**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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

**A close up of a sign

Description automatically generated**

**BÀI TẬP LỚN**

**AN TOÀN VÀ BẢO MẬT THÔNG TIN**

***ĐỀ TÀI***

***Giấu tin trong ảnh sử dụng kết hợp mã hóa AES và kỹ thuật giấu tin trên sai phân***

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

|  |  |  |
| --- | --- | --- |
| CBHD | : | ThS. Trần Phương Nhung |
| Nhóm | : | 2 |
| Mã lớp | : | *20214IT6001001* |

**Hà Nội, 2022TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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

**A close up of a sign

Description automatically generated**

**BÀI TẬP LỚN**

**AN TOÀN VÀ BẢO MẬT THÔNG TIN**

***ĐỀ TÀI***

**Giấu tin trong ảnh sử dụng kết hợp mã hóa AES và kỹ thuật giấu tin trên sai phân**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Giáo viên hướng dẫn | | : | *ThS. Trần Phương Nhung* | |
| Mã lớp | | : | *20214IT6001001* | |
| Sinh viên thực hiện | | : |  | |
|  | * Nguyễn Hoàng Anh * Hoàng Quốc Cường   - Cao mạnh Cường  ­- Lê Đình Chính | | | 2019605582  2017603385  2020601976  2019605619 |

**Hà Nội, 2022**

Mục lục

[**CHƯƠNG I** 3](#_Toc113548929)

[**1.1Tổng quan về an toàn bảo mật thông tin** 3](#_Toc113548930)

[**1.1.2 bảo vệ an toàn thông tin bằng mã hóa thông tin** 3](#_Toc113548931)

[**1.1.3 Mã hóa thông tin ở hiện tại** 5](#_Toc113548932)

[**1.1.1Mã hóa khóa công khai** 6](#_Toc113548933)

[**1.1.2Mã hóa khóa bí mật** 7](#_Toc113548934)

[**1.2Tiêu chuẩn mã hóa dữ liệu AES** 9](#_Toc113548935)

[**1.2.1Giới thiệu** 9](#_Toc113548936)

[**1.2.2Mã hóa và giải mã** 10](#_Toc113548937)

[**1.3 Tổng quan về giấu tin** 22](#_Toc113548938)

[**1.3.1 Các kĩ thuật giấu tin** 22](#_Toc113548939)

[**1.3.2 Các ứng dụng chính của giấu tin** 23](#_Toc113548940)

[**1.4 Cấu trúc ảnh bit map** 23](#_Toc113548941)

[**1.5 kĩ thuật giấu tin trên sai phân** 27](#_Toc113548942)

[**1.5.1 Điểm ảnh và hình ảnh** 27](#_Toc113548943)

[**1.5.2 Ẩn các số trên hình ảnh** 27](#_Toc113548944)

[CHƯƠNG II 32](#_Toc113548945)

[KẾT QUẢ NGHIÊN CỨU 32](#_Toc113548946)

[**2.1 Giới thiệu** 32](#_Toc113548947)

[**2.1.1. Nhiệm vụ** 32](#_Toc113548948)

[**2.2 Nội dung thuật toán** 32](#_Toc113548949)

[**2.3 Chương trình minh họa** 35](#_Toc113548950)

[**CHƯƠNG III** 37](#_Toc113548951)

[**KẾT LUẬN** 37](#_Toc113548952)

[**3.1 kiến thức lĩnh hội và bài học kinh nghiệm** 37](#_Toc113548953)

[**3.2 Hướng phát triển** 41](#_Toc113548954)

[**Tài liệu tham khảo** 42](#_Toc113548955)

**LỜI NÓI ĐẦU**

Chúng ta đang sống trong thế kỷ thứ 21, thế kỷ mà công nghệ thông tin phát triển vô cùng mạnh mẽ. Thời đại mà gần như mọi ngành mọi nghề đều bị ảnh hưởng ít nhiều bởi công nghệ, từ những bóng đèn thông minh đơn giản đến những người máy siêu AI. Cuộc cách mạng công nghệ đã, đang và sẽ vẫn tiếp tục ảnh hưởng đến kinh tế- xã hội của loài người, trong thời đại này năng lực cạnh tranh tập trung vào năng lực sáng tạo, xử lí và lưu trữ thông tin.

Với sự bùng nổ của mạng internet- công cụ chia sẻ dữ liệu mạnh mẽ nhất hành tinh, thì vấn đề an toàn bảo mật thông tin đi cùng cũng là vấn đề quan trọng được ưu tiên hàng đầu. Các nguy cơ mất an toàn thông tin từ bị tấn công an ninh mạng, hay đánh cắp dữ liệu đều mang lại thiệt hại nghiêm trọng và hậu quả khôn lường, đặc biệt là những thông tin cá nhân hay thông tin bảo mật của quốc gia. Do đó an toàn bảo mật thông tin là vô cùng quan trọng với cá nhân hay kể cả là một quốc gia.

Qua môn học an toàn bảo mật do giảng viên tận tình giảng dạy về cách thức hoạt động chung của hệ thống an toàn bảo mật thông tin cùng một số các hệ mã bảo mật cổ điển đến hiện đại. Chúng em làm báo cáo tìm hiểu về hệ mã AES, do kiến thức hạn hẹp và thời gian có hạn nên báo cáo của chúng em vẫn có nhiều thiếu sót cũng như chỉ dừng lại ở mức tìm hiểu cơ bản, chưa đi vào chuyên sâu, nhóm em chân thành cảm ơn cô Trần Phương Nhung đã giảng dạy, hướng dẫn và giúp đỡ để chúng em hoàn thành báo cáo này.

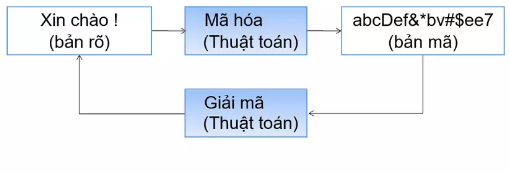
# **CHƯƠNG I**

* 1. **Tổng quan về an toàn bảo mật thông tin**

An toàn thông tin là sự bảo vệ thông tin và các hệ thống thông tin, tránh bị truy nhập, sử dụng, tiết lộ, gián đoạn, sửa đổi hoặc phá hoại trái phép nhằm bảo đảm tính nguyên vẹn, tính bảo mật và tính khả dụng của thông tin.

Bảo mật là việc bảo vệ những thứ có giá trị. Trong đó, bảo mật thông tin là một chủ đề rộng bao gồm tất cả các vấn đề bảo mật có liên quan đến lưu trữ và xử lý thông tin. Lĩnh vực nghiên cứu chính của bảo mật thông tin gồm các vấn đề pháp lý, như hệ thống chính sách, các quy định, yếu tố con người; các vấn đề thuộc tổ chức như kiểm toán xử lý dữ liệu điện tử, quản lý, nhận thức; và các vấn đề kỹ thuật như kỹ thuật mật mã, bảo mật mạng, công nghệ thẻ thông minh… Bảo mật thông tin là bảo quản tính bảo mật, tính toàn vẹn và tính sẵn có của thông tin.

**1.1.2 bảo vệ an toàn thông tin bằng mã hóa thông tin**

****

*ảnh 1.1. Mã hóa và giải mã thông tin*

- Mã hóa: Quá trình chuyển đổi dữ liệu gốc thành dữ liệu được mã hóa sao cho người khác không thể đọc hiểu được.

- Giải mã: Là quá trình ngược lại của mã hóa, biến đổi dữ liệu đã được mã hóa thành dạng gốc ban đầu.

- Bản mã: Tệp dữ liệu đã được mã hóa.

Một hệ thống mã hóa bao gồm các thành phần sau:

- PlainText : Bản tin sẽ được mã hóa hay bản tin gốc.

- CipherText : Bản tin đã được mã hóa hay bản tin mã.

- Thuật toán mã hóa và giải mã :

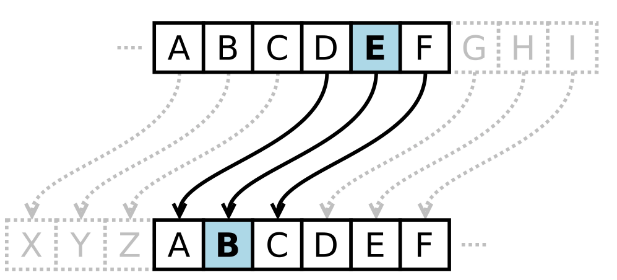
+ Encryption: quá trình chuyển bản tin gốc sang dạng mật mã.

+ Decryption: quá trình giải bản tin dạng mật mã trở về bản tin gốc.

Mã hóa thông tin là một hình thức biến đổi dữ liệu thành một dạng dữ liệu khác có ý nghĩa khác với dữ liệu trước khi bị biến đổi ban đầu, với mục đích chỉ cho phép một số người nhất định có thể đọc được dữ liệu ban đầu, thông qua việc giải mã dữ liệu sau khi biến đổi.

Biến đổi dữ liệu là một quy tắc nào đó biến đổi một lượng dữ liệu này thành một lượng dữ liệu khác. Nếu nhìn theo khía cạnh toán học, thì biến đổi dữ liệu chính là một dạng hàm số y = f(x) với x là dữ liệu ban đầu, y là dữ liệu sau khi biến đổi từ dữ liệu x và f là hàm biến đổi.

Trong mật mã học, khi nghiên cứu về mã hóa thông tin, dữ liệu ban đầu được gọi là Plaintext (kí hiệu là P), dữ liệu sau khi mã hóa được gọi là Ciphertext (kí hiệu là C), hàm biến đổi/mã hóa được gọi là phương pháp mã hóa và được kí hiệu là E (Encryption).



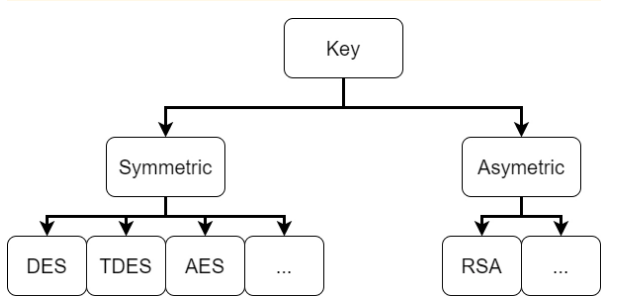
*Ảnh 1.2 Biến đổi dữ liệu đơn giản của Hệ mã Caesar*

### **1.1.3 Mã hóa thông tin ở hiện tại**

Trong mật mã học, mã hóa thông tin được chia làm hai loại là mã hóa khóa đối xứng (Symmetric-key) và mã hóa khóa công khai (Asymmetric cryptography hay Public-key cryptography).

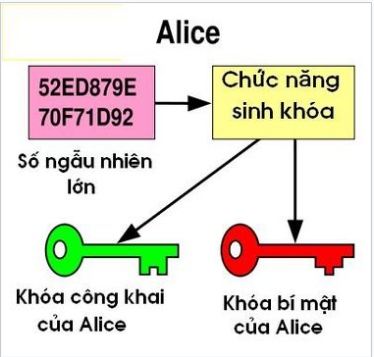
Mã hóa khóa đối xứng sử dụng cùng một loại chìa khóa trong việc mã hóa và giải mã. Tuy nhiên, chìa khóa phải đảm bảo bí mật giữa hai bên trao đổi thông tin. Nếu như bên thứ ba (Eve) có được chìa khóa thì sẽ có thể sử dụng để giải mã thông tin được truyền đi. Một số loại mã hóa khóa đối xứng nổi tiếng như DES, AES,...

Mã hóa khóa công khai sử dụng hai loại khóa khác nhau: khóa công khai là khóa không cần giữ bí mật với các bên khác và khóa bí mật, loại khóa chỉ được biết bởi chính người sở hữu. Một số loại mã hóa có thể kể đến là RSA,...



*Ảnh 1.4 Hai loại khóa mã hóa*

* + 1. **Mã hóa khóa công khai**

****

*Ảnh 1.5 khóa công khai*

Mật mã hóa khóa công khai là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật).

Thuật ngữ mật mã hóa khóa bất đối xứng thường được dùng đồng nghĩa với mật mã hóa khóa công khai mặc dù hai khái niệm không hoàn toàn tương đương. Có những thuật toán mật mã khóa bất đối xứng không có tính chất khóa công khai và bí mật như đề cập ở trên mà cả hai khóa (cho mã hóa và giải mã) đều cần phải giữ bí mật.

Trong mật mã hóa khóa công khai, khóa cá nhân phải được giữ bí mật trong khi khóa công khai được phổ biến công khai. Trong 2 khóa, một dùng để mã hóa và khóa còn lại dùng để giải mã. Điều quan trọng đối với hệ thống là không thể tìm ra khóa bí mật nếu chỉ biết khóa công khai.

Hệ thống mật mã hóa khóa công khai có thể sử dụng với các mục đích:

* Mã hóa: giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
* Tạo chữ ký số: cho phép kiểm tra một văn bản có phải đã được tạo với một khóa bí mật nào đó hay không.
* Thỏa thuận khóa: cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên.

Thông thường, các kỹ thuật mật mã hóa khóa công khai đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng những lợi điểm mà chúng mang lại khiến cho chúng được áp dụng trong nhiều ứng dụng.

* + 1. **Mã hóa khóa bí mật**

Mã hóa khóa bí mật, còn gọi là mã hóa đối xứng hay mã hóa khóa riêng, là sử dụng một khóa cho cả quá trình mã hóa (được thực hiện bởi người gửi thông tin) và quá trình giải mã (được thực hiện bởi người nhận).

Quá trình mã hóa khóa bí mật được thực hiện như sau:

Một khách hàng (Anne) muốn gửi tới người bán hàng (Bob) một đơn đặt hàng, nhưng chỉ muốn một mình Bob có thể đọc được. Anne mã hóa đơn đặt hàng (dưới dạng văn bản gốc) của mình bằng một mã khóa rồi gửi đơn đặt hàng đã mã hóa đó cho Bob.

Tất nhiên, ngoài Bob và Anne ra, không ai có thể đọc được nội dung thông điệp lộn xộn đã mã hóa.

Khi nhận được thông điệp mã hóa, Bob giải mã thông điệp này bằng khóa giải mã và đọc các thông tin của đơn đặt hàng. Điều đáng chú ý là trong kĩ thuật mã hóa khóa bí mật, khóa để mã hóa thông điệp và khóa để giải mã thông điệp không giống như nhau.

Người gửi thông điệp sử dụng một khóa mật mã để mã hóa thông điệp và người nhận thông điệp cũng sử dụng một khóa như vậy để đọc mật mã hoặc giải mã thông điệp.

Kĩ thuật mã hóa khóa bí mật này đã được IBM phát triển, áp dụng cho các cơ quan của Chính Phủ Mỹ năm 1977 được gọi là Tiêu chuẩn mã hóa dữ liệu (DES- data encryption standard).

* 1. **Tiêu chuẩn mã hóa dữ liệu AES**
     1. **Giới thiệu**

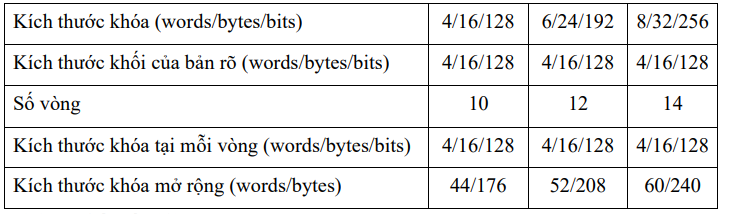
Vào năm 1999, cục tiêu chuẩn quốc gia Hoa Kỳ (NIST) đã ban hành một phiên bản mới của tiêu chuẩn DES chỉ ra rằng DES chỉ nên được sử dụng cho các hệ thống kế cũ và DES ba lần được sử dụng. DES ba lần có hai ưu điểm đảm bảo cho việc sử dụng rộng rãi trong vài năm tới. Đầu tiên, với độ dài khóa 168-bit, nó khắc phục được lỗ hổng đối với cuộc tấn công vét cạn của DES. Thứ hai, thuật toán mã hóa cơ bản trong DES ba lần cũng giống như trong DES. Thuật toán này đã được giám sát kỹ lưỡng hơn bất kỳ thuật toán mã hóa nào khác trong một khoảng thời gian dài và không có cuộc tấn công phá mã hiệu quả nào dựa trên thuật toán thay vì vét cạn được tìm thấy. Do đó, DES ba lần có khả năng chống phá mã rất tốt. Nếu bảo mật là yếu tố duy nhất được xem xét, thì DES ba lần sẽ là lựa chọn thích hợp cho thuật toán mã hóa tiêu chuẩn trong nhiều thập kỷ tới.

Hạn chế chính của DES ba lần là thuật toán tương đối chậm trong phần mềm. DES ban đầu được thiết kế để triển khai bằng phần cứng giữa những năm 1970 và không tạo ra mã phần mềm hiệu quả. DES ba lần, có số vòng gấp ba lần DES, do đó thực hiện chậm hơn DES ban đầu. Một nhược điểm phụ là cả DES và DES ba lần đều sử dụng kích thước khối 64-bit. Vì lý do cả hiệu quả và bảo mật, kích thước khối lớn hơn là cần thiết.

Vì những nhược điểm này, DES ba lần không phải là ứng cử viên thích hợp để sử dụng lâu dài. Để thay thế, vào năm 1997 NIST đã đưa ra lời kêu gọi đề xuất Tiêu chuẩn mã hóa nâng cao (AES) mới, tiêu chuẩn này phải có sức mạnh bảo mật bằng hoặc tốt hơn DES ba lần và cải thiện đáng kể hiệu quả. Ngoài các yêu cầu chung này, NIST quy định rằng AES phải là mật mã khối đối xứng với độ dài khối 128 bit và hỗ trợ độ dài khóa có thể là 128, 192 và 256 bit.

Trong vòng đánh giá đầu tiên, 15 thuật toán được đề xuất đã được chấp nhận. Vòng thứ hai thu hẹp còn 5 thuật toán. NIST đã hoàn thành quá trình đánh giá của mình và xuất bản tiêu chuẩn cuối cùng vào tháng 11 năm 2001. NIST đã chọn Rijndael làm thuật toán AES được đề xuất. Hai nhà nghiên cứu đã phát triển và gửi Rijndael cho AES đều là những nhà mật mã học đến từ Bỉ: Tiến sĩ Joan Daemen và Tiến sĩ Vincent Rijmen.

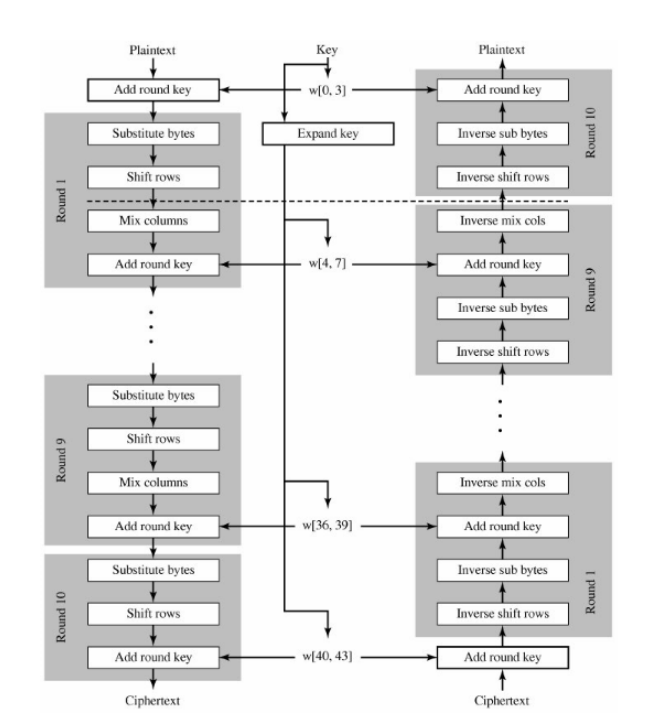
Cuối cùng, AES được thiết kế để thay thế DES ba lần, nhưng quá trình này sẽ mất một số năm. NIST dự đoán rằng DES ba lần vẫn sẽ là một thuật toán được sử dụng trong tương lai gần

****

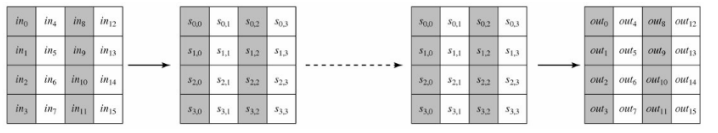
*Ảnh 1.6 Tham số của AES*

* + 1. **Mã hóa và giải mã**

Cấu trúc tổng thể của mã hóa AES được mô tả trên hình 4.10. 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 hình1.6. 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 1.7. 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 1.7 cấu trúc và giải mã AES*

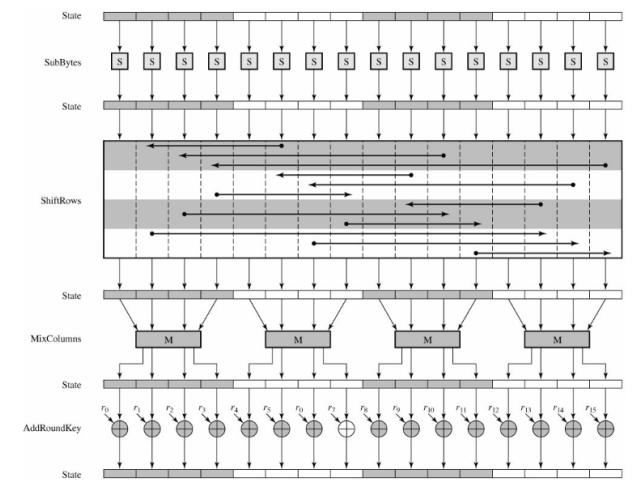


*Hình 1.8 Đầu vào, mảng trạng thái và đầu ra*

**

*Hình 1.9 khóa và mở rộng khóa*

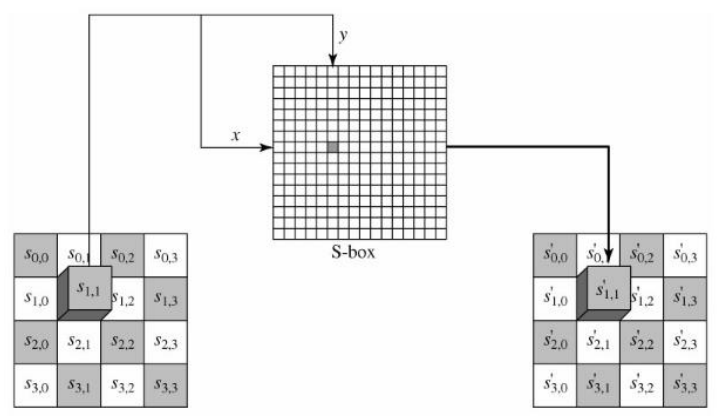
Cấu trúc của thuật toán AES tương đối đơn giản. Cả thuật toán mã hóa và giải mã đều bắt đầu giai đoạn AddRoundKey, tiếp theo là 9 vòng, mỗi vòng đầy đủ 4 giai đoạn: Thay thế các bytes (Substitute bytes) sử dụng hộp S để thực hiện việc thay thế từng byte của khối; dịch các dòng (ShiftRows) đơn giản là 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 1.10 minh họa một vòng mã hóa đầy đủ



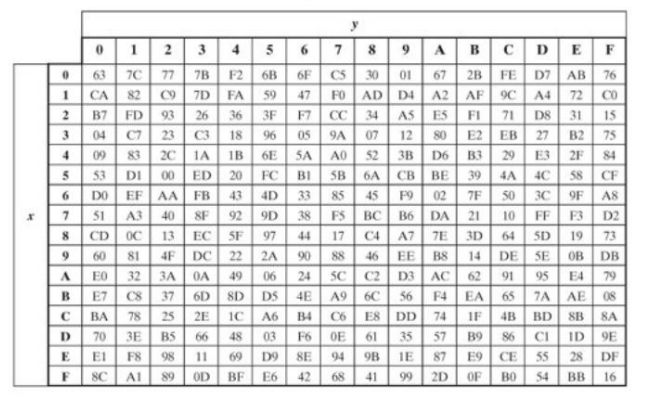
*Hình 1.10 Minh họa một vòng mã AES*

**1.4.2.1 thay thế byte**

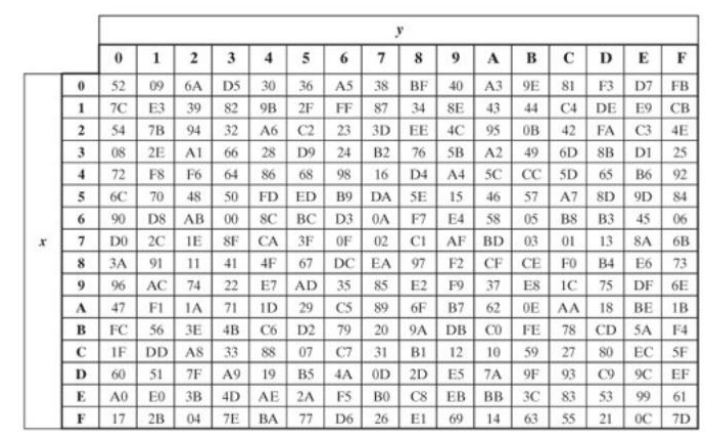
Thay thế byte đơn giản chỉ là tra cứu trong bảng 16 x 16, mỗi ô là 1 byte và được gọi là hộp S như bảng 1.12 và hộp S đảo bảng1.13. Minh họa việc tra cứu hộp S như hình



Hình 1.11 Phép thay thế byte sử dụng hộp S



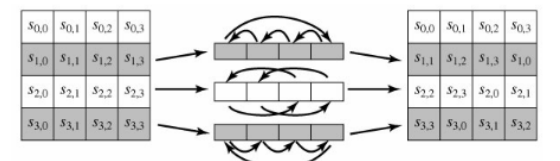
Bảng 1.12 hộp S



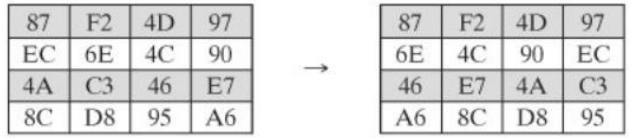
Bảng 1.13 Hộp S đảo

**1.4.2.2** **Dịch dòng**

Hình 1.14 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 1.15 ví dụ minh họa phép dịch dòng. Hình 1.14. Minh họa phép dịch dòng Hình 1.15. 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.



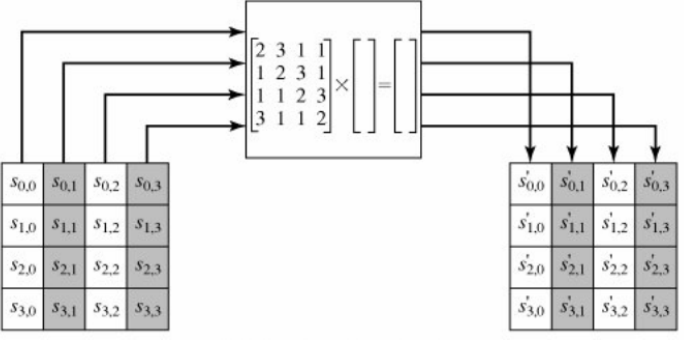
Hình 1.14



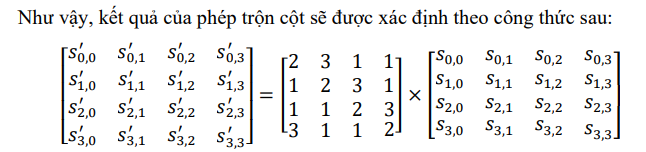
Hình 1.15

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.

**1.4.2.3 Trộn cột**

****

Hình 1.16 minh họa phép trộn cột



Áp dụng phép nhân hai ma trận ta thu được:

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

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

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

𝑠3,𝑗′ = (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:

2. 𝑠𝑖,𝑗 = { 𝑏6𝑏5𝑏4𝑏3𝑏2𝑏10 𝑛ế𝑢 𝑏7 = 0

𝑏6𝑏5𝑏4𝑏3𝑏2𝑏10 + 00011011 𝑛ế𝑢 𝑏7 = 1

3 ∙ 𝑠𝑖,𝑗 = 𝑠𝑖,𝑗 + 2 ∙ 𝑠𝑖,𝑗

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. 𝑠0,0′ = 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

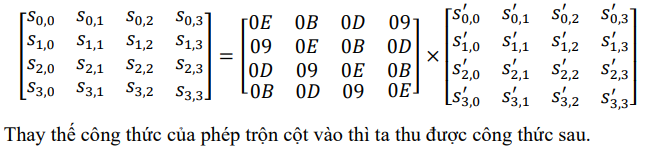
A6 = 1 0 1 0 0 1 1 0

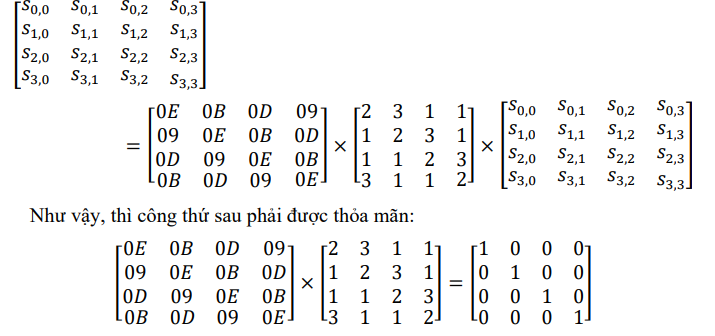
XOR 0 1 0 0 0 1 1 1 = 47h

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:





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

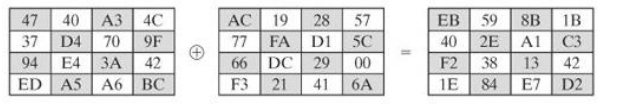
0D = 0 0 0 0 1 1 0 1

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.

**1.4.2.4 Cộng với khóa**

****

Hình1.17 ví dụ minh họa phép cộng khóa

**1.4.2.5 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

}

}

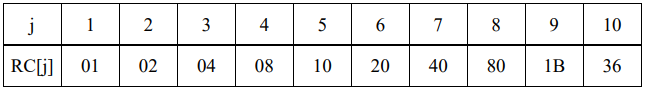
Trong đó, phép toán RotWord là thực hiện phép quay trái 1 byte, tức là đầu vào l từ

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 1.18 ở hệ thập lục phân (hexadecimal).



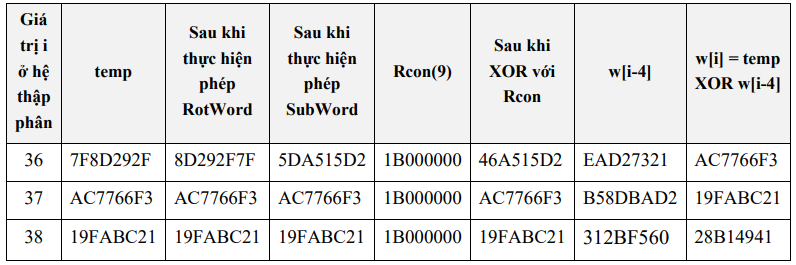
Bảng 1.18 minh họa cách xác định khóa của một vòng

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

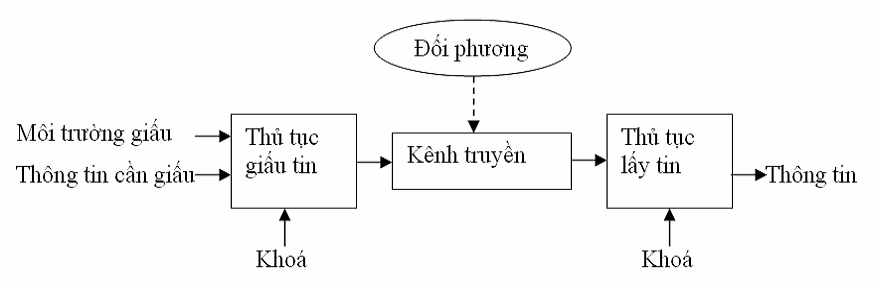
tại vòng 9 được xác định như bảng sau:



Bảng 1.19 ví dụ xác định khóa tại vòng 8

## **1.3 Tổng quan về giấu tin**

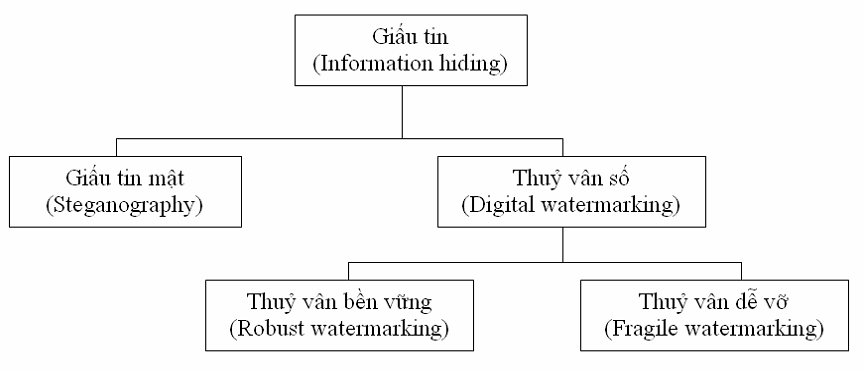
Để thực hiện giấu tin cần xây dựng được các thủ tục giấu tin. Các thủ tục này sẽ thực hiện nhúng thông tin cần giấu vào môi trường giấu tin. Các thủ tục giấu tin thường được thực hiện với một khóa giống như trong các hệ mật mã để tăng tính bảo mật. Sau khi giấu tin ta thu được chứa thông tin giấu và có thể phân phối đối tượng đó trên kênh thông tin. Để giải mã thông tin cần nhận được đối tượng có chứa thông tin đã giấu,sử dụng thủ tục giải mã cùng với khóa đã dùng trong quá trình giấu để lấy lại thông tin.



Hình 2.1 mô hình giấu và lấy tin

### **1.3.1 Các kĩ thuật giấu tin**

Các kĩ thuật giấu tin mới được chú ý phát triển mạnh trong khoảng 10 năm trở lại đây,nên việc phân loại còn chưa được thống nhất. Sơ đồ phân loại sau được đưa ra năm 1999 và được nhiều người chấp nhận:



Hình 2.2: Phân loại các mô hình giấu tin

Theo sơ đồ này, giấu tin được chia thành hai hướng chính là giấu tin mật và thủy vân số.Giấu tin mật quan tâm chủ yếu đến lượng tin có thể giấu,còn thủy vân số quan tâm đến tính bền vững của thông tin giấu. Trong từng hướng chính lại được chia ra các hướng nhỏ hơn,chẳng hạn với thủy vân số thì có thủy vân bền vững và thủy vân dễ vỡ. Thủy vân bền vững cần được bảo toàn được các thông tin thủy vân trước các tấn công như dịch chuyển ,cắt xén,xoay đối với ảnh. Ngược lại thủy vân dễ vỡ cần phải dễ bị phá hủy khi gặp các tấn công nói trên.

### **1.3.2 Các ứng dụng chính của giấu tin**

**Giấu tin mật (Steganography)**: các thông tin cần bảo mật được giấu trong các đối tượng vỏ và các đối tượng này có thể được truyền công khai tới người nhận mà không gây bất cứ sự chú ý nào của đối phương. Người nhận sẽ sử dụng thuật toán và khóa nào đó( đã thỏa thuận giữa 2 người) để khôi phục lại thông tin mật. Yêu cầu kĩ thuật là tỉ lệ giấu tin cần lớn nhưng bằng các giác quan không thể nhận thấy được sự khác biệt của đối tượng trước và sau khi giấu tin vào.

**Bảo vệ bản quyền (Copyright Protection):** đây là ứng dụng phổ biến nhất của thủy vân số. Một thông tin nào đó mang ý nghĩa quyền sở hữu tác giả(gọi là thủy vân) được nhúng vào trong các sản phẩm số. Yêu cầu kĩ thuật là việc nhúng thủy vân không ảnh hưởng đáng kể đến cảm nhận sản phẩm và phải bền vững trước các tấn công,tồn tại lâu dài cùng sản phẩm.

**Xác thực thông tin (authentication):** một thông tin được giấu trong đối tượng số để nhận biết xem đối tượng đó có bị thay đổi hay không. Yêu cầu kĩ thuật là việc nhúng thủy vân không ảnh hưởng đáng kể đến cảm nhận đối tượng và rất dễ bị phá hủy trước các tấn công. Đây là ứng dụng của thủy vân dễ vỡ.

## **1.4 Cấu trúc ảnh bit map**

**Ảnh BITMAP** là định dạng ảnh do microsoft đề xuất , có phần mở rộng là BMP , loại ảnh này truyền tải , sử dụng rộng rãi trên máy tính , và các thiết bị điện tử khác. Ảnh bitmap được chia thành ba dạng : ảnh nhị phân ( ảnh đen trăng ) , ảnh đa mức xám , ảnh màu.

· **Ảnh đen trắng** : là ảnh mà mỗi điểm ảnh chỉ thể hiện một trong hai trạng thái 0 và 1 để biểu diễn trạng thái điểm ảnh đen hay trắng.

· **Ảnh đa mức xám** : là ảnh mà mỗi điểm ảnh được biểu diễn bởi một giá trị và đó là cường độ sáng của điểm ảnh.

· **Ảnh màu** : là ảnh mà mỗi điểm ảnh được biểu diễn bởi ba đại lượng R, G, B . Số lượng màu có thể của loại ảnh này lên tới 265^3 màu khác nhau.Nhưng số lượng màu trên thực tế của một ảnh nào đó thường khá nhỏ. Để tiết kiệm bộ nhớ với các ảnh có số lượng màu nhỏ hơn 256 thì màu các điểm ảnh được lưu trữ dưới dạng bảng màu.Với ảnh có số màu lớn thì các điểm ảnh không tổ chức dưới dạng bảng màu , khi đó giá trị của các điẻm ảnh chinh là giá trị của các thành phần màu R,G,B. Với ảnh có số lượng màu lớn ,tùy theo chất lượng ảnh mà quyết định số bit để biểu diễn cho mỗi màu thường là 24 bit, hoặc 32 bit . Với ảnh 24 bit mỗi thành phần màu được biểu diễn bởi một byte(8 bit).

Ảnh bitmap đựợc lưu trữ dưới dạng nhị phân, một tệp dạng bitmap được chia thành các phần cơ bản như :

-Phần tiêu để tệp (Bitmap header)

-Thông tin về ảnh (Bitmap Infor)

-Bảng màu (Palette Table)

-Vùng dữ liệu(Data)

Thứ tự được lưu trữ trong bộ nhớ như sau: Bitmap Header->Bitmap Infor->Palette Table->Data

**Ý nghĩa của các phần trong tệp ảnh bitmap**.

-Bitmap Header: Mô tả thông tin chung về tệp định dạng bitmap, độ lớn của phần này cố định với mọi tệp bitmap.

-Bitmap Infor: Mô tả thông tin về ảnh được lưu trữ, độ lớn của phần này cố định.

-Pallete Table: Bảng màu của ảnh bitmap, độ lớn của phần này có thể bằng không ( không có bảng màu) đối với ảnh đen trắng và ảnh màu có số lượng màu lớn hơn 256 màu.

-Data: Thông tin về từng điểm ảnh , độ lớn của phần này phụ thuộc vào kich thước ảnh. Phần Data lưu trữ ảnh theo hướng từ dưới lên trên và từ trái qua phải.

Kích thước và giá trị các trường trong tệp ảnh

**Bitmap Header**

Phần này có độ lớn cố định là 14 bytes , phần này dùng để mô tả thông tin chung về tệp như :

Kiểu tệp, độ lớn và một số thông tin liên qua đến tệp.



*Bảng : Ý nghĩa từng trường trong vùng Bitmap Header*

Trong phần Bitmap Header có mô tả thông tin về độ dài tệp , thông tin này thực sự cần thiết với mọi chương trình .Tuy nhiên qua thực nghiệm thấy rằng với một số tệp thông tin này không chính xác. Do đó trong báo cáo đưa cách tính kích thước tệp bitmap thông qua công thức:

Tệp\_Size=Sizeof(Bitmap Header ) + Sizeof(Bitmap Infor) + Sizeof(Pallete) + Sizeof(Data)

Địa chỉ vị (offset) của vùng dữ liệu có thể được xác định thông qua công thức :

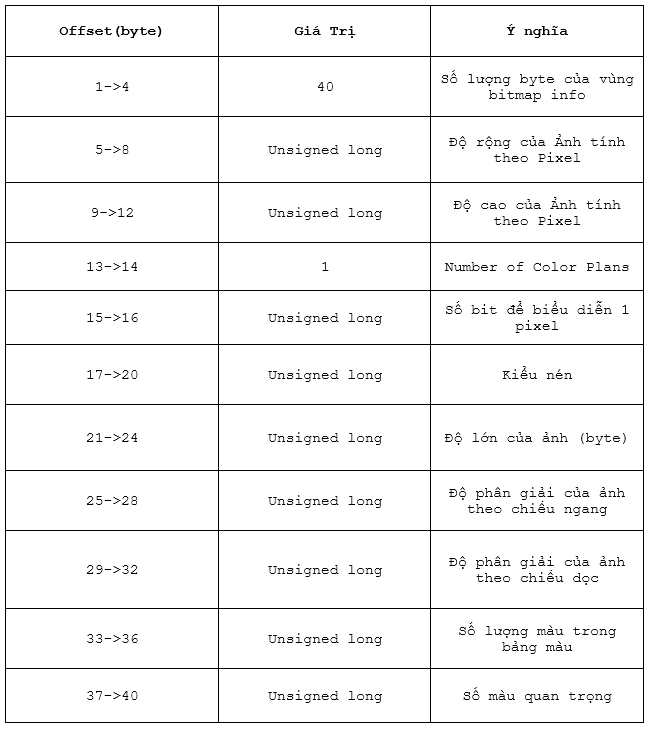
Địa chỉ vùng data = 54+ Sizeof(Pallete)

Đối với ảnh đen trắng và ảnh màu có số lượng màu lớn hơn 256 thì giá trị địa chỉ vị cố định là 54

**Bitmap Infor**:

Phần bitmap infor dùng để mô tả thông tin về ảnh đang dùng được lưu trữ trong tệp kích thước của phần này cố định là 40 byte.

Ý nghĩa và giá trị của từng trường trong vùng Bitmap Infor được mô tả chi tiết trong bảng :



*Ý nghĩa từng trường trong vùng BitmapInfor*

## **1.5 kĩ thuật giấu tin trên sai phân**

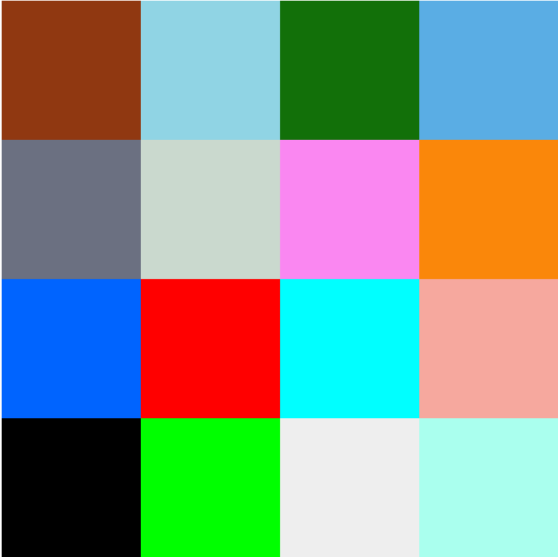
### **1.5.1 Điểm ảnh và hình ảnh**

Trước khi bắt tay vào công cuộc mã hóa hình ảnh với các thông điệp bí mật, trước tiên chúng ta phải hiểu bản chất của cách hình ảnh được biểu diễn trong máy tính

Điểm ảnh có thể được coi là đơn vị nhỏ nhất cấu thành một hình ảnh. Về bản chất, nó là một hình vuông nhỏ có màu được kết hợp với các hình vuông có màu khác (tức là các pixel khác) để hoàn thành một hình ảnh, điều quan trọng về pixel là chúng được thể hiện bằng các con số . Những con số này có thể ở dạng mã RGB (Đỏ, Xanh lá cây, Xanh lam), mã hex (mã thập lục phân), HSL (Hue, Saturation, Lightness) hoặc một số mã khác.

Vậy, điều rút ra chính ở đây là màu sắc được biểu thị bằng các con số. Những con số này - giống như tất cả những con số mà chúng ta biết - có thể được chuyển đổi sang hệ nhị phân. Và ẩn một thông điệp bên trong hình ảnh thường liên quan đến việc thao tác các chữ số có nghĩa nhỏ nhất trong các số nhị phân này.

### **1.5.2 Ẩn các số trên hình ảnh**



Hình ảnh gốc không có thông báo ẩn

Hình ảnh này bao gồm 16 pixel với nhiều màu sắc khác nhau và được sắp xếp theo định dạng 4x4, tức là nó có 4 hàng và 4 cột.

Ẩn các số chỉ có thể được biểu diễn bằng 3 chữ số nhị phân tức là các số 0-7.

Vì mã RGB chỉ cho phép 3 số riêng biệt (đỏ, lục và lam). Và ẩn từng bit nhị phân của số bị ẩn của trong bit quan trọng nhất của mỗi giá trị RGB.

Giả sử muốn ẩn số "6" trong pixel đầu tiên của hình ảnh. Mã RGB cho pixel đầu tiên là rgb (144, 56, 17).

Đầu tiên, chuyển đổi các giá trị RGB thành nhị phân.

nhị phân của 6 (số muốn ẩn) là 110 . Ẩn bit ngoài cùng bên trái của 110 (tức là 1 )trong bit ngoài cùng bên phải của giá trị nhị phân cho màu đỏ.

Đỏ = 1001 000 1 (ban đầu là 1001 0000) = 145

Tương tự, bit ở giữa (tức là 1 khác ) được ẩn trong bit ngoài cùng bên phải của giá trị nhị phân cho màu xanh lá cây.

tức là Green = 0011 100 1 (ban đầu là 0011 1000) = 57

Và cuối cùng, bit ngoài cùng bên phải của 110 (tức là 0 ) được ẩn trong bit ngoài cùng bên phải của giá trị nhị phân cho màu xanh lam

tức là Blue = 0001 000 0 (ban đầu là 0001 0001) = 16

Điều này dẫn đến pixel được mã hóa là rgb (145, 57, 16) .



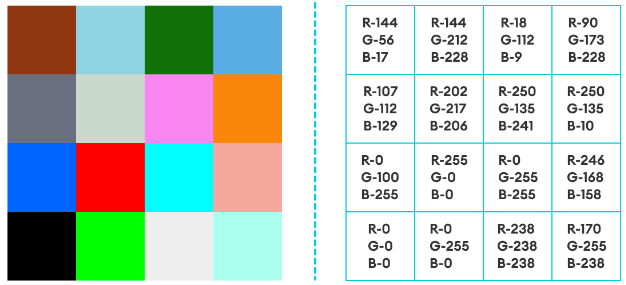
Hình ảnh 1 pixel được mã hóa bằng số 6( không thể nhận ra khác biệt bằng mắt thường)

Làm tương tự với các pixel còn lại, ta được hình ảnh sau khi mã hóa thông tin ẩn

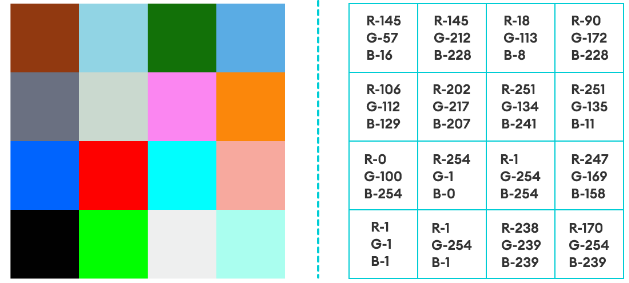


Hai hình ảnh trông gần giống hệt nhau. Tuy nhiên, chúng khác nhau một cách tinh tế do sự sửa đổi tinh vi đối với mã RGB của mỗi pixel.

Hình ảnh bên dưới hiển thị cả hình ảnh gốc cùng với các giá trị RGB của mỗi pixel:



Hình ảnh dưới đây cho thấy hình ảnh được mã hóa với các giá trị RGB tương ứng của mỗi pixel:



Giải mã hình ảnh sẽ cho chúng ta những con số:

6420135702467531

Nếu chúng ta xem xét hình ảnh trên và thuật toán đơn giản này kỹ hơn một chút, chúng ta có thể thấy rõ rằng nếu một thành phần của mã RGB là một số chẵn, thì số đó tương ứng với số nhị phân 0 , trong khi một số lẻ tương ứng với số nhị phân 1.

Để làm rõ hơn điểm này, chúng ta hãy xem mã RGB của hai pixel ngẫu nhiên, giả sử các pixel trong hàng 1 cột 3 (chúng ta hãy gọi nó là R1C3) và hàng 3 cột 4 (R3C4).

Đối với Pixel R1C3 ,

Đỏ = 18 (chẵn) ----> 0

Màu xanh lá cây = 113 (lẻ) ----> 1

Xanh lam = 8 (chẵn) ----> 0

Kết hợp các số này từ đỏ sang xanh, chúng ta nhận được 010 , là mã nhị phân của 2 .

Khi kiểm tra thông điệp bí mật, số 2 ở cùng một vị trí được đại diện bởi pixel (xem thông điệp bí mật ở trên hoặc hình ảnh của giải pháp cuối cùng bên dưới).

Tương tự, đối với Pixel R3C4 ,

Đỏ = 247 (lẻ) ----> 1

Màu xanh lá cây = 169 (lẻ) ----> 1

Xanh lam = 158 (chẵn) ----> 0

Kết hợp các số này từ đỏ sang xanh, ta nhận được 110 , là mã nhị phân của 6 . là số đã được mã hóa ở trên

Điều này này rất quan trọng vì nó có thể rút ngắn đáng kể thời gian dành cho việc giải mã hình ảnh, vì điều này có nghĩa là không cần phải chuyển đổi mã RGB sang hệ nhị phân. Thay vào đó, chúng ta chỉ xem số đó là số chẵn hay lẻ và gán các giá trị cho phù hợp. Sau đó, chuyển đổi số nhị phân cuối cùng, được giải mã thành số thập phân.

Vì vậy, thay vì tổng cộng 4 chuyển đổi số trên mỗi pixel (3 từ thập phân sang nhị phân cho màu đỏ, xanh lục, xanh lam; và 1 từ nhị phân sang thập phân cho thông điệp nhị phân được giải mã cuối cùng), ta chỉ cần thực hiện 1 chuyển đổi số (thập phân- thành nhị phân) trên mỗi pixel.

# **CHƯƠNG II**

# **KẾT QUẢ NGHIÊN CỨU**

## **2.1 Giới thiệu**

### **2.1.1. Nhiệm vụ**

1. Tìm hiểu mật mã AES.

2. Tìm hiểu về dấu tin trong ảnh và cấu trúc ảnh bitmap.

3. Nghiên cứu kỹ thuật giấu tin trên sai phân.

4. Tìm hiểu ngôn ngữ lập trình Matlab 7.7, Java,C#,C++

5. Ứng dụng xây dựng chương trình giấu tin trong ảnh sử dụng kết hợp mã

hóa AES và kỹ thuật giấu tin trên sai phân.

6. Demo chương trình

## **2.2 Nội dung thuật toán**

**Thuật toán giấu tin**

Đầu vào :

➢ Ảnh sử dụng để giấu tin.

➢ Thông điệp.

Đầu ra :

➢ Ảnh đã giấu tin.

Các bước thực hiện :

**Bước 1**: Sử dụng kỹ thuật mã hóa AES cho thông điệp cần giấu. Sau đó chuyển chuỗi thông điệp đã mã hóa sang nhị phân.

Bước 2: Từ ảnh cấp xám dùng để giấu tin ta có ma trận ảnh tương ứng. Tính giá

trị di bằng cách trừ hai điểm ảnh liên tiếp cho nhau di = gi +1 - gi, với gi, gi +1 ∈

[0,. . . , 255], i=1:mxn (với m và n là kích cỡ ảnh).

**Bước 3:** Các giá trị tuyệt đối | di|, | di| ∈ [0,. . . , 255], được phân loại vào các

phạm vi Ri (Ri gồm các miền giá trị sau [0 7], [8 15], [16 31], [32 63], [64 127]

và [128 255]), với i = 1, 2,. . . n. Cận trên cận dưới và độ rộng của mỗi Ri được

ký hiệu lần lượt là li, ui và wi. Giả sử | di

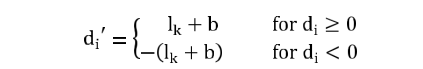
| thuộc Rk với k ∈ [1,2. . . 6].

**Bước 4:** Ta xác định được số bit sẽ nhúng theo công thức 1

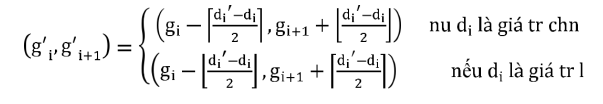


Sau đó chọn n bit từ chuỗi bit cần giấu giấu vào mỗi giá di

**Bước 5:** Tính lại d được theo công thức 2



**Bước 6:** Tính các giá trị màu xám mới (gi, gi+1) cho ảnh sau khi giấu tin theo công thức 3.



Cuối cùng ta được ảnh đã giấu tin.

**Thuật toán tách tin**

Đầu vào :

➢ Ảnh đã giấu tin.

Đầu ra :

➢ Thông điệp đã giấu.

Các bước thực hiện :

**Bước 1:** Từ ảnh đã giấu tin ta có ma trận ảnh tương ứng. Tính giá trị di bằng

cách trừ hai điểm ảnh liên tiếp cho nhau di = gi +1 - gi, với gi, gi +1 ∈ [0,. . . , 255],

i=1:mxn (với m và n là kích cỡ ảnh)

**Bước 2:** Xác định phạm vi Ri của mỗi giá trị tuyệt đối | di | được li, ui và wi. Từ

đó tính được số bit thông tin đã giấu n = .

Giá trị thập phân đã giấu vào di được tính bằng công thức (4).

Chuyển b sang n bit nhị phân ta được chuỗi bit thông điệp đã giấu.

**Bước 3:** Lặp lại bước 2 cho đến khi tách được số bit đã giấu.

**Bước 4**: Chuyển chuỗi bit thông điệp đã giấu ở dạng nhị phân sang thập phân ta được

chuỗi thông điệp mã hóa đã giấu ở dạng thập phân.

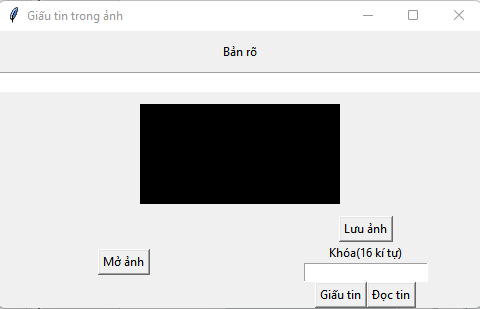
**Bước 5:** Sử dụng kỹ thuật giải mã AES cho chuỗi thông điệp mã hóa đã giấu ta được

chuỗi thông điệp cần tách.

## **2.3 Chương trình minh họa**

* Chương trình gồm các chức năng sau:
* Mở ảnh trong máy
* Mã hóa ảnh với khóa nhập từ bàn phím
* Lưu ảnh sau khi mã hóa
* Đọc tin đã mã hóa từ ảnh trước đó

Kết quả cài đặt:



Ngôn ngữ cài đặt: Python

Môi trường cài đặt: Visual code

Thư viện bổ sung :

* numpy==1.17.1
* opencv-python==4.1.1.26
* Pillow==6.2.0
* pycryptodome==3.9.0

- Mã hóa và giải mã

+ Người sử dụng nhập hay chọn các thông tin (file) cần mã hóa vào ô bản rõ

+ Người sử dụng chọn chức năng mã hóa (hoặc giải mã tương ứng). Nếu đầu vào là bản rõ cần mã hóa thì sẽ mã hóa thông tin bản rõ vào trong hình ảnh, và lưu ảnh . Ngược lại nếu cần giải mã sẽ chọn ảnh đã mã hóa trước đó và ấn giải mã, thông tin bản rõ sẽ hiển thị trong ô bản rõ

+ Nếu người dùng không nhập khóa hoặc khóa sai, thiếu thì chương trình thông báo nhập khóa. Ngược lại, chương trình tiến hành mã hóa,giải mã theo khóa đã nhâp trước đó.

## **2.4 Phân công công việc**

|  |  |  |
| --- | --- | --- |
| Tên sinh viên | Tên công việc | Ngôn Ngữ  Demo |
| Nguyễn Hoàng Anh | Tìm hiểu về tiêu chuẩn mã hóa dữ liệu AES | Java |
| Hoàng Quốc Cường  (Thư ký) | Tìm kĩ thuât giấu tin trên sai phân | Python |
| Cao Mạnh Cường | Tổng quan về an toàn bảo mật thông tin | C++ |
| Lê Đình Chính  (nhóm trưởng) | Tìm hiểu về cấu trúc ảnh bitmap | PHP |
|  |  |  |

# **CHƯƠNG III**

# **KẾT LUẬN**

## **3.1 kiến thức lĩnh hội và bài học kinh nghiệm**

**3.1.1 Nội dung đã thực hiện**

* + Thông qua bài tập lớn nhóm em thực hiện , nhóm em đã học hỏi được thêm rất nhiều kiến thức và những kí năng bổ ích, cụ thể:
  + Nắm được định nghĩa về an toàn và bảo mật thông tin:
    - Đảo bảo an toàn thông tin
    - - Bảo mật thông tin và tầm quan trọng, cũng như các phương pháp đảm bảo an toàn thông tin
  + Các phương pháp mã hóa thông tin cơ bản , cấu trúc của hệ mã hóa thông tin
  + Các thành phần của hệ mã hóa cổ điển, hiện đại
  + Tìm hiểu về hệ mã hóa hiện đại
  + Tìm hiểu và nắm được hiểu biết về 2 loại:
    - mã hóa khóa đối xứng (Symmetric-key)
    - Mã hóa khóa công khai(Asymmetric cryptography hay Public-key cryptography).
  + Nắm được mục đích sử dụng chính của hệ mã khóa công khai
  + Mã hóa: giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
  + Tạo chữ ký số: cho phép kiểm tra một văn bản có phải đã được tạo với một khóa bí mật nào đó hay không.
  + Thỏa thuận khóa: cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên.
  + Nắm được ưu nhược điểm của mã hóa khóa đối xứng và khóa công khai:
  + **-**Các thuật toán mã hóa đối xứng có tốc độ nhanh hơn và đòi hỏi sức mạnh tính toán thấp hơn, nhưng nhược điểm khi sử dụng loại mã hóa này là cần gửi khóa. Vì phương thức này sử dụng cùng một khóa để mã hóa và giải mã thông tin, nên khóa đó phải được gửi cho bất kỳ ai cần truy cập dữ liệu, và điều này sẽ tự nhiên mở ra các nguy cơ về bảo mật
  + **-**Ngược lại, mã hóa bất đối xứng giải quyết vấn đề gửi khóa vì phương thức này sử dụng khóa công khai để mã hóa và khóa cá nhân để giải mã. Tuy nhiên, đổi lại, các hệ thống mã hóa bất đối xứng có tốc độ chậm hơn nhiều so với hệ thống đối xứng và đòi hỏi nhiều khả năng tính toán hơn vì khóa của chúng có độ dài lớn hơn nhiều.
* Nắm được các ứng dụng của mã hóa khóa đối xứng:
* Do có tốc độ nhanh hơn nên mã hóa đối xứng được sử dụng rộng rãi để bảo vệ thông tin trong các hệ thống máy tính hiện đại.
* Ví dụ, Hoa Kỳ sử dụng Tiêu chuẩn Mã hóa Nâng cao (AES) để mã hóa thông tin mật và nhạy cảm. AES đã thay thế Tiêu chuẩn Mã hóa Dữ liệu (DES) cũ, tiêu chuẩn này được phát triển vào những năm 1970 như một tiêu chuẩn cho mã hóa đối xứng.
  + -Mã hóa bất đối xứng
    - -Có thể sử dụng mã hóa bất đối xứng cho các hệ thống mà trong đó nhiều người dùng có thể cần mã hóa và giải mã tệp hoặc bộ dữ liệu, đặc biệt trong trường hợp không có giới hạn về tốc độ và sức mạnh tính toán. Ví dụ, có thể sử dụng phương thức mã hóa này trong hệ thống email được mã hóa, trong đó khóa công khai được sử dụng để mã hóa các email và khóa cá nhân được sử dụng để giải mã chúng.
  + -Hệ thống kết hợp
    - -Mã hóa đối xứng và bất đối xứng được sử dụng kết hợp trong nhiều hệ thống. Ví dụ, một số hệ thống kết hợp điển hình là các giao thức mã hóa Security Sockets Layer (SSL) và Transport Layer Security (TLS), được thiết kế cho giao tiếp bảo mật trong mạng Internet. Các giao thức SSL hiện được coi là không an toàn và nên được ngừng sử dụng. Ngược lại, các giao thức TLS được coi là an toàn và đã được sử dụng rộng rãi bởi tất cả các trình duyệt web chính.
  + -Nắm được hiểu biết về chuẩn mã hóa dữ liệu AES, ưu nhược điểm của AES
    - -cũng như các hệ mã hóa khác , AES cũng bị tác động với phương pháp tấn công kênh bên( side channel attack) tức là việc tấn công lên các hệ thống thực hiện thuật toán có nguy cơ làm lộ dữ liệu.
    - Chính vì việc mô tả toàn học khác đơn giản nên nhiều nhà mật mã học lo sợ rằng AES sẽ bị phá trong tương lai gần với sự trợ giúp của các công nghệ hiện đại hơn.

**-**Nắm được tổng quan về các kĩ thuật giấu tin phổ biến:

* + -Giấu tin mật: các thông tin cần bảo mật được giấu trong các đối tượng vỏ và các đối tượng này có thể được truyền công khai tới người nhận mà không gây bất cứ sự chú ý nào của đối phương. Người nhận sẽ sử dụng thuật toán và khóa nào đó( đã thỏa thuận giữa 2 người) để khôi phục lại thông tin mật. Yêu cầu kĩ thuật là tỉ lệ giấu tin cần lớn nhưng bằng các giác quan không thể nhận thấy được sự khác biệt của đối tượng trước và sau khi giấu tin vào.
    - Thủy vân số:
  + -Thủy vân bền vững: Bảo vệ bản quyền (Copyright Protection): đây là ứng dụng phổ biến nhất của thủy vân số. Một thông tin nào đó mang ý nghĩa quyền sở hữu tác giả(gọi là thủy vân) được nhúng vào trong các sản phẩm số. Yêu cầu kĩ thuật là việc nhúng thủy vân không ảnh hưởng đáng kể đến cảm nhận sản phẩm và phải bền vững trước các tấn công,tồn tại lâu dài cùng sản phẩm.
  + -Thủy vân dễ vỡ: Xác thực thông tin (authentication): một thông tin được giấu trong đối tượng số để nhận biết xem đối tượng đó có bị thay đổi hay không. Yêu cầu kĩ thuật là việc nhúng thủy vân không ảnh hưởng đáng kể đến cảm nhận đối tượng và rất dễ bị phá hủy trước các tấn công. Đây là ứng dụng của thủy vân dễ vỡ.
* -có thêm hiểu biết về ảnh bitmap và cá dạng của ảnh bitmap:
  + - **Ảnh đen trắng** : là ảnh mà mỗi điểm ảnh chỉ thể hiện một trong hai trạng thái 0 và 1 để biểu diễn trạng thái điểm ảnh đen hay trắng.
    - **Ảnh đa mức xám** : là ảnh mà mỗi điểm ảnh được biểu diễn bởi một giá trị và đó là cường độ sáng của điểm ảnh.
    - **Ảnh màu** : là ảnh mà mỗi điểm ảnh được biểu diễn bởi ba đại lượng R, G, B .
    - Hiểu được ý nghĩa của các phần trong tệp ảnh bitmap
    - biết thêm kiến thức về kĩ thuật giấu tin trên sai phân
    - nắm được thông tin về cấu tạo sơ cấp nhất của hình ảnh - pixel

- biết được bản chất của các mã màu cấu tạo nên các pixel ảnh

* + -Hiểu rõ vai trò của hệ mật mã và các ưu điểm của hệ mật mã sử dụng.
  + -Qua đó rút ra chiến lược cũng như phương pháp bảo đảm bảo an toàn thông tin. Hiểu được an toàn thông tin bằng mật mã có vai trò pháp quan trọng và được ứng dụng rộng khắp không chỉ trong ngành công nghệ thông tin mà còn dùng để bảo mật những thông tin và tài liệu quan trọng ngoài đời.
  + -Kỹ năng quản lý.
  + -Kỹ năng thu thập dữ liệu đọc tài liệu bằng tiếng anh.
  + -Kỹ năng phân tích và giải quyết bài toán.
  + -Kỹ năng lập trình, thuật toán, …

## **3.2 Hướng phát triển**

* + Tìm hiểu sâu hơn về hệ mã AES và các ứng dụng của AES
  + Xây dựng thêm nhiều tính năng hơn cho sản phẩm như:
  + Đọc văn bản từ các tệp word hay txt
  + Xuất thông điệp được giải mã sang các file txt và doc
  + Xử lí được nhiều định dạng ảnh hơn
  + Giao diện người dùng tốt hơn
  + Tối ưu các đoạn code

## **Tài liệu tham khảo**

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

[Steganography: How to Hide Information inside Pictures (programiz.com)](https://www.programiz.com/blog/steganography-hiding-information-inside-pictures/)

[Tkinter – Quản lý Layout – Phở Code (phocode.com)](https://phocode.com/python/tkinter/tkinter-quan-ly-layout/)

[(PDF) Hiding Information in Images. (researchgate.net)](https://www.researchgate.net/publication/221129299_Hiding_Information_in_Images)

[raytano · GitHub](https://github.com/raytano)

[Thuật toán khóa đối xứng – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n_kh%C3%B3a_%C4%91%E1%BB%91i_x%E1%BB%A9ng)