

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO SINH HOẠT HỌC THUẬT
ĐỀ TÀI: Thuật toán Hill

Giảng viên thực hiện : ThS. Dương Chí Thiện

HÀ NỘI – 2024

MỞ ĐẦU

Hiện nay, cuộc cách mạng 4.0 đang phát triển mạnh mẽ như vũ bão, môi trường mạng đã trở nên rất phổ biến trên toàn cầu. Thông qua môi trường mạng, mọi người có thể kết nối và trao đổi thông tin với nhau một cách nhanh chóng và thuận tiện, kéo theo đó là những hệ lụy phát sinh trong quá trình trao đổi thông tin với nhau. Với trình độ phát triển của khoa học kỹ thuật, những kẻ xâm nhập luôn tìm cách để đánh cắp thông tin nhằm mục đích thương mại hay với những mục đích khác nhau, đã tấn công với quy mô lớn, cũng như tần suất nhiều hơn gây thiệt hại cho các tổ chức, cá nhân một cách nghiêm trọng. Điều này đã khiến những nhà sản xuất, kinh doanh và quản lý trên môi trường mạng phải đối diện với những khó khăn lớn là làm thế nào để bảo vệ dữ liệu thông tin quan trọng không bị tấn công, truy xuất dữ liệu một cách bất hợp pháp, đảm bảo thông tin không bị sai lệch hoặc bị lộ do xâm nhập của kẻ thứ ba (không được phép).

Trước tình hình đó, một số giải thuật mã hóa cũng như các công cụ mã hóa đã được xây dựng nhằm đảm bảo tính an toàn của dữ liệu tại nơi lưu trữ cũng như khi dữ liệu được truyền qua mạng. Việc tìm hiểu và phát triển thuật toán Hill cũng nhằm vào mục đích này.

MỤC LỤC

MỞ ĐẦU.....	2
CHƯƠNG 1: TỔNG QUAN AN TOÀN VÀ BẢO MẬT THUẬT TOÁN.....	4
1. Tổng quan an toàn bảo mật hệ thống thông tin	4
1.1. Khái quát về an toàn thông tin.....	4
1.1.1. An toàn thông tin là gì?.....	4
1.1.2. Các thành phần của an toàn thông tin.....	4
1.2. Khái quát về an toàn hệ thống thông tin.....	4
1.2.1. Các thành phần của hệ thống thông tin.....	4
1.2.2. An toàn hệ thống thông tin là gì?	5
1.3. Các yêu cầu đảm bảo an toàn hệ thống thông tin	5
1.3.1. Bí mật.....	5
1.3.2. Toàn vẹn.....	5
1.3.3. Sẵn dùng.....	6
1.4. Các mối đe dọa đối với một hệ thống và các biện pháp ngăn chặn	6
2. Cơ sở toán học	7
3. Chuyên đề	8
CHƯƠNG 2: HỆ MẬT MÃ HILL.....	9
1. Lý do, mục đích chọn hệ mật mã Hill	9
2. Tổng quan về hệ mật mã Hill.....	9
3. Cài đặt công cụ sử dụng thuật toán Hill	15
3.1. Một số đặc trưng của ngôn ngữ lập trình Python	15
3.2. Cài đặt công cụ	16
4. Hệ mật mã Hill	17
CHƯƠNG 3: KẾT LUẬN VÀ ĐỊNH HƯỚNG	22
1. Ưu và nhược điểm của hệ mật mã Hill	22
2. Kết luận về hệ mật mã Hill	23
3. Định hướng phát triển hệ mật mã Hill.....	23
TÀI LIỆU THAM KHẢO.....	24

CHƯƠNG 1: TỔNG QUAN AN TOÀN VÀ BẢO MẬT THUẬT TOÁN

1. Tổng quan an toàn bảo mật hệ thống thông tin

1.1. Khái quát về an toàn thông tin

1.1.1. An toàn thông tin là gì?

An toàn thông tin (Information security) là việc bảo vệ chống truy cập, sử dụng, tiết lộ, sửa đổi, hoặc phá hủy thông tin một cách trái phép.

An toàn thông tin là việc bảo vệ các thuộc tính bí mật (confidentiality), tính toàn vẹn (integrity) và tính sẵn dùng (availability) của các tài sản thông tin trong quá trình chúng được lưu trữ, xử lý, hoặc truyền tải.

1.1.2. Các thành phần của an toàn thông tin

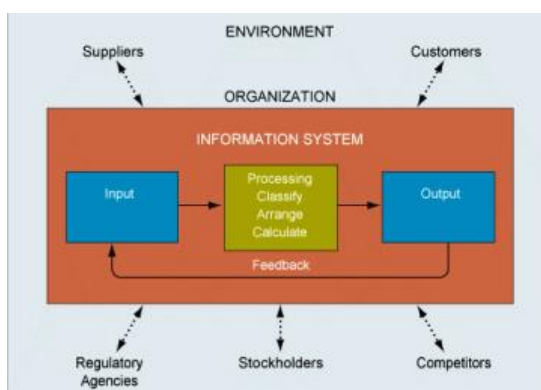
An toàn thông tin có thể được chia thành ba thành phần chính:

- + An toàn máy tính và dữ liệu (Computer & data security)
- + An ninh mạng (Network security)
- + Quản lý an toàn thông tin (Management of information security).

Ba thành phần của an toàn thông tin có quan hệ mật thiết và giao thoa với nhau, trong đó phần chung của cả ba thành phần trên là chính sách an toàn thông tin (Policy).

1.2. Khái quát về an toàn hệ thống thông tin

1.2.1. Các thành phần của hệ thống thông tin



Mô hình hệ thống thông tin cơ quan, tổ chức

Hệ thống thông tin là hệ thống tích hợp các thành phần nhằm phục vụ việc thu thập, lưu trữ, xử lý thông tin, và chuyển giao thông tin, tri thức và các sản phẩm số.

Trong nền kinh tế số, hệ thống thông tin đóng vai trò rất quan trọng trong hoạt động của các tổ chức, cơ quan và doanh nghiệp (gọi chung là tổ chức). Có thể nói, hầu hết các tổ chức đều sử dụng các hệ thống thông tin với các quy mô khác nhau để quản lý các hoạt động của mình.

Hệ thống thông tin dựa trên máy tính (Computerbased information system), hay sử dụng công nghệ máy tính để thực thi các nhiệm vụ là lớp hệ thống thông tin được sử dụng rộng rãi nhất. Hệ thống thông tin dựa trên máy tính thường gồm các thành phần: phần cứng (Hardware) để thu thập, lưu trữ, xử lý và biểu diễn dữ liệu; phần mềm (Software) chạy trên phần cứng để xử lý dữ liệu; cơ sở dữ liệu (Databases) để lưu trữ dữ liệu; mạng (Networks) là hệ thống truyền dẫn thông tin/dữ liệu; và các thủ tục (Procedures) là tập hợp các lệnh kết hợp các bộ phận nêu trên để xử lý dữ liệu, đưa ra kết quả mong muốn.

1.2.2. An toàn hệ thống thông tin là gì?

An toàn hệ thống thông tin là việc đảm bảo các thuộc tính an ninh, an toàn của hệ thống thông tin, bao gồm tính *bí mật* (confidentiality), *tính toàn vẹn* (Integrity) và tính *sẵn dùng* (Availability).

1.3. Các yêu cầu đảm bảo an toàn hệ thống thông tin

1.3.1. Bí mật

Tính bí mật đảm bảo rằng chỉ người dùng có thẩm quyền mới được truy nhập thông tin, hệ thống. Các thông tin bí mật có thể bao gồm:

- + Dữ liệu riêng của cá nhân
- + Các thông tin thuộc quyền sở hữu trí tuệ của các doanh nghiệp hay các cơ quan, tổ chức
- + Các thông tin có liên quan đến an ninh của các quốc gia và các chính phủ.

Thông tin bí mật lưu trữ hoặc trong quá trình truyền tải cần được bảo vệ bằng các biện pháp phù hợp, tránh bị lộ lọt hoặc bị đánh cắp. Các biện pháp có thể sử dụng để đảm bảo tính bí mật của thông tin như bảo vệ vật lý, hoặc sử dụng mật mã (cryptography).

1.3.2. Toàn vẹn

Tính toàn vẹn đảm bảo rằng thông tin và dữ liệu chỉ có thể được sửa đổi bởi những người dùng có thẩm quyền. Tính toàn vẹn liên quan đến tính hợp lệ (validity) và chính xác (accuracy) của dữ liệu.

Trong nhiều tổ chức, thông tin và dữ liệu có giá trị rất lớn, như bản quyền phần mềm, bản quyền âm nhạc, bản quyền phát minh, sáng chế. Mọi thay đổi không có thẩm quyền có thể ảnh hưởng rất nhiều đến giá trị của thông tin. Thông tin hoặc dữ liệu là toàn vẹn nếu nó thỏa mãn ba điều kiện:

- + Không bị thay đổi
- + Hợp lệ
- + Chính xác

1.3.3. Sẵn dùng

Tính sẵn dùng, hoặc khả dụng đảm bảo rằng thông tin, hoặc hệ thống có thể truy nhập bởi người dùng hợp pháp bất cứ khi nào họ có yêu cầu. Tính sẵn dùng có thể được đo bằng các yếu tố:

- Thời gian cung cấp dịch vụ (Uptime);
- Thời gian ngừng cung cấp dịch vụ (Downtime);
- Tỷ lệ phục vụ: $A = (\text{Uptime}) / (\text{Uptime} + \text{Downtime})$;
- Thời gian trung bình giữa các sự cố;
- Thời gian trung bình ngừng để sửa chữa;
- Thời gian khôi phục sau sự cố.

1.4. Các mối đe dọa đối với một hệ thống và các biện pháp ngăn chặn

Có 3 hình thức chủ yếu đe dọa đối với hệ thống:

- 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 đe dọa đối với một hệ thống thông tin có thể đến từ nhiều nguồn và được thực hiện bởi các đối tượng khác nhau. Chúng ta có thể chia thành 3 loại đối tượng như sau:

+ Các đối tượng từ ngay trong hệ thống(insider): đây là những người có quyền truy cập hợp pháp đối với hệ thống.

+ Những đối tượng ngoài hệ thống (hacker, cracker): Thường các đối tượng này tấn công qua những đường kết nối với hệ thống như internet

+ Các phần mềm (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.

- 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 quy định của các tổ chức nhằm đảm bảo tính an toàn bảo mật của hệ thống.

2. Cơ sở toán học

Hệ mật mã Hill dựa trên cơ sở toán học của đại số tuyến tính, đặc biệt là trong lĩnh vực ma trận. Trong hệ mật mã Hill, thông điệp được biểu diễn dưới dạng các vector cột và được mã hóa bằng ma trận khóa. Quá trình mã hóa và giải mã trong hệ mật mã Hill đều sử dụng phép nhân ma trận và ma trận nghịch đảo.

Để bắt đầu quá trình mã hóa, thông điệp cần được chia thành các khối có kích thước phù hợp với ma trận khóa. Mỗi khối sẽ được biểu diễn dưới dạng vector cột và nhân với ma trận khóa để tạo ra thông điệp đã được mã hóa. Để giải mã, ta cần tính toán ma trận nghịch đảo của ma trận khóa và nhân với thông điệp đã mã hóa để khôi phục lại thông điệp gốc.

Trong quá trình thực hiện phép nhân ma trận, ta cần chú ý đến việc thực hiện phép toán trong không gian modulo một số nguyên dương nào đó, thường là kích thước của bảng chữ cái. Điều này giúp bảo vệ thông điệp trước các cuộc tấn công mật mã thông thường và tăng cường tính bảo mật của hệ thống mật mã Hill.

Nhờ vào cơ sở toán học vững chắc trong đại số tuyến tính và ma trận, hệ mật mã Hill được áp dụng trong nhiều lĩnh vực như an ninh mạng, truyền thông và bảo mật thông tin. Điều này làm cho hệ mật mã Hill trở thành một công cụ quan trọng trong việc bảo vệ thông tin và dữ liệu quan trọng.

3. Chuyên đề

Thuật toán Hill là một phương pháp mã hóa đối xử với thông tin bí mật thông qua việc sử dụng ma trận khóa để thực hiện quá trình mã hóa và giải mã. Tuy nhiên, như mọi hệ thống mật mã khác, thuật toán Hill cũng đối diện với một số vấn đề liên quan đến an toàn và bảo mật. Dưới đây là một số vấn đề chung mà thuật toán Hill cần xem xét:

- *Kích thước ma trận:* Việc chọn kích thước ma trận trong thuật toán Hill có thể ảnh hưởng đến tính bảo mật của hệ thống. Kích thước ma trận quá nhỏ có thể dễ dàng bị tấn công bằng các phương pháp phân tích tần số hoặc tấn công độ phức tạp tuyến tính. Ngược lại, kích thước ma trận quá lớn có thể làm tăng độ phức tạp tính toán và tạo ra khó khăn trong quá trình thực hiện mã hóa và giải mã.
- *Khóa mã hóa:* Việc quản lý và bảo vệ khóa mã hóa trong thuật toán Hill cũng là một vấn đề quan trọng. Nếu khóa mã hóa bị lộ hoặc không đủ mạnh, thông tin mã hóa có thể bị dễ dàng giải mã bởi kẻ tấn công.
- *Tấn công tương quan:* Một vấn đề khác là tấn công tương quan, trong đó kẻ tấn công có thể sử dụng các quan hệ tương quan giữa các phần tử trong ma trận để phá vỡ hệ thống mã hóa.
- *Khả năng chống lại các tấn công:* Thuật toán Hill cần có khả năng chống lại các loại tấn công phổ biến như tấn công brute-force, tấn công tương quan, tấn công từ chối dịch vụ (DoS),...
- *Hiệu suất và tính linh hoạt:* Để được ứng dụng rộng rãi, thuật toán Hill cần đảm bảo hiệu suất cao và tính linh hoạt trong việc tích hợp vào các hệ thống và ứng dụng khác nhau mà không ảnh hưởng đến trải nghiệm người dùng.

Việc xem xét và giải quyết những vấn đề an toàn và bảo mật là quan trọng khi triển khai thuật toán Hill.

CHƯƠNG 2: HỆ MẬT MÃ HILL

1. Lý do, mục đích chọn hệ mật mã Hill

Lý do chọn hệ mật mã Hill:

- *Đa dạng hóa bảo mật:* Hill Cipher sử dụng phép toán ma trận trong quá trình mã hóa, tạo ra một cách tiếp cận độc đáo và hiệu quả trong việc bảo vệ thông tin.
- *Khả năng linh hoạt:* Với khả năng thay đổi kích thước ma trận khóa, Hill Cipher cung cấp sự linh hoạt cho việc điều chỉnh mức độ bảo mật theo nhu cầu cụ thể.
- *Thách thức cho kẻ tấn công:* Việc sử dụng ma trận trong quá trình mã hóa tạo ra một thách thức đối với kẻ tấn công, đồng thời giúp tăng cường tính an toàn của hệ thống.

Mục đích chọn hệ mật mã Hill

- *Bảo vệ thông tin quan trọng:* Hill Cipher được chọn để bảo vệ thông tin quan trọng và nhạy cảm, đảm bảo rằng dữ liệu không bị lộ ra ngoài một cách dễ dàng.
- *Sử dụng trong các ứng dụng cần mức độ bảo mật trung bình:* Hill Cipher thích hợp cho các ứng dụng cần mức độ bảo mật trung bình, nơi mà việc mã hóa thông tin là cần thiết nhưng không cần phức tạp quá nhiều.
- *Nâng cao hiểu biết về mật mã:* Việc chọn Hill Cipher cũng có thể nhằm mục đích nâng cao hiểu biết và kỹ năng trong lĩnh vực mật mã của người sử dụng.

Tuy nhiên, cũng cần lưu ý rằng như bất kỳ hệ mật mã nào khác, hệ mật mã Hill cũng có nhược điểm và yếu tố cần cân nhắc, như khả năng dễ dàng bị tấn công bằng cách phân tích tuyến tính nếu không áp dụng các biện pháp phòng ngừa phù hợp.

2. Tổng quan về hệ mật mã Hill

Hệ mật mã Hill được phát minh vào năm 1918 bởi Lester S. Hill

Hệ mật mã này được thực hiện trên từng bộ m ký tự liên tiếp, điều khác là mỗi ký tự của bản mã được xác định bởi một tổ hợp tuyến tính (trên vành Z_{26}) của

m ký tự trong bản rõ. Như vậy, khóa sẽ được cho bởi một ma trận cấp m, tức là một phần tử của $K \in Z^{m \times n}$.

Để phép biến đổi tuyến tính xác định bởi ma trận K có phép nghịch đảo, bản thân ma trận K cũng phải có ma trận nghịch đảo K^{-1} theo mod 26; mà điều kiện cần và đủ để K có nghịch đảo là định thức của nó, ký hiệu $\det K$, nguyên tố 26.

Mã Hill được mô tả như sau:

Tập khóa K	Hàm mã hóa e	Hàm giải mã d
$K = \{K \in (Z_{26})^{m \times m} : \det(K, 26) = 1\}$	Bản mã $y \in (Z_{26})^m$	Bản rõ $x \in (Z_{26})^m$
Cho m là số nguyên dương $P = C = (Z_{26})^m$	$y = (y_1, y_2, y_3, \dots, y_m) =$ $= e_K(x_1, x_2, x_3, \dots, x_m) =$ $= (x_1, x_2, x_3, \dots, x_m) \cdot K \text{ mod } 26$	$x = (x_1, x_2, x_3, \dots, x_m) =$ $= d_K(y_1, y_2, y_3, \dots, y_m) =$ $= (y_1, y_2, y_3, \dots, y_m) \cdot K^{-1} \text{ mod } 26$

Ý tưởng ở đây là lấy m tổ hợp tuyến tính của m ký tự trong một phần tử của bản rõ để tạo ra m ký tự ở một phần tử của bản mã.

Có thể lấy một ma trận K kích thước $m \times m$ làm khóa. Nếu một phần tử ở hàng i và cột j của K là $k_{i,j}$ thì có thể viết

$K = (k_{i,j})$, với $x = (x_1, x_2, \dots, x_m) \in P$ và $K \in K$, ta tính $y = (y_1, y_2, \dots, y_m)$ như sau:

$$(y_1, \dots, y_m) = (x_1, \dots, x_m) \begin{pmatrix} k_{1,1} & k_{1,2} & \dots & k_{1,m} \\ k_{2,1} & k_{2,2} & \dots & k_{2,m} \\ \dots & \dots & \dots & \dots \\ k_{m,1} & k_{m,2} & \dots & k_{m,m} \end{pmatrix}$$

Nói một cách khác $y = xK$.

Chúng ta nói rằng bản mã nhận được từ bản rõ nhờ phép biến đổi tuyến tính. Ta sẽ xét xem phải thực hiện giải mã như thế nào, tức là làm thế nào để tính x từ y. Đã làm quen với đại số tuyến tính sẽ thấy rằng phải dùng ma trận nghịch đảo K^{-1} để giải mã. Bản mã được giải mã bằng công thức yK^{-1} .

Sau đây là một số định nghĩa về những khái niệm cần thiết lấy từ đại số tuyến tính. Nếu $A = (a_{i,j})$ là một ma trận cấp $l \times m$ và $B = (b_{i,k})$ là một ma trận cấp $m \times n$ thì tích ma trận $AB = (c_{i,k})$ được định nghĩa theo công thức:

$$c_{1,k} = \sum_{j=1}^m a_{i,j} b_{j,k}$$

Với $1 \leq i \leq l$ và $1 \leq k \leq l$. Tức là các phần tử ở hàng i và cột thứ k của AB được tạo ra bằng cách lấy hàng thứ i của A và cột thứ k của B , sau đó nhân tương ứng các phần tử với nhau và cộng lại. Cần để ý rằng AB là một ma trận cấp $l \times n$.

Theo định nghĩa này, phép nhân ma trận là kết hợp (tức $(AB)C = A(BC)$) nhưng không giao hoán (không phải lúc nào $AB = BA$, thậm chí đối với ma trận vuông A và B).

Ma trận đơn vị $m \times m$ (ký hiệu là I_m) là ma trận cấp $m \times m$ có các số 1 nằm ở đường chéo chính và các số 0 ở vị trí còn lại. Ma trận đơn vị cấp 2 là:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

I_m được gọi là ma trận đơn vị vì $AI_m = A$ với mọi ma trận cấp $l \times m$ và $I_m B = B$ với mọi ma trận cấp $m \times n$. Ma trận nghịch đảo của ma trận A cấp $m \times m$ (nếu tồn tại) là ma trận A^{-1} sao cho $A^{-1}A = A^{-1}A = I_m$. Không phải mọi ma trận đều có nghịch đảo, nhưng nếu tồn tại thì nó duy nhất.

Với các định nghĩa trên, có thể dễ dàng xây dựng công thức giải mã đã nêu: Vì $y = xK$, ta có thể nhân cả 2 vế của đẳng thức với K^{-1} và nhận được:

$$yK^{-1} = (x(K K^{-1})) = xI_m = x$$

Tính khả nghịch của một ma trận vuông phụ thuộc vào giá trị định thức của nó. Để tránh sự tổng quát hoá không cần thiết, ta chỉ giới hạn trong trường hợp 2×2 .

Định nghĩa

Định thức của ma trận $A = (a_{i,j})$ cấp 2×2 là giá trị

$$\det A = a_{1,1} a_{2,2} - a_{1,2} a_{2,1}$$

Nhận xét: Định thức của một ma trận vuông cấp $m \times m$ có thể được tính theo các phép toán hằng sơ cấp.

Hai tính chất quan trọng của định thức là $\det I_m = 1$ và quy tắc nhân $\det(AB) = \det A \times \det B$.

Một ma trận thức K là có nghịch đảo khi và chỉ khi định thức của nó khác 0. Tuy nhiên, điều quan trọng cần nhớ là ta đang làm việc trên Z_{26} . Kết quả tương ứng là ma trận K có nghịch đảo theo modulo 26 khi và chỉ khi $\text{UCLN}(\det K, 26) = 1$.

Sau đây sẽ chứng minh ngắn gọn kết quả này.

Trước tiên, giả sử rằng $\text{UCLN}(\det K, 26) = 1$. Khi đó $\det K$ có nghịch đảo trong Z_{26} . Với $1 \leq i \leq m$, $1 \leq j \leq m$, định nghĩa K_{ij} ma trận thu được từ K bằng cách loại bỏ hàng thứ i và cột thứ j . Và định nghĩa ma trận K^* có phần tử (i,j) của nó nhận giá trị $(-1)^{i+j} \det K_{ji}$ (K^* được gọi là ma trận bù đại số của K). Khi đó có thể chứng tỏ rằng:

$$K^{-1} = (\det K)^{-1} K^*.$$

Bởi vậy K là khả nghịch.

Ngược lại K có nghịch đảo K^{-1} . Theo quy tắc nhân của định thức:

$$1 = \det I = \det (KK^{-1}) = \det K \det K^{-1}$$

Bởi vậy $\det K$ có nghịch đảo trong Z_{26} .

Nhận xét: Công thức đối với ở trên không phải là một công thức tính toán có hiệu quả trừ các trường hợp m nhỏ (chẳng hạn $m = 2, 3$). Với m lớn, phương pháp thích hợp để tính các ma trận nghịch đảo phải dựa vào các phép toán hằng sơ cấp.

Trong trường hợp 2×2 , ta có công thức sau:

Định lý

Giả sử $A = (a_{ij})$ là một ma trận cấp 2×2 trên Z_{26} sao cho $\det A = a_{1,1}a_{2,2} - a_{1,2}a_{2,1}$ có nghịch đảo. Khi đó:

$$A^{-1} = (\det A)^{-1} \begin{pmatrix} a_{2,2} & -a_{1,2} \\ -a_{2,1} & a_{1,1} \end{pmatrix}$$

Ví dụ minh họa về mã hóa và giải mã của thuật toán Hill

VD1: Cho bản rõ là chữ TUDO, chọn $m = 2$ và $K = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$

$$K^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$$

Mã hóa:

(19,20) tương ứng với TU và (3,14) tương ứng với DO

$$\begin{aligned} (19,20) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} &= (209 + 60, 152 + 140) \\ &= (269, 292) \\ &= (9, 6) \end{aligned}$$

$$= (J, G)$$

$$(3, 14) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (33 + 42, 24 + 98)$$

$$= (75, 122)$$

$$= (23, 18)$$

$$= (X, S)$$

Bản mã số: 9 6 23 18 tương ứng với bản mã chữ: JGXS

Giải mã:

Từ bản mã chữ JGXS có bản mã số: 9 6 23 18

$$(9, 6) \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (63 + 138, 162 + 66)$$

$$= (201, 228)$$

$$= (19, 29)$$

$$(23, 18) \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (161 + 414, 414 + 612)$$

$$= (575, 612)$$

$$= (3, 14)$$

$$= (D, O)$$

Vậy bản rõ số là 19 29 3 14 , bản rõ chữ là TUDO

VD2: Giả sử cần mã hóa bản rõ: JULY. Vẫn sử dụng khóa K như VD1

Mã hóa:

(9, 20) tương ứng với JU và (11, 24) tương ứng với LY

$$(9, 20) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (99 + 60, 72 + 140)$$

$$= (159, 212)$$

$$= (3, 4)$$

$$= (D, E)$$

$$(11, 24) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (121 + 72, 88 + 168)$$

$$= (193, 256)$$

$$= (11, 22)$$

$$= (L, W)$$

Bản mã số: 3 4 11 22 tương ứng với bản mã chữ: DELW

Giải mã:

$$(3, 4) \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (21 + 92, 54 + 44)$$

$$= (113, 98)$$

$$= (9, 20)$$

$$(11, 22) \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = (77 + 506, 198 + 242)$$

$$= (583, 440)$$

$$= (11, 24)$$

Vậy bản rõ số là 9 20 11 24 , bản rõ chữ là JULY

Thám mã

Hệ mã Hill là một hệ mật khó pha hơn nếu tấn công chỉ với bản mã. Tuy nhiên hệ mật này dễ bị phá nếu tấn công bằng bản rõ đã biết. Trước tiên, giả sử rằng, thám mã đã biết được giá trị m đang sử dụng. Giả sử thám mã có ít nhất m cặp véc tơ khác nhau $x_j = (x_{1,j}, x_{2,j}, \dots, x_{m,j})$ và $y_j = (y_{1,j}, y_{2,j}, \dots, y_{m,j})$ ($1 \leq j \leq m$) sao cho $y_j = e_K(x_j)$, $1 \leq j \leq m$. Nếu xác định hai ma trận: $X = (x_{i,j})$ $Y = (y_{i,j})$ cấp $m \times m$ thì ta có phương trình ma trận $Y = XK$, trong đó ma trận K cấp $m \times m$ là khoá chưa biết. Với điều kiện ma trận Y là khả nghịch. Oscar có thể tính $K = X^{-1} Y$ và nhờ vậy phá được hệ mật. (Nếu Y không khả nghịch thì cần phải thử các tập khác gồm m cặp rõ - mã).

Ví dụ:

Giả sử bản rõ *Friday* được mã hoá bằng mã Hill với $m = 2$, bản mã nhận được là PQCFKU.

Ta có $e_K(5,17) = (15,16)$, $e_K(8,3) = (2,5)$ và $e_K(0,24) = (10,20)$. Từ hai cặp rõ - mã đầu tiên, ta nhận được phương trình ma trận:

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} K$$

Dùng định lý dễ dàng tính được:

$$\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}$$

Bởi vậy:

$$k = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$$

Ta có thể dùng cặp rõ - mã thứ 3 để kiểm tra kết quả này.

Vấn đề ở đây là thám mã phải làm gì nếu không biết m ?

Giả sử rằng m không quá lớn, khi đó thám mã có thể thử với $m = 2, 3, \dots$ cho tới khi tìm được khoá. Nếu một giá trị giả định của m không đúng thì ma trận $m \times m$ tìm được theo thuật toán đã mô tả ở trên sẽ không tương thích với các cặp rõ - mã khác. Phương pháp này, có thể xác định giá trị m nếu chưa biết.

3. Cài đặt công cụ sử dụng thuật toán Hill

3.1. Một số đặc trưng của ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1990. Ngôn ngữ này sở hữu những điểm mạnh như dễ đọc, dễ nhớ và dễ học nên vì vậy Python trở thành 1 trong 3 ngôn ngữ mà những ai bắt đầu học lập trình nên học.

Đối với các bạn đang muốn bắt đầu học lập trình, thì đây là ngôn ngữ nên được xếp đầu hàng mục học tập của bạn, vì nó vừa dễ học, dễ hiểu, lại có thể ứng dụng ngay trong công việc và đời sống, cũng như có nhiều cơ hội phát triển trong tương lai.

Python tương thích với nhiều hệ điều hành như: Windows, MacOS, Linux... Chương trình Python có thể di chuyển được qua lại giữa các nền tảng mà không có thay đổi nào.

Thư viện trong ngôn ngữ Python được thiết kế có khả năng lưu trữ số lượng lớn công việc nhờ đó trong quá trình học tập và làm việc bạn được hỗ trợ tối đa. Các thư viện của Python cho phép bạn kiểm tra, đánh giá mà không ảnh hưởng đến các dòng code hay ứng dụng bạn đang thiết kế.

Ở phạm vi cài đặt và triển khai công cụ mã hóa này, chúng ta sẽ sử dụng Python để cài đặt phần lõi của ứng dụng, tức bản chất thuật toán của nó, nhằm đạt được hiệu quả thực thi tính toán cao nhất.

3.2. Cài đặt công cụ

Bước 1: Cài đặt Visual Studio Code

- Truy cập trang chính thức của Visual Studio Code: <https://code.visualstudio.com/>.
- Tải và cài đặt phiên bản mới nhất cho hệ điều hành của bạn.

Bước 2: Cài đặt Extension cho VSCode

- Mở VSCode.
- Trong thanh bên trái, chọn biểu tượng “Extensions” (hoặc nhấn Ctrl+Shift+X).
- Tìm kiếm “Python” trong ô tìm kiếm.
- Chọn extension “Python” được phát triển bởi Microsoft và nhấn nút “Install”.

Bước 3: Cài đặt Numpy trong Python

- Mở terminal hoặc command prompt và chạy lệnh sau để cài đặt NumPy:
pip install numpy

Bước 4: Tạo và Chạy một File Python

- Mở VScode

Chọn “File” > “New File” để tạo một file mới.

- Lưu file với đuôi .py, ví dụ: hello.py.
- Viết mã Python vào file. Ví dụ: pythonCopy codeprint("Hello, World!")
- Chọn “View” > “Terminal” để mở terminal tích hợp tại đáy của VSCode.
- Gõ lệnh sau để chạy script Python: bashCopy codepython hello.py hoặc bashCopy codepython3 hello.py

Bước 5: Debugging

- Đặt breakpoints bằng cách nhấn vào cạnh số dòng code bạn muốn dừng lại.
- Nhấn F5 để bắt đầu chế độ debug. Bạn cũng có thể sử dụng các biểu tượng debug trên thanh công cụ.

Bước 6: Cài đặt Thêm (Optional)

- Python Docstring Generator:

Cài đặt extension “Python Docstring Generator” để tự động tạo docstrings cho hàm và phương thức.

- Pylint hoặc flake8:

Cài đặt extension “Pylint” hoặc “flake8” để kiểm tra mã nguồn Python và cải thiện chất lượng mã.

- Jupyter (Nếu cần):

Cài đặt extension “Jupyter” để hỗ trợ Jupyter Notebooks trong VSCode.

Bước 6: Cập nhật Extension và VSCode

Hãy thường xuyên kiểm tra và cập nhật các extension Python và VSCode để đảm bảo bạn sử dụng phiên bản mới nhất với các tính năng và sửa lỗi.

4. Hệ mật mã Hill

CODE MÃ HÓA:

```
import numpy as np
from docx import Document
# Khai báo biến ký tự tiếng Việt và hàm tạo ma trận khóa
vietnamese_characters =
'aàãâäåăąảãầấẫẩậbcdđeèềếệêễểễệìĩỉifghjklmnoòốồỗôồỗỗộơờ  
ởỡớợpqrstuùũủừữứựvwxyỳỷỹyy0123456789!"#$%&"()*+,-  
./:;<=>?@[\\]^_`{|}~'

def generate_invertible_matrix(n):
    while True:
        key_matrix = np.random.randint(0,
len(vietnamese_characters), size=(n, n))
        if np.linalg.det(key_matrix) % len(vietnamese_characters)
!= 0 and np.linalg.det(key_matrix) % 2 != 0:
            return key_matrix


# Hàm mã hóa một chuỗi thông điệp
def hill_encrypt(message, key_matrix):
    message = message.lower()
    n = key_matrix.shape[0]
    while len(message) % n != 0:
```

```

        message += 'x'
    message_vector = [vietnamese_characters.index(char) for
char in message]
    encrypted_vector = []
    for i in range(0, len(message_vector), n):
        block = message_vector[i:i+n]
        encrypted_block = np.dot(key_matrix, block) %
len(vietnamese_characters)
        encrypted_vector.extend(encrypted_block)
    encrypted_message = ".join(vietnamese_characters[num] for
num in encrypted_vector)
    return encrypted_message

```

Hàm ghi thông điệp đã mã hóa vào tệp tin

```

def save_encrypted_message(encrypted_message, file_path):
    with open(file_path, 'w', encoding='utf-8') as file:
        file.write(encrypted_message)

```

```

def process_text(text):

```

```

    processed_text = text.lower()
    valid_chars = [char for char in processed_text if char in
vietnamese_characters]
    return ".join(valid_chars)

```

Đọc nội dung từ tệp DOCX

```

def read_docx(file_path):
    doc = Document(file_path)
    full_text = []
    for para in doc.paragraphs:
        processed_para = process_text(para.text)
        if processed_para: # Chỉ thêm vào danh sách nếu đoạn
văn bản không rỗng

```



```

        return None

# Hàm giải mã một chuỗi đã được mã hóa
def hill_decrypt(encrypted_message, key_matrix):
    n = key_matrix.shape[0]
    det = int(np.round(np.linalg.det(key_matrix)))
    det_inv = mod_inverse(det % len(vietnamese_characters),
len(vietnamese_characters))

    if det == 0 or det_inv is None:
        raise ValueError("Matrix is not invertible (det = 0 )")
    adj_matrix = np.round(det *
np.linalg.inv(key_matrix)).astype(int) % len(vietnamese_characters)
    key_inverse_matrix = (det_inv * adj_matrix) %
len(vietnamese_characters)

    encrypted_vector = [vietnamese_characters.index(char) for
char in encrypted_message]
    decrypted_vector = []
    for i in range(0, len(encrypted_vector), n):
        block = encrypted_vector[i:i+n]
        decrypted_block = np.dot(key_inverse_matrix, block) %
len(vietnamese_characters)
        decrypted_vector.extend(decrypted_block)
    decrypted_message = ".join(vietnamese_characters[num] for
num in decrypted_vector)
    return decrypted_message

# Hàm ghi thông điệp đã giải mã vào tệp tin
def save_decrypted_message(decrypted_message, file_path):
    with open(file_path, 'w', encoding='utf-8') as file:
        file.write(decrypted_message)

if __name__ == "__main__":

```

```
with open('encrypted_message.txt', 'r', encoding='utf-8') as
file:
    encrypted_message = file.read().strip()
    key_matrix = np.array([[62, 96], [115, 128]])
    decrypted_message = hill_decrypt(encrypted_message,
key_matrix)
    save_decrypted_message(decrypted_message,
'decrypted_message.txt')
    print("Decrypted Message saved to decrypted_message.txt")
```

CHƯƠNG 3: KẾT LUẬN VÀ ĐỊNH HƯỚNG

1. Ưu và nhược điểm của hệ mật mã Hill

Ưu điểm:

Thực hiện đơn giản (dựa trên phép nhân ma trận).

- So với mã dịch vòng, mã thay thế thường, mã Vigenere và mã Affine thì mã Hill sử dụng phép nhân ma trận cho độ bảo mật thông tin cao hơn.

- Không gian khóa lớn $n^{m \times m}$, số lượng khóa không có công thức tường minh để tính.

- Thuật toán dễ cài đặt.

- Mật mã Hill thuộc mật mã khóa bí mật, trong khi mô hình của mật mã khóa bí mật có thể được thiết kế cho các dạng dữ liệu yêu cầu tốc độ xử lý cao. Một số phần cứng thực thi mật mã khóa bí mật có thể thực hiện mã hóa với tốc độ hàng trăm Mbytes/s, trong khi các phần mềm thực thi mật mã khóa bí mật cũng có thể thực hiện tốc độ Mbytes/s. Do đó, mật mã Hill cũng có ưu điểm này.

- Khóa sử dụng cho mật mã Hill là ma trận vuông, tương đối ngắn, ngắn hơn rất nhiều so sánh với khóa sử dụng trong một số hệ mật thuộc mật mã khóa công khai.

- Mật mã Hill có thể được sử dụng kết hợp với một số mật mã khác như mã dịch vòng, mã Vigenere... để tạo nên hệ mật an toàn hơn, độ phức tạp cao hơn từ các nguyên lý khác nhau.

Nhược điểm:

- Hệ mật Hill là hệ mật cổ điển, hiện nay ít được sử dụng do có nhiều hệ mật mới tối ưu hơn.

- Nếu biết m cũng có thể dễ dàng dùng máy tính để giải mã.

- Do sử dụng phép nhân ma trận nên thời gian lập mã lâu hơn so với một số hệ mật cổ điển khác như mã Vigenere, mã dịch vòng...

- Trong quá trình liên lạc giữa hai đầu mỗi, khóa bí mật cần phải được thỏa thuận hoặc phân phối trước qua một kênh bí mật, an toàn.

- Số cặp khóa phải phân phối giữa các bên liên lạc tăng nhanh theo quy mô của mạng. Đồng thời việc thay đổi các đầu mối liên lạc trong mạng phức tạp và cần có quy trình nghiêm ngặt hơn so với mật mã khóa công khai.

- Khó sử dụng trong việc xác thực công khai trong các sơ đồ chữ ký số.

2. Kết luận về hệ mật mã Hill

Hệ mật mã Hill trong Python là một phương pháp mã hóa đơn giản nhưng hiệu quả, sử dụng ma trận để mã hóa và giải mã văn bản. Nó có nhiều ưu điểm như dễ dàng triển khai, hiệu quả tính toán và có thể mở rộng cho các bảng chữ cái khác nhau. Tuy nhiên, hệ mật mã Hill cũng có một số hạn chế như dễ bị tấn công bẻ khóa nếu ma trận khóa được tiết lộ và không phù hợp để mã hóa các tập dữ liệu lớn.

3. Định hướng phát triển hệ mật mã Hill

Dưới đây là một số định hướng phát triển tiềm năng cho hệ mật mã Hill trong Python:

- *Cải thiện tính bảo mật:* Có thể sử dụng các kỹ thuật mã hóa tiên tiến hơn kết hợp với ma trận Hill để tăng cường tính bảo mật cho hệ thống. Ví dụ, có thể sử dụng mã hóa khóa công khai hoặc kết hợp với các phương pháp mã hóa khác như mã hóa luồng.
- *Tăng tốc độ:* Đối với các tập dữ liệu lớn, có thể tối ưu hóa thuật toán mã hóa và giải mã Hill để tăng tốc độ xử lý.
- *Mở rộng khả năng:* Có thể mở rộng hệ thống để hỗ trợ các bảng chữ cái khác nhau, bao gồm cả các bảng chữ cái phi Latinh.
- *Phát triển giao diện người dùng:* Có thể phát triển giao diện người dùng thân thiện để người dùng dễ dàng sử dụng hệ thống mã hóa Hill.
- *Tích hợp với các ứng dụng khác:* Có thể tích hợp hệ thống mã hóa Hill với các ứng dụng khác, ví dụ như ứng dụng nhắn tin hoặc lưu trữ đám mây.

Ngoài ra, cũng cần nghiên cứu và phát triển các phương pháp tấn công và phòng thủ mới cho hệ mật mã Hill để đảm bảo tính bảo mật và hiệu quả của hệ thống.

TÀI LIỆU THAM KHẢO

- [1] An toàn thông tin, Ts. Lê Văn Phùng, NXB Thông tin và truyền thông
- [2] Giáo trình an toàn và bảo mật thông tin, Trường Đại học Giao thông vận tải