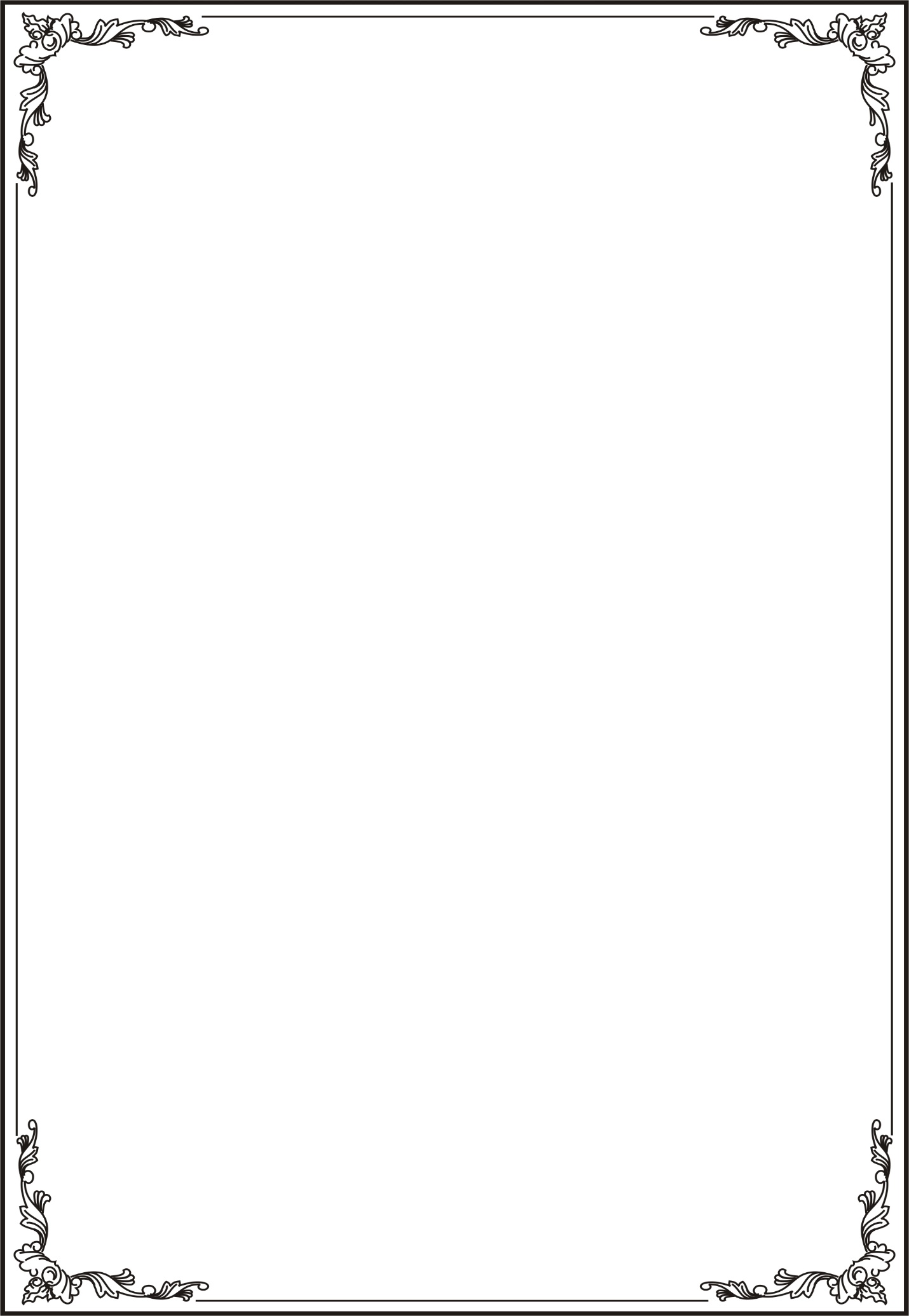
**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC KINH TẾ-KỸ THUẬT CÔNG NGHIỆP**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**BÀI TẬP LỚN**

**MÔN: AN TOÀN THÔNG TIN**

**Đề tài: Hệ mã hóa Vigenere**

**Giảng viên hướng dẫn: Nguyễn Thu Hiền**

**Nhóm SV thực hiện: -** Đỗ Thị Thùy Nhương

* Nguyễn Việt Trinh
* Nguyễn Ngọc Tân

**Lớp: DHTI13A5HN**

***Hà Nội, ngày 13 tháng 11 năm 2022***

# **Lời mở đầu**

Trao đổi thông tin luôn là nhu cầu cần thiết của con người, đặc biệt là trong cuộc sống hiện đại ngày nay khi mà mạng máy tính và Internet phát triển một cách mạnh mẽ và giữ vai trò quan trọng trong mọi lĩnh vực đời sống xã hội như: chính trị, quân sự, học tập, mua sắm, kinh doanh, … Tất cả những thông tin liên quan đến những công việc đều được máy tính quản lý và truyền đi trên hệ thống mạng. Đối với những thông tin bình thường thì không ai chú ý đến, nhưng đối với những thông tin mang tính chất sống còn đối với một cá nhân hay một tổ chức thì vẫn đề bảo mật thông tin là rất quan trọng và được đặt lên hàng đầu. Chính vì vậy, nên có rất nhiều tổ chức cá nhân đã nghiên cứu, tìm kiếm và đưa ra rất nhiều giải pháp bảo mật thông tin. Trong đó hệ mã hóa Vigenere rất thích hợp trong truyền thông tin dữ liệu và có tính bảo mật khá cao.

Bài báo cáo này được biên soạn dựa trên những kiến thức mà cô giáo đã chỉ dạy và thông qua sự tìm hiểu, nghiên cứu tích cực. Bài báo cáo này sẽ tập trung trình bày về hệ mã hóa Vigenere. Bài báo cáo còn dựa trên kinh nghiệm nhỏ bé, nếu có một vài thiếu sót mong thầy cô đóng góp ý kiến.

# **Mục lục**

**Phân công công việc:**

1. **Tìm hiểu khái niệm về hệ mã hóa Vigenère 4**
2. **Giới thiệu về hệ mã hóa Vigenère 4**
3. **Hệ mã hóa Vigenère 5**
4. **Nguyên lý thực hiện 6**
5. **Thuật toán 7**
6. **Tài liệu tham khảo 8**

# **Công tác chuẩn bị:**

|  |  |
| --- | --- |
| Thành viên nhóm | Nhiệm vụ |
| 1. Đỗ Thị Thùy Nhương | * **Tổng hợp làm file báo cáo** * **Thiết kế giao diện chương trình** * **Code thuật toán** |
| 1. Nguyễn Việt Trinh | * **Tìm hiểu lý thuyết về hệ mã hóa Vingenere** * **Code thuật toán** |
| 1. Nguyễn Ngọc Tân | * **Tìm hiểu lý thuyết về hệ mã hóa Vingenere** * **Code thuật toán** |

# **Đề tài: Tìm hiểu hệ mã hóa Vigenère?**

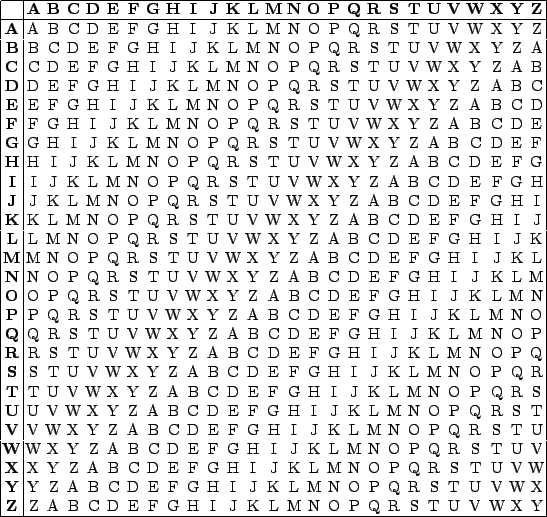
## **Tìm hiểu khái niệm về hệ mã hóa Vigenere:**

### **Giới thiệu về hệ mã hóa Vigenere:**

* Mật mã Vigenère được đặt tên theo Blaise de Vigenère (hình), mặc dù [Giovan Battista Bellaso](https://vi.wikipedia.org/w/index.php?title=Giovan_Battista_Bellaso&action=edit&redlink=1) đã phát minh mật mã trước đó. Vigenère đã phát minh một cơ chế tạo khóa tự động mạnh hơn.
* **Mật mã Vigenère đã kéo dài hàng trăm năm mà không thể phá vỡ với khóa đủ dài. Nhưng nếu sử dụng các khóa ngắn hoặc nếu các nhà thám mã có đủ nhiều bản mã so với độ dài khóa thì mật mã Vigenère lại bị thám mã là khá dễ dàng.**
* Mật mã Vigenère là một phương pháp mã hóa chữ văn bản tiếng Anh, lần đầu tiên được mô tả bởi Giovan Battista Bellaso vào năm 1553. Phương pháp mã hóa mật mã Vigenère dễ hiểu và dễ thực hiện, nhưng chỉ đến năm 1863 với nhiều nỗ lực suốt ba thế kỷ, Friedrich Kasiski mới xuất bản một phương pháp chung để giải mã mật mã Vigenère.
* Mật mã Vigenère là tập hợp các quy tắc thay thế chữ cái đơn trong bảng chữ cái tiếng Anh qua việc sử dụng 26 mật mã Caesar với các bước dịch chuyển từ 0 đến 25 tương ứng từ chữ ‘a’ đến chữ ‘z’. Cụ thể, bản mã Vigenère được lập theo công thức sau:
* *ci = (pi + ki) mod 26, i=1, …, l*
* trong đó, *C = {c1…cl}* là bản mã, *P= {p1…pl}* là bản gốc, *K = {k1…kl}* là dãy khóa và l là độ dài bản rõ. Tương tự, bản gốc *P* có thể được tính nếu biết khóa và bản mã theo công thức:
* *pj = (cj – kj) mod 26, j = 1, …, l*
* Mật mã Vigenère không thể phá vỡ trong trường hợp sử dụng các khóa đủ dài. Nhưng với các khóa ngắn hoặc nếu nhà thám mã có đủ nhiều bản mã so với độ dài khóa thì khá dễ để phá vỡ. Việc thám mật mã Vigenère thường tiến hành theo hai bước là: xác định độ dài chu kỳ của khóa trước, sau đó tìm khóa cụ thể.

### **Hệ mã hóa Vigenere:**

* **Mật mã Vigenère** là một phương pháp mã hóa văn bản bằng cách sử dụng xen kẽ một số phép [mã hóa Caesar](https://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_Caesar) khác nhau dựa trên các [chữ cái](https://vi.wikipedia.org/wiki/Ch%E1%BB%AF_c%C3%A1i) của một từ khóa. Nó là một dạng đơn giản của mật mã thay thế dùng nhiều bảng chữ cái.
* Trong phép mã hóa Caesar, mỗi ký tự của bảng chữ cái được dịch đi một khoảng nhất định, ví dụ với bước dịch là 3, A trở thành D, B trở thành E... Mật mã Vigenère là sự kết hợp xen kẽ vài phép mã hóa Caesar với các bước dịch khác nhau.

 Hình 1

* Để mã hóa, ta dùng một hình vuông Vigenère (hình 1). Nó gồm 26 hàng, mỗi hàng dịch về bên trái một bước so với hàng phía trên, tạo thành 26 bảng mã Caesar. Trong quá trình mã hóa, tùy theo từ khóa mà mỗi thời điểm ta dùng một dòng khác nhau để mã hóa văn bản.
* Ví dụ, ta có văn bản cần mã hóa như sau:

ATTACKATDAWN

* Người gửi lựa chọn một từ khóa và viết nó lặp lại nhiều lần trên một dòng đến khi số chữ cái trên dòng bằng số chữ cái trong thông điệp, với từ khóa "LEMON" thì:

LEMONLEMONLE

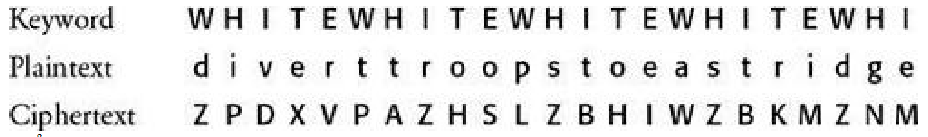
* Chữ cái đầu tiên của văn bản, A, được mã hóa bằng bảng bắt đầu với L (chữ cái đầu tiên của từ khóa). Nó sẽ được mã hóa thành chữ cái trên dòng L và cột A của hình vuông Vigenère, đó là chữ L. Tương tự như vậy, chữ cái thứ hai của văn bản sẽ được mã hóa bằng chữ cái thứ hai của từ khóa: chữ trên dòng E và cột T là X. Sau đây là bản mã:

|  |  |
| --- | --- |
| Văn bản: | ATTACKATDAWN |
| Từ khóa: | LEMONLEMONLE |
| Bản mã: | LXFOPVEFRNHR |

## **Nguyên lý thực hiện:**

* Đầu tiên cần lưu ý là chu kỳ của khóa tìm được có thể không đúng với thực tế được sử dụng. Nếu bản mã đủ dài thì có thể là chính xác, các phương pháp được cung cấp ở đây là gần đúng.
* Mật mã Vigenère áp dụng các mật mã Caesar khác nhau cho các chữ cái liên tiếp.
* Mật mã Caesar là một dạng của mật mã thay thế, theo đó mỗi ký tự trong bản rõ được thay thế bằng một ký tự cách nó một đoạn trong bảng chữ cái để tạo thành bản mã. Giả sử với khóa là 3 (dịch 3 vị trí trong bảng chữ cái), thì chữ ‘a’ sẽ được thay bằng chữ ‘d’, chữ ‘b’ sẽ được thay bằng ‘e’ và cứ thế đến hết bản rõ. Phương pháp này được đặt tên là Caesar, vị Hoàng đế đã sử dụng loại mật mã này thường xuyên trong công việc.
* Nếu mật mã Vigenère sử dụng khóa có chu kỳ 3 là 'PUB', thì chữ cái rõ đầu tiên được mã hóa bằng mật mã Caesar với khóa là 16 (P là chữ cái thứ 16 của bảng chữ cái), chữ cái thứ hai được mã với khóa là 21 (chữ cái U) và chữ cái thứ ba được mã với khóa là 2 (chữ cái B). Chữ cái rõ thứ 4 được mã hóa quay lại bằng chữ khóa thứ nhất (khóa 16). Kết quả là, các chữ cái ở các vị trí 1, 4, 7, 10, ... đều được mã hóa bằng cùng một mật mã Caesar với chữ khóa là P. Các chữ cái ở các vị trí 2, 5, 8, 11, ... và 3, 6, 9, 12, ... được mã hóa bằng mật mã Caesar với khóa tương ứng là chữ U và B.
* Như vậy, trình tự chính xác sẽ phụ thuộc vào chu kỳ của khóa mật mã, tức là độ dài khóa, như với ví dụ trên thì độ dài chu kỳ khóa là 3.
* Ví dụ:

Giả sử ta cần mã hóa câu: **divert troops to east ridge** (chuyển quân sang bờ phía đông). Đầu tiên chọn ra một cụm từ làm khóa. Ví dụ **WHITE.**Ta sẽ viết cụm WHITE này lặp lại đến khi bằng độ dài bản rõ. Rồi lập sơ đồ sau:

****

* Theo sơ đồ, kí tự đầu tiên của bản rõ, chữ d được mã hóa theo mã Caesar với key là a -> W, hay nói cách khác k = 22, vậy d mã hóa thành Z. Tiếp tục, kí tự thứ 2, chữ i mã hóa theo mã Caesar với key là a -> H, hay k = 7, vậy i -> P. Tương tự đến hết. Bản rõ **divert troops to east ridge**sẽ được mã hóa thành **ZPDXVP AZHSLZ BH IWZB KMZNM.**

## **Thuật toán:**

* Một số dòng code nổi bật để giải mã và mã hóa thuật toán mã hóa Vigenere:
* Mã hóa:

public static void VigenereEncrypt(ref StringBuilder s, string key)

{

for (int i = 0; i < s.Length; i++)

s[i] = Char.ToUpper(s[i]);

key = key.ToUpper();

int j = 0;

for (int i = 0; i < s.Length; i++)

{

if (Char.IsLetter(s[i]))

{

s[i] = (char)(s[i] + key[j] - 'A');

if (s[i] > 'Z') s[i] = (char)(s[i] - 'Z' + 'A' - 1);

}

j = j + 1 == key.Length ? 0: j + 1;

}

}

* Giải mã

public static void VigenereDecrypt(ref StringBuilder s, string key)

{

for (int i = 0; i < s.Length; i++)

s[i] = Char.ToUpper(s[i]);

key = key.ToUpper();

int j = 0;

for (int i = 0; i < s.Length; i++)

{

if (Char.IsLetter(s[i]))

{

s[i] = s[i] >= key[j]?

(char) (s[i] - key[j] + 'A'):

(char) ('A' + ('Z' - key[j] + s[i] - 'A') + 1);

}

j = j + 1 == key.Length ? 0: j + 1;

}

}

## **Tài liệu tham khảo:**

* Bài giảng An Toàn Thông Tin – Khoa CNTT – trường ĐH KTKTCN.
* [*https://drx.home.blog/2018/07/22/series-mat-ma-06-ma-vigenere/*](https://drx.home.blog/2018/07/22/series-mat-ma-06-ma-vigenere/)
* [*https://tailieu.vn/doc/bai-giang-mon-an-toan-va-bao-mat-thong-tin-doanh-nghiep-nguyen-thi-hoi-1681334.html*](https://tailieu.vn/doc/bai-giang-mon-an-toan-va-bao-mat-thong-tin-doanh-nghiep-nguyen-thi-hoi-1681334.html)
* [*http://antoanthongtin.vn/gp-mat-ma/tham-mat-ma-vigenere-106294*](http://antoanthongtin.vn/gp-mat-ma/tham-mat-ma-vigenere-106294)

*~ The end ~*