HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

GIÁO TRÌNH

CƠ SỞ MẬT MÃ HỌC

Chủ biên: GS.TS Nguyễn Bình
Cộng tác viên: TS. Ngô Đức Thiện
Khoa KTĐT1 - Học viện CNBCVT

Hà Nội - 2013

MỤC LỤC

LỜI NÓI ĐẦU ..... **i**

MỤC LỤC ..... **iii**

CHƯƠNG 1. NHẬP MÔN MẬT MÃ HỌC **1**

1.1. SƠ ĐỒ KHỐI ĐƠN GIẢN CỦA MỘT HỆ THỐNG THÔNG TIN SỐ 1

1.2. SƠ LƯỢC VỀ MẬT MÃ HỌC 2

1.3. THUẬT TOÁN VÀ ĐỘ PHỨC TẠP 3

1.3.1. Khái niệm về thuật toán 3

1.3.2. Độ phức tạp của thuật toán 4

1.4. LÝ THUYẾT THÔNG TIN TRONG CÁC HỆ MẬT 7

1.4.1. Độ mật hoàn thiện. 7

1.4.2. Entropy 13

BÀI TẬP CHƯƠNG 1. 22

CHƯƠNG 2. MẬT MÃ KHÓA BÍ MẬT **24**

2.1. SƠ ĐỒ KHỐI MỘT HỆ TRUYỀN TIN MẬT 24

2.2. MẬT MÃ THAY THẾ 25

2.2.1. Mật mã dịch vòng (MDV) 25

2.2.2. Mã thay thế (MTT) 26

2.2.3. Mật mã Vigenère 26

2.3. MẬT MÃ HOÁN VỊ (MHV) 31

2.4. MẬT MÃ HILL 32

2.5. HỆ MẬT XÂY DỰNG TRÊN CÁC CẤP SỐ NHÂN XYCLIC CỦA VÀNH ĐA THỨC 36

2.5.1. Nhóm nhân của vành 36

2.5.2. Các phần tử cấp n và các nhóm nhân xyclic cấp n 37

2.5.3. Hệ mật xây dựng trên các cấp số nhân xyclic 38

2.6. CÁC HỆ MẬT MÃ TÍCH 44

2.7. CÁC HỆ MÃ DÒNG 46

2.7.1. Sơ đồ chức năng của hệ mật mã dòng 46

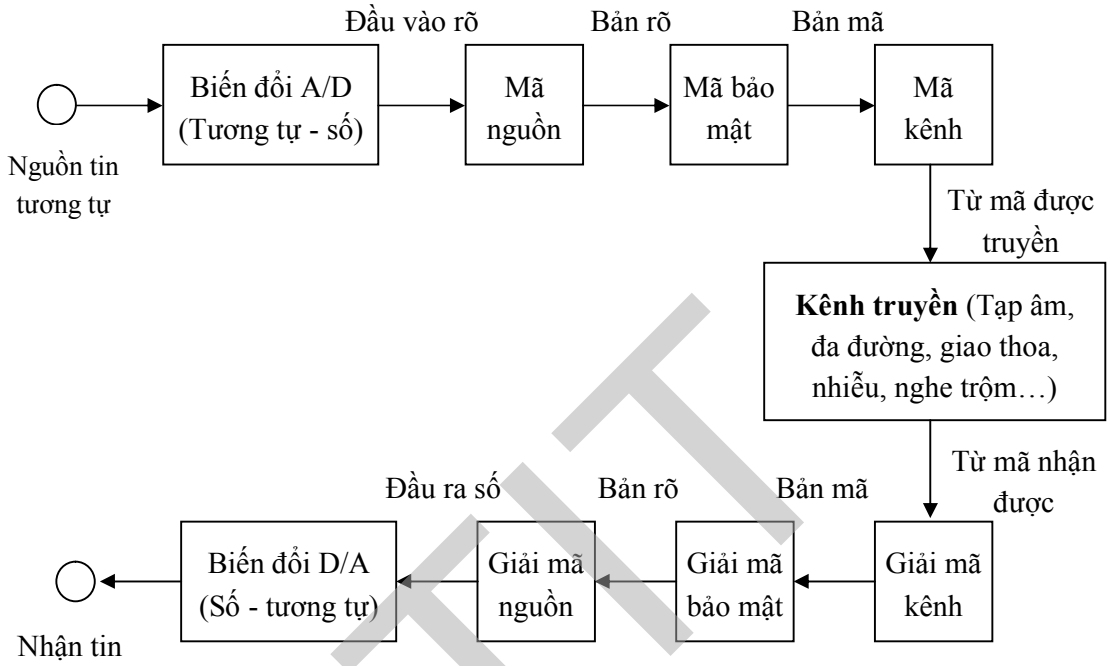
2.7.2. Tạo dãy giả ngẫu nhiên (M-dãy) 48

2.8. CHUẨN MÃ DỮ LIỆU 53

2.8.1. Mở đầu 53

CHƯƠNG 1. NHẬP MÔN MẬT MÃ HỌC

1.1. SƠ ĐỒ KHỐI ĐƠN GIẢN CỦA MỘT HỆ THỐNG THÔNG TIN SỐ



Hình 1.1. Sơ đồ hệ thống thông tin số

Trường hợp nguồn tin đầu vào là nguồn tin số thì không cần bộ biến đổi A/D ở đầu vào và bộ biến đổi D/A ở đầu ra

Trong hệ thống này khối mã bảo mật có chức năng bảo vệ cho thông tin không bị khai thác bất hợp pháp, chống lại các tấn công sau:

- Thám mã thụ động: bao gồm các hoạt động:
  + Thu chặn.
  + Dò tìm.
  + So sánh tương quan.
  + Suy diễn.
- Thám mã tích cực: bao gồm các hoạt động:
  + Giả mạo.
  + Ngụy trang.
  + Sử dụng lại.
  + Sửa đổi.

1.2. SƠ LƯỢC VỀ MẬT MÃ HỌC

Khoa học về mật mã (cryptology) bao gồm:

-Mật mã học (cryptography).

-Phân tích mật mã (cryptanalysis).

Mật mã học là khoa học nghiên cứu cách ghi bí mật thông tin nhằm biến bản tin rõ thành các bản mã.

Phân tích mã là khoa học nghiên cứu cách phá các hệ mật nhằm phục hồi bản rõ ban đầu từ bản mã. Việc tìm hiểu các thông tin về khóa và các phương pháp biến đổi thông tin cũng là một nhiệm vụ quan trọng của phân tích mật mã.

Có ba phương pháp tấn công cơ bản của thám mã:

- Tìm khóa vét cạn.
- Phân tích thống kê.
- Phân tích toán học.

Việc tấn công của thám mã có thể được thực hiện với các giả định:

- Tấn công chỉ với bản mã.
- Tấn công với bản rõ đã biết.
- Tấn công với các bản rõ được chọn.
- Tấn công với các bản mã được chọn.

Chú ý:

- Một hệ mật có thể bị phá chỉ với bản mã thường là hệ mật có độ an toàn thấp.
- Một hệ mật là an toàn với kiểu tấn công có các bản rõ được chọn thường là một hệ mật có độ an toàn cao.

Có 4 loại hệ mật mã sau:

- Hệ mật mã dòng.
- Hệ mật mã khối đối xứng.
- Hệ mật mã có hồi tiếp mật mã.
- Hệ mật mã khóa công khai (Bất đối xứng).

Ta sẽ nghiên cứu các loại hệ mật trên ở các chương sau.

Khi xây dựng một hệ mật người ta thường xem xét tới các tiêu chuẩn sau:

- Độ mật cần thiết.
- Kích thước không gian khóa.
- Tính đơn giản và tốc độ mã hóa và giải mã.
- Tính lan truyền sai.
- Tính mở rộng bản tin.

1.3. THUẬT TOÁN VÀ ĐỘ PHỨC TẠP

1.3.1. Khái niệm về thuật toán

1.3.1.1. Định nghĩa thuật toán

Có thể định nghĩa thuật toán theo nhiều cách khác nhau. Ở đây ta không có ý định trình bày chặt chẽ về thuật toán mà sẽ hiểu khái niệm thuật toán theo một cách thông thường nhất.

Định nghĩa 1.1:

Thuật toán là một quy tắc để với những dữ liệu ban đầu đã cho, tìm được lời giải của bài toán được xét sau một khoảng thời gian hữu hạn.

Để minh họa cách ghi một thuật toán cũng như tìm hiểu các yêu cầu đề ra cho thuật toán, ta xét trên các ví dụ cụ thể sau đây:

Cho n số $X[1],X[2],...,X[n]$ ta cần tìm m và j sao cho:

$$m = X[j] = \max_{1 \le k \le n} X[k]$$

Và j là lớn nhất có thể. Điều đó có nghĩa là cần tìm cực đại của các số đã cho và chỉ số lớn nhất trong các số cực đại.

Với mục tiêu tìm số cực đại với chỉ số lớn nhất, ta xuất phát từ giá trị $X[n]$. Bước thứ nhất, vì mới chỉ có một số ta có thể tạm thời xem $m = X[n]$ và $j = n$. Tiếp theo ta so sánh $X[n]$ với $X[n-1]$. Nếu $X[n]$ không nhỏ hơn $X[n-1]$ thì ta giữ nguyên, trong trường hợp ngược lại, $X[n-1]$ chính là số cực đại trong hai số đã xét và ta phải thay đổi m và j. Đặt $m = X[n-1]$, $j = 1,...,n-1$. Với cách làm như trên, ở mỗi bước ta luôn nhận được số cực đại trong số những số đã xét. Bước tiếp theo là so sánh nó với những số đứng trước hoặc kết thúc thuật toán trong trường hợp không còn số nào đứng trước nó.

1.3.1.2. Thuật toán tìm cực đại

M1: [Bước xuất phát] đặt $j \leftarrow n$, $k \leftarrow n-1$, $m \leftarrow X[n]$

M2: [Đã kiểm tra xong?]. Nếu $k = 0$, thuật toán kết thúc.

M3: [So sánh]. Nếu $X[k] \le m$, chuyển sang M5

M4: [Thay đổi m]. Đặt $j \leftarrow k$, $m \leftarrow X[k]$ (Tạm thời m đang là cực đại).

M5: [Giảm k]. Đặt $k \leftarrow k-1$ quay về M2.

Dấu "$\leftarrow$" dùng để chỉ một phép toán quan trọng là phép thay chỗ (replacement).

Trên đây ta ghi thuật toán bằng ngôn ngữ thông thường. Trong trường hợp thuật toán được viết bằng ngôn ngữ của máy tính, ta có một chương trình.

Trong thuật toán có những số liệu ban đầu được cho trước khi thuật toán bắt đầu làm việc được gọi là các đầu vào (input). Trong thuật toán trên đầu vào là các số $X[1],X[2],...,X[n]$.

Một thuật toán có thể có một hoặc nhiều đầu ra (output). Trong thuật toán trên các đầu ra là m và j.

Có thể thấy rằng thuật toán vừa mô tả thỏa mãn các yêu cầu của một thuật toán nói chung, đó là:

- *Tính hữu hạn:* Thuật toán cần phải kết thúc sau một số hữu hạn bước. Khi thuật toán ngừng làm việc ta phải thu được câu trả lời cho vấn đề đặt ra. Thuật toán m rõ ràng thỏa mãn điều kiện này, vì ở mỗi bước ta luôn chỉ từ việc xem xét một số sang số đứng trước nó và số các số là hữu hạn.
- *Tính xác định:* Ở mỗi bước thuật toán cần phải xác định, nghĩa là chỉ rõ việc cần làm. Nếu đối với người đọc thuật toán trên chưa thỏa mãn được điều kiện này thì đó là lỗi của người viết.

Ngoài những yếu tố kể trên, ta còn phải xét đến tính hiệu quả của thuật toán. Có rất nhiều thuật toán về mặt lý thuyết là hữu hạn bước, tuy nhiên thời gian "hữu hạn" đó vượt quá khả năng làm việc của chúng ta. Những thuật toán đó sẽ không được xét đến ở đây, vì chúng ta chỉ quan tâm những thuật toán có thể sử dụng thực sự trên máy tính.

Cũng do mục tiêu trên, ta còn phải chú ý đến độ phức tạp của các thuật toán. Độ phức tạp của một thuật toán có thể được đo bằng không gian tức là dung lượng bộ nhớ của máy tính cần thiết để thực hiện thuật toán và bằng thời gian, tức là thời gian máy tính làm việc. Ở đây khi nói đến độ phức tạp của thuật toán ta luôn hiểu là độ phức tạp của thời gian.

1.3.2. Độ phức tạp của thuật toán

Thời gian làm việc của máy tính khi chạy một thuật toán nào đó không chỉ phụ thuộc vào thuật toán mà còn phụ thuộc vào máy tính được sử dụng. Vì thế, để có một tiêu chuẩn chung, ta sẽ đo độ phức tạp của một thuật toán bằng số các phép tính phải làm khi thực hiện thuật toán. Khi tiến hành cùng một thuật toán, số các phép tính phải thực hiện còn phụ thuộc vào cỡ của bài toán, tức là độ lớn của đầu vào. Vì thế độ phức tạp của thuật toán sẽ là một hàm số của độ lớn đầu vào. Trong những ứng dụng thực tiễn, chúng ta không cần biết chính xác hàm này mà chỉ cần biết "cỡ" của chúng, tức là cần có một ước lượng đủ tốt của chúng.

Trong khi làm việc, máy tính thường ghi các chữ số bằng bóng đèn "sáng, tắt", bóng đèn sáng chỉ số 1, bóng đèn tắt chỉ số 0. Vì thế để thuận tiện nhất là dùng hệ đếm cơ số 2, trong đó để biểu diễn một số, ta chỉ cần dùng hai ký hiệu 0 và 1. Một ký hiệu 0 hoặc 1 được gọi là 1bit "viết tắt của binary digit". Một số nguyên n biểu diễn bởi k chữ số 1 và 0 được gọi là một số $k-bit$.

Độ phức tạp của một thuật toán được đo bằng số các phép tính bit. Phép tính bit là một phép tính logic hay số học thực hiện trên các số một bit 0 và 1.

Để ước lượng độ phức tạp của thuật toán ta dùng khái niệm bậc O lớn.

Định nghĩa 1.2:

Giả sử $f[n]$ và $g[n]$ là hai hàm xác định trên tập hợp các số nguyên dương. Ta nói $f[n]$ có bậc O-lớn của $g[n]$ và viết $f[n]=O(g[n])$, nếu tồn tại một số $C>0$ sao cho với n đủ lớn. Các hàm $f[n]$ và $g[n]$ đều dương thì $f[n]<C\,g[n]$.

Ví dụ 1.1. :

1) Giả sử $f[n]$ là đa thức: $f[n]=a_d n^d+a_{d-1}n^{d-1}+...+a_1 n+a_0$ trong đó $a_d>0$. Dễ chứng minh $f[n]=O(n^d)$.

2) Nếu $f[n]=O(g[n])$, $f_2[n]=O(g[n])$ thì $f_1+f_2=O(g)$.

3) Nếu $f_1=O(g_1)$, $f_2=O(g_2)$ thì $f_1 f_2=O(g_1 g_2)$.

4) Nếu tồn tại giới hạn hữu hạn:

$$\lim_{n\to\infty}\frac{f[n]}{g[n]}$$

thì $f=O(g)$

5) Với mọi số $\varepsilon>0$, $\log n=O(n^{\varepsilon})$

Định nghĩa 1.3:

Một thuật toán được gọi là có độ phức tạp đa thức hoặc có thời gian đa thức, nếu số các phép tính cần thiết để thực hiện thuật toán không vượt quá $O(\log^d n)$, trong đó n là độ lớn của đầu vào và d là số nguyên dương nào đó.

Nói cách khác nếu đầu vào là các số k bit thì thời gian thực hiện thuật toán là $O(k^d)$, tức là tương đương với một đa thức của k.

Các thuật toán với thời gian $O(n^{\alpha})$, $\alpha>0$ được gọi là thuật toán với độ phức tạp mũ hoặc thời gian mũ.

Chú ý rằng nếu một thuật toán nào đó có độ phức tạp $O(g)$ thì cũng có thể nói nó có độ phức tạp $O(h)$ với mọi hàm $h>g$. Tuy nhiên ta luôn luôn cố gắng tìm ước lượng tốt nhất có thể để tránh hiểu sai về độ phức tạp thực sự của thuật toán.

Cũng có những thuật toán có độ phức tạp trung gian giữa đa thức và mũ. Ta thường gọi đó là thuật toán dưới mũ. Chẳng hạn thuật toán nhanh nhất được biết hiện nay để phân tích một số nguyên n ra thừa số là thuật toán có độ phức tạp:

$$\exp\left(\sqrt{\log n\,\log\log n}\right)$$

Khi giải một bài toán không những ta chỉ cố gắng tìm ra một thuật toán nào đó, mà còn muốn tìm ra thuật toán "tốt nhất". Đánh giá độ phức tạp là một trong những cách để phân tích, so sánh và tìm ra thuật toán tối ưu. Tuy nhiên độ phức tạp không phải là tiêu chuẩn duy nhất để đánh giá thuật toán. Có những thuật toán về lý thuyết thì có độ phức tạp cao hơn một thuật toán khác, nhưng khi sử dụng lại có kết quả (gần đúng) nhanh hơn nhiều. Điều này còn tùy thuộc những bài toán cụ thể, những mục tiêu cụ thể và cả kinh nghiệm của người sử dụng.

Chúng ta cần lưu ý thêm một số điểm sau đây. Mặc dù định nghĩa thuật toán mà chúng ta đưa ra chưa phải là chặt chẽ, nó vẫn quá "cứng nhắc" trong những ứng dụng thực tế. Bởi vậy chúng ta còn cần đến các thuật toán "xác suất", tức là các thuật toán phụ thuộc vào một hay nhiều tham số ngẫu nhiên. Những "thuật toán" này về nguyên tắc không được gọi là thuật toán vì chúng có thể với xác suất rất bé, không bao giờ kết thúc. Tuy nhiên thực nghiệm chỉ ra rằng, các thuật toán xác suất thường hữu hiệu hơn các thuật toán không xác suất. Thậm chí trong rất nhiều trường hợp, chỉ có các thuật toán như thế là sử dụng được.

Khi làm việc với các thuật toán xác suất, ta thường hay phải sử dụng các số "ngẫu nhiên". Khái niệm chọn số ngẫu nhiên cũng cần được chính xác hóa. Thường thì người ta sử dụng một "máy" sản xuất số giả ngẫu nhiên nào đó. Tuy nhiên ở đây khi nói đến việc chọn số ngẫu nhiên ta hiểu đó là được thực hiện trên máy.

Cần chú ý ngay rằng, đối với các thuật toán xác suất, không thể nói đến thời gian tuyệt đối, mà chỉ có thể nói đến thời gian hy vọng (expected).

Để hình dung được phần nào "độ phức tạp" của các thuật toán khi làm việc với những số lớn, ta xem Bảng 1.1 dưới đây cho khoảng thời gian cần thiết để phân tích một số nguyên n ra thừa số nguyên tố bằng thuật toán nhanh nhất được biết hiện nay.

Bảng 1.1. Độ phức tạp để phân tích số nguyên ra thừa số nguyên tố

Số chữ số thập phân	**Số phép tính bit**	**Thời gian**
50	$1,4.10^{10}$	3,9 giờ
75	9.10^{12}	104 ngày
100	$2,3.10^{15}$	74 năm
200	$1,2.10^{23}$	$3,8.10^{9}$ năm
300	$1,5.10^{29}$	$4,9.10^{15}$ năm
500	$1,3.10^{39}$	$4,2.10^{25}$ năm

Từ Bảng 1.1 trên, ta thấy rằng ngay với một thuật toán dưới mũ, thời gian làm việc với các số nguyên lớn là quá lâu. Vì thế nói chung người ta luôn cố gắng tìm những thuật toán đa thức.

1.4. LÝ THUYẾT THÔNG TIN TRONG CÁC HỆ MẬT

Năm 1949, Claude Shannon đã công bố một bài báo có nhan đề "Lý thuyết thông tin trong các hệ mật" trên tạp chí "The Bell System Technical Journal". Bài báo đã có ảnh hưởng lớn đến việc nghiên cứu khoa học mật mã. Trong chương này ta sẽ thảo luận một vài ý tưởng trong lý thuyết của Shannon.

1.4.1. Độ mật hoàn thiện.

Có hai quan điểm cơ bản về độ an toàn của một hệ mật.

1.4.1.1. Độ an toàn tính toán

Độ đo này liên quan đến những nỗ lực tính toán cần thiết để phá một hệ mật. Một hệ mật là an toàn về mặt tính toán nếu một thuật toán tốt nhất để phá nó cần ít nhất N phép toán, N là số rất lớn nào đó. Vấn đề là ở chỗ, không có một hệ mật thực tế đã biết nào có thể được chứng tỏ là an toàn theo định nghĩa này. Trên thực tế, người ta gọi một hệ mật là "an toàn về mặt tính toán" nếu có một phương pháp tốt nhất phá hệ này nhưng yêu cầu thời gian lớn đến mức không chấp nhận được. (Điều này tất nhiên là rất khác với việc chứng minh về độ an toàn).

Một quan điểm chứng minh về độ an toàn tính toán là quy độ an toàn của một hệ mật về một bài toán đã được nghiên cứu kỹ và bài toán này được coi là khó. Ví dụ, ta có thể chứng minh một khẳng định có dạng. "Một hệ mật đã cho là an toàn nếu không thể phân tích ra thừa số một số nguyên n cho trước". Các hệ mật loại này đôi khi gọi là "An toàn chứng minh được". Tuy nhiên cần phải hiểu rằng, quan điểm này chỉ cung cấp một chứng minh về độ an toàn có liên quan để một bài toán khác chứ không phải là một chứng minh hoàn chỉnh về độ an toàn. (Tình hình này cũng tương tự như việc chứng minh một bài toán là NP đầy đủ: Có thể chứng tỏ bài toán đã cho chí ít cũng khó như một bài toán NP đầy đủ khác, song không phải là một chứng minh hoàn chỉnh về độ khó tính toán của bài toán).

1.4.1.2. Độ an toàn không điều kiện

Độ đo này liên quan đến độ an toàn của các hệ mật khi không có một hạn chế nào được đặt ra về khối lượng tính toán mà Oscar được phép thực hiện. Một hệ mật được gọi là an toàn không điều kiện nếu nó không thể bị phá thậm chí với khả năng tính toán không hạn chế.

Khi thảo luận về độ an toàn của một hệ mật, ta cũng phải chỉ ra kiểu tấn công đang được xem xét. Trong chương ta thấy rằng, không một hệ mật nào trong các hệ mã dịch vòng, mã thay thế và mã Vigenère được coi là an toàn về mặt tính toán với phương pháp tấn công chỉ với bản mã (Với khối lượng bản mã thích hợp).

Điều mà ta sẽ làm trong phần này là để phát triển lý thuyết về các hệ mật có độ an toàn không điều kiện với phương pháp tấn công chỉ với bản mã. Có thể thấy rằng, cả ba hệ mật nêu trên đều là các hệ mật an toàn vô điều kiện chỉ khi mỗi phần tử của bản rõ được mã hoá bằng một khoá cho trước.

Rõ ràng là độ an toàn không điều kiện của một hệ mật không thể được nghiên cứu theo quan điểm độ phức tạp tính toán vì thời gian tính toán cho phép không hạn chế. Ở đây lý

thuyết xác suất là nền tảng thích hợp để nghiên cứu về độ an toàn không điều kiện. Tuy nhiên ta chỉ cần một số kiến thức sơ đẳng trong xác suất; các định nghĩa chính sẽ được nêu dưới đây.

Định nghĩa 1.4:

Giả sử X và Y là các biến ngẫu nhiên. Kí hiệu xác suất để X nhận giá trị x là p(x) và để Y nhận giá trị y là $p(y)$. *Xác suất đồng thời* $p(x,y)$ *là xác suất để X nhận giá trị x và Y nhận giá trị y. Xác suất có điều kiện* $p(x|y)$ *là xác suất để X nhận giá trị x với điều kiện Y nhận giá trị y. Các biến ngẫu nhiên X và Y được gọi là độc lập nếu* $p(x,y) = p(x)p(y)$ *với mọi giá trị có thể x của X và y của Y.*

Quan hệ giữa xác suất đồng thời và xác suất có điều kiện được biểu thị theo công thức:

$$p(x,y) = p(x|y)\,p(y) \tag{1.1}$$

Đổi chỗ x và y ta có:

$$p(x,y) = p(y|x)\,p(x) \tag{1.2}$$

Từ hai biểu thức trên ta có thể rút ra kết quả sau:(được gọi là định lý Bayes)

Định lý 1.1: (Định lý Bayes)

Nếu $p(y) > 0$ thì:

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)} \tag{1.3}$$

Hệ quả 1.1.

x và y là các biến độc lập khi và chỉ khi: $p(x|y) = p(x)$, $\forall x,y$.

Trong phần này ta giả sử rằng, một khoá cụ thể chỉ dùng cho một bản mã. Giả sử có một phân bố xác suất trên không gian bản rõ $\mathcal{P}$. Kí hiệu xác suất tiên nghiệm để bản rõ xuất hiện là $p_{\mathcal{P}}(x)$. Cũng giả sử rằng, khóa *K* được chọn (bởi Alice và Bob) theo một phân bố xác suất xác định nào đó. (Thông thường khoá được chọn ngẫu nhiên, bởi vậy tất cả các khoá sẽ đồng khả năng, tuy nhiên đây không phải là điều bắt buộc). Kí hiệu xác suất để khóa *K* được chọn là $p_{\mathcal{K}}(K)$. Cần nhớ rằng khóa được chọn trước khi Alice biết bản rõ. Bởi vậy có thể giả định rằng khoá *K* và bản rõ x là các sự kiện độc lập.

Hai phân bố xác suất trên $\mathcal{P}$ và $\mathcal{K}$ sẽ tạo ra một phân bố xác suất trên $\mathcal{C}$. Thật vậy, có thể dễ dàng tính được xác suất $p_{\mathcal{C}}(y)$ với y là bản mã được gửi đi. Với một khoá $K \in \mathcal{K}$, ta xác định:

$$C(K) = \{e_K(x) : x \in \mathcal{P}\}$$

Ở đây $C(K)$ biểu thị tập các bản mã có thể nếu *K* là khóa. Khi đó với mỗi $y \in C$, ta có:

$$p_C(y) = \sum_{\{K : y \in C(K)\}} p_{\mathcal{K}}(K) p_{\mathcal{P}}(d_K(y)) \tag{1.4}$$

Nhận thấy rằng, với bất kì $y \in C$ và $x \in \mathcal{P}$, có thể tính được xác suất có điều kiện $p_C(y|x)$. (Tức là xác suất để y là bản mã với điều kiện bản rõ là x):

$$p_C(y|x) = \sum_{\{K : x = d_K(y)\}} p_{\mathcal{K}}(K) \tag{1.5}$$

Bây giờ ta có thể tính được xác suất có điều kiện $p_{\mathcal{P}}(x|y)$ (tức xác suất để x là bản rõ với điều kiện y là bản mã) bằng cách dùng định lý Bayes. Ta thu được công thức sau:

$$p_{\mathcal{P}}(y|x) = \frac{p_{\mathcal{P}}(x) = \sum_{\{K : x = d_K(y)\}} p_{\mathcal{K}}(K)}{\sum_{\{K : y \in c(K)\}} p_{\mathcal{K}}(K) p_{\mathcal{P}}(d_K(y))} \tag{1.6}$$

Các phép tính này có thể thực hiện được nếu biết được các phân bố xác suất.

Sau đây sẽ trình bày một ví dụ đơn giản để minh hoạ việc tính toán các phân bố xác suất này.

Ví dụ 1.2.

Giả sử $\mathcal{P} = \{a, b\}$ với $p_{\mathcal{P}}(a) = 1/4$, $p_{\mathcal{P}}(b) = 3/4$.

Cho $K = \{K_1, K_2, K_3\}$ với $p_{\mathcal{K}}(K_1) = 1/2$, $p_{\mathcal{K}}(K_2) = p_{\mathcal{K}}(K_3) = 1/4$.

Giả sử $C = \{1, 2, 3, 4\}$ và các hàm mã được xác định là $e_{K_1}(a) = 1, e_{K_1}(b) = 2$, $e_{K_2}(a) = 2, e_{K_2}(b) = 3, e_{K_3}(a) = 3, e_{K_3}(b) = 4$. Hệ mật này được biểu thị bằng ma trận mã hoá sau:

	a	b
K_1	1	2
K_2	2	3
K_3	2	4

Tính phân bố xác suất p_C ta có:

$$p_C(1) = 1/8$$
$$p_C(2) = 3/8 + 1/16 = 7/16$$
$$p_C(3) = 3/16 + 1/16 = 1/4$$
$$p_C(4) = 3/16$$

Bây giờ ta đã có thể các phân bố xác suất có điều kiện trên bản rõ với điều kiện đã biết bản mã. Ta có:

$$p_{\mathcal{P}}(a|1)=1 \quad p_{\mathcal{P}}(b|1)=0 \quad p_{\mathcal{P}}(a|2)=1/7 \quad p_{\mathcal{P}}(b|2)=6/7$$
$$p_{\mathcal{P}}(a|3)=1/4 \quad p_{\mathcal{P}}(b|3)=3/4 \quad p_{\mathcal{P}}(a|4)=0 \quad p_{\mathcal{P}}(b|4)=1$$

Bây giờ ta đã có đủ điều kiện để xác định khái niệm về độ mật hoàn thiện. Một cách không hình thức, độ mật hoàn thiện có nghĩa là Oscar với bản mã trong tay không thể thu được thông tin gì về bản rõ. Ý tưởng này sẽ được làm chính xác bằng cách phát biểu nó theo các thuật ngữ của các phân bố xác suất định nghĩa ở trên như sau:

Định nghĩa 1.5:

Một hệ mật có độ mật hoàn thiện nếu $p_{\mathcal{P}}(x|y)=p_{\mathcal{P}}(x)$ *với mọi* $x\in\mathcal{P}, y\in C$. *Tức xác suất hậu nghiệm để bản rõ là* x *với điều kiện đã thu được bản mã* y *là đồng nhất với xác suất tiên nghiệm để bản rõ là* x.

Trong ví dụ trên chỉ có bản mã 3 mới thoả mãn tính chất độ mật hoàn thiện, các bản mã khác không có tính chất này.

Sau đây sẽ chứng tỏ rằng, mã dịch vòng (MDV - xem chương 2) có độ mật hoàn thiện. Về mặt trực giác, điều này dường như quá hiển nhiên. Với mã dịch vòng, nếu đã biết một phần tử bất kỳ của bản mã $y\in Z_{26}$, thì một phần tử bất kỳ của bản rõ $x\in Z_{26}$ cũng có thể là bản mã đã giải của y tuỳ thuộc vào giá trị của khoá. Định lý sau cho một khẳng định hình thức hoá và được chứng minh theo các phân bố xác suất.

Định lý 1.2:

Giả sử 26 khoá trong MDV có xác suất như nhau và bằng1/26. Khi đó MDV sẽ có độ mật hoàn thiện với mọi phân bố xác suất của bản rõ.

Chứng minh: Ta có $\mathcal{P}=C=\mathcal{K}=Z_{26}$ và với $0\le K\le 25$, quy tắc mã hoá e_K là $e_K(x)=x+K \bmod 26$ ($x\in Z_{26}$). Trước tiên tính phân bố p_C. Giả sử $y\in Z_{26}$, khi đó:

$$\begin{aligned} p_C(y) &= \sum_{K\in Z_{26}} p_{\mathcal{K}}(K)\,p_{\mathcal{P}}(d_K(y)) \\ &= \sum_{K\in Z_{26}} 1/26\,p_{\mathcal{P}}(y-K) \\ &= 1/26\sum_{K\in Z_{26}} p_{\mathcal{P}}(y-K) \end{aligned} \qquad (1.7)$$

Xét thấy với y cố định, các giá trị $y-K \bmod 26$ sẽ tạo thành một hoán vị của Z_{26} và $p_{\mathcal{P}}$ là một phân bố xác suất. Bởi vậy ta có:

$$\sum_{K\in Z_{26}} p_{\mathcal{P}}(y-K)=\sum_{K\in Z_{26}} p_{\mathcal{P}}(y)=1$$

Do đó: $p_C(y)=1/26$ với bất kỳ $y\in Z_{26}$.

Tiếp theo ta có:

$$p_C(y|x) = p_K(y - x \bmod 26) = 1/26$$

Với mọi x, y vì với mỗi cặp x, y khóa duy nhất K (khoá đảm bảo $e_K(x) = y$) là khoá $K = y - x \bmod 26$. Bây giờ sử dụng định lý Bayes, ta có thể dễ dàng tính:

$$\begin{aligned} p_C(x|y) &= \frac{p_{\mathcal{P}}(x)p_C(y|x)}{p_C(y)} \\ &= \frac{p_{\mathcal{P}}(x).(1/26)}{(1/26)} \\ &= p_{\mathcal{P}}(x) \end{aligned}$$

Bởi vậy, MDV có độ mật hoàn thiện.

Như vậy, mã dịch vòng là hệ mật không phá được miễn là chỉ dùng một khoá ngẫu nhiên đồng xác suất để mã hoá mỗi ký tự của bản rõ.

Sau đây sẽ nghiên cứu độ mật hoàn thiện trong trường hợp chung. Trước tiên thấy rằng, (sử dụng định lý Bayes) điều kiện để $p_{\mathcal{P}}(x|y) = p_{\mathcal{P}}(x)$ với mọi $x \in \mathcal{P}, y \in \mathcal{P}$ là tương đương với $p_C(y|x) = p_C(y)$ với mọi $x \in \mathcal{P}, y \in \mathcal{P}$.

Giả sử rằng $p_C(y) > 0$ với mọi $y \in C\,(p_C(y) = 0)$ thì bản mã sẽ không được dùng và có thể loại khỏi C). Cố định một giá trị nào đó $x \in \mathcal{P}$. Với mỗi $y \in C$ ta có $p_C(y|x) = p_C(y) > 0$. Bởi vậy, với mỗi $y \in C$ phải có ít nhất một khoá K và một x sao cho $e_K(x) = y$. Điều này dẫn đến $|\mathcal{K}| \geq |C|$. Trong một hệ mật bất kỳ ta phải có $|C| \geq |\mathcal{P}|$ vì mỗi quy tắc mã hoá là một đơn ánh. Trong trường hợp giới hạn, $|\mathcal{K}| = |C| = |\mathcal{P}|$, ta có định lý sau (Theo Shannon).

Định lý 1.3:

Giả sử $(\mathcal{P}, C, \mathcal{K}, \mathcal{E}, \mathcal{D})$ là một hệ mật , trong đó $|\mathcal{K}| = |C| = |\mathcal{P}|$. Khi đó, hệ mật có độ mật hoàn thiện khi và chỉ khi khoá K được dùng với xác suất như nhau bằng $1/|\mathcal{K}|$, và với mỗi $x \in \mathcal{P}$, mỗi $y \in C$ có một khoá duy nhất K sao cho $e_K(x) = y$.

Chứng minh

Giả sử hệ mật đã cho có độ mật hoàn thiện. Như đã thấy ở trên, với mỗi $x \in \mathcal{P}$ và $y \in C$, phải có ít nhất một khoá K sao cho $e_K(x) = y$. Bởi vậy ta có bất đẳng thức:

$$|C| = |\{e_K(x) : K \in \mathcal{K}\}| = |\mathcal{K}|$$

Tuy nhiên, ta giả sử rằng $|C| = |\mathcal{K}|$, bởi vậy ta phải có:

$$|\{e_K(x) : K \in C\}| = |\mathcal{K}|$$

Tức là ở đây không tồn tại hai khoá K_1 và K_2 khác nhau để $e_{K_1}(x) = e_{K_2}(x) = y$. Như vậy ta đã chứng tỏ được rằng, với bất kỳ $x \in \mathcal{P}$ và $y \in C$ có đúng một khoá K để $e_K(x) = y$.

Ký hiệu $n = |\mathcal{K}|$. Giả sử $\mathcal{P} = \{x_i : 1 \le i \le n\}$ và cố định một giá trị $y \in C$ Ta có thể ký hiệu các khoá $K_1, K_2, ..., K_n$ sao cho $e_{K_i}(x_i) = y_i$, $1 \le i \le n$. Sử dụng định lý Bayes ta có:

$$p_{\mathcal{P}}(x_i|y) = \frac{p_C(y|x_i)p_{\mathcal{P}}(x_i)}{p_C(y)}$$
$$= \frac{p_{\mathcal{K}}(K_i).(p_{\mathcal{P}}(x_i))}{p_C(y)}$$

Xét điều kiện độ mật hoàn thiện $p_{\mathcal{P}}(x_i|y) = p_{\mathcal{P}}(x_i)$. Điều kiện này kéo theo $p_{\mathcal{K}}(K_i) = p_C(y)$ với $1 \le i \le n$. Tức là khoá được dùng với xác suất như nhau (chính bằng $p_C(y)$). Tuy nhiên vì số khoá là $n = |\mathcal{K}|$ nên ta có $p_{\mathcal{K}}(K) = 1/|\mathcal{K}|$ với mỗi $K \in \mathcal{K}$.

Ngược lại, giả sử hai điều giả định đều thoả mãn. Khi đó dễ dàng thấy được hệ mật có độ mật hoàn thiện với mọi phân bố xác suất bất kỳ của bản rõ (tương tự như chứng minh định lý 2.3). Các chi tiết dành cho bạn đọc xem xét.

Mật mã khoá sử dụng một lần của Vernam (One-Time-Pad: OTP) là một ví dụ quen thuộc về hệ mật có độ mật hoàn thiện. Gillbert Vernam lần đầu tiên mô tả hệ mật này vào năm 1917. Hệ OTP dùng để mã và giải mã tự động các bản tin điện báo. Điều thú vị là trong nhiều năm OTP được coi là một hệ mật không thể bị phá nhưng không thể chứng minh cho tới khi Shannon xây dựng được khái niệm về độ mật hoàn thiện hơn 30 năm sau đó.

Mô tả về hệ mật dùng một lần nêu trên hình 1.2.

Giả sử $n \ge 1$ là số nguyên và $\mathcal{P} = C = \mathcal{K} = (Z_2)^n$. Với $\mathcal{K} \in (Z_2)^n$, ta xác định $e_K(x)$ là tổng vector theo modulo 2 của K và x (hay tương đương với phép hoặc loại trừ của hai dãy bit tương ứng). Như vậy, nếu $x = (x_1, ..., x_n)$ và $K = (K_1, ..., K_n)$ thì:

$$e_K(x) = (x_1 + K_1, ..., x_n + K_n) \bmod 2$$

Phép mã hoá là đồng nhất với phép giải mã. Nếu $y = (y_1, ..., y_n)$ thì:

$$d_K(x) = (y_1 + K_1, ..., y_n + K_n) \bmod 2$$

Hình 1.2. Hệ mật sử dụng khoá một lần (OTP)

Sử dụng định lý 2.4, dễ dàng thấy rằng OTP có độ mật hoàn thiện. Hệ thống này rất hấp dẫn do dễ thực hiện mã và giải mã.

Vernam đã đăng ký phát minh của mình với hy vọng rằng nó sẽ có ứng dụng thương mại rộng rãi. Đáng tiếc là có những nhược điểm quan trọng đối với các hệ mật an toàn không điều kiện, chẳng hạn như OTP. Điều kiện $|\mathcal{K}| \ge |\mathcal{P}|$ có nghĩa là lượng khóa (cần được thông

báo một cách bí mật) cũng lớn như bản rõ. Ví dụ , trong trường hợp hệ OTP, ta cần n bit khoá để mã hoá n bit của bản rõ. Vấn đề này sẽ không quan trọng nếu có thể dùng cùng một khoá để mã hoá các bản tin khác nhau; tuy nhiên, độ an toàn của các hệ mật an toàn không điều kiện lại phụ thuộc vào một thực tế là mỗi khoá chỉ được dùng cho một lần mã. Ví dụ OTP không thể đứng vững trước tấn công chỉ với bản rõ đã biết vì ta có thể tính được K bằng phép hoặc loại trừ xâu bit bất kỳ x và $e_K(x)$. Bởi vậy, cần phải tạo một khóa mới và thông báo nó trên một kênh bảo mật đối với mỗi bản tin trước khi gửi đi. Điều này tạo ra khó khăn cho vấn đề quản lý khoá và gây hạn chế cho việc sử dụng rộng rãi OTP. Tuy nhiên OTP vẫn được áp dụng trong lĩnh vực quân sự và ngoại giao, ở những lĩnh vực này độ an toàn không điều kiện có tầm quan trọng rất lớn.

Lịch sử phát triển của mật mã học là quá trình cố gắng tạo các hệ mật có thể dùng một khoá để tạo một xâu bản mã tương đối dài (tức có thể dùng một khoá để mã nhiều bản tin) nhưng chí ít vẫn còn giữ được độ an toàn tính toán. Chuẩn mã dữ liệu (DES) là một hệ mật thuộc loại này.

1.4.2. Entropy

Trong phần trước ta đã thảo luận về khái niệm độ mật hoàn thiện và đặt mối quan tâm vào một trường hợp đặc biệt, khi một khoá chỉ được dùng cho một lần mã. Bây giờ ta sẽ xét điều sẽ xảy ra khi có nhiều bản rõ được mã bằng cùng một khoá và bằng cách nào mà thám mã có thể thực hiện có kết quả phép tấn công chỉ với bản mã trong thời gian đủ lớn.

Công cụ cơ bản trong nghiên cứu bài toán này là khái niệm Entropy. Đây là khái niệm trong lý thuyết thông tin do Shannon đưa ra vào năm 1948. Có thể coi Entropy là đại lượng đo thông tin hay còn gọi là độ bất định. Nó được tính như một hàm của phân bố xác suất.

Giả sử ta có một biến ngẫu nhiên X nhận các giá trị trên một tập hữu hạn theo một phân bố xác suất $p(X)$. Thông tin thu nhận được bởi một sự kiện xảy ra tuân theo một phân bố $p(X)$ là gì? Tương tự, nếu sự kiện còn chưa xảy ra thì cái gì là độ bất định và kết quả bằng bao nhiêu? Đại lượng này được gọi là Entropy của X và được kí hiệu là $H(X)$.

Các ý tưởng này có vẻ như khá trừu tượng, bởi vậy ta sẽ xét một ví dụ cụ thể hơn. Giả sử biến ngẫu nhiên X biểu thị phép tung đồng xu. Phân bố xác suất là: p(mặt xấp) = p(mặt ngửa) = 1/2. Có thể nói rằng, thông tin (hay Entropy) của phép tung đồng xu là một bit vì ta có thể mã hoá mặt xấp bằng 1 và mặt ngửa bằng 0. Tương tự Entropy của n phép tung đồng tiền có thể mã hoá bằng một xâu bit có độ dài n .

Xét một ví dụ phức tạp hơn một chút. Giả sử ta có một biến ngẫu nhiên X có 3 giá trị có thể là x_1, x_2, x_3 với các xác suất tương ứng bằng 1/2, 1/4, 1/4. Cách mã hiệu quả nhất của 3 biến cố này là mã hoá x_1 là 0, mã của x_2 là 10 và mã của x_3 là 11. Khi đó số bit trung bình trong phép mã hoá này là:

$$1/2 \times 1 + 1/4 \times 2 + 1/4 \times 2 = 3/2.$$

Các ví dụ trên cho thấy rằng, một biến cố xảy ra với xác suất 2^{-n} có thể mã hoá được bằng một xâu bit có độ dài n . Tổng quát hơn, có thể coi rằng, một biến cố xảy ra với xác suất

p có thể mã hoá bằng một xâu bit có độ dài xấp xỉ $-\log_2 p$. Nếu cho trước phân bố xác suất tuỳ ý $p_1, p_2, ..., p_n$ của biến ngẫu nhiên X, khi đó độ đo thông tin là trọng số trung bình của các lượng $-\log_2 p_i$. Điều này dẫn tới định nghĩa hình thức hoá sau.

Định nghĩa 1.6:

Giả sử X là một biến ngẫu nhiên lấy các giá trị trên một tập hữu hạn theo phân bố xác suất p(X). Khi đó entropy của phân bố xác suất này được định nghĩa là lượng:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 P_i \tag{1.8}$$

Nếu các giá trị có thể của X là x_i, $1 \le i \le n$ thì ta có:

$$H(X) = -\sum_{i=1}^{n} p(X = x_i) \log_2 P(X = x_i) \tag{1.9}$$

Nhận xét:

Nhận thấy rằng $\log_2 p_i$ không xác định nếu $p_i = 0$. Bởi vậy đôi khi entropy được định nghĩa là tổng tương ứng trên tất cả các xác suất khác 0. Vì $\lim_{x \to 0} x \log_2 x = 0$ nên trên thực tế cũng không có trở ngại gì nếu cho $p_i = 0$ với giá trị i nào đó. Tuy nhiên ta sẽ tuân theo giả định là khi tính entropy của một phân bố xác suất p_i, tổng trên sẽ được lấy trên các chỉ số i sao cho $p_i \neq 0$. Ta cũng thấy rằng việc chọn cơ số của logarit là tuỳ ý; cơ số này không nhất thiết phải là 2. Một cơ số khác sẽ chỉ làm thay đổi giá trị của entropy đi một hằng số.

Chú ý rằng, nếu $p_i = 1/n$ với $1 \le i \le n$ thì $H(X) = \log_2 n$. Cũng dễ dàng thấy rằng $H(X) \ge 0$ và $H(X) = 0$ khi và chỉ khi $p_i = 1$ với một giá trị i nào đó và $p_j = 0$ với mọi $j \neq i$.

Xét entropy của các thành phần khác nhau của một hệ mật. Ta có thể coi khoá là một biến ngẫu nhiên *K* nhận các giá trị tuân theo phân bố xác suất p_K và bởi vậy có thể tính được $H(K)$. Tương tự ta có thể tính các entropy $H(P)$ và $H(C)$ theo các phân bố xác suất tương ứng của bản mã và bản rõ.

Ví dụ 1.2: (tiếp)

Ta có:

$$\begin{aligned} H(P) &= -1/4 \log_2 1/4 - 3/4 \log_2 3/4 \\ &= -1/4(-2) - -3/4(\log_2 3 - 2) \\ &= 2 - 3/4 \log_2 3 \\ &\approx 0,81 \end{aligned}$$

bằng các tính toán tương tự, ta có $H(K) = 1,5$ và $H(C) \approx 1,85$.

1.4.2.1. Các tính chất của Entropy

Trong phần này sẽ chứng minh một số kết quả quan trọng liên quan đến Entropy. Trước tiên ta sẽ phát biểu bất đẳng thức Jensen. Đây là một kết quả cơ bản và rất hữu ích. Bất đẳng thức Jensen có liên quan đến hàm lồi có định nghĩa như sau.

Định nghĩa 1.7:

Một hàm có giá trị thực f là lồi trên khoảng I nếu:

$$f\left(\frac{x+y}{2}\right) \geq \frac{f(x)+f(y)}{2} \tag{1.10}$$

với mọi $x, y \in I$. f là hàm lồi thực sự trên khoảng I nếu:

$$f\left(\frac{x+y}{2}\right) > \frac{f(x)+f(y)}{2} \tag{1.11}$$

với mọi $x, y \in I$, $x \neq y$.

Sau đây ta sẽ phát biểu mà không chứng minh bất đẳng thức Jensen.

Định lý 1.4: (Bất đẳng thức Jensen).

Giả sử h là một hàm lồi thực sự và liên tục trên khoảng l,

$$\sum_{i=1}^{n} a_i = 1$$

và $a_i > 0; 1 \leq i \leq n$ Khi đó:

$$\sum_{i=1}^{n} a_i f(x_i) \leq f\left(\sum_{i=1}^{n} a_i x_i\right) \tag{1.12}$$

trong đó $x_i \in I; 1 \leq i \leq n$. Ngoài ra dấu "=" chỉ xảy ra khi và chỉ khi $x_1 = x_2 = ... = x_n$

Bây giờ ta sẽ đưa ra một số kết quả về Entropy. Trong định lý sau sẽ sử dụng khẳng định: hàm $\log_2 x$ là một hàm lồi thực sự trong khoảng (0, ∞) (Điều này dễ dàng thấy được từ những tính toán sơ cấp vì đạo hàm cấp 2 của hàm logarith là âm trên khoảng (0, ∞)).

Định lý 1.5:

Giả sử X là biến ngẫu nhiên có phân bố xác suất $p_1, p_2, ..., p_n$ trong đó $p_i > 0;\ 1 \leq i \leq n$. Khi đó $H(X) < \log_2 n$. Dấu "=" xảy ra khi và chỉ khi $p_i = 1/n$, $1 \leq i \leq n$

Chứng minh:

Áp dụng bất đẳng thức Jensen, ta có:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i = \sum_{i=1}^{n} p_i \log_2 (1/p_i)$$
$$\leq \log_2 \sum_{i=1}^{n} (p_i \times 1/p_i)$$
$$= \log_2 n$$

Ngoài ra, dấu "=" chỉ xảy ra khi và chỉ khi $p_i = 1/n$, $1 \leq i \leq n$.

Định lý 1.6:

$$H(X,Y) \leq H(X) + H(Y) \tag{1.13}$$

Đẳng thức (dấu "=") xảy ra khi và chỉ khi X và Y là các biến cố độc lập

Chứng minh.

Giả sử X nhận các giá trị $x_i, 1 \leq i \leq m$; Y nhận các giá trị $y_j, 1 \leq j \leq n$. Kí hiệu: $p_i = p(X = x_i), 1 \leq i \leq m$ và $q_i = p(Y = y_j), 1 \leq j \leq n$. Kí hiệu $r_{ij} = p(X = x_i, Y = y_j)$, $1 \leq i \leq m$, $1 \leq j \leq n$. (Đây là phân bố xác suất hợp).

Nhận thấy rằng

$$p_i = \sum_{j=1}^{n} r_{ij} \; ; \; (1 \leq i \leq m)$$

và

$$q_j = \sum_{i=1}^{m} r_{ij} \; ; \; (1 \leq j \leq n)$$

Ta có

$$H(X) + H(Y) = -(\sum_{i=1}^{m} p_i \log_2 p_i + \sum_{j=1}^{n} q_j \log_2 q_j)$$
$$= -(\sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} \log_2 p_i + \sum_{j=1}^{n} \sum_{i=1}^{m} r_{ij} \log_2 q_j)$$
$$= -\sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} \log_2 p_i q_j$$

Mặt khác:

$$H(X,Y) = -\sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} \log_2 r_{ij}$$

Kết hợp lại ta thu được kết quả sau:

$$H(X,Y)-H(X)-H(Y)=\sum_{i=1}^{m}\sum_{j=1}^{n}r_{ij}\log_2(1/r_{ij})+\sum_{i=1}^{m}\sum_{j=1}^{n}r_{ij}\log_2 p_i q_j$$
$$\le \log_2\sum_{i=1}^{m}\sum_{j=1}^{n}p_i q_j$$
$$=\log_2 1$$
$$=0$$

(Ở đây đã áp dụng bất đẳng thức Jensen khi biết rằng các r_{ij} tạo nên một phân bố xác suất).

Khi đẳng thức xảy ra, có thể thấy rằng phải có một hằng số c sao cho $p_{ij}/r_{ij}=c$ với mọi i,j . Sử dụng đẳng thức sau:

$$=\sum_{i=1}^{m}\sum_{j=1}^{n}r_{ij}\log_2(p_i q_j/r_{ij})$$
$$\sum_{j=1}^{n}\sum_{i=1}^{m}r_{ij}=\sum_{j=1}^{n}\sum_{i=1}^{m}p_i q_j=1$$

Điều này dẫn đến $c=1$. Bởi vậy đẳng thức (dấu "=") sẽ xảy ra khi và chỉ khi $r_{ij}=p_j q_j$, nghĩa là:

$$p\left(X=x_j,Y=y_j\right)=p\left(X=x_j\right)p\left(Y=y_j\right)$$

với $1\le i\le m, 1\le j\le n$. Điều này có nghĩa là *X* và *Y* độc lập.

Tiếp theo ta sẽ đưa ra khái niệm Entropy có điều kiện

Định nghĩa 1.8:

Giả sử X và Y là hai biến ngẫu nhiên. Khi đó với giá trị xác định bất kỳ y của Y, ta có một phân bố xác suất có điều kiện $p(X|y)$. *Rõ ràng là:*

$$H(X|y)=-\sum_{x}p(x|y)\log_2 p(x|y) \tag{1.14}$$

Ta định nghĩa Entropy có điều kiện H(X|Y) là trung bình có trọng số (ứng với các xác suất $p(y)$*) của Entropy H(X|y) trên mọi giá trị có thể y. H(X|y) được tính bằng:*

$$H(X|Y)=-\sum_{y}\sum_{x}p(y)p(x|y)\log_2 p(x|y) \tag{1.15}$$

Entropy có điều kiện đo lượng thông tin trung bình về X do Y mang lại.

Sau đây là hai kết quả trực tiếp (Bạn đọc có thể tự chứng minh)

Định lý 1.7:

$$H\left(X,Y\right)=H\left(Y\right)+H\left(X|Y\right) \tag{1.16}$$

Hệ quả 1.2.

$$H(X|Y) \le H(X)$$

Dấu bằng chỉ xảy ra khi và chỉ khi X và Y độc lập.

1.4.2.2. Các khoá giả và khoảng duy nhất

Trong phần này chúng ta sẽ áp dụng các kết quả về Entropy ở trên cho các hệ mật. Trước tiên sẽ chỉ ra một quan hệ cơ bản giữa các Entropy của các thành phần trong hệ mật. Entropy có điều kiện $H(K|C)$ được gọi là độ bất định về khoá. Nó cho ta biết về lượng thông tin về khoá thu được từ bản mã.

Định lý 1.8:

Giả sử $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ *là một hệ mật. Khi đó:*

$$H(K|C) = H(K) + H(P) - H(C) \tag{1.17}$$

Chứng minh:

Trước tiên ta thấy rằng $H(K,P,C) = H(C|K,P) + H(K,P)$. Do $y = e_K(x)$ nên khoá và bản rõ sẽ xác định bản mã duy nhất. Điều này có nghĩa là $H(C|K,P) = 0$. Bởi vậy $H(K,P,C) = H(K,P)$. Nhưng K và P độc lập nên $H(K,P) = H(K) + H(P)$. Vì thế:

$$H(K,P,C) + H(K,P) = H(K) + H(P)$$

Tương tự vì khoá và bản mã xác định duy nhất bản rõ (tức $x = d_K(y)$) nên ta có $H(K,P,C) = 0$ và bởi vậy $H(K,P,C) = H(K,P)$. Bây giờ ta sẽ tính như sau:

$$\begin{aligned} H(K|C) &= H(K,C) - H(C) \\ &= H(K,P,C) - H(C) \\ &= H(K) + H(P) - H(C) \end{aligned}$$

Đây là nội dung của định lý.

Ta sẽ quay lại ví dụ 2.1 để minh hoạ kết quả này.

Ví dụ 1.1 (tiếp)

Ta đã tính được $H(P) \approx 0,81$, $H(K) = 1,5$ và $H(C) \approx 1,85$. Theo định lý 1.8 ta có $H(K|,C) \approx 1,5 + 0,81 - 0,85 \approx 0,46$. Có thể kiểm tra lại kết quả này bằng cách áp dụng định nghĩa về Entropy có điều kiện như sau. Trước tiên cần phải tính các xác suất xuất $p(K_j|j)$, $1 \le i \le 3$; $1 \le j \le 4$. Để thực hiện điều này có thể áp dụng định lý Bayes và nhận được kết quả như sau:

$$p(K_1|1)=1 \qquad p(K_2|1)=0 \qquad p(K_3|1)=0$$

$$p(K_1|2)=6/7 \qquad p(K_2|2)=6/7 \qquad p(K_3|2)=0$$

$$p(K_1|3)=0 \qquad p(K_2|3)=3/4 \qquad p(K_3|3)=1/4$$

$$p(K_1|4)=0 \qquad p(K_2|4)=0 \qquad p(K_3|4)=1$$

Bây giờ ta tính:

$$H(K|C) = 1/8\times 0+7/16\times 0,59+1/4\times 0,81+3/16\times 0 = 0,46$$

Giá trị này bằng giá trị được tính theo định lý 2.10.

Giả sử $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ là hệ mật đang được sử dụng. Một xâu của bản rõ $x_1,x_2,..,x_n$ sẽ được mã hoá bằng một khoá để tạo ra bản mã $y_1,y_2,...,y_n$. Nhớ lại rằng, mục đích cơ bản của thám mã là phải xác định được khoá. Ta xem xét các phương pháp tấn công chỉ với bản mã và coi Oscar có khả năng tính toán vô hạn. Ta cũng giả sử Oscar biết bản rõ là một văn bản theo ngôn ngữ tự nhiên (chẳng hạn văn bản tiếng Anh). Nói chung Oscar có khả năng rút ra một số khoá nhất định (các khoá có thể hay các khoá chấp nhận được) nhưng trong đó chỉ có một khoá đúng, các khoá có thể còn lại (các khoá không đúng) được gọi là các khoá giả.

Ví dụ, giả sử Oscar thu được một xâu bản mã WNAJW mã bằng phương pháp mã dịch vòng. Dễ dàng thấy rằng, chỉ có hai xâu bản rõ có ý nghĩa là *river* và *arena* tương ứng với các khoá F(= 5) và W(= 22). Trong hai khoá này chỉ có một khoá đúng, khoá còn lại là khoá giả. (Trên thực tế, việc tìm một bản mã của MDV có độ dài 5 và 2 bản giải mã có nghĩa không phải quá khó khăn, bạn đọc có thể tìm ra nhiều ví dụ khác). Mục đích của ta là phải tìm ra giới hạn cho số trung bình các khoá giả. Trước tiên, phải xác định giá trị này theo Entropy (cho một kí tự) của một ngôn ngữ tự nhiên L (kí hiệu là H_L). H_L là lượng thông tin trung bình trên một kí tự trong một xâu có nghĩa của bản rõ. (Chú ý rằng, một xâu ngẫu nhiên các kí tự của bảng chữ cái sẽ có Entropy trên một kí tự bằng $\log_2 26 \approx 4,76$). Ta có thể lấy H(P) là xấp xỉ bậc nhất cho H_L. Trong trường hợp L là Anh ngữ, ta tính được $H(P) \approx 4,19$.

Dĩ nhiên các kí tự liên tiếp trong một ngôn ngữ không độc lập với nhau và sự tương quan giữa các kí tự liên tiếp sẽ làm giảm Entropy. Ví dụ, trong Anh ngữ, chữ Q luôn kéo theo sau là chữ U. Để làm xấp xỉ bậc hai, tính Entropy của phân bố xác suất của tất cả các bộ đôi rồi chia cho 2. Một cách tổng quát, ta định nghĩa P^n là biến ngẫu nhiên có phân bố xác suất của tất cả các bộ n của bản rõ. Ta sẽ sử dụng tất cả các định nghĩa sau:

Định nghĩa 1.9:

Giả sử L là một ngôn ngữ tự nhiên. Entropy của L được xác định là lượng sau:

$$H_L = \lim_{n\to\infty} \frac{H(P^n)}{n} \tag{1.18}$$

Độ dư của L là: $$R_L = 1-\left(H_L/\log_2|\mathcal{P}|\right) \tag{1.19}$$

Nhận xét: H_L đo Entropy trên mỗi kí tự của ngôn ngữ L. Một ngôn ngữ ngẫu nhiên sẽ có Entropy là $\log_2 |\mathcal{P}|$. Bởi vậy đại lượng R_L đo phần "kí tự vượt trội" là phần dư.

Trong trường hợp Anh ngữ, dựa trên bảng chứa một số lớn các bộ đôi và các tần số, ta có thể tính được $H(P^2)$. Ước lượng theo cách này, ta tính được $H(P^2) \approx 3,90$. Cứ tiếp tục như vậy bằng cách lập bảng các bộ ba v.v... ta thu được ước lượng cho H_L. Trên thực tế, bằng nhiều thực nghiệm khác nhau, ta có thể đi tới kết quả sau $1,0 \le H_L \le 1,5$. Tức là lượng thông tin trung bình trong tiếng Anh vào khoảng 1 bit tới 1,5 bit trên mỗi kí tự.

Giả sử lấy 1,25 là giá trị ước lượng của giá trị của H_L. Khi đó độ dư vào khoảng 0,75. Tức là tiếng Anh có độ dư vào khoảng 75%! (Điều này không có nghĩa loại bỏ tuỳ ý 3 trên 4 kí tự của một văn bản tiếng Anh mà vẫn có khả năng đọc được nó. Nó chỉ có nghĩa là tìm được một phép mã Huffman cho các bộ n với n đủ lớn, phép mã này sẽ nén văn bản tiếng Anh xuống còn 1/4 độ dài của bản gốc).

Với các phân bố xác suất $\mathcal{C}^n$ đã cho trên $\mathcal{K}$ và $\mathcal{P}^n$. Có thể xác định phân bố xác suất trên là tập các bộ n của bản mã. (Ta đã làm điều này trong trường hợp $n = 1$). Ta đã xác định P^n là biến ngẫu nhiên biểu diễn bộ n của bản rõ. Tương tự C^n là biến ngẫu nhiên biểu thị bộ n của bản mã.

Với $y \in C^n$, định nghĩa: $K(y) = \left\{K \in \mathcal{K}; \exists x \in \mathcal{P}^n, p_{\mathcal{P}^n(x)} > 0, e_K(x) = y\right\}$ nghĩa là $K(y)$ là tập các khoá K sao cho y là bản mã của một xâu bản rõ độ dài n có nghĩa, tức là tập các khoá "có thể" với y là bản mã đã cho. Nếu y là dãy quan sát được của bản mã thì số khoá giả sẽ là $|K(y)0| - 1$ vì chỉ có một khoá là khoá đúng trong số các khoá có thể. Số trung bình các khoá giả (trên tất cả các xâu bản mã có thể độ dài n) được kí hiệu là $\overline{s_n}$ và nó được tính như sau:

$$\begin{aligned}\overline{s_n} &= \sum_{y \in C^n} p(y)\left(|K(y)| - 1\right) \\ &= \sum_{y \in C^n} p(y)|K(y)| - \sum_{y \in C^n} p(y) \\ &= \sum_{y \in C^n} p(y)|K(y)| - 1\end{aligned}$$

Từ định lý 1.8 ta có:

$$H\left(K|C^n\right) = H(K) + H\left(P^n\right) - H\left(C^n\right)$$

Có thể dùng ước lượng sau:

$$H\left(P^n\right) \approx nH_L = n(1 - R_L)\log_2 |\mathcal{P}|$$

với điều kiện n đủ lớn. Hiển nhiên là:

$$H\left(C^n\right) \le n \log_2 |C|$$

Khi đó nếu $|\mathcal{P}| = |C|$ thì:

$$H(K|C^n) \geq H(K) - nR_L \log_2 |\mathcal{P}| \tag{1.20}$$

Tiếp theo xét quan hệ của lượng $H(K|C^n)$ với số khoá giả s_n. Ta có:

$$\begin{aligned} H(K|C^n) &= \sum_{y \in C^n} p(y)(|K|y) \\ &\leq \sum_{y \in C^n} p(y) \log_2 |K(y)| \\ &\leq \sum_{y \in C^n} p(y) |K(y)| \\ &= \log_2 (\overline{s_n} + 1) \end{aligned}$$

Ở đây ta áp dụng bất đẳng thức Jensen (định lý 1.5) với $f(x) = \log_2(x)$. Bởi vậy ta có bất đẳng thức sau:

$$H(K|C^n) \leq \log_2(\overline{s_n} + 1) \tag{1.21}$$

Kết hợp hai bất đẳng thức (1.20) và (1.21), ta có:

$$\log_2(\overline{s_n} + 1) \geq H(K) - nR_L \log_2 |P|$$

Trong trường hợp các khoá được chọn đồng xác suất (Khi đó $H(K)$ có giá trị lớn nhất) ta có kết quả sau.

Định lý 1.9:

Giả sử $(\mathcal{P}, C, K, \mathcal{E}, \mathcal{D})$ là một hệ mật trong đó $|C| = |\mathcal{P}|$ và các khoá được chọn đồng xác suất. Giả sử R_L là độ dư của ngôn ngữ gốc. Khi đó với một xâu bản mã độ dài n cho trước (n là số đủ lớn), số trung bình các khoá giả s_n thoả mãn bất đẳng thức như sau:

$$\overline{s_n} \geq \{|K| / (|P| nR_L)\} - 1$$

Lượng $|K| / (|P| nR_L) - 1$ tiến tới 0 theo hàm mũ khi n tăng. Ước lượng này có thể không chính xác với các giá trị n nhỏ. Đó là do $H(P^n)/n$ không phải là một ước lượng tốt cho H_L nếu n nhỏ.

Ta đưa ra đây một khái niệm nữa

Định nghĩa 1.10:

Khoảng duy nhất của một hệ mật được định nghĩa là giá trị của n mà ứng với giá trị này, số khoá giả trung bình bằng 0 (kí hiệu giá trị này là n_0). Điều đó có nghĩa là n_0 là độ dài trung bình cần thiết của bản mã để thám mã có thể tính toán khoá một cách duy nhất với thời gian đủ lớn.

Nếu đặt $s_n = 0$ trong định lý 1.11 và giải theo n ta sẽ nhận được ước lượng cho khoảng duy nhất:

$$n_0 \approx \log_2 |\mathcal{K}| / R_L \log_2 |\mathcal{P}|$$

Ví dụ với mã thay thế, ta có $|\mathcal{P}| = 26$ và $|\mathcal{K}| = 26!$. Nếu lấy $R_L = 0,75$ thì ta nhận được ước lượng cho khoảng duy nhất bằng:

$$n_0 \approx 88,4/(0,75 \times 4,7) \approx 25$$

Điều đó có nghĩa là thông thường nếu mã thám có được xâu bản mã với độ dài tối thiểu là 25, anh ta có thể nhận được bản giải mã duy nhất.

BÀI TẬP CHƯƠNG 1.

Bài 1.1: Cho n là một số nguyên dương. Một hình vuông lớn latin cấp $n(L)$ là một bảng $n \times n$ các số nguyên $1,...,n$ sao cho mỗi một số trong n số nguyên này chỉ xuất hiện đúng một lần ở hàng và mỗi cột của *L*. Ví dụ hình vuông Latin cấp 3 có dạng:

1	2	3
3	1	2
2	3	1

Với một hình vuông Latin *L* bất kỳ cấp n , ta có thể xác định một hệ mã tương ứng. Giả sử $\mathcal{P} = C = \mathcal{K} = \{1,...,n\}$. Với $1 \le i \le n$, quy tắc mã hóa e_1 được xác định là $e_1(j) = L(i,j)$ (Do đó mỗi hàng của L sẽ cho một quy tắc mã hóa).

Chứng minh rằng hệ mật hình vuông Latin này có độ mật hoàn thiện.

Bài 1.2: Hãy chứng tỏ rằng mã Affine có độ mật hoàn thiện

Bài 1.3: Giả sử một hệ mật đạt được độ hoàn thiện với phân bố xác suất p_0 nào đó của bản rõ. Hãy chứng tỏ rằng độ mật hoàn thiện vẫn còn giữ được đối với một phân bố xác suất bất kỳ của bản rõ.

Bài 1.4: Hãy chứng tỏ rằng nếu một hệ mật có độ hoàn thiện và $|\mathcal{K}| = |C| = |\mathcal{P}|$ thì mọi bản mã là đồng xác suất.

Bài 1.5: Hãy chứng tỏ rằng $H(X,Y) = H(Y) + H(X|Y)$. Sau đó hãy chứng minh bổ đề là $H(X|Y) \le H(X)$, đẳng thức chỉ xảy ra khi và chỉ khi *X* và *Y* độc lập.

Bài 1.5: Chứng minh rằng một hệ mật có độ mật hoàn thiện khi và chỉ khi $H(P|C) = H(P)$.

Bài 1.6: Chứng minh rằng trong một hệ mật $H\left(K\left|C\right.\right) \geq H\left(PC\right)$ (về mặt trực giác kết quả này nói rằng với bản mã cho trước độ bất định của thám mã về khóa ít nhất cũng lớn bằng độ bất định khi thám mã rõ).

Bài 1.7: Xét một hệ mật trong đó $\mathcal{P} = \{a,b,c\}$, $\kappa = \{K_1, K_2, K_3\}$ và $C = \{1,2,3,4\}$. Giả sử ma trận mã hóa như sau:

	a	b	c
K_1	1	2	3
K_2	2	3	4
K_3	3	4	1

Giả sử các khóa được chọn đồng xác suất và phân bố xác suất của bản rõ là $p_{\mathcal{P}}(a) = 1/2$, $p_{\mathcal{P}}(b) = 1/3$, $p_{\mathcal{P}}(c) = 1/6$. Hãy tính $H(P), H(C), H(K)$, $H\left(K\left|C\right.\right)$ và $H\left(P\left|C\right.\right)$.

LỜI NÓI ĐẦU

Trong sự phát triển của xã hội loài người, kể từ khi có sự trao đổi thông tin, an toàn thông tin trở thành một nhu cầu gắn liền với nó như hình với bóng. Từ thủa sơ khai, an toàn thông tin được hiểu đơn giản là giữ được bí mật và điều này được xem như một nghệ thuật chứ chưa phải là một ngành khoa học. Với sự phát triển của khoa học kỹ thuật và công nghệ, cùng với các nhu cầu đặc biệt có liên quan tới an toàn thông tin, ngày nay các kỹ thuật chính trong an toàn thông tin bao gồm:

- Kỹ thuật mật mã (Cryptography)
- Kỹ thuật ngụy trang (mã ẩn) (Steganography)
- Kỹ thuật tạo bóng mờ, thủy vân số (Watermarking)

Kỹ thuật mật mã nhằm đảm bảo ba dịch vụ an toàn cơ bản:

- Bí mật (Confidential)
- Xác thực (Authentication)
- Đảm bảo tính toàn vẹn (Intergrity)

Có thể thấy rằng mật mã học là một lĩnh vực khoa học rộng lớn có liên quan rất nhiều ngành toán học như: Đại số tuyến tính, Lý thuyết thông tin, Lý thuyết độ phức tạp tính toán...

Bởi vậy việc trình bày đầy đủ mọi khía cạnh của mật mã học trong khuôn khổ một giáo trình là một điều khó có thể làm được. Chính vì lý do đó, trong giáo trình này chúng tôi chỉ dừng ở mức mô tả ngắn gọn các thuật toán mật mã chủ yếu. Các thuật toán này hoặc đang được sử dụng trong các chương trình ứng dụng hiện nay hoặc không còn được dùng nữa, nhưng vẫn được xem như là một ví dụ hay, cho ta hình dung rõ hơn bức tranh tổng thể về sự phát triển của mật mã học cả trên phương diện lý thuyết và ứng dụng. Còn một nội dung rất lý thú chưa được nêu trong giáo trình này là vấn đề thám mã. Bạn đọc quan tâm có thể tham khảo thêm trong các tài liệu [1], [2], [3].

Nội dung giáo trình bao gồm sáu chương:

Chương I - Nhập môn mật mã học: Trình bày những khái niệm và sơ lược về mật mã học, độ phức tạp tính toán, và cơ sở lý thuyết thông tin trong các hệ mật.

Chương II - Mật mã khóa bí mật: Trình bày các phương pháp xử lý thông tin của các hệ mật khóa bí mật (hệ mật khóa đối xứng) bao gồm các thuật toán hoán vị, thay thế và chuẩn mã dữ liệu của Mỹ (DES) và AES.

Chương III - Mật mã khóa công khai: Trình bày một số bài toán một chiều và các thuật toán mật mã khóa công khai (hay mật mã khoá bất đối xứng) liên quan bao

gồm: hệ mật RSA, Merkle-Hellman, Rabin, McEliece, hệ mật trên đường cong elliptic...

Chương IV - Hàm băm, xác thực và chữ ký số: Khái quát về hàm băm, một số vấn đề về xác thực, đảm bảo tính toàn vẹn và chữ ký số.

Chương V - Các thủ tục và các chú ý trong thực tế khi sử dụng mã hóa

Chương VI: Các chuẩn và áp dụng

Sau mỗi chương đều có các câu hỏi và bài tập nhằm giúp cho bạn đọc nắm vững hơn các vấn đề đã được trình bày

Phần phụ lục cung cấp chương trình nguồn của DES.

Do thời gian và trình độ còn hạn chế, việc lựa chọn và trình bày các thuật toán này không thể tránh khỏi những khiếm khuyết nhất định. Rất mong bạn đọc đóng góp ý kiến về mặt cấu trúc, các nội dung được trình bày và các sai sót cụ thể.

Các đóng góp ý kiến xin gửi về

KHOA KỸ THUẬT ĐIỆN TỬ 1 - HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KM 10. ĐƯỜNG NGUYỄN TRÃI - HÀ ĐÔNG

Email: KhoaDT1@hn.vnn.vn

Hoặc: nguyenbinh1999@yahoo.com

Xin chân thành cảm ơn!

NGƯỜI BIÊN SOẠN

2.8.2. Mô tả DES 53
2.8.3. Một số ý kiến thảo luận về DES 64
2.8.4. DES trong thực tế 65
2.8.5. Chuẩn mã dữ liệu tiên tiến (AES) 68
2.9. ƯU VÀ NHƯỢC ĐIỂM CỦA MẬT MÃ KHÓA BÍ MẬT 72
2.9.1. Ưu điểm 72
2.9.2. Nhược điểm 72
BÀI TẬP CHƯƠNG 2 73
CHƯƠNG 3. MẬT MÃ KHOÁ CÔNG KHAI 78
3.1. SỐ HỌC MODULO 78
3.1.1. Số nguyên 78
3.1.2. Các thuật toán trong 80
3.1.3. Các số nguyên modulo n 82
3.1.4. Các thuật toán trong 88
3.1.5. Các ký hiệu Legendre và Jacobi 89
3.1.6. Các số nguyên Blum 94
3.1.7. Ví dụ (Số nguyên Blum) 94
3.2. GIỚI THIỆU VỀ MẬT MÃ KHOÁ CÔNG KHAI 95
3.3. SƠ ĐỒ CHỨC NĂNG CỦA HỆ MẬT KHÓA CÔNG KHAI 96
3.4. BÀI TOÁN LOGARIT RỜI RẠC VÀ CÁC HỆ MẬT LIÊN QUAN 97
3.4.1. Bài toán logarit rời rạc 97
3.4.2. Một số hệ mật xây dựng trên bài toán logarit rời rạc 100
3.5. BÀI TOÁN PHÂN TÍCH THỪA SỐ VÀ HỆ MẬT RSA 105
3.5.1. Bài toán phân tích thừa số 105
3.5.2. Hệ mật RSA (Rivest – Shamir – Adleman) 105
3.5.3. Vấn đề điểm bất động trong RSA 109
3.5.4. Hệ mật Rabin 110
3.6. BÀI TOÁN XẾP BA LÔ VÀ HỆ MẬT MERKLE – HELLMAN 112
3.6.1. Bài toán xếp ba lô 112
3.6.2. Hệ mật Merkle - Hellman 113
3.6.3. Hệ mật Chor-Rivest (CR) 116
3.7. BÀI TOÁN MÃ SỬA SAI VÀ HỆ MẬT McELIECE 120

3.7.1. Bài toán mã sửa sai 120
3.7.2. Hệ mật McEliece 122
3.8. ĐƯỜNG CONG ELLIPTIC VÀ CÁC HỆ MẬT LIÊN QUAN 125
3.8.1. Các đường cong Elliptic. 125
3.8.2. Các đường cong Elliptic trên trường Galois 126
3.8.3. Các phép toán cộng và nhân trên các nhóm E 127
3.8.4. Các hệ mật trên đường cong elliptic 130
3.8.5. Độ an toàn của hệ mật trên đường cong Elliptic 133
3.9. ƯU VÀ NHƯỢC ĐIỂM CỦA MẬT MÃ KHÓA CÔNG KHAI 133
3.9.1. Ưu điểm 133
3.9.2. Nhược điểm 134
3.10. XÂY DỰNG CÁC CHƯƠNG TRÌNH ỨNG DỤNG KIẾN TRÚC PGP 134
BÀI TẬP CHƯƠNG 3 135
CHƯƠNG 4. HÀM BĂM, XÁC THỰC VÀ CHỮ KÝ SỐ 139
4.1. CÁC HÀM BĂM VÀ TÍNH TOÀN VẸN CỦA DỮ LIỆU 139
4.1.1. Khái niệm về hàm băm 139
4.1.2. Các định nghĩa, tính chất cơ bản và phân loại hàm băm 140
4.1.3. Các hàm băm không có khóa (MDC) 141
4.1.4. Các hàm băm có khoá (MAC) 145
4.1.5. Tính toàn vẹn của dữ liệu và xác thực thông báo 146
4.2. CHỮ KÝ SỐ 147
4.2.1. Sơ đồ chữ ký số 147
4.2.2. Sơ đồ chữ ký số RSA 148
4.3. HỆ MẬT DỰA TRÊN ĐỊNH DANH 150
4.3.1. Ý tưởng cơ bản 150
4.3.2. Sơ đồ trao đổi khoá Okamoto-Tanaka 150
CHƯƠNG 5. CÁC THỦ TỤC VÀ CÁC CHÚ Ý TRONG THỰC TẾ KHI SỬ DỤNG MÃ HOÁ 152
5.1. CÁC THỦ TỤC: HÀNH VI CÓ THỨ TỰ 152
5.1.1. Định nghĩa thủ tục 152
5.1.2. Các loại thủ tục 153
5.1.3. Các thủ tục có trọng tài 153

5.1.4. Các thủ tục có phán xét 154
5.1.5. Các thủ tục tự ràng buộc 155
5.2. CÁC THỦ TỤC ĐỂ GIẢI QUYẾT CÁC VẤN ĐỀ 156
5.2.1. Phân phối khoá 156
5.2.2. Các chữ ký số 168
5.2.3. Giao kèo về khoá 174
5.2.4. Chơi bài qua thư tín 177
5.2.5. Bỏ phiếu bằng máy tính 181
5.2.6. Chuyển giao không nhớ 184
5.2.7. Ký thoả thuận 186
5.2.8. Thư tín được chứng thực 189
5.3. SỬ DỤNG MÃ HOÁ NHƯ THẾ NÀO 190
5.3.1. Mức độ bảo mật 191
5.3.2. Quản lý khoá 192
5.3.3. Các khoá bị mất (bị lộ) 192
5.3.4. Độ phức tạp mã hoá 193
5.3.5. Lan truyền sai 194
5.3.6. Kích thước bản mã 194
5.4. CẢI THIỆN ĐỘ MẬT CỦA HỆ MẬT 194
5.4.1. Ngăn ngừa và phát hiện sai 195
5.4.2. Mã hoá một chiều 198
5.5. CÁC CHẾ ĐỘ MÃ HOÁ 201
5.5.1. Chế độ xích khối mật mã (CBC) 201
5.5.2. Chế độ hồi tiếp mật mã (CFB) 201
5.5.3. Hai khoá cho hiệu quả tương đương một khoá 112 bit 203
5.6. TÓM LƯỢC VỀ CÁC THỦ TỤC VÀ CÁC ỨNG DỤNG THỰC TẾ 204
BÀI TẬP CHƯƠNG 5 205
CHƯƠNG 6. CÁC CHUẨN VÀ ÁP DỤNG 206
6.1. BẢO MẬT THƯ ĐIỆN TỬ SỬ DỤNG PRETTY GOOD PRIVACY (PGP) 206
6.1.1. Mở đầu 206
6.1.2. Ký hiệu 206
6.1.3. Mô tả hoạt động 207

6.2. GIAO DỊCH ĐIỆN TỬ AN TOÀN - SET 211
6.2.1. Mở đầu 211
6.2.2. Mô tả SET 211
6.3. ỨNG DỤNG XÁC THỰC - KERBEROS 215
6.3.1. Mở đầu 215
6.3.2. Kerberos V.4 217
BÀI TẬP CHƯƠNG 6 221
PHỤ LỤC 1: MÃ NGUỒN DES 222

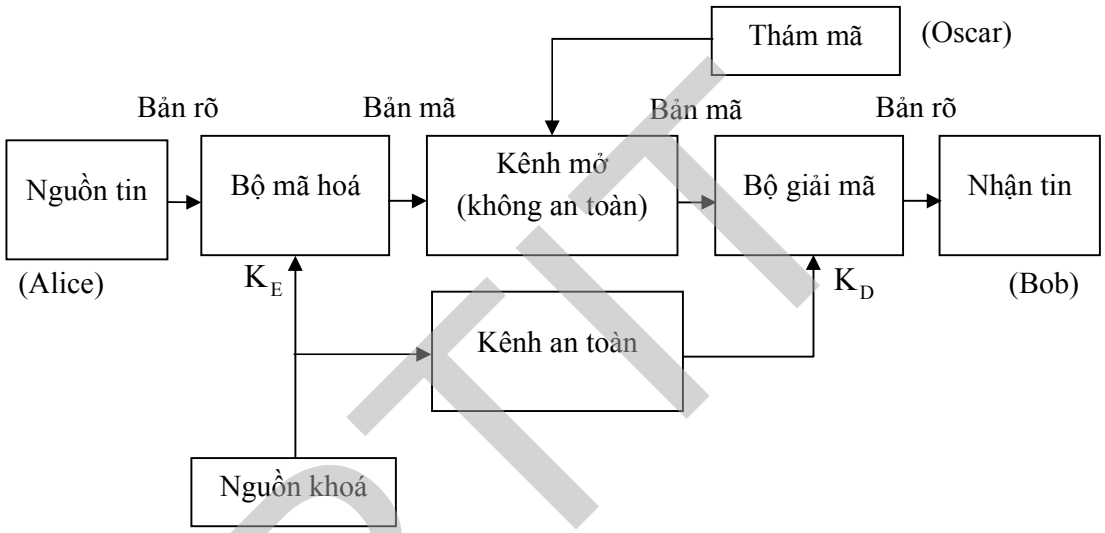
CHƯƠNG 2. MẬT MÃ KHÓA BÍ MẬT

Có ba phương pháp chính trong mật mã cổ điển (còn gọi là mật mã khoá riêng, mật mã khoá bí mật hay mật mã khóa đối xứng):

- Hoán vị
- Thay thế
- Xử lý bit (chủ yếu nằm trong các ngôn ngữ lập trình)

Ngoài ra còn có phương pháp hỗn hợp thực hiện kết hợp các phương pháp trên mà điển hình là chuẩn mã dữ liệu (DES – Data Encryption Standard) của Mỹ.

2.1. SƠ ĐỒ KHỐI MỘT HỆ TRUYỀN TIN MẬT



Hình 2.1. Sơ đồ hệ mật khóa bí mật

Định nghĩa 2.1:

Một hệ mật là một bộ 5 $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ thoả mãn các điều kiện sau:

a) $\mathcal{P}$ là một tập hữu hạn các bản rõ có thể

b) $\mathcal{C}$ là một tập hữu hạn các bản mã có thể

c) $\mathcal{K}$ là một tập hữu hạn các khoá có thể (không gian khoá)

d) Đối với mỗi $k \in \mathcal{K}$ có một quy tắc mã $e_k \in \mathcal{E}$

$$e_k : \mathcal{P} \to \mathcal{C} \tag{2.1}$$

và một quy tắc giải mã tương ứng $d_k \in \mathcal{D}$

$$d_k : \mathcal{C} \to \mathcal{P} \tag{2.2}$$

sao cho: $d_k(e_k(x)) = x$ với $\forall x \in \mathcal{P}$.

2.2. MẬT MÃ THAY THẾ

2.2.1. Mật mã dịch vòng (MDV)

Giả sử $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbf{Z}_{26}$ với $0 \le k \le 25$, ta định nghĩa:

$$e_k(x) = x + k \bmod 26$$
$$d_k(y) = y - k \bmod 26 \qquad (2.3)$$
$$(x, y \in \mathbf{Z}_{26})$$

Ta sử dụng MDV (với modulo 26) để mã hoá một văn bản tiếng Anh thông thường bằng cách thiết lập sự tương ứng giữa các ký tự và các thặng dư theo mod 26 như sau:

Ký tự	A	B	C	D	E	F	G	H	I	J	K	L	M
Mã tương ứng	0	1	2	3	4	5	6	7	8	9	10	11	12
Ký tự	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Mã tương ứng	13	14	15	16	17	18	19	20	21	22	23	24	25

Ví dụ 2.1:

Giả sử khoá cho MDV là $k = 5$ và bản rõ là: **meet me at sunset**.

Trước tiên, ta biến đổi bản rõ thành chữ in hoa và biến đổi thành dãy các số nguyên theo bảng trên (không biến đổi dấu cách (space) giữa 2 từ):

12.4.4.19.12.4.0.19.18.20.13.18.4.19

Sau đó ta cộng 5 vào mỗi giá trị ở trên và rút gọn tổng theo mod 26, ta được dãy số sau:

17.9.9.24.17.9.5.24.23.25.18.23.9.24

Cuối cùng, ta lại biến đổi dãy số nguyên trên thành các ký tự tương ứng, ta có bản mã sau:

RJJY RJ FY XZSXJY

Để giải mã cho bản mã này, trước tiên ta biến bản mã thành dãy số nguyên rồi trừ mỗi giá trị cho 5 (rút gọn theo modulo 26), và cuối cùng là lại biến đổi lại dãy số nhận được này thành các ký tự.

Nhận xét:

Khi $k = 3$, hệ mật này thường được gọi là mã Caesar đã từng được Hoàng đế Caesar sử dụng.

MDV (theo mod 26) là không an toàn vì nó có thể bị thám theo phương pháp tìm khoá vét cạn (thám mã có thể dễ dàng thử mọi khoá d_k có thể cho tới khi tìm được bản rõ có nghĩa). Trung bình có thể tìm được bản rõ đúng sau khi thử khoảng $(26/2) = 13$ quy tắc giải mã.

Từ ví dụ trên ta thấy rằng, điều kiện cần để một hệ mật an toàn là phép tìm khoá vét cạn phải không thể thực hiện được. Tuy nhiên, một không gian khoá lớn vẫn chưa đủ để đảm bảm độ mật.

2.2.2. Mã thay thế (MTT)

Cho $\mathcal{P} = C = \mathbf{Z}_{26}$. $\mathcal{K}$ chứa mọi hoán vị có thể có của 26 ký tự từ 0 đến 25. Với mỗi phép hoán vị $\pi \in \mathcal{K}$, ta định nghĩa:

$$e_\pi(x) = \pi(x)$$

và

$$d_\pi(y) = \pi^{-1}(y)$$

trong đó π^{-1} là hoán vị ngược của π

Sau đây là một ví dụ về phép hoán vị ngẫu nhiên π tạo nên một hàm mã hoá (tương tự như trên, các ký tự của bản rõ được viết bằng chữ thường, còn các ký tự của bản mã được viết bằng chữ in hoa).

Ký tự bản rõ	a	b	c	d	e	f	g	h	i	j	k	l	m
Ký tự bản mã	X	N	Y	A	H	P	O	G	Z	Q	W	B	T
Ký tự bản rõ	n	o	p	q	r	s	t	u	v	w	x	y	z
Ký tự bản mã	S	F	L	R	C	V	M	U	E	K	J	D	I

Như vậy, $e_\pi(a) = X, e_\pi(b) = N, \ldots$

Hàm giải mã là phép hoán vị ngược. Điều này được thực hiện bằng cách viết hàng thứ hai lên trước rồi sắp xếp theo thứ tự chữ cái. Ta có:

Ký tự bản mã	A	B	C	D	E	F	G	H	I	J	K	L	M
Ký tự bản rõ	d	l	r	y	v	o	h	e	z	x	w	p	t
Ký tự bản mã	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ký tự bản rõ	b	g	f	j	q	n	m	u	s	k	a	c	i

Ví dụ 2.2:

Với phép thay thế trên, từ bản rõ: **meet me at sunset**

ta thu được bản rõ sau: **THHM TH XM VUSHM**

Sử dụng phép hoán vị ngược, ta dễ dàng tìm lại được bản rõ ban đầu.

Mỗi khoá của mã thay thế là một phép hoán vị của 26 ký tự. Số các hoán vị này là $26! > 4.10^{26}$. Đây là một số rất lớn nên khó có thể tìm được khoá bằng phép tìm khoá vét cạn. Tuy nhiên, bằng phương pháp thống kê, ta có thể dễ dàng thám được các bản mã loại này.

2.2.3. Mật mã Vigenère

Trong hai hệ MDV và MTT ở trên, một khi khoá đã được chọn thì mỗi ký tự sẽ được ánh xạ vào một ký tự duy nhất. Vì vậy, các hệ trên còn được gọi là các hệ thay thế đơn biểu. Sau đây ta sẽ trình bày một hệ thay thế đa biểu được gọi là hệ mật Vigenere.

Sử dụng phép tương ứng $A \leftrightarrow 0, B \leftrightarrow 1, ..., Z \leftrightarrow 25$ mô tả ở trên, ta có thể gắn cho mỗi khoá k một chuỗi ký tự có độ dài m, được gọi là từ khoá. Mật mã Vigenère sẽ mã hoá đồng thời m ký tự: mỗi phần tử của bản rõ tương đương với m ký tự.

Ví dụ 2.3:

Giả sử $m = 6$ và từ khoá là CIPHER. Từ khoá này tương ứng với dãy số $k = (2,8,15,7,4,17)$. Giả sử bản rõ là:

meet me at sunset

Ta sẽ biến đổi các phần tử của bản rõ thành các thặng dư theo mod 26, viết chúng thành các nhóm 6 rồi cộng với từ khoá theo modulo 26 như sau:

12	4	4	19	12	4	0	19	18	20	13	18	4	19	Bản rõ
2	8	15	7	4	17	2	8	15	7	4	17	2	8	Khoá
14	12	19	0	16	21	2	1	7	1	17	9	6	1	Bản mã

Như vậy, dãy ký tự tương ứng với xâu bản mã sẽ là:

OMTA QV CB HBRJGB

Ta có thể mô tả mật mã Vigenère như sau:

Cho m là một số nguyên dương cố định nào đó.

Ta định nghĩa $P = C = K = (\mathbf{Z}_{26})^n$

Với khoá $k = (k_1, k_2, \ldots, k_m)$, ta xác định:

$e_k(x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$

và $d_k(y_1, y_2, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m)$

trong đó tất cả các phép toán được thực hiện trong $\mathbf{Z}_{26}$.

Chú ý: Để giải mã, ta có thể dùng cùng từ khoá nhưng thay cho cộng, ta trừ nó theo modulo 26.

Ta thấy rằng, số các từ khoá có thể với độ dài m trong mật mã Vigenere là 26^m. Bởi vậy, thậm chí với m khá nhỏ, phương pháp tìm kiếm vét cạn cũng yêu cầu thời gian khá lớn. Ví dụ, với $m = 6$ thì không gian khoá cũng có kích thước lớn hơn 3.10^8 khoá.

2.2.3.1. Mã Affine

MDV là một trường hợp đặc biệt của MTT chỉ gồm 26 trong số 26! các hoán vị có thể của 26 phần tử. Một trường hợp đặc biệt khác của MTT là mã Affine được mô tả dưới đây. Trong mã Affine, ta giới hạn chỉ xét các hàm mã có dạng:

$$e(x) = ax + b \bmod 26;\ a, b \in \mathbf{Z}_{26}$$

Các hàm này được gọi là các hàm Affine (chú ý rằng khi $a = 1$, ta có MDV).

Để việc giải mã có thể thực hiện được, yêu cầu cần thiết là hàm Affine phải là đơn ánh. Nói cách khác, với bất kỳ $y \in \mathbf{Z}_{26}$, ta muốn có đồng nhất thức sau:

$$ax + b \equiv y \bmod 26$$

phải có nghiệm x duy nhất. Đồng dư thức này tương đương với:

$$ax = y - b \pmod{26}$$

Vì y thay đổi trên $\mathbf{Z}_{26}$ nên $y - b$ cũng thay đổi trên $\mathbf{Z}_{26}$. Bởi vậy, ta chỉ cần nghiên cứu phương trình đồng dư:

$$ax \equiv y \bmod 26$$

Ta biết rằng, phương trình này có một nghiệm duy nhất đối với mỗi y khi và chỉ khi $\text{ƯCLN}(a, 26) = 1$ (ở đây hàm ƯCLN là ước chung lớn nhất của các biến của nó).

Trước tiên ta giả sử rằng, $\text{ƯCLN}(a, 26) = d > 1$. Khi đó, đồng dư thức $ax \equiv 0 \bmod 26$ sẽ có ít nhất hai nghiệm phân biệt trong $\mathbf{Z}_{26}$ là $x = 0$ và $x = 26/d$. Trong trường hợp này, $e(x) = ax + b \bmod 26$ không phải là một hàm đơn ánh và bởi vậy nó không thể là hàm mã hoá hợp lệ.

Ví dụ 2.4:

Do $\text{ƯCLN}(4, 26) = 2$ nên $4x + 7$ không là hàm mã hoá hợp lệ: x và $x + 13$ sẽ mã hoá thành cùng một giá trị đối với bất kỳ $x \in \mathbf{Z}_{26}$.

Ta giả thiết $\text{ƯCLN}(a, 26) = 1$. Giả sử với x_1 và x_2 nào đó thoả mãn:

$$ax_1 \equiv ax_2 \bmod 26$$

Khi đó:

$$a(x_1 - x_2) \equiv 0 \bmod 26$$

bởi vậy

$$26 \mid a(x_1 - x_2)$$

Bây giờ ta sẽ sử dụng một tính chất của phép chia sau: Nếu $\text{ƯCLN}(a, b) = 1$ và $a \mid bc$ thì $a \mid c$. Vì $26 \mid a(x_1 - x_2)$ và $\text{ƯCLN}(a, 26) = 1$ nên ta có:

$$26 \mid (x_1 - x_2)$$

tức là

$$x_1 \equiv x_2 \bmod 26$$

Tới đây ta đã chứng tỏ rằng, nếu $\text{ƯCLN}(a, 26) = 1$ thì một đồng dư thức dạng $ax \equiv y \bmod 26$ chỉ có (nhiều nhất) một nghiệm trong $\mathbf{Z}_{26}$. Do đó, nếu ta cho x thay đổi trên

$\mathbf{Z}_{26}$ thì $ax \bmod 26$ sẽ nhận được 26 giá trị khác nhau theo modulo 26 và đồng dư thức $ax \equiv y \bmod 26$ chỉ có một nghiệm y duy nhất.

Không có gì đặc biệt đối với số 26 trong khẳng định này. Bởi vậy, bằng cách tương tự, ta có thể chứng minh được kết quả sau:

Định lý 2.1:

Đồng dư thức $ax \equiv b \bmod m$ *chỉ có một nghiệm duy nhất* $x \in \mathbf{Z}_m$ *với mọi* $b \in \mathbf{Z}_m$ *khi và chỉ khi* ƯCLN$(a,m) = 1$.

Vì $26 = 2 \times 13$ nên các giá trị $a \in \mathbf{Z}_{26}$ thoả mãn ƯCLN$(a,26) = 1$ là a = 1 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 và 25. Tham số b có thể là một phần tử bất kỳ trong $\mathbf{Z}_{26}$. Như vậy , mã Affine có $12 \times 26 = 312$ khoá có thể (dĩ nhiên, con số này là quá nhỏ để bảo đảm an toàn).

Bây giờ, ta sẽ xét bài toán chung với modulo m . Ta cần một định nghĩa khác trong lý thuyết số.

Định nghĩa 2.2:

Giả sử $a \geq 1$ *và* $m \geq 2$ *là các số nguyên.* ƯCLN$(a,m) = 1$ *thì ta nói rằng* a *và* m *là nguyên tố cùng nhau. Số các số nguyên trong* $\mathbf{Z}_m$ *nguyên tố cùng nhau với* m *thường được ký hiệu là* $\varphi(m)$ *(hàm này được gọi là hàm phi-Euler).*

Một kết quả quan trọng trong lý thuyết số cho ta giá trị của $\varphi(m)$ theo các thừa số trong phép phân tích theo luỹ thừa các số nguyên tố của m . (Một số nguyên $p > 1$ là số nguyên tố nếu nó không có ước dương nào khác ngoài 1 và p). Mọi số nguyên $m > 1$ có thể phân tích được thành tích của các luỹ thừa các số nguyên tố theo cách duy nhất.

Ví dụ $60 = 2^3 \times 3 \times 5$ và $98 = 2 \times 7^2$).

Ta sẽ ghi lại công thức cho $\Phi(m)$ trong định lí sau:

Định lý 2.2:

Giả sử $m = \prod_{i=1}^{n} p_i^{e_i}$;

Trong đó các số nguyên tố p_i khác nhau và $e_i > 0; \quad 1 \leq i \leq n$. Khi đó :

$$\varphi(m) = \prod_i \left(p_i^{e_i} - p_i^{e_i - 1} \right) \tag{2.4}$$

Định lý này cho thấy rằng, số khoá trong mã Affine trên $\mathbf{Z}_{26}$ bằng $m.\varphi(m)$, trong đó $\varphi(m)$ được cho theo công thức trên. (Số các phép chọn của b là m và số các phép chọn của a là $\phi(m)$ với hàm mã hoá là $e(x) = ax + b$).

Ví dụ, khi $m = 60$, $\varphi(60) = 2 \times 2 \times 4 = 16$ và số các khoá trong mã Affine là 960.

Bây giờ, ta sẽ xét xem các phép toán giải mã trong mật mã Affine với modulo $m = 26$. Giả sử ƯCLN$(a,m) = 1$. Để giải mã cần giải phương trình đồng dư $y = ax + b \bmod 26$ theo x. Từ thảo luận trên thấy rằng, phương trình này có một nghiệm duy nhất trong $\mathbf{Z}_{26}$. Tuy nhiên, ta vẫn chưa biết một phương pháp hữu hiệu để tìm nghiệm. Điều cần thiết ở đây là có một thuật toán hữu hiệu để làm việc đó. Rất may là một số kết quả tiếp sau về số học modulo sẽ cung cấp một thuật toán giải mã hữu hiệu cần tìm.

Định nghĩa 2.3:

Giả sử $a \in \mathbf{Z}_m$. Phần tử nghịch đảo (theo phép nhân) của a là phần tử $a^{-1} \in \mathbf{Z}_m$ sao cho $aa^{-1} = a^{-1}a = 1 \bmod m$.

Bằng các lý luận tương tự như trên, có thể chứng tỏ rằng a có nghịch đảo theo modulo m khi và chỉ khi ƯCLN$(a,m) = 1$, và nếu nghịch đảo này tồn tại thì nó phải là duy nhất. Ta cũng thấy rằng, nếu $b = a^{-1}$ thì $a = b^{-1}$. Nếu p là số nguyên tố thì mọi phần tử khác không của $\mathbf{Z}_p$ đều có nghịch đảo. Một vành trong đó mọi phần tử khác 0 đều có nghịch đảo được gọi là một trường.

Trong [3] có một thuật toán hữu hiệu để tính các nghịch đảo của $\mathbf{Z}_m$ với m tuỳ ý. Tuy nhiên, trong $\mathbf{Z}_{26}$, chỉ bằng phương pháp thử và sai cũng có thể tìm được các nghịch đảo của các phần tử nguyên tố cùng nhau với 26:

$1^{-1} = 1,\ 3^{-1} = 9,\ 5^{-1} = 21,\ 7^{-1} = 15,\ 11^{-1} = 19,\ 17^{-1} = 23,\ 25^{-1} = 25$. (Có thể dễ dàng kiểm chứng lại điều này, ví dụ: $7 \times 15 = 105 \equiv 1 \bmod 26$, bởi vậy $7^{-1} = 15$).

Xét phương trình đồng dư $y \equiv ax + b \bmod 26$. Phương trình này tương đương với

$$ax \equiv y - b (\bmod 26)$$

Vì ƯCLN$(a,26) = 1$ nên a có nghịch đảo theo modulo 26. Nhân cả hai vế của đồng dư thức với a^{-1}, ta có:

$$a^{-1}(ax) \equiv a^{-1}(y - b) \bmod 26$$

Áp dụng tính kết hợp của phép nhân modulo:

$$a^{-1}(ax) \equiv (a^{-1}a)x = 1x = x$$

Kết quả là $x \equiv a^{-1}(y - b) \bmod 26$. Đây là một công thức tường minh cho x. Như vậy hàm giải mã là:

$$d(y) = a^{-1}(y - b) \bmod 26$$

Hình 2.2 cho mô tả đầy đủ về mã Affine. Sau đây là một ví dụ nhỏ.

Cho $P = C = \mathbf{Z}_{26}$ và giả sử:

$$K = \{ (a,b) \in \mathbf{Z}_{26} \times \mathbf{Z}_{26} : UCLN(a, 26) = 1 \}$$

Với $k = (a,b) \in K$, ta định nghĩa:

$$e_k(x) = ax + b \bmod 26$$

Và $$d_k(y) = a^{-1}(y - b) \bmod 26$$

$$x, y \in Z_{26}$$

Hình 2.2. Mã Affine

Ví dụ 2.5:

Giả sử $k = (7,3)$. Như đã nêu ở trên, $7^{-1} = 15$. Hàm mã hoá là:

$$e_k(x) = 7x + 3$$

Và hàm giải mã tương ứng là: $d_k(x) = 15(y - 3) = 15y - 19$

Ở đây, tất cả các phép toán đều thực hiện trên $\mathbf{Z}_{26}$. Ta sẽ kiểm tra liệu $d_k(e_k(x)) = x$ với mọi $x \in \mathbf{Z}_{26}$ không. Dùng các tính toán trên $\mathbf{Z}_{26}$ ta có:

$$d_k(e_k(x)) = d_k(7x + 3) = 15(7x + 3) - 19 = x + 45 - 19 = x$$

Để minh hoạ, ta hãy mã hoá bản rõ **"hot"**. Trước tiên, biến đổi các chữ **h, o, t** thành các thặng dư theo modulo 26. Ta được các số tương ứng là 7, 14 và 19. Bây giờ sẽ mã hoá:

$$7 \times 7 + 3 \bmod 26 = 52 \bmod 26 = 0$$
$$7 \times 14 + 3 \bmod 26 = 101 \bmod 26 = 23$$
$$7 \times 19 + 3 \bmod 26 = 136 \bmod 26 = 6$$

Bởi vậy, ba ký hiệu của bản mã là 0, 23 và 6, tương ứng với xâu ký tự **AXG**. Việc giải mã sẽ do bạn đọc thực hiện như một bài tập.

2.3. MẬT MÃ HOÁN VỊ (MHV)

Khác với MTT, ý tưởng của MHV là giữ các ký tự của bản rõ không thay đổi nhưng sẽ thay đổi vị trí của chúng bằng cách sắp xếp lại các ký tự này. Ở đây không có một phép toán đại số nào cần thực hiện khi mã hoá và giải mã.

Ví dụ 2.6:

Giả sử $m = 6$ và khoá là phép hoán vị sau:

1	2	3	4	5	6
3	5	1	6	4	2

Khi đó, phép hoán vị ngược sẽ là:

1	2	3	4	5	6
3	6	1	5	2	4

Giả sử ta có bản rõ: **a second class carriage on the train**

Trước tiên, ta nhóm bản rõ thành các nhóm 6 ký tự:

asecon | dclass | carria | geonth | etrain

Sau đó, mỗi nhóm 6 chữ cái lại được sắp xếp lại theo phép hoán vị π, ta có:

EOANCS | LSDSAC | RICARA | OTGHNE | RIENAT

Cuối cùng, ta có bản mã sau:

EOANCSLSDSACRICARAOTGHNERIENAT

Khi sử dụng phép hoán vị ngược π^{-1} trên dãy bản mã (sau khi đã nhóm lại theo các nhóm 6 ký tự), ta sẽ nhận lại được bản rõ ban đầu.

Từ ví dụ trên, ta có thể định nghĩa MHV như sau:

Cho m là một số nguyên dương xác định nào đó.

Cho $\mathcal{P} = C = (\mathbf{Z}_{26})^m$ và cho $\mathcal{K}$ là tất cả các hoán vị có thể có của:

$$\{ 1, 2, \ldots, m \}.$$

Đối với một khoá π (tức là một phép hoán vị nào đó), ta xác định:

$$e_\pi = (x_1, \ldots, x_m) = (x_{\pi(1)}, \ldots, x_{\pi(m)})$$

và $$d_\pi = (x_1, \ldots, x_m) = (y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(m)})$$

trong đó π^{-1} là phép hoán vị ngược của π

2.4. MẬT MÃ HILL

Trong phần này sẽ mô tả một hệ mật thay thế đa biểu khác được gọi là mật mã Hill. Mật mã này do Lester S.Hill đưa ra năm 1929. Giả sử m là một số nguyên dương, đặt $\mathcal{P} = C = (\mathbf{Z}_{26})^m$. 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.

Ví dụ nếu $m = 2$ ta có thể viết một phần tử của bản rõ là $x = (x_1, x_2)$ và một phần tử của bản mã là $y = (y_1, y_2)$. Ở đây, y_1 cũng như y_2 đều là một tổ hợp tuyến tính của x_1 và x_2. Chẳng hạn, có thể lấy:

$$\begin{aligned} y_1 &= 11x_1 + 3x_2 \\ y_2 &= 8x_1 + 7x_2 \end{aligned} \qquad (2.5)$$

Tất nhiên có thể viết gọn hơn theo ký hiệu ma trận như sau:

$$(y_1 \quad y_2) = (x_1 \quad x_2)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} \tag{2.6}$$

Nói chung, có thể lấy một ma trận k kích thước $m \times m$ làm khoá. Nếu một phần tử ở hàng i và cột j của k là $k_{i,j}$ thì có thể viết $k = \left(k_{i,j}\right)$, với $x = (x_1, x_2, ..., x_m) \in \mathcal{P}$ và $k \in \mathcal{K}$, ta tính $y = e_k(x) = (y_1, y_2, ..., y_m)$ như sau :

$$(y_1, y_2, ..., y_m) = (x_1, x_2, ..., x_m)\begin{pmatrix} k_{1,1} & k_{1,2} & ... & k_{1,m} \\ k_{2,1} & k_{2,2} & ... & k_{2,m} \\ ... & ... & ... & ... \\ k_{m,1} & k_{m,2} & ... & k_{m,m} \end{pmatrix} \tag{2.7}$$

Nói 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. Bạn đọc 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 $x = yk^{-1}$.

Sau đây là một số định nghĩa về những khái niệm cần thiết lấy từ đại số tuyến tính. Nếu $\mathrm{A} = \left(x_{i,j}\right)$ là một ma trận cấp $1 \times m$ và $\mathrm{B} = \left(b_{l,k}\right)$ là một ma trận cấp $m \times n$ thì tích ma trận $\mathrm{AB} = \left(c_{l,k}\right)$ được định nghĩa theo công thức :

$$c_{l,k} = \sum_{j=1}^{m} a_{i,j} b_{j,k} \tag{2.8}$$

với $1 \le i \le l$ và $1 \le k \le l$. 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 $1 \times n$.

Theo định nghĩa này, phép nhân ma trận là kết hợp (tức $(\mathrm{AB})\mathrm{C} = \mathrm{A}(\mathrm{BC})$ nhưng nói chung là không giao hoán (không phải lúc nào $\mathrm{AB} = \mathrm{BA}$, 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. Như vậy, ma trận đơn vị 2×2 là:

$$\mathrm{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.9}$$

I_m được gọi là ma trận đơn vị vì $\mathrm{AI}_m = \mathrm{A}$ với mọi ma trận cấp $1 \times m$ và $\mathrm{I}_m\mathrm{B} = \mathrm{B}$ 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 $\mathrm{AA}^{-1} = \mathrm{I}_m$. 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 hai vế của đẳng thức với k^{-1} và nhận được:

$$yk^{-1} = (xk)k^{-1} = x\mathrm{I}_m = x \tag{2.10}$$

(Chú ý: sử dụng tính chất kết hợp)

Có thể thấy rằng, ma trận mã hoá ở trên có nghịch đảo trong $\mathbf{Z}_{26}$:

$$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$$

vì

$$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = \begin{pmatrix} 11\times 7 + 8\times 23 & 11\times 18 + 8\times 11 \\ 3\times 7 + 7\times 23 & 3\times 18 + 7\times 11 \end{pmatrix}$$
$$= \begin{pmatrix} 261 & 286 \\ 182 & 131 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(Hãy nhớ rằng mọi phép toán số học đều được thực hiện theo modulo 26).

Sau đây là một ví dụ minh hoạ cho việc mã hoá và giải mã trong hệ mật mã Hill.

Ví dụ 2.7:

Giả sử khoá $k = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$

Từ các tính toán trên, ta có:

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

Giả sử cần mã hoá bản rõ "July". Ta có hai phần tử của bản rõ để mã hoá: $(9,20)$ (ứng với Ju) và $(11,24)$ (ứng với ly). Ta tính như sau:

$$(9 \quad 20)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = \begin{pmatrix} 99+60 & 72+140 \end{pmatrix} = \begin{pmatrix} 3 & 4 \end{pmatrix}$$

$$(11 \quad 24)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = \begin{pmatrix} 121+72 & 88+168 \end{pmatrix} = \begin{pmatrix} 11 & 22 \end{pmatrix}$$

Bởi vậy, bản mã của **July** là **DELW**. Để giải mã, Bob sẽ tính

$$\begin{pmatrix} 3 & 4 \end{pmatrix}k^{-1} = \begin{pmatrix} 9 & 20 \end{pmatrix} \text{ và } \begin{pmatrix} 11 & 22 \end{pmatrix}k^{-1} = \begin{pmatrix} 11 & 24 \end{pmatrix}$$

Như vậy, Bob đã nhận được bản đúng.

Cho tới lúc này, ta đã chỉ ra rằng có thể thực hiện phép giải mã nếu k có một nghịch đảo. Trên thực tế, để phép giải mã là có thể thực hiện được, điều kiện cần là k phải có nghịch

đảo. (Điều này dễ dàng rút ra từ đại số tuyến tính sơ cấp, tuy nhiên sẽ không chứng minh ở đây). Bởi vậy, ta chỉ quan tâm tới các ma trận k khả nghịch.

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 2.4:

Định thức của ma trận $A=\left(a_{i,j}\right)$ *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 hàng sơ cấp (hãy xem một giáo trình bất kỳ về đại số tuyến tính).

Hai tính chất quan trọng của định thức là:

$$\det \mathrm{I}_m = 1$$

và quy tắc nhân: $\det(\mathrm{AB}) = \det\mathrm{A}\times\det\mathrm{B}$.

Một ma trận thực 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 $\mathbf{Z}_{26}$. *Kết quả tương ứng là ma trận* k *có nghịch đảo theo modulo 26 khi và chỉ khi ƯCLN* $(\det k, 26) = 1$.

Sau đây sẽ chứng minh ngắn gọn kết quả này.

Trước tiên, giả sử rằng $\text{ƯCLN}(\det k, 26) = 1$. Khi đó $\det k$ có nghịch đảo trong $\mathbf{Z}_{26}$. Với $1\le i\le m$, $1\le j\le m$, định nghĩa k_{ij} là 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 $(-1)^{i+j}\det k_{ij}$ (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 $\mathbf{Z}_{26}$.

Nhận xét: Công thức đối với k^{-1} 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 hàng sơ cấp.

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

Định lý 2.3:

Giả sử $\mathrm{A}=(a_{ij})$ là một ma trận cấp 2×2 trên $\mathbf{Z}_{26}$ sao cho:

$\det A = (a_{1,1}a_{2,2} - a_{1,2}a_{2,1})$ có nghịch đảo. Khi đó:

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

Trở lại ví dụ đã xét ở trên. Trước hết ta có:

$$\det\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = 11\times 7 - 8\times 3 \bmod 2 = 77 - 24 \bmod 2 = 53 \bmod 2 = 1$$

Vì $1^{-1} \bmod 26 = 1$ nên ma trận nghịch đảo là:

$$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$$

Đây chính là ma trận đã có ở trên. (Chú ý: $-8 = 18; -3 = 23$)

Bây giờ ta sẽ mô tả chính xác mật mã Hill trên $\mathbf{Z}_{26}$ (Hình 2.3).

Cho m là một số nguyên dương cố định. Cho $\mathcal{P} = C = (\mathbf{Z}_{26})^m$ và cho $\mathcal{K}$ = { các ma trận khả nghịch cấp $m \times m$ trên $\mathbf{Z}_{26}$ }

Với một khoá $k \in \mathcal{K}$ ta xác định:

$$e_k(x) = xk$$

và $$d_k(y) = yk^{-1}$$

Tất cả các phép toán được thực hiện trong $\mathbf{Z}_{26}$

Hình 2.3. Mật mã Hill

2.5. HỆ MẬT XÂY DỰNG TRÊN CÁC CẤP SỐ NHÂN XYCLIC CỦA VÀNH ĐA THỨC

Trong phần này ta xét một ứng dụng của nhóm nhân xyclic trên vành đa thức $\mathbf{Z}_x[x]/x^n + 1$ với $n = 2^k$. Đây là một trường hợp đặc biệt không được xem xét tới khi xây dựng các mã khống chế sai.Tuy nhiên,trường hợp này lại có những ứng dụng khá lý thú trong mật mã [4].

2.5.1. Nhóm nhân của vành

Bổ đề 2.1:

Trong vành $\mathbf{Z}_x[x]/x^n + 1$ với $n = 2^k$, tập các đa thức có trọng số lẻ sẽ tạo nên một nhóm nhân các đa thức theo modulo $x^n + 1$.

Chứng minh: Vì $n = 2^k$ nên: $(x^n + 1) = (1 + x)^n$. Do đó, mọi đa thức $a(x)$ có trọng số lẻ đều thoả mãn điều kiện:

$$\left(a(x),(1+x)^n\right)=1 \tag{2.11}$$

Các đa thức này sẽ tạo nên một nhóm nhân G có luỹ đẳng $e(x)=1$ và có cấp bằng: $|G|=2^{n-1}$.

Bổ đề 2.2:

Mọi phần tử trong nhóm nhân G có cấp là 2^k hoặc có cấp là ước của 2^k.

Chứng minh: Đây là một trường hợp riêng của định lý ở phần 2.4.10. Ta có thể chứng minh bằng qui nạp:

$k=1$: vành này chứa nhóm nhân cấp 2 là nhóm nhân xyclic đơn vị I.

$k=i$: Giả sử $A=\{a(x),a^2(x),a^3(x),...,a^n(x)\}$ là một nhóm nhân xyclic cấp n trong vành ($n=2^i$).

$k=i+1$: Bình phương các phần tử của A ta có nhóm nhân xyclic sau:

$$A^2=\{a^2(x),a^4(x),a^6(x),...,a^{2n}(x)\}$$

Nhóm nhân xyclic này hiển nhiên là nhóm con của nhóm nhân xyclic cấp $2.2^i=2^{i+1}$ có phần tử sinh là một trong các căn bậc hai của $a(x)$.

Gọi Q là tập các thặng dư bậc hai trong G. Ta có bổ đề sau:

Bổ đề 2.3:

Số các thặng dư bậc hai trong nhóm nhân G của vành được xác định theo biểu thức sau :

$$|Q|=2^{2^{k-1}-1} \tag{2.12}$$

Chứng minh: Xét $f(x)\in Q$. Giả sử căn bậc hai của $f(x)$ là $g(x)$, tức là:

$$g^2(x)=f(x)\bmod x^n+1$$

Nếu $g(x)=\sum g_i x^i$ thì $f(x)=\sum g_i x^{2i}$.

Tức là $f(x)$ (có trọng số lẻ) chỉ gồm một số lẻ các đơn thức có mũ chẵn.

Số lượng các đa thức này bằng:

$$|Q|=C_{n/2}^1+C_{n/2}^3+...+C_{n/2}^{(n/2)-1}$$

2.5.2. Các phần tử cấp n và các nhóm nhân xyclic cấp n

Xét $a(x)\in G$. $a(x)=\sum a_i x^i$. Ta có bổ đề sau:

Bổ đề 2.4:

Đa thức $a(x)$ là phần tử cấp n khi nó có chứa một số lẻ các đơn thức có mũ lẻ có cấp n và một số chẵn các đơn thức có mũ chẵn có cấp là ước của n. Số các đa thức cấp n bằng 2^{n-2}.

Chứng minh: Vì $a(x) \in G$ nên nó có trọng số lẻ. Số lượng các đơn thức có cấp n là $(n/2)$ và số lượng các đơn thức còn lại là $(n/2)$. Như vậy, số các đa thức $a(x)$ có cấp n bằng:

$$\left(\sum_i C_{n/2}^{2i-1}\right)\left(\sum_j C_{n/2}^{2j}\right) = 2^{(n/2)-1}2^{(n/2)-1} = 2^{n-2}$$

Ví dụ 2.8:

Với trường hợp $n = 8$, có tất cả $2^6 = 64$ các phần tử cấp n.

Ta có thể sử dụng các phần tử này để xây dựng các nhóm nhân xyclic cấp n.

$$A_i = \{a_i(x), a_i^2(x), a_i^3(x), ..., a_i^n(x) = a_i^0(x) = 1\}$$

Có tất cả 2^{n-2} các nhóm nhân xyclic cấp n và nhóm nhân I cũng thuộc vào lớp các nhóm nhân này. Ta gọi nó là nhóm nhân xyclic đơn vị.

2.5.3. Hệ mật xây dựng trên các cấp số nhân xyclic

2.5.3.1. Các cấp số nhân xyclic cấp n

Nếu ta nhân các phần tử của một nhóm nhân xyclic cấp n với một phần tử bất kỳ trong nhóm nhóm nhân G của vành đa thức ta sẽ thu được một cấp số nhân xyclic có công bội là phần tử sinh của nhóm nhân và có số hạng ban đầu là đa thức đem nhân.

Bổ đề 2.5:

Số các cấp số nhân xyclic cấp n xây dựng được trong G xác định theo biểu thức sau:

$$N = 2^{2^k-1}.2^{2^k-2} \tag{2.13}$$

Ví dụ 2.9:

$n = 8$ $\quad N = 2^{8-1}.2^{8-2} = 2^{13} = 8.192$

$n = 16$ $\quad N = 2^{16-1}.2^{16-2} = 2^{29} = 536.870.912$

$n = 32$ $\quad N = 2^{32-1}.2^{32-2} = 2^{61}$

$n = 64$ $\quad N = 2^{64-1}.2^{64-2} = 2^{125}$

$n = 128$ $\quad N = 2^{128-1}.2^{128-2} = 2^{253}$

2.5.3.2. Hệ mật xây dựng trên các cấp số nhân xyclic

Mỗi cấp số nhân xyclic cấp n có thể coi là một phép biến đổi tuyến tính của vector mã ban đầu (được coi là nhóm nhân xyclic đơn vị I).

Gọi α là phần tử sinh của một nhóm nhân xyclic cấp n. Ta có bổ đề sau:

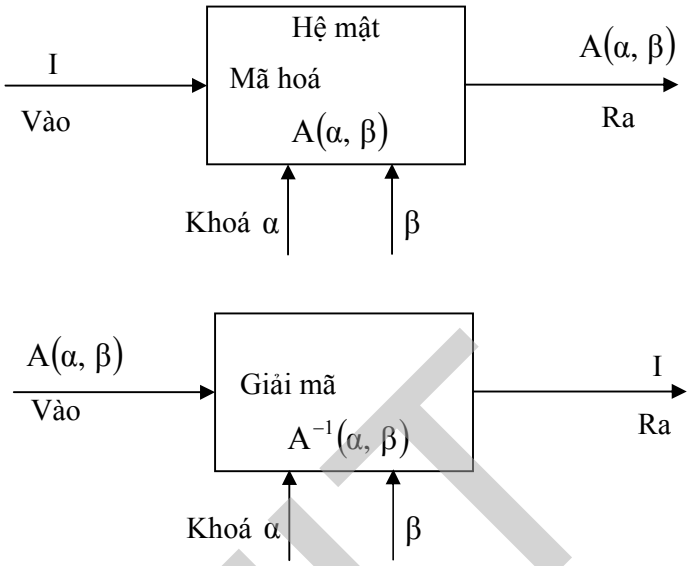
Bổ đề 2.6:

Tổng các số hạng của một cấp số nhân xyclic cấp n có công bội α và số hạng đầu β được xác định theo biểu thức sau:

$$S_n = \beta\left[\prod_{i=0}^{k-1}\left(1+\alpha^{2^i}\right)\right] \tag{2.14}$$

Hiển nhiên là $S_n \neq 0$.

Hệ mật xây dựng trên các cấp số nhân này có thể được mô tả theo sơ đồ khối sau:



Hình 2.4.

Mỗi phép biến đổi (mã hoá) A có thể được đặc trưng bởi một ma trận vuông cấp n có dạng sau:

$$A = \begin{pmatrix} \beta.\alpha \\ \beta.\alpha^2 \\ \vdots \\ \beta.\alpha^0 \end{pmatrix}$$

A là một ma trận không suy biến và bởi vậy, luôn tồn tại A^{-1} thoả mãn:

$$A.A^{-1} = I$$

Tập các phép biến đổi này là một tập kín đối với phép tính (nhân ma trận) và tạo nên một nhóm nhân có phần tử đơn vị là phép biến đổi đồng nhất (ma trận đơn vị I).

Nhóm nhân trong vành các ma trận vuông này là nhóm tuyến tính đầy đủ và được ký hiệu là $GL\left(n, GF(2)\right)$.

Thuật toán mã hoá khá đơn giản, chỉ dựa trên phép toán nhân và bình phương một đa thức $a(x) \in G$ theo modulo $(x^n + 1)$ ($a(x)$ có cấp n) với một đa thức $b(x)$ bất kỳ $\in G$.

2.5.3.3. Vấn đề giải mã

Để giải mã ta phải tìm phép biến đổi ngược A^{-1} là ma trận nghịch đảo của ma trận A. Tuy nhiên ta có thể dễ dàng thực hiện giải mã dựa trên bổ đề sau:

Bổ đề 2.7:

Ma trận A có cấp (order) hoặc là n , hoặc là ước của n . Tức là ta luôn có:

$$A^n = I$$

Hay $$\underbrace{\left(\left((A^2)^2\right)^2 ...\right)^2}_{k \text{ lần}}$$

Ở đây, A được xem là phần tử sinh của một nhóm nhân xyclic có cấp bằng n hoặc bằng ước của n .

Ví dụ 2.10: $n = 8$

$$A = \{(012),(024),(01356),(4),(456),(046),(12457),(0)\}$$

Ma trận tương ứng:

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

$$A^2 = \{(014),(2),(236),(4),(045),(6),(267),(0)\}$$

$$A^3 = \{(124),(024),(01235),(4),(056),(046),(14567),(0)\}$$

$$A^4 = I = \{(1),(2),(3),(4),(5),(6),(7),(0)\}$$

Chú ý: Ở đây ta biểu diễn các đa thức qua các số mũ của các thành phần khác không. Ví dụ: $(01235) = 1 + x + x^2 + x^3 + x^5$.

Vào	Mã hoá	Ra	Vào	Giải mã	Ra
I $\rightarrow$	A $\rightarrow$	A	A $\rightarrow$	$(A^2)^2$ =	I

Ví dụ 2.11:

Xét cấp số nhân có công bội (023) với số hạng đầu (023) (012) = (015).

$$B = \{(015),(12457),(03467),(456),(145),(01356),(02347),(012)\}$$

$$B^2 = \{(124),(136),(346),(035),(056),(257),(027),(147)\}$$

$$B^3 = \{(02567),(047),(167),(23567),(12346),(034),(235),(12367)\}$$

$$B^4 = \{(02456),(13567),(02467),(01357),(01246),(12357),(02346),(13457)\}$$

$$B^5 = \{(347),(12345),(01245),(146),(037),(01567),(01456),(025)\}$$

$$B^6 = \{(245),(123),(467),(345),(016),(567),(023),(017)\}$$

$$B^7 = \{(24567),(236),(127),(01347),(01236),(267),(356),(03457)\} = B^{-1}$$

$$B^8 = \{(1),(2),(3),(4),(5),(6),(7),(0)\}$$

$$I = \left(\left(B^2\right)^2\right)^2$$

Thuật toán giải mã chỉ là một thuật toán lặp của thuật toán mã hoá. Số lần lặp tối đa là k .

2.5.3.4. Các ma trận luân hoàn

Khi sử dụng cấp số nhân có công bội x và có số hạng đầu là một đa thức $a(x) \in G$ ta sẽ có một lớp các biến đổi đặc biệt, được đặc trưng bởi một loại ma trận đặc biệt, được gọi là ma trận luân hoàn.

Định nghĩa 2.5:

Ma trận vuông A $n \times n$ trên trường F được gọi là ma trận luân hoàn nếu nó có dạng sau:

$$A = \begin{pmatrix} a(x) \\ xa(x) \\ \vdots \\ x^{n-1}a(x) \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & \dots & a_{n-1} \\ a_{n-1} & a_0 & \dots & a_{n-2} \\ \vdots & \vdots & & \vdots \\ a_1 & a_2 & \dots & a_0 \end{pmatrix}; \qquad a \in F$$

Bổ đề 2.8:

Đại số các ma trận luân hoàn cấp n trên trường F đẳng cấu với đại số $F[x]/(x^n - 1)$ đối với phép ánh xạ các ma trận luân hoàn thành các đa thức dạng:

$$a(x) = \sum_{i=0}^{n-1} a_i x^i$$

Bổ đề 2.9:

Tổng và tích của hai ma trận luân hoàn là một ma trận luân hoàn.

Ta có: A.B = C

Trong đó: $c(x) = a(x)b(x) \bmod (x^n - 1)$

Bổ đề 2.10:

Ma trận luân hoàn A là khả nghịch khi và chỉ khi đa thức $a(x)$ là nguyên tố cùng nhau với $(x^n - 1)$. Ma trận nghịch đảo B nếu tồn tại sẽ tương ứng với $b(x)$ thoả mãn điều kiện:

$$a(x)b(x) \equiv 1 \bmod (x^{2^k} - 1)$$

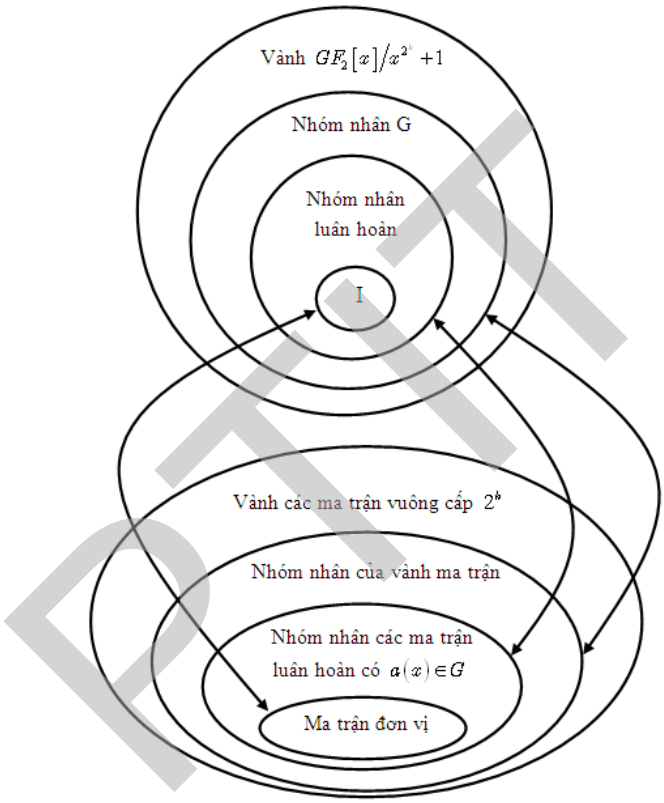
Trong trường hợp vành $GF_2[x]/(x^n+1)$ và $a(x) \in G$, ta luôn có:

$$\left(a(x),(x^{2^k}+1)\right)=\left(a(x),(x+1)^{2^k}\right)=1$$

Bổ đề 2.11:

Tập các ma trận luân hoàn A ứng với $a(x) \in G$ sẽ tạo nên một nhóm con nhân Abel trong nhóm nhân của vành các ma trận vuông. Trong nhóm này tồn tại các nhóm con là các nhóm nhân xyclic có cấp bằng n hoặc ước của n.

Mối quan hệ giữa nhóm nhân của vành đa thức và nhóm nhân của vành các ma trận vuông được mô tả trên hình sau (Hình 2.5).



Hình 2.5. Quan hệ giữa vành đa thức và vành ma trận

Bổ đề 2.12:

Cấp của ma trận luân hoàn A bằng cấp của đa thức $a(x)$ tương ứng của nó.

Khi $ord(a(x))=2$ thì ma trận luân hoàn A tương ứng là một ma trận tự nghịch đảo.

Bổ đề 2.13:

Số các ma trận luân hoàn dùng để lập mã bằng số các phần tử của nhóm nhân trong vành đa thức.

Trong trường hợp ma trận luân hoàn, thuật toán mã hoá chỉ là một phép cộng với n bước dịch vòng.

Thuật toán giải mã bao gồm một phép tính nghịch đảo của một đa thức theo modulo $(x^n + 1)$ và n bước dịch vòng tương ứng của phần tử nghịch đảo này.

Ví dụ 2.12: $a(x) = 1 + x + x^2$

$$A = \{(012),(123),(234),(345),(456),(567),(067),(017)\}$$

$$A^2 = \{(124),(135),(246),(357),(046),(157),(026),(137)\}$$

$$A^3 = \{(01356),(12467),(02357),(01346),(12457),(02356),(13467),(02457)\}$$

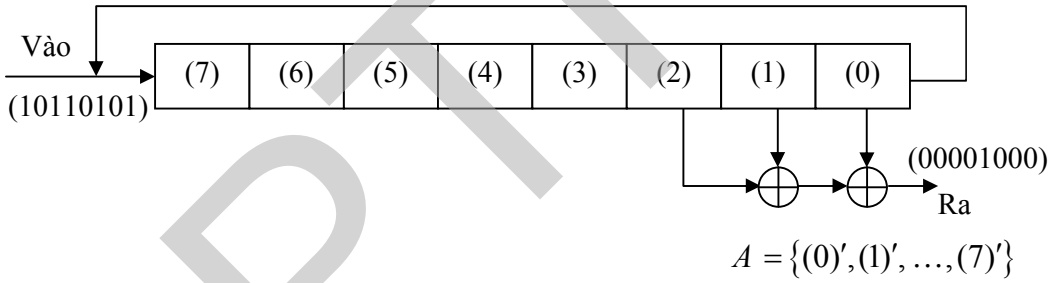
$$A^4 = \{(4),(5),(6),(7),(0),(1),(2),(3)\}$$

$$A^5 = \{(456),(567),(067),(017),(012),(123),(234),(345)\}$$

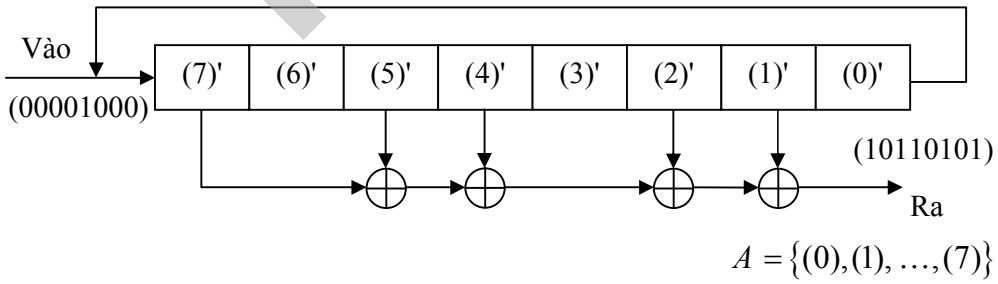
$$A^6 = \{(046),(157),(026),(137),(024),(135),(246),(357)\}$$

$$A^7 = \{(12457),(02356),(13467),(02457),(01356)$$
$$(12467),(02357),(01346)\} = A^{-1}$$

$$A^8 = \{(1),(2),(3),(4),(5),(6),(7),(0)\} = I$$



Hình 2.6. Sơ đồ thiết bị mã hoá



Hình 2.7. Sơ đồ thiết bị giải mã

$$a^{-1}(x) = x + x^2 + x^4 + x^5 + x^7$$

Ta có:

$$AA^{-1} = \begin{pmatrix} 1&1&1&0&0&0&0&0\\ 0&1&1&1&0&0&0&0\\ 0&0&1&1&1&0&0&0\\ 0&0&0&1&1&1&0&0\\ 0&0&0&0&1&1&1&0\\ 0&0&0&0&0&1&1&1\\ 1&0&0&0&0&0&1&1\\ 1&1&0&0&0&0&0&1 \end{pmatrix} \times \begin{pmatrix} 0&1&1&0&1&1&0&1\\ 1&0&1&1&0&1&1&0\\ 0&1&0&1&1&0&1&1\\ 1&0&1&0&1&1&0&1\\ 1&1&0&1&0&1&1&0\\ 0&1&1&0&1&0&1&1\\ 1&0&1&1&0&1&0&1\\ 1&1&0&1&1&0&1&0 \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix} = I$$

2.6. CÁC HỆ MẬT MÃ TÍCH

Một phát minh khác do Shannon đưa ra trong bài báo của mình năm 1949 là ý tưởng kết hợp các hệ mật bằng cách tạo tích của chúng. Ý tưởng này có tầm quan trọng to lớn trong việc thiết kế các hệ mật hiện nay (chẳng hạn, chuẩn mã dữ liệu - DES).

Để đơn giản, trong phần này chỉ hạn chế xét các hệ mật trong đó $C = \mathcal{P}$: các hệ mật loại này được gọi là tự đồng cấu. Giả sử $S_1 = (\mathcal{P}, \mathcal{P}, \mathcal{K}_1, \mathcal{E}_1, \mathcal{D}_1)$ và $S_2 = (\mathcal{P}, \mathcal{P}, \mathcal{K}_2, \mathcal{E}_2, \mathcal{D}_2)$ là hai hệ mật tự đồng cấu có cùng các không gian bản mã và rõ. Khi đó, tích của S_1 và S_2 (kí hiệu là $S_1 \times S_2$) được xác định là hệ mật sau:

$$(\mathcal{P}, \mathcal{P}, \mathcal{K}_1 \times \mathcal{K}_2, \mathcal{E}, \mathcal{D})$$

Khoá của hệ mật tích có dạng $k = (k_1, k_2)$ trong đó $k_1 \in \mathcal{K}_1$ và $k_2 \in \mathcal{K}_2$. Các quy tắc mã và giải mã của hệ mật tích được xác định như sau: Với mỗi $k = (k_1, k_2)$, ta có một quy tắc mã e_k xác định theo công thức:

$$e_{(k_1,k_2)}(x) = e_{k_2}\left(e_{k_1}(x)\right)$$

và quy tắc giải mã:

$$d_{(k_1,k_2)}(y) = d_{k1}\left(d_{k_2}(y)\right)$$

Nghĩa là, trước tiên ta mã hoá x bằng e_{k_1} rồi mã lại mã hóa bản kết quả bằng e_{k_2}. Quá trình giải mã tương tự nhưng thực hiện theo thứ tự ngược lại:

$$d_{(k_1,k_2)}\left(e_{(k_1,k_2)}(x)\right) = d_{(k_1,k_2)}\left(e_{k_2}\left(e_{k_1}(x)\right)\right) = d_{k_1}\left(d_{k_2}\left(e_{k_2}\left(e_{k_1}(x)\right)\right)\right) = d_{k_1}\left(e_{k_1}(x)\right) = x$$

Ta biết rằng, các hệ mật đều có các phân bố xác suất ứng với các không gian khoá của chúng. Bởi vậy, cần phải xác định phân bố xác suất cho không gian khoá K của hệ mật tích. Hiển nhiên ta có thể viết:

$$p_K(k_1, k_2) = p_{K_1}(k_1) \times p_{K_2}(k_2)$$

Nói một cách khác, ta chọn k_1 có phân bố $p_{K_1}(k_1)$ rồi chọn một cách độc lập k_2 có phân bố $p_{K_2}(k_2)$.

Sau đây là một ví dụ đơn giản để minh hoạ khái niệm hệ mật tích. Giả sử định nghĩa hệ mật mã nhân như trong Hình 2.8 sau.

Giả sử $P = C = \mathbf{Z}_{26}$ và giả sử:

$$k = \{\, a, \mathbf{Z}_{26} : UCLN(a, 26) = 1 \,\}$$

Với $a \in K$, ta xác định: $e_a(x) = ax \mod 26$

và $d_a(y) = a^{-1}y \mod 26$

$(x, y) \in \mathbf{Z}_{26}$

Hình 2.8. Mật mã tích

Cho M là một hệ mã nhân (với các khoá được chọn đồng xác suất) và S là MDV (với các khoá chọn đồng xác suất). Khi đó dễ dàng thấy rằng $M \times S$ chính là hệ mã Affine (cùng với các khoá được chọn đồng xác suất). Tuy nhiên, việc chứng tỏ $S \times M$ cũng là hệ mã Affine khó hơn một chút (cũng với các khóa đồng xác suất).

Ta sẽ chứng minh các khẳng định này. Một khoá dịch vòng là phần tử $k \in \mathbf{Z}_{26}$ và quy tắc mã hóa tương ứng là $e_k(x) = x + k \mod 26$. Còn khoá trong hệ mã nhân là phần từ $a \in \mathbf{Z}_{26}$ sao cho ƯCLN$(a, 26) = 1$. Quy tắc mã tương ứng là $e_a(x) = a \mod 26$. Bởi vậy, một khoá trong mã tích $M \times S$ có dạng (a, k), trong đó

$$e_{(a,k)} = ax + k \mod 26$$

Đây chính là định nghĩa về khoá trong hệ mã Affine. Hơn nữa, xác suất của một khoá trong hệ mã Affine là: $1/312 = (1/12)(1/26)$. Đó là tích của xác suất tương ứng của các khoá a và k. Bởi vậy $M \times S$ là hệ mã Affine.

Bây giờ ta sẽ xét $S \times M$. Một khoá này trong hệ mã này có dạng (k, a), trong đó:

$$e_{(k,a)}(x) = a(x + k) = ax + ak \mod 26$$

Như vậy, khoá (k, a) của mã tích $S \times M$ đồng nhất với khoá (a, ak) của hệ mã Affine. Vấn đề còn lại là phải chứng tỏ rằng mỗi khoá của mã Affine xuất hiện với cùng xác suất 1/312 như trong mã tích $S \times M$. Nhận thấy rằng $ak = k_1$ khi và chỉ khi $k = a^{-1}k_1$, (hãy nhớ lại rằng ƯCLN$(a, 26) = 1$, bởi vậy a có phần tử nghịch đảo). Nói cách khác, khoá (a, k_1) của hệ mã Affine tương đương với khoá $(a^{-1}k_1, a)$ của mã tích $S \times M$. Bởi vậy, ta có một song ánh giữa hai không gian khoá. Vì mỗi khoá là đồng xác suất nên có thể thấy rằng $S \times M$ thực sự là mã Affine.

Ta chứng minh rằng $M \times S = S \times M$. Bởi vậy, hai hệ mật là giao hoán. Tuy nhiên, không phải mọi cặp hệ mật đều giao hoán; có thể tìm ta được các cặp phản ví dụ. Mặt khác ta thấy rằng phép tích luôn kết hợp:

$$(S_1 \times S_2) \times S_3 = S_1 \times (S_2 \times S_3)$$

Nếu lấy tích của một hệ mật tự đồng cấu với chính nó thì ta thu được hệ mật $S \times S$ (kí hiệu là S^2). Nếu lấy tích n lần thì hệ mật kết quả là S^n. Ta gọi S^n là hệ mật lặp.

Một hệ mật S được gọi là luỹ đẳng nếu $S^2 = S$. Có nhiều hệ mật đã nghiên cứu trong chương 1 là hê mật luỹ đẳng. Chẳng hạn các hệ MDV, MTT, Affine, Hill, Vigenère và hoán vị đều là luỹ đẳng. Hiển nhiên là nếu hệ mật S là luỹ đẳng thì không nên sử dụng hệ mật tích S^2 vì nó yêu cầu lượng khoá cực lớn mà không có độ bảo mật cao hơn.

Nếu một hệ mật không phải là luỹ đẳng thì có thể làm tăng độ mật bằng cách lặp nhiều lần. Ý tưởng này đã được dùng trong chuẩn mã dữ liệu (DES). Trong DES dùng 16 phép lặp, tất nhiên hệ mật ban đầu phải là hệ mật không luỹ đẳng. Một phương pháp có thể xây dựng các hệ mật không luỹ đẳng đơn giản là lấy tích của hai hệ mật đơn giản khác nhau.

Nhận xét:

Có thể dễ dàng chứng tỏ rằng, nếu cả hai hệ mật S_1 và S_2 là luỹ đẳng và giao hoán thì S_1 và S_2 cũng là luỹ đẳng. Điều này rút ra từ các phép toán đại số sau:

$$\begin{aligned}(S_1 \times S_2) \times (S_1 \times S_2) &= S_1 \times (S_2 \times S_1) \times S_2 \\ &= S_1 (S_1 \times S_2) \times S_2 \\ &= (S_1 \times S_1) \times (S_2 \times S_2) \\ &= S_1 \times S_2\end{aligned}$$

(Chú ý: Dùng tính chất kết hợp trong chứng minh trên).

Bởi vậy, nếu cả S_1 và S_2 đều là luỹ đẳng và ta muốn $S_1 \times S_2$ là không luỹ đẳng thì điều kiện cần là S_1 và S_2 không giao hoán.

Rất may mắn là nhiều hệ mật đơn giản thoả mãn điều kiện trên. Kỹ thuật thường được sử dụng trong thực tế là lấy tích các hệ mã kiểu thay thế và các hệ mã kiểu hoán vị.

2.7. CÁC HỆ MÃ DÒNG

2.7.1. Sơ đồ chức năng của hệ mật mã dòng

Trong các hệ mật nghiên cứu ở trên, các phần tử liên tiếp của bản rõ đều được mã hoá bằng cùng một khoá k. Tức xâu bản mã y nhận được có dạng:

$$y = y_1 y_2 \ldots = e_k(x_1) e_k(x_2) \ldots$$

Các hệ mật thuộc dạng này thường được gọi là các mã khối. Một quan điểm sử dụng khác là mật mã dòng. Ý tưởng cơ bản ở đây là tạo ra một dòng khoá $z = z_1 z_2 \ldots$ và dùng nó để mã hoá một xâu bản rõ $x = x_1 x_2 \ldots$ theo quy tắc:

$$y = y_1 y_2 \ldots = e_{z_1}(x_1) e_{z_2}(x_2) \ldots$$

Sơ đồ mã hóa và giải mã của hệ mật mã dòng mô tả trong Hình 2.9

a) Mã hóa *b) Giải mã*

Hình 2.9. Sơ đồ chức năng hệ mật mã dòng

Mã dòng hoạt động như sau: Giả sử $k \in \mathcal{K}$ là khoá, và $x = x_1 x_2 ...$ là xâu bản rõ. Hàm f_i được dùng để tạo z_i (z_i là phần tử thứ i của dòng khoá), trong đó f_i là một hàm của khoá k và $i-1$ là ký tự đầu tiên của bản rõ:

$$z_i = f_i(k, x_1, ..., x_{i-1})$$

Phần tử z_i của dòng khoá được dùng để mã x_i tạo ra $y_i = e_{z_i}(x_i)$. Bởi vậy, để mã hoá xâu bản rõ $x = x_1 x_2 ...$ ta phải tính liên tiếp $z_1, y_1, z_2, y_2, ...$

Việc giải mã xâu bản mã $y_1 y_2 ...$ có thể được thực hiện bằng cách tính liên tiếp $z_1, x_1, z_2, x_2, ...$

Sau đây là định nghĩa dưới dạng toán học:

Định nghĩa 3.6:

Mật mã dòng là một bộ $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{L}, \mathcal{F}, \mathcal{E}, \mathcal{D})$ *thoả mãn các điều kiện sau:*

(1) $\mathcal{P}$ *là một tập hữu hạn các bản rõ có thể.*

(2) $\mathcal{C}$ *là tập hữu hạn các bản mã có thể.*

(3) $\mathcal{K}$ là *tập hữu hạn các khoá có thể (không gian khoá)*

(4) $\mathcal{L}$ *là tập hữu hạn các bộ chữ của dòng khoá.*

(5) $\mathcal{F} = (f_1 f_2 ...)$ *là bộ tạo dòng khoá. Với* $i \geq 1$

$$f_i : \mathcal{K} \times \mathcal{P}^{i-1} \to \mathcal{L}$$

(6) Với mỗi $z \in \mathcal{L}$ *có một quy tắc mã* $e_z \in \mathcal{E}$ *và một quy tắc giải mã tương ứng* $d_z \in \mathcal{D}$. $e_z : \mathcal{P} \to \mathcal{C}$ *và* $d_z : \mathcal{C} \to \mathcal{P}$ *là các hàm thoả mãn* $d_z(e_z(x)) = x$ *với mọi bản rõ* $x \in \mathcal{P}$.

Ta có thể coi mã khối là một trường hợp đặc biệt của mã dòng, trong đó dùng khoá không đổi: $z_i = k$ với mọi $i \geq 1$.

Nhận xét:

- Để hệ thống an toàn, dãy bit khóa ngẫu nhiên và $|k| \geq |m|$. (Dãy ngẫu nhiên được lấy là kết quả của quá trình tung đồng xu: $p(0) = p(1) = 0,5$).

- Việc tạo dãy ngẫu nhiên rất tốn kém và việc lưu trữ không hiệu quả, do vậy ta phải sử dụng dãy giả ngẫu nhiên, các dãy này có tính tiền định và được xây dựng từ các bit mầm.

2.7.2. Tạo dãy giả ngẫu nhiên (M-dãy)

2.7.2.1. Tạo dãy giả ngẫu nhiên theo đa thức nguyên thủy

Định nghĩa 2.6: Đa thức nguyên thủy

Đa thức bất khả quy bậc m được gọi là đa thức nguyên thủy nếu nó là ước của x^n+1 với $n=2^m-1$ nhưng không là ước của $x^\ell+1$ với $\ell<n$.

(chú ý: đa thức bất khả quy là đa thức chia hết cho 1 và chính nó, tương đương số nguyên tố)

Ví dụ 2.13:

+ $m=3\rightarrow n=7$, ta có: $x^7+1=(1+x)(1+x+x^3)(1+x^2+x^3)$.

Trên vành đa thức x^7+1 có hai đa thức: $f_1(x)=(1+x+x^3)$ và $f_2(x)=(1+x^2+x^3)$ là hai đa thức nguyên thủy vì nó không là ước của $x^\ell+1$ với $\ell\le 6$.

+ $m=4\rightarrow n=15$, ta có

$$x^{15}+1=(1+x)(1+x+x^2)(1+x+x^4)(1+x^3+x^4)(1+x+x^2+x^3+x^4)$$

Với trường hợp này chỉ có hai đa thức $1+x+x^4$ và $1+x^3+x^4$ là nguyên thủy, còn đa thức $1+x+x^2+x^3+x^4$ không phải nguyên thủy vì nó là ước của x^5+1.

$$x^5+1=(1+x)(1+x+x^2+x^3+x^4)$$

Bổ đề 2.14:

Dãy giả ngẫu nhiên (M-dãy) được tạo từ phương trình đồng dư sau đây:

$$\boxed{a(x)\equiv b(x)x^i \bmod g(x)};\ i=1,2,...,2^n-1 \qquad (2.15)$$

Với: $g(x)$ là đa thức nguyên thủy bậc m

$b(x)$ là đa thức mầm ứng với m bit, được chọn ngẫu nhiên thỏa mãn $\deg b(x)\le m-1$

Ví dụ 2.14:

Với $m=4; g(x)=1+x+x^4$, M-dãy được tạo như sau:

$$a(x)\equiv b(x)x^i \bmod 1+x+x^4$$

Coi $1+x+x^4=0\rightarrow x^4=1+x$. Giả sử $b(x)=1+x\leftrightarrow 1100$

Trạng thái của M-dãy này được mô tả trong Bảng 2.1

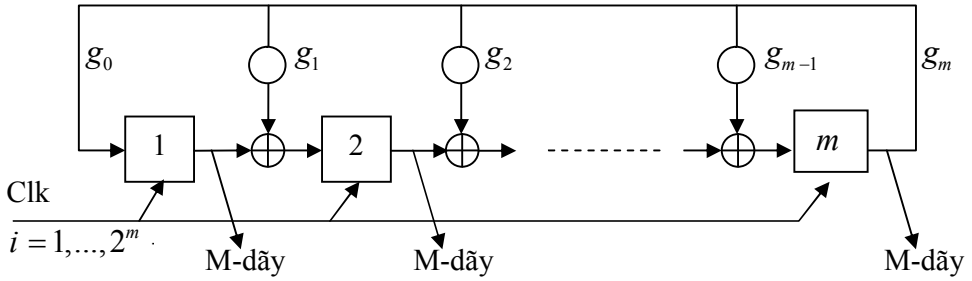
Bảng 2.1. Trạng thái của M-dãy

TT	$a(x)$	$\vec{a}$
0	$1+x$	1 1 0 0
1	$x+x^2$	0 1 1 0
2	x^2+x^3	0 0 1 1
3	$1+x+x^3$	1 1 0 1
4	$1+x^2$	1 0 1 0
5	$x+x^3$	0 1 0 1
6	$1+x+x^2$	1 1 1 0
7	$x+x^2+x^3$	0 1 1 1
8	$1+x+x^2+x^3$	1 1 1 1
9	$1+x^2+x^3$	1 0 1 1
10	$1+x^3$	1 0 0 1
11	1	1 0 0 0
12	x	0 1 0 0
13	x^2	0 0 1 0
14	x^3	0 0 0 1
15	$1+x$	1 1 0 0

Nhận xét:

- Khi lấy bất kỳ một cột nào trong 4 cột của $\vec{a}$ ta sẽ được một M-dãy.
- Chu kỳ của dãy: $t = 2^m - 1$ với trường hợp này $m = 4 \rightarrow t = 15$.
- Số con "1" trong dãy: $N_1 = 2^m$, với $m = 4 \rightarrow N_1 = 8$.
- Số con "0" trong dãy: $N_0 = 2^m - 1$, với $m = 4 \rightarrow N_0 = 7$.
- Khi $m \rightarrow \infty$ ta có: $\lim_{m\to\infty} p(0) = \lim_{m\to\infty} p(1) = 1/2$

Cấu trúc tổng quát mạch điện phần cứng M-dãy được thực hiện bằng các thanh ghi dịch hồi tiếp tuyến tính (LFSR) và có dạng như Hình 2.10.

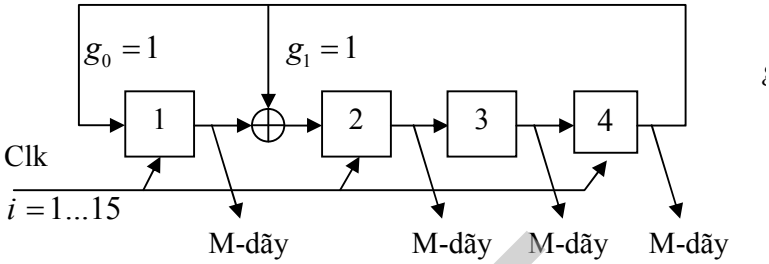


Hình 2.10. Mạch điện thực hiện M-dãy

Trong sơ đồ Hình 2.10 các g_i nhận giá trị "0" hoặc "1" là các hệ số của đa thức nguyên thủy $g(x) = g_0 + g_1x + g_2x^2 + ... + g_mx^m$. Riêng $g_0 = g_m = 1$ (luôn bằng "1"). Nếu $g_i = 1$ thì mạch điện sẽ nối tắt, còn $g_i = 0$ thì sẽ hở mạch.

Ví dụ 2.15:

Giả sử $g(x) = 1 + x + x^4$, $(m = 4)$, ta thấy $g_0 = g_1 = g_4 = 1; g_2 = g_3 = 0$, mạch điện tạo M-dãy như sau:



Hình 2.11. Mạch điện tạo M-dãy với $g(x) = 1 + x + x^4$

Bảng 2.2. Trạng thái hoạt động của các ô nhớ

Nhịp i	**Trạng thái các ô nhớ**			
	1	**2**	**3**	**4**
0	1	1	0	0
1	0	1	1	1
2	0	0	1	1
3	1	1	0	1
4	1	0	1	0
5	0	1	0	1
6	1	1	1	0
7	0	1	1	1
8	1	1	1	1
9	1	0	1	1
10	1	0	0	1
11	1	0	0	0
12	0	1	0	0
13	0	0	1	0
14	0	0	0	1
15	1	1	0	0

Trong Bảng 2.2 thì trạng thái các ô nhớ tại dòng đầu tiên tương ứng với các bit mầm (đa thức $b(x) = 1 + x$).

2.7.2.2. Tạo dãy giả ngẫu nhiên trên vành đa thức có hai lớp kề xyclic

Định nghĩa 2.7: Vành đa thức có hai lớp kề xyclic là vành có phân tích như sau:

$$x^n + 1 = (1 + x)\sum_{i=0}^{n-1} x^i$$

Trong đó: $(1+x)$ và $\sum_{i=0}^{n-1} x^i$ là các đa thức bất khả quy.

Ví dụ: $n = 5,11,19,...$

$$x^5 + 1 = (1+x)(1+x+x^2+x^3+x^4)$$

Bổ đề 2.15:

M-dãy trên vành đa thức có hai lớp kề xyclic $x^n + 1$ được tạo từ phương trình đồng dư sau:

$$\boxed{a(x) \equiv b(x)c^i(x) \bmod \sum_{i=0}^{n-1} x^i} \quad (2.16)$$

Trong đó: $\sum_{i=0}^{n-1} x^i$ là đa thức bất khả quy

$b(x)$ là đa thức bất kỳ, thỏa mãn $\deg b(x) \le n-2$

$c(x)$ là đa thức thỏa mãn $ord\, c(x) = \max = 2^{n-1} - 1$

Định nghĩa 2.8:

Cấp của đa thức $c(x)$ là cấp của nhóm nhân xyclic sinh bởi $c(x)$

$$C = \{c^i(x), i = 1,2,...\} \Rightarrow ord\, c(x) = |C|$$

Ví dụ 2.16:

Xét trường hợp $n = 5$ và $\sum_{i=0}^{4} x^i$ là đa thức bất khả quy, khi đó M-dãy được xây dựng theo công thức (2.16) có dạng như sau:

$$a(x) \equiv b(x)c^i(x) \bmod \sum_{i=0}^{4} x^i$$

Chú ý: để lấy modulo theo $\sum_{i=0}^{4} x^i = 1+x+x^2+x^3+x^4$ ta coi $1+x+x^2+x^3+x^4 = 0$ tức là coi $x^4 = 1+x+x^2+x^3$.

Chọn $b(x) = 1 \leftrightarrow (0)$ và $c(x) = 1+x^2+x^4 \leftrightarrow (024)$, chú ý (024) là dạng biểu diễn số mũ của $c(x)$. Nhóm nhân xây dựng từ đa thức sinh $c(x)$ như sau:

$$C = \{c^i(x), i = 1,2,...\} = \{(024),(034),(1),(013),(014),(2),(124)$$
$$(012),(3),(023),(123),(4),(134),(234),(0)\}$$

Trên cơ sở nhóm nhân C ta tính được M-dãy như sau:

$$A=\left\{c^i(x) \bmod \sum_{i=0}^{4} x^i\right\}=\{(13),(12),(1),(013),(23),(2),(03),(012)$$
$$(3),(023),(123),(0123),(02),(01),(0)\}$$

Bảng 2.3. M-dãy trên vành x^5+1

TT	Đa thức dạng số mũ	Đa thức	Dạng vectơ
1	(13)	$x+x^3$	0 1 0 1 ← M -dãy
2	(12)	$x+x^2$	0 1 1 0
3	(1)	x	0 1 0 0
4	(013)	$1+x+x^3$	1 1 0 1
5	(23)	x^2+x^3	0 0 1 1
6	(2)	x^2	0 0 1 0
7	(03)	$1+x^3$	1 0 0 1
8	(012)	$1+x+x^2$	1 1 1 0
9	(3)	x^3	0 0 0 1
10	(023)	$1+x^2+x^3$	1 0 1 1
11	(123)	$x+x^2+x^3$	0 1 1 1
12	(0123)	$1+x+x^2+x^3$	1 1 1 1
13	(02)	$1+x^2$	1 0 1 0
14	(01)	$1+x$	1 1 0 0
15	(0)	1	1 0 0 0

M-dãy trong Bảng 2.3 thực chất là một hoán vị của M-dãy trong Bảng 2.1.

Số các đa thức nguyên thủy (các phần tử sinh) tính theo công thức sau:

$$N=\varphi(|C|)$$

Trong đó φ là hàm Phi-Euler, giá trị của $\varphi(|C|)$ cho biết số các số nguyên tố cùng nhau với $|C|$. Cách tính φ đã được trình bày ở mục 1.1.12.

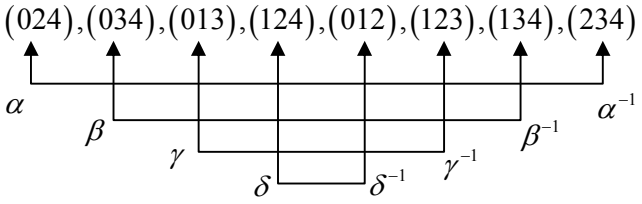
Ví dụ 2.17:

Xét $n=15=3.5$, khi đó:

$$\Phi(15)=15\left(1-\frac{1}{3}\right)\left(1-\frac{1}{5}\right)=8$$

Nếu α là một phần tử nguyên thủy thì α^i cũng là phần tử nguyên thủy với $(i,n)=1$.

Với trường hợp $n = 15$ ta có: $i = \{1,2,4,7,8,11,13,14\}$ tức là có 8 phần tử bằng với $\varphi(15)$. Và 8 phần tử nguyên thủy này sẽ tạo thành một nhóm nhân, các phần tử đều có nghịch đảo như mô tả như hình 2.12 sau đây:



Hình 2.12.

$$(024) = (234)^{-1} \; hay \; 1 + x^2 + x^4 = \frac{1}{x^2 + x^3 + x^4}$$

2.8. CHUẨN MÃ DỮ LIỆU

2.8.1. Mở đầu

Ngày 15/5/1973. Uỷ ban tiêu chuẩn quốc gia Mỹ đã công bố một khuyến nghị cho các hệ mật trong Hồ sơ quản lý liên bang. Điều này cuối cùng đã dẫn đến sự phát triển của Chuẩn mã dữ liệu (DES) và nó đã trở thành một hệ mật được sử dụng rộng rãi nhất trên thế giới. DES được IBM phát triển và được xem như một cải biên của hệ mật LUCIPHER. DES được công bố lần đầu tiên trong Hồ sơ Liên bang vào ngày 17/3/1975. Sau nhiều cuộc tranh luận công khai, DES đã được chấp nhận chọn làm chuẩn cho các ứng dụng không được coi là mật vào 5/1/1977. Kể từ đó cứ 5 năm một lần, DES lại được Uỷ ban Tiêu chuẩn Quốc gia xem xét lại. Lần đổi mới gần đây nhất của DES là vào tháng 1/1994 và tiếp tới sẽ là 1998. Người ta đoán rằng DES sẽ không còn là chuẩn sau 1998.

2.8.2. Mô tả DES

Mô tả đầy đủ của DES được nêu trong Công bố số 46 về các chuẩn xử lý thông tin Liên bang (Mỹ) vào 15/1/1977. DES mã hoá một xâu bit x của bản rõ độ dài 64 bằng một khoá 54 bit. Bản mã nhận được cũng là một xâu bit có độ dài 64. Trước hết ta mô tả ở mức cao về hệ thống.

Thuật toán tiến hành theo 3 giai đoạn:

1. Với bản rõ cho trước x, một xâu bit x_0 sẽ được xây dựng bằng cách hoán vị các bit của x theo phép hoán vị cố định ban đầu IP. Ta viết:

   $x_0 = \text{IP}(x) = L_0 R_0$

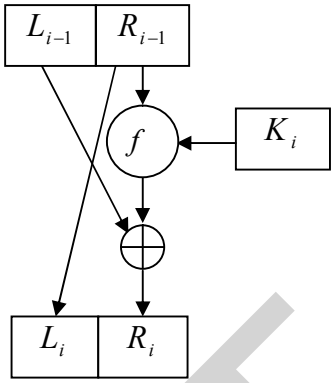
   trong đó L_0 gồm 32 bit đầu và R_0 là 32 bit cuối.

2. Sau đó tính toán 16 lần lặp theo một hàm xác định. Ta sẽ tính $L_i, R_i, 1 \le i \le 16$ theo quy tắc sau:

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, k_i) \end{cases}$$

trong đó $\oplus$ kí hiệu phép hoặc loại trừ của hai xâu bit (cộng theo modulo 2). f là một hàm mà ta sẽ mô tả ở sau, còn $k_1,k_2,...,k_{16}$ là các xâu bit độ dài 48 được tính như hàm của khoá k. (trên thực tế mỗi k_i là một phép chọn hoán vị bit trong k).

$k_1,k_2,...,k_{16}$ sẽ tạo thành bảng khoá. Một vòng của phép mã hoá được mô tả trên Hình 2.13.

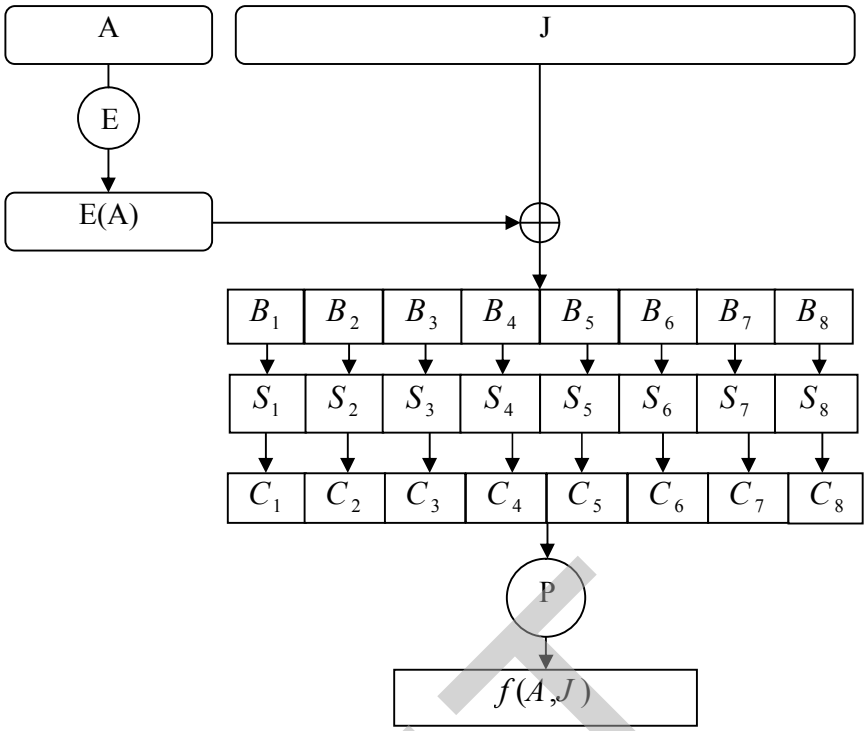


Hình 2.13. Một vòng của DES

3. Áp dụng phép hoán vị ngược IP^{-1} cho xâu bit $R_{16}L_{16}$, ta thu được bản mã y. Tức là $y = \text{IP}^{-1}(R_{16}L_{16})$. Hãy chú ý thứ tự đã đảo của R_{16} và L_{16}.

Hàm f có hai biến vào: biến thứ nhất *A* là xâu bit độ dài 32, biến thứ hai *J* là một xâu bit độ dài 48. Đầu ra của *f* là một xâu bit độ dài 32. Các bước sau được thực hiện:

1. Biến thứ nhất *A* được mở rộng thành một xâu bit độ dài 48 theo một hàm mở rộng cố định *E*(*EA*) gồm 32 bit của *A* (được hoán vị theo cách cố định) với 16 bit xuất hiện hai lần.
2. Tính $E(A) \oplus J$ và viết kết quả thành một chuỗi 8 xâu 6 bit = $B_1B_2B_3B_4B_5B_6$.
3. Bước tiếp theo dùng 8 bảng $S_1,S_2,...,S_8$ (được gọi là các hộp S). Với mỗi S_i là một bảng 4×16 cố định có các hàng là các số nguyên từ 0 đến 15. Với xâu bit có độ dài 6 (kí hiệu $B_i = b_1b_2b_3b_4b_5b_6$), ta tính $S_j(B_j)$ như sau: hai bit b_1b_6 xác định biểu diễn nhị phân của hàng r của $S_i(0 \le r \le 3)$ và bốn bit $b_2b_3b_4b_5$ xác định biểu diễn nhị phân của cột c của $S_i(0 \le c \le 15)$. Khi đó, $S_j(B_j)$ sẽ xác định phần tử $S_j(r,c)$; phần tử này viết dưới dạng nhị phân là một xâu bit có độ dài 4. (Bởi vậy, mỗi S_j có thể được coi là một hàm mã mà đầu vào là một xâu bit có độ dài 2 và một xâu bit có độ dài 4, còn đầu ra là một xâu bit có độ dài 4). Bằng cách tương tự tính các $C_j = S_j(B_j), 1 \le j \le 8$.
4. Xâu bit $C = C_1C_2..C_8$ có độ dài 32 được hoán vị theo phép hoán vị cố định P. Xâu kết quả là $P(C)$ được xác định là $f(A,J)$.



Hình 2.14. Hàm *f* của DES

Hàm *f* được mô tả trong Hình 2.14. Chủ yếu nó gồm một phép thế (sử dụng hộp *S*), tiếp sau đó là phép hoán vị *P*. 16 phép lặp của *f* sẽ tạo nên một hệ mật tích nêu như ở mục 2.6.

Trong phần còn lại của mục này, ta sẽ mô tả hàm cụ thể được dùng trong DES. Phép hoán vị ban đầu IP như sau:

Bảng 2.4. Bảng IP của DES

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Bảng này có nghĩa là bit thứ 58 của x là bit đầu tiên của $\mathrm{IP}(x)$; bit thứ 50 của x là bit thứ hai của $\mathrm{IP}(x)$.v.v...

Phép hoán vi ngược IP^{-1} là:

Bảng 2.5. Bảng IP^{-1} của DES

IP^{-1}							
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

Hàm mở rộng E được xác đinh theo bảng sau:

Bảng 2.6. Hàm mở rộng E của DES

Bảng chọn E bit					
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

Tám hộp S như sau:

S_1															
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

S_2															
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

S_3															
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

S_4															
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
S_5															
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	1	14	2	13	6	15	0	9	10	4	5	3

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	15	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

S_7															
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

S_8															
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

Và phép hoán vị P có dạng:

Bảng 2.7. Phép hoàn vị P trong hàm *f* của DES

P			
16	7	20	21
29	12	28	17
1	15	23	26
2	8	24	14
5	18	31	10
32	27	3	9
19	13	30	6
22	11	4	25

Cuối cùng, ta cần mô tả việc tính toán bảng khoá từ khoá k. Trên thực tế, k là một xâu bit độ dài 64, trong đó 56 bit là khoá và 8 bit để kiểm tra tính chẵn lẻ nhằm phát hiện sai. Các bit ở các vị trí 8,16,..., 64 được xác định sao cho mỗi byte chứa một số lẻ các số "1". Bởi vậy, một sai sót đơn lẻ có thể phát hiện được trong mỗi nhóm 8 bit. Các bit kiểm tra bị bỏ qua trong quá trình tính bảng khoá.

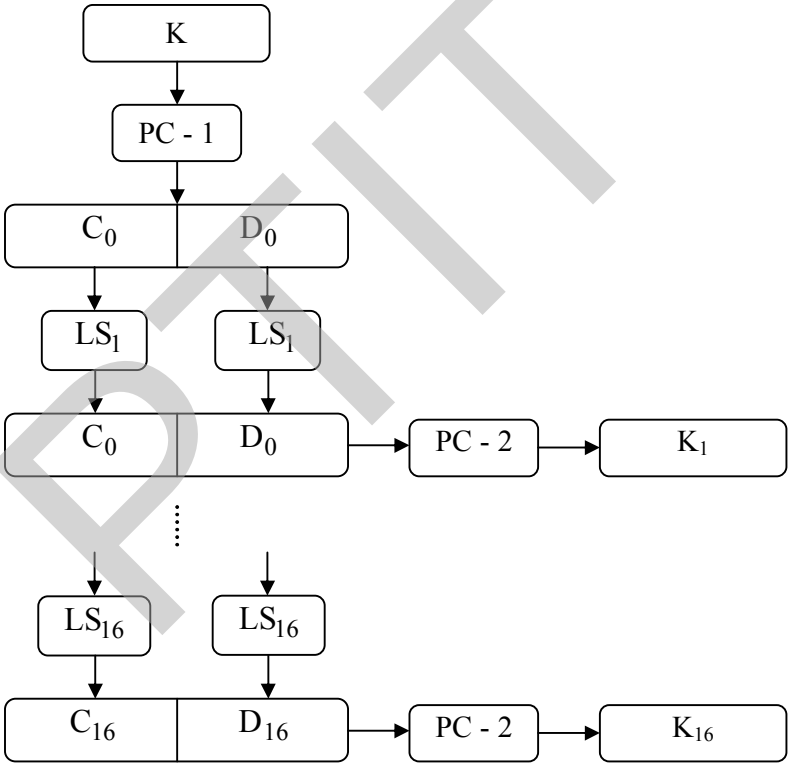
Với một khoá k 64 bit cho trước, ta loại bỏ các bit kiểm tra tính chẵn lẻ và hoán vị các bit còn lại của k theo phép hoán vị cố định PC-1. Ta viết:

$$PC-1(k)=C_0D_0$$

Với i thay đổi từ 1 đến 16:

$$C_i = LS_i(C_{i-1})$$
$$D_i = LS_i(D_{i-1})$$

Việc tính bảng khoá được mô tả trên Hình 2.15.



Hình 2.15. Tính bảng khoá DES

Các hoán vị PC-1 và PC-2 được dùng trong bảng khoá.

Bảng 2.8. Hoán vị PC-1

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 2.9. Hoán vị PC-2

PC-2					
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

Bây giờ ta sẽ đưa ra bảng khoá kết quả. Như đã nói ở trên, mỗi vòng sử dụng một khoá 48 bit gồm 48 bit nằm trong K. Các phần tử trong các bảng dưới đây biểu thị các bit trong K trong các vòng khoá khác nhau.

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

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

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

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

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

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

Vòng 8
36 41 60 50 10 43 59 18 57 35 9 3
58 25 52 51 34 19 49 27 26 17 44 2
12 54 61 13 31 30 6 20 62 47 45 23
55 15 28 22 37 46 39 4 7 21 14 53

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

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

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

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

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

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

Vòng 16
18 59 42 3 57 25 41 36 10 17 27 50
11 43 34 33 52 1 2 9 44 35 26 49
30 5 47 62 45 12 55 38 13 61 31 37
6 29 46 4 23 28 53 22 21 7 63 39

Phép giải mã được thực hiện nhờ dùng cùng thuật toán như phép mã nếu đầu vào là y nhưng dùng bảng khoá theo thứ tự ngược lại $K_{16}...K_1$. Đầu ra của thuật toán sẽ là bản rõ x.

Một ví dụ về DES

Sau đây là một ví dụ về phép mã DES. Giả sử ta mã bản rõ (ở dạng mã hexa):

0 1 2 3 4 5 6 7 8 9 A B C D E F

Bằng cách dùng khoá

1 2 3 4 5 7 7 9 9 B B C D F F 1

Khoá ở dạng nhị phân (không chứa các bit kiểm tra) là:

0001001001101001010110111100100110110111101101111111000

Sử dụng IP, ta thu được L_0 và R_0 (ở dạng nhị phân) như sau:

L_0 = 11001100000000000110011001111111
$L_1 = R_0$ = 11110000101010101111000010101010

Sau đó thực hiện 16 vòng của phép mã như sau:

$E(R_0)$ = 011110100001010101010101011110100001010101010101 K_1 = 000110110000001011101111111111000111000001110010 $E(R_0) \oplus K_1$ = 011000010001011110111010100001100110010100100111 S-box outputs 01011100100000101011010110010111 $f(R_0,K_1)$ = 00100011010010101010100110111011 $L_2 = R_1$ = 11101111010010100110010101000100
$E(R_1)$ = 011101011110101001010100001100001010101000001001 K_2 = 011110011010111011011001110110111100100111100101 $E(R_1) \oplus K_2$ = 000011000100010010001101111010110110001111101100 S-box outputs 11111000110100000011101010101110 $f(R_1,K_2)$ = 00111100101010111000011110100011 $L_3 = R_2$ = 11001100000000001011011100001001
$E(R_2)$ = 11100101100000000000001010111010111010000101010011 K_3 = 01010101111111001000101001000010110011111100111001 $E(R_2) \oplus K_3$ = 10110000011111001000100011111000001001111100101O S-box outputs 00100111000100001110000101101111 $f(R_2,K_3)$ = 01001101000101100110111010110000 $L_4 = R_3$ = 10100010010111000000101111110100
$E(R_3)$ =01010000010000101111100000000101011111111010100 K_4 = 01110010101011011101011011011011001101010001101 $E(R_3) \oplus K_4$ = 00100010111011110010111011011110010010101011010O S-box outputs 00100001111011011001111100111010 $f(R_3,K_4)$ = 10111011001000110111011101001100 $L_5 = R_4$ = 01110111001000100000000001000101
$E(R_4)$ = 10111010111010010000010000000000000000001000001010 K_5 = 01111100111011000000011111101011010100111010100O $E(R_4) \oplus K_5$ = 11000110000001010000001111101011010100011010001O S-box outputs 01010000110010000011000111101011 $f(R_4,K_5)$ = 00101000000100111010110111000011 $L_6 = R_5$ = 10001010010011111010011000110111

$E(R_5)$ = 11000101010000100101111111010000110000011010111
K_6 = 01100011101001010011111001010000011110110010111
$E(R_5) \oplus K_6$ =10100110111001110110000110000000101110101000000
S-box outputs 01000001111100110100110000111101
$f(R_5,K_6)$ = 10011110010001011100110100101100
$L_7 = R_6$ = 11101001011001111100110101101001
$E(R_6)$ = 11110101001010110000111111100101101010110101010011
K_7 = 11101100100001001011011111111011000011000101111100
$E(R_6) \oplus K_7$ = 00011001101011111011100000010011101100111111101111
S- box outputs 00010000011101010100000010101101
$f(R_6,K_7)$ = 10001100000001010001110000100111
$L_8 = R_7$ = 00000110010010101011101000010000
$E(R_7)$ = 00000000110000100101010101011111010000001010000
K_8 = 11110111100010100011101011000001001110111111111011
$E(R_7) \oplus K_8$ = 11110111010010000110111110011110011110110101011011
S-box outputs 01101100000110000111110010101110
$f(R_7,K_8)$ = 00111100000011101000011011111001
$L_9 = R_8$ = 11010101011010010100101110010000
$E(R_8)$ = 01101010101010110101001010100101011111100101000001
K_9 = 11100000110110111110101111101101111001111000000001
$E(R_8) \oplus K_9$ = 10001010011100001011100101001000100110110010000
S-box outputs 00010001000011000101011101110111
$f(R_8,K_9)$ = 00100010001101100111110001101010
$L_{10} = R_9$ = 00100100011111001100011001111010
$E(R_9)$ = 00010000100000111111100101100000110000111111010
K_{10} = 10110001111100110100011110111010010000110010011111
$E(R_9) \oplus K_{10}$ = 10100001011100001011111011011010100001011011101
S-box outputs 11011010000001000101001001110101
$f(R_9,K_{10})$ = 01100010101111001001110000100010
$L_{11} = R_{10}$ = 10110111110101011101011110110010
$E(R_{10})$ = 01011010111111101010101111110101011111101101001
K_{11} = 00100001010111111101001111011110110100111000011
$E(R_{10}) \oplus K_{11}$ = 01111011101000010111100000110100001011100010001
S-box outputs 01110011000001011101000100000001
$f(R_{10},K_{11})$ = 11100001000001001111101000000010
$L_{12} = R_{11}$ = 11000101011110000011110001111000
$E(R_{11})$ = 01100000101010111111000000011111100000111111000
K_{12} = 01110101011100011111010110010100011001111110100
$E(R_{11}) \oplus K_{12}$ = 00010101110110100000010110001011111001000001100
S-box outputs 01110011000001011101000100000001
$f(R_{11},K_{12})$ = 11000010011010001100111111101010
$L_{13} = R_{12}$ = 01110101101111010001100001011000
$E(R_{12})$ = 00111010101111011111101010001111000000101111000
K_{13} = 10010111110001011101000111111010101110100100000
$E(R_{12}) \oplus K_{13}$ = 10101101011110000010101101110101101110001011000
Sbox outputs 10011010110100011000101101001111
$f(R_{12},K_{13})$ = 11011101101110110010100100100010

$L_{14} = R_{13} = 00011000110000110001010101011010$

$E(R_{13}) = 000011110001011000000110100010101010101011110100$ $K_{13} = 010111110100001110110111111100101110011100111010$ $E(R_{13}) \oplus K_{14} = 010100000101010110110001011110000100110111001110$ S-box outputs 01100100011110011001101011110001 $f(R_{13},K_{14}) = 10110111001100011000111001010101$ $L_{15} = R_{14} = 11000010100011001001011000001101$
$E(R_{14}) = 11100000010101000101100101001010110000000001011011$ $K_{15} = 10111111100100011000110100111101001111110000101 0$ $E(R_{14}) \oplus K_{15} = 01011111110001011101010001110111111111111010100 01$ S-box outputs 10110010111010001000110100111100 $f(R_{14},K_{15}) = 01011011100000010010011101101110$ $R_{15} = 01000011010000100011001000110100$
$E(R_{15}) = 00100000011010100000010000011010010000011010 1000$ $K_{16} = 11001011001111011000101100001110000101111111 0101$ $E(R_{15}) \oplus K_{16} = 11101011010101111000111100010100010101100101 1101$ S-box outputs 10100111100000110010010000101001 $f(R_{15},K_{16}) = 11001000110000000100111110011000$ $R_{16} = 00001010010011001101100110010101$

Cuối cùng, áp dụng IP^{-1} vào L_{16},R_{16} ta nhận được bản mã hexa là:

85E813540F0AB405

2.8.3. Một số ý kiến thảo luận về DES

Khi DES được đề xuất như một chuẩn mật mã, đã có rất nhiều ý kiến phê phán. Một lý do phản đối DES có liên quan đến các hộp *S*. Mọi tính toán liên quan đến DES ngoại trừ các hộp *S* đều tuyến tính, tức việc tính phép hoặc loại trừ của hai đầu ra cũng giống như phép hoặc loại trừ của hai đầu vào rồi tính toán đầu ra. Các hộp *S* - chứa đựng thành phần phi tuyến của hệ mật là yếu tố quan trong nhất đối với độ mật của hệ thống (Ta đã thấy là các hệ mật tuyến tính - chẳng hạn như Hill - có thể dễ dàng bị mã thám khi bị tấn công bằng bản rõ đã biết). Tuy nhiên, tiêu chuẩn xây dựng các hộp S không được biết đầy đủ. Một số người đã gợi ý là các hộp S phải chứa các "cửa sập" được dấu kín, cho phép Cục An ninh Quốc gia Mỹ (NSA) giải mã được các thông báo nhưng vẫn giữ được mức độ an toàn của DES. Dĩ nhiên ta không thể bác bỏ được khẳng định này, tuy nhiên không có một chứng cớ nào được đưa ra để chứng tỏ rằng trong thực tế có các cửa sập như vậy.

Năm 1976 NSA đã khẳng định rằng, các tính chất sau của hộp *S* là tiêu chuẩn thiết kế:

- Mỗi hàng trong mỗi hộp *S* là một hoán vị của các số nguyên 0, 1,... , 15.
- Không một hộp *S* nào là một hàm Affine hoặc tuyến tính các đầu vào của nó.
- Việc thay đổi một bit vào của S phải tạo nên sự thay đổi ít nhất là hai bit ra.
- Đối với hộp *S* bất kỳ và với đầu vào x bất kỳ $S(x)$ và $S(x \oplus 001100)$ phải khác nhau tối thiểu là hai bit (trong đó x là xâu bit độ dài 6).
- Hai tính chất khác nhau sau đây của các hộp *S* có thể coi là được rút ra từ tiêu chuẩn thiết kế của NSA.

- Với hộp *S* bất kỳ, đầu vào x bất kỳ và với $e,f \in \{0,1\} : S(x) \neq S(x \oplus 1 \text{lef} 00)$.
- Với hộp *S* bất kỳ, nếu cố định một bit vào và xem xét giá trị của một bit đầu ra cố định thì các mẫu vào để bit ra này bằng 0 sẽ xấp xỉ bằng số mẫu ra để bit đó bằng 1. (Chú ý rằng, nếu cố định giá trị bit vào thứ nhất hoặc bit vào thứ 6 thì có 16 mẫu vào làm cho một bit ra cụ thể bằng 0 và có 16 mẫu vào làm cho bit này bằng 1. Với các bit vào từ bit thứ hai đến bit thứ 5 thì điều này không còn đúng nữa. Tuy nhiên, phân bố kết quả vẫn gần với phân bố đều. Chính xác hơn, với một hộp *S* bất kỳ, nếu ta cố định giá trị của một bit vào bất kỳ thì số mẫu vào làm cho một bit ra cố định nào đó có giá trị 0 (hoặc 1) luôn nằm trong khoảng từ 13 đến 19).

Người ta không biết rõ là liệu có còn một chuẩn thiết kế nào đầy đủ hơn được dùng trong việc xây dựng hộp S hay không.

Sự phản đối xác đáng nhất về DES chính là kích thước của không gian khoá: 2^{56} là quá nhỏ để đảm bảo an toàn thực sự. Nhiều thiết bị chuyên dụng đã được đề xuất nhằm phục vụ cho việc tấn công với bản rõ đã biết. Phép tấn công này chủ yếu thực hiện tìm khoá theo phương pháp vét cạn. Tức với bản rõ x 64 bit và bản mã y tương ứng, mỗi khoá đều có thể được kiểm tra cho tới khi tìm được một khoá k thoả mãn $e_k(x) = y$. (Cần chú ý là có thể có nhiều hơn một khoá k như vậy).

Ngay từ năm 1977, Diffie và Hellman đã gợi ý rằng có thể xây dựng một chip VLSI (mạch tích hợp mật độ lớn) có khả năng kiểm tra được 10^6 khoá /giây. Một máy có thể tìm toàn bộ không gian khoá cỡ 10^6 trong khoảng 1 ngày. Họ ước tính chi phí để tạo một máy như vậy khoảng 2.10^7\$.

Trong cuộc hội thảo tại hội nghị CRYPTO'93, Michael Wiener đã đưa ra một thiết kế rất cụ thể về máy tìm khoá. Máy này xây dựng trên một chip tìm khoá, có khả năng thực hiện đồng thời 16 phép mã và tốc độ tới 5×10^7 khoá/giây. Với công nghệ hiện nay, chi phí chế tạo khoảng 10,5\$/chip. Giá của một khung máy chứa 5760 chip vào khoảng 100.000\$ và như vậy nó có khả năng tìm ra một khoá của DES trong khoảng 1,5 ngày. Một thiết bị dùng 10 khung máy như vậy có giá chừng 10^6 \$ sẽ giảm thời gian tìm kiếm khoá trung bình xuống còn 3,5 giờ.

2.8.4. DES trong thực tế

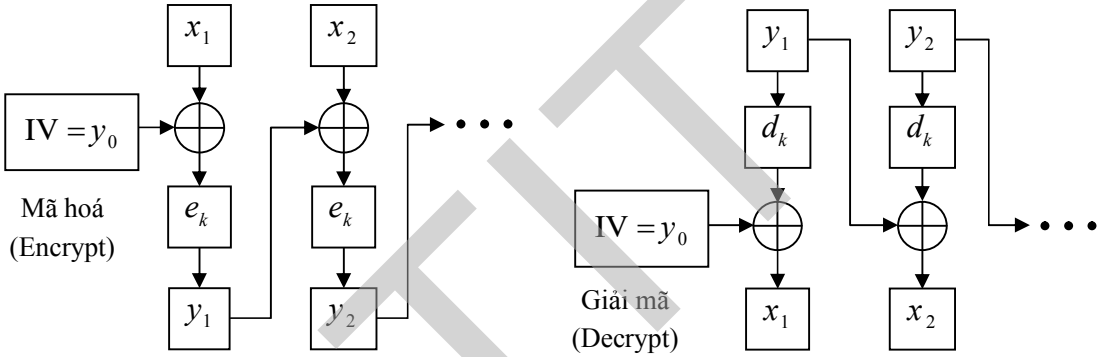
Mặc dù việc mô tả DES khá dài dòng song người ta có thể thực hiện DES rất hữu hiệu bằng cả phần cứng lẫn phần mềm. Các phép toán duy nhất cần được thực hiện là phép hoặc loại trừ các xâu bit. Hàm mở rộng *E*, các hộp *S*, các hoán vị IP và *P* và việc tính toán các giá trị $K_1,...,K_{16}$ đều có thể thực hiện được cùng lúc bằng tra bảng (trong phần mềm) hoặc bằng cách nối cứng chúng thành một mạch.

Các ứng dụng phần cứng hiện thời có thể đạt được tốc độ mã hoá cực nhanh. Công ty Digital Equipment đã thông báo tại hội nghị CRYPTO'92 rằng họ đã chế tạo một chip có 50 ngàn tranzistor có thể mã hoá với tốc độ 1 Gbit/s bằng cách dùng nhịp có tốc độ 250MHz. Giá của chip này vào khoảng 300\$. Tới năm 1991 đã có 45 ứng dụng phần cứng và chương trình cơ sở của DES được Uỷ ban tiêu Chuẩn quốc gia Mỹ (NBS) chấp thuận.

Một ứng dụng quan trọng của DES là trong giao dịch ngân hàng Mỹ - (ABA) DES được dùng để mã hoá các số định danh cá nhân (PIN) và việc chuyển tài khoản bằng máy thủ quỹ tự động (ATM). DES cũng được Hệ thống chi trả giữa các nhà băng của Ngân hàng hối đoái (CHIPS) dùng để xác thực các giao dịch vào khoảng trên $1{,}5\times10^{12}$ USA/tuần. DES còn được sử dụng rộng rãi trong các tổ chức chính phủ. Chẳng hạn như Bộ năng lượng, Bộ Tư pháp và Hệ thống dự trữ liên bang.

2.8.4.1. Các chế độ hoạt động của DES

Có 4 chế độ làm việc đã được phát triển cho DES: Chế độ quyển mã điện tử (ECB), chế độ phản hồi mã (CFB), chế độ liên kết khối mã (CBC) và chế độ phản hồi đầu ra (OFB). Chế độ ECB tương ứng với cách dùng thông thường của mã khối: với một dãy các khối bản rõ cho trước $x_1, x_2, \ldots$ (mỗi khối có 64 bit), mỗi x_i sẽ được mã hoá bằng cùng một khoá k để tạo thành một chuỗi các khối bản mã $y_1, y_2, \ldots$ theo quy tắc $y_i = e_k(y_{i-1} \oplus x_i)$. Việc sử dụng chế độ CBC được mô tả trên Hình 2.16.



Hình 2.16. Chế độ CBC

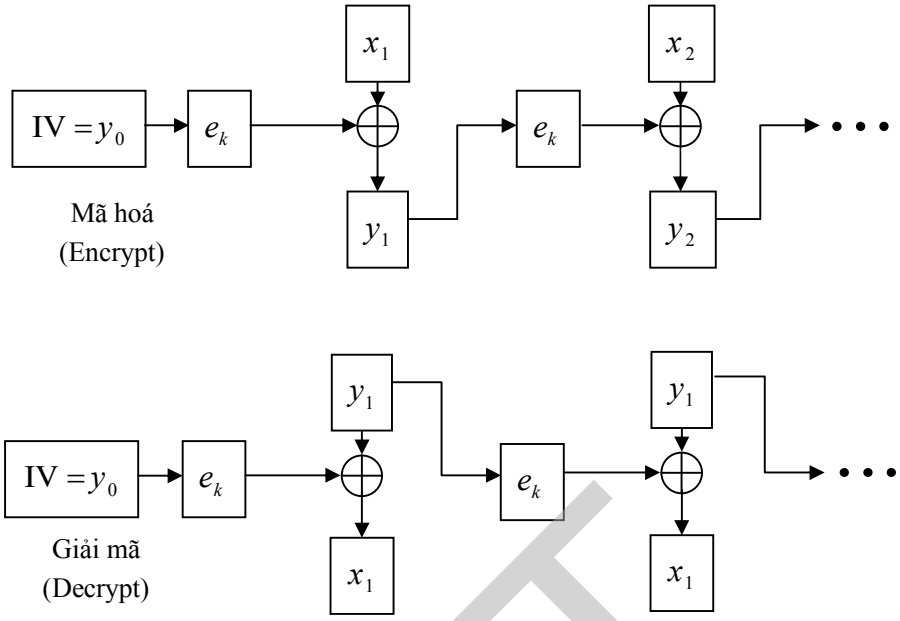
Trong các chế độ OFB và CFB dòng khoá được tạo ra sẽ được cộng mod 2 với bản rõ (tức là nó hoạt động như một hệ mã dòng, xem phần 3.8). OFB thực sự là một hệ mã dòng đồng bộ: dòng khoá được tạo bởi việc mã lặp vector khởi tạo 64 bit (vector IV). Ta xác định $z_0 = \text{IV}$ và rồi tính dòng khoá $z_1, z_2, \ldots$ theo quy tắc $z_i = e_k(z_{i-1}), i \geq 1$. Dãy bản rõ $x_1, x_2, \ldots$ sau đó sẽ được mã hoá bằng cách tính $y_i = x_i \oplus z_i, i \geq 1$.

Trong chế độ CFB, ta bắt đầu với $y_0 = \text{IV}$ (là một vector khởi tạo 64 bit) và tạo phần tử z_i của dòng khoá bằng cách mã hoá khối bản mã trước đó. Tức $z_i = e_k(y_{i-1}), i \geq 1$. Cũng như trong chế độ OFB: $y_i = x_i \oplus z_i, \ i \geq 1$. Việc sử dụng CFB được mô tả trên Hình 2.17 (chú ý rằng hàm mã DES e_k được dùng cho cả phép mã và phép giải mã ở các chế độ CFB và OFB).

Cũng còn một số biến thể của OFB và CFB được gọi là các chế độ phản hồi k bit $(1 < k < 64)$. Ở đây, ta đã mô tả các chế độ phản hồi 64 bit. Các chế độ phản hồi 1 bit và 8 bit thường được dùng trong thực tế cho phép mã hoá đồng thời 1 bit (hoặc byte) số liệu.

Bốn chế độ công tác có những ưu, nhược điểm khác nhau. Ở chế độ ECB và OFB, sự thay đổi của một khối bản rõ x_i 64 bit sẽ làm thay đổi khối bản mã y_i tương ứng, nhưng các

khối bản mã khác không bị ảnh hưởng. Trong một số tình huống, đây là một tính chất đáng mong muốn. Ví dụ, chế độ OFB thường được dùng để mã khi truyền vệ tinh.



Hình 2.17. Chế độ CFB

Mặt khác ở các chế độ CBC và CFB, nếu một khối bản rõ x_i bị thay đổi thì y_i và tất cả các khối bản mã tiếp theo sẽ bị ảnh hưởng. Như vậy các chế độ CBC và CFB có thể được sử dụng rất hiệu quả cho mục đích xác thực. Đặc biệt hơn, các chế độ này có thể được dùng để tạo mã xác thực bản tin (MAC - message authentication code). MAC được gắn thêm vào các khối bản rõ để thuyết phục Bob tin rằng, dãy bản rõ đó thực sự là của Alice mà không bị Oscar giả mạo. Như vậy MAC đảm bảo tính toàn vẹn (hay tính xác thực) của một bản tin (nhưng tất nhiên là MAC không đảm bảo độ mật).

Ta sẽ mô tả cách sử dụng chế độ BCB để tạo ra một MAC. Ta bắt đầu bằng vector khởi tạo IV chứa toàn số 0. Sau đó dùng chế độ CBC để tạo các khối bản mã $y_1,...,y_n$ theo khoá K . Cuối cùng ta xác định MAC là y_n . Alice sẽ phát đi dãy các khối bản rõ $x_1,...,x_n$ cùng với MAC. Khi Bob thu được $x_1,...,x_n$ anh ta sẽ khôi phục lại $y_1,...,y_n$ bằng khoá K bí mật và xác minh xem liệu y_n có giống với MAC mà mình đã thu được hay không?

Nhận thấy Oscar không thể tạo ra một MAC hợp lệ do anh ta không biết khoá K mà Alice và Bob đang dùng. Hơn nữa Oscar thu chặn được dãy khối bản rõ $x_1,...,x_n$ và thay đổi ít nhiều nội dung thì thì chắc chắn là Oscar không thể thay đổi MAC để được Bob chấp nhận.

Thông thường ta muốn kết hợp cả tính xác thực lẫn độ bảo mật. Điều đó có thể thực hiện như sau: Trước tiên Alice dùng khoá K_1 để tạo MAC cho $x_1,...,x_n$. Sau đó Alice xác định x_{n+1} là MAC rồi mã hoá dãy $x_1,...,x_{n+1}$ bằng khoá thứ hai K_2 để tạo ra bản mã $y_1,...,y_{n+1}$. Khi Bob thu được $y_1,...,y_{n+1}$, trước tiên Bob sẽ giải mã (bằng K_2) và kiểm tra xem x_{n+1} có phải là MAC đối với dãy $x_1,...,x_n$ dùng K_1 hay không.

Ngược lại, Alice có thể dùng K_1 để mã hoá $x_1,...,x_n$ và tạo ra được $y_1,...,y_n$, sau đó dùng K_2 để tạo MAC y_{n+1} đối với dãy $y_1,...,y_n$. Bob sẽ dùng K_2 để xác minh MAC và dùng K_1 để giải mã $y_1,...,y_n$.

2.8.4.2. Mã nguồn DES (Xem phụ lục 1)

2.8.5. Chuẩn mã dữ liệu tiên tiến (AES)

Vào 1997, Viện tiêu chuẩn và công nghệ quốc gia (NIST) Của Mỹ đã phát động cuộc thi nhằm xây dựng một chuẩn mã dữ liệu mới thay thế cho chuẩn mã dữ liệu cũ DES đã được đưa ra năm 1974. Qua quá trình tuyển chọn vào tháng 10 năm 2000, NIST đã công bố chuẩn mã dữ liệu mới được lựa chọn là thuật toán Rijndael. Đây là một mật mã khối đối xứng với ba kích thước khóa có thể lựa chọn (128 bit, 192 bit và 256 bit). Sau đây ta sẽ mô tả thuật toán AES này.

2.8.5.1. Cơ sở toán học của AES

Trong AES các phép toán cộng và nhân được thực hiện trên các byte trong trường hữu hạn $\mathrm{GF}(2^8)$.

Phép cộng:

Phép cộng giữa hai phần tử (các byte) trong trường hữu hạn được thực hiện bằng cách cộng theo modulo 2 các bit tương ứng trong biểu diễn của các byte này. Phép cộng các byte A và B với:

$$A = (a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8)$$
$$B = (b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6 \quad b_7 \quad b_8)$$

là $C = A + B$ với $C = (c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7 \quad c_8)$

trong đó $C_i = a_i + b_i \bmod 2$ với $i = \overline{1,8}$

Các phần tử của trường hữu hạn còn có thể được biểu diễn dưới dạng đa thức. Ví dụ tổng của $A = 73_H$ và $B = 4E_H$ (viết dưới dạng cơ số 16 - hexa) là:

$$73_H + 4E_H = 3D_H$$

Viết dưới dạng nhị phân:

$$01110011 + 01001110 = 00111101$$

Viết dưới dạng đa thức:

$$(x^6 + x^5 + x^4 + x + 1) + (x^6 + x^3 + x^2 + x) = (x^5 + x^4 + x^3 + x^2 + 1)$$

Phép nhân:

Phép nhân được thực hiện trên $GF(2^8)$ bằng cách nhân hai đa thức rút gọn theo modulo của một đa thức bất khả quy $m(x)$.

Trong AES đa thức bất khả quy này là $m(x) = x^8 + x^4 + x^3 + x + 1$

Ví dụ 2.18: $A = \text{C3}_H$, $B = 85_H$ tương ứng với:

$$a(x) = x^7 + x^6 + x + 1 \text{ và } b(x) = x^7 + x^2 + 1$$

Khi đó C = A.B

$$c(x) = a(x)\,b(x) \bmod \left(x^8 + x^4 + x^3 + x + 1\right)$$

$$c(x) = x^7 + x^5 + x^3 + x^2 + x$$

hay $C = AE_H = 10101110$

2.8.5.2. Thuật toán AES

AES mã hóa một khối bản rõ M 128 bit thành một khối bản mã C 128 bit bằng cách dùng một khóa mã K có độ dài 128 bit (hoặc 192 hoặc 256 bit) tương ứng với $AES-128$ (hoặc $AES-192$ hoặc $AES-256$). Thuật toán thực hiện trên các byte và kích thước khối đối với đầu vào đầu ra và khóa được biểu thị bằng các từ 32 bit (4 byte).

AES sẽ thực hiện một số vòng mã hóa N_r phụ thuộc vào độ dài khóa được sử dụng (Bảng 2.10)

Bảng 2.10. Số các vòng mã hóa của AES

Thuật toán AES	Độ dài đầu vào/đầu ra	Độ dài khóa N_k	Số vòng N_r
$AES-128$	4 từ	4 từ	10 vòng
$AES-192$	4 từ	6 từ	12 vòng
$AES-256$	4 từ	8 từ	14 vòng

Mã hóa AES:

Mõi vòng gồm 4 phép biến đổi mật mã theo byte

- Thay thế byte
- Dịch các hàng của mảng trạng thái (State Array)
- Trộn dữ liệu trong một cột của State Array
- Cộng khóa vòng vào State Array

Phép thay thế byte: SubBytes( )

Phép biến đổi AES đầu tiên là một phép thay thế byte phi tuyến gọi là phép biến đổi SubBytes( ), nó hoạt động độc lập trên mỗi byte. Trước tiên nó sẽ tính nghịch đảo của phép nhân trong $GF\left(2^8\right)$, sau đó sử dụng một phép biến đổi trên nghịch đảo này.

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

trong đó b_i biểu thị bit thứ i của byte b

Dịch các hàng của State Array; Phép biến đổi ShiftRows( )

Phép biến đổi tiếp theo của AES là dịch các hàng của State Array. Lượng dịch $\text{Shift}(r, N_b)$ phụ thuộc vào số hàng r. Các khối đầu vào (bản rõ) vào các khoái đầu ra (bản mã) là các khối 128 bit gồm $N_b = 4$ từ 32 bit

Phép biến đổi ShiftRows( ) được biểu thị như sau:

$$s'_{r,c} = s_r\left(c + shift\left(r, N_b\right)\right) \bmod N_b$$

trong đó $0 \le c \le N_b$

Hàng đầu tiên sẽ không dịch, tức là $\text{shift}(0, N_b = 4) = 0$

Với các hàng còn lại lượng dịch sẽ tùy theo số hàng

$$\text{shift}(0,4) = 0$$
$$\text{shift}(1,4) = 1$$
$$\text{shift}(2,4) = 2$$
$$\text{shift}(3,4) = 3$$

Trộn dữ liệu trong một cột State Array: Phép biến đổi Mixcolumns( )

Phép biến đổi Mixcolumns( ) được dùng để trộn dữ liệu trong một cột của ma trận trạng thái. Các cột được xem như các đa thức trong $GF\left(2^8\right)$. Đầu ra của Mixcolumns( ) là $s'(x)$ được tạo bằng cách nhân cột với $s(x)$ với đa thức $a(x)$ và rút gọn theo $\bmod\left(X^4 + 1\right)$

$$s'(x) = a(x)\, s(x) \bmod\left(x^4 + 1\right)$$

trong đó: $a(x) = 03_H x^3 + 01_H x + 02_H$

Ở dạng ma trận phép biến đổi này có thể viết như sau:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02_H & 03_H & 01_H & 01_H \\ 01_H & 02_H & 03_H & 01_H \\ 01_H & 01_H & 02_H & 03_H \\ 03_H & 01_H & 01_H & 02_H \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Ở đây $0 \le c < N_b$

Mở rộng khóa AES: KeyExpansion( )

Thuật toán AES sẽ tạo từ khóa mã 128 bit (hoặc 192 hoặc 256 bit) một tập khởi tạo N_b từ 32 bit và N_b từ 32 bit cho mỗi vòng bao gồm $N_b\left(N_r + 1\right)$ từ 32 bit. Chương trình giải mã KeyExpansion( ) chứa các SubWord( ) và RotWord( ).

Hàm SubWord( ) là một phép thay thế (hộp S) một từ vào 4 byte bằng một từ ra 4 byte.

Hàm RotWord( ) thực hiện phép hoán vị vòng các byte trong một từ 4 byte (32 bit) W_i:

$$RotWord(a_0,a_1,a_2,a_3)=(a_1,a_2,a_3,a_0)$$

KeyExpansion $\left(byte\,key\left[4_*N_k\right],word\,w\left[N_{b*}\left(N_r+1\right)\right],N_k\right)$

Begin

$i=0$

while $\left(i<N_k\right)$

$w[i]=word\left[key\left[4_*i\right],key\left[4_*i+1\right],key\left[4_*i+2\right],key\left[4_*i+3\right]\right]$

$i=i+1$

end while

$i \square N_k$

while $\left(i<N_{b*}\left(N_r+1\right)\right)$

word $temp=w[i-1]$

if $\left(i \bmod N_k=0\right)$

$temp=\text{SubWord}\left(\text{RotWord}\left(temp\right)\right)xor\ Rconw\left[i/N_k\right]$

else if $\left(N_k=8\ and\ i \bmod N_k=4\right)$

$temp=\text{SubWord}\left(temp\right)$

end if

$w[i] \square w\left[i-N_k\right]=xor\ temp$

$i=i+1$

end while

end

Chương trình giải mã của AES

Cipher $\left(byte\,in\left[4_*N_b\right],byte\,out\left[4_*N_b\right],word\,w\left[N_{b*}\left(N_r+1\right)\right]\right)$

Begin byte state $\left[4,N_b\right]$ state = in AddRoundKey(state,w)

for round = 1 step 1 to N_r-1

SubBytes (state), ShifRows (state),

Mixcolumns(state), AddRoundKey(state,w+round $_*N_b$)

end for

SubBytes (state), ShifRows (state)

AddRoundKey(state,w+$N_{r*}N_b$)

out = state

end

2.9. ƯU VÀ NHƯỢC ĐIỂM CỦA MẬT MÃ KHÓA BÍ MẬT

2.9.1. Ưu điểm

- Mật mã khóa bí mật (mật mã cổ điển) nói chung đơn giản, tức là các yêu cầu về phần cứng không phức tạp, thời gian tính toán nhanh.
- Mật mã khóa bí mật có tính hiệu quả, thông thường tốc độ mã $R_{mã} = 1$ (số bit đầu ra mã hóa bằng với số bit đầu vào).

Từ các ưu điểm này cho thấy mật mã cổ điển dễ sử dụng cho các dịch vụ nhạy cảm với độ trễ và các dịch vụ di động.

2.9.2. Nhược điểm

- Với mật mã khóa bí mật phải dùng kênh an toàn để truyền khóa, điều này dẫn đến chi phi sẽ cao hơn và việc thiết lập kênh an toàn khó khăn hơn.
- Việc tạo khóa và giữ bí mật khóa phức tạp. Khi làm việc trên mạng nếu dùng mật mã khóa bí mật sẽ phải tạo và lư trữ số lượng khóa nhiều.
- Nếu sử dụng mật mã khóa bí mật sẽ khó xây dựng các dịch vụ an toàn khác như các dịch vụ đảm bảo tính toàn vẹn của dữ liệu, dịch vụ xác thực và chữ ký số. Các dịch vụ này sẽ được thực hiện bởi mật mã khóa công khai.

BÀI TẬP CHƯƠNG 2

Bài 2.1: Thám mã thu được bản mã sau:

PSZI QIERW RIZIV LEZMRK XS WEC CSY EVI WSVVC

Biết rằng đây là bản mã của mật Caesar với khoá k chưa biết. Hãy dùng phương pháp tìm khoá vét cạn để tìn được bản rõ tiếng Anh tương ứng.

Ghi chú: Phương pháp tìm khoá vét cạn là phương pháp thử giải mã bằng mọi khoá có thể có.

Bài 2.2: Thám mã thu được bản mã sau:

_EHOHWSI_ON_E_TREVADYC_YQNOREUGNIOS_ EMAEFH
R_SATONEL_NRA DEEHTES_ERCO_TL_FEFI

Hãy chỉ ra rằng đây là một hệ mật hoán vị và thực hiện thám mã bằng phương pháp tìm khoá vét cạn. (Ký hiệu (_) là khoảng trắng (space)).

Bài 2.3: Thực hiện thám mã các bản mã sau của một hệ mật mã dịch vòng với khoá k chưa biết, bằng các phương pháp tìm khóa vét cạn và Thống kê, biết rằng các ký tự có xác suất xuất hiện lớn trong tiếng Anh được sắp xếp theo thứ tự sau:

_,E,T,A,H,O,N

Với giả sử ‘‘khoảng trống’’ (_) được xem là 1 ký tự

a) XMQIDMWDQSVIDZEPYEFPIDXLERDQSRIBDBSYDGERDKIXDQ
SVIDQSRIBDFYXDBSYDGERRSXDKIXDQSVIDXMQI

b) YMJEKTTQNXMERFSEXJJPXEMFUUNSJXXENSEYMJEI
NXYFSHJEYMJEANXJELWTAXENYEZSIJWEMNXEKJJY

c) ZNKFZX_KFYOMTFULFOTZKRROMKTIKFOYFTUZFQTUBRKJMKFH_ZFOSG
MOTGZOUT

d) IDRIZIVDORS_DXLIDPSZIDSJDSYVDTEVIRXWDJSVDYWDXM
PPD_IDLEZIDFIGSQIDTEVIRXW

Bài 2.4: Dưới đây là 4 bản mã thu được từ mã thay thế. Một bản thu được từ mã thay thế, một từ mã Vigenère, một từ mật mã Affine và một bản chưa xác định. Nhiệm vụ ở đây là xác định bản rõ trong mỗi trường hợp.

Hãy mô tả các bước cần thực hiện để giải mã mỗi bản mã (bao gồm tất cả các phân tích thống kê và các tính toán cần thực hiện).

Hai bản rõ đầu lấy từ cuốn "**The Diary of Samuel Marchbanks**" của Robertson Davies, Clack Iriwin,1947; bản rõ thứ tư lấy từ "**Lake Wobegon Days**" của Garrison Keillor, Viking Penguin, 1985.

a) Mã thay thế.

EMGLOSUDCGDNCUSWYSFHNSFCYKDPUMLWGYICOXYSIPJCK

QPKUGKMGOUCGINCGACKSNISACYKZSCKXEOCKSHYSXCG

OIDPKZCNKSHICGIWYGKKGKGOLDSILKGOIUSIGLEDSPWZU

GFZCCNDGYYSFUSZCNXEOJNCGYEOWEUPXEZGACGNFGLKNS

ACIGOIYCKXOUOUZCFZCCNDGYYSFEUEKUZCSOCFZCCNC

IACZEJNCSHFZEJZEGMXCYHCIUMGKUSY

Chỉ dẫn: **F** sẽ giải mã thành **w**.

b) Hệ mã Vigenère

KCCPKBGUFDPHQTYAVINRRTMVGRKDNBVFĐETDGILTXRGUD

DKOTFMBPVGEGLTGCKQRACQCWDNAWCRXLZAKFTLEWRPTVC

QKYVXCHKFTPONCQQRHJVAJUWETMCMSPKQDYHJVDAHCTRL

SVSKCGCZQọDZXGSFRLSWCWSJTBHAFSLASPRJAHKJRJUMV

GKMITZHFPDLSPZLVLGWTFPLKKEBDPGCEBSHCTJRWXBAFS

PEZQNRWXCVYCGAONWDDKACKAWBBIKFTLOVKCGGHJVLNHI

FFSQESVYCLACNVRWBBIREPBBVFEXOSCDYGZWPFDTKFQLY

CWHJVTNHIQ/BTKH/VNPIST

c) Hệ mã Affine.

KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP

KRLOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABĐKP

BCPFEQPKAZBKRHALBKAPCCIBURCCDKDCCJC/DFUIXPAFF

ERBICZDFKABICBBENEFCUPLCVKABPCYDCCDPKBCOCPERK

IVKSCPICBRKLJPKABL

d) Hệ mã chưa xác định được.

BNVSNSIHQCEELSSKKYERIFJKXUMBGVKAMQLJTYAVFBKVT

DVBPVVRJYYLAOKYMPQSCGDLFSRLLPROYGESEBUUALRWXM

MASAZLGLEĐFJBZAVVPXWI CGJXASCBYEHOSNMULKCEAHTQ

OKMFLEBKFXLRRFDTZXCIWBJSICBGAWDVYDHAVFJXZIBKC

GJIWEAHTTOEWTUHKRQVVRGZBXYIREMMASCSPBNLHJMBLR

FFJELHWEYLWISTFVVYFJCMHYUYRUFSFMGESIGRLWALSVVM

NUHSIMYYITCCQPZSICEHBCCMZFEGVJYOCDEMMPGHVAAUM

ELCMOEHVLTIPSUYILVGFLMVWDVYDBTHFRAYISYSGKVSUU

HYHGGCKTMBLRX

Bài 2.5:

a) Có bao nhiêu ma trận khả nghịch cấp 2×2 trên $\mathbf{Z}_{26}$?

b) Giả sử p là số nguyên tố. Hãy chứng tỏ số các ma trận khả nghịch cấp 2×2 trên $\mathbf{Z}_p$ là $\left(p^2-1\right)\left(p^2-p\right)$.

Chỉ dẫn Vì p là số nguyên tố nên $\mathbf{Z}_p$ là một trường. Hãy sử dụng khẳng định sau: Một ma trận trên một trường là khả nghịch khi và chỉ khi các hàng của nó là các véc tơ độc lập tuyến tính (tức không tồn tại một tổ hợp tuyến tính các hàng khác 0 mà tổng của chúng là một véc tơ toàn số 0).

c) Với p là số nguyên tố và m là một số nguyên $m\geq 2$. Hãy tìm công thức tính số các ma trận khả nghịch cấp $m\times m$ trên $\mathbf{Z}_p$.

Bài 2.6: Giả sử ta đã biết rằng bản rõ "**conversation**" sẽ tạo nên bản mã "**HIARRTNUYTUS**" (được mã theo hệ mã Hill nhưng chưa xác định được m). Hãy xác định ma trận mã hoá.

Bài 2.6: Hệ mã Affine - Hill là hệ mã Hill được sửa đổi như sau: Giả sử m là một số nguyên dương và $P=C=\left(\mathbf{Z}_{26}\right)^m$. Trong hệ mật này, khoá K gồm các cặp $\left(L,b\right)$, trong đó L là một ma trận khả nghịch cấp $m\times m$ trên $\mathbf{Z}_{26}$ và $b\in\left(\mathbf{Z}_{26}\right)^m$ theo công thức $y=xL+b$. Bởi vậy, nếu $L=\left(l_{ij}\right)$ và $b=\left(b_1,...,b_m\right)$ thì:

$$\left(y_1,...,y_m\right)=\left(x_1,...,x_m\right)\begin{pmatrix} l_{1,1} & l_{1,2} & \dots & l_{1,m} \\ l_{2,1} & l_{2,2} & \dots & l_{2,m} \\ \vdots & \vdots & & \vdots \\ l_{m,1} & l_{m,2} & \dots & l_{m,m} \end{pmatrix}+\left(b_1,...,b_m\right)$$

Giả sử Oscar đã biết bản rõ 1à ***"adisplayedequation"*** và bản mã tương ứng là "**DSRMSIOPLXLJBZULLM**". Oscar cũng biết $m=3$. Hãy tính khoá và chỉ ra tất cả các tính toán cần thiết?

Bài 2.7: Sau đây là cách thám mã hệ mã Hill sử dụng phương pháp tấn công chỉ với bản mã. Giả sử ta biết $m=2$. Chia các bản mã thành các khối có độ dài 2 kí tự (các bộ đôi). Mỗi bộ đôi này là bản mã của một bộ đôi của bản rõ nhờ dùng một ma trận mã hoá chưa biết. Hãy nhặt ra các bộ đôi thường gặp nhất trong bản mã và coi rằng đó là mã của một bộ đôi thường gặp trong danh sách ở bảng 1.1 (ví dụ TH và ST). Với mỗi giả định, hãy thực hiện phép tấn công với bản rõ đã biết cho tới. khi tìm được ma trận giải mã đúng.

Sau đây là một ví dụ về bản mã để bạn giải mã theo phương pháp đã nêu:

LMQETXYEAGTXCTUIEWNCTXLZEWUAISPZYVAPEWLMGQWVA

XFTGMSQCADAGTXLMDXNXSNPJQSYVAPRIQSMHNOCVAXFV.

Bài 2.8: Ta sẽ mô tả một trường hợp đặc biệt của mã hoán vị. Giả sử m, n là các số nguyên dương. Hãy viết bản rõ theo thành từng hàng thành một hình chữ nhật $m \times n$. Sau đó tạo ra bản mã bằng cách lấy các cột của hình chữ nhật này Ví dụ, nếu $m = 4, n = 3$ thì ta sẽ mã hoá bản rõ "**cryptography**" bằng cách xây dựng hình chữ nhật :

cryp

togr

aphy

Bản mã sẽ là: "**CTAROPYGHPRY**"

a) Hãy mô tả cách Bob giải mã một bản mã (với m, n đã biết).

b) Hãy giải mã bản mã sau: (nhận được theo phương pháp đã nêu):

MYAMRARUYIQTENCTORAHROYWĐSOYEOUARRGĐERNOGW

Bài 2.9: Hãy chứng minh rằng phép giải mã DES có thể thực hiện bằng cách áp dụng thuật toán mã hoá DES cho bản rõ với bảng khoá đảo ngược.

Bài 2.10: Cho $DES(x, K)$ là phép mã hoá DES của bản rõ x với khoá K. Giả sử $y = DES(x, K)$ và $y' = DES\left(c(x), c(K)\right)$ trong đó $c(.)$ kí hiệu là phần bù theo các bit của biến. Hãy chứng minh rằng $y = c(y)$ (tức là nếu lấy phần bù của bản rõ và khoá thì bản mã kết quả cũng là phần bù của bản mã ban đầu). Chú ý rằng kết quả trên có thể chứng minh được chỉ bằng cách sử dụng mô tả "mức cao" của DES - cấu trúc thực tế của các hộp S và các thành phần khác của hệ thống không ảnh hưởng tới kết quả này.

Bài 2.11: Mã kép là một cách để làm mạnh thêm cho DES: với hai khóa K_1 và K_2 cho trước, ta xác định $y = e_{K_2}(e_{K_1}(x))$ (dĩ nhiên đây chính là tích của DES với chính nó). Nếu hàm mã hoá e_{K_2} giống như hàm giải mã d_{K_1} thì K_1 và K_2 được gọi là các khoá đối ngẫu (đây là trường hợp không mong muốn đối với phép mã kép vì bản mã kết quả lại trùng với bản rõ). Một khoá được gọi là tự đối ngẫu nếu nó đối ngẫu với chính nó.

a) Hãy chứng minh rằng nếu C_0 gồm toàn các số 0 hoặc gồm toàn các số 1 và D_0 cũng vậy thì K là tự đối ngẫu.

b) Hãy tự chứng minh rằng các khoá sau (cho ở dạng hexa) là tự đối ngẫu:

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

F E E F E F E F E F E F E F E F

1 F 1 F 1 F 1 F 0 F 0 F 0 F 0 F

E 0 E 0 E 0 E 0 F 1 F 1 F 1 F 1

c) Hãy chứng tỏ rằng nếu $C_0 = 0101...01$ hoặc $1010...10$ (ở dạng nhị phân) thì XOR các xâu bit C_i và C_{17-i} là $111...11$, với $1 \le i \le 16$ (khẳng định tương tự cũng đúng đối với D_i).

d) Hãy chứng tỏ các cặp khoá sau là đối ngẫu:

E001E001F101F101

FE1FFE1FF0EFE0E

E01FE01FFF10FF10

01E001E001F101F1

1FEFE1FFE0EFE0EFE

1FE01FE00EF10EF1