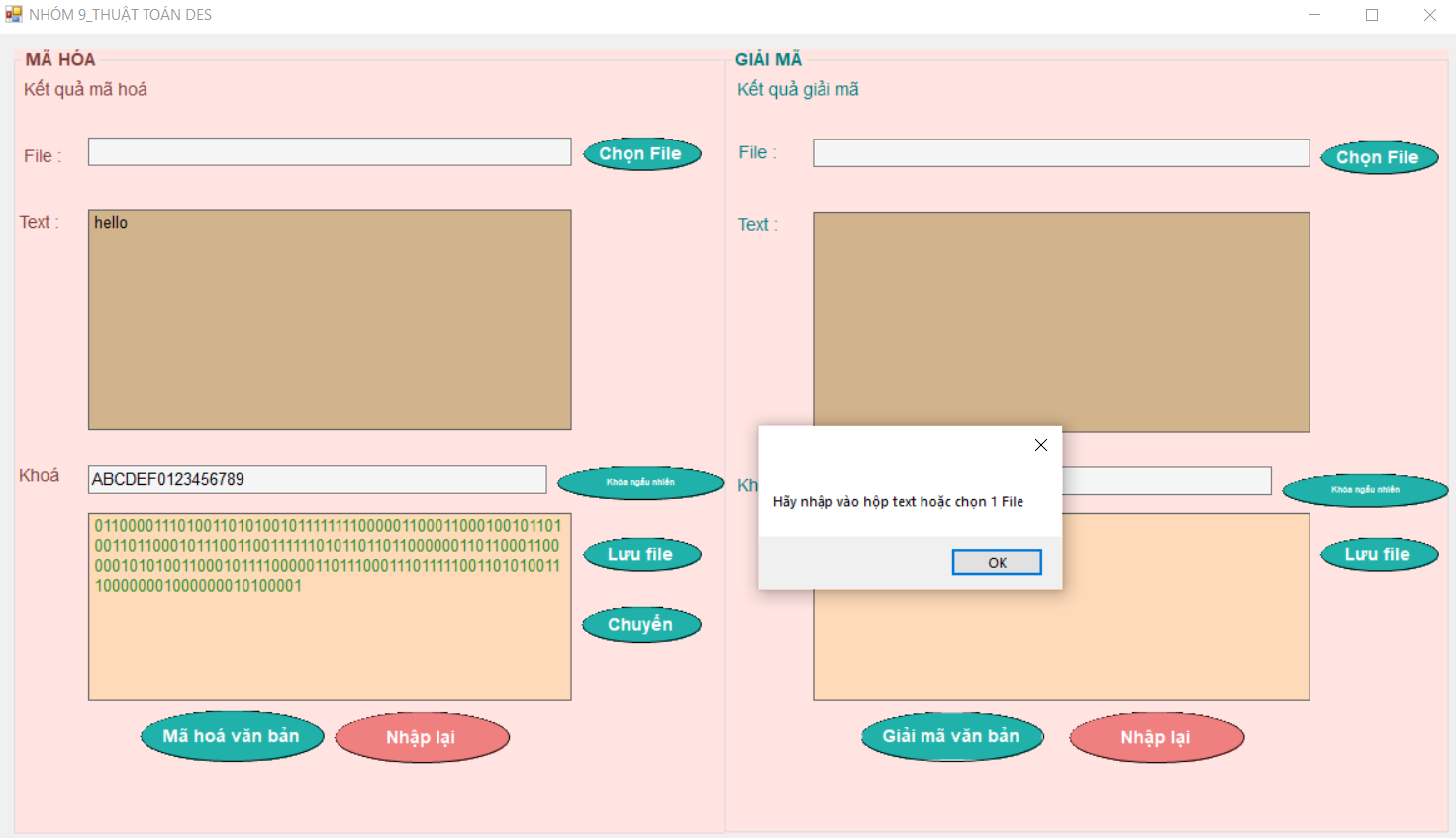


**Giải mã**

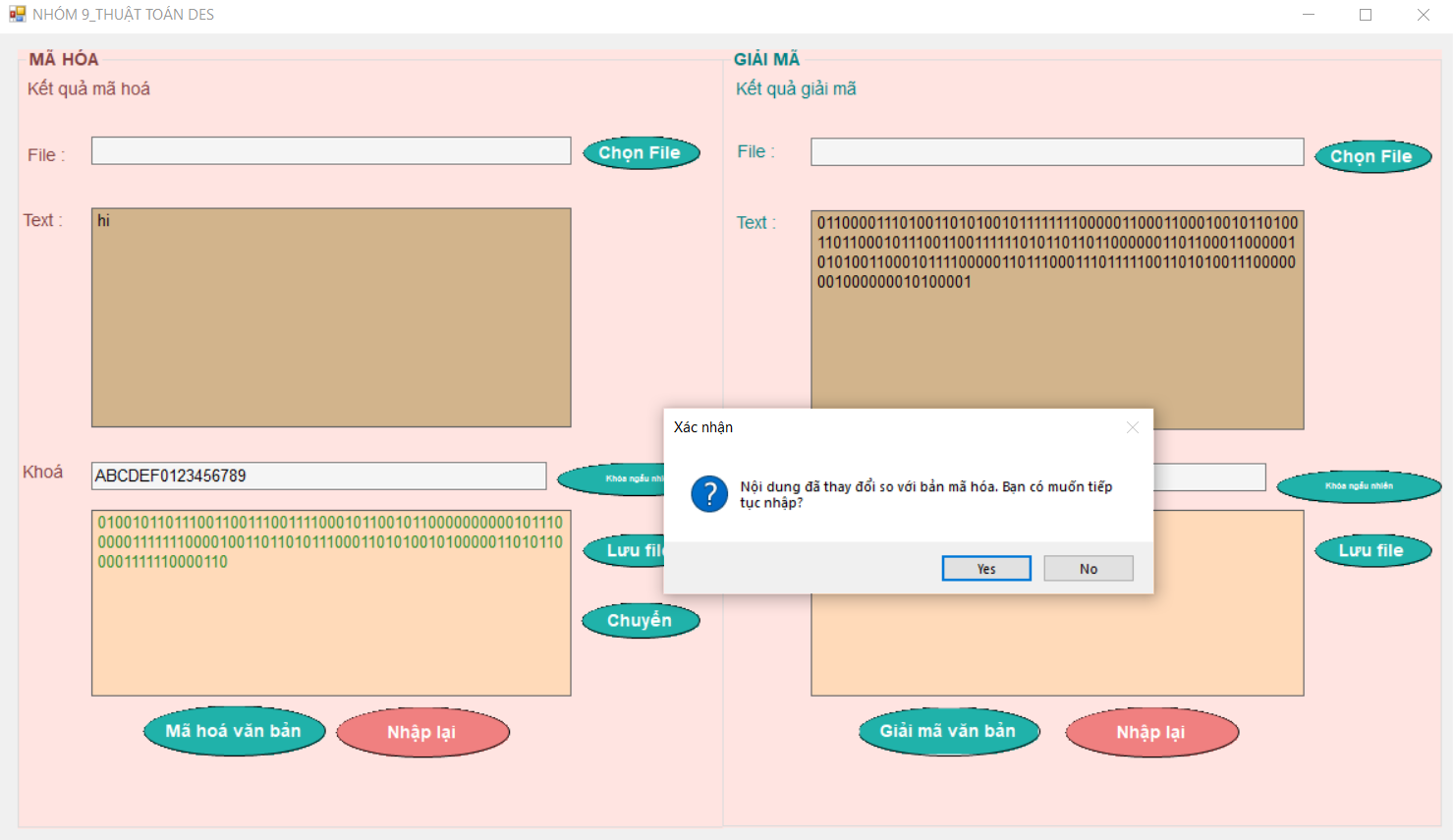
***Bước 1: Chọn file văn bản cần giải mã hoặc nhập vào ô “Text”***

Ở đây thực hiện bắt lỗi nhập liệu :

- Lỗi chưa chọn file hoặc lỗi chưa nhập vào ô “Text”

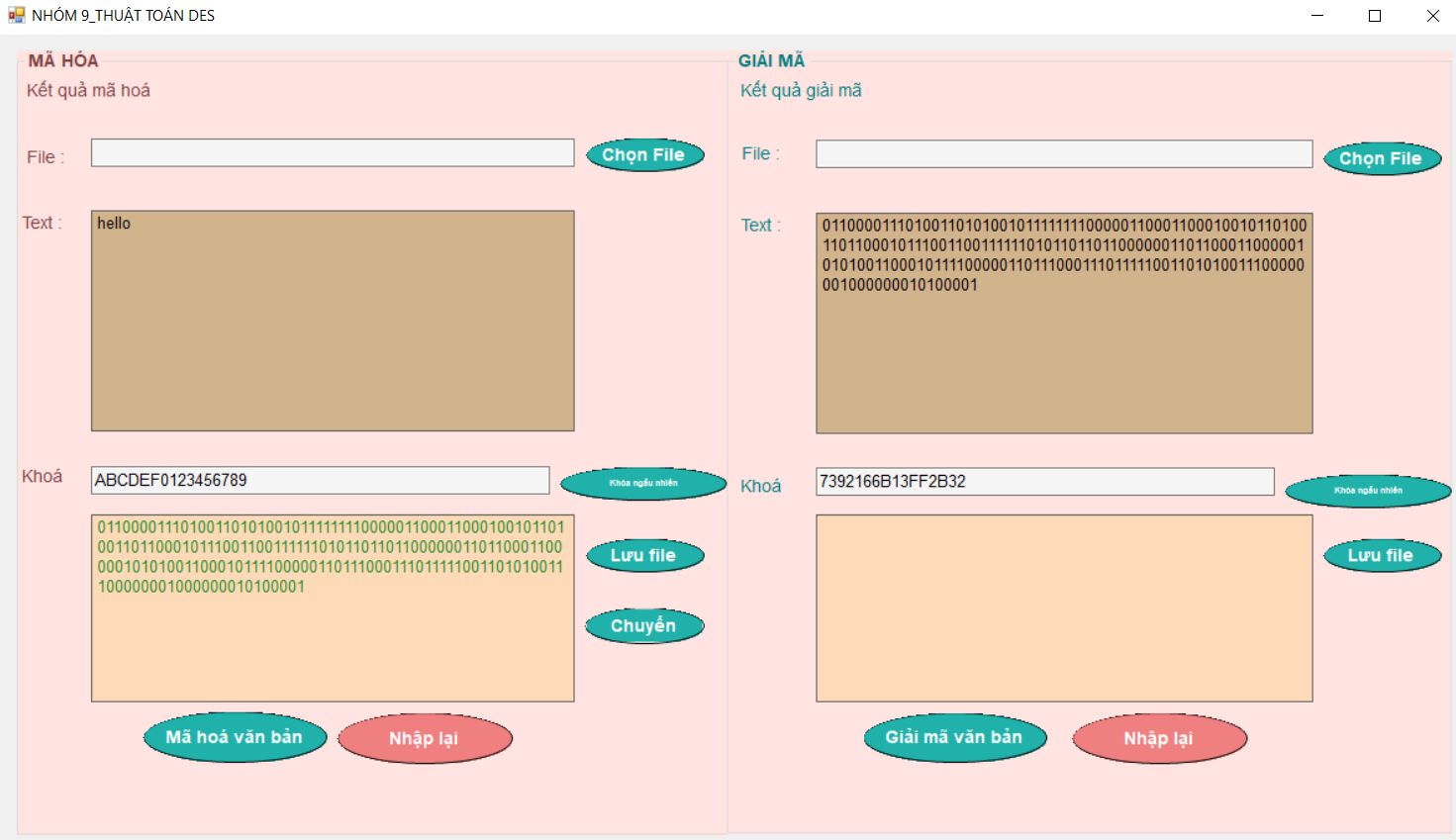


- Xác nhận nội dung của ô text thay đổi so với kết quả của mã hóa để tiến hành giải mã:



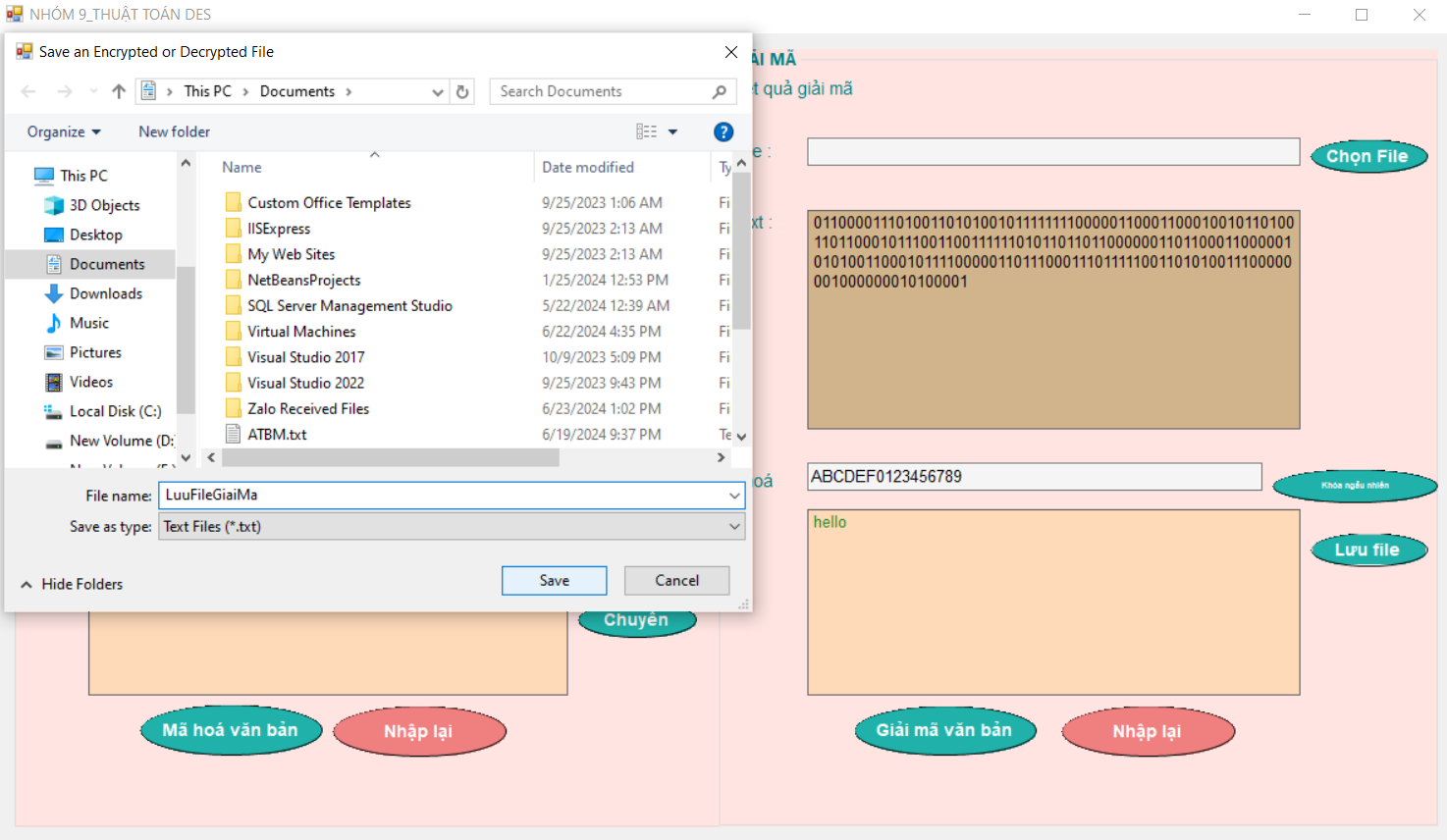
***Bước 2: Tạo khóa ngẫu nhiên***

Khi người dùng kích vào nút “Khóa ngẫu nhiên”, chương trình sẽ tự động sinh ra khóa ngẫu nhiên nào đó.

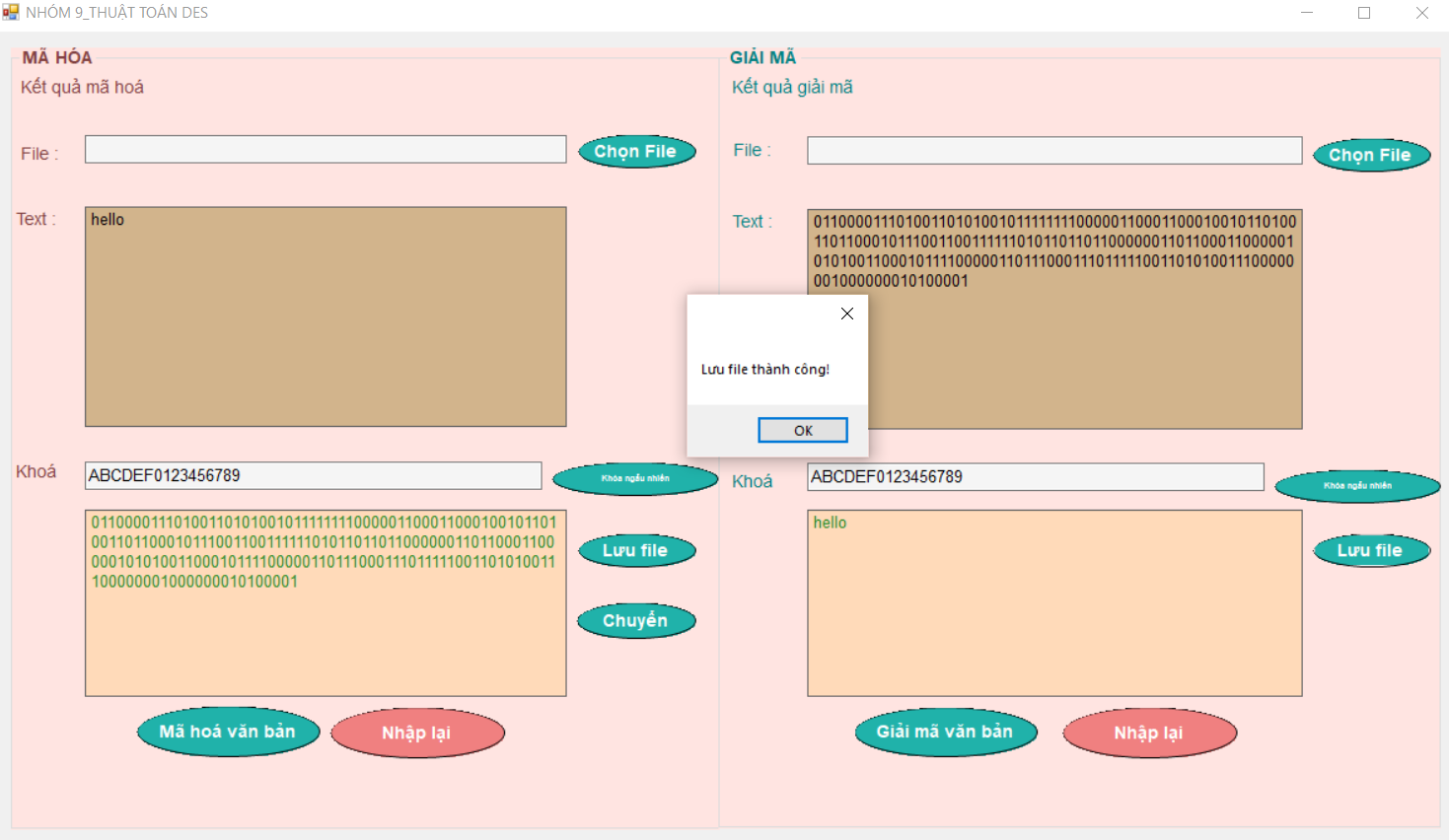


***Bước 3: Lưu file khi thực hiện giải mã thành công***

Người dùng kích vào nút “Lưu file”, màn hình sẽ hiện lên màn hình yêu cầu người dùng lưu file giải mã theo tên file

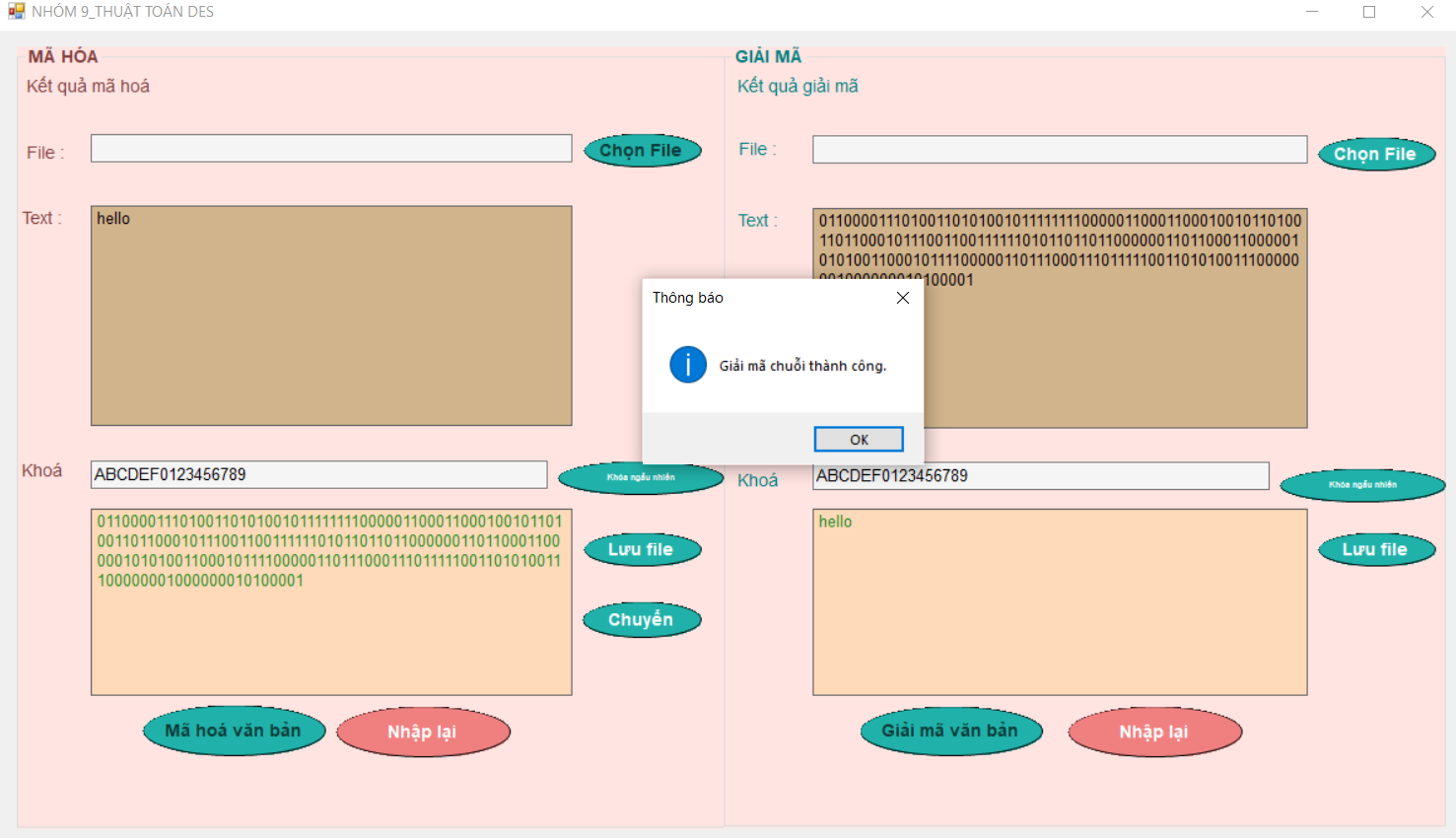
****

- Kết quả khi lưu file thành công:

****

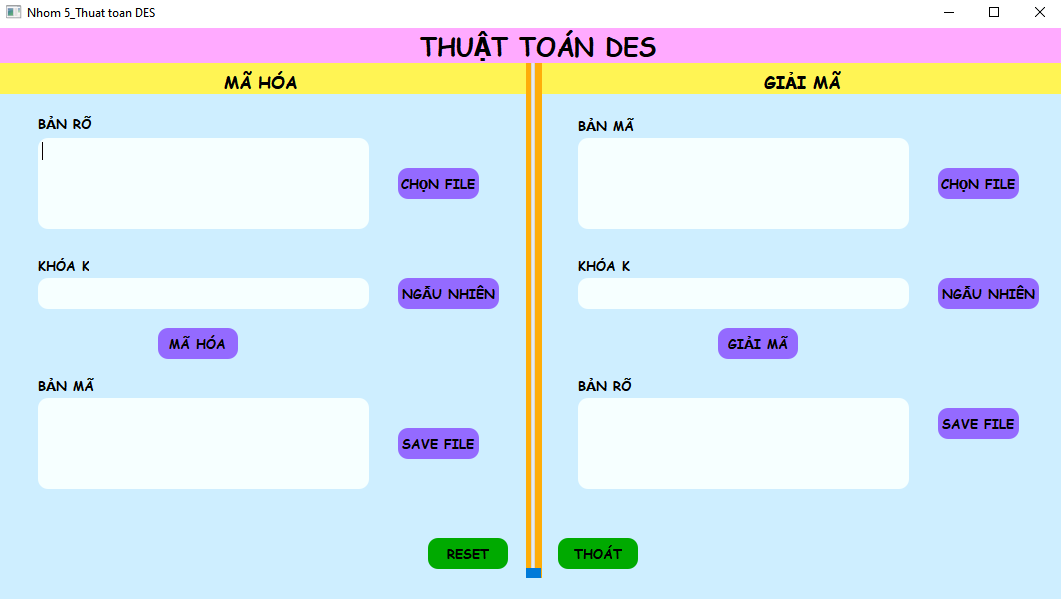
***Bước 4: Giải mã thành công***

Khi thực hiện giải mã thành công, chương trình sẽ hiển thị nội dung của file vào ô text, hiển thị khóa Hexa gồm 16 ký tự và kết quả giải mã sẽ hiển thị với thông báo “Giải mã chuỗi thành công” như sau:



#### 2.3.3.2. Chương trình C++

* **Chương trình với ngôn ngữ C++**

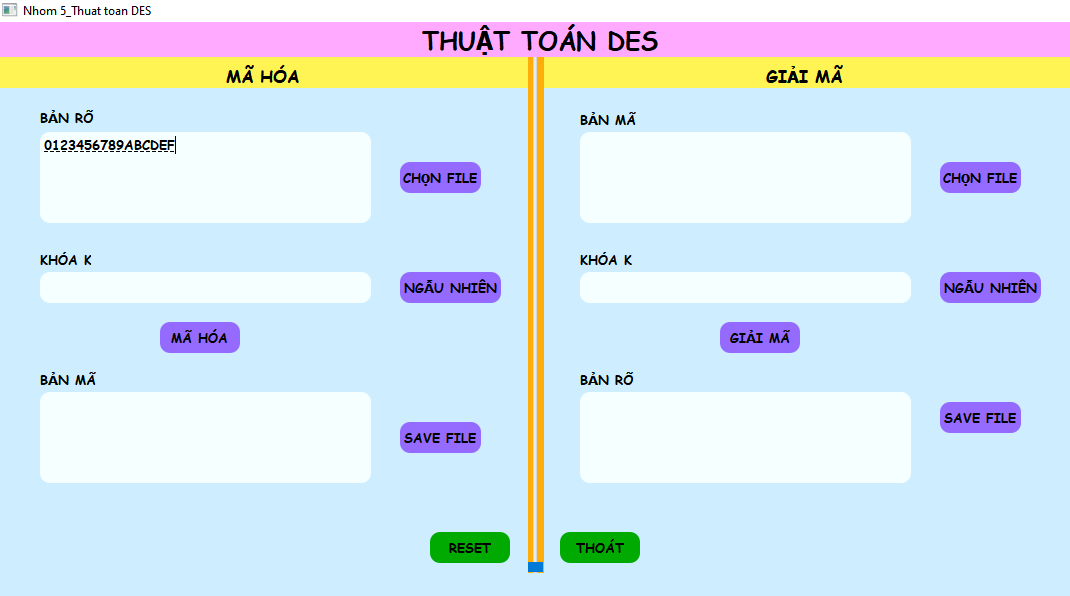


*Hình 2.3.3.2. Giao diện demo mã hóa và giải mã DES ngôn ngữ C++*

* **Mã hóa**

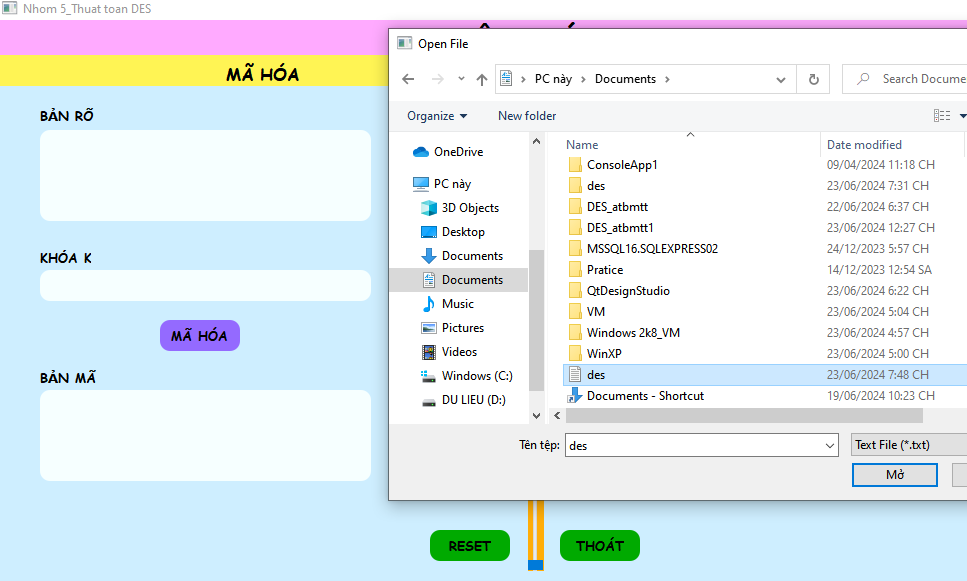
***Bước 1: Tạo bản rõ***

Cách 1: Nhập trực tiếp bản rõ vào ô text

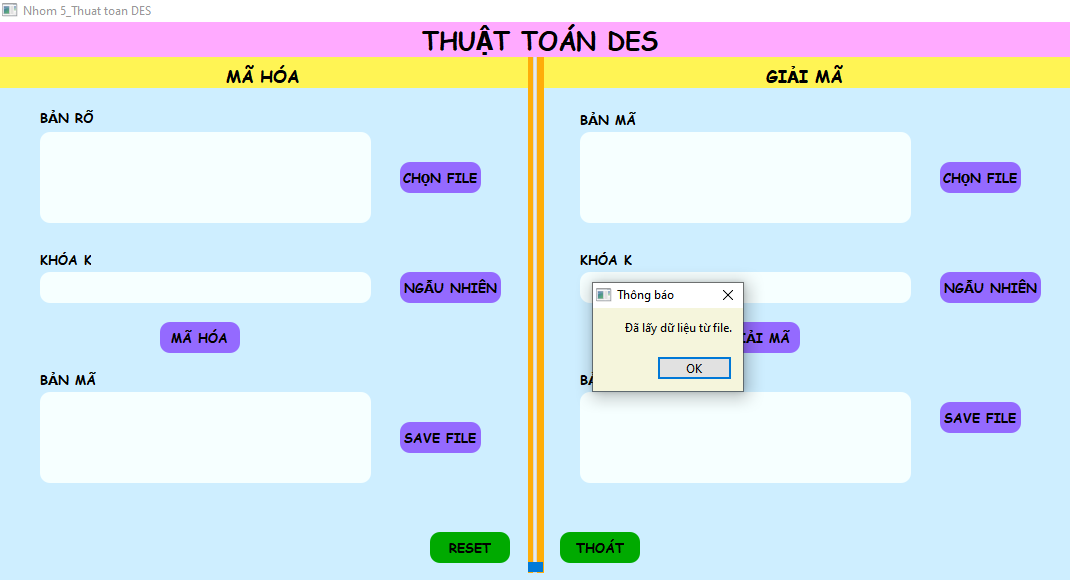


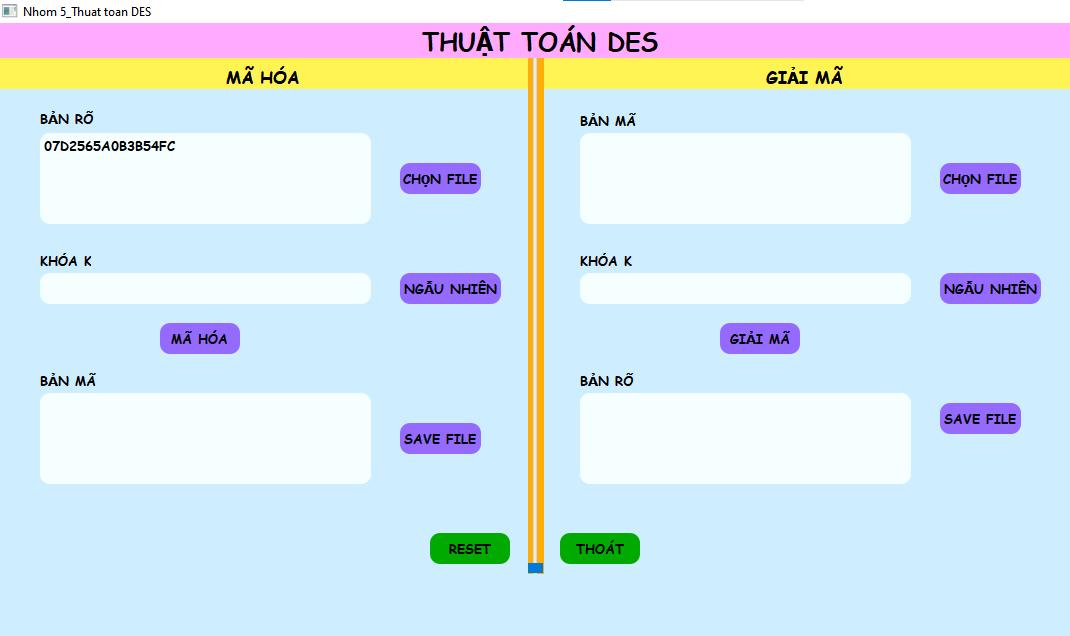
Cách 2: Bấm vào nút “ Chọn File”

Khi bấm vào nút “ Chọn File”, màn hình sẽ hiển thị ra các file. Khi đó hãy chọn 1 file .txt bất kì đã có dữ liệu của bản rõ để thực hiện mã hóa.

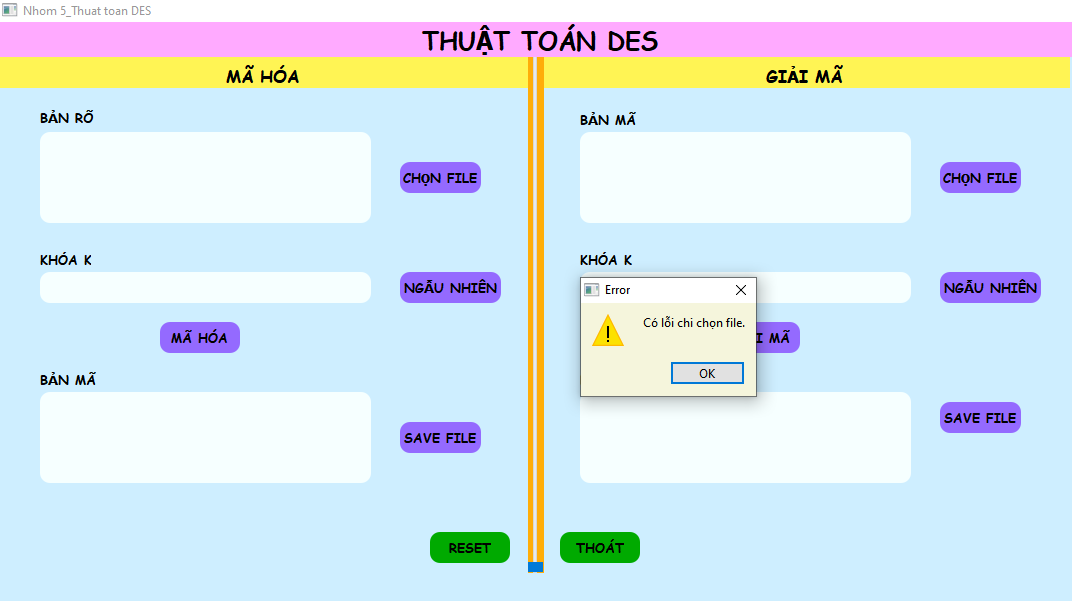


Khi file được chọn, màn hình sẽ hiển thị thông báo “Đã lấy dữ liệu từ file” khi đó dữ liệu file sẽ được lấy và hiển thị lên ô Bản rõ

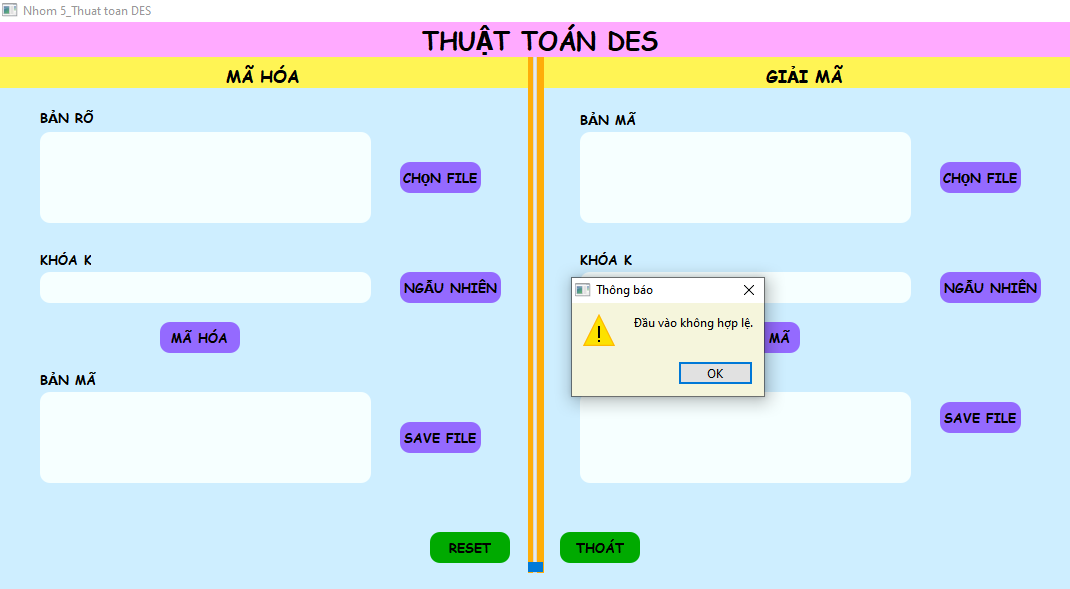




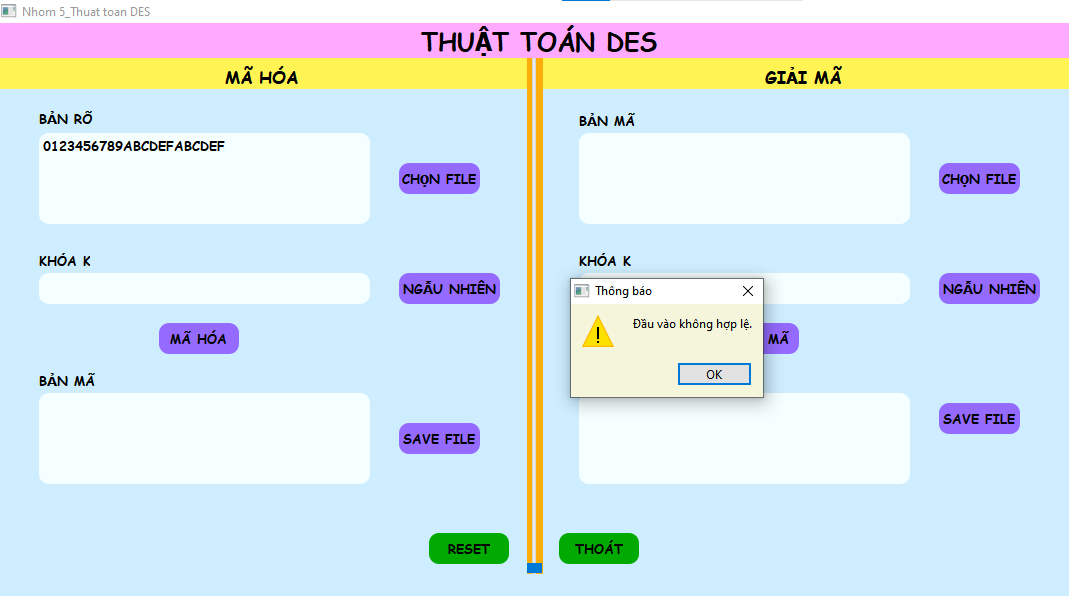
Nếu ấn vào nút “Hủy” khi chọn file màn hình sẽ hiển thị ra thông báo “Lỗi khi chọn file”



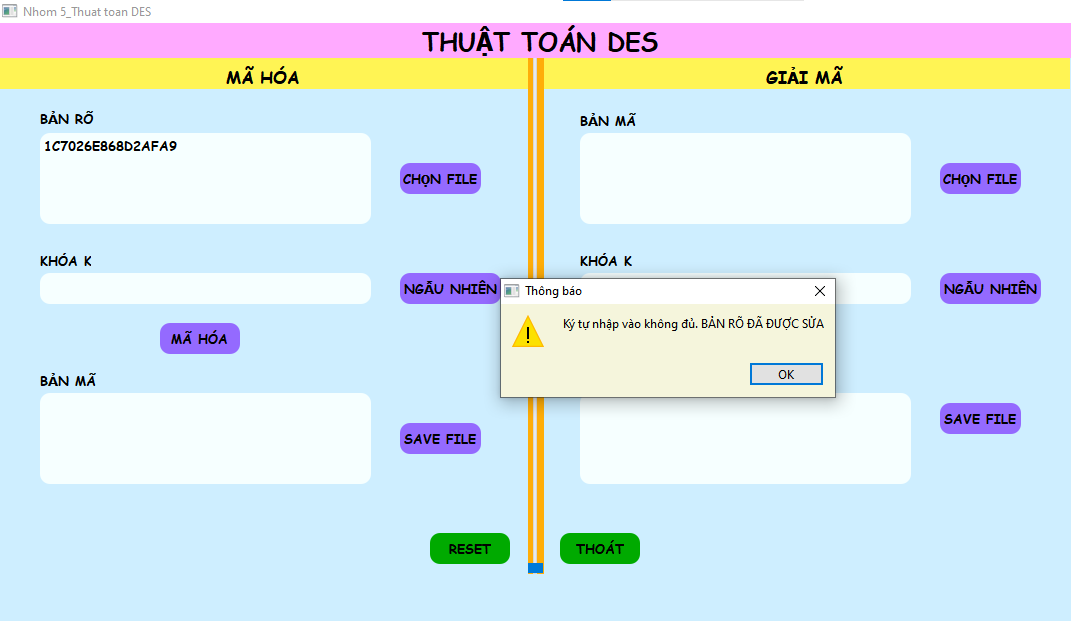
Nếu Bản rõ để trống tức là chưa được nhập hoặc chưa được chọn file, khi thực hiện mã hóa màn hình sẽ hiển thị ra thông báo “Đầu vào không hợp lệ”



Nếu Bản rõ vượt quá 16 ký tự, màn hình sẽ hiển thị ra thông báo “Đầu vào không hợp lệ”

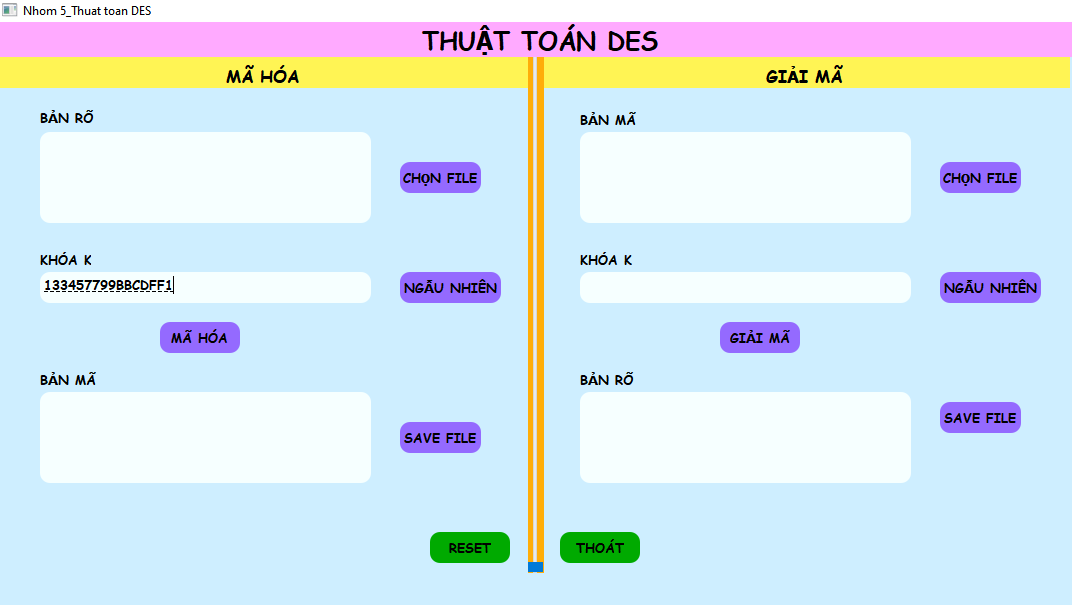


Nếu Bản rõ nhập không đủ 16 ký tự, khi đó nó sẽ được bổ sung các ký tự bất kì để đủ 16 ký tự và thông báo ra màn hình “Ký tự nhập không đủ. BẢN RÕ ĐÃ ĐƯỢC SỬA”



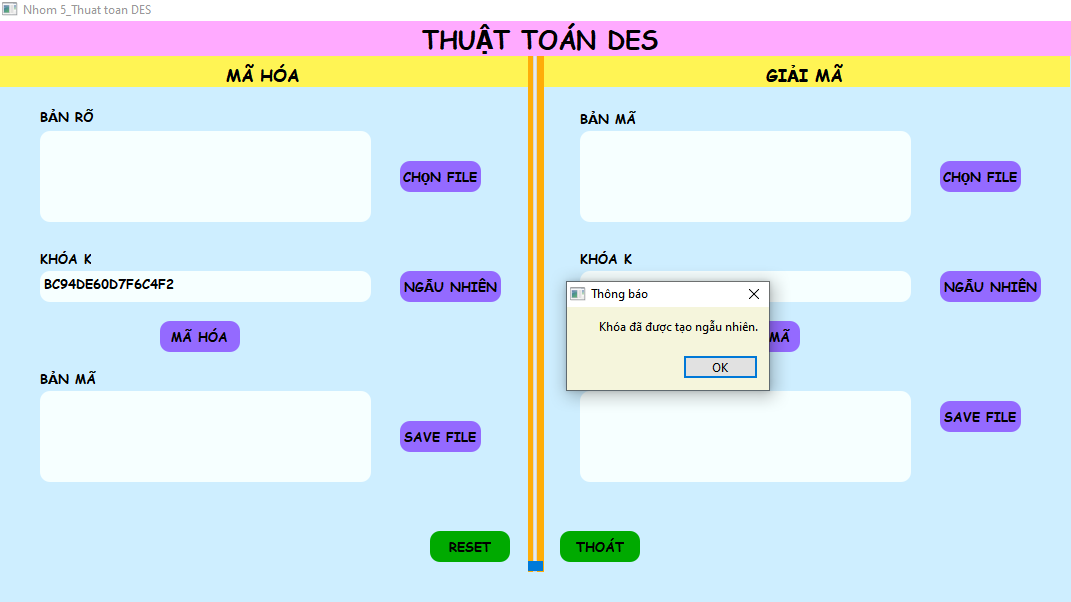
***Bước 2: Tạo khóa K***

Cách 1: Nhập khóa K vào ô text



Cách 2: Tạo khóa ngẫu nhiên

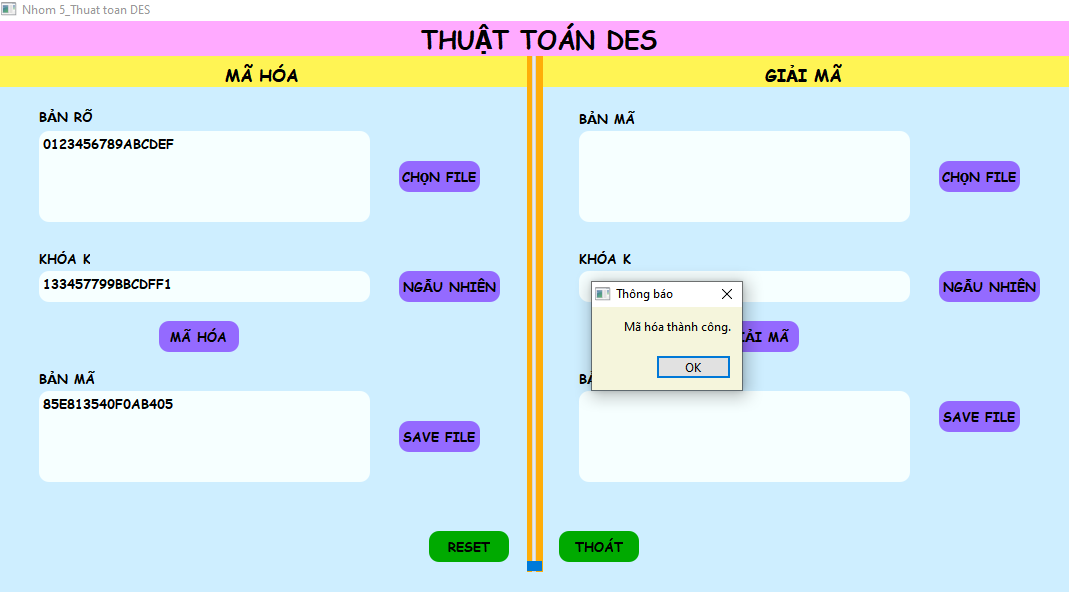
Khi ấn vào nút “Ngẫu nhiên” khóa ngẫu nhiên sẽ được tạo và thông báo ra màn hình “Khóa đã được tạo ngẫu nhiên”



Tương tự như bản rõ, nếu khóa K chưa được tạo hoặc vượt quá 16 ký tự sẽ thông báo “Dữ liệu đầu vào không hợp lệ”, nếu khóa K được nhập nhưng chưa đủ ký tự.

***Bước 3: Mã hóa***

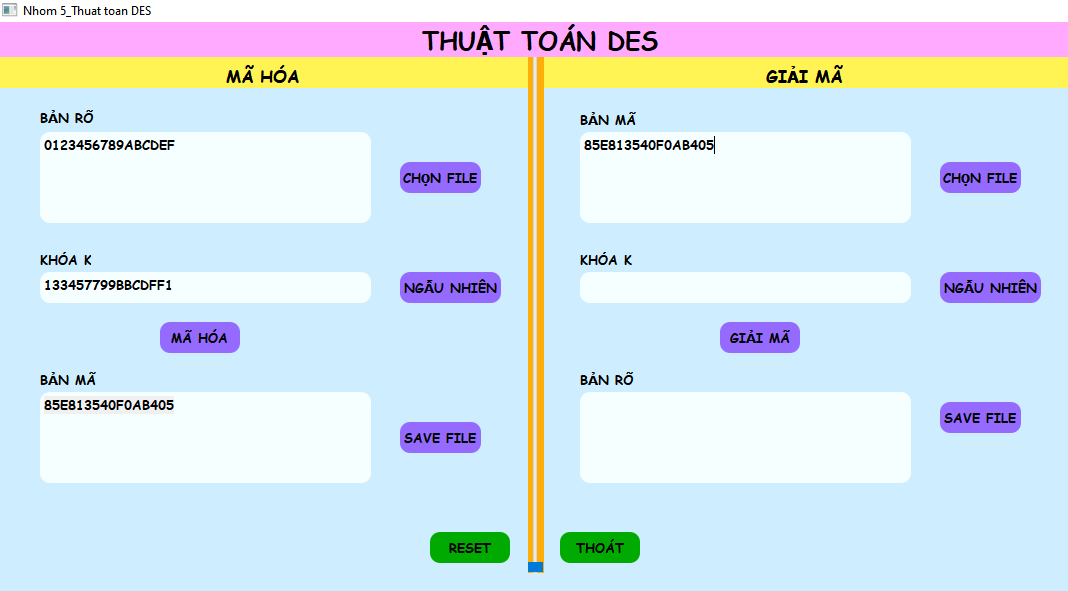
Khi ấn vào nút “Mã hóa”, chương trình sẽ thực hiện mã hóa và đưa ra kết quả ra ô bản mã và thông báo “Mã hóa thành công”



* **Giải mã**

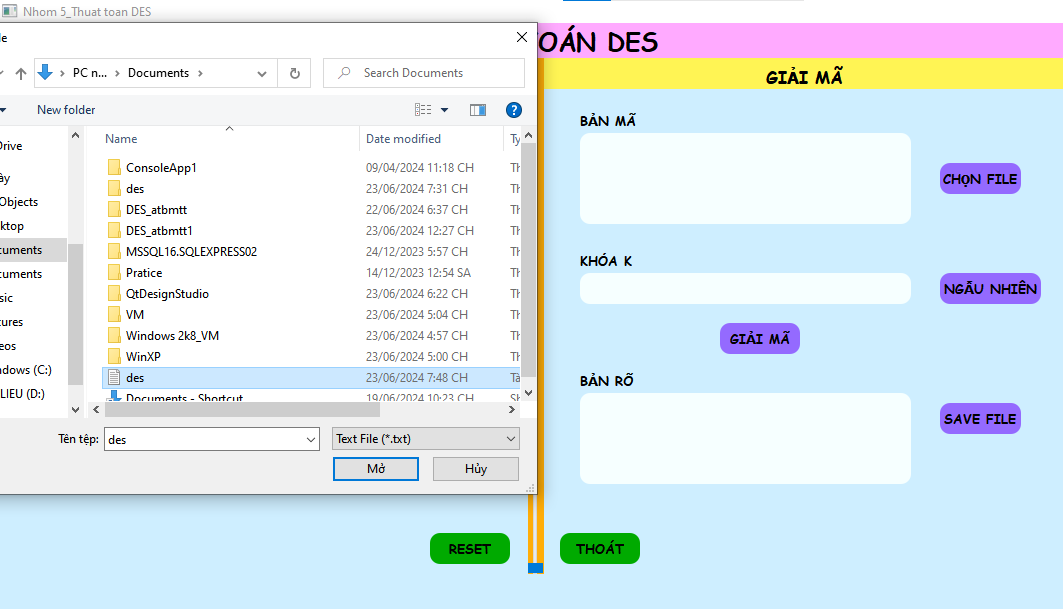
***Bước 1: Tạo bản mã***

Cách 1: Nhập trực tiếp bản mã vào ô text

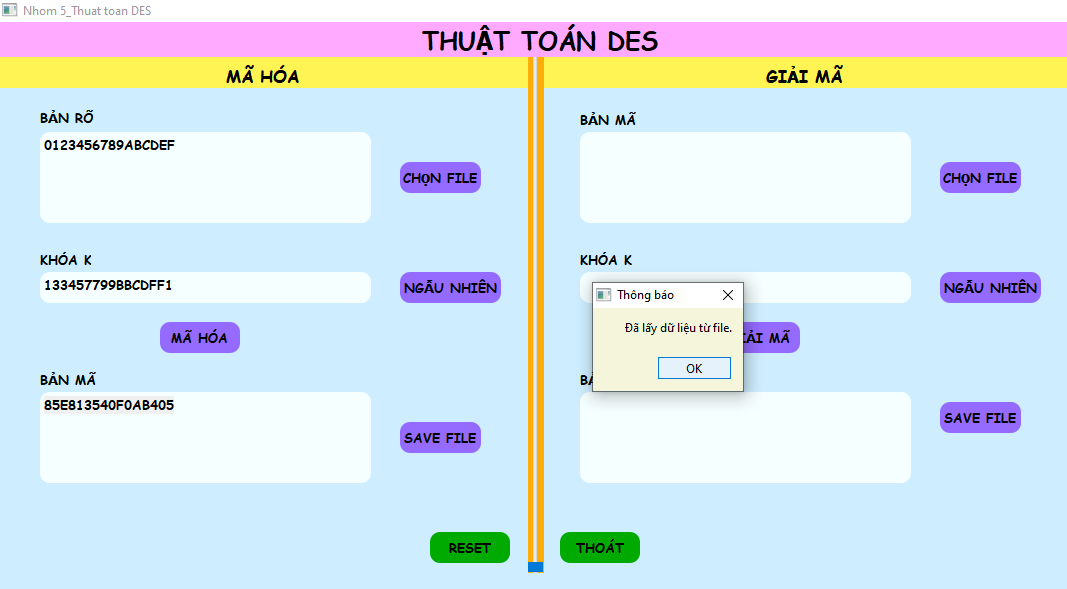


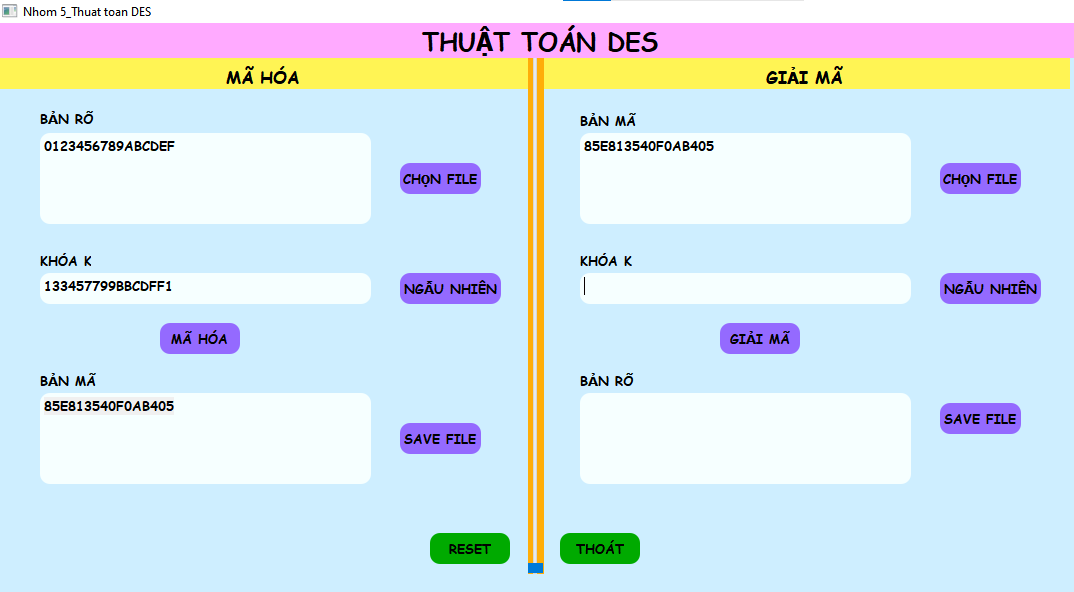
Cách 2: Bấm vào nút “ Chọn File”

Khi bấm vào nút “ Chọn File”, màn hình sẽ hiển thị ra các file. Khi đó hãy chọn 1 file .txt bất kì đã có dữ liệu của bản mã để thực hiện giải mã.

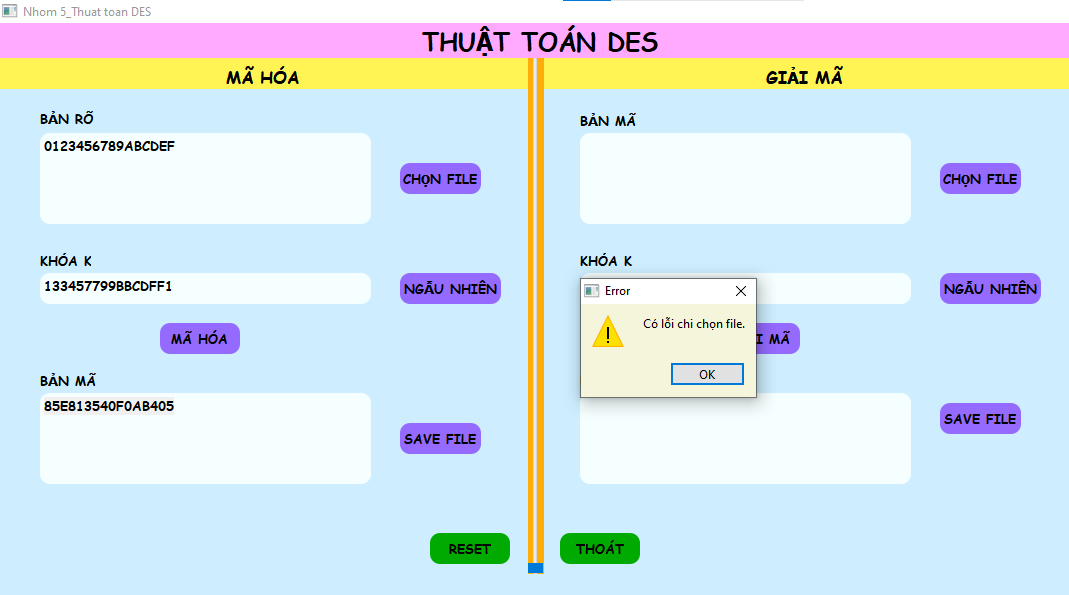


Khi file được chọn, màn hình sẽ hiển thị thông báo “Đã lấy dữ liệu từ file” khi đó dữ liệu file sẽ được lấy và hiển thị lên ô Bản mã

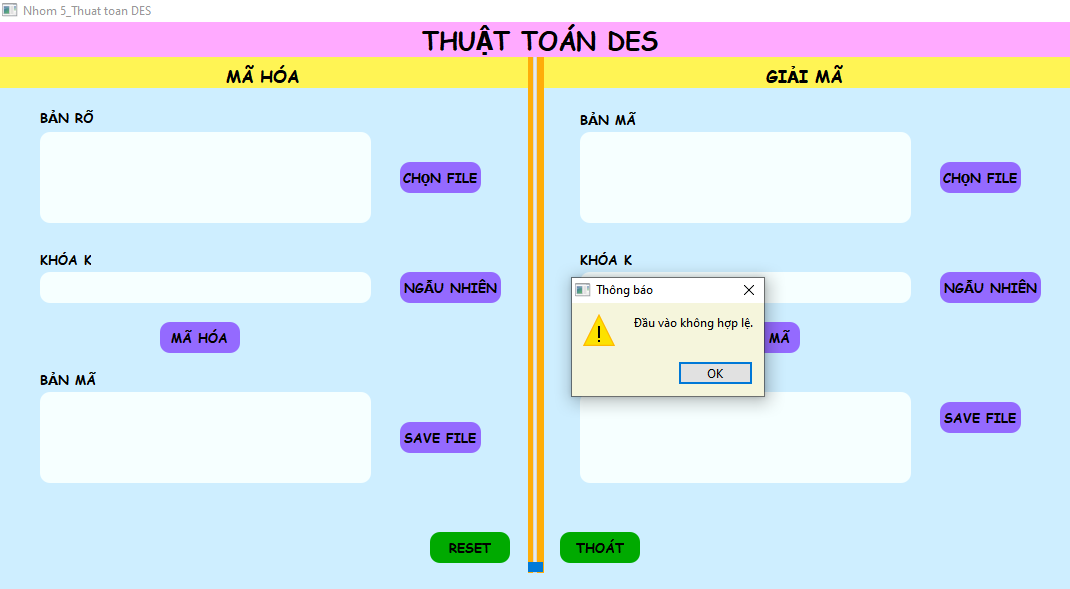




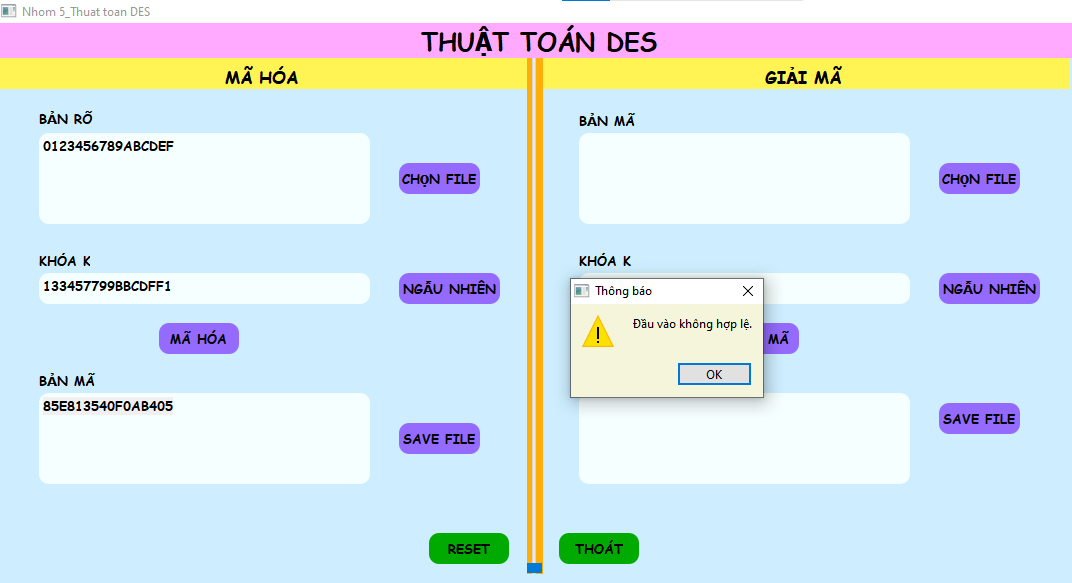
Nếu ấn vào nút “Hủy” khi chọn file màn hình sẽ hiển thị ra thông báo “Lỗi khi chọn file”



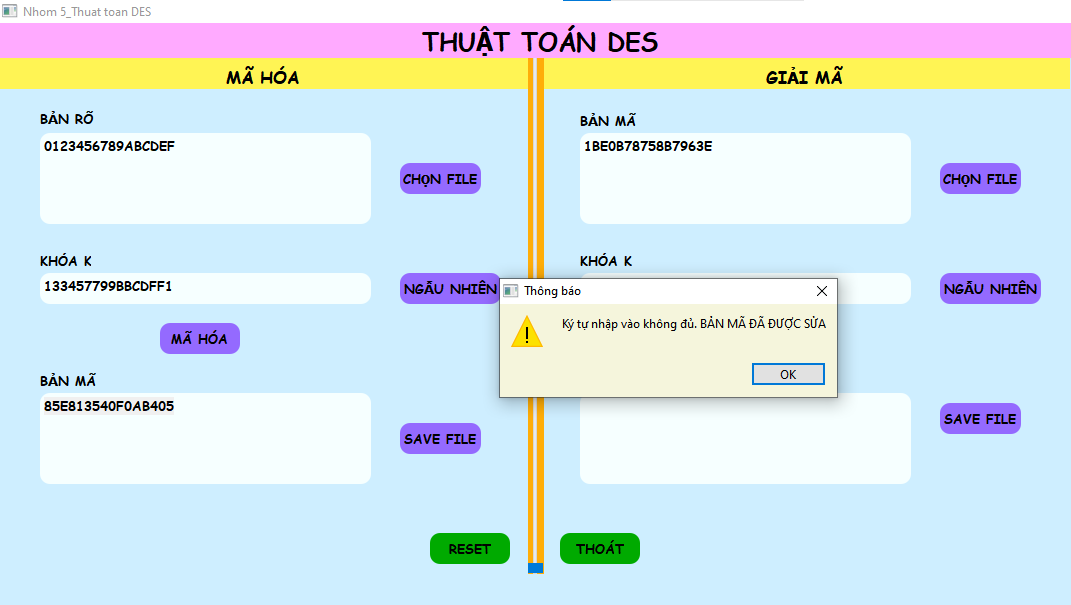
Nếu Bản mã để trống tức là chưa được nhập hoặc chưa được chọn file, khi thực hiện giải mã màn hình sẽ hiển thị ra thông báo “Đầu vào không hợp lệ”



Nếu Bản mã vượt quá 16 ký tự, màn hình sẽ hiển thị ra thông báo “Đầu vào không hợp lệ”

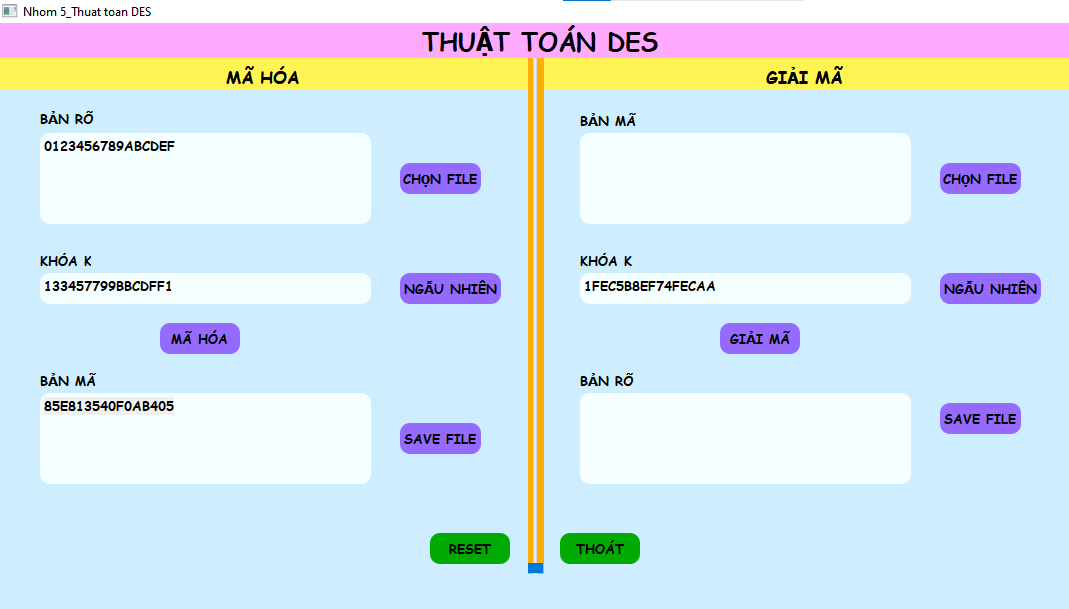


Nếu Bản mã nhập không đủ 16 ký tự, khi đó nó sẽ được bổ sung các ký tự bất kì để đủ 16 ký tự và thông báo ra màn hình “Ký tự nhập không đủ. BẢN MÃ ĐÃ ĐƯỢC SỬA”



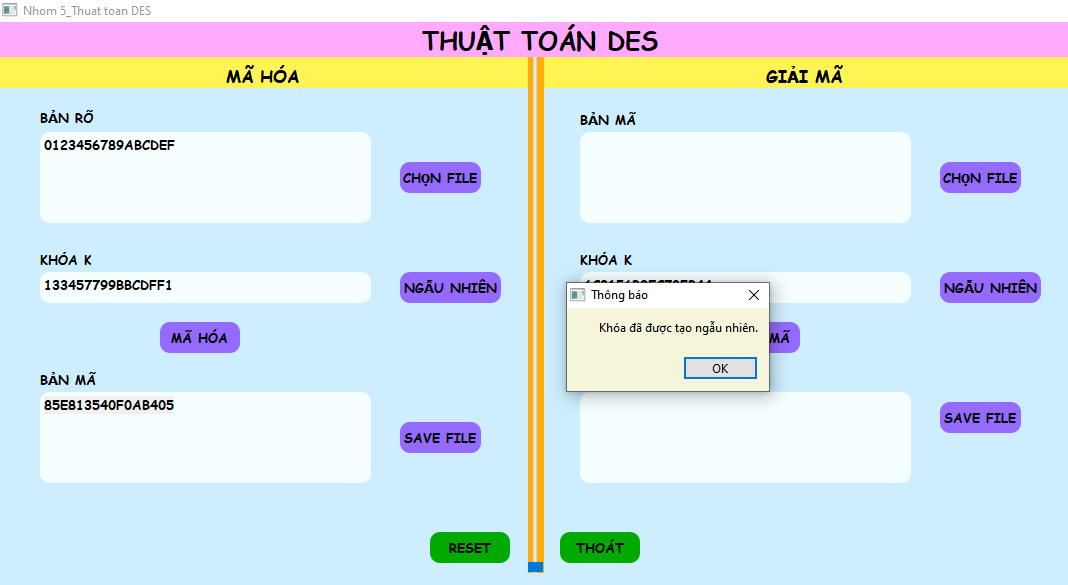
***Bước 2: Tạo khóa K***

Cách 1: Nhập khóa K vào ô text



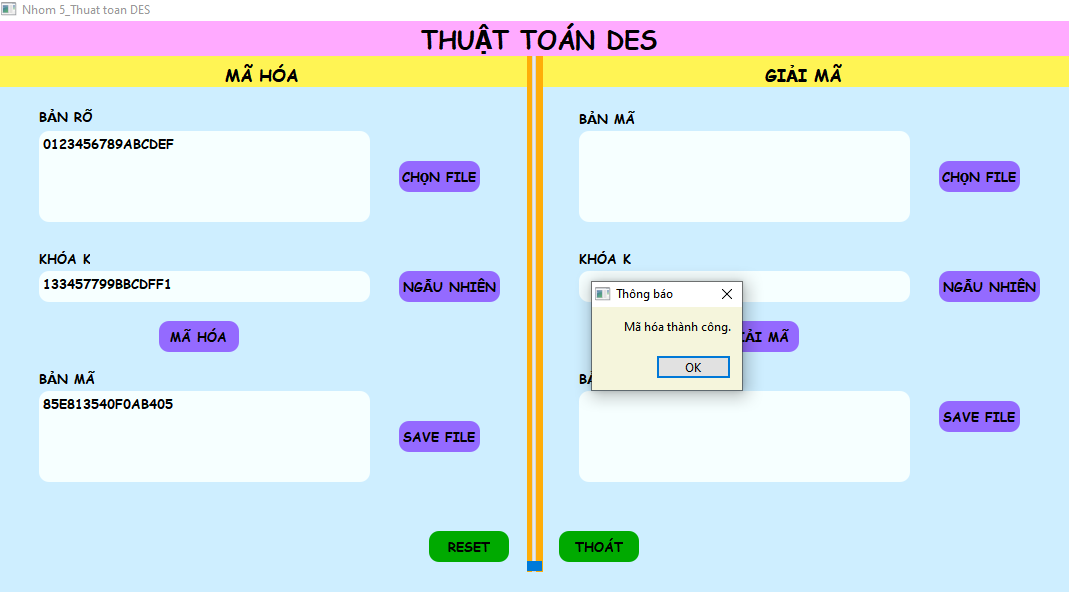
Cách 2: Tạo khóa ngẫu nhiên

Khi ấn vào nút “Ngẫu nhiên” khóa ngẫu nhiên sẽ được tạo và thông báo ra màn hình “Khóa đã được tạo ngẫu nhiên”



***Bước 3: Mã hóa***

Khi ấn vào nút “Mã hóa”, chương trình sẽ thực hiện mã hóa và đưa ra kết quả ra ô bản mã và thông báo “Mã hóa thành công”



# 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.

Trong quá trình thực hiện đề tài mã hóa và giải mã DES, nhóm đã học được rất nhiều kiến thức và kỹ năng quan trọng bao gồm cả lý thuyết và thực hành.

### 3.1.1. Kiến thức lý thuyết

1. Lý thuyết mã hóa DES: Hiểu hơn về thuật toán mã hóa DES, từ các bước phân chia khối, các phép thế và hoán vị, cho đến các vòng lặp Feistel. DES (Data Encryption Standard) là một trong những thuật toán mã hóa khóa đối xứng đầu tiên được chấp nhận rộng rãi, do đó việc hiểu rõ các cơ chế hoạt động của DES là cực kỳ quan trọng.
2. Khái niệm mật mã học: Nắm vững các khái niệm cơ bản trong mật mã học, bao gồm mã hóa đối xứng, khóa bí mật, và sự an toàn của các thuật toán mã hóa.
3. Chuyển đổi hệ cơ số: Kỹ năng chuyển đổi giữa các hệ cơ số (nhị phân, thập phân, và thập lục phân) để xử lý dữ liệu mã hóa. Điều này bao gồm việc chuyển đổi văn bản sang dạng nhị phân để mã hóa và ngược lại.

### 3.1.2. Kỹ năng lập trình

1. Lập trình C#, C++: Cải thiện kỹ năng lập trình trong ngôn ngữ C#, C++ đặc biệt là làm việc với chuỗi nhị phân và các thao tác bit. Hiểu biết về cấu trúc dữ liệu và thuật toán trong C# được củng cố thông qua việc triển khai các bước mã hóa và giải mã.

### 3.1.3. Kỹ năng nghiên cứu, phân tích và làm việc nhóm

1. Tìm kiếm, thu thập và phân tích thông tin từ các nguồn tài liệu khác nhau để hỗ trợ cho việc nghiên cứu và phát triển đề tài​​.
2. Học cách phân công công việc hợp lý và phối hợp hiệu quả giữa các thành viên trong nhóm.
3. Kỹ năng phân tích yêu cầu và thiết kế giải pháp mã hóa và giải mã hiệu quả.

## 3.2. Bài học kinh nghiệm

Trong quá trình thực hiện, nhóm đã rút ra được rất nhiều bài học và kinh nghiệm. Trước khi bắt tay vào triển khai mã nguồn, cần hiểu rõ yêu cầu và cấu trúc của thuật toán. Việc nghiên cứu kỹ lưỡng về lý thuyết DES và cách nó hoạt động là rất quan trọng để đảm bảo triển khai chính xác.

1. Lập kế hoạch và quản lý thời gian cụ thể: Việc lập kế hoạch chi tiết cho từng giai đoạn của đề tài giúp quản lý thời gian hiệu quả hơn. Điều này bao gồm việc phân chia công việc, đặt ra các mốc thời gian cụ thể và kiểm tra tiến độ thường xuyên.
2. Kỹ năng làm việc nhóm: Học cách làm việc nhóm hiệu quả giữa các thành viên trong nhóm là yếu tố quan trọng để đảm bảo sự phối hợp tốt và thống nhất trong công việc.
3. Kiểm tra và thử nghiệm kỹ lưỡng: Thử nghiệm các phần của mã nguồn một cách độc lập trước khi kết hợp lại thành hệ thống hoàn chỉnh giúp phát hiện và sửa lỗi sớm, giảm thiểu rủi ro trong giai đoạn triển khai cuối cùng.

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

1. Tính khả thi của chủ đề

Chủ đề mã hóa và giải mã DES có tính khả thi cao trong nghiên cứu và ứng dụng thực tế. DES là một trong những thuật toán mã hóa cổ điển và nền tảng trong lĩnh vực mật mã học. Việc nghiên cứu DES không chỉ giúp hiểu rõ nguyên lý mã hóa mà còn cung cấp nền tảng cho việc nghiên cứu các thuật toán mã hóa hiện đại hơn như AES.

1. Thuận lợi

* Có nhiều tài liệu, sách vở và nguồn tài nguyên trực tuyến chi tiết về DES và mật mã học. Điều này tạo điều kiện thuận lợi cho việc nghiên cứu và triển khai thuật toán.
* Kiến thức về DES có thể được áp dụng trong nhiều lĩnh vực như an ninh mạng, bảo mật thông tin và ứng dụng công nghệ thông tin. Dù DES không còn phổ biến trong thực tế hiện nay, hiểu biết về nó vẫn giúp nâng cao khả năng bảo mật tổng thể.
* DES là thuật toán mã hóa cơ bản, cung cấp nền tảng tốt để tiếp tục nghiên cứu các thuật toán mã hóa phức tạp hơn.

1. Khó khăn

* DES có cấu trúc phức tạp, yêu cầu hiểu biết sâu rộng về các phép toán bit và kỹ thuật mã hóa. Việc hiểu và triển khai các bước mã hóa như hoán vị ban đầu (IP), hoán vị cuối (IP-1), các bảng S-box, và các phép dịch chuyển vòng có thể gặp nhiều khó khăn.
* Việc triển khai và kiểm thử thuật toán cần nhiều thời gian và công sức, đòi hỏi tính kiên nhẫn và tỉ mỉ.
* DES hiện không còn được coi là an toàn cho nhiều ứng dụng do độ dài khóa ngắn. Việc nghiên cứu cần bao gồm cả việc tìm hiểu và đánh giá các phương pháp mã hóa thay thế và nâng cao. Điều này đòi hỏi sự cập nhật liên tục về các phương pháp và công nghệ mới trong lĩnh vực bảo mật.

# 

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

[1] Slide Bài giảng An toàn và Bảo mật thông tin, Ths. Trần Phương Nhung, Giảng viên Đại học Công nghiệp Hà Nội.

[2] Phan Đình Diệu (2002). Lý thuyết mật mã và an toàn thông tin.NXB Đại học quốc gia Hà Nội.

[3] Giáo trình An toàn và bảo mật thông tin (2008), Đại học Nha Trang, Biên soạn Trần Minh Văn

**Website**

[1] Secure Communicating Systems: Design, Analysis, and Implementation, Tác giả Michael Huth, Michael R. A. Huth. (Link sách)

[2] <https://vi.wikipedia.org/wiki/DES_(m%C3%A3_h%C3%B3a)>

[3] <https://en.wikipedia.org/wiki/Secret_sharing>

[4] <https://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing>

[5] https://en.wikipedia.org/wiki/Lagrange\_polynomial#Finite\_fields