BAN CƠ YẾU CHÍNH PHỦ

HỌC VIỆN KỸ THUẬT MẬT MÃ

GS. NGUYỄN BÌNH, ThS. HOÀNG THU PHƯƠNG

GIÁO TRÌNH

CƠ SỞ LÍ THUYẾT MẬT MÃ

HÀ NỘI, 2013

BAN CƠ YẾU CHÍNH PHỦ

HỌC VIỆN KỸ THUẬT MẬT MÃ

GS. NGUYỄN BÌNH, ThS. HOÀNG THU PHƯƠNG

GIÁO TRÌNH

CƠ SỞ LÍ THUYẾT MẬT MÃ

HÀ NỘI, 2013

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

AES	Advanced Encryption Standard	Chuẩn mã dữ liệu tiên tiến
CBC	Cipher Block Chaining	Chế độ liên kết khối mã
CFB	Cipher Feedback	Chế độ phản hồi mã
CRHF	Collission Resistant Hash Function	Hàm băm kháng va chạm
DES	Data Encryption Standard	Chuẩn mã dữ liệu
DSS	Digital Signature Standard	Chuẩn chữ kí số
ECB	Electronic Code Book	Chế độ quyển mã điện tử
LAN	Local Area Network	Mạng cục bộ
LFSR	Linear Feedback Sequence Register	Thanh ghi hồi tiếp tuyến tính
LSB	Least Signification Bit	Bít thấp nhất (có giá trị nhỏ nhất)
MAC	Massage Authentication Code	Mã xác thực thông báo
MDC	Manipulation Detection Code	Mã phát hiện sự sửa đổi
MDV		Mã dịch vòng
MHV		Mã hoán vị
MTT		Mã thay thế
OWHF	One Way Hash Function	Hàm băm một chiều.
OTP	One Time Pad	Hệ mật khóa dùng một lần
RSA	Rivest – Shamir - Adleman	Thuật toán RSA

DANH MỤC BẢNG

DANH MỤC HÌNH VẼ

LỜI NÓI ĐẦU

Đảm bảo an toàn là một trong những chỉ tiêu chất lượng cơ bản của hệ thống truyền tin số. Ngoài việc đảm bảo hệ thống là khả dụng (có đủ tài nguyên cần thiết cho dịch vụ tương ứng) có ba loại dịch vụ chính phải thực hiện:

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

Các dịch vụ này được thực hiện thông qua việc kết hợp các thuật toán cơ bản trong mật mã học. Giáo trình này là một giáo trình cơ sở giúp cho sinh viên bước đầu tìm hiểu các vấn đề và các thuật toán cơ bản trong mật mã học nhằm thực hiện các dịch vụ trên.

Nội dung giáo trình bao gồm 4 chương:

Chương 1: Nhập môn mật mã học: Trình bày một số khái niệm, định nghĩa cơ bản và cơ sở lý thuyết thông tin áp dụng cho các hệ mật

Chương II: Mật mã khóa bí mật: Trình bày các thuật toán mật mã khoá bí mật bao gồm các thuật toán hoán vị, thay thế và các thuật toán kết hợp mà chủ yếu là DES và AES.

Chương III: Mật mã khóa công khai: Trình bày các thuật toán cơ bản trong mật mã khóa công khai bao gồm các các hệ mật RSA, Merkle-Hellman, Rabin, ElGamal, hệ mật trên đường cong Elliptic và hệ mật McEliece.

Chương IV: Hàm băm, xác thực và chữ ký số: Trình bày khái niệm hàm băm các ứng dụng trong việc xác thực và đảm bảo tính toàn vẹn của dữ liệu.

Sau mỗi chương đều có các bài tập nhằm giúp cho sinh viên có thể nắm, hiểu cụ thể và sâu sắc hơn các vấn đề lý thuyết được trình bày.

Với kinh nghiệm và thời gian hạn chế, việc chọn lọc và trình bày các vấn đề không thể tránh khỏi các thiếu sót nhất định. Rất mong nhận được các ý kiến đóng góp quý báu của độc giả.

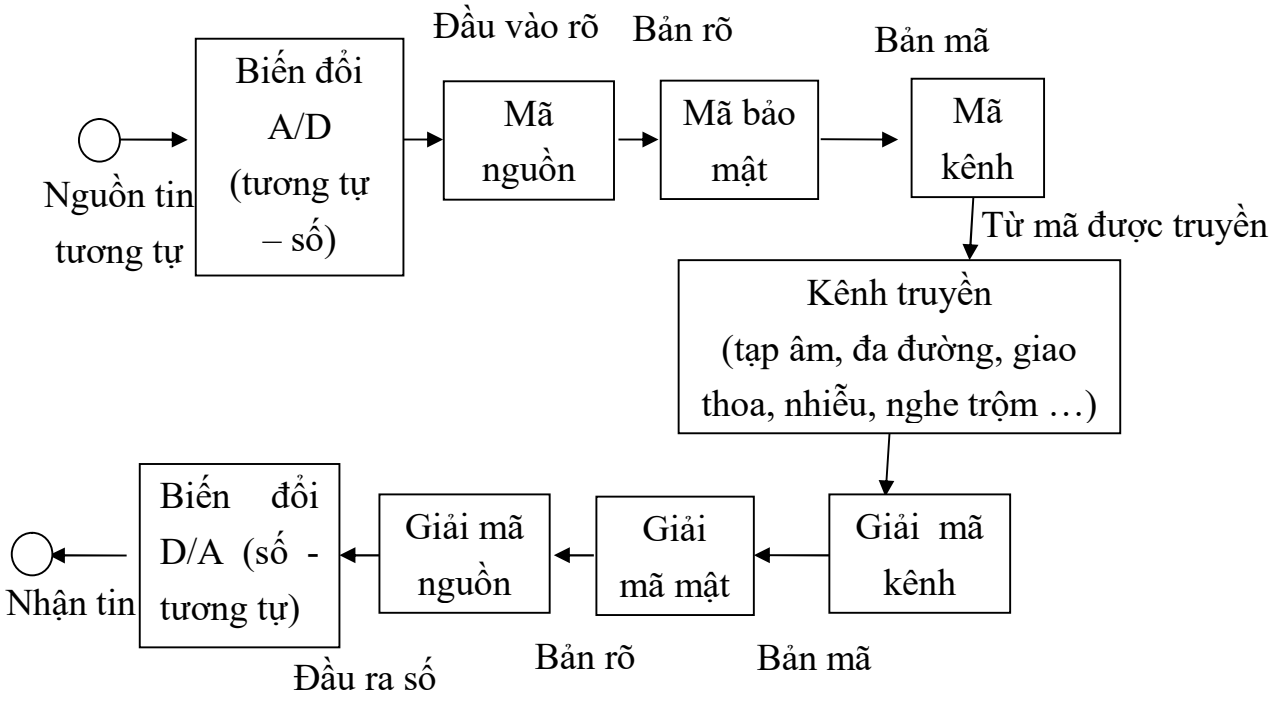
CÁC TÁC GIẢ

GS. TS NGUYỄN BÌNH

ThS. HOÀNG THU PHƯƠNG

CHƯƠNG 1
NHẬP MÔN MẬT MÃ HỌC

1.1. SƠ ĐỒ KHỐI CHỨC NĂNG CỦA MỘT HỆ THỐNG THÔNG TIN SỐ.



Hình 1-1. Sơ đồ khối của một 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. CÁC CHỈ TIÊU CHẤT LƯỢNG CƠ BẢN CỦA MỘT HỆ THỐNG THÔNG TIN SỐ

a) Tính hữu hiệu

Thể hiện trên các mặt sau:

- Tốc độ truyền tin cao
- Truyền được đồng thời nhiều tin khác nhau
- Chi phí cho một bit thấp

b) Độ tin cậy

Đảm bảo độ chính xác của việc thu nhận tin cao, xác suất thu sai (BER) thấp.

Hai chỉ tiêu trên mâu thuẫn nhau. Giải quyết mâu thuẫn trên là nhiệm vụ của lý thuyết thông tin

c) An toàn

- Bí mật:
  + Không thể khai thác thông tin trái phép
  + Chỉ có người nhận hợp lệ mới hiểu được thông tin
- Xác thực: Gắn trách nhiệm của bên gửi – bên nhận với bản tin (chữ ký số)

- Toàn vẹn

+ Thông tin không bị bóp méo (cắt xén, xuyên tạc, sửa đổi)

+ Thông tin được nhận phải nguyên vẹn cả về nội dung và hình thức

- Khả dụng: mọi tài nguyên và dịch vụ của hệ thống phải được cung cấp đầy đủ cho người dùng hợp pháp

d) Đảm bảo chất lượng dịch vụ (QoS)

Đây là một chỉ tiêu rất quan trọng đặc biệt là đối với các dịch vụ thời gian thực, nhạy cảm với độ trễ (truyền tiếng nói, hình ảnh,…)

1.3. ĐỘ MẬT HOÀN THIỆN

Năm 1949, 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 sẽ trình bày một vài ý tưởng trong lý thuyết của Shannon.

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

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 đến một bài toán khác chứ không phải là một chứng minh hoàn chỉnh về độ an toàn (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).

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(người nhận-giải mã) 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 sau 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 cơ bản trong xác suất; các định nghĩa chính sẽ được nêu dưới đây.

Định nghĩa 1.1.

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:

$$p(x,y) = p(x|y)\,p(y) \qquad (1.1)$$

Đổi chỗ x và y ta có :

$$p(x,y) = p(y|x)\,p(x) \qquad (1.2)$$

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: $p(x|y) = \dfrac{p(x)p(y|x)}{p(y)}$ (1.3)

Hệ quả 1.1.

X và Y là các biến độc lập khi và chỉ khi: $p(x|y) = p(x)$ với mọi 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 P. Kí hiệu xác suất tiên nghiệm để bản rõ xuất hiện là p_P (x). Cũng giả sử rằng, khóa K được chọn (bởi Alice(bên gửi-mã hóa) và Bob(bên nhận-giải mã)) 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 $p_K(K)$. 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 P và K sẽ tạo ra một phân bố xác suất trên C. Thật vậy, có thể dễ dàng tính được xác suất $p_C(y)$ với y là bản mã được gửi đi. Với một khoá $K \in \mathsf{K}$, ta xác định:

$$C(K) = \{e_K(x) : x \in \mathcal{P}\} \qquad (1.4)$$

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

$$p_C(y) = \sum_{\{K:y\in C(K)\}} p_{\mathcal{K}}(K)\, p_{\mathcal{P}}(d_K(y)) \qquad (1.5)$$

Nhận thấy rằng, với bất kì $y \in \mathsf{C}$ và $x \in \mathsf{P}$, 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):

$$p_C(y|x) = \sum_{\{K:x=d_K(y)\}} p_{\mathcal{K}}(K) \qquad (1.6)$$

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:

$$p_P(x|y) = \frac{p_P(x).\sum_{\{K:x=d_K(y)\}} p_{\mathcal{K}}(K)}{\sum_{\{K:y\in C(K)\}} p_{\mathcal{K}}(K)\, p_P(d_K(y))} \qquad (1.7)$$

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 1.1.

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 $e_{K1}(a) = 1$, $e_{K1}(b) = 2$, $e_{K2}(a) = 2$, $e_{K2}(b) = 3$, $e_{K3}(a) = 3$, $e_{K3}(b) = 4$. Hệ mật này được biểu thị bằng ma trận mã hoá sau:

	a	b
K_1	1	2
K_2	2	3
K_3	3	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 :

$p_P(a \mid 1) = 1$ $p_P(b \mid 1) = 0$ $p_P(a \mid 2) = 1/7$ $p_P(b \mid 2) = 6/7$

$p_P(a \mid 3) = 1/4$ $p_P(b \mid 3) = 3/4$ $p_P(a \mid 4) = 0$ $p_P(b \mid 4) = 1$

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 1.2.

Một hệ mật có độ mật hoàn thiện nếu $p_{\mathcal{P}}(x|y) = p_{\mathcal{P}}(x)$ *với mọi* $x \in P$, $y \in C$. *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, 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 $P = C = K = Z_{26}$ và với $0 \le K \le 25$, quy tắc mã hoá e_K là $e_K(x) = x + K \bmod 26$ ($x \in 26$). Trước tiên tính phân bố P_C . Giả sử $y \in Z_{26}$, khi đó:

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

Xét thấy với y cố định, các giá trị $y - K \bmod 26$ sẽ tạo thành một hoán vị của Z_{26} và p_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:

$$p_C(y|x) = p_{\mathcal{K}}(y - x \bmod 26) = 1/26$$

Với mọi x,y vì với mỗi cặp x,y, khóa duy nhất K (khoá đảm bảo $e_K(x) = y$) là khoá $K = y\text{-}x \bmod 26$. 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 dồng 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 $p_P(x \mid y) = p_P(x)$ với mọi $x \in P$, $y \in P$ là tương đương với $p_C(y \mid x) = p_C(y)$ với mọi $x \in P$, $y \in P$.

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 $x \in P$. Với mỗi $y \in C$ ta có $p_C(y \mid x) = p_C(y) > 0$. 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 $|C| \geq |P|$ vì mỗi quy tắc mã hoá là một đơn ánh. Trong trường hợp giới hạn, $|\mathcal{K}| = |C| = |P|$, ta có định lý sau (Theo Shannon).

Định lý 1.3

Giả sử (P, C, $\mathcal{K}$, E, D) là một hệ mật , trong đó $|\mathcal{K}| = |C| = |P|$. 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 $x \in P$, 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 $x \in P$ và $y \in C$, phải có ít nhất một khoá K sao cho $e_K(x) = y$. Bởi vậy ta có bất đẳng thức:

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

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

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

Tức là ở đây không tồn tại hai khoá K_1 và K_2 khác nhau để $e_{K1}(x) = e_{K2}(x) = y$. Như vậy ta đã chứng tỏ được rằng, với bất kỳ $x \in P$ và $y \in C$ có đúng một khoá K để $e_K(x)=y$.

Ký hiệu $n = |\mathcal{K}|$. Giả sử $P = \{ x_i: 1 \le i \le n \}$ và cố định một giá trị $y \in C$. Ta có thể ký hiệu các khoá $K_1, K_2, \ldots, K_n$ sao cho $e_{Ki}(x_i) = y_i$, $1 \le i \le n$. 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 $|\mathcal{K}|$ 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 1.2). 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.

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. ENTROPY VÀ CÁC TÍNH CHẤT CỦA ENTROPY

1.4.1. Entropy

Trong phần trước ta đã đề cập 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).

Xét ví dụ cụ thể: 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 bít có độ dài n.

Xét ví dụ phức tạp hơn: 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 bít 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 bít 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 bít 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 1.3

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:

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

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

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

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 $1 \leq i \leq n$ thì $H(X) = \log_2 n$. Cũng dễ dàng thấy rằng $H(X) \geq 0$ 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 rõ và bản mã.

Ví dụ 1.1: (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) ≈1,85.

1.4.2. 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 1.4

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

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

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

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

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 f là một hàm lồi thực sự và liên tục trên khoảng I,

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

và $a_i > 0, 1 \leq i \leq n$. Khi đó:

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

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

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 $(0, \infty)$ (Điều này dễ dàng thấy được từ những tính toán sơ cấp vì đạo hàm cấp 2 của hàm logarit là âm trên khoảng $(0, \infty)$).

Định lý 1.5.

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

Chứng minh:

Áp dụng bất đẳng thức Jensen, ta có:

$$\begin{aligned} H(X) &= -\sum_{i=1}^{n} p_i \log_2 p_i = \sum_{i=1}^{n} p_i \log_2 (1/p_i) \\ &\le \log_2 \sum_{i=1}^{n} (p_i \times 1/p_i) \\ &= \log_2 n \end{aligned}$$

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

Định lý 1.6.

$$H(X,Y) \le H(X) + H(Y) \qquad (1.13)$$

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

Nhận thấy rằng

$$p_i = \sum_{j=1}^{n} r_{ij} \quad (1 \le i \le m) \qquad (1.14)$$

và

$$q_j = \sum_{i=1}^{m} r_{ij} \quad (1 \le j \le n) \qquad (1.15)$$

Ta có

$$H(X)+H(Y) = -(\sum_{i=1}^{m} p_i \log_2 p_i + \sum_{j=1}^{n} q_j \log_2 q_j)$$
$$= -(\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} \log_2 p_i + \sum_{j=1}^{n}\sum_{i=1}^{m} r_{ij} \log_2 q_j) \qquad (1.16)$$
$$= -\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} \log_2 p_i q_j$$

Mặt khác $$H(X,Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} \log_2 r_{ij} \qquad (1.17)$$

Kết hợp lại ta thu được kết quả sau:

$$H(X,Y) - H(X) - H(Y) = \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} \log_2 (1/r_{ij}) + \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} \log_2 p_i q_j$$
$$\leq \log_2 \sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j \qquad (1.18)$$
$$= \log_2 1$$
$$= 0$$

(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:

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

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

$$p(X = x_j, Y = y_j) = p(X = x_j) p(Y = y_j)$$

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

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

Định nghĩa 1.5

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 :

$$H(X \mid y) = -\sum_{x} p(x \mid y) \log_2 p(x \mid y)$$

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

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

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.

$$H(X,Y) = H(Y) + H(X \mid Y)$$

Hệ quả 1.2.

$$H(X \mid Y) \leq H(X)$$

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

1.5. CÁC KHÓA GIẢ VÀ KHOẢNG GIẢI MÃ 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 (P, C, K, E, D) là một hệ mật. Khi đó:

H(K|C) = H(K) + H(P) - H(C)

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(P | K,C) = 0 và bởi vậy H(K,P,C) = H(K,P). Bây giờ ta sẽ tính như sau:

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

Ta sẽ quay lại ví dụ 1.1 để minh hoạ kết quả này.

Ví dụ 1.1 (tiếp)

Ta đã tính được H(P) ≈ 0,81, H(K) = 1,5 và H(C) ≈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 $p(K_j \mid C_j)$, $1 \le i \le 3$, $1 \le j \le 4$. Để thực hiện điều này có thể áp dụng định lý Bayes và nhận được kết quả như sau:

$P(K_1 \mid 1) = 1$	$p(K_2 \mid 1) = 0$	$p(K_3 \mid 1) = 0$
$P(K_1 \mid 2) = 6/7$	$p(K_2 \mid 2) = 1/7$	$p(K_3 \mid 2) = 0$

$P(K_1 \mid 3) = 0$ $p(K_2 \mid 3) = 3/4$ $p(K_3 \mid 3) = 1/4$

$P(K_1 \mid 4) = 0$ $p(K_2 \mid 4) = 0$ $p(K_3 \mid 4) = 1$

Bây giờ ta tính:

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

Giá trị này bằng giá trị được tính theo định lý 1.8.

Giả sử (P, C, K, E, D) là hệ mật đang được sử dụng. Một xâu của bản rõ $x_1x_2 \ldots x_n$ sẽ được mã hoá bằng một khoá để tạo ra bản mã $y_1y_2 \ldots y_n$. 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 entropi trên một kí tự bằng $\log_2$

26 ≈ 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) ≈ 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 1.6

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

$$H_L = \lim_{n\to\infty} \frac{H(P^n)}{n} \qquad (1.24)$$

Độ dư của L là: $R_L = 1 - (H_L / \log_2 |P|)$ (1.25)

Nhận xét: H_L đo entropy trên mỗi kí tự của ngôn ngữ L. Một ngôn ngữ ngẫu nhiên sẽ có entropi là $\log_2 |P|$. 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 $H\left(P^2\right) \approx 3,90$. 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 bít tới 1,5 bít 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 (đây là một phép mã hóa nén thực hiện theo nguyên tắc các tin có xác suất xuất hiện lớn phải được mã hóa bằng các từ mã có độ dài nhỏ và ngược lại) 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 đã cho trên K và P^n. Có thể xác định phân bố xác suất trên C^n là tập các bộ n của bản mã. (Ta đã làm điều này trong trường hợp n =1). Ta đã xác định $\mathcal{P}^n$ 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 $|K(y)|-1$ 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:

$$\begin{aligned}\bar{s}_n &= \sum_{y \in C^n} p(y)\left(|K(y)|-1\right) \\ &= \sum_{y \in C^n} p(y)|K(y)| - \sum_{y \in C^n} p(y) \\ &= \sum_{y \in C^n} p(y)|K(y)| - 1\end{aligned} \qquad (1.26)$$

Từ định lý 1.8 ta có:

$$H\left(K\middle|C^n\right) = H(K) + H\left(P^n\right) - H\left(C^n\right) \tag{1.27}$$

Có thể dùng ước lượng sau:

$$H\left(P^n\right) \approx nH_L = n\left(1 - R_L\right)\log_2\left|\mathcal{P}\right| \tag{1.28}$$

với điều kiện n đủ lớn. Hiển nhiên là:

$$H\left(C^n\right) \le n\log_2\left|C\right| \tag{1.29}$$

Khi đó nếu $\left|\mathcal{P}\right| = \left|C\right|$ thì:

$$H\left(K\middle|C^n\right) \ge H(K) - nR_L\log_2\left|\mathcal{P}\right| \tag{1.30}$$

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

$$\begin{aligned} H\left(K\middle|C^n\right) &= \sum_{y\in C^n} p(y)\left(\middle|K\middle|y\right) \\ &\le \sum_{y\in C^n} p(y)\log_2\left|K(y)\right| \\ &\le \sum_{y\in C^n} p(y)\left|K(y)\right| \\ &= \log_2\left(\overline{s_n} + 1\right) \end{aligned} \tag{1.31}$$

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:

$$H(K\left|C^n\right.) \le \log_2(\overline{s_n} + 1) \tag{1.32}$$

Kết hợp hai bất đẳng thức (1.1) và (1.2), ta có :

$$\log_2(\overline{s_n} + 1) \ge H(K) - nR_L\log_2\left|P\right| \tag{1.33}$$

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 (P, C, K, E, D) là một hệ mật trong đó |C | = |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 $\overline{s_n}$ thoả mãn bất đẳng thức như sau:

$$\overline{s_n} \geq \left\{ |K| / \left(|P| nR_L \right) \right\} - 1 \qquad (1.34)$$

Lượng $|K| / \left(|P| nR_L \right) - 1$ 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 1.7.

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 $\overline{s_n}$ = 0 trong định lý 1.9 và giải theo n ta sẽ nhận được ước lượng cho khoảng duy nhất:

$$n_0 \approx \log_2 |\mathcal{K}| / R_L \log_2 |\mathcal{P}| \qquad (1.35)$$

Ví dụ với MTT, ta có |P| = 26 và |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.

1.6. PHÂN TÍCH MẬT MÃ VÀ ĐỘ PHỨC TẠP TÍNH TOÁN

1.6.1. Phân tích mật mã

Khoa học về mật mã (cryptology) bao gồm:

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

- Phân tích mật mã (cryptanalysis):là khoa học nghiên cứu cách phá các hệ mật nhằm phục hồi bản rõ ban đầu từ bản mã.

Mật mã được sử dụng trước hết là để đảm bảo tính bí mật cho các thông tin được trao đổi, và do đó bài toán quan trọng nhất của phân tích mật mã (hay còn được gọi là thám mã) cũng là bài toán phá bỏ tính bí mật đó, tức là từ bản mật mã có thể thu được dễ dàng (trên các kênh truyền tin công cộng) ngườoi thám mã phải phát hiện được nội dung thông tin bị che giấu trong bản mật mã đó. Tình huống thường gặp là bản thân sơ đồ hệ thống mật mã, kể cả các phép lập mã và giải mã (tức là cá thuật toán $\mathcal{E}$ và $\mathcal{D}$), không nhất thiết là bí mật, do đó bài toán quy về việc tìm khóa mật mã K, hay khóa giải mã K' nếu hệ mật mã có khóa phi đối xứng. Như vậy, ta có thể quy ước xem bài toán thám mã cơ bản là bài toán tìm khóa mật mã K (hay khóa giải mã K'). Để giải bài toán đó, giải thiết người thám mã biết thông tin về sơ đồ hệ mật mã được dùng, kể cả các phép lập mã và giải mã tổng quát $\mathcal{E}$ và $\mathcal{D}$. Ngoài ra, người thám mã có thể biết thêm một số thông tin khác, tùy theo những thông tin được biết thêm này mà ta có thể phân loại bài toán thám mã thành các bài toán cụ thể như sau:

- Bài toán thám mã *chỉ biết bản mã*: là bài toán phổ biến nhất, khi người thám mã chỉ biết một bản mật mã Y.

- Bài toán thám mã khi *biết cả bản rõ*: người thám mã biết một bản mật mã Y cùng với bản rõ tương ứng X.

- Bài toán thám mã khi *có bản rõ được chọn*: người thám mã có thể chọn một bản rõ X, và biết bản mật mã tương ứng Y. Điều này có thể xảy ra khi người thám mã chiếm được (tạm thời) máy lập mã

- Bài toán thám mã khi *có bản mã được chọn*: người thám mã có thể chọn một bản mật mã Y, và biết bản rõ tương ứng X. Điều này có thể xảy ra khi người thám mã chiếm được tạm thời máy giải mã.

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.

1.6.2. Độ phức tạp tính toán

1.6.2.1.Khái niệm về độ phức tạp tính toán

Lí thuyết thuật toán và các hàm số tính được ra đời từ những năm 30 của thế kỉ 20 đã đặt nền móng cho việc nghiên cứu các vấn đề "tính được", "giải được" trong toán học, đưa đến nhiều kết quả quan trọng và lý thú. Nhưng từ cái "tính được" một cách trừu tượng, hiểu theo nghĩa tiềm năng, đến việc tính được trong thực tế của khoa học tính toán bằng máy tính điện tử, là cả một khoảng cách rất lớn. Biết bao nhiêu thứ được chứng minh là tính được một cách tiềm năng, nhưng không tính được trong thực tế, dù có sự hỗ trợ của những máy tính điện tử. Vấn đề là do ở chỗ những đòi hỏi về không gian vật chất và về thời gian để thực hiện các tiến trình tính toán nhiều khi vượt quá xa những khả năng thực tế. Từ đó, vào khoảng giữa những năm 60 (của thế kỉ trước), một lý thuyết về độ phức tạp tính toán bắt đầu được hình thành và phát triển nhanh chóng, cung cấp cho chúng ta nhiều hiểu biết sâu sắc về bản chất phức tạp của các thuật toán và các bài toán, cả những bài toán thuần túy lý thuyết đến những bài toán thường

gặp trong thực tế. Sau đây ta giới thiệu sơ lược một số khái niệm cơ bản và vài kết quả sẽ được dùng đến của lý thuyết đó.

Trước hết, ta hiểu *độ phức tạp tính toán* (về không gian hay về thời gian) của một tiến tình tính toán là số ô nhớ được dùng hay số các phép toán sơ cấp được thực hiện trong tiến trình tính toán đó.

Dữ liệu đầu vào đối với một thuật toán thường được biểu diễn qua các từ trong một bảng kí tự nào đó. *Độ dài của một từ* là số kí tự trong từ đó.

Cho một thuật toán $\mathcal{A}$ trên bảng kí tự $\sum$ (tức có đầu vào là các từ trong $\sum$). Độ phức tạp tính toán của thuật toán $\mathcal{A}$ được hiểu là một hàm số $f_A(n)$ sao cho với mỗi số n, $f_A(n)$ là số ô nhớ, hay số phép toán sơ cấp tối đa mà $\mathcal{A}$ cần để thực hiện tiến trình tính toán của mình trên các dữ liệu có độ dài $\leq n$. Ta nói thuật toán $\mathcal{A}$ có độ phức tạp thời gian *đa thức*, nếu có một đa thức P(n) sao cho với mọi n đủ lớn ta có $f_A(n) \leq P(n)$, trong đó $f_A(n)$ là độ phức tạp tính toán theo thời gian của $\mathcal{A}$.

Về sau khi nói đến các bài toán, ta hiểu đó là các *bài toán quyết định*, mỗi bài toán P như vậy được xác định bởi:

- Một tập các dữ liệu I (trong một bảng kí tự $\sum$ nào đó)

- Một câu hỏi Q trên các dữ liệu vào, sao cho với mỗi dữ liệu vào $x \in I$, câu hỏi Q có một trả lời *đúng* hoặc *sai*.

Ta nói bài toán quyết định P là *giải được*, nếu có thuật toán để giải nó, tức là thuật toán làm việc có kết thúc trên mọi dữ liệu vào của bài toán, và cho kết quả *đúng* hoặc *sai* tùy theo câu hỏi Q trên dữ liệu đó có trả lời đúng hoặc sai. Bài toán P là *giải được trong thời gian đa thức*, nếu có thuật toán giải nó với độ phức tạp thời gian đa thức.

1.6.2.2. Lớp phức tạp

Ta xét một vài lớp các bài toán được xác định theo độ phức tạp tính toán của chúng. Trước hết, ta định nghĩa $\mathcal{P}$ là lớp tất cả các bài toán có thể giải được bởi thuật toán trong thời gian đa thức.

Giả sử cho hai bài toán P_1, P_2 với các tập dữ liệu trong hai bảng kí tự tương ứng là $\sum_1$ và $\sum_2$. Một thuật toán f: $\sum_1^* \to \sum_2^*$ được gọi là một *phép qui dẫn* bài toán P_1 về bài toán P_2, nếu nó biến mỗi dữ liệu x của bài toán P_1 thành một dữ liệu f(x) của bài toán P_2, và sao cho câu hỏi của P_1 trên x có trả lời đúng khi và chỉ khi câu hỏi của P_2 trên f(x) cũng có trả lời đúng. Ta nói bài toán P_1 *quy dẫn được* về bài toán P_2 trong *thời gian đa thức*, và kí hiệu $P_1 \alpha P_2$, nếu có thuật toán f với độ phức tạp thời gian đa thức quy dẫn bài toán P_1 về bài toán P_2. Ta dễ dàng thấy rằng, nếu $P_1 \alpha P_2$ và $P_2 \in P$ thì cũng có $P_1 \in P$.

Một lớp quan trọng các bài toán đã được nghiên cứu nhiều là lớp các bài toán khá thường gặp trong thực tế nhưng cho đến nay chưa có khả năng nào chứng tỏ là chúng có thể giải được trong thời gian đa thức. Đó là lớp các bài toán *NP đầy đủ* được trình bày sau đây:

Cùng với khái niệm thuật toán tất định thông thường (có thể mô tả chính xác chẳng hạn bởi máy Turing tất định), ta xét khái niệm thuật toán *không đơn định* với một ít thay đổi như sau: nếu đối với máy Turing tất định, khi máy đang ở một trạng thái q và đang đọc kí tự a thì cặp (q, a) xác định duy nhất một hành động kế tiếp của máy, còn đối với máy Turing không đơn định, ta quy ước rằng (q, a) xác định không phải duy nhất mà là một tập hữu hạn các hành động kế tiếp, máy có thể thực hiện trong bước kế tiếp một trong các hành động đó. Như vậy, đối với một dữ liệu vào x, một thuật toán không đơn định (được xác định chẳng hạn bởi một máy Turing không đơn định) không phải chỉ có một tiến trình

tính toán duy nhất, mà có thể có một số hữu hạn những tiến trình tính toán khác nhau. Ta nói thuật toán không đơn định $\mathcal{A}$ *chấp nhận* dữ liệu x, nếu với dữ liệu vào x thuật toán $\mathcal{A}$ có ít nhất một tiến trình tính toán kết thúc ở trạng thái chấp nhận (tức với kết quả *đúng*). Một bài toán P được gọi là *giải được bởi thuật toán không đơn định trong thời gian đa thức* nếu có một thuật toán không đơn định $\mathcal{A}$ và một đa thức p(n) sao cho với mọi dữ liệu vào x có độ dài n, $x \in P$ (tức câu hỏi của P có trả lời đúng trên x) khi và chỉ khi thuật toán $\mathcal{A}$ chấp nhận x bởi một tiến trình tính toán có độ phức tạp thời gian $\leq p(n)$. Ta kí hiệu lớp tất cả các bài toán giải được bởi thuật toán không đơn định trong thời gian đa thức là NP.

Người ta đã chứng tỏ được rằng tất cả những bài toán trong các thí dụ kể trên và rất nhiều các bài toán tổ hợp thường gặp khác đều thuộc lớp NP, dù rằng hầu hết chúng để chưa được chứng tỏ là thuộc P. Một bài toán P được gọi là *NP – đầy đủ*, nếu $P \in NP$ và với mọi $Q \in NP$ đều có $Q \alpha P$

Lớp NP có một số tính chất sau đây:

- $P \subseteq NP$

- Nếu $P_1 \alpha P_2$ và $P_2 \in NP$ thì $P_1 \in NP$

- Nếu $P_1, P_2 \in NP$, $P_1 \alpha P_2$ và P_1 là NP đầy đủ, thì P_2 cũng là NP đầy đủ

- Nếu có P sao cho P là NP đầy đủ và $P \in \mathcal{P}$, thì $\mathcal{P} = NP$

Từ các tính chất đó ta có thể xem rằng trong lớp NP, $\mathcal{P}$ là lớp con các bài toán "dễ" nhất, còn các bài toán NP đầy đủ là các bài toán "khó" nhất, nếu có ít nhất một bài toán NP đầy đủ được chứng minh là thuộc $\mathcal{P}$ thì lập tức suy ra $\mathcal{P} = NP$, dù rằng cho đến nay tuy đã có nhiều cố gắng nhưng toán học vẫn chưa tìm được con đường nào hi vọng đi đến giải quyết vấn đề [$\mathcal{P} = NP?$], thậm chí vấn đề

đó còn được xem là một trong bảy vấn đề khó nhất của toán học trong thiên niên kỉ mới!

1.6.2.3.Độ phức tạp tính toán

Định nghĩa 1.8: 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 $f[n]<Cg[n]$.

Ví dụ :

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

4. tồn tại giới hạn hữu hạn:

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

thì $f = O(g)$

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

Định nghĩa 1.9: 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 bít thì thời gian thực hiện thuật toán là $O\left(k^d\right)$, tức là tương đương với một đa thức của k.

Các thuật toán với thời gian $O\left(n^\alpha\right)$, $\alpha > 0$ 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.

1.7. BỔ TÚC VỀ LÝ THUYẾT SỐ

1.7.1. Số học modulo và đồng dư

1.7.1.1.Số nguyên

Tập các số nguyên $\{\ldots,-3,-2,-1,0,1,2,3,\ldots\}=Z$

Định nghĩa 1.10:

Cho $a,b\in Z$, a là ước của b nếu $\exists c\in Z$: $b=a.c$. Ký hiệu $a\,|\,b$

Định nghĩa 1.11 (Thuật toán chia đối với các số nguyên)

Nếu a và b là các số nguyên với $b\geq 1$

thì $a=qb+r$, $0\leq r<b$

q và r là duy nhất .

Phần dư của phép chia a và b được ký hiệu $a \bmod b = r$

Thương của phép chia a và b được ký hiệu $a \operatorname{div} b = q$

Ta có $a \operatorname{div} b=\left[\frac{a}{b}\right]$, $a \bmod b = a-b\left[\frac{a}{b}\right]$

Ví dụ: $a=73,\ b=17.$

$73 \operatorname{div} 17 = 4$, $73 \bmod 17 = 5$

Định nghĩa 1.12: Ước chung.

c là ước chung của a và b nếu $c\,|\,a$ & $c\,|\,b$

Định nghĩa 1.13: Ước chung lớn nhất (ƯCLN)

Số nguyên dương d là ƯCLN của các số nguyên a và b (Ký hiệu $d=(a,b)$) nếu:

(i) d là ước chung của a và b.

(ii) Nếu có $c\,|\,a$ và $c\,|\,b$ thì $c\,|\,d$.

Như vậy (a,b) là số nguyên dương lớn nhất ước của cả a và b không kể $(0,0)=0$.

Ví dụ: Các ước chung của 12 và 18 là $\{\pm 1,\pm 2,\pm 3,\pm 6\}$

$$(12, 18) = 6$$

Định nghĩa 1.14: Bội chung nhỏ nhất (BCNN)

Số nguyên dương d là BCNN của các số nguyên a và b (Ký hiệu $d = \text{BCNN}(a, b)$) nếu:

(i) $a \mid d$, $b \mid d$.

(ii) Nếu có $a \mid c$, $b \mid c$ thì $d \mid c$.

Như vậy d là số nguyên dương nhỏ nhất là bội của cả a và b.

Tính chất

$$\text{BCNN}(a, b) = \frac{a \cdot b}{(a, b)}$$

Ví dụ: $(12, 18) = 6 \quad \Rightarrow \quad \text{BCNN}(12, 18) = \frac{12 \cdot 18}{6} = 36$

Định nghĩa 1.15:

Hai số nguyên dương a và b được gọi là nguyên tố cùng nhau nếu: $(a, b) = 1$

Định nghĩa 1.16:

Số nguyên $P \geq 2$ được gọi là số nguyên tố nếu các ước dương của nó chỉ là 1 và P. Ngược lại P được gọi là hợp số.

Định lý cơ bản của số học:

Với mỗi số nguyên $r \geq 2$ ta luôn phân tích được dưới dạng tích của luỹ thừa của các số nguyên tố.

$$n = p_1^{e_1} \, p_2^{e_2} \ldots p_k^{e_k}$$

Trong đó p_i là các số nguyên tố khác nhau và e_i là các số nguyên dương. Hơn nữa phân tích trên là duy nhất.

Định nghĩa 1.17:

Với $n \geq 2$, hàm $\Phi(n)$ được xác định là số các số nguyên trong khoảng $[1, n]$ nguyên tố cùng nhau với n.

Các tính chất của hàm $\Phi(n)$

(i) Nếu p là các số nguyên tố thì $\Phi(p) = p - 1$.

(ii) Nếu $(m, n) = 1$ thì $\Phi(m.n) = \Phi(m).\Phi(n)$.

(iii)Nếu $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$ là phân tích ra thừa số nguyên tố của n thì:

$$\Phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\dots\left(1 - \frac{1}{p_k}\right)$$

Định lý 1.10:

Với $\forall n \geq 5$:

$$\Phi(n) > \frac{n}{6 \ln \ln n}$$

1.7.1.2. Các số nguyên modulo n.

Định nghĩa 1.18:

Nếu a và b là các số nguyên thì a được gọi là đồng dư với b theo modulo (ký hiệu là $a \equiv b \bmod n$) nếu $n \mid (a - b)$.

Số nguyên n được gọi là modulo đồng dư.

Ví dụ: $24 \equiv 9 \bmod 5$ vì $24 - 9 = 3.5$

$-11 \equiv 17 \bmod 7$ vì $-11 - 17 = -4.7$

Các tính chất:

Đối với $a, a_1, b, b_1, c \in Z$ ta có:

(1) $a \equiv b (\bmod n)$ nếu và chỉ nếu a và b cũng có phần dư khi chia cho n.

(2) Tính phản xạ : $a \equiv a (\bmod n)$.

(3) Tính đối xứng : Nếu $a \equiv b (\bmod n)$ thì $b \equiv a (\bmod n)$

(4) Tính bắc cầu : Nếu $a \equiv b (\bmod n)$ và $b \equiv c (\bmod n)$ thì

$$a \equiv c (\bmod n)$$

(5) Nếu $a \equiv a_1 (\bmod n)$ và $b \equiv b_1 (\bmod n)$ thì

$$a + b \equiv a_1 + b_1 (\bmod n) \text{ và } a.b \equiv a_1.b_1 (\bmod n)$$

Lớp tương đương của một số nguyên a là tập các số nguyên đồng dư với a modulo n. Từ các tính chất (2), (3) và (5) ở trên ta có thể thấy rằng đối với n cố

định, quan hệ đồng dư theo modulo n sẽ phân hoạch Z thành các lớp tương đương.

Nếu $a = qn + r$ với $0 \le r \le n$ thì $a \equiv r(\mathrm{mod}\, n)$.

Bởi vậy mỗi số nguyên a là đồng dư theo modulo n với một số nguyên duy nhất nằm trong khoảng từ 0 tới $n-1$, số này được gọi là thặng dư tối thiểu của $a \bmod n$. Như vậy a và r có thể được dùng để biểu thị cho lớp tương đương này.

Định nghĩa 1.19:

Các số nguyên modulo n (ký hiêu Z_n) là tập (các lớp tương đương) của các số nguyên $\{0,1,2,\ldots,n-1\}$. Các phép cộng , trừ, nhân trong Z_n được thực hiện theo modulo n.

Ví du: $Z_{25} = \{0,1,\ldots,24\}$. Trong Z_{25} ta có:

$$13 + 16 = 4 \text{ vì } 13 + 16 = 29 \equiv 4 (\mathrm{mod}\ 25)$$

Tương tự $13.16 = 8$ trong Z_{25}.

Định lý 1.11 (Phần dư Trung hoa).

Nếu các số nguyên $n_1, n_2, \ldots, n_k$ là nguyên tố cùng nhau từng đôi một thì hệ các phương trình đồng dư:

$$x \equiv a_1 (\mathrm{mod}\ n_1)$$
$$x \equiv a_2 (\mathrm{mod}\ n_2)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$x \equiv a_k (\mathrm{mod}\ n_k)$$

sẽ có nghiệm duy nhất theo modulo n $(n = n_1 . n_2 \ldots n_k)$

Thuật toán Gausse.

Nghiệm x của hệ phương trình đồng dư trong định lý phần dư China có thể đươc tính bằng:

$$x = \sum_{i=1}^{k} a_i N_i M_i \bmod n$$

Trong đó $N_i = n / n_i$ và $M_i = N_i^{-1} \bmod n_i$

Các tính toán này có thể được thực hiện trong $0((\lg n)^2)$ các phép toán trên bit.

Ví dụ: Cặp phương trình đồng dư $x \equiv 3 (\bmod 7)$, $x \equiv 7 (\bmod 13)$ có nghiệm duy nhất $x \equiv 59 (\bmod 91)$

Định lý 1.12:

Nếu $(n_1, n_2) = 1$ thì cặp phương trình đồng dư.

$$x \equiv a (\bmod n_1),\ x \equiv a (\bmod n_2)$$

có một nghiệm duy nhất $x \equiv a \pmod{n_1.n_2}$

Định nghĩa 1.20:

Nhóm nhân của Z_n là $Z_n^* = \{a \in Z_n \mid (a, n) = 1\}$

Đặc biệt, nếu n là số nguyên tố thì $Z_n^* = \{a \mid 1 \le a \le n-1\}$

Định nghĩa 1.21:

Cấp của Z_n^* là số các phần tử trong Z_n^* (ký hiệu $|Z_n^*|$)

Theo định nghĩa của hàm Phi-Euler ta thấy:

$$|Z_n^*| = \Phi(n)$$

Cần để ý rằng nếu $a \in Z_n^*$ và $b \in Z_n^*$ thì $a.b \in Z^*_n$ và bởi vậy Z_n^* là đóng đối với phép nhân.

Định lý 1.13:

(1) *Định lý Euler:* Nếu $a \in Z_n^*$ thì $a^{\Phi(n)} \equiv 1 (\bmod n)$.

(2) Nếu n là tích của các số nguyên khác nhau và nếu $r \equiv s (\bmod \Phi(n))$ thì $a^r \equiv a^s (\bmod n)$ đối với mọi số nguyên a. Nói một cách khác khi làm việc với modulo n thì các số mũ có thể được rút gọn theo modulo $\Phi(n)$.

Định lý 1.14: Cho p là một số nguyên tố:

(1) *Định lý Ferma:* Nếu $(a,p)=1$ thì $a^{p-1} \equiv 1 \pmod{p}$.

(2) Nếu $r \equiv s \pmod{p-1}$ thì $a^r \equiv a^s \pmod{p}$ đối với mọi số nguyên a. Nói một cách khác khi làm việc với modulo của một số nguyên tố p thì các luỹ thừa có thể được rút gọn theo modulo $p-1$.

(3) Đặc biệt $a^p \equiv a \pmod{p}$ với mọi số nguyên a.

Định nghĩa 1.22:

Cho $a \in Z_n^*$. Cấp của a (ký hiệu là $\text{ord}(a)$) là số nguyên dương nhỏ nhất t sao cho $a^t \equiv 1 \pmod{n}$.

Định nghĩa 1.23:

Cho $a \in Z_n^*$, $\text{ord}(a)=t$ và $a^s \equiv 1 \pmod{n}$ khi đó t là ước của s. Đặc biệt $t \mid \Phi(n)$.

Ví dụ: Cho $n=21$, khi đó $Z_{21}^* = \{1,2,4,5,8,10,11,13,16,17,19,20\}$

Chú ý rằng $\Phi(21) = \Phi(7)\Phi(3) = 12 = |Z_{21}^*|$. Cấp của các phần tử trong Z_{21}^* được nêu trong bảng sau:

Bảng 1-1. Cấp của các phần tử trong Z^*_{21}

$a \in Z_{21}^*$	1	2	4	5	8	10	11	13	16	17	19	20
$\text{ord}(a)$	1	6	3	6	2	6	6	2	3	6	6	2

Định nghĩa 1.24:

Cho $\alpha \in Z_n^*$. Nếu cấp của α là $\Phi(n)$ thì α được gọi là phần tử sinh hay phần tử nguyên thuỷ của Z_n^*. Nếu Z_n^* có một phần tử sinh thì Z_n^* được gọi là cyclic.

Các tính chất của các phần tử sinh của Z_n^*

(1) Z_n^* có phần tử sinh nếu và chỉ nếu $n = 2, 4, p^k$ hoặc là $2p^k$, trong đó p là một số nguyên tố lẻ và $k \geq 1$. Đặc biệt, nếu p là một số nguyên tố thì Z_n^* có phần tử sinh.

(2) Nếu α là một phần tử sinh của Z_n^* thì:

$$Z_n^* = \{\alpha^i \bmod n \mid 0 \leq i \leq \Phi(n) - 1\}$$

(3) Giả sử rằng α là một phần tử sinh của Z_n^* khi đó $b = \alpha^i \bmod n$ cũng là một phần tử sinh của Z_n^* nếu và chỉ nếu $(i, \Phi(n)) = 1$. Từ đó ta rút ra rằng nếu Z_n^* là cyclic thì số các phần tử sinh là $\Phi(\Phi(n))$.

(4) $\alpha \in Z_n^*$ là một phần tử sinh của Z_n^* nếu và chỉ nếu $\alpha^{\Phi(n)/p} \neq 1 (\bmod n)$ đối với mỗi nguyên tố p của $\Phi(n)$

Ví dụ: Z_{21}^* không là cyclic vì nó không chứa một phần tử có cấp $\Phi(21) = 12$ (Chú ý rằng 21 không thoả mãn điều kiện (1) ở trên).

Z_{25}^* là cyclic và có một phần tử sinh $\alpha = 2$

Định nghĩa 1.25:

Cho $a \in Z_n^*$, a được gọi là thặng dư bậc hai modulo n (hay bình phương của modulo n) nếu tồn tại $x \in Z_n^*$ sao cho $x^2 \equiv a (\bmod n)$. Nếu không tồn tại x như vậy thì a được gọi là thặng dư không bậc hai $\bmod n$. Tập tất cả các thặng dư bậc hai modulo n được ký hiệu là Q_n, còn tập tất cả các thặng dư không bậc hai được ký hiệu là $\overline{Q_n}$. Cần chú ý rằng theo định nghĩa $0 \notin Z_n^*$. Bởi vậy $0 \notin Q_n$ và $0 \notin \overline{Q_n}$.

Định lý 1.15:

Cho p là một số nguyên tố lẻ và α là một phần tử sinh của Z_p^*. Khi đó $a \in Z_p^*$ là một thặng dư bậc hai modulo p nếu và chỉ nếu $a = \alpha^i \bmod p$, trong đó i là một số nguyên chẵn. Từ đó rút ra rằng $|Q_p| = \frac{(p-1)}{2}$ và $|\overline{Q_p}| = \frac{(p-1)}{2}$, tức là một nửa số phần tử trong Z_p^* là các thặng dư bậc hai và nửa còn lại thặng dư không bậc hai.

Ví dụ: $\alpha = 6$ là một phần tử sinh của Z_{13}^*. Các luỹ thừa của α được liệt kê ở bảng sau:

i	0	1	2	3	4	5	6	7	8	9	10	11
$\alpha^i \bmod 13$	1	6	10	8	9	2	12	7	3	5	4	11

Bởi vậy $Q_{13} = \{1,3,4,9,10,12\}$, $\overline{Q}_{13} = \{2,5,6,7,8,11\}$

Định lý 1.16:

Cho n là tích của hai số nguyên tố lẻ khác nhau q và p, $n = p.q$, khi đó $a \in Z_n^*$ là một thặng dư bậc hai modulo n nếu và chỉ nếu $a \in Q_p$ và $a \in Q_q$. Điều đó dẫn tới $|Q_n| = |Q_q|.|Q_p| = \frac{(p-1)(q-1)}{4}$

và $|\overline{Q}_n| = \frac{3(p-1)(q-1)}{4}$

Ví dụ: Cho $n = 21$. Khi đó $Q_{21} = \{1,4,16\}$ $\overline{Q}_{21} = \{2,5,8,10,11,13,17,19,20\}$

Định nghĩa 1.26:

Cho $a \in Q_n$. Nếu $x \in Z_n^*$ thoả mãn $x^2 \equiv a \pmod n$ thì x được gọi là căn bậc hai của a mod n.

Định lý 1.17 (Số các căn bậc hai).

(1) Nếu p là một số nguyên tố lẻ và $a \in Q_n$ thì a được gọi là thặng dư bậc hai theo modulo p.

(2) Tổng quát hơn, cho $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$, trong đó p_i là các số nguyên tố lẻ phân biệt và $e_i \geq 1$. Nếu $a \in Q_n$ thì có đúng 2^k căn bậc hai khác nhau theo modulo n.

Ví dụ: Các căn bậc 2 của $12 \bmod 37$ là 7 và 30. Các căn bậc 2 của $121 \bmod 315$ là 11, 74, 101, 151, 164, 214, 241 và 304.

1.7.2. Nghịch đảo nhân

Định nghĩa 1.27 (Phần tử nghịch đảo).

Cho $a \in Z_n$, Phần tử nghịch đảo (ngược theo phép nhân) của $a \bmod n$ là một số nguyên $x \in Z_n$ sao cho: $a\,x \equiv 1 (\bmod n)$

Nếu x tồn tại thì nó là duy nhất, a được gọi là khả nghịch. Phần tử nghịch đảo của a được ký hiệu là a^{-1}.

Định nghĩa 1.28:

Phép chia của với a cho là tích của a và $b^{-1} \bmod n$ tích này được xác định nếu b là phần tử khả nghịch

Định lý 1.18:

Cho $a \in Z_n$, khi đó a là khả nghịch nếu và chỉ nếu : $(a,n) = 1$

Ví dụ: Các phần tử khả nghịch trong Z_9 là 1, 2, 4, 5, 7 và 8.

Chẳng hạn $4^{-1} = 7$ vì $4.7 \equiv 1 (\bmod 9)$.

Định lý 1.19:

Cho $d = (a,n)$. Phương trình đồng dư $ax \equiv b (\bmod n)$ có nghiệm x nếu và chỉ nếu $d \mid b$, trong trường hợp này có đúng d nghiệm nằm giữa 0 và $n-1$, những nghiệm này là tất cả các đồng dư theo modulo $n \mid d$.

1.7.3. Thuật toán Euclide

Tính UCLN của 2 số nguyên

VÀO : Hai số nguyên không âm a và b với $a > b$

RA : ƯCLN của a và b.

(1) While $b \neq 0$ do

$$r \leftarrow a \bmod b \ , \ a \leftarrow b \ , \ b \leftarrow r$$

(2) Return (a).

Định lý 1.20:

Thuật toán trên có thời gian chạy chừng $0\left((\lg n)^2\right)$ các phép toán bit.

Ví dụ: Sau đây là các bước chia của thuật toán trên khi tính:

$$(4864, 3458) = 38$$
$$4864 = 1.3458 + 1406$$
$$3458 = 2.1406 + 646.$$
$$1406 = 2.646 + 76$$
$$646 = 5.114 + 38$$
$$76 = 2.38 + 0$$

Thuật toán trên có thể được mở rộng để không những chỉ tính được ƯCLN của 2 số nguyên a và b mà còn tính được các số nguyên x và y thoả mãn $ax + by = d$.

1.7.4. Thuật toán nghịch đảo

Thuật toán Euclide mở rộng:

VÀO : Hai số nguyên không âm a và b với $a \geq b$

RA : $d = UCLN(a, b)$ và các số nguyên x và y thoả mãn $ax + by = d$.

(1) Nếu $b = 0$ thì đặt $d \leftarrow a \ , \ x \leftarrow 1 \ , \ y \leftarrow 0$ và return (d, x, y)

(2) Đặt $x_2 \leftarrow 1 \ , \ x_1 \leftarrow 0 \ , \ y_2 \leftarrow 0 \ , \ y_1 \leftarrow 1$

(3) While $b > 0$ do

3.1. $q \leftarrow \lfloor a / b \rfloor, r \leftarrow a - qb, x \leftarrow x_2 - qx_1, y \leftarrow y_2 - qy_1$

3.2. $a \leftarrow b$, $b \leftarrow r$, $x_2 \leftarrow x_1$, $x_1 \leftarrow x$, $y_2 \leftarrow y_1$, $y_1 \leftarrow y$

(4) Đặt $d \leftarrow a$, $x \leftarrow x_2$, $y \leftarrow y_2$ và return (d, x, y)

Định lý 1.21:

Thuật toán trên có thời gian chạy cỡ $0((\lg n)^2)$ các phép toán bit.

Ví dụ: Bảng 1.2 sau chỉ ra các bước của thuật toán trên với các giá trị vào $a = 4864$ và $b = 3458$

Bảng 1-2. Thuật toán Euclide mở rộng và các giá trị vào a = 4864, b = 3458

Q	r	x	y	a	b	x_2	x_1	y_2	y_1
-	-	-	-	4864	3458	1	0	0	1
1	1406	1	-1	3458	1406	0	1	1	-1
2	646	-2	3	1406	646	1	-2	-1	3
2	114	5	-7	646	114	-2	5	3	-7
5	76	-27	38	114	76	5	-27	-7	38
1	38	32	-45	76	38	-27	32	38	-45
2	0	-91	128	38	0	32	-91	-45	128

Bảng 1.2: Thuật toán Euclide mở rộng với các đầu vào $a = 4864$ và $b = 3458$

Bởi vậy ta có $\text{UCLN}(4864, 3458) = 38$

và $(4864)(32) + (3458)(-45) = 38$

Thuật toán tính nghịch đảo trong Z_n

VÀO : $a \in Z_n$

RA : $a^{-1} \bmod n$ (nếu tồn tại).

(1) Dùng thuật toán Euclide mở rộng để tìm các số nguyên x và y sao cho $ax + ny = d$ trong đó $d = (a, n)$.

(2) Nếu $d > 1$ thì $a^{-1} \bmod n$ không tồn tại. Ngược lại return (x).

Phép luỹ thừa theo modulo có thể được thực hiện có hiệu quả bằng thuật toán nhân và bình phương có lặp. Đây là một thuật toán rất quan trọng trong nhiều thủ tục mật mã. Cho biểu diễn nhị phân của k là:

$\sum_{i=0}^{t} k_i 2^i$ trong đó mỗi $k_i \in \{0,1\}$ khi đó

$$a^k = \prod_{i=0}^{t} a^{k_i 2^i} = \left(a^{2^0}\right)^{k_0} \left(a^{2^1}\right)^{k_1} \ldots \left(a^{2^t}\right)^{k_t}$$

Thuật toán nhân và bình phương có lặp để lấy luỹ thừa trong Z_n.

VÀO : $a \in Z_n$ và số nguyên k, $(0 \le k < n)$ có biểu diễn nhị phân:

$$k = \sum_{i=0}^{t} k_i 2^i$$

RA : $a^k \bmod n$

(1) Đặt $b \leftarrow 1$. Nếu $k = 0$ thì return (b)

(2) Đặt $A \leftarrow a$.

(3) Nếu $k_0 = 1$ thì đặt $b \leftarrow a$.

(4) For i from 1 to t do

4.1. Đặt $A \leftarrow A^2 \bmod n$.

4.2. Nếu $k_i = 1$ thì đặt $b \leftarrow A.b \bmod n$

(5) Return (b)

Ví dụ: Bảng 1.3 sau chỉ ra các bước tính toán $5^{596} \bmod 1234 = 1013$

Bảng 1-3. Tính $5^{596} \bmod 1234$

i	0	1	2	3	4	5	6	7	8	9

k_i	0	0	1	0	1	0	1	0	0	1
A	5	25	625	681	1011	369	421	779	947	925
b	1	1	625	625	67	67	1059	1059	1059	1013

Số các phép toán bit đối với phép toán cơ bản trong Z_n được tóm lược trong bảng 1.4.

Bảng 1-4. Độ phức tạp bit của các phép toán cơ bản trong Z_n

Phép toán		Độ phức tạp bit
Cộng module	$a + b$	$0(\lg n)$
Trừ modulo	$a - b$	$0(\lg n)$
Nhân modulo	$a.b$	$0((\lg n)^2)$
Nghịch đảo modulo	$a^{-1} \bmod n$	$0((\lg n)^2)$
Luỹ thừa modulo	$a^k \bmod n$, $k < n$	$0((\lg n)^3)$

1.7.5. Các kí hiệu Legendre và Jacobi

1.7.5.1.Định nghĩa 1.29:

Cho p là một số nguyên tố lẻ và a là một số nguyên. Ký hiệu Legendre $\left(\frac{a}{p}\right)$ được xác định như sau:

$$\left(\frac{a}{p}\right)=\begin{cases}0 & p \mid a\\ 1 & a \in Q_p\\ -1 & a \in \overline{Q}_p\end{cases}$$

1.7.5.2. Các tính chất của kí hiệu Legendre

Cho p là một số nguyên tố lẻ và $a, b \in Z$. Khi đó ký hiệu Legendre có các tính chất sau:

(1) $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod p$. Đặc biệt $\left(\frac{1}{p}\right)=1$ và $\left(-\frac{1}{p}\right)=(-1)^{(p-1)/2}$

Bởi vậy $-1 \in Q_p$ nếu $p \equiv 1 \pmod 4$ và $-1 \in \overline{Q}_p$ nếu $p \equiv 3 \pmod 4$

(2) $\left(\frac{a.b}{p}\right) \equiv \left(\frac{a}{p}\right).\left(\frac{b}{p}\right)$. Bởi vậy nếu $a \in Z_p^*$ thì $\left(\frac{a^2}{p}\right)=1$.

(3) Nếu $a \equiv b \pmod p$ thì $\left(\frac{a}{p}\right)=\left(\frac{b}{p}\right)$.

(4) $\left(\frac{2}{p}\right)=(-1)^{(p^2-1)/8}$. Bởi vậy $\left(\frac{2}{p}\right)=1$ nếu $p \equiv 1$ hoặc $7 \pmod 8$ và $\left(\frac{2}{p}\right)=-1$ nếu $p \equiv 3$ hoặc $5 \pmod 8$.

(5) Luật thuận nghịch bậc 2:

Giả sử p là một số nguyên tố lẻ khác với q, khi đó:

$$\left(\frac{p}{q}\right)=\left(\frac{q}{p}\right)(-1)^{(p-1)(q-1)/4}$$

Nói một cách khác $\left(\frac{p}{q}\right)=\left(\frac{q}{p}\right)$ trừ phi cả p và q là đồng dư với $3 \pmod 4$, trong trường hợp này $\left(\frac{p}{q}\right)=-\left(\frac{q}{p}\right)$.

Dấu hiệu Jacobi là tổng quát hoá của ký hiệu Legendre đối với các số nguyên lẻ n không nhất thiết là một số nguyên tố.

1.7.5.3. Định nghĩa 1.30:

Cho $n \geq 3$ là các số nguyên tố lẻ có phân tích $n = p_1^{e_1} . p_2^{e_2} \dots p_k^{e_k}$. Khi đó ký hiệu Jacobi $\left(\frac{a}{n}\right)$ được định nghĩa là

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \dots \left(\frac{a}{p_k}\right)^{e_k}$$

Ta thấy rằng nếu n là một số nguyên tố thì ký hiệu Jacobi chính là ký hiệu Legendre.

1.7.5.4. Các tính chất của kí hiệu Jacobi

Cho $n \geq 3$ là các số nguyên tố lẻ $a, b \in Z$. Khi đó ký hiệu Jacobi có các tính chất sau:

(1) $\left(\frac{a}{n}\right) = 0,1$ hoặc -1. Hơn nữa $\left(\frac{a}{n}\right) = 0$ nếu và chỉ nếu $UCLN(a,n) \neq 1.$

(2) $\left(\frac{a.b}{n}\right) \equiv \left(\frac{a}{n}\right).\left(\frac{b}{n}\right)$. Bởi vậy $a \in Z_n^*$ thì $\left(\frac{a^2}{n}\right) = 1$

(3) $\left(\frac{a}{m.n}\right) \equiv \left(\frac{a}{m}\right).\left(\frac{a}{n}\right).$

(4) Nếu $a \equiv b(\mathrm{mod}\, n)$ thì $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right).$

(5) $\left(\frac{1}{n}\right) = 1$

(6) $\left(-\frac{1}{n}\right) = (-1)^{(n-1)/2}$. Bởi vậy $\left(-\frac{1}{n}\right) = 1$ nếu $n \equiv 1(\mathrm{mod}\, 4)$

$$\left(-\frac{1}{n}\right) = -1 \text{ nếu } n \equiv 3 (\text{mod } 4)$$

(7) $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$. Bởi vậy $\left(\frac{2}{n}\right) = 1$ nếu $n \equiv 1$ hoặc $7 (\text{mod } 8)$

$$\left(\frac{2}{n}\right) = -1 \text{ nếu } n \equiv 3 \text{ hoặc } 5 (\text{mod } 8)$$

(8) $\left(\frac{m}{n}\right) = \left(\frac{n}{m}\right)(-1)^{(m-1)(n-1)/4}$

Nói một cách khác $\left(\frac{m}{n}\right) = \left(\frac{n}{m}\right)$ trừ phi cả hai số m và n đều đồng dư với $3 (\text{mod } 4)$, trong trường hợp này $\left(\frac{m}{n}\right) = -\left(\frac{n}{m}\right)$.

Từ các tính chất của ký hiệu Jacobi ta thấy rằng n lẻ và $a = 2^e a_1$ trong đó a_1 là một số lẻ thì:

$$\left(\frac{a}{n}\right) = \left(\frac{2^e}{n}\right)\left(\frac{a_1}{n}\right) = \left(\frac{2}{n}\right)^e \left(\frac{n \bmod a_1}{a_1}\right)(-1)^{(a_1-1)(n-1)/4}$$

Từ công thức này ta có thể xây dựng thuật toán đệ quy sau để tính $\left(\frac{a}{n}\right)$ mà không cần phải phân tích n ra các thừa số nguyên tố.

1.7.5.5. Thuật toán tính toán kí hiệu Jacobi (và kí hiệu Legendre)

JACOBI (a, n)

VÀO: Số nguyên lẻ $n \geq 3$ số nguyên a, $(0 \leq a \leq n)$

RA: Ký hiệu Jacobi $\left(\frac{a}{n}\right)$ (Sẽ là ký hiệu Legendre khi n là số nguyên tố)

(1) Nếu $a = 0$ thì return (0)

(2) Nếu $a = 1$ thì return (1)

(3) Viết $a = 2^e a_1$, trong đó a_1 là một số lẻ

(4) Nếu e chẵn thì đặt $s \leftarrow 1$. Ngược lại hãy đặt $s \leftarrow 1$ nếu $n = 1$ hoặc $7(\text{mod } 8)$

(5) Nếu $n \equiv 3(\text{mod } 4)$ và $a_1 \equiv 3(\text{mod } 4)$ thì đặt $s \leftarrow -s$

(6) Đặt $n_1 \leftarrow n \bmod a_1$

(7) Return $(s.\text{JACOBI}(n_1, a_1))$

Thuật toán trên có thời gian chạy chừng $0((\lg n)^2)$ các phép toán bit.

1.7.5.6. Nhận xét (tìm các thặng dư bậc hai theo modulo của số nguyên tố p)

Cho p là một số nguyên tố lẻ . Mặc dù đã biết rằng một nửa các phần tử trong Z_p^* là các thặng dư không bậc hai theo modulo p nhưng không có một thuật toán xác định theo thời gian đa thức nào được biết để tìm.

Một thuật toán ngẫu nhiên tìm một thặng dư không bậc hai là chọn ngẫu nhiên các số nguyên $a \in Z_p^*$ cho tới khi số đó thoả mãn $\left(\frac{a}{p}\right) = -1$. Phép lặp đối với số được chọn trước khi tìm được một thặng dư bậc hai là 2 và bởi vậy thuật toán được thực hiện theo thời gian đa thức.

1.7.5.7. Ví dụ tính toán kí hiệu Jacobi

Cho $a = 158$ và $n = 235$. Thuật toán trên tính $\left(\frac{158}{235}\right)$ như sau:

$$\left(\frac{158}{235}\right) = \left(\frac{2}{235}\right)\left(\frac{79}{235}\right) = (-1)\left(\frac{235}{79}\right)(-1)^{78 \cdot 234/4} = \left(\frac{77}{79}\right)$$

$$= \left(\frac{77}{79}\right)(-1)^{76 \cdot 78/4} = \left(\frac{2}{77}\right) = -1$$

Khác với ký hiệu Legendre, ký hiệu Jacobi $\left(\frac{a}{n}\right)$ không cho biết liệu a có phải là một thặng dư bậc 2 theo modulo n hay không. Sự thực là nếu $a \in Q_n$ thì $\left(\frac{a}{n}\right) = 1$

Tuy nhiên $\left(\frac{a}{n}\right) = 1$ thì không có nghĩa là $a \in Q_n$.

1.7.5.8. Ví dụ (các thặng dư bậc 2 và không thặng dư bậc 2)

Bảng 1-5. Các ký hiệu Jacobi của các phần tử trong Z_{21}^*

$a \in Z_{21}^*$	1	2	4	5	8	10	11	13	16	17	19	20
$a^2 \bmod n$	1	4	16	4	1	16	16	1	4	16	4	1
$\left(\frac{a}{3}\right)$	1	–1	1	–1	–1	1	–1	1	1	–1	1	–1
$\left(\frac{a}{7}\right)$	1	1	1	–1	1	–1	1	–1	1	–1	–1	–1
$\left(\frac{a}{21}\right)$	1	–1	1	1	–1	–1	–1	–1	1	1	–1	–1

Bảng 1.5 liệt kê các phần tử trong Z_{21}^* và các ký hiệu Jacobi của chúng. Từ ví dụ trong phần c ta có $Q_{21} = \{1,4,16\}$. Ta thấy rằng $\left(\frac{5}{21}\right) = 1$ nhưng $5 \notin Q_{21}$.

1.7.5.9. Định nghĩa 1.31

Cho $n \geq 3$ là các số nguyên tố lẻ và cho $J_n = \left\{a \in Z_n^* \middle| \left(\frac{a}{n}\right) = 1\right\}$ tập các thặng dư giả bậc 2 theo modulo n (Ký hiệu $\hat{Q}_n$) được định nghĩa là tập $J_n - Q_n$.

1.7.5.10. Định lí 1.22.

Cho $n = p.q$ là tích của hai số nguyên tố lẻ khác nhau. Khi đó $|Q_n| = |\widetilde{Q}_n| = (p-1)(q-1)/4$ tức là một nửa các phần tử trong J_n là các thặng dư giả bậc hai.

1.7.6. Căn nguyên thủy

- **Thuật toán tính căn bậc hai modulo số nguyên tố p**

VÀO: số nguyên tố lẻ p và số nguyên a, $1 \le a \le p - 1$

RA: hai căn bậc hai của a modulo p, giả thiết rằng a là thặng dư bình phương modulo p

1. Tính kí hiệu Legendre $\left(\frac{a}{p}\right)$. Nếu $\left(\frac{a}{p}\right)$= -1 thì trả về "a không có căn bậc hai modulo p" và dừng
2. Chọn ngẫu nhiên b, $1 \le b \le p - 1$ cho đến khi tìm được b với $\left(\frac{b}{p}\right)$= -1 (b là không thặng dư bình phương modulo p)
3. Bằng cách chia liên tiếp cho 2, viết $p - 1 = 2^s t$ với t lẻ
4. Tính a^{-1} mod p bằng thuật toán Euclide mở rộng
5. $c \leftarrow b^t \bmod p$ và $r \leftarrow a^{(t+1)/2} \bmod p$
6. For i from 1 to s – 1 do
    1. Tính $d = (r^2.a^{-1})^{2^{s-i-1}} \bmod p$
    2. Nếu $d \equiv -1 \bmod p$ thì đặt $r \leftarrow r.c \bmod p$
    3. Đặt $c \leftarrow c^2 \bmod p$
7. Return (r, -r)

- **Thuật toán tính căn bậc hai modulo p khi $p \equiv 3 \bmod 4$**

VÀO: số nguyên tố lẻ p với $p \equiv 3 \bmod 4$ và $a \in Q_p$

RA: hai căn bậc hai của a modulo p

1. Tính $r = a^{(p+1)/4} \bmod p$
2. Return (r, -r)

- **Thuật toán tính căn bậc hai modulo p khi p ≡ 5 mod 8**

VÀO: số nguyên tố lẻ p với $p \equiv 5 \bmod 8$ và $a \in Q_p$

RA: hai căn bậc hai của a modulo p

1. Tính $d = a^{(p+1)/4} \bmod p$
2. Nếu d = 1 thì $r = a^{(p+3)/8} \bmod p$
3. Nếu d = -1 thì $r = 2a(4a)^{(p-5)/8} \bmod p$
4. Return (r, -r)

- **Thuật toán tính căn bậc hai modulo n, với n là hợp số**

VÀO: số nguyên n, các nhân tử nguyên tố của nó p và q (trong đó $p \equiv 3 \bmod 4$, $q \equiv 3 \bmod 4$), $c \in Q_n$

RA: bốn căn bậc hai của c modulo n

1. Dùng thuật toán Euclide mở rộng tìm a, b: ap + bq = 1
2. Tính

$r = c^{(p+1)/4} \bmod p$

$s = c^{(q+1)/4} \bmod p$

$x = (aps + bqr) \bmod n$

$y = (aps - bqr) \bmod n$

3. Return (±x, ±y)

1.7.7. Các số nguyên Blum

Định nghĩa 1.32.

Số nguyên Blum là một hợp số có dạng $n = p.q$, trong đó p và q là các số nguyên tố khác nhau và thoả mãn:

$$p \equiv 3 \bmod 4$$
$$q \equiv 3 \bmod 4$$

Định lý 1.23:

Cho $n = p.q$ là một số nguyên Blum và cho $a \in Q_n$. Khi đó a có đúng 4 căn bậc hai modulo n và chỉ có một số nằm trong Q_n.

Định nghĩa 1.33:

Cho n là một số nguyên Blum và cho $a \in Q_n$. Căn bậc hai duy nhất của a nằm trong Q_n được gọi là căn bậc hai chính $a \bmod n$.

Ví dụ (Số nguyên Blum).

Đối với số nguyên Blum $n = 21$. Ta có $J_n = \{1,4,5,16,17,20\}$ và $\tilde{Q}_n = \{5,17,20\}$. Bốn căn bậc 2 của $a = 4$ là 2, 5, 16 và 19, trong đó chỉ có 16 là cũng nằmg trong Q_n. Bởi vậy 16 là căn bậc 2 chính của $4 \bmod 21$.

Định lý 1.24:

Nếu $n = p.q$ là một số nguyên Blum thì ánh xạ.

$f : Q_n \to Q_n$ được xác định bởi $f(x) = x^2 \bmod n$ là một phép hoán vị.

Ánh xạ ngược của f là: $f^{-1}(x) = x^{((p-1)(q-1)+4/8)} \bmod n$.

1.8. CÂU HỎI ÔN TẬP

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 $\mathcal{K} = C = \mathcal{P} = \{1,...,n\}$. 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.

2. Hãy chứng tỏ rằng mã Affine có độ mật hoàn thiện
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.
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.
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.
6. Chứng minh rằng một hệ mật có độ mật hoàn thiện khi và chỉ khi $H(P|C) = H(P)$.
7. Chứng minh rằng trong một hệ mật $H(K|C) \ge H(P|C)$ (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).
8. 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), $H(K|C)$ và $H(P|C)$.

9. Sử dụng thuật toán Euclide mở rộng để tìm ước chung lớn nhất của hai số a = 1573, b = 308.

10. Hãy tính $3^{22} \bmod 23$ bằng cách dùng thuật toán nhân và bình phương có lặp.

11. Hãy tính các căn bậc hai của $12 \bmod 37$.

12. Tìm tất cả các phần tử nguyên thuỷ của nhóm nhân Z_{19}^*.

13. Tìm phần tử nghịch đảo của 3 trong Z_{31}^*.

14. Với $m, n, s \in N$ và p_i là các số nguyên tố. Hãy chứng minh các tính chất sau của hàm φ-Euler

    i. $\varphi(p^s) = p^s\left(1 - \frac{1}{p}\right)$.

    ii. $\varphi(m,n) = \varphi(m)\varphi(n)$ nếu $UCLN(m,n) = 1$.

    iii. $\varphi(n) = m\left(1 - \frac{1}{p_1}\right)...\left(1 - \frac{1}{p_r}\right)$ trong đó $m = p_1^{e_1}...p_r^{e_1}$ là phân tích của m thành tích của thừa số nguyên tố.

15. Hãy tính $\varphi(490)$ và $\varphi(768)$.

16. Giải hệ phương trình đồng dư sau:

$$5x \equiv 20 \bmod 6$$
$$6x \equiv 6 \bmod 5$$
$$4x \equiv 5 \bmod 77$$

17. Hãy dùng thuật toán Euclide mở rộng để tính các phần tử nghịch đảo sau:

a. $17^{-1} \bmod 101$

b. $357^{-1} \bmod 1234$

c. $3125^{-1} \bmod 9987$

18. Ta nghiên cứu một số tính chất của các phần tử nguyên thuỷ:

(a) 97 là một số nguyên tố. Hãy chứng minh rằng $x \neq 0$ là một phần tử nguyên thuỷ theo $\text{modulo } 97$ khi và chỉ khi:

$$x^{32} \neq 1 \bmod 97 \text{ và } x^{48} \neq 1 \bmod 97$$

(b) Hãy dùng phương pháp này để tìm phần tử nguyên thuỷ nhỏ nhất theo modulo 97.

(c) Giả sử p là một số nguyên tố và $p-1$ có phân tích ra luỹ thừa của các nguyên tố sau:

$$p-1 = \prod_{i=1}^{n} p_i^{e_i}$$

Ở đây p_i là các số nguyên tố khác nhau. Hãy chứng tỏ rằng $x \neq 0$ là một phần tử nguyên thuỷ theo $\text{modulo } p$ khi và chỉ khi $x^{(p-1)/p_i} \neq 1 \bmod p$ với $1 \leq i \leq n$.

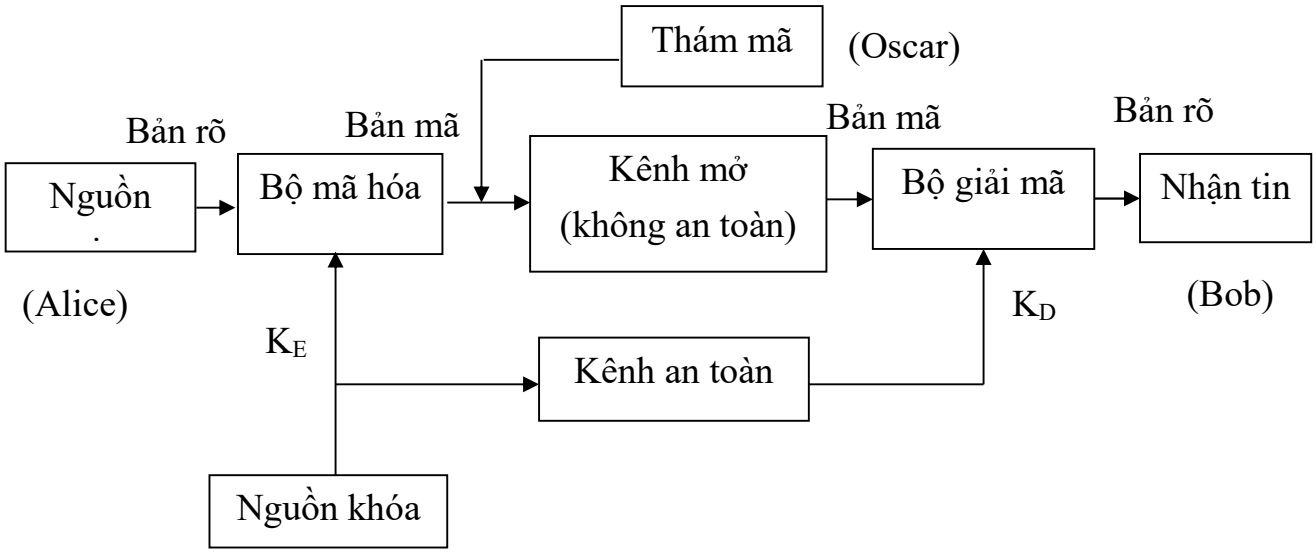
CHƯƠNG 2 CÁC HỆ MẬT KHÓA BÍ MẬT

Có ba phương pháp chính trong mật mã khoá bí mật (mật mã khoá riêng hay mật mã cổ điển):

- 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 CỦA MỘT HỆ TRUYỀN TIN MẬT



Hình 2-1. Sơ đồ khối của hệ truyền tin mật

Định nghĩa 2.1:

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

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

b) 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 $e_k \in \mathcal{E}$

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

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

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

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

2.2. CÁC HỆ MẬT THAY THẾ ĐƠN GIẢN

2.2.1. Các hệ mật thay thế đơn biểu

2.2.1.1.Mã dịch vòng (MDV)

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

$$e_k(x) = x + k \bmod 26$$
$$d_k(y) = y - k \bmod 26$$
$$(x, y \in Z_{26})$$

Hình 2-2. Mã dịch vòng

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:

Kí 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
Kí 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 meetmeatsunset.

Trước tiên, ta biến đổi bản rõ thành dãy các số nguyên theo bảng trên:

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:

RJJYRJFYXZSXJY

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 bảo độ mật.

2.2.1.2.Mã Affine

Trong mã Affine, ta giới hạn chỉ xét các hàm mã có dạng:

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

$a, b \in Z_{26}$. 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 $y \in Z_{26}$, ta muốn có đồng nhất thức sau:

$$ax + b \equiv y \pmod{26}$$

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

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

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

$$ax \equiv y \pmod{26} \qquad (y \in Z_{26})$$

Ta biết rằng, phương trình này có một nghiệm duy nhất đối với mỗi y khi và chỉ khi $UCLN(a, 26) = 1$ (ở đây hàm UCLN là ước chung lớn nhất của các biến của nó). Trước tiên ta giả sử rằng, $UCLN(a, 26) = d > 1$. Khi đó, đồng dư thức $ax \equiv 0 \pmod{26}$ sẽ có ít nhất hai nghiệm phân biệt trong 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 2.2: Do $UCLN(4, 26) = 2$ 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 kì $x \in Z_{26}$.

Ta giả thiết $UCLN(a, 26) = 1$. Giả sử với x_1 và x_2 nào đó thoả mãn:

$$ax_1 \equiv ax_2 \pmod{26}$$

Khi đó:

$$a(x_1 - x_2) \equiv 0 \pmod{26}$$

bởi vậy

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

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

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

tức là

$$x_1 \equiv x_2 \pmod{26}$$

Tới đây ta đã chứng tỏ rằng, nếu $UCLN(a, 26) = 1$ thì một đồng dư thức dạng $ax \equiv y \pmod{26}$ chỉ có (nhiều nhất) một nghiệm trong Z_{26}. Do đó, nếu ta cho x thay đổi trên Z_{26} thì $ax \bmod 26$ sẽ nhận được 26 giá trị khác nhau theo modulo 26 và đồng dư thức $ax \equiv y \pmod{26}$ 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* $x \in Z_m$ *với mọi* $b \in Z_m$ *khi và chỉ khi* $UCLN(a, m) = 1$.

Vì $26 = 2 \times 13$ nên các giá trị $a \in Z_{26}$ thoả mãn $UCLN(a, 26) = 1$ là a = 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23 và 25. Tham số b có thể là một phần tử bất kỳ trong 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.* $UCLN(a, m) = 1$ *thì ta nói rằng* a *và* m *là nguyên tố cùng nhau . Số các số nguyên trong* Z_m *nguyên tố cùng nhau với* m *thường được ký hiệu là* $\phi(m)$ *(hàm này được gọi là hàm phi-Euler) .*

Một kết quả quan trọng trong lý thuyết số cho ta giá trị của $\phi(m)$ 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 $$m = \prod_{i=1}^{n} p_i^{e_i} \qquad (2.1)$$

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

$$\phi(m) = \prod_{i=1} \left(p_i^{e_i} - p_i^{e_i - 1} \right) \qquad (2.2)$$

Định lý này cho thấy rằng, số khoá trong mã Affine trên Z_m bằng $m\phi(m)$, trong đó $\phi(m)$ 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 $m = 60, \phi(60) = 2 \times 2 \times 4 = 16$ 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 $UCLN(a, 26) = 1$. Để giải mã cần giải phương trình đồng

dư $y \equiv ax + b \pmod{26}$ theo x. Từ thảo luận trên thấy rằng, phương trình này có một nghiệm duy nhất trong 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 $a \in Z_m$. *Phần tử nghịch đảo (theo phép nhân) của* a *là phần tử* $a^{-1} \in Z_m$ *sao cho* $a.a^{-1} = a^{-1}.a = 1 \pmod{m}$.

Bằng các lý luận tương tự như trên, có thể chứng tỏ rằng a có nghịch đảo theo modulo m khi và chỉ khi $UCLN(a, m) = 1$, 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 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 Z_m với m tuỳ ý. Tuy nhiên, trong 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: $7 \times 5 = 105 \equiv 1 \bmod 26$, bởi vậy $7^{-1} = 15$).

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

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

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

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

Áp dụng tính kết hợp của phép nhân modulo:

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

Kết quả là $x \equiv a^{-1}(y-b) \pmod{26}$. 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 .3 cho mô tả đầy đủ về mã Affine.

Cho $\mathcal{P} = C = Z_{26}$ và giả sử: $K = \{ (a, b) \in Z_{26} \times Z_{26} : UCLN(a, 26) = 1 \}$

Với $k = (a, b) \in \mathcal{K}$ ta định nghĩa:

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

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

Hình 2-3. Mã Affine

Sau đây là một ví dụ nhỏ.

Ví dụ 2.3:

Giả sử $k = (7, 3)$. Như đã nêu ở trên, $7^{-1} \bmod 26 = 15$. 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 Z_{26}. Ta sẽ kiểm tra liệu $d_k(e_k(x)) = x$ với mọi $x \in Z_{26}$ không?. Dùng các tính toán trên Z_{26}, ta có:

$$\begin{aligned} d_k(e_k(x)) &= d_k(7x+3) \\ &= 15(7x+3) - 19 \\ &= x + 45 - 19 \\ &= x \end{aligned}$$

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.

2.2.2. Các phép thay thế đơn giản khác

Mã thay thế (MTT)

Cho $\mathcal{P} = C = Z_{26}$. K chứa mọi hoán vị có thể có của 26 kí tự từ 0 đến 25. Với mỗi phép hoán vị $\pi \in K$, 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

Hình 2-4. Mã thay thế

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).

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

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

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:

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

Ví dụ 2.4:

Với phép thay thế trên, từ bản rõ:

meetmeatsunset

ta thu được bản mã sau:

THHMTHXMVUSVHM

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.

2.3. CÁC HỆ MẬT THAY THẾ ĐA BIỂU

2.3.1. Hệ mật Vigenere

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 2.5:

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:

meetmeatsunset

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:

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

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

OMTAQVCBHBRJGB

Ta có thể mô tả mật mã Vigenere như sau:

Cho m là một số nguyên dương cố dịnh nào đó. Ta định nghĩa $\mathcal{P} = C = \mathcal{K} = (Z_{26})^n$

Với khóa $k = (k_1, k_2, ..., k_m)$ ta xác định:

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

và

$d_k(y_1, y_2, ..., y_m) = (y_1 - k_1, y_2 - k_2, ..., y_m - k_m)$

trong đó tất cả các phép toán được thực hiện trong 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.

2.3.2. Hệ mật 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 $\mathcal{P} = C = (Z_{26})^m$. 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:

$$y_1 = 11x_1 + 3x_2$$
$$y_2 = 8x_1 + 7x_2$$

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

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

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

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

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 $A = (x_{i,j})$ là một ma trận cấp $l \times m$ và $B = (b_{l,k})$ là một ma trận cấp $m \times n$ thì tích ma trận $AB = (c_{l,k})$ được định nghĩa theo công thức :

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

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 $l \times n$.

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

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

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

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

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

$$yk^{-1} = (xk)k^{-1} = x(kk^{-1}) = xI_m = x$$

(Chú ý: sử dụng tính chất kết hợp)

Có thể thấy rằng, ma trận mã hoá ở trên có nghịch đảo trong 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 2.6:

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:

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

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

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

$(3 \quad 4).k^{-1} = (9 \quad 20)$ và $(11 \quad 22).k^{-1} = (11 \quad 24)$

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 2.3:

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

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

Nhận xét: Định thức của một ma trận vuông cấp m x 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 $\det I_m = 1$ và quy tắc nhân $\det(AB) = \det A \times \det B$.

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 Z_{26}. Kết quả

tương ứng là ma trận k có nghịch đảo theo modulo 26 khi và chỉ khi $\text{UCLN}(\det k, 26) = 1$.

Sau đây sẽ chứng minh ngắn gọn kết quả này.

Trước tiên, giả sử rằng $\text{UCLN}(\det k, 26) = 1$. Khi đó $\det k$ có nghịch đảo trong Z_{26}. Với $1 \le i \le m$, $1 \le j \le m$, 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 $(-1)^{i+j} \det k_{ji}$ (k^* được gọi là ma trận bù đại số của k). Khi đó, có thể chứng tỏ rằng:

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

Bởi vậy k là khả nghịch.

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

$$1 = \det I = \det\left(k\, k^{-1}\right) = \det k \ \det k^{-1}$$

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

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

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

$$\det\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix} = 11\times 7 - 8\times 3 \bmod 2$$
$$= 77 - 24 \bmod 26 = 53 \bmod 26$$
$$= 1$$

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.

Bây giờ ta sẽ mô tả chính xác mật mã Hill trên Z_{26} (hình 2.6).

Cho m là một số nguyên dương cố định. Cho $\mathcal{P} = C = \left(Z_{26}\right)^m$ và cho:

$\mathcal{K}$ = {các ma trận khả nghịch cấp $m \times m$ trên Z_{26}}

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

$$e_k(x) = xk$$

Hình 2-6. Mật mã Hill

2.3.3. Hệ mật Playfair

Mật mã Playfair hay hình vuông Playfair là một kĩ thuật mã hoá đối xứng thủ công và là thuật toán thay thế chữ ghép đầu tiên. Hệ mật này được Charles Wheatstone phát minh ra năm 1854 nhưng lại được gọi theo tên của Lord Playfair người đã phổ biến để sử dụng làm mật mã.

Kĩ thuật này mã hoá từng cặp kí tự (bộ ghép) thay cho các kí tự đơn trong mã hoá thay thế đơn và cách sử dụng phức tạp hơn mã hoá Vigenere. Tấn công Playfair khó vì việc phân tích tần suất vẫn thường sử dụng cho mật mã thay thế đơn không dùng được tuy nhiên phân tích tần suất các bộ chữ ghép thì vẫn có thể nhưng khó hơn nhiều và nói chung phải cần số lượng bản mã rất lớn.

Cách dùng Playfair

Playfair dùng một bảng 5 × 5 chứa từ hoặc cụm từ khóa. Ta cần phải nhớ từ khóa và 4 quy tắc thực hiện.

Tạo bảng khóa chứa 25 chữ cái khác nhau (bỏ chữ "Q" trong bảng chữ cái hoặc phiên bản khác thì coi chữ "I" và chữ "J" thuộc cùng một ô trong bảng khóa), trước tiên lấp đầy bảng bởi các chữ cái của từ khóa (bỏ các chữ cái lặp lại) rồi lấp các chữ cái còn lại của bảng chữ cái vào các chỗ trống của bảng khóa theo thứ tự. Khóa có thể được viết ở các hàng đầu của bảng, từ trái qua phải.

Mã hóa thông điệp, tách thông điệp thành các nhóm gồm 2 kí tự rồi ánh xạ chúng vào bảng khóa.

Bốn quy tắc:

1) Xen chữ "X" vào giữa hai chữ cái giống nhau (hoặc chèn vào sau nếu chỉ còn lại một chữ cái cuối cùng) của bản rõ rồi mã hóa cặp mới và tiếp tục thực hiện.

Chú ý: Phiên bản khác thì chèn chữ "Q" thay cho chữ "X".

2) Nếu hai chữ cái của một cặp xuất hiện trên cùng một hàng của khóa thì thay thế chúng bằng các chữ cái ở ngay bên phải tương ứng (nếu chữ cái nằm ở tận cùng bên phải của hàng thì thay bằng chữ cái đầu tiên của hàng đó)

3) Nếu hai chữ cái của cặp xuất hiện trên cùng một cột của khóa thì thay thế chúng bởi các chữ ở ngay dưới tương ứng (nếu chữ cái nằm ở tận cùng bên dưới của cột thì thay thế bởi chữ cái đầu tiên của cột đó)

4) Nếu hai chữ cái của cặp không cùng hàng và cột thì thay thế chúng bởi các chữ cái trên cùng hàng tương ứng và ở các góc của hình chữ nhật mà hai chữ cái của cặp này tạo nên trên khóa.

Ví dụ

Dùng khóa "playfair example" để mã hóa bản rõ "Hide the gold in the tree stump".

Khóa sẽ được bố trí như sau:

P L A Y F

I R E X M

B C D G H

J K N O S

T U V W Z

Ở đây các kí tự lặp lại sẽ bị bỏ đi, sau đó thêm các chữ cái trong bảng chữ cái mà chưa xuất hiện trong khóa sau khi đã bỏ các chữ lặp lại vào để lấp đầy ô trống của bảng khóa 5×5.

Bản rõ được tách như sau:

HI DE TH EG OL DI NT HE TR EE ST UM P

Nhưng có cặp EE thì ta phải xen chữ X vào giữa, kết quả được là:

HI DE TH EG OL DI NT HE TR EX ES TU MP

Mã hóa:

Chiếu từng cặp của bản rõ sau khi đã tách vào bảng khóa theo các quy tắc mã hóa ta được:

HI→ BM (Khác hàng, khác cột)

DE→ND (Cùng cột)

TH→ZB (Khác hàng, khác cột)

EG→XD (Khác hàng, khác cột)

OL→KY (Khác hàng, khác cột)

DI→BE (Khác hàng, khác cột)

NT→JV (Khác hàng, khác cột)

HE→DM (Khác hàng, khác cột)

TR→UI (Khác hàng, khác cột)

EX→XM (Cùng hàng)

ES→MN (Khác hàng, khác cột)

TU→UV (Cùng hàng)

MP→IF (Khác hàng, khác cột)

Vậy bản mã là: "BMNDZBXDKYBEJVDMUIXMMNUVIF”

Giải mã: Vẫn dùng khóa giống như mã hóa còn 4 quy tắc thì thay thế theo chiều ngược lại.

Thám mã Playfair

Việc lấy được khóa là tương đối dễ nếu biết cả bản rõ và bản mã. Nếu chỉ biết bản mã thì phương pháp giải mã theo vét cạn bao gồm việc tìm kiếm qua không gian khóa và phân tích tần suất của các chữ ghép trong ngôn ngữ giả định của thông điệp gốc.

Giải mã Playfair dựa vào sự liên quan của các chữ cái nên việc xác định các xâu bản rõ ứng cử viên dễ dàng hơn. Đặc biệt là chữ ghép Playfair và ngược của chữ ghép đó (ví dụ AB và BA) sẽ giải mã thành cùng một kiểu mẫu chữ cái trong bản rõ (Ví dụ RE và ER). Trong tiếng Anh, có rất nhiều từ chứa những bộ chữ ghép ngược này như REceivER và DEpartED. Việc xác định những bộ chữ ghép ngược mà gần nhau trong bản mã và ghép kiểu mẫu thành một danh sách các từ rõ đã biết chứa kiểu mẫu là đơn giản từ đó đưa ra được các xâu bản rõ có thể rồi sẽ xác định khóa.

Một cách tiếp cận khác là phương pháp Shotgun hill climbing. Bắt đầu với hình vuông các chữ cái ngẫu nhiên sau đó thực hiện những sự thay đổi nhỏ (ví dụ như chuyển các chữ cái, các hàng hay phản xạ toàn bộ hình vuông) để thấy được nếu bản rõ ứng cử viên giống bản rõ chuẩn hơn so với trước khi thay đổi (có thể bằng cách so sánh các nhóm ba thành lược đồ tần suất đã biết). Nếu hình vuông mới có sự cải tiến thì nó sẽ được chấp nhận và sau sẽ được biến đổi thêm để tìm ra một ứng cử viên tốt hơn. Cuối cùng là dù chọn phương pháp phân loại nào thì cũng tìm ra bản rõ hoặc một văn bản rất gần bản rõ với khả năng đúng là lớn

nhất. Máy tính có thể chấp nhận thuật toán này để phá các mật mã Playfair với số lượng văn bản tương đối nhỏ.

Phương pháp Playfair thường được áp dụng trong việc giải các trò chơi đố các ô chữ.

2.4. CÁC HỆ MẬT THAY THẾ KHÔNG TUẦN HOÀN

2.4.1. Hệ mật khóa tự sinh

Cho $\mathcal{P} = C = \mathcal{K} = \mathcal{L} = Z_{26}$

Cho $z_1 = k$ và $z_i = x_{i-1}$ $(i \geq 2)$

Với $0 \leq z \leq 25$ ta xác định

$e_z(x) = x + z \bmod 26$

$d_z(y) = y - z \bmod 26$

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

Hình 2-7. Mật mã khóa tự sinh

Lý do sử dụng thuật ngữ "khoá tự sinh" là ở chỗ bản rõ được dùng làm khoá (ngoài "khoá khởi thuỷ" ban đầu k).

Sau đây là một ví dụ minh hoạ.

Ví dụ 2.7:

Giả sử khoá là k = 8 và bản rõ là *rendezvous*. Trước tiên, ta biến đổi bản rõ thành dãy các số nguyên:

17 4 13 3 4 25 21 14 20 18

Dòng khoá như sau:

8 17 4 13 3 4 25 21 14 20

Bây giờ ta cộng các phần tử tương ứng rồi rút gọn theo modulo 26:

25 21 17 16 7 3 20 9 8 12

Bản mã ở dạng ký tự là: ZVRQHDUJIM .

Bây giờ ta xem Bob giải mã bản mã này như thế nào. Trước tiên, Bob biến đổi xâu kí tự thành dãy số:

25 21 17 16 7 3 20 9 8 12

Sau đó anh ta tính:

$$x_1 = d_8(25) = 25 - 8 \bmod 26 = 17$$

và $$x_2 = d_{17}(21) = 21 - 17 \bmod 26 = 4$$

và cứ tiếp tục như vậy.

Mỗi khi Bob nhận được một ký tự của bản rõ, cô ta sẽ dùng nó làm phần tử tiếp theo của dòng khoá. Dĩ nhiên là mã dùng khoá tự sinh là không an toàn do chỉ có 26 khoá.

2.4.2. Hệ mật Vernam

Giả sử n ≥1 là số nguyờn và P = C = K = $(Z_2)^n$. Với K thuộc $(Z_2)^n$, ta xác định $e_K(x)$ là tổng véc tơ 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 hóa là đồng nhất với phép giải mã. Nếu y = $(y_1,..., y_n)$ thì:

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

Hình 2-8. Hệ mật OTP

Sử dụng định lý 1.3, 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 đó 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 |K | ≥ | 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 hóa 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 bít 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 rói 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.

2.5. CÁC HỆ MẬT HOÁN VỊ

Khác với MTT, ý tưởng của mã hoán vị (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 2.8:

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: asecondclasscarriageonthetrain

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

a sec on|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 $\mathcal{P} = C = (Z_{26})^m$ và cho là tất cả các hoán vị có thể có của {1, 2, …,m}

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

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

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

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

Hình 2-9. Mã hoán vị

2.6. CÁC HỆ MẬT 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 $\mathcal{C} = \mathcal{P}$: 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:

$$e_{(k_1, k_2)}(x) = e_{k_2}(e_{k_1}(x))$$

và quy tắc giải mã:

$$d_{(k_1, k_2)}(y) = d_{k_1}(d_{k_2}(y))$$

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

$$\begin{aligned} d_{(k_1, k_2)}(e_{(k_1, k_2)}(x)) &= d_{(k_1, k_2)}(e_{k_2}(e_{k_1}(x))) \\ &= d_{k_1}(d_{k_2}(e_{k_2}(e_{k_1}(x)))) \\ &= d_{k_1}(e_{k_1}(x)) \\ &= x \end{aligned}$$

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} 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 2.10 sau.

Giả sử $\mathcal{P} = \mathcal{C} = Z_{26}$ và giả sử:

$K = \{ a \in Z_{26} : \text{UCLN}(a, 26) = 1 \}$

Với $a \in \mathcal{K}$, ta xác định: $e_a(x) = ax \bmod 26$

Hình 2-10. Mã nhân

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 $k \in Z_{26}$ và quy tắc giải mã tương ứng là $e_k(x) = x + k \bmod 26$. Còn khoá trong hệ mã nhân là phần từ $a \in Z_{26}$ sao cho $\text{UCLN}(a, 26) = 1$. Quy tắc mã tương ứng là $e_a(x) = a \bmod 26$. Bởi vậy, một khoá trong mã tích $M \times S$ có dạng (a, k), trong đó

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

Đây chính là định nghĩa về khoá trong hệ mã Affine. Hơn nữa, xác suất của một khoá trong hệ mã Affine là: $1/312 = (1/12)\times(1/26)$. 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 :

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

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 $UCLN(a, 26) = 1$, 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 này là hệ 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 lớn hơn 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.

Nhật 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:

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

(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. CHUẨN MÃ DỮ LIỆU (DES)

2.7.1. 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 sau là 1998. Tới tháng 10.2000 DES đã không còn là chuẩn mã dữ liệu nữa.

2.7.2. 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 56 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:

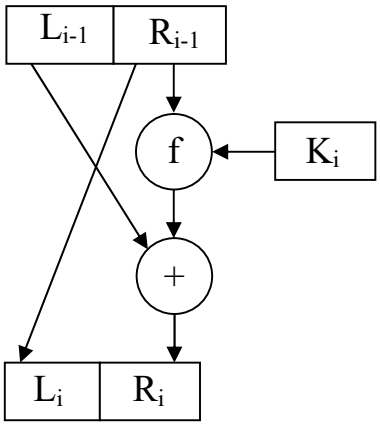
$x_0 = IP(x) = L_0R_0$, trong đó L_0 gồm 32 bit đầu và R_0 là 32 bit cuối.

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

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

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 $k_1, k_2, \ldots, k_{16}$ 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).

$k_1, k_2, \ldots, k_{16}$ sẽ tạo thành bảng khoá. Một vòng của phép mã hoá được mô tả trên hình 2.11



Hình 2-11. 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 $y = IP^{-1}(R_{16}L_{16})$. Hãy chú ý thứ tự đã đảo của L_{16} và R_{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. $E(A)$ gồm 32 bit của A (được hoán vị theo cách cố định) với 16 bit xuất hiện hai lần.

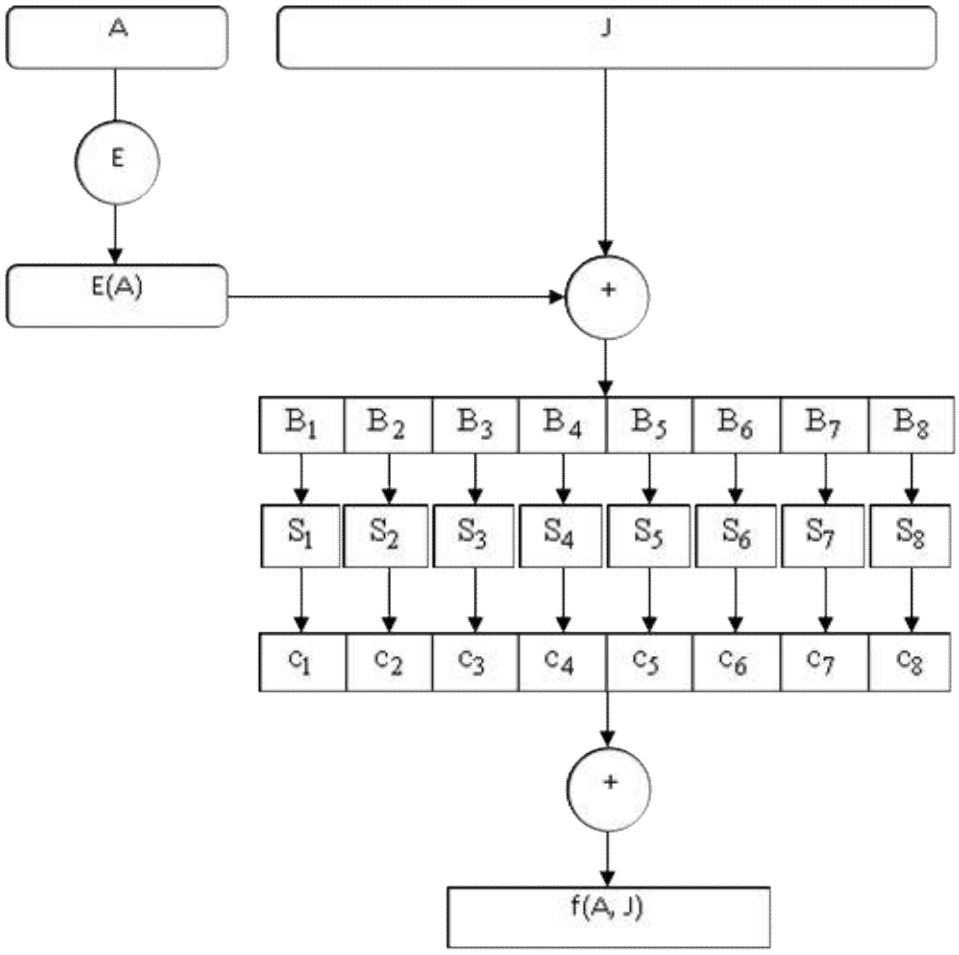
2. Tính $E(A) \oplus J$ và viết kết quả thành một chuỗi 8 xâu 6 bit là

$B_1B_2B_3\ B_4B_5B_6B_7B_8$.

3. Bước tiếp theo dùng 8 bảng $S_1, S_2, \ldots, S_8$ (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_j $(0 \le r \le 3)$ và bốn bit $(b_2b_3b_4b_5)$ xác định biểu diễn nhị phân của cột c của S_j $(0 \le c \le 15)$. 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 $C = C_1C_2 \ldots C_8$ 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 2-12. Hàm f của DES

Hàm f được mô tả trong hình 2.12. 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.

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:

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:

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 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 là:

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
S4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	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
S7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8															

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Và phép hoán vị P có dạng:

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

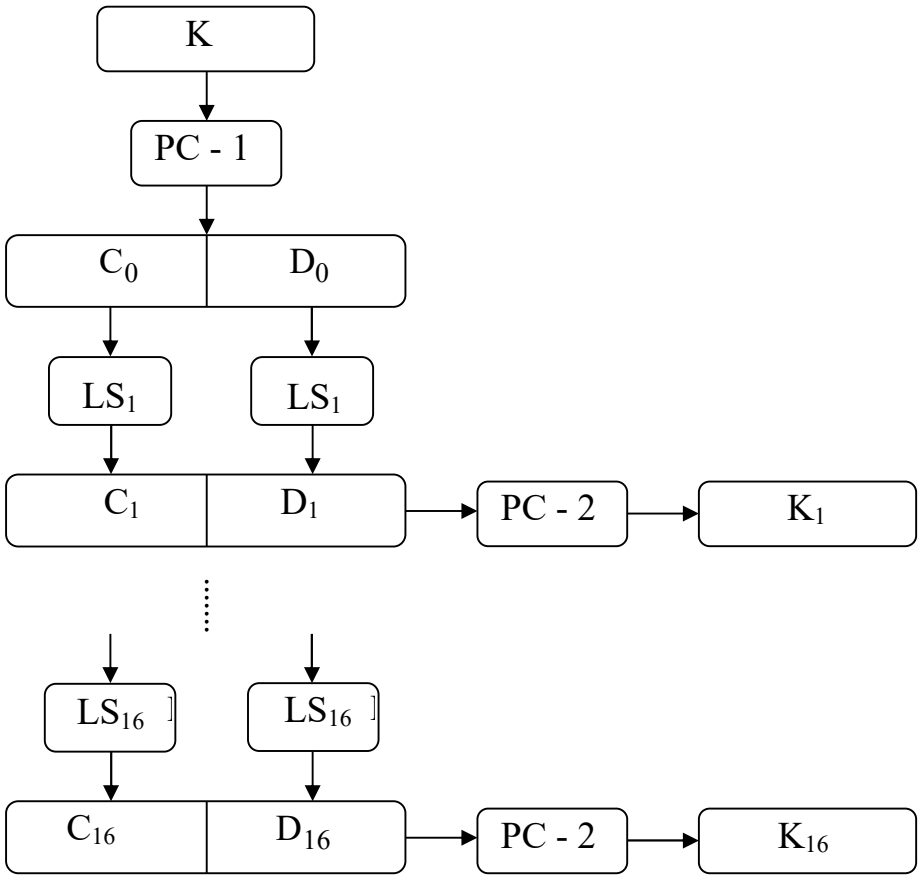
1. Với một khoá k 64 bit cho trước, ta loại bỏ các bit kiểm tra tính chẵn lẻ và hoán vị các bit còn lại của k theo phép hoán vị cố định PC-1. Ta viết:

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

2. 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 2.13



Hình 2-13. Tính bảng khóa DES

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

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

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.

2.7.3. 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- hệ đếm 16):

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 = 11001100000000001100110011111111

$L_1 = R_0$ = 11110000101010101111000010101010

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

$E(R_0)$ = 011110100001010101010101011110100001010101010101

K_1 = 000110110000000101110111111111110001110000011100 10

$E(R_0) \oplus K_1$ = 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

$L_3 = R_2$ = 11001100000000010111011100001001

$E(R_2)$ = 111001011000000000000010101110101110100001010011

K_3 = 010101011111110010001010010000101100111110011001

$E(R_2) \oplus K_3$ = 101100000111110010001000111110000010011111001010

S-box outputs 00100111000100001110000101101111

$f(R_2,K_3)$ = 01001101000101100110111010110000

$L_4 = R_3$ = 10100010010111000000101111110100

$E(R_3)$ =010100000100001011111000000001010111111111010100

K_4 = 011100101010110111010110110110110011010100011101

$E(R_3) \oplus K_4$ = 001000101110111100101110110111100100101010110100

S-box outputs 00100001111011011001111100111010

$f(R_3,K_4)$ = 10111011001000110111011101001100

$L_5 = R_4$ = 01110111001000100000000001000101

$E(R_4)$ = 101110101110100100000100000000000000001000001010

K_5 = 011111001110110000000111111010110101001110101000

$E(R_4) \oplus K_5$ = 110001100000010100000011111010110101000110100010

S-box outputs 01010000110010000011000111101011

$f(R_4,K_5)$ = 00101000000100111010110111000011

$L_6 = R_5$ = 10001010010011111010011000110111

$E(R_5)$ = 110001010100001001011111110100001100000110101111

K_6 = 011000111010010100111110010100000111101100101111

$E(R_5) \oplus K_6$ =101001101110011101100001100000001011101010000000

S-box outputs 01000001111100110100110000111101

$f(R_5,K_6)$ = 10011110010001011100110100101100

$L_7 = R_6$ = 11101001011001111100110101101001

$E(R_6)$ = 11110101001010110000111111100101101010110101010011

K_7 = 11101100100001001011011111110110000110001011110 0

$E(R_6) \oplus K_7$ = 00011001101011111011100000010011101100111110 1111

S- box outputs 00010000011101010100000010101101

$f(R_6,K_7)$ = 10001100000001010001110000100111

$L_8 = R_7$ = 00000110010010101011101000010000

$E(R_7)$ = 00000000110000100101010101011111010000001010000 0

K_8 = 11110111100010100011101011000001001110111111111011

$E(R_7) \oplus K_8$ = 11110111010010000110111110011110011110110101 1011

S-box outputs 01101100000110000111110010101110

$f(R_7,K_8)$ = 00111100000011101000011011111001

$L_9 = R_8$ = 11010101011010010100101110010000

$E(R_8)$ = 01101010101010110101001010100101011111100101000 01

K_9 = 11100000110110111110101111101101111001111000000 1

$E(R_8) \oplus K_9$ = 10001010011100001011100101001000100110110010 0000

S-box outputs 00010001000011000101011101110111

$f(R_8,K_9)$ = 00100010001101100111110001101010

$L_{10} = R_9$ = 00100100011111001100011001111010

$E(R_9)$ = 00010000100000111111100101100000110000111111010 0

K_{10} = 10110001111100110100011110111010010001100100 1111

$E(R_9) \oplus K_{10}$ = 10100001011100001011111011011010100001011011 1011

S-box outputs 11011010000001000101001001110101

$f(R_9,K_{10})$ = 01100010101111001001110000100010

$L_{11} = R_{10}$ = 10110111110101011101011110110010

$E(R_{10})$ = 01011010111111101010101111101010111111011010 0101

K_{11} = 00100001010111111101001111011110110100111000011 0

$E(R_{10}) \oplus K_{11}$ = 01111011101000010111100000110100001011100010 0011

S-box outputs 01110011000001011101000100000001

$f(R_{10},K_{11}) = 11100001000001001111101000000010$

$L_{12} = R_{11} = 11000101011110000011110001111000$

$E(R_{11}) = 01100000101010111111000000011111100000111111 0001$

$K_{12} = 011101010111000111110101100101000110011111101001$

$E(R_{11}) \oplus K_{12} = 000101011101101000000101100010111110010000011000$

S-box outputs 01110011000001011101000100000001

$f(R_{11},K_{12}) = 11000010011010001100111111101010$

$L_{13} = R_{12} = 01110101101111010001100001011000$

$E(R_{12}) = 00111010101111011111101010001111000000101111 0000$

$K_{13} = 100101111100010111010001111110101011101001000001$

$E(R_{12}) \oplus K_{13} = 101011010111100000101011011101011011100010110001$

Sbox outputs 10011010110100011000101101001111

$f(R_{12},K_{13}) = 11011101101110110010100100100010$

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

$E(R_{13}) = 00001111000101100000011010001010101010111110100$

$K_{13} = 010111110100001110110111111100101110011100111010$

$E(R_{13}) \oplus K_{14} = 010100000101010110110001011110000100110111001110$

S-box outputs 01100100011110011001101011110001

$f(R_{13},K_{14}) = 10110111001100011000111001010101$

$L_{15} = R_{14} = 11000010100011001001011000001101$

$E(R_{14}) = 11100000010101000101100101001010110000000 1011011$

$K_{15} = 101111111001000110001101001111010011111100001010$

$E(R_{14}) \oplus K_{15} = 010111111100010111010100011101111111111101010001$

S-box outputs 10110010111010001000110100111100

$f(R_{14},K_{15}) = 01011011100000010010011101101110$

$R_{15} = 01000011010000100011001000110100$

$E(R_{15}) = 00100000011010100000010000011010010000011010 1000$

$K_{16} = 110010110011110110001011000011100001011111110101$

$E(R_{15}) \oplus K_{16} = 111010110101011110001111000101000101011001011101$

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:

8 5 E 8 1 3 5 4 0 F 0 A B 4 0 5

2.7.4. 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 trọng 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 kì và với đầu vào x bất kì $S(x)$ 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 kì, đầu vào x bất kì và với e, $f \in \{0, 1\}: S(x) \neq S(x \oplus 11ef00)$.

- Với hộp S bất kì, 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 kì, nếu ta cố định giá trị của một bit vào bất kì 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 10^6khoá/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.

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 tri 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.

Các tính chất và sức mạnh của DES:

DES có một số tính chất dễ nhận thấy và đồng thời chúng ta cũng sẽ sơ bộ đánh giá độ an toàn của DES thông qua các tấn công mạnh nhất hiện nay.

- *Tính chất bù:*

Kí hiệu phép mã hóa DES là E, và x^* là phần bù của x. Khi đó ta có: nếu $y = E_k(x)$ thì $y^* = E_K(x^*)$

- *Các khóa yếu và khóa nửa yếu:*

Định nghĩa: Một khóa yếu của DES là khóa K sao cho $E_K(E_K(x)) = x$ với mọi x. Một cặp khóa nửa yếu của DES là cặp (K_1, K_2) sao cho $E_{K1}(E_{K2}(x)) = x$ với mọi x.

DES có 4 khóa yếu và 6 cặp khóa nửa yếu

- *Các điểm bất động*

Với mỗi khóa yếu của DES sẽ có tương ứng 2^{32} điểm bất động, tức là x thỏa mãn $E_K(x) = x$.

Có 4 trong 12 khóa nửa yếu của DES mỗi cái sẽ có 2^{32} điểm phản bất động, tức là x sao cho $E_K(x) = x^*$.

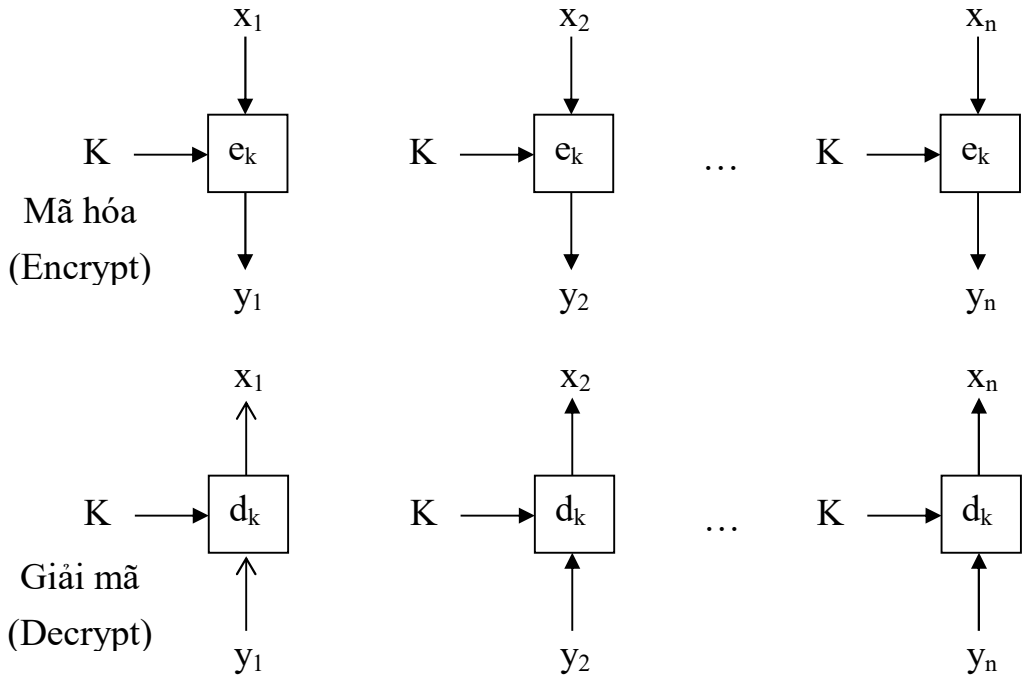
- *DES không phải là một nhóm dưới phép hợp hàm*

2.7.5. Các chế độ hoạt động của DES

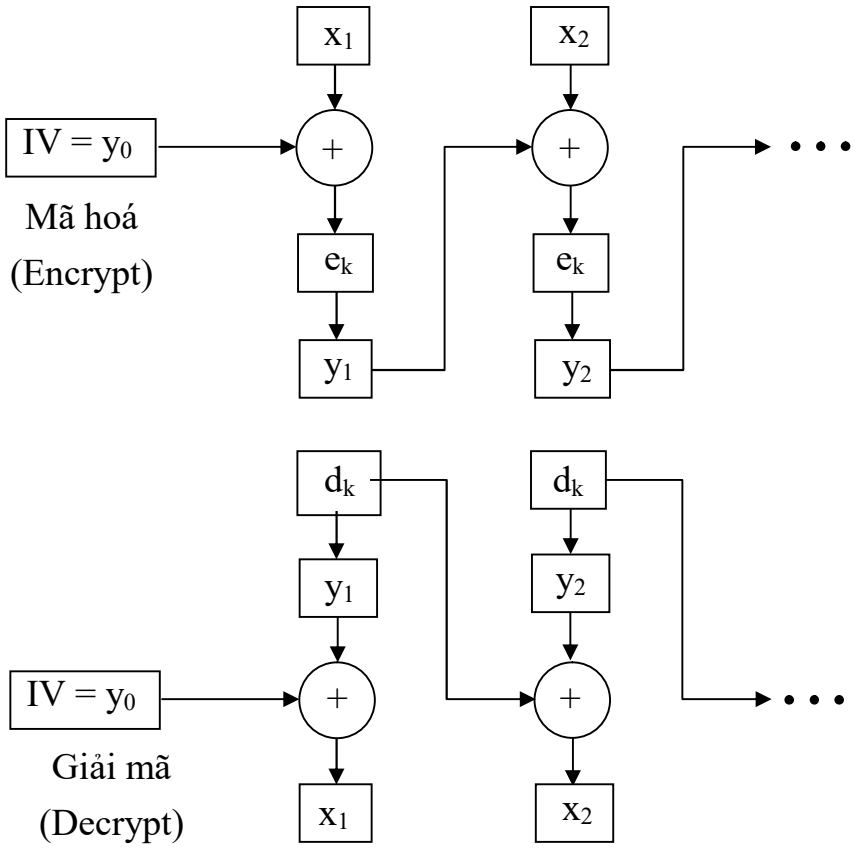
Có 4 chế độ làm việc đã được phát triển cho DES: Chế độ quyển mã điện tử (ECB), chế độ phản hồi mã (CFB), chế độ liên kết khối mã (CBC) và chế độ phản hồi đầu ra (OFB). Chế độ ECB tương ứng với cách dùng thông thường của mã khối: với một dãy các khối bản rõ cho trước $x_1, x_2, \ldots$ (mỗi khối có 64 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 $y_i = e_k(y_{i-1} \oplus x_i), i \geq 1$. Việc sử dụng chế độ CBC được mô tả trên hình 2.15.

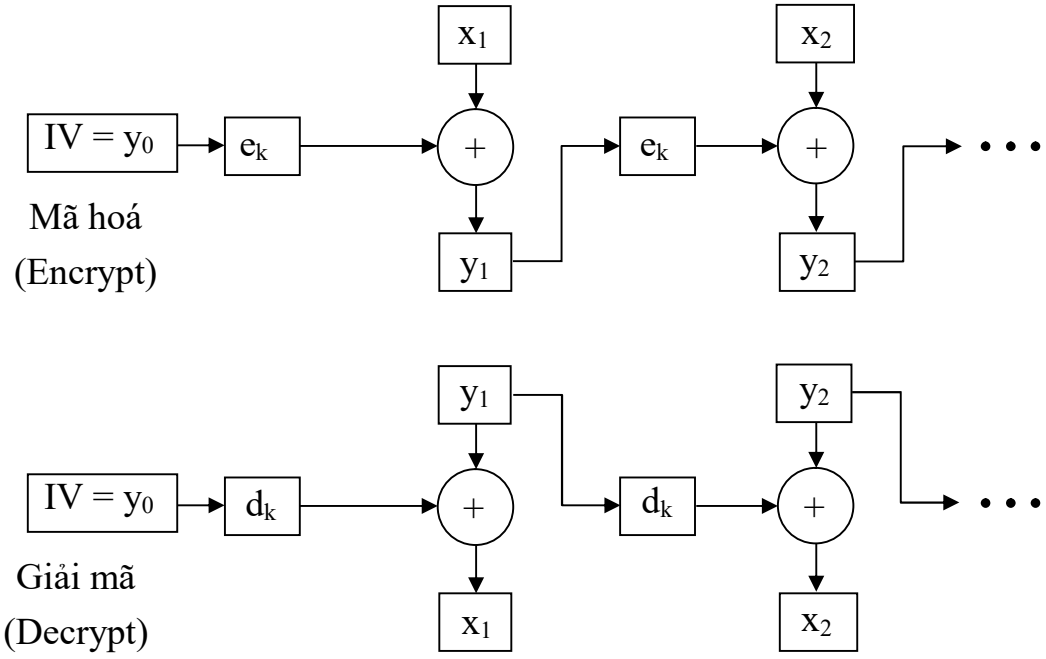
Trong các chế độ OFB và CFB dòng khoá được tạo ra sẽ được cộng mod 2 với bản rõ. OFB thực sự là một hệ mã dòng đồng bộ: dòng khoá được tạo bởi việc mã lặp vector khởi tạo 64 bit (vector IV). Ta xác định $z_0 = IV$ và rồi tính dòng khoá $z_1, z_2, \ldots$ theo quy tắc $z_i = e_k(z_{i-1}), i \geq 1$. Dãy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



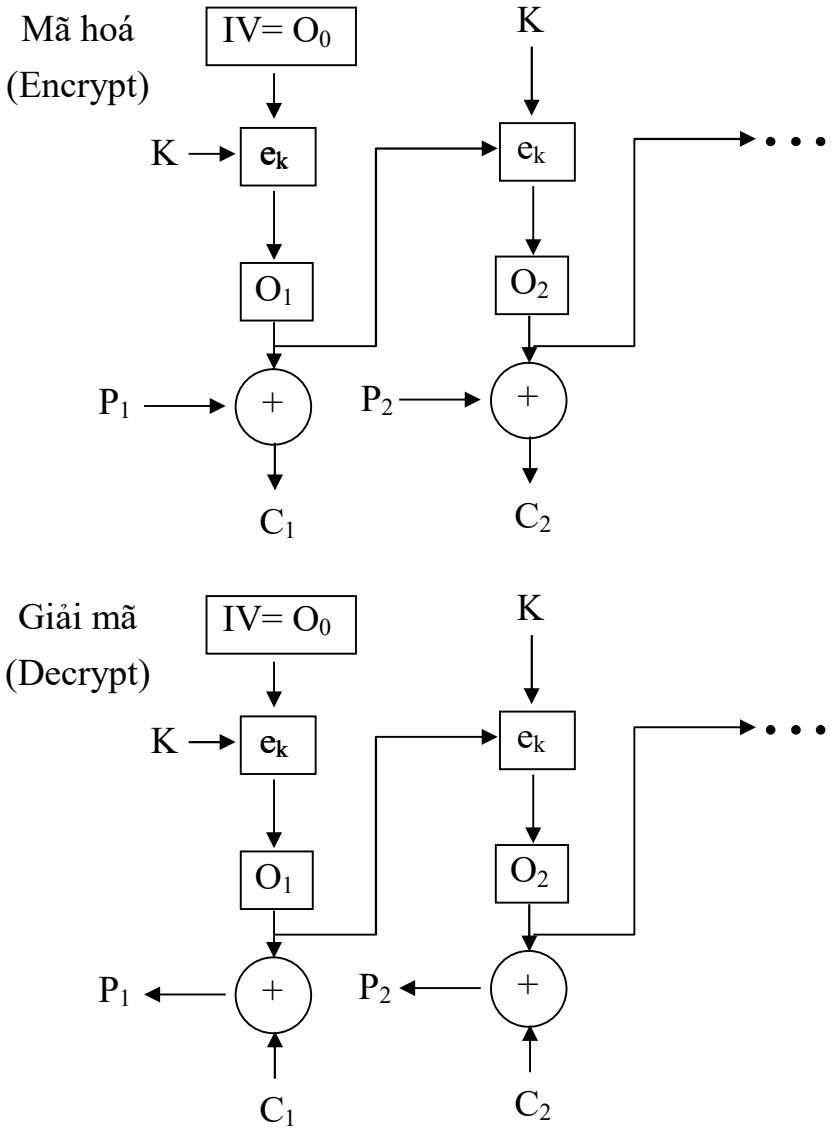
Hình 2-14. Chế độ ECB



Hình 2-15. Chế độ CBC



Hình 2-16. Chế độ CFB



Hình 2-17. Chế độ OFB

Trong chế độ CFB, ta bắt đầu với $y_0 = IV$ (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 2.16 (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 tấu 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.

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 CBC 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 $y_1, \ldots, y_n$ theo khoá K. Cuối cùng ta xác định MAC là y_n. Alice sẽ phát đi dãy các khối bản rõ $x_1, \ldots, x_n$ cùng với MAC. Khi Bob thu được $x_1 \ldots x_n$ anh ta sẽ khôi phục lại $y_1, \ldots, y_n$ 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 $x_1, \ldots, x_n$ và thay đổi ít nhiều nội dung thì thì chắc chắn là Oscar không thể thay đổi MAC để được Bob chấp nhận.

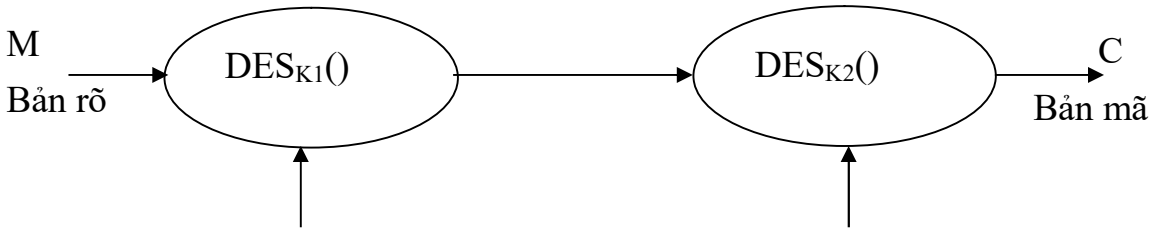
Thông thường ta muốn kết hợp cả tính xác thực lẫn độ bảo mật. Điều đó có thể thực hiện như sau: Trước tiên Alice dùng khoá K_1 để tạo MAC cho $x_1, \ldots, x_n$. Sau đó Alice xác định x_{n+1} là MAC rồi mã hoá dãy $x_1, \ldots, x_{n+1}$ bằng khoá thứ hai K_2 để tạo ra bản mã $y_1, \ldots, y_{n+1}$. Khi Bob thu được $y_1, \ldots, y_{n+1}$,

trước tiên Bob sẽ giải mã (bằng K_2) và kiểm tra xem x_{n+1} có phải là MAC đối với dãy $x_1, \ldots, x_n$ dùng K_1 hay không.

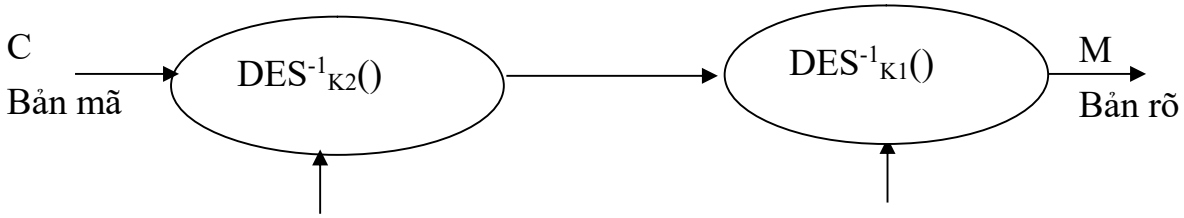
Ngược lại, Alice có thể dùng K_1 để mã hoá $x_1, \ldots, x_n$ và tạo ra được $y_1, \ldots, y_n$, sau đó dùng K_2 để tạo MAC y_{n+1} đối với dãy $y_1, \ldots, y_n$. Bob sẽ dùng K_2 để xác minh MAC và dùng K_1 để giải mã $y_1, \ldots, y_n$.

2.7.6. Một số biến thể của DES

2.7.6.1. DES bội hai (Double DES)



a. Mã hóa DES bội hai



b. Giải mã DES bội hai

Hình 2-18. Des bội hai

Mã hóa: $C = DES_{K2}\left[DES_{K1}(M)\right]$

Giải mã: $M = DES^{-1}_{K1}\left[DES^{-1}_{K2}(C)\right]$

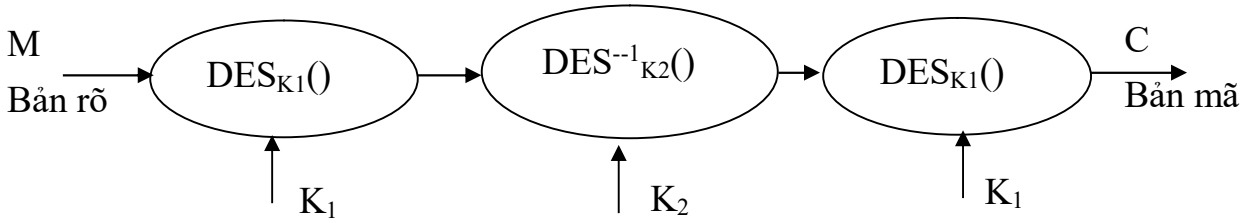
Mặc dù có 2^{56} sự lựa chọn cho khóa K_1 và 2^{56} sự lựa chọn đối với khóa K_2. Điều này dẫn tới có 2^{112} sự lựa chọn cho cặp khóa (K_1, K_2) nhưng sức mạnh của DES bội hai không lớn tới mức như vậy.

2.7.6.2. DES bội ba (Triple DES – TDES)

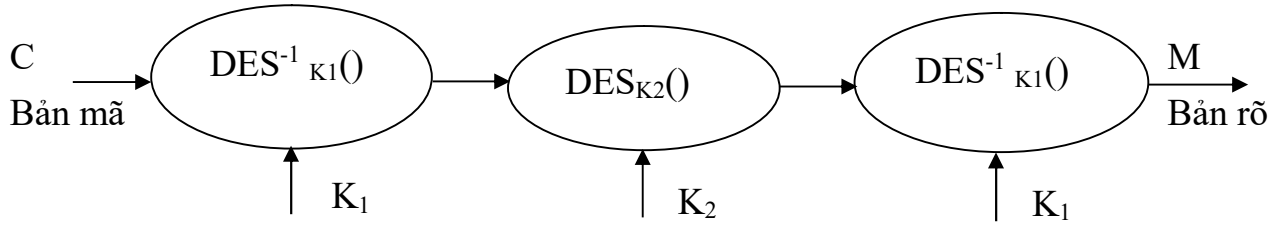
DES bội hai có thể bị tấn công bằng cách thám mã từ hai phía theo đề xuất của Diffie – Hellman. Để khắc phục yếu điểm này người ta đã xây dựng TDES với hai khóa K_1 và K_2 như sau:

Mã hóa: $C = DES_{K1}\left\{DES^{-1}_{K2}\left[DES_{K1}(M)\right]\right\}$

Giải mã: $M = DES^{-1}_{K1}\left\{DES_{K2}\left[DES^{-1}_{K1}(C)\right]\right\}$



a. Mã hóa TDES với hai khóa



b. Giải mã TDES với hai khóa

Hình 2-19. Mã hóa và giải mã TDES với hai khóa

Với TDES việc tìm kiếm vét cạn yêu cầu khoảng $2^{112} = 5{,}1923.10^{33}$ phép tính TDES, bởi vậy trên thực tế khó có thể thám mã thành công.

2.7.6.3. DES với các khóa con độc lập

Có thể sử dụng DES với 16 khóa con độc lập để tăng độ mật. Nếu 16 véctơ 48 bít được dùng cho các vòng mã hóa của DES thì người ta phải tạo một khóa k có độ dài 768 bít. Cách tấn công tìm kiếm vét cạn yêu cầu tìm kiếm trong không

gian khóa có kích thước 2^{768}. Cách tấn công từ hai phía có thể giảm không gian tìm kiếm xuống 2^{384}, giá trị này vẫn còn rất lớn trong thực tế. Tuy nhiên bằng cách sử dụng thám mã vi sai hệ mật này có thể bị phá với 2^{61} bản rõ được chọn.

2.7.6.4. DES tổng quát (Generalize DES – GDES)

Vào năm 1981 Johanmuller – Bilch đã đưa ra GDES nhằm tăng tốc độ mã hóa. Thuật toán GDES được mô tả trên hình 2.18

Thay cho việc sử dụng các khối thông báo 64 bít trong DES, GDES chia thông báo thành q khối 32 bít. Giả sử m là thông báo được dùng để mã hóa bằng GDES.

Trong đó $M_i = m_{i1}, m_{i2}, ... m_{i32}$

Ở vòng lặp đầu tiên GDES sẽ mã hóa khối con 32 bít cuối cùng:

$$B_0^{(q)} = M_q = m_{q1}, m_{q2}, ... m_{q32}$$

bằng 1 khóa con 48 bít K_1

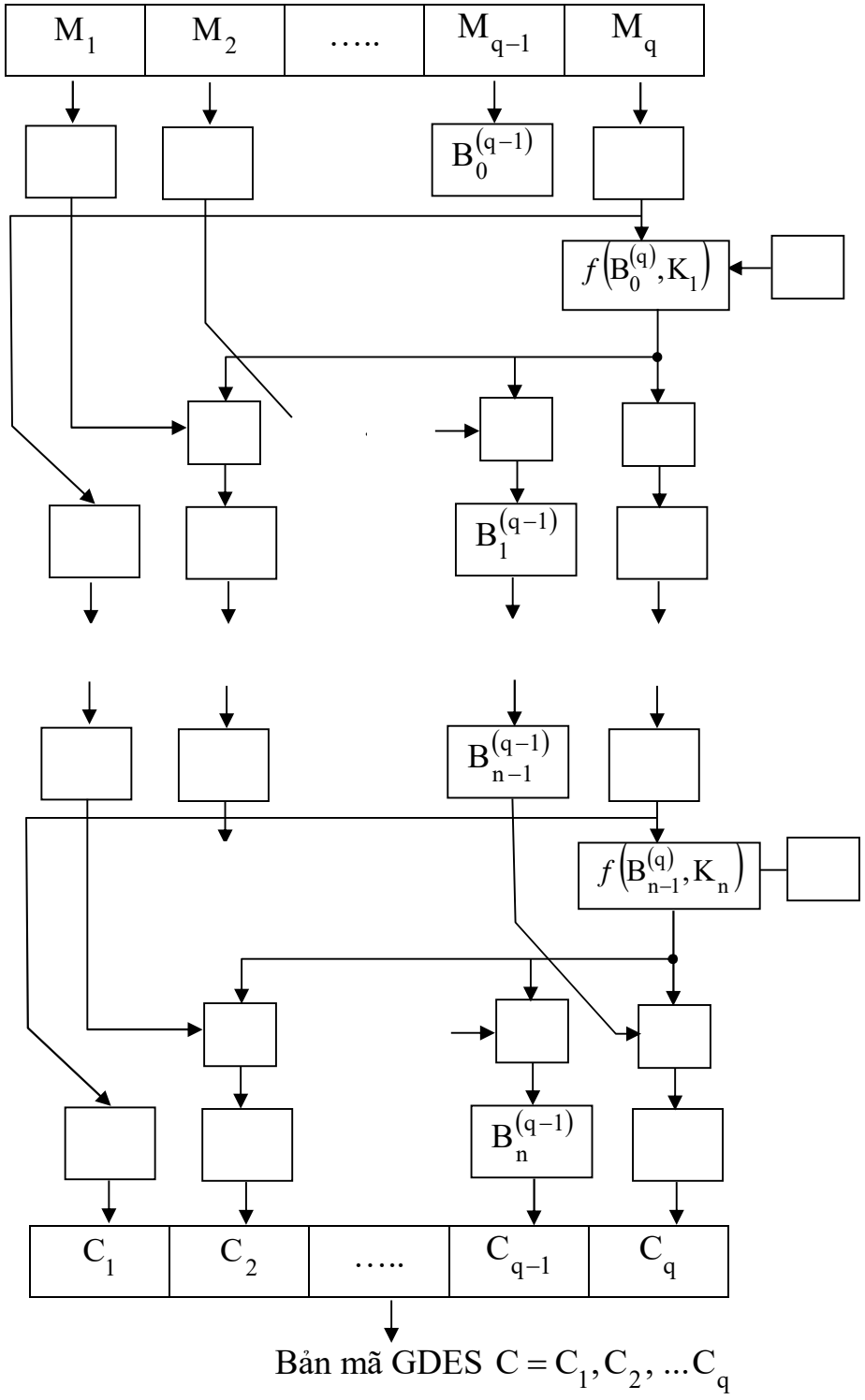
$$f\left(B_0^{(q)}, K_1\right) = \left\{S\left[K_1 \oplus E\left(B_0^{(q)}\right)\right]\right\}$$

Trong đó $S\left[K_1 \oplus E\left(B_0^{(q)}\right)\right]$ biểu thị phép thay thế trên véctơ 48 bít $K_1 \oplus E\left(B_0^{(q)}\right)$.

Véctơ 32 bít kết quả $f\left(B_0^{(q)}, K_1\right)$ sau đó được cộng mod 2 theo từng bít với các nội dung của $(q-1)$ thanh ghi 32 bít còn lại:

$$\begin{aligned}
B_1^{(2)} &= f\left(B_0^{(q)}, K_1\right) \oplus B_0^{(1)} \\
B_1^{(3)} &= f\left(B_0^{(q)}, K_1\right) \oplus B_0^{(2)} \\
&\vdots \\
B_1^{(q-1)} &= f\left(B_0^{(q)}, K_1\right) \oplus B_0^{(q-2)} \\
B_1^{(q)} &= f\left(B_0^{(q)}, K_1\right) \oplus B_0^{(q-1)}
\end{aligned}$$

Các nội dung trước đó của thanh ghi $B_0^{(q)}$ sẽ được lưu vào thanh ghi tận cùng bên trái $B_1^{(1)} = B_0^{(q)}$



Hình 2-20. Thuật toán mã hóa GDES

2.7.7. Thám mã vi sai và thám mã tuyến tính

Phương pháp thám mã truyền thống đối với các mật mã khối (chẳng hạn DES) với bản rõ đã biết là tìm kiếm, vét cạn trên toàn bộ không gian khóa. Tuy nhiên phương pháp tấn công tổng lực này không thể áp dụng được với DES bội đôi và DES bội ba. Các phương pháp tấn công tinh tế hơn đã được đề xuất trong những năm gần đây nhằm làm giảm độ phức tạp tính toán cho thám mã. Sau đây là 2 phương pháp quan trọng nhất.

2.7.7.1. Thám mã vi sai (thám mã dựa trên sự khác biệt)

Thám mã vi sai được đề xuất từ 1990 để thám các mật mã khối như PES, LUCIFER …Thám mã vi sai xoay quanh việc phân tích phân bố của sự khác biệt (cộng mod 2 theo từng bít) giữa hai bản rõ X_1 và X_2 và hai bản mã Y_1 và Y_2.Ở đây các bản rõ X_1 và X_2 là các nội dung 32 bít của thanh ghi dịch phải trước phép hoán vị mở rộng E(X) trong 1 vòng DES. Hai bản mã Y_1 và Y_2 đầu ra 32 bít từ phép hoán vị P© sau các hộp thay thế. Giả sử ΔX là hiệu của hai bản rõ đã biết X_1 và X_2 :

$$\Delta X = X_1 \oplus X_2$$

Ở đây $X_1 \oplus X_2$ biểu thị phép cộng mod 2 theo từng bít của hai véctơ bản rõ. Trong cách tấn công bản rõ có lựa chọn, hai bản rõ X_1 và X_2 được chọn sao cho có ΔX mong muốn. Vì $\Delta X = X_1 \oplus X_2$ và $A = E(X)$ đơn giản là một phép hoán vị mở rộng của các bít của bản rõ A nên ta cũng biết được ΔA .

$$\Delta A = A_1 \oplus A_2$$
$$\Delta A = E(X_1) + E(X_2)$$
$$\Delta A = E(\Delta X)$$

Ở mỗi vòng của DES, khóa con 48 bít K_i được cộng vào véctơ A 48 bít ở đầu ra của hộp hoán vị mở rộng:

$$B_1 = A_1 \oplus K_i$$
$$B_2 = A_2 \oplus K_i$$

Vì K_i là chưa biết nên B_1 và B_2 cũng chưa biết. Tuy nhiên ta lại biết được hiệu của chúng:

$$\Delta B = B_1 \oplus B_2$$
$$\Delta B = (A_1 + K_i) \oplus (A_2 + K_i)$$
$$\Delta B = A_1 \oplus A_2$$
$$\Delta B = \Delta A$$
$$\Delta B = E(\Delta X)$$

Bởi vậy bằng cách chọn X_1 và X_2 (tương ứng là ΔX) ta có thể tìm được các đầu vào của 8 hộp thay thế ngay cả khi không biết khóa con.

Từ các bản mã đã biết Y_1 và Y_2 thu được từ việc mã hóa các bản rõ X_1 và X_2 ta cũng xác định được hiệu ΔY của chúng:

$$\Delta Y = Y_1 \oplus Y_2$$

Cả hai véctơ Y_1 và Y_2 đều là các hoán vị của các đầu ra 32 bít C_1 và C_2 của các hộp thay thế.

$$Y_1 = P(C_1)$$
$$Y_2 = P(C_2)$$

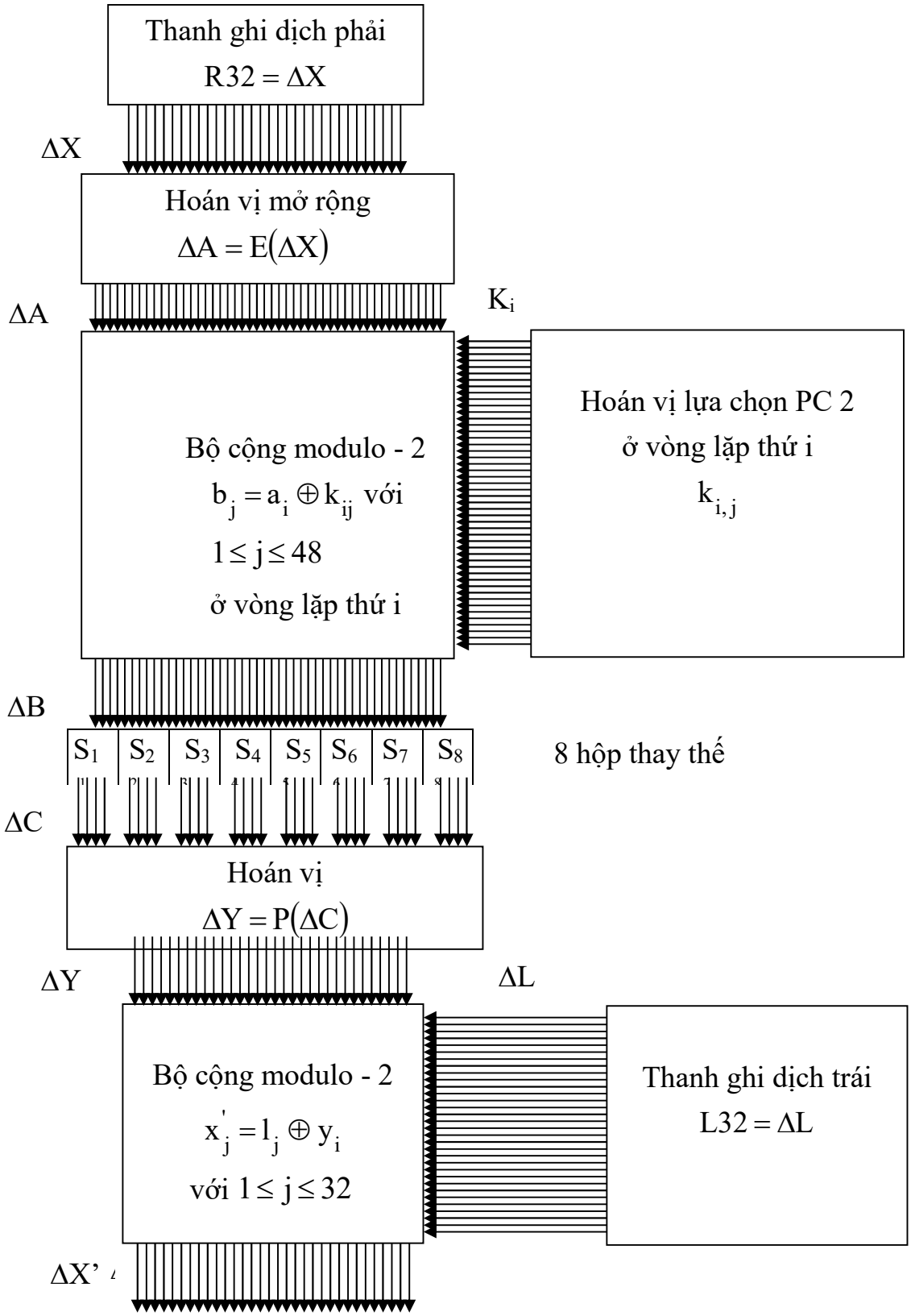
Ta có thể biểu thị các đầu ra C_1 và C_2 của các hộp thay thế như các hàm của Y_1 và Y_2:

$$C_1 = P^{-1}(Y_1)$$
$$C_2 = P^{-1}(Y_2)$$

Như vậy sự khác biệt ở đầu ra của các hộp thay thế ΔC là:

$$\Delta C = C_1 \oplus C_2$$
$$\Delta C = (P^{-1}(Y_1)) \oplus (P^{-1}(Y_2))$$
$$\Delta C = P^{-1}(\Delta Y)$$

Thám mã vi sai sẽ so sánh phân bố của ΔX đối với các cặp bản rõ X_1 và X_2 với phân bố của ΔY đối với các cặp bản mã Y_1 và Y_2 tương ứng.



Hình 2-21. Thám mã vi sai của một vòng DES

Trong cách tấn công với cặp rõ – mã được chọn, bản rõ được chọn sao cho tạo được ΔX mong muốn. Có một thực tế là các sai khác của bản rõ ΔX và các sai khác của bản mã ΔY là không như nhau. Một số sai khác trong các cặp bản rõ có xác suất gây nên sự khác biệt trong các cặp bản mã lớn hơn.

Với mỗi bộ 8 hộp thay thế của DES ta có thể tạo nên một bảng cho mối quan hệ giữa ΔX và ΔY (Xem bảng 2.1)

p_{ij} trong bảng biểu thị số các trường hợp mà ΔX_i tạo nên ΔY_j

	ΔY_1	…	ΔY_j	…
ΔX_1	p_{11}	…	p_{1j}	…
⋮	⋮		⋮	…
ΔX_i	p_{i1}	…	p_{ij}	…
⋮	⋮	…	⋮	…

Biham và Shamin đã trình diễn một thám mã DES16 vòng dùng 2^{47} cặp rõ – mã được chọn hoặc 2^{55} cặp rõ – mã đã biết với 2^{37} phép toán DES. Điều này chứng tỏ rằng các thám mã này với DES cũng chưa được hiệu quả.

2.7.7.2. Thám mã tuyến tính (TMTT)

Ý tưởng cơ bản của phương pháp này là cố gắng biểu thị (xấp xỉ) một vòng của DES bằng một phép biến đổi tuyến tính. Hình 2.17 biểu thị cách mà thám mã tuyến tính có thể dùng trên một vòng của DES.

Trong cách tấn công với bản rõ đã biết ta biết được bản rõ M và bản mã C tương ứng. Vì đầu ra IP(M) sau phép hoán vị ban đầu đã biết nên ta cũng biết được nội dung của các thanh ghi dịch trái và phải.

Giả sử $X = x_1, x_2, ... x_{32}$ là nội dung của thanh ghi dịch phải. 32 bít này sẽ qua một phép hoán vị mở rộng $A = E(X)$: véctơ 48 bít kết quả $A = a_1, a_2, ... a_{48}$ sẽ được cộng mod 2 theo từng bít với khóa con 48 bít $K_i = k_{i1}, k_{i2}, ... k_{i48}$ ở vòng lặp thứ i lấy ra từ phép biến đổi hoán vị lựa chọn PC 2.

Véctơ 48 bít $B = b_1, b_2, ... b_{48}$ sẽ được đưa qua 8 hộp thay thế $\{S_k\}_{k=1,...,48}$. Ở đó mỗi véctơ vào 6 bít $(b_1, b_2, b_3, b_4, b_5, b_6)$ sẽ được thay thế bằng một véctơ ra 4 bít (c_1, c_2, c_3, c_4). Véctơ 32 bít $C = c_1, c_2, ... c_{32}$ lại được biến đổi qua một phép hoán vị P và véctơ 32 bít.

$Y = y_1, y_2, ... y_{32}$ sẽ được cộng với nội dung của thanh ghi dịch trái. Thanh ghi dịch phải được cập nhật bằng véctơ 32 bít kết quả này.

$$Y = P(C)$$
$$C = P^{-1}(Y)$$

Từ hình 2.17 ta thấy rằng nếu biết đầu vào X (bản rõ sau phép hoán vị ban đầu IP) thì đầu ra của phép hoán vị mở rộng $A = E(X)$ cũng đã biết. Tuy nhiên vì khóa con $K_i = k_{ij}$ với $j = 1, 2, ...48$ ở vòng lặp thứ i (ta có thể bắt đầu với i = 1) là chưa biết nên ta không thể xác định được tổng ở đầu ra của các bộ cộng moduo 2: $b_j = a_i \oplus k_{ij}$ với $1 \le j \le 48$. Các bít ở đầu ra các bộ cộng $(\{b_j\} j = 1, 2, ...48)$ là các bít vào của 8 hộp thay thế S_k.

Bây giờ ta quay trở lại nội dung của thanh ghi dịch trái L và nội dung trước đó của thanh thanh ghi dịch phải X' (trên thực tế: Thanh ghi dịch tạm thời TEMP 32 từ vòng lặp trước của DES), ta có thể xác định được véctơ 32 bít Y. Vì Y là kết quả của phép hoán vị chuẩn P của đầu ra từ các hộp thay thế:

$$C = P^{-1}(Y)$$

Véctơ 32 bít $C = c_1, c_2, ... c_{32}$ ở đầu ra của các hộp thay thế cũng được xác định.

Các hộp thay thế $\{S_k\}_{k=1,...,48}$ phải ngẫu nhiên và không chệch. Với một đầu vào 6 bít bất kỳ b_1, b_2, b_3, b_4, b_5 và b_6, các bít ra phải có phân bố chuẩn đều. Bây giờ bằng cấu tạo của bảng của tất cả 64 véctơ vào của mỗi hộp thay thế, mỗi bít vào $b_i = 0$ ở một nửa số lần và $b_i = 1$ ở một nửa số lần khác. Nói một cách

khác, ta có thể nói rằng mỗi một bít vào (trong 6 bít) bằng 0 với xác suất $p = \frac{1}{2}$ và mỗi một bít ra (trong 4 bít) bằng 0 với xác suất $p = \frac{1}{2}$.

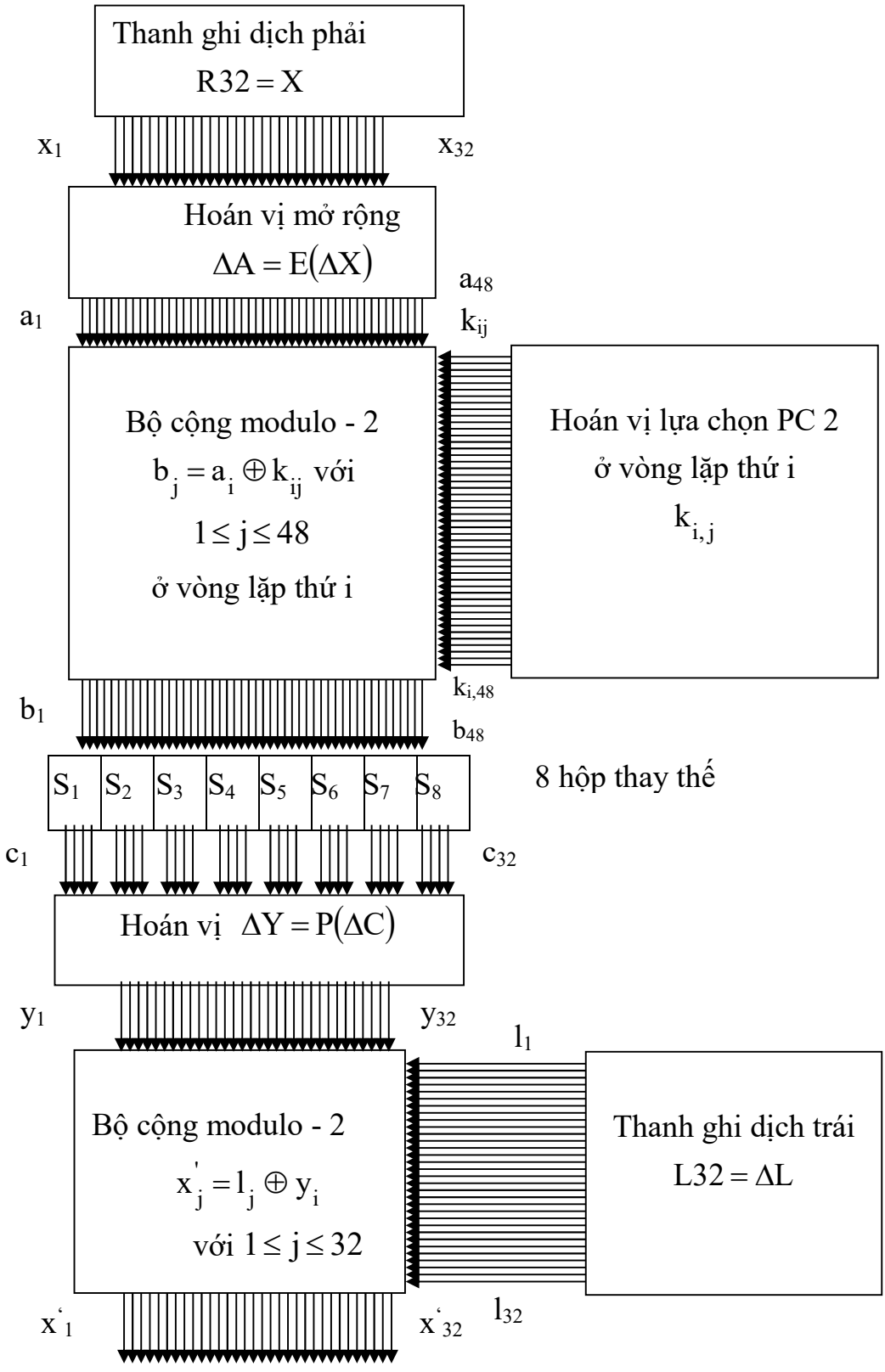
Tuy nhiên ta có thể suy ra đầu vào của một hộp thay thế nếu có thể khai thác được mối quan hệ giữa các đầu vào và các đầu ra của nó. Chẳng hạn nếu ta quan sát 4 bít c_1, c_2, c_3 và c_4 ở đầu ra của một hộp thay thế S_k và cộng chúng với nhau theo modulo 2 thì đối với 64 véctơ vào khác nhau $b_1, ..., b_6$, kết quả sẽ là $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 0$ với một nửa số trường hợp (32 trường hợp) và $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 1$ với một nửa số trường hợp còn lại. (Mỗi một giá trị trong 16 véctơ ra sẽ xuất hiện 4 lần trong bảng thay thế).

Ta có thể thấy rằng quan hệ vào – ra của các hộp thay thế không hoàn toàn không chệch. Chẳng hạn, hộp thay thế S_5 là chệch nhất trong các hộp thay thế và ta có thể khai thác nó để suy ra khóa. Bảng 2.2 chỉ ra quan hệ giữa 6 bít vào b_{25}, $b_{26}, b_{27}, b_{28}, b_{29}$ và b_{30} và 4 bít ra c_{17}, c_{18}, c_{19} và c_{20} trong hộp thay thế S_5. Từ bảng 2.2 ta có thể thấy rằng ngay cả khi bít vào $b_{26} = 0$ trong một nửa số trường hợp (tức là với xác suất $p = \frac{1}{2}$) và tổng $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 0$

Với xác suất $p = \frac{1}{2}$ thì phương trình sau:

$$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$$

chỉ đúng có 12 lần trong số 62 lần (đúng với xác suất $p = \frac{12}{64} = \frac{3}{16}$) 12 trường hợp này được chỉ ra ở cột kiểm tra trong bảng 2.2.



Hình 2-22. Thám mã tuyến tính của một vòng DES

6 bít vào						Ra	4 bít ra				Kiểm tra
b_1 b_{25}	b_2 b_{26}	b_3 b_{27}	b_4 b_{28}	b_5 b_{29}	b_6 b_{30}		c_1 c_{17}	c_2 c_{18}	c_3 c_{19}	c_4 c_{20}	
0	0	0	0	0	0	2	0	0	1	0	
0	0	0	0	0	1	14	1	1	1	0	
0	0	0	0	1	0	12	1	1	0	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0	0	0	0	1	1	11	1	0	1	1	
0	0	0	1	0	0	4	0	1	0	0	
0	0	0	1	0	1	2	0	0	1	0	
0	0	0	1	1	0	1	0	0	0	1	
0	0	0	1	1	1	12	1	1	0	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0	0	1	0	0	0	7	0	1	1	1	
0	0	1	0	0	1	4	0	1	0	0	
0	0	1	0	1	0	10	1	0	1	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0	0	1	0	1	1	7	0	1	1	1	
0	0	1	1	0	0	11	1	0	1	1	
0	0	1	1	0	1	13	1	1	0	1	
0	0	1	1	1	0	6	0	1	1	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0	0	1	1	1	1	1	0	0	0	1	
0	1	0	0	0	0	8	1	0	0	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0	1	0	0	0	1	5	0	1	0	1	
0	1	0	0	1	0	5	0	1	0	1	
0	1	0	0	1	1	0	0	0	0	0	
0	1	0	1	0	0	3	0	0	1	1	
0	1	0	1	0	1	15	1	1	1	1	
0	1	0	1	1	0	15	1	1	1	1	
0	1	0	1	1	1	10	1	0	1	0	
0	1	1	0	0	0	13	1	1	0	1	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0	1	1	0	0	1	3	0	0	1	1	
0	1	1	0	1	0	0	0	0	0	0	
0	1	1	0	1	1	9	1	0	0	1	
0	1	1	1	0	0	14	1	1	1	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0	1	1	1	0	1	8	1	0	0	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$

0	1	1	1	1	0	9	1	0	0	1	
0	1	1	1	1	1	6	0	1	1	0	

Hình 2-23. Quan hệ vào ra trong hộp thay thế S_5 (bắt đầu)

6 bít vào						Ra	4 bít ra				Kiểm tra
b_1 b_{25}	b_2 b_{26}	b_3 b_{27}	b_4 b_{28}	b_5 b_{29}	b_6 b_{30}		c_1 c_{17}	c_2 c_{18}	c_3 c_{19}	c_4 c_{20}	
1	0	0	0	0	0	4	0	1	0	0	
1	0	0	0	0	1	11	1	0	1	1	
1	0	0	0	1	0	2	0	0	1	0	
1	0	0	0	1	1	8	1	0	0	0	
1	0	0	1	0	0	1	0	0	0	1	
1	0	0	1	0	1	12	1	1	0	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1	0	0	1	1	0	11	1	0	1	1	
1	0	0	1	1	1	7	0	1	1	1	
1	0	1	0	0	0	10	1	0	1	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1	0	1	0	0	1	1	0	0	0	1	
1	0	1	0	1	0	13	1	1	0	1	
1	0	1	0	1	1	14	1	1	1	0	
1	0	1	1	0	0	7	0	1	1	1	
1	0	1	1	0	1	2	0	0	1	0	
1	0	1	1	1	0	8	1	0	0	0	
1	0	1	1	1	1	13	1	1	0	1	
1	1	0	0	0	0	15	1	1	1	1	
1	1	0	0	0	1	6	0	1	1	0	
1	1	0	0	1	0	9	1	0	0	1	
1	1	0	0	1	1	15	1	1	1	1	
1	1	0	1	0	0	12	1	1	0	0	
1	1	0	1	0	1	0	0	0	0	0	
1	1	0	1	1	0	5	0	1	0	1	
1	1	0	1	1	1	9	1	0	0	1	
1	1	1	0	0	0	6	0	1	1	0	
1	1	1	0	0	1	10	1	0	1	0	
1	1	1	0	1	0	3	0	0	1	1	

1	1	1	0	1	1	4	0	1	0	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1	1	1	1	0	0	0	0	0	0	0	
1	1	1	1	0	1	5	0	1	0	1	
1	1	1	1	1	0	14	1	1	1	0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1	1	1	1	1	1	3	0	0	1	1	

Hình 2-24. Quan hệ vào ra trong hộp thay thế S_5 (kết thúc)

Ta có thể thấy rằng xác suất để có $b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$ là $\frac{3}{16}$ sẽ được dùng để trợ giúp cho việc phá DES. Khi đó với xác suất $p = \frac{3}{16}$

$$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$$
$$a_{26} \oplus k_{i26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$$

Nhưng vì $A = E(X)$ nên $a_{26} = x_{17}$.

Tương tự việc biết ánh xạ của hàm hoán vị chuẩn $Y = P(C)$ giúp ta có thể thay c_{17}, c_{18}, c_{19} và c_{20} bằng các giá trị của bản mã đã biết y_3, y_8, y_{14} và y_{25}.

Bởi vậy, với xác suất $p = \frac{3}{16}$

$$k_{i26} = a_{26} \oplus c_1 \oplus c_2 \oplus c_3 \oplus c_4$$
$$k_{i26} = a_{26} \oplus c_{17} \oplus c_{18} \oplus c_{19} \oplus c_{20}$$
$$k_{i26} = x_{17} \oplus y_3 \oplus y_8 \oplus y_{14} \oplus y_{25}$$

Vì cặp bản rõ X và bản mã Y ở một vòng đã biết nên điều này cung cấp chứng cứ để coi bít k_{i26} là phần bù của $x_{17} \oplus y_3 \oplus y_8 \oplus y_{14} \oplus y_{25}$.

Phân tích một vòng này sẽ được tổng quá hóa lên cho 16 vòng của DES. Điều này có thể thực hiện được vì các nội dung của thanh ghi phải ở vòng lặp thứ hai là một hàm của các kết quả ở vòng lặp thứ nhất.

Thám mã tuyến tính đối với DES vẫn còn khó khả thi vì nó cần tới 2^{47} cặp rõ - mã đã biết để tìm một bít khóa riêng lẻ. Bít khóa thứ hai có thể tìm được.

Người ta đã chỉ ra rằng sử dụng phép xấp xỉ tuyến tính cho DES 14 vòng và đánh giá (phán đoán) 6 bít khóa con k_{i25}, $k_{i26}, k_{i27}, k_{i28}, k_{i29}$ và k_{i30} theo 6 bít vào của hộp thay thế S_5 cho các vòng 2 và 14, điều này tương đương với việc thực hiện 2^{12} phép phân tích tuyến tính song song và sẽ tạo ra 26 bít khóa. Điều này sẽ làm giảm không gian khóa cần tìm kiếm từ 2^{56} (khi tìm kiếm vét cạn) xuống còn $2^{30} = 1.073.741.824$.

2.8. 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 bít, 192 bít và 256 bít). Sau đây ta sẽ mô tả thuật toán AES này.

2.8.1. Cơ sở toán học của AES

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

Phép cộng:

Phép cộng giữa hai phần tử (các byte) trong trường hữu hạn được thực hiện bằng cách cộng theo modulo 2 các bít 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:

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

Phép nhận:

Phép nhân được thực hiện trên $GF\left(2^8\right)$ bằng cách nhân hai đa thức rút gọn theo mođulo 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: $A = C3_H$, $B = 85_H$ tương ứng với:

$a(x) = x^7 + x^6 + x + 1$ 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 $C = AE_H = 10101110$

2.8.2. Thuật toán AES

AES mã hóa một khối bản rõ M 128 bít thành một khối bản mã C 128 bít bằng cách dùng một khóa mã K có độ dài 128 bít (hoặc 192 hoặc 256 bít) 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 bít (4 byte).

AES sẽ thực hiện một số vòng mã hóa N_r phụ thuộc vào độ dài khóa được sử dụng (Xem bảng 2.1)

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

Hình 2-25. Số các vòng mã hóa của AES

Mã hóa AES:

Mỗi 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 afin 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 bít 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 khối đầu ra (bản mã) là các khối 128 bít gồm $N_b = 4$ từ 32 bít

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

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

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 $\text{GF}(2^8)$. 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 $\mathrm{mod}\left(X^4+1\right)$

$$s'(x)=a(x).s(x)\mathrm{mod}\left(X^4+1\right)$$

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 $0 \le c < N_b$

<u>Mở rộng khóa AES: KeyExpansion( )</u>

Thuật toán AES sẽ tạo từ khóa mã 128 bít (hoặc 192 hoặc 256 bít) một tập khởi tạo N_b từ 32 bít và N_b từ 32 bít cho mỗi vòng bao gồm $N_b\left(N_r+1\right)$ từ 32 bít . 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 bít) W_i:

$$\mathrm{RotWord}\left(a_0,a_1,a_2,a_3\right)=\left(a_1,a_2,a_3,a_0\right)$$

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

Begin

$i=0$

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

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

$i=i+1$

```
        end while
    i  N_k
    while (i < N_b*(N_r + 1))
        word temp = w[i - 1]
        if (i mod N_k = 0)
            temp = SubWord(RotWord(temp)) xor Rconw[i/N_k]
        else if (N_k = 8 and i mod N_k = 4)
                    temp = SubWord(temp)
        end if
    w[i]  w[i - N_k] = xor temp
    i = i + 1
   end while
end
```

(nguồn trích dẫn: Đặc tả thô AES: http://csrc.nist.gov/encryption/aes/)

Chương trình giải mã của AES

```
Cipher (byte in[4* N_b], byte out[4* N_b], word w[N_b*(N_r + 1)])
    Begin  byte state [4, N_b] 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 NHƯỢC ĐIỂM CỦA CÁC HỆ MẬT KHÓA BÍ MẬT

Ưu điểm của các hệ mật khóa bí mật:

- Mô hình khá đơn giản
- Dễ dàng tạo ra thuật toán mã hóa đối xứng cho cá nhân
- Dễ cài đặt và hoạt động hiệu quả
- Hoạt động nhanh và hiệu quả do tốc độ mã hóa và giải mã cao

Nhược điểm:

- Dùng chung khóa nên nhiều nguy cơ mất an toàn
- Khóa dùng chung rất dễ bị hóa giải ("bẻ khóa") do phải truyền trên kênh truyền tin đến bên nhận
- Việc gửi thông tin cùng khóa cho số lượng lớn là khó khăn.

2.10. BÀI TẬP

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 Xeda với khoá k chưa biết. Hãy dùng phương pháp tìm khoá vét cạn để tìm 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.

2. Dưới đây là 4 bản mã thu được từ mã thay thế Một bản thu được 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
KYVXCHKFTPONCQQRHJVAJUWETMCMSPKQDYHJVDAHCTRL
SVSKCGCZQọDZXGSFRLSWCWSJTBHAFSLASPRJAHKJRJUMV
KMITZHFPDLSPZLVLGWTFPLKKEBDPGCEBSHCTJRWXBAFS
PEZQNRWXCVYCGAONWDDKACKAWBBIKFTLOVKCGGHJVLNHI
FFSQESVYCLACNVRWBBIREPBBVFEXOSCDYGZWPFDTKFQLY
CWHJVTNHIQ/BTKH/VNPIST

c. *Hệ mã Affine.*

KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRLOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABĐKP
BCPFEQPKAZBKRHALBKAPCCIBURCCDKDCCJC/DFUIXPAFF
ERBICZDFKABICBBENEFCUPLCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKLJPKABL

d. *Hệ mã chưa xác định được.*

BNVSNSIHQCEELSSKKYERIFJKXUMBGVKAMQLJTYAVFBKVT
DVBPVVRJYYLAOKYMPQSCGDLFSRLLPROYGESEBUUALRWM
MASAZLGLEĐFJBZAVVPXWI
CGJXASCBYEHOSNMULKCEAHTQ
OKMFLEBKFXLRRFDTZXCIWBJSICBGAWDVYDHAVFJXZIBKC
GJIWEAHTTOEWTUHKRQVVRGZBXYIREMMASCSPBNLHJMBLR
FFJELHWEYLWISTFVVYFJCMHYUYRUFSFMGESIGRLWALSVM
NUHSIMYYITCCQPZSICEHBCCMZFEGVJYOCDEMMPGHVAAUM
ELCMOEHVLTIPSUYILVGFLMVWDVYDBTHFRAYISYSGKVSUU

HYHGGCKTMBLRX

3. Có bao nhiêu ma trận khả nghịch cấp 2x2 trên Z_{26}.
   a. Giả sử p là số nguyên tố. Hãy chứng tỏ số các ma trận khả nghịch cấp 2x2 trên Z_p là $\left(p^2-1\right)\left(p^2-p\right)$.

      Chỉ dẫn Vì p là số nguyên tố nên 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).

   b. 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 mxm trên Z_p.

4. 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.

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

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

   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.

6. 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.

7. 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 x 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

8. 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.

9. Cho DES(x,K) là phép mã hoá DES của bản rõ x với khoá K. Giả sử $y = \mathrm{DES}(x, K)$ và $y' = \mathrm{DES}(c(x), c(K))$ trong đó c(.) kí hiệu là phần bù theo các bít 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.

10. Mã kép là một cách để làm mạnh thêm cho DES: với hai khóa K_1 và K_2 cho trước, ta xác định $y = e_{K2}\left(e_{K1}(x)\right)$ (dĩ nhiên đây chính là tích của DES với chính nó). Nếu hàm mã hoá e_{K2} giống như hàm giải mã d_{K1} 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 chứag tỏ rằng nếu $C_0 = 0101\ldots01$ hoặc $1010\ldots10$ (ở dạng nhị phân) thì XOR các xâu bít C_i và C_{17-i} là $111\ldots11$, với $1 \le i \le 16$ (khẳng định tương tự cũng đúng đối với D_i).

   c. Hãy chứng tỏ các cặp khoá sau là đối ngẫu:

E001E001F101F101	01E001E001F101F1
FE1FFE1FF0EFE0E	1FFE1FFE0EFE0EFE
E01FE01FFF10FF10	1FE01FE00EF10EF1

CHƯƠNG 3 CÁC HỆ MẬT KHÓA CÔNG KHAI

3.1. GIỚI THIỆU VỀ MẬT MÃ KHÓA CÔNG KHAI

Trong mô hình mật mã cổ điển trước đây mà hiện nay đang được nghiên cứu Alice (người gửi) và Bob (người nhận) chọn một cách bí mật khoá K. Sau đó dùng K để tạo luật mã hoá e_k và luật giải mã d_k. Trong hệ mật này d_k hoặc giống e_k hoặc dễ dàng nhận được từ nó (ví dụ trong hệ DES quá trình giải mã hoàn toàn tương tự như quá trình mã nhưng thủ tục khoá ngược lại). Các hệ mật thuộc loại này được gọi là hệ khoá bí mật, nếu để lộ e_k thì làm cho hệ thống mất an toàn.

Nhược điểm của hệ mật này là nó yêu cầu phải có thông tin trước về khoá K giữa Alice và Bob qua một kênh an toàn trước khi gửi một bản mã bất kỳ. Trên thực tế điều này rất khó đảm bảo. Chẳng hạn khi Alice và Bob ở cách xa nhau và họ chỉ có thể liên lạc với nhau bằng thư tín điện tử (E.mail). Trong tình huống đó Alice và Bob không thể tạo một kênh bảo mật với giá phải chăng.

Ý tưởng xây dựng một hệ mật khoá công khai (hay dùng chung) là tìm một hệ mật không có khả năng tính toán để xác định d_k khi biết e_k. Nếu thực hiện được như vậy thì quy tắc mã e_k có thể được công khai bằng cách công bố nó trong một danh bạ (bởi vậy nên có thuật ngữ *hệ mật khoá công khai*). Ưu điểm của hệ mật khoá công khai là ở chỗ Alice (hoặc bất kỳ ai) có thể gửi một bản tin đã mã cho Bob (mà không cần thông tin trước về khoá mật) bằng cách dùng mật

mã công khai e_k. Người nhận A sẽ là người duy nhất có thể giải được bản mã này bằng sử dụng luật giải bí mật d_k của mình.

Có thể hình dung hệ mật này tương tự như sau. Alice đặt một vật vào một hộp kim loại và rồi khoá nó lại bằng một khoá số do Bob để lại. Chỉ có Bob là người duy nhất có thể mở được hộp vì chỉ có anh ta mới biết tổ hợp mã của khoá số của mình.

Ý tưởng về một hệ mật khoá công khai được Diffie và Hellman đưa ra vào năm 1976. Còn việc hiện thực hoá nó thì do Rivesrt, Shamir và Adleman đưa ra lần đầu tiên vào năm 1977, họ đã tạo nên hệ mật nổi tiếng RSA (sẽ được nghiên cứu trong chương này). Kể từ đó đã công bố một số hệ, độ mật của chúng dựa trên các bài tính toán khác nhau. Trong đó, quan trọng nhất là các hệ mật khoá công khai sau:

- *Hệ mật RSA:*

  Độ bảo mật của hệ RSA dựa trên độ khó của việc phân tích ra thừa số nguyên lớn. Hệ này sẽ được mô tả trong phần 4.2.

- *Hệ mật xếp ba lô Merkle - Hellman:*

  Hệ này và các hệ liên quan dựa trên tính khó giải của bài toán tổng các tập con (bài toán này là bài toán NP đầy đủ - là một lớp khá lớn các bài toán không có giải thuật được biết trong thời gian đa thức). Tuy nhiên tất cả các hệ mật xếp ba lô khác nhau đều đã bị chứng tỏ là không an toàn (ngoại trừ hệ mật Chor-Rivest).

- *Hệ mật McEliece:*

  Hệ này dựa trên lý thuyết mã đại số và vẫn còn được coi là an toàn. Hệ mật McEliece dựa trên bài toán giải mã cho các mã tuyến tính (cũng là một bài toán NP đầy đủ). Hệ mật McEliece được trình bày ở phần 4.6.

- *Hệ mật ElGamal:*

  Hệ mật ElGamal dựa trên tính khó giải của bài toán logarithm rời rạc trên các trường hữu hạn

- *Hệ mật Chor-Rivest:*

  Hệ mật Chor-Rivest cũng được xem như một hệ mật xếp ba lô. Tuy nhiên nó vẫn được coi là an toàn

- *Hệ mật trên các đường cong Elliptic:*

  Các hệ mật này là biến tướng của các hệ mật khác (chẳng hạn như hệ mật ElGamal), chúng làm việc trên các đường cong Elliptic chứ không phải là trên các trường hữu hạn. Hệ mật này đảm bảo độ mật với số khoá nhỏ hơn các hệ mật khoá công khai khác.

Một chú ý quan trọng là một hệ mật khoá công khai không bao giờ có thể đảm bảo được độ mật tuyệt đối (an toàn vô điều kiện). Sở dĩ như vậy vì đối phương khi nghiên cứu một bản mã, y có thể mã lần lượt các bản tin rõ bằng luật mã hoá công khai e_k cho tới khi anh ta tìm được bản rõ duy nhất x đảm bảo $y = e_k(x)$. Bản rõ này chính là kết quả giải mã của y. Bởi vậy, ta chỉ nghiên cứu độ mật về mặt tính toán của các hệ mật này.

Một khái niệm có ích khi nghiên cứu hệ mật khoá công khai là khái niệm về hàm cửa sập một chiều. Ta sẽ định nghĩa khái niệm này một cách không hình thức.

Hàm mã khoá công khai e_k của Bob phải là một hàm dễ tính toán. Song việc tìm hàm ngược (hàm giải mã) rất khó khăn (đối với bất kỳ ai không phải là Bob). Đặc tính khó tính toán hàm ngược thường được gọi là đặc tính một chiều. Bởi vậy điều kiện cần thiết là e_k phải là hàm một chiều.

Các hàm một chiều đóng vai trò quan trọng trong mật mã học, chúng rất quan trọng trong các hệ mật khoá công khai và trong nhiều lĩnh vực khác. Đáng tiếc là mặc dù có rất nhiều hàm được coi là hàm một chiều nhưng cho đến nay vẫn không tồn tại một hàm nào có thể chứng minh được là hàm một chiều.

Sau đây là một ví dụ về một hàm được coi là hàm một chiều. Giả sử n là tích của hai số nguyên tố lớn p và q, giả sử b là một số nguyên dương. Khi đó ta xác định ánh xạ $f : Z_n \to Z_n$ là $f(x) = x^b \bmod n$ (với b và n đã được chọn thích hợp thì đây chính là hàm mã RSA, sau này ta sẽ nói nhiều hơn về nó).

Để xây dựng một hệ mật khoá công khai thì việc tìm được một hàm một chiều vẫn chưa đủ. Ta không muốn e_k là hàm một chiều đối với Bob vì anh ta phải có khả năng giải mã các bản tin nhận được một cách hiệu quả. Điều cần thiết là Bob phải có một cửa sập chứa thông tin bí mật cho phép dễ dàng tìm hàm của e_k. Như vậy Bob có thể giải mã một cách hữu hiệu vì anh ta có một hiểu biết tuyệt mật nào đó về K. Bởi vậy một hàm được gọi là cửa sập một chiều nếu nó là một hàm một chiều và nó trở nên dễ tính ngược nếu biết một cửa sập nhất định.

3.2. BÀI TOÁN PHÂN TÍCH THỪA SỐ VÀ CÁC HỆ MẬT CÓ LIÊN QUAN

3.2.1. Bài toán phân tích thừa số

Bài toán phân tích một số nguyên > 1 thành thừa số nguyên tố cũng được xem là một bài toán khó thường được sử dụng trong lý thuyết mật mã. Biết một số n là hợp số thì việc phân tích n thành thừa số mới là có nghĩa, do đó thường khi để giải bài toán phân tích n thành thừa số, ta thử trước n có là hợp số hay không; và bài toán phân tích n thành thừa số có thể dẫn về bài toán *tìm một ước số* của n, vì khi biết một ước số d của n thì tiến trình phân tích n được tiếp tục thực hiện bằng cách phân tích d và n|d.

Bài toán phân tích thành thừa số, hay bài toán tìm ước số của một số nguyên cho trước, đã được nghiên cứu nhiều, nhưng cũng chưa có một thuật toán hiệu quả nào để giải nó trong trường hợp tổng quát mà người ta có xu hướng giải bài toán này theo những trường hợp đặc biệt của số cần phải phân tích, chẳng hạn khi n có một ước số nguyên tố p với p – 1 là B-mịn với một cận B > 0 nào đó, hoặc khi n là số Blum, tức là số có dạng tích của hai số nguyên tố lớn nào đó (n = p.q).

Ta xét trường hợp thứ nhất với (p – 1)-thuật toán Pollard như sau: Một số nguyên n được gọi là B-mịn nếu tất cả các ước số nguyên tố của nó đều ≤ B. Ý chính chứa trong (p – 1)-thuật tóan Pollard như sau: Giả sử n là B-mịn. Kí hiệu Q là bội chung bé nhất của tất cả các lũy thừa của các số nguyên tố ≤ B mà bản thân chúng ≤ n. Nếu $q^l \le n$ thì $l \ln q \le \ln n$, tức $l \le \left\lfloor \frac{\ln n}{\ln q} \right\rfloor$ ($\lfloor x \rfloor$ là số nguyên bé nhất lớn hơn x)

Ta có:

$$Q = \prod_{q \le B} q^{\lfloor \ln n / \ln q \rfloor}$$

Trong đó tích lấy theo tất cả các số nguyên tố khác nhau q ≤ B. Nếu p là một thừa số nguyên tố của n sao cho p – 1 là B-mịn thì p-1|Q và do đó với mọi a bất kì thỏa mãn gcd(a, p) = 1, theo định lý Fermat ta có $a^Q \equiv 1 \bmod p$. Vì vậy, nếu lấy d = gcd(a^Q - 1 ,n) thì p|d. Nếu d = n thì coi như thuật toán không cho ta điều mong muốn, tuy nhiên điều đó chắc không xảy ra nếu n có ít nhất hai thừa số nguyên tố khác nhau. Từ những lập luận đó ta có:

(p – 1)-thuật toán Pollard phân tích thành thừa số:

VÀO: một hợp số n không phải lũy thừa của một số nguyên tố

RA: một thừa số không tầm thường của n.

1. Chọn một cận cho độ mịn B

2. Chọn ngẫu nhiên một số nguyên a, $2 \le a \le n - 1$, và tính d = gcd(a, n). Nếu $d \ge 2$ thì cho ra kết quả (d)

3. Với mỗi số nguyên tố $q \le B$ thực hiện:

3.1. Tính $l = \left\lfloor \frac{\ln n}{\ln q} \right\rfloor$

3.2. Tính $a \leftarrow a^{q^l} \bmod n$

4. Tính d = gcd(a – 1, n)

5. Nếu $1 < d < n$ thì cho ra kết quả (d). Nếu ngược lại thì thuật toán coi như không có kết quả.

Ví dụ: Dùng thuật toán cho số n = 19048567. Ta chọn B = 19, và a = 3 và tính được gcd(3, n) = 1. Chuyển sang thực hiện bước 3 ta được bảng sau đây (mỗi hàng ứng với một giá trị của q):

Bảng 3-1. Kết quả tính bước 3 của thuật toán Pollard

q	l	a
2	24	2293244
3	15	13555889
5	10	16937223
7	8	15214586
11	6	9685355
13	6	13271154
17	5	11406961
19	5	554506

Sau đó ta tính d = gcd (554506 – 1, 19048567) = 5281. Vậy ta được một thừa số p = 5281, và do đó một thừa số nữa là q = n/p = 3607. Cả hai thừa số đó đều là số nguyên tố.

Chú ý rằng ở đây $p - 1 = 2^5.3.5.11$, có tất cả các ước số nguyên tố đều ≤ 19, do đó chắc chắn thuật toán sẽ kết thúc có kết quả. Thuật toán sẽ kết thúc không có kết quả khi độ mịn B được chọn quá bé để không một thừa số nguyên tố p nào của n mà p – 1 chỉ chứa các ước số nguyên tố ≤ B. Như vậy, có thể xem (p-1)-thuật toán Pollard phân tích n thành thừa số nguyên tố là có hiệu quả đối với những số nguyên n là B-mịn, người ta tính được thời gian cần để thực hiện thuật toán đó là cỡ $O(B\ln n/\ln B)$ phép nhân theo môđuyn.

Bây giờ ta xét trường hợp các số nguyên Blum, tức là các số có dạng n = p.q, tích của hai số nguyên tố lớn. Trước hết ta chú ý rằng nếu ta biết hai số nguyên khác nhau x, y sao cho $x^2 \equiv y^2 \bmod n$ thì ta dễ tìm được một thừa số của n. Thực vậy, từ $x^2 \equiv y^2 \bmod n$ ta có $x^2 - y^2 = (x - y)(x + y)$ chia hết cho n, do n không là ước số của x + y hoặc x – y nên gcd(x – y, n) phải là một ước số của n, tức bằng p hoặc q.

Ta biết nếu n = p.q là số Blum thì phương trình đồng dư

$$x^2 \equiv a^2 \bmod n$$

có 4 nghiệm, hai nghiệm tầm thường là x = a và x = -a. Hai nghiệm không tầm thường khác là ± b, chúng là nghiệm của hai hệ phương trình đồng dư bậc nhất sau đây:

$$\begin{cases} x \equiv a \bmod p \\ x \equiv -a \bmod q \end{cases} \qquad \begin{cases} x \equiv -a \bmod p \\ x \equiv a \bmod q \end{cases}$$

Bằng lập luận như trên ta thấy rằng nếu n là số Blum, a là một số nguyên tố với n và ta biết một nghiệm không tầm thường của phương trình $x^2 \equiv a^2 \bmod n$, tức biết một $x \neq \pm a$ sao cho $x^2 \equiv a^2 \bmod n$ thì gcd(x-a, n) sẽ là một ước số của n. Những điều trên đây là căn cứ cho một số phương pháp tìm ước số nguyên tố của một số nguyên dạng Blum; ý chung của các phương pháp đó là dẫn về việc

tìm một nghiệm không tầm thường của một phương trình dạng $x^2 \equiv a^2 \bmod n$, chẳng hạn như phương trình $x^2 \equiv 1 \bmod n$.

Một trường hợp khá lý thú trong lý thuyết mật mã là khi ta biết hai số a, b là nghịch đảo của nhau theo mod $\phi(n)$ (nhưng không biết $\phi(n)$) và tìm một phân tích thành thừa số của n. Bài toán được đặt ra cụ thể là: Biết n có dạng Blum, biết a và b sao cho $ab \equiv 1 \bmod \phi(n)$. Hãy tìm một ước số nguyên tố của n, hay tìm một nghiệm không tầm thường của phương trình $x^2 \equiv 1 \bmod n$. Ta giả thiết $ab - 1 = 2^s.r$ với r là số lẻ. Ta phát triển một thuật toán xác suất kiểu Las Vegas như sau: Ta chọn một số ngẫu nhiên v ($1 \le v \le n - 1$). Nếu may mắn v là bội số của p hay q, thì ta được ngay một ước số của n là gcd(v, n). Nếu v nguyên tố với n, thì ta tính các bình phương liên tiếp kể từ v^r, được v^r, v^{2r}, v^{4r},… cho đến khi được $v^{2^t.r} \equiv 1 \bmod n$ với một t nào đó. Số t như vậy bao giờ cũng đạt được vì có $2^s.r \equiv 0 \bmod \phi(n)$ nên có $v^{2^t.r} \equiv 1 \bmod n$. Như vậy, ta đã tìm được một số $x = v^{2^{t-1}.r}$ sao cho $x^2 \equiv 1 \bmod n$. Tất nhiên có $x \neq 1 \bmod n$. Nếu cũng có $x \neq -1 \bmod n$ thì x là nghiệm không tầm thường của $x^2 \equiv 1 \bmod n$, từ đó ta có thể tìm ước số của n. Nếu không thì thuật toán coi như thất bại, cho ta kết quả *không đúng*. Người ta có thể ước lượng xác suất cho kết quả *không đúng* với một lần thử với một số v là < 1/2, do đó nếu ta thiết kế thuật toán với m số ngẫu nhiên v_1, …, v_m, thì sẽ có thể đạt được xác suất cho kết quả không đúng là $<1/2^m$!

3.2.2. Hệ mật RSA

3.2.2.1. Thuật toán 1: Tạo khóa.

Tóm lược: Mỗi đầu cần tạo một khoá công khai và một khóa riêng tương ứng theo các bước sau:

a. Tạo 2 số nguyên tố lớn ngẫu nhiên và khác nhau p và q. p và q có độ lớn xấp xỉ nhau.

b. Tính $n = p.q$ và $\Phi(n) = (p-1)(q-1)$.

c. Chọn một số nguyên ngẫu nhiên e, $1 < e < \Phi$, sao cho $(e, \Phi) = 1$.

d. Tính một số nguyên d duy nhất, $1 < d < \Phi$ thoả mãn $ed \equiv 1 (\mathrm{mod}\, \Phi)$.

e. Khoá công khai là cặp số (n, e). Khoá riêng bí mật là d.

3.2.2.2. Thuật toán 2: Mã hóa công khai RSA

Tóm lược: B mã hoá một thông báo m để gửi cho A bản mã cần giải.

Mã hoá: B phải thực hiện:

1. Thu nhận khoá công khai (n, e) của A.
2. Biểu diễn bản tin dưới dạng một số nguyên m trong khoảng $[0, n-1]$
3. Tính $c = m^e \bmod n$.
4. Gửi bản mã c cho A.

Giải mã: Khôi phục bản rõ m từ c. A phải thực hiện phép tính sau bằng cách dùng khoá riêng $m = c^d \bmod n$

Chứng minh hoạt động giải mã:

Vì $ed \equiv 1 (\mathrm{mod}\, \Phi)$ nên luôn tồn tại một số nguyên k sao cho $ed = 1 + k\Phi$. Bây giờ nếu $(m, p) = 1$ theo định lý Ferma ta có: $m^{p-1} \equiv 1 (\mathrm{mod}\, p)$. Luỹ thừa cả hai vế của đồng dư thức trên với số mũ $k(q-1)$ và rồi nhân cả hai vế với m ta có:

$$m^{1+k(q-1)(p-1)} \equiv m (\mathrm{mod}\, p)$$

Mặt khác nếu $\mathrm{UCLN}(m, p) = p$ thì đồng dư thức cuối cùng ở trên vẫn đúng vì mỗi vế đều đồng dư với $0 \bmod p$. Bởi vậy, trong mọi trường hợp ta đều có:

$$m^{ed} \equiv m (\mathrm{mod}\, p)$$

Bằng lập luận tương tự ta lại có: $m^{ed} \equiv m (\mathrm{mod}\, q)$

Cuối cùng vì p và q là các số nguyên tố khác nhau nên $m^{ed} \equiv m (\mathrm{mod}\, n)$ và bởi vậy $c^d \equiv (m^e)^d \equiv m (\mathrm{mod}\, n)$.

3.2.2.3. Ví dụ

Tạo khoá

A chọn các số nguyên tố $p = 2357,\ q = 2551$ và tính $n = p.q = 6012707$ và $\Phi = (p-1)(q-1) = 6007800$. A chọn $e = 3674911$ và dùng thuật toán Euclide mở rộng để tìm được d = 422191 thoả mãn $ed \equiv 1 (\mathrm{mod}\, \Phi)$. Khoá công khai của A là cặp số ($n = 6012707, e = 3674911$), khoá bí mật của A là d = 422191.

Mã hoá

Để mã hoá thông báo m = 5234673, B sử dụng thuật toán lấy luỹ thừa theo modulo để tính.

$$c = m^e \bmod n = 5234673^{3674911} \bmod 6012707 = 3650502$$

rồi gửi c cho A.

Giải mã

Để giải mã bản mã c, A tính:

$$c^d \bmod n = 3650502^{422191} \bmod 6012707 = 5234673$$

Chú ý (Số mũ vạn năng).

Số $\lambda = \mathrm{BCNN}(p-1, q-1)$ đôi khi được gọi là số mũ vạn năng của n, λ có thể được dùng thay cho $\Phi = (p-1)(q-1)$ khi tạo khoá RSA. Cần chú ý rằng λ là ước thực sự của Φ. Sử dụng λ có thể thu được số mũ giải mã d nhỏ hơn (làm cho giải mã nhanh hơn). Tuy nhiên, nếu p và q được chọn ngẫu nhiên thì $\mathrm{UCLN}(p-1, q-1)$ sẽ khá nhỏ và bởi vậy Φ và λ sẽ là các số có kích thước xấp xỉ.

3.2.2.4. Vấn đề điểm bất động trong RSA

Giả sử rằng cặp khóa công khai là $(e, n) = (17, 35)$.

Giả sử thông báo có giá trị bằng 8.

Ta có $8^{17} \equiv 8 \bmod 35$.

Như vậy mã hóa của thông báo vẫn là thông báo ban đầu. Nói một cách khác với khóa mã là 17 thì thông tin không được che dấu. Rõ ràng là phải tránh được tình trạng này định lý sau cho ta tính được số bản tin không thể che dấu được với một lựa chọn cho trước của (e, n).

Định lý:

Nếu các thông báo được mã bằng hệ mật RSA với cặp khóa công khai (e, n) với $n = p.q$ thì số các thông báo không thể che dấu được bằng:

$$N = (1 + UCLN(e-1, p-1))(1 + UCLN(d-1, q-1))$$

Chứng minh:

Một thông báo là không thể che dấu được nếu $M^e \equiv M \bmod n$

Ta có: $M^e \equiv M \bmod p$ và $M^e \equiv M \bmod q$.

Ta có thể viết lại các phương trình trên như sau:

$$M^{e-1} \equiv 1 \bmod p \text{ hoặc } M^{e-1} \equiv 0 \bmod p$$

$$M^{e-1} \equiv 1 \bmod q \text{ hoặc } M^{e-1} \equiv 0 \bmod q$$

Chú ý rằng phương trình đồng dư $M^{e-1} \equiv 0 \bmod p$ chỉ có một nghiệm tương tự với q ta có được kết quả của định lý

Ví dụ: $n = 35$

Giả sử e = 3 ta có $(1 + UCLN(2,4))(1 + UCLN(2,6)) = 9$

Các thông báo không thể che dấu được là 9 thông báo sau:

$$\{0, 1, 6, 14, 15, 20, 21, 29, 34\}$$

Giả sử e = 17. ta có $(1 + UCLN(6,4))(1 + UCLN(16,6)) = 15$

Các thông báo không thể che dấu được là 15 thông báo sau:

$$\{0, 1, 6, 7, 8, 13, 14, 15, 20, 21, 22, 27, 28, 29, 34\}$$

Giả sử $p = 2p'+1$ và $q = 2q'+1$ trong đó p' và q' là các số nguyên tố. Khi đó:

$$UCLN(e-1, 2p') = 1; 2 \text{ hoặc } p'$$

Nếu $UCLN(e-1, 2p')$ không phải là p' và $UCLN(e-1, 2q')$ không phải là q' thì số thông báo không thể che dấu chỉ nhiều nhất là 9.

Nếu $UCLN(e-1, 2p') = p'$ thì số các thông báo không thể che dấu tối thiểu là $2(p'+1)$. Tuy nhiên xác suất để xảy ra điều này là rất nhỏ $(\text{bằng } 1/p')$

3.2.3. Hệ mật Rabin

3.2.3.1. Thuật toán 1: Tạo khóa

Tóm lược: Mỗi đầu tạo một khoá công khai và một khoá bí mật tương ứng theo các bước sau:

(1) Tạo 2 số nguyên tố lớn, ngẫu nhiên và phân biệt p và q có kích thước xấp xỉ nhau.

(2) Tính $n = p.q$.

(3) Khoá công khai là n, khoá bí mật là các cặp số (p, q).

3.2.3.2. Thuật toán 2: Mã hóa công khai Rabin

Mã hoá: B phải thực hiện các bước sau:

(1) Nhận khoá công khai của A: n.

(2) Biểu thị bản tin dưới dạng một số nguyên m nằm trong dải $[0, n-1]$.

(3) Tính $c = m^2 \bmod n$.

(4) Gửi bản mã c cho A.

Giải mã: Để khôi phục bản rõ m từ c, A phải thực hiện các bước sau:Tìm 4 căn bậc hai của $c \bmod n$ là m_1, m_2, m_3 hoặc m_4.

(1) Thông báo cho người gửi là một trong 4 giá trị m_1, m_2, m_3 hoặc m_4. Bằng một cách nào đó A sẽ quyết định m là giá trị nào.

3.2.3.3. Ví dụ

Tạo khoá.

A chọn các số nguyên tố p = 277 và q = 331. A tính n = p . q = 91687. Khoá công khai của A là 91687. Khoá bí mật của A là cặp số (p = 277 , q = 331).

Mã hoá

Giả sử rằng 6 bit cuối cùng của bản tin gốc được lặp lại trước khi thực hiện mã hoá. Việc thêm vào độ thừa này nhằm giúp cho bên giải mã nhận biết được bản mã đúng.

Để mã hoá bản tin 10 bit $\overline{m} = 1001111001$, B sẽ lặp lại 6 bit cuối cùng của $\overline{m}$ để có được bản tin 16 bit sau: m = 1001111001111001, biểu diễn thập phân tương ứng là m = 40596.

Sau đó B tính $c = m^2 \bmod n = 40596^2 \bmod 91687 = 62111$ rồi gửi c cho A

Giải mã

Để giải mã bản mã c, A tính bốn giá trị căn bậc 2 của $c \bmod n$:

$$m_1 = 69654, \quad m_2 = 22033, \quad m_3 = 40596, \quad m_4 = 51118$$

Biểu diễn nhị phân tương ứng của các số trên là:

$$m_1 = 10001000000010110 , \qquad m_2 = 101011000010001$$
$$m_3 = 1001111001111001 , \qquad m_4 = 1100011110101110$$

Vì chỉ có m_3 mới có độ thừa cần thiết nên A sẽ giải mã c bằng m_3 và khôi phục lại bản tin gốc là $\overline{m} = 1001111001$.

Đánh giá hiệu quả

Thuật toán mã hoá Rabin là một thuật toán cực nhanh vì nó chỉ cần thực hiện một phép bình phương modulo đơn giản. Trong khi đó, chẳng hạn với thuật toán RSA có e = 3 phải cần tới một phép nhân modulo và một phép bình phương modulo. Thuật toán giải mã Rabin có chậm hơn thuật toán mã hoá, tuy nhiên về mặt tốc độ nó cung tương đương với thuật toán giải mã RSA.

3.3. BÀI TOÁN LOGARIT RỜI RẠC VÀ CÁC HỆ MẬT CÓ LIÊN QUAN

3.3.1. Bài toán logarit rời rạc

Giả sử cho Z_p là một trường hữu hạn với p là một nguyên tố lớn.

Cho g là phần tử sinh của nhóm nhân Z_p^* tức là với một phần tử $a \neq 0$ bất kỳ thuộc Z_p ta có thể tìm được một số nguyên tố x duy nhất thỏa mãn:

$$a = g^x$$

Ta có thể viết: $\log_g a = x$.

Bài toán logarit rời rạc chính là bài toán tìm x.

Ví dụ: Xét Z_{19}, phần tử sinh g = 2. Ta có bảng sau:

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_2 x$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9

Từ bảng trên ta có: $2^{13} \equiv 3 \bmod 19$.

Nhìn chung đây là một bài toán rất khó khi p đủ lớn (chẳng hạn $p \approx 10^{200}$). Khi đó ngay cả với các máy tính cực mạnh ta cũng phải chịu bó tay. Tuy nhiên, trên thực tế bài toán này chỉ thực sự khó khi $p-1$ không phải là tích của các số nguyên tố nhỏ. Nói chung bài toán logarit rời rạc trên trường hữu hạn $GF(p)$ có độ phức tạp lớn hơn so với trên $GF(2^m)$.

Thuật toán bước đi lớn bước đi nhỏ (Baby step giant step):

VÀO: số nguyên n, phần tử sinh α của Z_n^*, và phần tử $\beta \in Z_n^*$

RA: $\log_\alpha \beta$ trên Z_n^*

1. Tính $m = \lceil \sqrt{ord(\alpha)} \rceil$
2. Lập bảng (j, $\alpha^j \bmod n$) với $j = \overline{0 \to m-1}$

3. Tính $\beta.(\alpha^{-m})^i \bmod n$ với $i = \overline{0 \to m-1}$

4. Tra bảng $(j, \alpha^j \bmod n)$ cho tới khi thỏa mãn $\beta.(\alpha^{-m})^i = \alpha^j$

5. Khi đó $\boldsymbol{log_\alpha \beta = m.i + j}$

3.3.2. Thuật toán trao đổi khóa Diffie – Hellman

Các bên A và B mỗi bên gửi cho nhau một thông báo trên kênh mở và sẽ đạt được khóa chia sẻ K giữa hai bên A và B

1. Thiết lập một lần: Một số nguyên tố phù hợp p và phần tử sinh $\alpha \in Z_p^*$ với $2 \le \alpha \le p - 2$ được lựa chọn và công bố.

2. Các thông báo của giao thức:

$A \to B: \alpha^x \bmod p$ (1)

$B \to A: \alpha^y \bmod p$ (2)

3. Các hành động của giao thức: thực hiện các bước sau đây mỗi khi khóa chia sẻ được yêu cầu:

(a) A chọn số ngẫu nhiên số nguyên bí mật x, $1 \le x \le p - 2$ và gửi thông báo (1) cho B.

(b) B chọn số ngẫu nhiên số nguyên bí mật y, $1 \le y \le p - 2$ và gửi thông báo (2) cho A.

(c) B nhận được α^x và tính khóa chia sẻ là $K = (\alpha^x)^y \bmod p$

(d) A nhận được α^x và tính khóa chia sẻ là $K = (\alpha^y)^x \bmod p$

3.3.3. Hệ mật Elgamal

3.3.3.1. Thuật toán tạo khóa

Tóm lược: Mỗi đầu liên lạc tạo một khoá công khai và một khoá bí mật tương ứng :

(1) Tạo 1 số nguyên tố p lớn và một phần tử sinh α của nhóm nhân Z_p^* của các số nguyên $\bmod p$.

(2) Chọn một số nguyên ngẫu nhiên a, $1 \le a \le p-2$ và tính $\alpha^a \bmod p$.

(3) Khoá công khai là bộ 3 số (p, α, α^a), khoá bí mật là a.

3.3.3.2. Thuật toán mã hóa công khai Elgamal

Tóm lược: B mã hoá một thông tin báo m để gửi cho A bản mã cần gửi.

Mã hoá: B phải thực hiện các bước sau:

(1) Nhận khoá công khai (p, α, α^a) của A.

(2) Biểu thị bản tin dưới dạng một số nguyên m trong dải $\{0,1,\ldots,p-1\}$.

(3) Chọn số nguyên ngẫu nhiên k, $1 \le k \le p-2$

(4) Tính $\gamma = \alpha^k \bmod p$ và $\delta = m(\alpha^a)^k \bmod p$.

(5)Gửi bản mã $c = (\alpha, \delta)$ cho A.

Giải mã: Để khôi phục bản rõ m từ c, A phải thực hiện các bước sau:

(1)Sử dụng khoá riêng a để tính $\gamma^{p-1-a} \bmod p$

(Chú ý $\gamma^{p-1-a} = \gamma^{-a} = \gamma^{-ak}$)

(2)Khôi phục bản rõ bằng cách tính $(\gamma^{-a})\delta \bmod p$.

Chứng minh hoạt động giải mã:

Thuật toán trên cho phép A thu được bản rõ vì:

$$\gamma^{-a}\,\delta \equiv \alpha^{-ak}.m\,\alpha^{ak} \equiv m \bmod p$$

3.3.3.3. Ví dụ:

Tạo khoá.

A chọn p = 2357 và một phần tử sinh $\alpha = 2$ của Z_{2357}^*. A chọn khoá bí mật a = 1751 và tính $\alpha^a \bmod p = 2^{1751} \bmod 2357 = 1185$. Khoá công khai của A là $(p = 2357, \alpha = 2, \alpha^a = 1185)$

Mã hoá

Để mã hoá bản tin m = 2035, B sẽ chọn một số nguyên ngẫu nhiên k = 1520 và tính:

$$\gamma = 2^{1520} \bmod 2357 = 1430$$

và $\delta = 2035.1185^{1520} \bmod 2357 = 697$

Sau đó B gửi $c = (1430, 697)$ cho A

Giải mã

Để giải mã A phải tính:

$$\gamma^{p-1-a} = 1430^{605} \bmod 2357 = 872$$

Sau đó khôi phục bản rõ m bằng cách tính: $m = 872.697 \bmod 2357 = 2035$.

3.4. BÀI TOÁN XẾP BALÔ VÀ HỆ MẬT MERKLE – HELLMAN

3.4.1. Định nghĩa dãy siêu tăng

Định nghĩa: Dãy các số nguyên dương $(a_1, a_2, \ldots, a_n)$ được gọi là dãy siêu tăng nếu $a_i > \sum_{j=1}^{i-1} a_j$ với $\forall i, 2 \le i \le n$

3.4.2. Bài toán xếp balô

Cho một đống các gói có các trọng lượng khác nhau, liệu có thể xếp một số gói này vào ba lô để ba lô có một trọng lượng cho trước hay không. Về mặt hình thức ta có thể phát biểu bài toán trên như sau:

Cho tập các giá trị $M_1, M_2, \ldots, M_n$ và một tổng S. Hãy tính các giá trị b_i để:

$$S = b_1M_1 + b_2 M_2 + \ldots + b_n M_n$$

với $b_i \in \{0,1\}$

$b_i = 1$: Có nghĩa là gói M_i được xếp vào ba lô.

$b_i = 0$: Có nghĩa là gói M_i không được xếp vào ba lô.

3.4.3. Giải bài toán xếp balô trong trường hợp dãy siêu tăng

Trong trường hợp $M = \{M_1, M_2, \ldots, M_n\}$ là một dãy siêu tăng thì việc tìm $b = (b_1, b_2, \ldots, b_n)$ tương đương như bài toán tìm biểu diễn nhị phân của một số S. Biểu diễn này sẽ tìm được sau tối đa là n bước.

Thuật toán giải:

VÀO: Dãy siêu tăng $M = \{M_1, M_2, \ldots, M_n\}$ và một số nguyên S là tổng của một tập con trong M

RA : $(b_1, b_2, \ldots, b_n)$ trong đó $b_i \in \{0,1\}$ sao cho: $\sum_{i=1}^{n} b_i M_i = S$

(1) $i \leftarrow n$

(2) Chừng nào $i \geq 1$ hãy thực hiện

a. Nếu $S \geq M_i$ thì : $x_i \leftarrow 1$ và $S \leftarrow S - M_i$ ngược lại: $x_i \leftarrow 0$

b. $i \leftarrow i - 1$

(3) Return (b)

Nếu M không phải là dãy siêu tăng thì lời giải của bài toán là một trong 2^n phương án có thể . Đây là một bài toán khó giải nếu n lớn.

3.4.4. Thuật toán mã công khai Merkle – Hellman

Tóm lược: B mã hoá bản tin m để gửi cho A bản mã cần phải giải mã.

Mã hoá: B phải thực hiện các bước sau:

(1) Nhận khoá công khai của A: $(a_1, a_2, \ldots, a_n)$

(2) Biểu thị bản tin m như một chuỗi nhị phân có độ dài n $m = m_1, m_2, \ldots, m_n$.

(3) Tính số nguyên $c = m_1 a_1 + m_2 a_2 + \ldots + m_n a_n$

(4) Gửi bản mã c cho A.

Giải mã: Để khôi phục bản rõ m từ c, A phải thực hiện các bước sau:

(1) Tính $d = W^{-1} c \bmod M$

(2)Sử dụng thuật giải xếp ba lô trong trường hợp dãy siêu tăng để tìm các số nguyên $r_1, r_2, \ldots, r_n$, $r_i \in \{0,1\}$ sao cho:

$$d = r_1 M_1 + r_2 M_2 + \ldots + r_n M_n$$

(3)Các bit của bản rõ là $m_i = r_{\pi(i)}$, $i = 1, 2, \ldots, n$

Chứng minh: Thuật toán trên cho phép A thu được bản rõ vì:

$$d \equiv W^{-1} c \equiv W^{-1} \sum_{i=1}^{n} m_i a_i \equiv \sum_{i=1}^{n} m_i M_{\pi(i)} \bmod M$$

Vì $0 \le d < M$, $d = \sum_{i=1}^{n} m_i M_{\pi(i)} \bmod M$, bởi vậy nghiệm của bài toán xếp ba lô ở bước (b) sẽ cho ta các bit của bản rõ sau khi sử dụng phép hoán vị π

3.4.5. Ví dụ:

Tạo khoá.

Cho n = 6. A chọn dãy siêu tăng sau: (12, 17, 33, 74, 157, 316), M = 737, W = 635 thoả mãn (W, M) = 1.

Phép hoán vị π của {1, 2, 3, 4, 5, 6} được xác định như sau:

$$\pi(1) = 3,\ \pi(2) = 6,\ \pi(3) = 1,\ \pi(4) = 2,\ \pi(5) = 5,\ \pi(6) = 4$$

Khoá công khai của A là tập (319, 196, 250, 477, 200, 559)

Khoá bí mật của A là $(\pi, M, W(12, 17, 33, 74, 157, 316))$

Mã hoá

Để mã hoá bản tin m = 101101, B tính:

$$c = 319 + 250 + 477 + 559 = 1605$$

và gửi c cho A.

Giải mã

Để giải mã A phải tính: $\left(W^{-1} = -224 = 513\right)$

$$d = W^{-1} c \bmod M = 136$$

và giải bài toán xếp ba lô trong trường hợp dãy siêu tăng sau:

$$136 = 12r_1 + 17r_2 + 33r_3 + 74r_4 + 157r_5 + 316r_6$$

và nhận được $136 = 12 + 17 + 33 + 74$

Bởi vậy $r_1 = r_2 = r_3 = r_4 = 1 \quad r_5 = r_6 = 0$

Sử dụng phép hoán vị π sẽ tìm được các bit của bản rõ như sau:

$$m_1 = r_3 = 1, \quad m_2 = r_6 = 0, \quad m_3 = r_1 = 1, \quad m_4 = r_2 = 1, \quad m_5 = r_5 = 0$$
$$m_6 = r_4 = 1$$

Vậy bản rõ m = 101101.

3.5. HỆ MẬT CHOR-RIVEST (CR)

Hệ mật CR là hệ mật khoá công khai xếp ba lô duy nhất hiện nay không sử dụng pháp nhân modulo để nguỵ trang bài toán tổng tập con.

3.5.1. Thuật toán tạo khoá.

Tóm lược: Mỗi bên liên lạc tạo một khoá công khai và một khoá riêng tương ứng. A thực hiện các bước sau:

(1) Chọn một trường hữu hạn F_q có đặc số q, trong đó $q = p^h$, $p \geq h$ và đối với nó bài toán logarit rời rạc là khó giải.

(2) Chọn một đa thức bất khả quy định chuẩn ngẫu nhiên $f(x)$ bậc h trên Z_p. Các phần tử của F_q sẽ được biểu diễn bằng các đa thức trong $Z_p[x]$ có bậc nhỏ hơn h với phép nhân được thực hiện theo $\bmod f(x)$.

(3) Chọn một phần tử nguyên thuỷ ngẫu nhiên $g(x)$ của F_q.

(4) Với mỗi phần tử của trường cơ sở $i \in Z_P$, tìm logarit rời rạc $a_i = \log_{g(x)}(x+i)$ của các phần tử $x+i$ theo cơ số $g(x)$.

(5) Chọn một phép hoán vị ngẫu nhiên π trên các số nguyên $\{1,2,\ldots,p-1\}$.

(6) Chọn một số nguyên ngẫu nhiên d, $0 \le d \le p^h - 2$

(7) Tính $C_i = (a_{\pi(i)} + d) \bmod (p^h - 1), 0 \le i \le p-1$.

(8) Khoá công khai của A là $((C_0, C_1, \ldots, C_{p-1}), p, h)$

Khoá riêng của A là $(f(x), g(x), \pi, d)$.

3.5.2. Thuật toán mã hoá.

Tóm lược: B mã hoá thông báo m để gửi cho A.

Mã hoá: B thực hiện các bước sau:

a) Nhập khoá công khai của A $((C_0, C_1, \ldots, C_{p-1}), p, h)$

b) Biểu diễn thông báo như một xâu bit có độ dài $\left[\lg\binom{p}{h}\right]$ trong đó $\binom{p}{h} = \frac{p!}{h!(p-h)!}$.

c) Xem m như là biểu diễn nhị phân của một số nguyên. Biến đổi số nguyên này thành môt véctơ nhị phân $M = (M_0, M_1, \ldots, M_{p-1})$ có độ dài p và có đúng h con 1 như sau:

i. Đặt $l \leftarrow h$

ii. For i from 1 to n do:

Nếu $m \ge \binom{P-i}{l}$ thì đặt $M_{i-1} \leftarrow 1, m \leftarrow m - \binom{p-i}{l}, l \leftarrow l-1$. Nếu không thì đặt

$$M_{i-1} \leftarrow 0 \quad \left(CY: \binom{n}{0} = 1 \qquad n \ge 0 \right.$$

$$\left. \binom{0}{1} = 0 \qquad l \ge 1 \right)$$

d) Tính $c = \sum_{i=1}^{p-1} M_i c_i \bmod (p^h - 1)$.

e) Gửi bản mã c cho A.

Giải mã.

Để khôi phục bản mã rõ m từ c, A phải thực hiện các bước lệnh sau:

a) Tính $r = (c - hd) \bmod (p^h - 1)$

b) Tính $u(x) = g^r(x) \bmod f(x)$

c) Tính $s(x) = u(x) + f(x)$ là một đa thức định chuẩn h trên Z_p.

d) Phân tích $s(x)$ thành các nhân tử bậc nhất trên Z_p.

$$s(x) = \prod_{j-1}^{h} (x + t_j) \text{ trong đó } t_j \in Z_p$$

e) Các thành phần có giá trị 1 của vectơ M có các chỉ số là $\pi^{-1}(t_j)$ với $1 \le j \le h$.

Các thành phần còn lại bằng 0

f) Thông báo m được khôi phục lại từ M như sau

i. Đặt $m \leftarrow 0, l \leftarrow h$

ii. For i from 1 to p do:

Nếu $M_{i-1} = 1$ thì đặt $m \leftarrow m + \binom{p-i}{l}, l \leftarrow l - 1$.

Chứng minh hoạt động giải mã:

Ta thấy

$$\begin{aligned} u(x) &= g^2(x) \bmod f(x) \\ &\equiv [g(x)]^{c-hd} \equiv [g(x)]^{\left(\sum_{i=0}^{p-1} M_i c_i\right) - hd} \\ &\equiv [g(x)]^{\left(\sum_{i=0}^{p-1} M_i (a_{\pi(i)} + d)\right) - hd} \\ &\equiv [g(x)]^{\sum_{i=0}^{p-1} M_i a_{\pi(i)}} \bmod f(x) \end{aligned}$$

$$u(x) \equiv \prod_{i=0}^{p-1} \left[g(x)^{a_{\pi(i)}}\right]^{M_i} \equiv \prod_{i=0}^{p-1} (x + \pi(i))^{M_i} \pmod{f(x)}$$

Vì $\prod_{i=0}^{p-1} (x + \pi(i))^{M_i}$ và $s(x)$ là các đa thức định chuẩn bậc h và đồng dư với nhau theo modulo $f(x)$ nên $s(x) = u(x) + f(x) = \prod_{i=0}^{p-1} (x + \pi(i))^{M_i}$

Bởi vậy tất cả các căn bậc h của $s(x)$ đều nằm trong Z_p và áp dụng π^{-1} đối với các căn này ta sẽ có các toạ độ của M là 1

3.5.3. Ví dụ.

Tạo khoá: A thực hiện các bước sau:

(1) Chọn $p = 7$ và $h = 4$.

(2) Chọn đa thức bất khả quy $f(x) = x^4 + 3x^3 + 5x^2 + 6x + 2$ có bậc 4 trên Z_7. Các phần tử của trường hữu hạn F_{7^4} được biểu diễn bằng các đa thức trong $Z_7[x]$.

(3) Chọn phần tử nguyên thuỷ ngẫu nhiên $g(x) = 3x^3 + 3x^2 + 6$.

(4) Tính các logarit rời rạc sau:

$$\begin{aligned}
a_0 &= \log_{g(x)}(x) &&= 1028\\
a_1 &= \log_{g(x)}(x+1) &&= 1935\\
a_2 &= \log_{g(x)}(x+2) &&= 2054\\
a_3 &= \log_{g(x)}(x+3) &&= 1008\\
a_4 &= \log_{g(x)}(x+4) &&= 379\\
a_5 &= \log_{g(x)}(x+5) &&= 1780\\
a_6 &= \log_{g(x)}(x+6) &&= 223
\end{aligned}$$

(5) Chọn phép hoán vị ngẫu nhiên trên $\{0,1,2,3,4,5,6\}$ như sau:

$$\begin{aligned}
&\pi(0) = 6 && \pi(3) = 2 && \pi(5) = 5\\
&\pi(1) = 4 && \pi(4) = 1 && \pi(6) = 3\\
&\pi(2) = 0
\end{aligned}$$

(6) Chọn số nguyên ngẫu nhiên $d = 1702$

(7) Tính

$$\begin{aligned}
C_0 &= (a_6 + d) \bmod 2400 &&= 1925\\
C_1 &= (a_4 + d) \bmod 2400 &&= 2081\\
C_2 &= (a_0 + d) \bmod 2400 &&= 330\\
C_3 &= (a_2 + d) \bmod 2400 &&= 1356
\end{aligned}$$

$$C_4 = (a_1 + d) \bmod 2400 = 1237$$
$$C_5 = (a_5 + d) \bmod 2400 = 1082$$
$$C_6 = (a_3 + d) \bmod 2400 = 310$$

(8) Khoá công khai của A là $((C_0, C_1, C_2, C_3, C_4, C_5, C_6), p = 7, h = 4)$

Khoá bí mật của A là $(f(x), g(x), \pi, d)$

Mã hoá.

Để mã hoá bản tin $m = 22$ gửi cho A, B làm như sau:

(1) Nhận khoá công khai của A.

(2) Biểu diễn m như một xâu bit độ dài 5: $m = 10110$ (Chú ý rằng $\left[\lg\binom{7}{4}\right] = 5$)

(3) Dùng phương pháp đã nêu ở trên bước c trong thuật toán trên để biến đổi m thành véctơ nhị phân M có độ dài M: $M = (1,0,1,1,0,0,1)$

(4) Tính $C = (C_0 + C_2 + C_3 + C_6) \bmod 2400 = 1521$

(5) Gửi $C = 1521$ cho A

Giải mã:

(1) Tính $r = (c - hd) \bmod 2400 = 1913$

(2) Tính $u(x) = g(x)^{1913} \bmod f(x) = x^3 + 3x^2 + 2x + 5$

(3) Tính $g(x) = u(x) + f(x) = x^4 + 4x^3 + x^2 + x$

(4) Phân tích $s(x) = x(x+2)(x+3)(x+6)$

(Do đó $t_1 = 0, t_2 = 2, t_3 = 3, t_4 = 6$)

(5) Các thành phần của M bằng 1 có các chỉ số

$\pi^{-1}(0) = 2 \quad \pi^{-1}(2) = 3 \quad \pi^{-1}(3) = 6 \quad \pi^{-1}(6) = 0$

Bởi vậy $M = (1,0,1,1,0,0,1)$

(6) Sử dụng bước f trong thuật toán giải mã để biến đổi M thành số nguyên $m = 22$ và như vậy khôi phục được bản rõ ban đầu

Chú ý:

- Hệ mật này được xem là an toàn nếu không bị lộ khoá bí mật.
- Có thể mở rộng hệ mật này cho trường hợp Z_p với p là luỹ thừa của một số nguyên tố .

- Để làm cho bài toán logarit rời rạc là dễ giải, các tham số p và h phải chọn sao cho $q = p^h - 1$ chỉ có các nhân tử có giá trị nhỏ.
- Trong thực tế kích thước khuyến nghị của các tham số là $p \approx 200, h \approx 25$ (Ví dụ $p = 197$ và $h = 24$)
- Trở ngại lớn nhất của thuật toán là khoá công khai với kích thước chừng $p.h \log p$ bit là quá lớn. Ví dụ với $p = 197$ và $h = 24$ khoá công khai có chừng 36.000 bit.

3.6. BÀI TOÁN MÃ SỬA SAI VÀ HỆ MẬT MC ELICE

Hệ mật McEliece sử dụng nguyên lý tương tự như hệ mật Merkle-Hellman. Phép giải mã là một trường hợp đặc biệt của bài toán NP đầy đủ nhưng nó được nguỵ trang giống như trường hợp chung của bài toán. Trong hệ thống này bài toán NP được áp dụng ở đây là bài toán giải mã cho một mã sửa sai (nhị phân) tuyến tính nói chung. Tuy nhiên, đối với nhiều lớp mã đặc biệt đều tồn tại các thuật toán giải mã với thời gian đa thức. Một trong những lớp mã này là mã Goppa, chúng được dùng làm cơ sở cho hệ mật McEliece.

3.6.1. Định nghĩa 1.

Giả sử k, n là các số nguyên dương, $k \le n$. Mã $C[n,k]$ là một không gian k chiều của $(Z_2)^n$ (không gian véctơ của tất cả các véctơ nhị phân n chiều).

Ma trận sinh của mã $C[n,k]$ là ma trận nhị phân $k \times n$, các hàng của ma trận này tạo nên cơ sở của C.

Giả sử $x, y \in (Z_2)^n$, trong đó $x = (x_1, \ldots, x_n)$ và $y = (y_1, \ldots, y_n)$. Ta xác định khoảng cách Hamming: $d(x,y) = |\{i : 1 \le i \le n, x_i \ne y_i\}|$ tức là số các toạ độ mà ở đó x và y khác nhau.

Khoảng cách mã C được định nghĩa như sau:

$$d(C) = \min\{d(x,y) : x, y \in C, x \ne y\}$$

Mã $[n,k]$ có khoảng cách d được ký hiệu là mã $[n,k,d]$.

Mã sửa sai được dùng để sửa các sai ngẫu nhiên xảy ra khi truyền số liệu (nhị phân) qua kênh có nhiễu. Điều đó được thực hiện như sau: Giả sử G là một ma trận sinh đối với mã $[n,k,d]$, x là véctơ nhị phân k chiều cần truyền đi. Người gửi Alice sẽ mã hoá x thành một véctơ n chiều $y = xG$ rồi truyền y qua kênh.

Giả sử Bob nhận được véctơ n chiều r không giống y, Bob sẽ giải mã r bằng chiến thuật giải mã "người láng giềng gần nhất". Theo chiến thuật này, Bob sẽ tìm thấy từ y' có khoảng cách tới r nhỏ nhất. Sau đó anh ta giải mã r thành y' , rồi xác định véctơ k chiều x' sao cho $y' = x'G$. Bob hy vọng $y' = y$ và bởi vậy $x' = x$ (tức là Bob tin rằng các sai số trên đường truyền đã được sửa).

Dễ dàng thấy rằng, nếu sai số trên đường truyền nhiều nhất là $(d-1)/2$ thì trên thực tế chiến thuật này sẽ sửa được tất cả các sai.

Ta xét trên thực tế, thuật toán giải mã này được thực hiện như thế nào? Vì $|C| = 2^k$ nên Bob so sánh r với mỗi từ mã anh ta phải kiểm tra 2^k véctơ là một số lớn theo hàm mũ so với k. Nói cách khác, thuật toán này không phải là thuật toán chạy trong thời gian đa thức.

Một biện pháp khác (tạo cơ sở cho nhiều thuật toán giải mã thực tế) dựa trên khái niệm về syndrom. Ma trận kiểm tra tính chẵn lẻ của mã $C[n,k,d]$(có ma trận sinh G) là một mã trận nhị phân $(n-k)xn$ chiều (ký hiệu là H). Các hàng của H sẽ tạo cơ sở cho các phần bù trực giao của C (ký hiệu là $C^{\perp}$) và được gọi là mã đối ngẫu với C. Nói cách khác, các hàng của H là những véctơ độc lập tuyến tính, còn $GH^{\perp}$ là một ma trận không cấp $kx(n-k)$

Cho véctơ $r \in (Z_2)^n$, ta xác định syndrom của r là $Hr^{\perp}$. Syndrom $Hr^{\perp}$ là một véctơ cột có $(n-k)$ thành phần.

3.6.2. Định lý 2

Giả sử C là một mã $[n,k]$ có ma trận sinh G và ma trận kiểm tra tính chẵn lẻ H. Khi đó $x \in (Z_2)^n$ là một từ mã khi và chỉ khi $Hx^T = [0\,0\ldots0]^T$.

Hơn nữa nếu $x \in C, e \in (Z_2)^n$ và $r = x + e$ thì $Hx^T = He^T$.

Ta coi e là vectơ sai xuất hiện trong quá trình truyền từ mã x. Khi đó r biểu diễn vectơ thu được. Định lý trên phát biểu rằng syndrom chỉ phụ thuộc vào các sai số mà không phụ thuộc vào từ mã cụ thể nào được truyền đi.

Điều này gợi ý tới một cách giải mã gọi là *giải mã theo syndrom*. Trước tiên tính $s = Hr^T$ nếu s là một vectơ không, thì ta giải mã r thành r. Nếu không thì ta sẽ lần lượt tạo tất cả các véctơ sai có trọng số 1. Với mỗi véctơ này, ta tính He^T. Nếu có một véctơ e nào đó thoả mãn $He^T = s$ thì ta giải mã r thành $r - e$. Ngược lại, lại tiếp tục tạo các vectơ sai có trọng số $2,3,\ldots,[(d-1)/2]$.

Theo thuật toán này, có thể giải mã cho một vectơ nhận được trong nhiều nhất $1 + \binom{n}{1} + \ldots + \binom{n}{[(d-1)/2]}$ bước.

Phương pháp này làm việc trên một mã tuyến tính bất kỳ. Đối với một số loại mã đặc biệt, thủ tục giải mã có thể nhanh chóng hơn. Tuy nhiên, trên thực tế, cách giải quyết này cho chiến thuật giải mã "người láng giếng gần nhất" vẫn là một bài toán NP đầy đủ. Như vậy, vẫn chưa có một thuật toán giải trong thời gian đa thức đã biết nào cho bài toán giải mã theo "người láng giềng gần nhất" tổng quát. (Khi số các sai số không bị giới hạn bởi $[(d-1)/2]$).

Cũng giống như bài toán tổng tập con, có thể chỉ ra một trường hợp đặc biệt "dễ", sau đó nguỵ trang sao cho nó giống với bài toán chung "khó". Để đưa ra lý thuyết sẽ rất dài dòng, bởi vậy ta sẽ chỉ tóm lược các kết quả ở đây. Một trường hợp khá dễ được McEliece đề nghị là dùng một mã trong lớp các mã Goppa.

Trên thực tế, các mã này có một thuật toán giải mã hữu hiệu. Hơn nữa các, các mã này rất dễ tạo và có một số lượng lớn các mã Goppa tương đương có cùng tham số.

Các tham số của mã Goppa có dạng $n = 2^m, d = 2t + 1$ và $k = n - mt$. Để áp dụng trong thực tế cho một hệ mật khoá công khai, McEliece đề nghị chọn $m = 10$ và $t = 50$. Điều này ứng với mã Goppa $[1024, 524, 101]$. Mỗi bản rõ là một véctơ nhị phân cấp 524 và mỗi bản mã là một véctơ nhị phân cấp 1024. Khoá công khai là một ma trận nhị phân cấp 524×1024 . Hình 3.3 sẽ mô tả hệ mật McEliece.

Cho G là một ma trận sinh của một mã Goppa $C[n, k, d]$, trong đó $n = 2^m$, $d = 2t + 1$ và $k = n - mt$. Cho s là một ma trận khả nghịch cấp $k \times k$ trên Z_2. Giả sử P là một ma trận hoán vị cấp $n \times n$, ta đặt $G' = SGP$. Cho $P = (Z_2)^2$, $C = (Z_2)^n$ và ký hiệu: $K = \{(G, S, P, G')\}$

Trong đó G, S, P được xây dựng như mô tả ở trên và được giữ kín, còn G' được công khai. Với $K = (G, S, P, G')$, ta định nghĩa : $e_k(x, e) = xG' + e$. Ở đây, $e \in (Z_2)^n$ là một véctơ ngẫu nhiên có trọng số t.

Bob giải mã bản mã $y \in (Z_2)^n$ theo các bước sau:

1. Tính $y_1 = yP^{-1}$.
2. Giải mã (Decode) y_1, Bob tìm được $y_1 = x_1 + e_1$, $x_1 \in C$.
3. Tính $x_0 \in (Z_2)^k$ sao cho $x_0 G = x_1$.
4. Tính $x = x_0 S^{-1}$

Hình 3-1. Hệ mật Mc Elice

Ví dụ: Ma trận:

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

là ma trận sinh của mã Hamming $[7,4,3]$. Giả sử Bob chọn ma trận S và ma trận P như sau:

$$S = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \quad \text{và} \quad P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Khi đó ma trận sinh công khai là:

$$G' = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Bây giờ giả sử Alice mã hoá bản rõ $x = (1,\ 1,\ 0,\ 1)$ bằng cách dùng một vectơ sai ngẫu nhiên trọng số 1 có dạng: $e = (0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0)$

Bản mã tính được là:

$$\begin{aligned} y &= x\,G' + e \\ &= (1,1,0,1)\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} + (0,0,0,0,1,0,0) \\ &= (0,1,1,0,0,1,0) + (0,0,0,0,1,0,0) \\ &= (0,1,1,0,1,1,0) \end{aligned}$$

Khi Bob nhận được bản mã y, trước hết anh ta tính

$$y_1 = y P^{-1} = (0, 1, 1, 0, 1, 1, 0)\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = (1, 0, 0, 0, 1, 1, 1)$$

Tiếp theo Bob giải mã y_1 để nhận được $x_1 = (1, 0, 0, 0, 1, 1, 0)$ (Cần để ý là $e_1 \neq e$ do phép nhân với P^{-1})

Sau đó anh ta lập $x_0 = (1, 0, 0, 0)$ (bốn thành phần đầu tiên của x_1).

Cuối cùng Bob tính: $$x = S^{-1} x_0 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}(1, 0, 0, 0) = (1, 1, 0, 1)$$

Đây chính là bản rõ mã Alice đã mã.

3.7. HỆ MẬT TRÊN ĐƯỜNG CONG ELLIPTIC

3.7.1. Các đường cong Elliptic

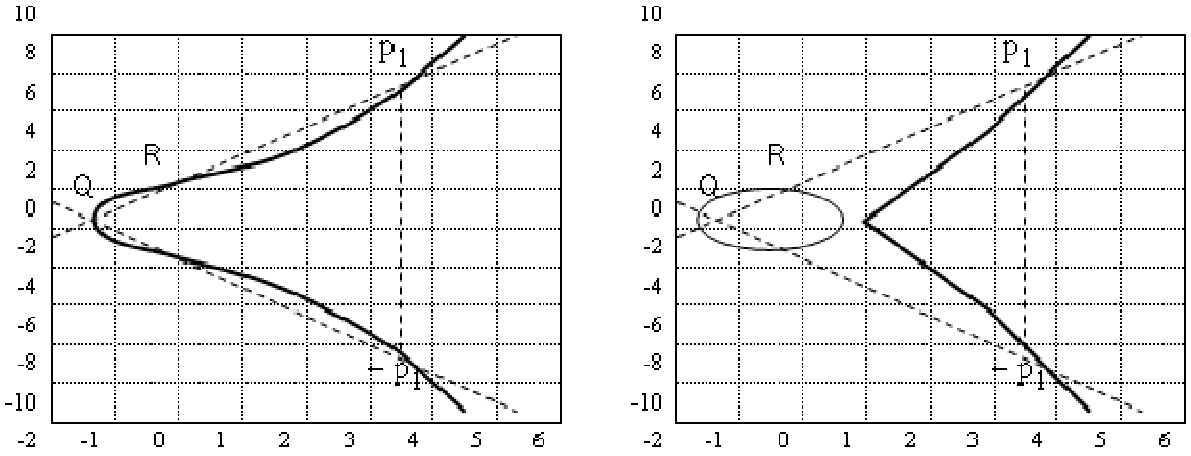
Một đường cong Elliptic là một phương trình bậc 3 có dạng sau:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

Trong đó a, b, c, d, e là các số thực.

Trên các đường cong E ta xác định một phép cộng đặc biệt với một điểm O được gọi là điểm vô cực. Nếu trên đường thẳng cắt đường cong E ở ba điểm thì tổng của chúng bằng điểm vô cực O (điểm O này có vai trò như phần tử đơn vị

trong phép cộng này). Hình 3.1 sau mô tả các đường cong E $y^2 = x^3 + 2x + 5$ và $y^2 = x^3 - 2x + 1$



Hình 3-2. Các đường cong $y^2 = x^3 + 2x + 5$ và $y^2 = x^3 - 2x + 1$

3.7.2. Các đường cong Elliptic trên trường Galois

Một nhóm E trên trường Galois $E_p(a,b)$ nhận được bằng cách tính $x^3 + ax + b \bmod p$ với $0 \le x < p$. Các hằng số a, b là các số nguyên không âm và nhỏ hơn số nguyên tố p và thỏa mãn điều kiện: $4a^3 + 27b^2 \bmod p \neq 0$. Với mỗi giá trị x ta cần xác định xem nó có là một thặng dư bậc hai hay không? Nếu x là thặng dư bậc hai thì có 2 giá trị trong nhóm Elliptic. Nếu x không là thặng dư bậc 2 thì điểm này không nằm trong nhóm $E_p(a,b)$.

Ví dụ: (cấu trúc của một nhóm E)

Giả sử p = 23, a = 1 và b = 1

Trước tiên ta kiểm tra lại:

$$\begin{aligned} 4a^3 + 27b^2 \bmod p &= 4.1^3 + 27.1^2 \bmod 23 \\ &= 4 + 27, \text{od} 23 = 31 \bmod 23 \\ &= 8 \neq 0 \end{aligned}$$

Xét mỗi giá trị có thể $x \in Z_{23}$, tính $x^3 + x + 1 \bmod 23$ và thử giải phương trình đối với y. Với giá trị x cho trước ta có thể kiểm tra xem liệu $z = x^3 + x + 1 \bmod 23$ có phải là một thặng dư bình phương hay không bằng cách áp dụng tiêu chuẩn Euler (*Tiêu chuẩn Euler:* Giả sử p là số nguyên tố, khi đó x là một thặng dư bậc hai theo modulo p khi và chỉ khi: $x^{(p-1)/2} \equiv 1 \bmod p$).

Ta đã có một công thức tường minh để tính các căn bậc hai của các thặng dư bình phương theo modulo p với các số nguyên tố $p \equiv 3 \bmod 4$. Áp dụng công thức này ta có các căn bậc hai của một thặng dư bình phương z là:

$$z^{(23+1)/4} = z^6 \bmod 23$$

Kết quả của phép tính này được nêu trong bảng 3.2 dưới đây:

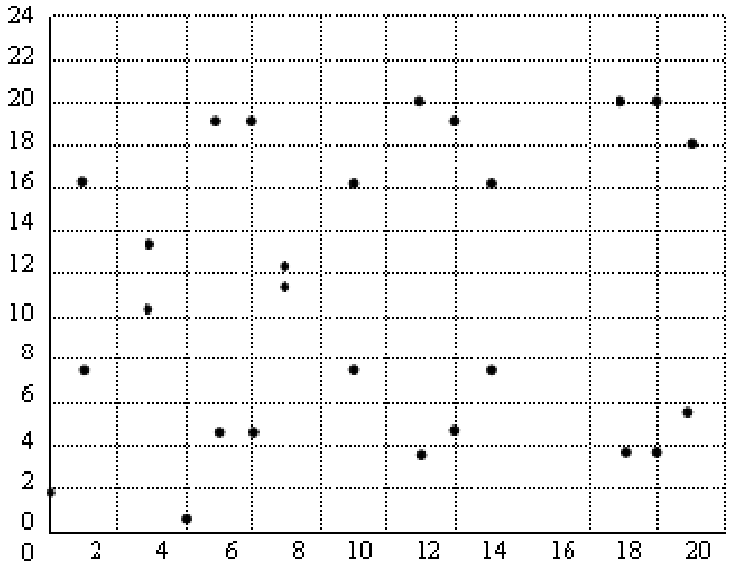
Bảng 3-2. Giá trị y tương ứng với x trên Z_{23}

x	$x^3+x+1 \bmod 23$	Có trong Q_{23}	y
0	1	Có	1, 22
1	3	Có	7, 16
2	11	Không	
3	8	Có	10, 13
4	0	Không	
5	16	Có	4, 19
6	16	Có	4, 19
7	6	Có	11, 12
8	15	Không	
9	3	Có	7, 16
10	22	Không	
11	9	Có	3, 20
12	16	Có	4, 19
13	3	Có	7, 16

14	22	Không	
15	10	Không	
16	19	Không	
17	9	Có	3, 20
18	9	Có	3, 20
19	2	Có	5, 18
20	17	Không	
21	14	Không	
22	22	Không	

Nhóm Elliptic $E_p(a,b) = E_{23}(1,1)$ sẽ gồm các điểm sau:

$$E_{23}(1,1) = \left\{ \begin{matrix} (0,1) & (0,22) & (1,7) & (1,16) & (3,10) & (3,13) & (4,0) \\ (5,4) & (5,19) & (6,4) & (6,19) & (7,11) & (7,12) & (9,7) \\ (9,16) & (11,3) & (11,20) & (12,4) & (12,19) & (13,7) & (13,16) \\ (17,3) & (17,20) & (18,3) & (18,20) & (19,5) & (19,18) & \end{matrix} \right\}$$



Hình 3-3. Nhóm E_{23} (1, 1)

3.7.3. Các phép toán cộng và nhân trên các nhóm E.

Giả sử $P=(x_1,y_1)$, $Q=(x_2,y_2)$ là các điểm trong nhóm $E_p(a,b)$, O là điểm vô cực. Các quy tắc đối với phép cộng trên nhóm con $E_p(a,b)$ như sau:

(1) $P + O = O + P = P$.

(2) Nếu $x_2 = x_1$ và $y_2 = -y_1$ tức là $P=(x_1,y_1)$ và $Q=(x_2,y_2)=(x_1,-y_1)=-P$ thì $P + Q = 0$.

(3) Nếu $Q \neq -P$ thì tổng $P+Q=(x_3,y_3)$ được cho bởi:

$$x^3 = \lambda^2 - x_1 - x_2 \bmod p$$
$$y^3 = \lambda(x_1 - x_3) - y_1 \bmod p$$

Trong đó:

$$\lambda \triangleq \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{nếu } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{nếu } P = Q \end{cases}$$

Ví dụ: Phép nhân trên nhóm $E_p(a,b)$.

Phép nhân trên nhóm $E_p(a,b)$ thực hiện tương tự như phép lũy thừa modulo trong RSA.

Giả sử $P=(3,10)\leftarrow E_{23}(1,1)$, khi đó $2P=(x_3,y_3)$ bằng:

$$2P = P + P = (x_1,y_1) + (x_1,y_1)$$

Vì P = Q và $x_2 = x_1$ nên các giá trị α, x_3 và y_3 là:

$$\lambda = \frac{3x_1^2 + a}{2y_1} \bmod p = \frac{3.3^2 + 1}{2.10} \bmod 23 = \frac{5}{20} \bmod 23 = 4^{-1} \bmod 23 = 6$$

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p = 6^2 - 3 - 3 \bmod 23 = 30 \bmod 23 = 7$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p = 6(3-7) - 10 \bmod 23 = -34 \bmod 23 = 12$$

Bởi vậy $2P = (x_3, y_3) = (7,12)$.

Phép nhân kP nhận được bằng cách thực hiện lặp k lần phép cộng.

Bảng 3-3. Bảng tính kP

	$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ (nếu $P \neq Q$) $\lambda = \frac{3x_1^2 + a}{2y_1}$ (nếu $P = Q$)	x_3 $\lambda^2 - x_1 - x_2 \bmod 23$	y_3 $\lambda(x_1 - x_3) - y_1 \bmod 23$	kP (x_3, y_3)
1				(3,10)
2	6	7	12	(7,12)
3	12	19	5	(19,5)
4	4	17	3	(17,3)
5	11	9	19	(9,16)
6	1	12	4	(12,4)
7	7	11	3	(11,3)
8	2	13	16	(13,16)
9	19	0	1	(0,1)
10	3	6	4	(6,4)
11	21	18	20	(18,20)
12	16	5	4	(5,4)
13	20	1	7	(1,7)

14	13	4	0	(4,0)
15	13	1	16	(1,16)
16	20	5	19	(5,19)
17	16	18	3	(18,3)
18	21	6	19	(6,19)
19	3	0	22	(0,22)
20	19	13	7	(13,7)
21	2	11	20	(11,20)
22	7	12	19	(12,19)
23	1	9	7	(9,7)
24	11	17	20	(17,20)
25	4	19	18	(19,18)
26	12	7	11	(7,11)
27	6	3	13	(3,13)

3.7.4. Mật mã trên đường cong Elliptic.

Trong hệ mật này bản rõ M được mã hóa thành một điểm P_M trong tập hữu hạn các điểm của nhóm $E_p(a,b)$.

Trước hết ta phải chọn một điểm sinh $G \in E_p(a,b)$ sao cho giá trị nhỏ nhất của n đảm bảo $nG = 0$ phải là một số nguyên tố rất lớn. Nhóm $E_p(a,b)$ và điểm sinh G được đưa ra công khai.

Mỗi người dùng chọn một khóa riêng $n_A < n$ và tính khóa công khai P_A như sau: $P_A = n_A G$.

Để gửi thông báo P_M cho bên B, A chọn một số nguyên ngẫu nhiên k và tính cặp bản mã P_C bằng cách dùng khóa công khai P_B của B:

$$P_C = \left[(kG), (P_M + kP_B)\right]$$

Sau khi thu cặp điểm P_C, B sẽ nhân điểm đầu tiên (kG) với khóa riêng n_B của mình rồi cộng kết quả với điểm thứ hai trong cặp điểm P_C (Điểm $(P_M + kP_B)$);

$$(P_M + kP_B) - n_B(kG) = (P_M + kn_BG) - n_B(kG) = P_M$$

Đây chính là điểm tương ứng với bản rõ M. Chỉ có B mới có khóa riêng n_B và mới có thể tách $n_B(kG)$ khỏi điểm thứ hai của P_C để thu thông tin về bản rõ P_M.

Ví dụ:

Xét đường cong E sau: $y^2 = x^3 + ax + b \bmod p$

$$y^2 = x^3 - x + 188 \bmod 751$$
$$(a = -1,\ b = 188,\ p = 751)$$

Nhóm E được tạo từ đường cong E ở trên là:

$$E_p(a,b) = E_{751}(-1, 188)$$

Cho điểm sinh $G = (0,\ 376)$. Khi đó phép nhân kG của G là $(1 \le k \le 751)$.

Nếu A muốn gửi cho B bản rõ m (được mã thành điểm bản rõ P_M) $P_M = (443, 253) \in E_{751}(-1, 188)$ thì A phải dùng khóa công khai của B để mã hóa nó.

Giả sử khóa bí mật của B là $n_B = 85$, khi đó khóa công khai của B là: $P_B = n_BG = 85(0, 376)$

$$P_B = (671,558)$$

A chọn số ngẫu nhiên $k = 113$ và dùng P_B để mã hóa P_M thành cặp điểm bản mã:

$$P_C = [(kG); (P_M + kP_B)]$$
$$P_C = [113.(0,376), (443,253) + (47,416)]$$
$$P_C = [(34,633),(443,253) + (47,416)]$$
$$P_C = [(34, 633), (217, 606)]$$

Dựa vào P_C nhận được, B sẽ dùng khóa riêng $n_B = 85$ để tính P_M như sau:

$$(P_M + kP_B) - n_B(kG) = (217, 606) - [85(34, 633)]$$
$$= (217, 606) - [(47,416)]$$
$$= (217, 606) + (47, 4 - 16) \text{ (vì } -P = (x_1, -y_1))$$
$$= (217, 606) + (47,335) \text{ (vì } -416 \equiv 335 \bmod 751)$$
$$= (443, 253)$$

Sau đó B ánh xạ điểm - điểm bản rõ P_M trở lại thông báo gốc M.

3.7.5. Độ an toàn của hệ mật trên đường cong Elliptic.

Sức mạnh ECC nằm ở sự khó khăn đối với thám mã khi phải xác định số ngẫu nhiên bí mật k từ kP và P. Phương pháp nhanh nhất để giải bài toán này là phương pháp phân tích S - Pollard. Để phá ECC độ phức tạp tính toán khi dùng phương pháp S – Pollard là $3{,}8.10^{10}$ MIPS - năm với kích thước khóa 150 bít (đây là số năm cần thiết với một hệ thống tính toán có tốc độ hàng triệu lệnh/giây). Để so sánh với phương pháp nhanh nhất phá RSA (là phương pháp sàng trường số để phân tích hợp số n thành tích của 2 số nguyên tố p và q) ta thấy rằng với n có kích thước 768 bít độ phức tạp tính toán là: 2.10^8 MIPS - năm , với n có kích thước 1024 bít, độ phức tạp tính tóan là 3.10^{11} năm .

Nếu độ dài khóa của RSA tăng lên tới 2048 bít thì cần 3.10^{20} MIPS - năm, trong khi đó với ECC chỉ cần độ dài khóa là 234 bít đã phải yêu cầu tới $1{,}6.10^{28}$ MIPS - năm.

3.8. ƯU NHƯỢC ĐIỂM CỦA HỆ MẬT KHÓA CÔNG KHAI

Vấn đề còn tồn đọng của hệ mật mã khoá đối xứng được giải quyết nhờ hệ mật mã khoá công khai. Chính ưu điểm này đã thu hút nhiều trí tuệ vào việc đề xuất, đánh giá các hệ mật mã công khai. Nhưng do bản thân các hệ mật mã khoá công khai đều dựa vào các giả thiết liên quan đến các bài toán khó nên đa số các hệ mật mã này đều có tốc độ mã dịch không nhanh lắm. Chính nhược điểm này làm cho các hệ mật mã khoá công khai khó được dùng một cách độc lập.

Một vấn đề nữa nảy sinh khi sử dụng các hệ mật mã khóa công khai là việc xác thực mà trong mô hình hệ mật mã đối xứng không đặt ra. Do các khoá mã côngkhai được công bố một cách công khai trên mạng cho nên việc đảm bảo rằng "khoá được công bố có đúng là của đối tượng cần liên lạc hay không?" là một kẽ hở có thể bị lợi dụng. Vấn đề xác thực này được giải quyết cũng chính bằng các hệ mật mã khoá công khai. Nhiều thủ tục xác thực đã được nghiên cứu và sử dụng như Kerberos, X.509… Một ưu điểm nữa của các hệ mật mã khoá công khai là các ứng dụng của nó trong lĩnh vực chữ ký số, cùng với các kết quả về hàm băm, thủ tục ký để bảo đảm tính toàn vẹn của một văn bản được giải quyết.

3.9. BÀI TẬP

1. Ví dụ về hệ mật RSA. Cho $p = 7$ và $q = 17$.
   a. Tính n.
   b. Cho e (số mũ mã hoá) bằng 5. Hãy tính số mũ giải mã d.
   c. Hãy mã hoá và giải mã cho các số 49 và 12.

2. Người ta biết rằng đối với hệ mật RSA, tập các bản rõ bằng tập các bản mã. Tuy nhiên bạn có cho rằng một số giá trị trong không gian thông báo (bản rõ) là không mong muốn?

3. Trong hệ mật Rabin, giả sử p = 199, q = 211.
   a. Xác định 4 căn bậc hai của 1 mod n, trong đó n = p.q.
   b. Tính bản mã của 32767.
   c. Xác định 4 bản giải mã có thể của bản mã trên.

4. Xét trường hợp đơn giản của hệ mật Merkle-Hellman sử dụng phép hoán vị đồng nhất. Giả sử dãy siêu tăng được chọn là (2, 3. 6, 13, 27, 52) giá trị ngẫu nhiên w được chọn là 31, modulo M được chọn là 105.
   a. Hãy xác định khoá bí mật.
   b. Bản tin ở dạng nhị phân có dạng 011000_110101_101110.
   Hãy tính bản mã và hãy giải mã để tìm lại bản tin ban đầu.

5. Đây là một ví dụ về hệ mật ElGamal áp dụng trong $GF(3^3)$. Đa thức $x^3 + x^2 + 1$ là một đa thức bất khả quy trên $Z_3[x]$ và bởi vậy $Z_3[x]/(x^3 + x^2 + 1)$ chính là $GF(3^3)$. Ta có thể gắn 26 chữ cái của bảng chữ cái tiếng Anh với 26 phần tử khác không của trường và như vậy có thể mã hoá một văn bản thông thường theo cách truyền thống. Ta sẽ dùng thứ tự theo từ điển của các đa thức khác không để thiết lập sự tương ứng.

$A \leftrightarrow 1$	$B \leftrightarrow 2$	$C \leftrightarrow x$
$D \leftrightarrow x+1$	$E \leftrightarrow x+2$	$F \leftrightarrow 2x$
$G \leftrightarrow 2x+1$	$H \leftrightarrow 2x+2$	$I \leftrightarrow x^2$
$J \leftrightarrow x^2+1$	$K \leftrightarrow x^2+2$	$L \leftrightarrow x^2+x$
$M \leftrightarrow x^2+x+1$	$N \leftrightarrow x^2+x+2$	$O \leftrightarrow x^2+2x$
$P \leftrightarrow x^2+2x+1$	$Q \leftrightarrow x^2+2x+2$	$R \leftrightarrow 2x^2$
$S \leftrightarrow 2x^2+1$	$T \leftrightarrow 2x^2+2$	$U \leftrightarrow 2x^2+x$
$V \leftrightarrow 2x^2+x+1$	$W \leftrightarrow 2x^2+x+2$	$X \leftrightarrow 2x^2+2x$
$Y \leftrightarrow 2x^2+2x+1$	$Z \leftrightarrow 2x^2+2x+2$	

Giả sử Bob dùng $\alpha = x$ và a = 11 trong hệ mật ElGamal, khi đó $\alpha^a = x+2$. Hãy chỉ ra cách mà Bob sẽ giải mã cho bản mã sau:

(K, H) (P,X) (N,K) (H, R) (T, F) (V, Y) (E, H) (F, A) (T, W) (J, D) (V, J).

6. Mã BCH (15, 7, 5) có ma trận kiểm tra sau:

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

Hãy giải mã cho các vectơ nhân được sau bằng phương pháp giải mã theo syndrom:

a. r=(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,)

b. r=(1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0)

c. r=(1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0)

CHƯƠNG 4 HÀM BĂM, XÁC THỰC VÀ CHỮ KÍ SỐ

4.1. VẤN ĐỀ XÁC THỰC

Trong các vấn đề liên quan tới an toàn thông tin, ngoài bài toán quan trọng nhất là bảo mật thông tin thì các bài toán kế tiếp là: xác nhận thông báo, xác nhận người gửi (cùng với thông báo), xưng danh và xác nhận danh tính của một chủ thể giao dịch, vv.. Bài toán bảo mật được đáp ứng bằng các giải pháp mật mã đã là nội dung của các chương 2 và 3, trong chương này ta sẽ đề cập đến các bài toán xác nhận và xác thực.

Xác thực mẩu tin liên quan đến các khía cạnh sau khi truyền tin trên mạng

- Bảo vệ tính toàn vẹn của mẩu tin: bảo vệ mẩu tin không bị thay đổi hoặc có các biện pháp phát hiện nếu mẩu tin bị thay đổi trên đường truyền.
- Kiểm chứng danh tính và nguồn gốc: xem xét mẩu tin có đúng do người xưng tên gửi không hay một kẻ mạo danh nào khác gửi.
- Không chối từ bản gốc: trong trường hợp cần thiết, bản thân mẩu tin chứa các thông tin chứng tỏ chỉ có người xưng danh gửi, không một ai khác có thể làm điều đó. Như vậy người gửi không thể từ chối hành động gửi, thời gian gửi và nội dung của mẩu tin.

Ngoài ra có thể xem xét bổ sung thêm các yêu cầu bảo mật như mã hoá. Với mong muốn đáp ứng các yêu cầu trên, có 3 hàm lựa chọn sau đây được sử dụng:

- Mã mẩu tin bằng mã đối xứng hoặc mã công khai.

- Mã xác thực mẩu tin (MAC): dùng khoá và một hàm nén mẩu tin cần gửi để nhận được một đặc trưng đính kèm với mẩu tin và người gửi đó.
- Hàm hash (hàm băm) là hàm nén mẩu tin tạo thành "dấu vân tay" cho mẩu tin.

4.2. HÀM BĂM

4.2.1. Các định nghĩa và tính chất cơ bản

4.2.1.1. Định nghĩa hàm băm

Hàm băm là một hàm h có ít nhất hai tính chất sau:

a) *Tính chất nén:* h sẽ ánh xạ một đầu vào x có độ dài bit hữu hạn tuỳ ý tới một đầu ra $h(x)$ có độ dài bit n hữu hạn.

b) *Tính chất dễ dàng tính toán:* Với h cho trước và một đầu vào x, có thể dễ dàng tính được $h(x)$.

4.2.1.2. Một số tính chất của các hàm băm không có khóa

Giả sử h là một hàm băm không có khoá, x và x' là các đầu vào và y và y' là các đầu ra. Ngoài hai tính chất cơ bản trên ta còn có 3 tính chất sau:

c) *Tính khó tính toán nghịch ảnh:*

Đối với hầu hết các đầu ra được xác định trước, không có khả năng tính toán để tìm một đầu vào bất kỳ mà khi băm sẽ cho ra đầu ra tương ứng (Tức là tìm một nghịch ảnh x' sao cho $h(x') = y$ với y cho trước và không biết đầu vào tương ứng).

d) *Khó tìm nghịch ảnh thứ hai:*

Không có khả năng tính toán để tìm một đầu vào đã cho trước (Tức là với x cho trước phải tìm $x' \neq x$ sao cho $h(x) = h(x')$)

e) *Tính kháng va chạm.* Không có khả năng về tính toán để tìm hai đầu vào khác nhau bất kỳ x và x' để $h(x) = h(x')$.

Hàm băm có thêm ba tính trên được gọi là hàm băm mật mã hay hàm băm an toàn.

4.2.1.3. Định nghĩa hàm băm một chiều (OWHF – one way hash function)

OWHF là một hàm băm (có hai tính chất cơ bản) có tính chất bổ xung là :

- Khó tìm nghịch ảnh
- Khó tìm nghịch ảnh thứ hai.

4.2.1.4. Định nghĩa hàm băm (CRHF: Collision resistant HF)

CRHF là một hàm băm (có hai tính chất cơ bản) có tính chất bổ xung là :

- Khó tìm nghịch ảnh thứ hai
- Khó và chạm

4.2.1.5. Chú ý về các thuật ngữ

Khó tìm nghịch ảnh ≡ Một chiều

Khó tìm nghịch ảnh thứ hai ≡ Kháng va chạm yếu.

Kháng va chạm ≡ Kháng va chạm mạnh

OWHF ≡ Hàm băm một chiều yếu.

CRHF ≡ Hàm băm một chiều mạnh.

4.2.1.6. Ví dụ

r bit kiểm tra của một mã xyclic (n, k) với k > r có thể coi là một hàm băm thoả mãn hai tính chất cơ bản (dễ tính toán và nén). Tuy nhiên nó không thoả mãn tính chất khó tìm nghịch ảnh thứ hai.

4.2.1.7. Định nghĩa mã xác thực thông báo(MAC)

Thuật toán MAC là một họ các hàm h_k (được tham số hoá bằng một khoá bí mật k) có các tính chất sau:

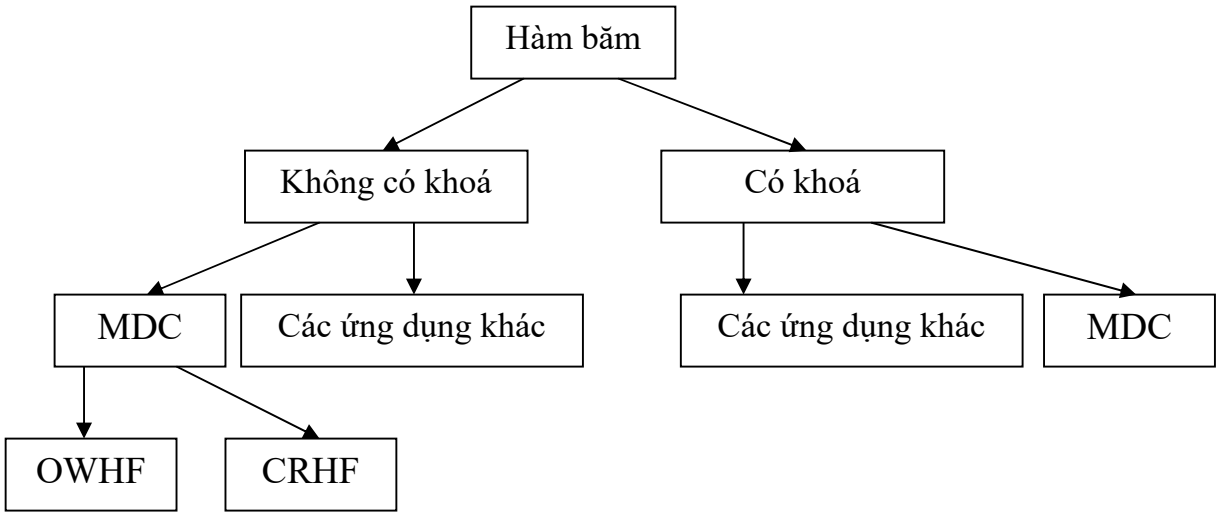
(1) *Dễ dàng tính toán:* Với h_k đã biết, giá trị k cho trước và một đầu vào x, $h_k(x)$ có thể được tính dễ dàng ($h_k(x)$ được gọi là giá trị MAC hay MAC).

(2) *Nén:* h_k ánh xạ một đầu vào x có độ dài bit hữu hạn tuỳ ý tới một đầu ra $h_k(x)$ có độ dài bit n cố định.

(3) *Khó tính toán:* Với các cặp giá trị $(x_i, h_k(x_i))$ không có khả năng tính một cặp $(x, h_k(x))$ với $x \neq x_i$ (kể cả có khả năng $h_k(x) = h_k(x_i)$ với một i nào đó).

Nếu tính chất 3 không thoả mãn thì thuật toán được coi là giả mạo MAC.

4.2.1.8. Phân loại các hàm băm mật mã và ứng dụng



Hình 4-1. Phân loại hàm băm

4.2.2. Các hàm băm không có khóa

4.2.2.1. Định nghĩa 4.1.

Mật mã khối (n, r) là một mã khối xác định một hàm khả nghịch từ các bản rõ n bit sang các bản mã n bit bằng cách sử dụng một khoá r bit. Nếu E là một phép mã hoá như vậy thì $E_k(x)$ ký hiệu cho phép mã hoá x bằng khoá k.

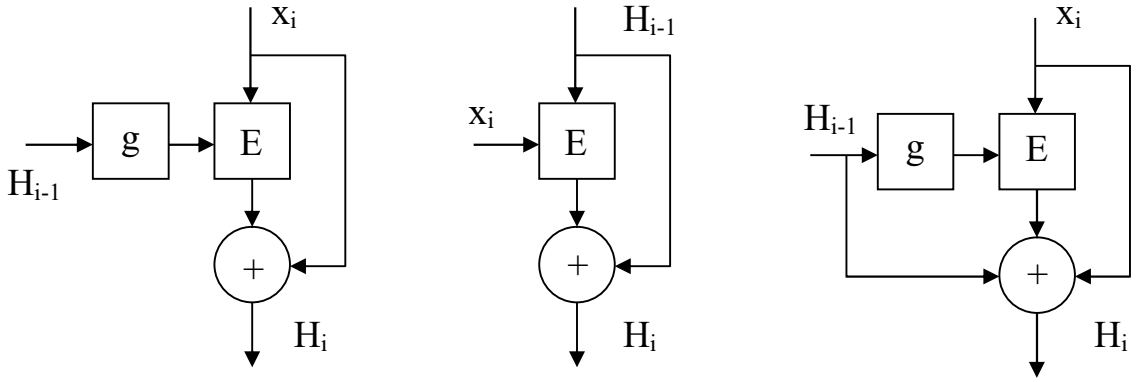
4.2.2.2. Định nghĩa 4.2:

Cho h là một hàm băm có lặp được xây dựng từ một mật mã khối với hàm nén *f* thực hiện s phép mã hoá khối để xử lý từng khối bản tin n bit. Khi đó tốc độ của h là 1/s.

4.2.2.3. MDC độ dài đơn.

Ba sơ đồ dưới đây có liên quan chặt chẽ với các hàm băm độ dài đơn, xây dựng trên các mật mã khối. Các sơ đồ này có sử dụng các thành phần được xác định trước như sau:

- Một mật mã khối n bit khởi sinh E_k được tham số hoá bằng một khoá đối xứng k.
- Một hàm g ánh xạ n bit vào thành khoá k sử dụng cho E (Nếu các khoá cho E cũng có độ dài n thì g có thể là hàm đồng nhất)
- Một giá trị ban đầu cố định IV thích hợp để dùng với E.



Matyas - Mayer - Oseas Davies - Mayer Miyaguchi - Preneel

Hình 4-2. MDC độ dài đơn

Thuật toán băm Matyas - Meyer - Oseas.

VÀO: Xâu bit x

RA : Mã băm n bit của x

(1) Đầu vào x được phân chia thành các khối n bit và được độn nếu cần thiết nhằm tạo khối cuối cùng hoàn chỉnh. Ta được t khối n bit: $x_1\ x_2\ \ldots\ x_t$. Phải xác định trước một giá trị ban đầu n bit (ký hiệu IV).

(2) Đầu ra là H_t được xác định như sau:

$$H_0 = IV\ ,\ \ H_i = E_{g(H_{i-1})}(x_i) \oplus x_i\ \ ,\ \ 1 \le i \le t$$

Thuật toán băm Davies - Meyer

VÀO: Xâu bit x

RA : Mã băm n bit của x

(3) Đầu vào x được phân thành các khối k bit (k là kích thước khoá) và được độn nếu cần thiết để tạo khối cuối cùng hoàn chỉnh. Biểu thị thông báo đã độn thành t khối k bit: $x_1\ x_2\ \ldots\ x_t$. Xác định trước một giá trị ban đầu n bit (ký hiệu IV).

(4) Đầu ra là H_t được xác định như sau:

$$H_0 = IV\ ,\ \ H_i = E_{x_i}(H_{i-1}) \oplus H_{i-1}\ \ ,\ \ 1 \le i \le t$$

Thuật toán băm Miyaguchi - Preneel

Sơ đồ này tương tự như C1 ngoại trừ H_{i-1} (đầu ra ở giai đoạn trước) được cộng $\mathrm{mod}\,2$ với tín hiệu ra ở giai đoạn hiện thời. Như vậy:

$$H_0 = IV\ ,\ \ H_i = E_{g(H_{i-1})}(x_i) \oplus x_i \oplus H_{i-1}\ \ ,\ \ 1 \le i \le t$$

Nhận xét: Sơ đồ D - M có thể coi là sơ đồ đối ngẫu với sơ đồ M - M - O theo nghĩa x_i và H_{i-1} đổi vai trò cho nhau.

4.2.2.4.MDC độ dài kép: MDC -2 và MDC - 4.

MDC -2 và MDC - 4 là các mã phát hiện sự sửa đổi yêu cầu tương ứng là 2 và 4 phép toán mã hoá khối trên mỗi khối đầu vào hàm băm. Chúng sử dụng 2 hoặc 4 phép lặp của sơ đồ M - M - O để tạo ra hàm băm có độ dài kép. Khi dùng DES chúng sẽ tạo ra mã băm 128 bit. Tuy nhiên trong cấu trúc tổng quát có thể dùng các hệ mật mã khối khác MDC-2 và MDC- 4 sử dụng các thành phần xác định như sau:

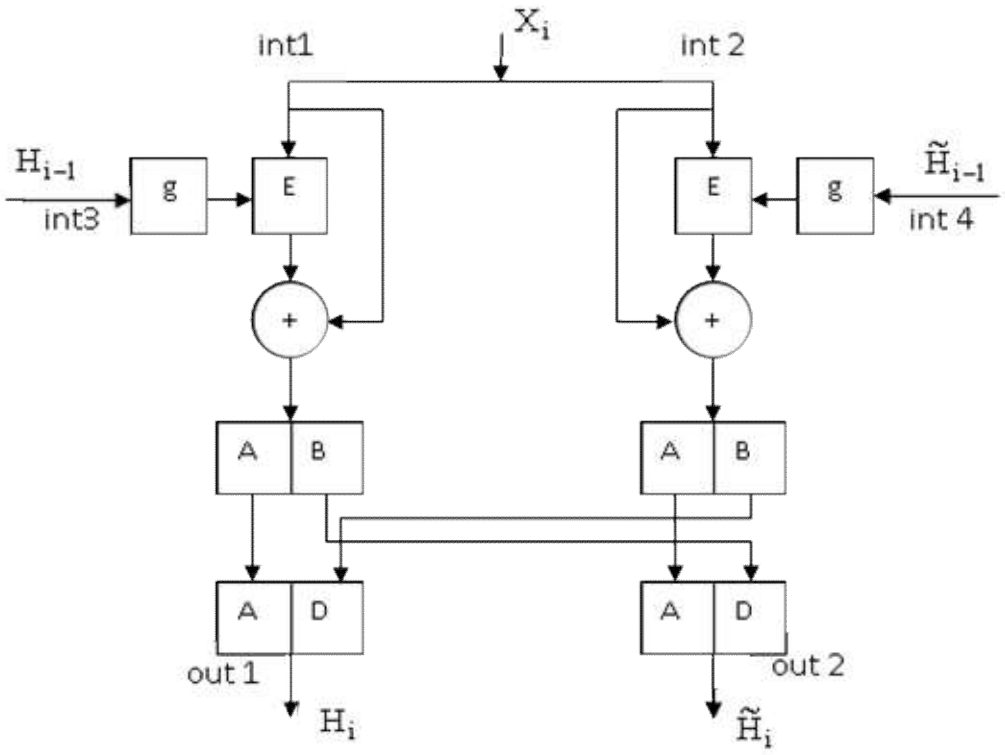
- DES được dùng làm mật mã khối E_k có đầu vào/ ra 64 bit và được tham số hoá bằng khoá k 56 bit.
- Hai hàm g và $\tilde{g}$ ánh xạ các giá trị 64 bit U thành các khoá DES 56 bit như sau:

Cho $U = u_1\ u_2\ \ldots\ u_{64}$, xoá mọi bit thứ 8 bắt đầu từ u_8 và đặt các bit thứ 2 và thứ 3 về "10" đối với g và "01" đối với $\tilde{g}$.

$$g(U) = u_1 10 u_4\ u_5\ u_6\ u_7\ u_9\ u_{10} \ldots u_{63}$$
$$\tilde{g}(U) = u_1 01 u_4\ u_5\ u_6\ u_7\ u_9\ u_{10} \ldots u_{63}$$

Đồng thời điều này cũng phải đảm bảo rằng chúng không phải là các khoá DES yếu hoặc nửa yếu vì các khoá loại này có bit thứ hai bằng bit thứ ba. Đồng thời điều này cũng đảm bảo yêu cầu bảo mật là $g(IV) \neq \tilde{g}(IV)$.

Thuật toán MDC -2 có thể được mô tả theo sơ đồ sau:



Hình 4-3. Thuật toán MDC – 2

Thuật toán MDC - 2

VÀO: Xâu bit x có độ dài r = 64t với $t \geq 2$.

RA : Mã băm 128 bit của x

(1) Phân x thành các khối 64 bit x_i: $x_1\ x_2 \ldots x_t$.

(2) Chọn các hằng số không bí mật IV và $I\tilde{V}$ từ một tập các giá trị khuyến nghị đã được mô tả trước. Tập ngầm định các giá trị cho trước này là (ở dạng HEXA)

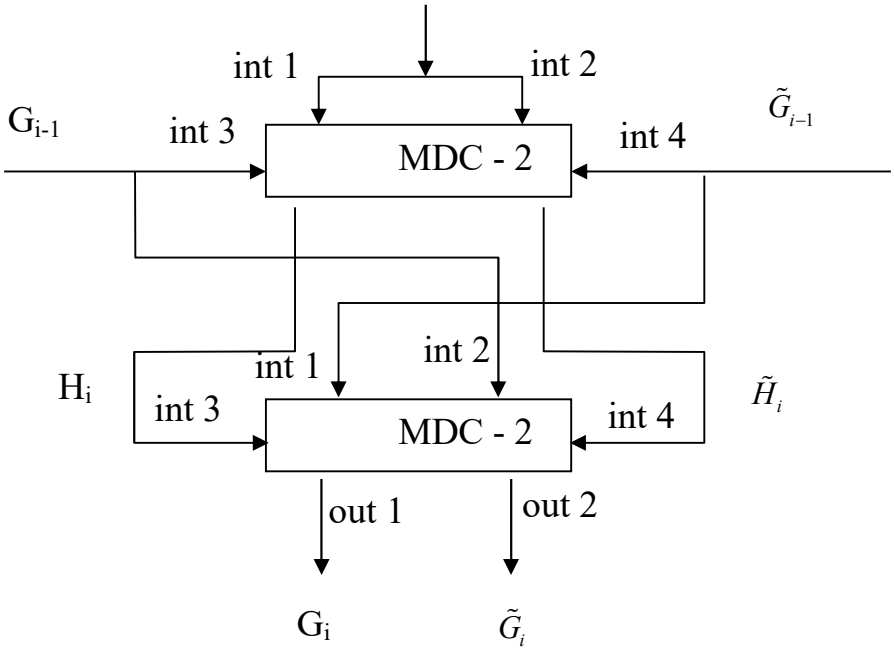
$$IV = 0x5252525252525252$$
$$I\tilde{V} = 0x2525252525252525$$

(3) Ký hiệu $||$ là phép ghép và C_i^L, C_i^R là các nửa 32 bit phải và trái của C_i

Đầu ra $h(x) = H_t \| \tilde{H}_t$ được xác định như sau: (với $1 \leq i \leq t$)

$$H_0 = IV\ ,\ \ k_i = g(H_{i-1})\,,\ C_i = E_{k_i}(x_i) \oplus x_i\ \ ,\ \ H_i = C_i^L \,\|\, \widetilde{C}_i^R$$
$$\widetilde{H}_0 = I\widetilde{V}\ ,\ \ \widetilde{k}_i = \widetilde{g}(\widetilde{H}_{i-1})\,,\ \widetilde{C}_i = E_{\widetilde{k}_i}(x_i) \oplus x_i\ \ ,\ \ \widetilde{H}_i = \widetilde{C}_i^L \,\|\, C_i^R$$

Thuật toán MDC - 4 có thể được mô tả theo sơ đồ sau:



Hình 4-4. Thuật toán MDC – 4

4.2.3. Các hàm băm có khóa (MAC)

Các hàm băm có khoá được sử dụng để xác thực thông báo và thường được gọi là *các thuật toán tạo mã xác thực thông báo (MAC).*

MAC dựa trên các mật mã khối.

Thuật toán

VÀO: Dữ liệu x, mật mã khối E, khoá MAC bí mật k của E.

RA : n bit MAC trên x (n là độ dài khối của E)

(1) Độn và chia khối: Độn thêm các bit vào x nếu cần. Chia dữ liệu đã độn thành từng khối n bit : $x_1\ x_2\ \ldots\ x_t$.

(2) Xử lý theo chế độ CBC.

Ký hiệu E_k là phép mã hoá E với khoá k.

Tính khối H_t như sau:

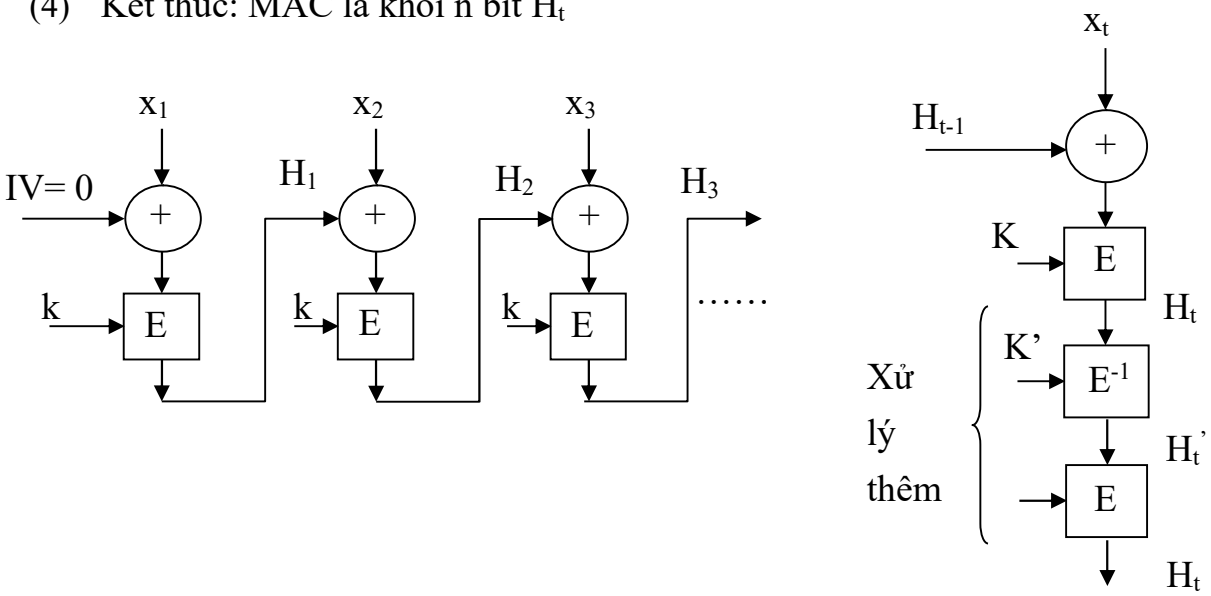
$$H_1 \leftarrow E_k(x_1)$$
$$H_i \leftarrow K_k(H_{i-1} \oplus x_i) \qquad 2 \le i \le t$$

(3) Xử lý thêm để tăng sức mạnh của MAC

Dùng một khoá bí mật thứ hai $k' \neq k$. Tính

$$H_t' \leftarrow E_{k'}^{-1}(H_t) \ , \ H_t = E_k(H_t')$$

(4) Kết thúc: MAC là khối n bit H_t



Hình 4-5. Thuật toán MAC dùng CBC

4.3. CHỮ KÍ SỐ

4.3.1. Sơ đồ Shamir

Chuỗi bít thông báo trước hết được tách thành các véctơ k bít M.

Giả sử $M \in [0, n-1]$

$$M = (m_1, \ldots, m_i, \ldots, m_k)$$

Một ma trận nhị phân bí mật $k \times 2k$ (ma trận H) được chọn ngẫu nhiên cùng với một giá trị modulus n, trong đó n là một số nguyên tố ngẫu nhiên k bít (thông

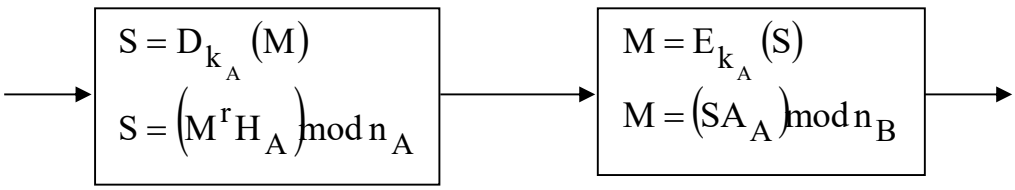
thường k = 100bít). Một vectơ A 2Kbit (được dùng làm khóa công khai) được chọn trên cơ sở giải hệ phương trình tuyến tính sau:

$$\begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,2k-1} & h_{1,2k} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,2k-1} & h_{2,2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{k,1} & h_{k,2} & \cdots & h_{k,2k-1} & h_{k,2k} \end{pmatrix} \times \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{2k} \end{pmatrix} \bmod n = \begin{pmatrix} 2^0 \\ 2^1 \\ \vdots \\ 2^{k-1} \end{pmatrix}$$

Nói một cách khác, các hệ số $\{h_{ij}\}$ được chọn là ngẫu nhiên sao cho thỏa mãn hệ phương trình tuyến tính sau:

$$\left(\sum_{j=1}^{2k} h_{ij} a_j \right) \bmod n = 2^{i-1} \bmod n \qquad \text{với } 1 \le i \le k$$

Đây là hệ k phương trình tuyến tính mođulo với 2k ẩn. Bởi vậy k giá trị đầu của véctơ A được xác định theo các phương trình trên. Véctơ A cùng với n (tức là cặp (A, n)) là các thông tin công khai, trong khi đó ma trận H được giữ kín.



Hình 4-6. Xác thực thông báo dùng sơ đồ chữ kí

4.3.1.1.Xác thực thông báo dùng sơ đồ Shamir

Người gửi A có thể chứng tỏ cho một người dùng khác trên mạng B tính xác thực của thông báo M bằng cách dùng khóa riêng của mình (H_A, n_A) đối với thông báo M.

$$S = D_{k_A}(M)$$

$$S = M^r \times H_A \bmod n_A$$

Trong đó M^r biểu thị véctơ đảo bít của M, tức là:

$$M^r = (m_k, m_{k-1}, \ldots, m_2, m_1)$$

Các bít của thông báo đã ký là:

$$s_i = \sum_{i=1}^{k} m_i h_{ij} \qquad \text{với } 1 \le j \le 2k$$

$$s_i \in [0, k]$$

Chỉ có A có thể tạo ra 2Kbít $\{s_i\}$ từ k bít của thông báo $\{m_i\}$ vì chỉ có A mới tạo được $2.k^2$ phần tử của ma trận $\{h_{i,j}\}$

4.3.1.2.Kiểm tra thông báo

Mỗi người dùng trên mạng có thể kiểm tra tính xác thực của thông báo do A gửi bằng cách dùng thông tin công khai (A_A, n_A):

$$E_{k_A}(S) = S \times A_A \bmod n_A$$

$$E_{k_A}(S) = (M^r \times H_A) \times A_A \bmod n_A$$

$$E_{k_A}(S) = M$$

Tức là :

$$\sum_{j=1}^{2k} s_j a_j \bmod n_A = \sum_{j=1}^{2k}\left[\sum_{i=1}^{k} m_i h_{ij}\right] a_j \bmod n_A$$

$$\sum_{j=1}^{2k} s_j a_j \bmod n_A = \sum_{i=1}^{k} m_i \left[\sum_{j=1}^{2k} h_{ij} a_j\right] \bmod n_A$$

$$= \sum_{i=1}^{k} m_i 2^{i-1} \bmod n_A$$

Ví dụ: Cho k = 3, n = 7

Khi đó thông báo $M \in [0,6]$, mỗi bít của thông báo $m_i \in [0,1]$

Ma trận H được chọn trước như sau:

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

Chẳng hạn ta chọn được k phần tử đầu tiên của véctơ A là: $a_1 = 1$, $a_2 = 3$, $a_3 = 4$. Khi đó k phần tử còn lại của A được xác định bằng cách giải:

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \\ 4 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} \bmod 7 = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \bmod 7$$

Kết quả ta có: $a_4 = 4$, $a_5 = 1$, $a_5 = 2$.

Khi đó véctơ khóa công khai A là: $A = (1,3,4,4,1,2)$.

Để xác thực thông báo M = 3 (tức là M = (0, 1, 1)) người gửi A dùng khóa riêng của mình là ma trận H và tính:

$$S = M^r \times H$$

$$S = (1,1,0) \times \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$S = (1,1,\ 2,\ 1,\ 1,\ 0)$$

Ở phía thu, người thu sẽ tạo lại thông báo dựa trên thông tin về khóa công khai của A và n.

$$M = S \times A = (1,1,\ 2,\ 1,\ 1,\ 0) \times \begin{pmatrix} 1 \\ 3 \\ 4 \\ 4 \\ 1 \\ 2 \end{pmatrix} \bmod 7$$

$$M = 17 \bmod 7 = 3$$

Như vậy thông báo M đã được xác thực vì chỉ có người gửi A mới có thể tạo ra một thông báo có nghĩa.

Sơ đồ chữ ký số Shamir được mô tả ở trên là không an toàn vì với một cặp bản rõ – bản mã thích hợp thám mã mới có thể xác định được ma trận H. Bằng cách ngẫu nhiên hóa thông báo M trước khi ký ta có thể tránh được nguy cơ này:

Véctơ A sẽ được nhân với một vectơ ngẫu nhiên R có 2Kbít: $R = (r_1, \ldots, r_{2k})$ rồi thực hiện phép biến đổi sau:

$$M' = (M - R \times A) \bmod n$$

Hay $M = (M' + R \times A) \bmod n$

Để ký cho thông báo đã biến đổi M’ ta cũng đảo ngược các bít và nhân nó với H. Tuy nhiên kết quả này lại được cộng với véctơ R.

$$S' = M'^{r} \times H + R$$

$$S' = \left(m'_k, \ldots, m'_1\right) \begin{pmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,2k-1} & h_{1,2k} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,2k-1} & h_{2,2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{k,1} & h_{k,2} & \ldots & h_{k,2k-1} & h_{k,2k} \end{pmatrix} + \left(r_1, \ldots, r_{2k}\right)$$

$$S' = \left(s_1, \ldots, s_{2k}\right) + \left(r_1, \ldots, r_{2k}\right)$$

$$S' = \left(s'_1, \ldots, s'_{2k}\right)$$

Ở điểm thu, người sử dụng kiểm tra tính xác thực của thông báo S' bằng cách véctơ khóa công khai A:

$$\begin{aligned} S' \times A \bmod n &= \left(M'^{r} \times H + R\right) \times A \bmod n \\ &= \left(M'^{r} \times H \times A + R \times A\right) \bmod n \\ &= \left(M' + R \times A\right) \bmod n \\ &= \left(M - R \times A + R \times A\right) \bmod n \\ &= M \end{aligned}$$

Cần chú ý rằng, vào năm 1984 Odlyzko đã phá được sơ đồ chữ ký này.

Ví dụ: Trở lại ví dụ trước với k = 3, n = 7.

Ma trận khóa công khai H có dạng:

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

Véctơ khóa công khai: $A = (1,3,4,4,1,2)$

Giả sử ra chọn ngẫu nhiên vectơ R 2Kbít như sau:

$$R = (1,1,0,0,0,1)$$

Khi đó thông báo M' là:

$$M' = M - (R \times A) = 3 - (1,1,0,0,0,1) \times \begin{pmatrix} 1 \\ 3 \\ 4 \\ 4 \\ 1 \\ 2 \end{pmatrix} \bmod 7$$

$$M' = 3 - 6 \bmod 7 = -3 \bmod 7 = 4$$

Thông báo đã ngẫu nhiên hóa $M' = 4 = (1, 0, 0)$

Chữ ký xác thực S' được tính như sau:

$$S' \times = M^{'r} \times H + R$$

$$S' = (0, 0, 1) \times \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} + (1,1,0,0,0,1)$$

$$S' = (1,0,0,0,1) + (1,1,0,0,0,1)$$

$$S' = (2,1,0,0,1,2)$$

Dựa trên S' nhận được, bên thu sẽ kiểm tra bằng cách sử dụng véctơ khóa công khai A:

$$M = S' \times A = (2,1,0,0,1,2) \times \begin{pmatrix} 1 \\ 3 \\ 4 \\ 4 \\ 1 \\ 2 \end{pmatrix} \bmod 7 = 10 \bmod 7 = 3$$

4.3.2. Sơ đồ Ong – Schnorr – Shamir

Sơ đồ xác thực này đã được Ong,Schnorr và Shamir đưa ra vào 1984. Trong sơ đồ này, người gửi (người sử dụng A) chọn một số nguyên lớn n_A (n_A không nhất thiết phải là một số nguyên tố). Sau đó A chọn một số ngẫu nhiên k_A

nguyên tố cùng nhau với n_A (tức là UCLN $(k_A, n_A) = 1$). Khóa công khai k_A được tính như sau:

$$K_A = -(k_A)^{-2} \bmod n_A$$

Cặp (k_A, n_A) được đưa công khai cho mọi người dùng trong mạng. Để xác thực một thông báo M (M nguyên tố cùng nhau với n_A), người gửi sẽ chọn một số ngẫu nhiên R_A (R_A cũng nguyên tố cùng nhau với n_A) rồi tính thông báo được xác thực là cặp $S = (S_1, S_2)$ sau:

$$S_1 = 2^{-1}\left[\left(MR_A^{-1}\right) + R_A\right] \bmod n_A$$
$$S_2 = 2^{-1} k_A \left[\left(MR_A^{-1}\right) - R_A\right] \bmod n_A$$

Sau đó A gửi S cho bên thu qua mạng.

Việc kiểm tra tính xác thực ở bên thu được thực hiện như sau:

$$S_1^2 + \left(K_A S_2^2\right) \bmod n_A = M$$

Thực vậy ta có:

$$\begin{aligned}
S_1^2 + (K_A S_2^2) \bmod n_A &= \left[2^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]\right]^2 + K_A\left[2^{-1} k_A \left[\left(MR_A^{-1}\right) - R_A\right]\right]^2 \bmod n_A \\
&= 4^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]^2 + 4^{-1} K_A k_A^2 \left[\left(MR_A^{-1}\right) - R_A\right]^2 \bmod n_A \\
&= 4^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]^2 - 4^{-1} k_A^2 k_A^{-2} \left[\left(MR_A^{-1}\right) - R_A\right]^2 \bmod n_A \\
&= 4^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]^2 - 4^{-1}\left[\left(MR_A^{-1}\right) - R_A\right]^2 \bmod n_A \\
&= 4^{-1}\left[M^2 R_A^{-2} + 2MR_A^{-1}R_A + R_A^{-2}\right] - \left[M^2 R_A^{-2} - 2MR_A^{-1}R_A + R_A^{-2}\right] \bmod n_A \\
&= 4^{-1}\left(M^2 R_A^{-2} + 2M + R_A^{-2} - M^2 R_A^{-2} + 2M - R_A^{-2}\right) \bmod n_A \\
&= 4^{-1}(2M + 2M) \bmod n_A \\
&= M
\end{aligned}$$

<u>Ví dụ:</u> Giả sử người gửi A chọn $n_A = 27$ và $k_A = 5$

(ta có .(27, 5) = 1). A tính K_A như sau:

$$K_A = -(k_A)^2 \bmod n_A = -(5)^{-2} \bmod 27$$
$$= -\left(5^{-1}\right)^2 \bmod 27 = -(11)^2 \bmod 27$$
$$= -121 \bmod 27 = 14$$

Khi đó thông tin khóa công khai là $(K_A, n_A) = (14,27)$.

Sau khi A chọn một cặp số ngẫu nhiên R_A với điều kiện $(R_A, n_A) = 1$ rồi tính cặp chữ ký $S = (S_1, S_2)$ từ R_A và thông báo M (với điều kiện $(M, n_A) = 1$). Chẳng hạn $R_A = 13$ và $M = 25$.

$$S_1 = 2^{-1}\left[\left(MR_A^{-1}\right) + R_A\right] \bmod n_A$$
$$= 14[(25.25) + 13] \bmod 27$$
$$= 14.638 \bmod 27 = 8932 \bmod 27 = 22$$
$$S_2 = 2^{-1} k_A \left[\left(MR_A^{-1}\right) - R_A\right] \bmod n_A$$
$$= 14.5[(25.25) - 13] \bmod 27$$
$$= 70.612 \bmod 27 = 42840 \bmod 27 = 18$$

(Ta có $2^{-1} \bmod 27 = 14$ và $13^{-1} \bmod 27 = 25$).

Sau đó cặp $S = (S_1, S_2) = (22, 18)$ sẽ được gửi qua mạng tới người nhận B.

B sẽ kiểm tra tính xác thực của thông báo bằng khóa công khai của A là cặp $(K_A, n_A) = (14,27)$. B tính :

$$S_1^2 + \left(K_A S_2^2\right) \bmod n_A = 22^2 + \left(14.18^2\right) \bmod 27$$
$$= 484 + 14.324 \bmod 27$$
$$= 5020 \bmod 27$$
$$= 25 = M$$

4.4. CÁC CHỮ KÍ SỐ CÓ NÉN

Trong thực tế, các bản tin có thể là một vài trang văn bản hoặc là các file dữ liệu lớn. Trong phần trên ta thấy rằng các chữ ký cho thông báo cũng có độ lớn

như bản thân các bản tin. Trong phần này ta sẽ mô tả một số sơ đồ chữ ký số mà độ lớn của nó thường là nhỏ hơn và không phụ thuộc vào độ lớn của bản tin. Đó là các chữ ký số có nén.

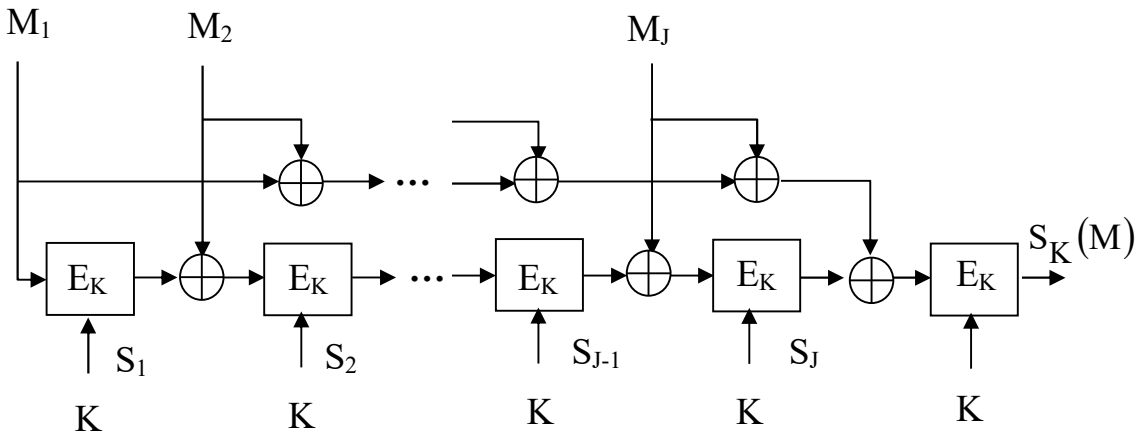
4.4.1. Nén chữ kí

Hình 4.7. chỉ ra một phương pháp nén chữ kí

$$S_1 = E_K(M_1)$$
$$S_2 = E_K(M_2 \oplus S_1)$$
$$\vdots$$
$$S_J = E_K(M_J \oplus S_{J-1})$$

Theo cách này ta tạo được một chữ ký $S_k(M)$

$$S_K(M) = E_K(M_1 \oplus M_2 \oplus \ldots \oplus M_J \oplus S_J)$$



Hình 4-7. Vòng nén chữ kí

4.4.2. Sơ đồ Diffie – Lamport

Trong sơ đồ này một chữ ký số cho n bít bản tin được tạo như sau:

(1). Chọn n cặp khóa ngẫu nhiên (chẳng hạn như khóa 56 bít của DES) được gửi bí mật:

$$\begin{aligned} i=1 &\Rightarrow \quad (K_{1,0}, K_{1,1}) \\ i=2 &\Rightarrow \quad (K_{2,0}, K_{2,1}) \\ &\vdots \\ i=n &\Rightarrow \quad (K_{n,0}, K_{n,1}) \end{aligned}$$

(2). Chọn một dãy S gồm n cặp véctơ ngẫu nhiên (chẳng hạn như các khối đầu vào 64 bít của DES), dãy này được đưa ra công khai:

$$S = \{(S_{1,0}, S_{1,1}), (S_{2,0}, S_{2,1}), \ldots, (S_{n,0}, S_{n,1})\}$$

(3). Tính R là dãy các khóa mã (chẳng hạn là các dãy ra của DES)

$$R = \{(R_{1,0}, R_{1,1}), (R_{2,0}, R_{2,1}), \ldots, (R_{n,0}, R_{n,1})\}$$

Trong đó : $R_{ij} = E_{K_{i,j}}(S_{i,j})$ với $1 \le i \le n$ và $j = 0,1$

Dãy R cũng được đưa công khai.

Chữ ký SG(M) của một bản tin n bít $M = (m_1, m_2, \ldots, m_n)$ chính là dãy khóa sau: $M = \left(K_{1,i_1}, K_{2,i_2}, \ldots, K_{n,i_n}\right)$ trong đó chỉ số khóa $i_j = m_j$.

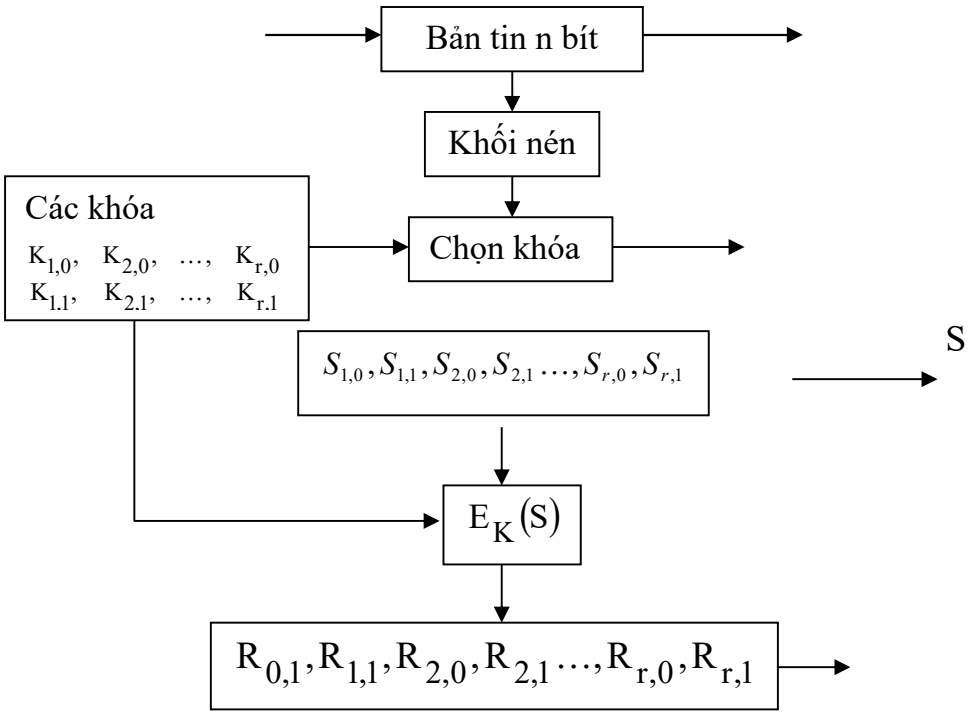
Ví dụ : Nếu thông báo M là :

$$\begin{aligned} M &= m_1 \quad m_2 \quad m_3 \quad m_4 \quad \ldots \quad m_{n-1} \quad m_n \\ M &= 1 \quad 0 \quad 0 \quad 1 \quad \ldots \quad 1 \quad 1 \end{aligned}$$

Thì chữ ký SG(M) là:

$$\begin{aligned} SG(M) &= K_{1,i_1} \quad K_{2,i_2} \quad K_{3,i_3} \quad K_{4,i_4} \quad \ldots \quad K_{n-1,i_{n-1}} \quad K_{n,i_n} \\ SG(M) &= K_{1,1} \quad K_{2,0} \quad K_{3,0} \quad K_{4,1} \quad \ldots \quad K_{n-1,1} \quad K_{n,1} \end{aligned}$$

Sơ đồ chữ ký Diffie-Lamport được mô tả trên hình sau:



Hình 4-8. Sơ đồ chữ kí D – L (đầu phát)

Bản tin M và chữ ký SG(M) đều được gửi tới nơi thu.

Bản tin có thể kiểm tra tính xác thực của thông báo bằng việc mã hóa các véctơ tương ứng của dãy S đã biết với chữ ký SG(M) đã nhận và so sánh bản mã tạo ra với dãy R đã biết.

$$E_{K_{1,i1}}\left(S_{1,i_1}\right) \stackrel{?}{=} R_{1,i_1}$$

$$E_{K_{2,i_2}}\left(S_{2,i_2}\right) \stackrel{?}{=} R_{2,i_2}$$

$$\vdots$$

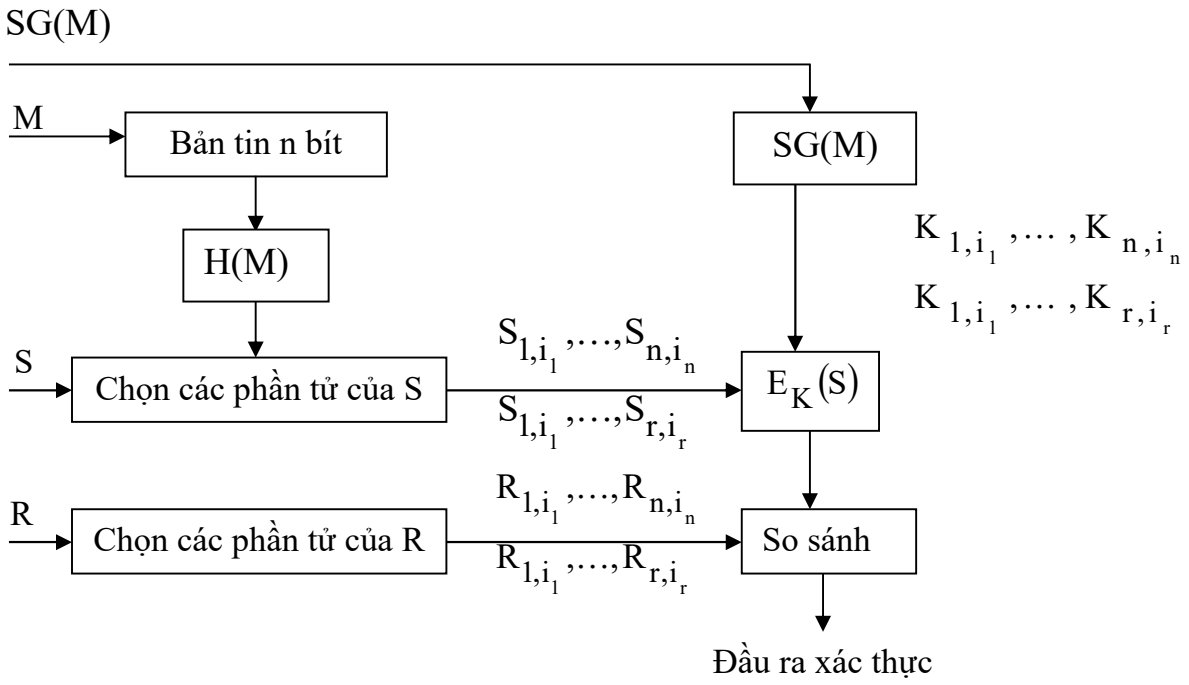
$$E_{K_{n,i_n}}\left(S_{n,i_n}\right) \stackrel{?}{=} R_{n,i_n}$$

Nếu dãy n véctơ này bằng nhau thì chữ ký được xem là đã xác thực.

$$\left(R_{1,i_1}, R_{2,i_2}, \ldots, R_{n,i_n}\right) = \left[E_{K_{1,i_1}}\left(S_{1,i_1}\right), \ldots, E_{K_{n,i_n}}\left(S_{n,i_n}\right)\right]$$

Cần chú ý rằng sơ đồ chữ ký D-L sẽ mở rộng độ dài chữ ký chứ không phải là nén nó ! Nếu DES được sử dụng thì một bản tin n bít sẽ cần một chữ ký số SG(M) có độ dài là 56.n bít. Vì vậy, để khắc phục nhược điểm này bản tin n cần được nén thành một bản tóm lược thông báo r bít $(r << n)$ bằng một hàm băm H(M) trước khi áp dụng sơ đồ D-L.

Hình 4.9 chỉ ra quá trình kiểm tra chữ ký.



Hình 4-9. Kiểm tra chữ kí D – L (đầu thu)

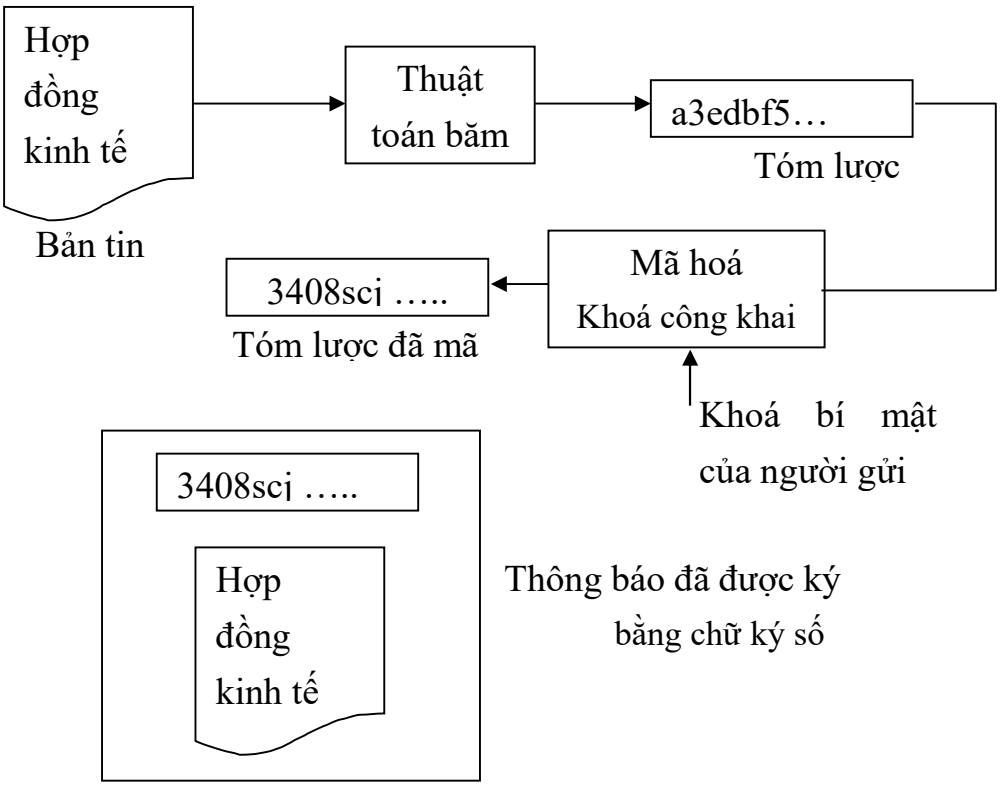
Cần chú ý rằng chữ ký ở đây chỉ còn là tập r khóa.

Một hạn chế khác cần phải nói tới là : vì một nửa số khóa đã bị lộ sau khi kiểm tra nên sơ đồ này chỉ có thể được sử dụng một lần với một cặp khóa cho trước. Để khắc phục nhược điểm này ta có thể sử dụng sơ đồ chữ ký dựa trên các hệ mật khóa công khai

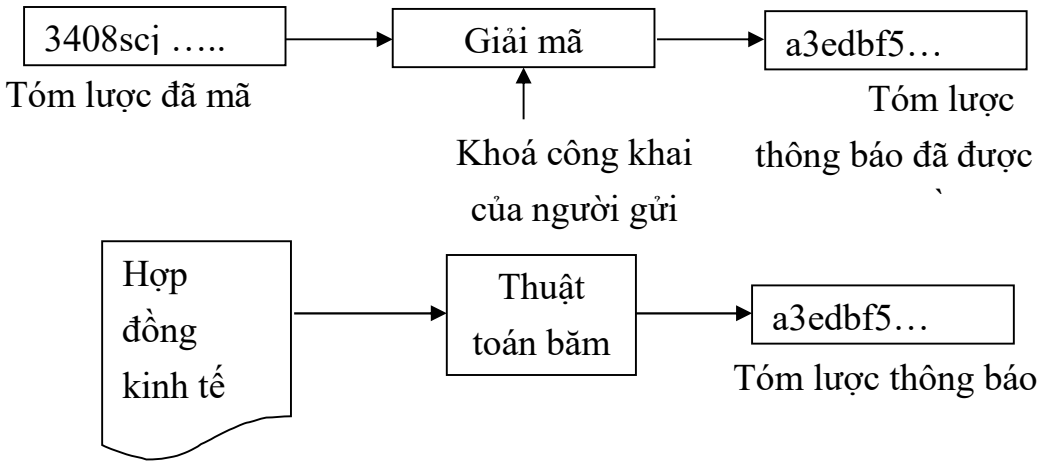
4.4.3. Sơ đồ chữ kí RSA

Chữ ký số được xây dựng trên cơ sở kết hợp mã hoá khoá công khai với hàm băm. Tuy nhiên cách sử dụng khóa ở đây khác với trong các hệ mật khóa công khai.

Các bước tạo chữ ký và kiểm tra chữ ký được mô tả trên hình sau:



Hình 4-10. Tạo một thông báo có kí bằng chữ



Hình 4-11. Các bước kiểm tra một thông báo đã kí

Ví dụ: Sơ đồ chữ ký số RSA.

Có thể coi bài toán xác thực là bài toán "đối ngẫu" với bài toán bảo mật. Vì vậy, sử dụng ngược thuật toán RSA ta có thể có được một sơ đồ chữ ký số RSA như sau:

Giả sử $n = pq$, trong đó p và q là các số nguyên tố lớn có kích thước tương đương.

$$K = \{(n, e, d) : d \in Z_n^*, ed \equiv 1 \bmod(n)\}$$

Với $K = (n, e, d)$ ta có $D = d$ là khoá bí mật, $E = (n, e)$ là khoá công khai, m là bản tin cần ký.

Tạo chữ ký : $S = sig_D(m) = m^d \bmod n$

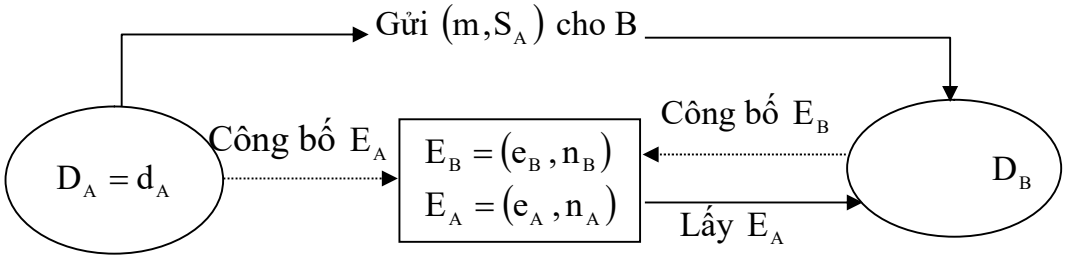
Kiểm tra chữ ký: $ver_E(m, s)$= đúng $\Leftrightarrow m \equiv S^e \bmod n$.

Hoạt động của sơ đồ chữ ký RSA có thể mô tả như sau:

a. Trường hợp bản tin rõ m không cần bí mật (Hình 4.13.).

A ký bản tin m và gửi cho B.

B kiểm tra chữ ký của A.



Ký: $S_A = m^{d_A} \bmod n_A$ Kiểm tra: $m = S_A^{e_A} \bmod n_A$

Hình 4-12. Sơ đồ kí số RSA (không bí mật bản tin)

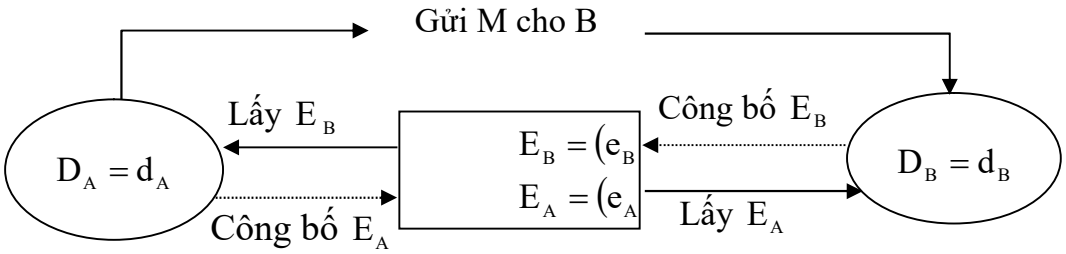
Giả sử A muốn gửi cho B bản tin rõ m có xác thực bằng chữ ký số của mình. Trước tiên A tính chữ ký số

$$S_A = sig_{D_A}(m) = m^{d_A} \bmod n_A$$

Sau đó A gửi cho B bộ đôi (m, S_A). B nhận được (m, S_A) và kiểm tra xem điều kiện $m \equiv S_A^{e_A} \bmod n_A$ có thoả mãn không. Nếu thoả mãn, thì khi đó B khẳng định rằng $ver_{E_A}(m, S_A)$ nhận giá trị Đúng và chấp nhận chữ ký của A trên m.

b. Trường hợp bản tin rõ m cần giữ bí mật (Hình 4.14).

A ký bản tin rõ m để được chữ ký S_A. Sau đó A dùng khoá mã công khai E_B của B để lập bản mã $M = E_B(m, S_A)$ rồi gửi đến B. Khi nhận được bản mã M, B dùng khoá bí mật D_B của mình để giải mã cho M và thu được m, S_A. Tiếp đó dùng thuật toán kiểm tra ver_{E_A} để xác nhận chữ ký của A.



A tính:

i. $S_A = m^{d_A} \bmod n_A$

ii. $X = m \| S_a$

iii. $M = X^{e_B} \bmod n_B$

B tính:

i. $X = M^{d_B} \bmod n_B$

ii. $X = m \| S_a$

iii. Kiểm tra: $m = S_A^{e_A} \bmod n_A$

Hình 4-13. Sơ đồ chữ kí số RSA (có bí mật bản tin)

4.5. CHUẨN CHỮ KÍ SỐ

Chuẩn chữ kí số(DSS) là phiên bản cải tiến của sơ đồ chữ kí Elgamal. Nó được công bố trong Hồ Sơ trong liên bang vào ngày 19/5/94 và được làm chuẩn voà 1/12/94 tuy đã được đề xuất từ 8/91. Trước hết ta sẽ nêu ra những thay đổi của nó so với sơ đồ Elgamal và sau đó sẽ mô tả cách thực hiện nó.

Trong nhiều tinh huống, thông báo có thể mã và giải mã chỉ một lần nên nó phù hợp cho việc dùng với hệ mật Bất kì (an toàn tại thời điểm được mã). Song trên thực tế, nhiều khi một bức điện được dùng làm một tài liệu đối chứng, chẳng hạn như bản hợp đồng hay một chúc thư và vì thế cần xác minh chữ kí sau nhiều năm kể từ lúc bức điện được kí. Bởi vậy, điều quan trọng là có phương án

dự phòng liên quan đến sự an toàn của sơ đồ chữ kí khi đối mặt với hệ thống mã. Vì sơ đồ Elgamal không an toàn hơn bài toán logarithm rời rạc nên cần dung modulo p lớn. Chắc chắn p cần ít nhất là 512 bít và nhiều người nhất trí là p nên lấy p=1024 bít để có độ an toàn tốt.

Tuy nhiên, khi chỉ lấy modulo p =512 thì chữ kí sẽ có 1024 bít. Đối với nhiều ứng dụng dùng thẻ thông minh thì cần lại có chữ kí ngắn hơn. DSS cải tiến sơ đồ Elgamal theo hướng sao cho một bức điện 160 bít được kí bằng chữ kí 302 bít song lại p = 512 bít. Khi đó hệ thống làm việc trong nhóm con Z_n^* kích thước 2^{160}. Độ mật của hệ thống dựa trên sự an toàn của việc tìm các logarithm rời rạc trong nhóm con Z_n^*.

Sự thay đổi đầu tiên là thay dấu " - " bằng "+" trong định nghĩa δ, vì thế:

$$\delta = (x + \alpha\ \gamma)k\text{-}1 \bmod (p\text{-}1)$$

thay đổi kéo theo thay đổi điều kiện xác minh như sau:

$$\alpha^x\ \beta^\gamma \equiv \gamma^\delta \pmod p \quad (1)$$

Nếu UCLN $(x + \alpha\gamma, p\text{-}1) = 1$ thì $\delta^{-1} \bmod (p\text{-}1)$ tồn tại và ta có thể thay đổi điều kiện (6.1) như sau:

$$\alpha x\delta^{-1}\beta\gamma\delta^{-1} \equiv \gamma \pmod p \quad (2)$$

Đây là thay đổi chủ yếu trong DSS. Giả sử q là số nguyên tố 160 bít sao cho q | (q-1) và α là căn bậc q của một modulo p. (Dễ dàng xây dựng một α như vậy: cho α_0 là phần tử nguyên thuỷ của Z_p và định nghĩa $\alpha = \alpha_0^{(p-1)/q} \bmod p$).

Khi đó β và γ cũng sẽ là căn bậc q của 1. vì thế các số mũ Bất kỳ của α, β và γ có thể rút gọn theo modulo q mà không ảnh hưởng đến điều kiện xác minh (6.2). Điều rắc rối ở đây là γ xuất hiện dưới dạng số mũ ở vế trái của (6.2) song không như vậy ở vế phải. Vì thế, nếu γ rút gọn theo modulo q thì cũng phải rút

gọn toàn bộ vế trái của (6.2) theo modulo q để thực hiện phép kiểm tra. Nhận xét rằng, sơ đồ (6.1) sẽ không làm việc nếu thực hiện rút gọn theo modulo q trên (6.1). DSS được mô tả đầy đủ trong hinh 6.3.

Chú ý cần có $\delta \not\equiv 0 \pmod q$ vì giá trị $\delta^{-1} \bmod q$ cần thiết để xác minh chữ kí (điều này tương với yêu cầu UCLN(δ, p-1) =1 khi biến đổi (1) thành (2). Nếu Bob tính $\delta \equiv 0 \pmod q$ theo thuật toán chữ kí, anh ta sẽ loại đi và xây dựng chữ kí mới với số ngẫu nhiên k mới. Cần chỉ ra rằng, điều này có thể không gần vấn đề trên thực tế: xác xuất để $\delta \equiv 0 \pmod q$ chắc sẽ xảy ra cở 2^{-160} nên nó sẽ hầu như không bao giờ xảy ra.

Dưới đây là một ví dụ minh hoạ nhỏ

Hình 4.14. Chuẩn chữ kí số.

Ví dụ:

Giả sử q =101, p = 78 q+1 =7879.3 là phần tử nguyên thuỷ trong Z_{7879} nên ta có thể lấy: $\alpha = 3^{78} \bmod 7879 = 170$

Giả sử a =75, khi đó :

$$\beta = \alpha^a \bmod 7879 = 4576$$

Bây giờ giả sữ Bob muốn kí bức điện x = 1234 và anh ta chọn số ngẫu nhiên k =50, vì thế :

$$k^{-1} \bmod 101 = 99$$

khi đó $\gamma = (170^{30} \bmod 7879) \bmod 101$

$= 2518 \bmod 101$

$= 94$

và $\delta = (1234 + 75 \times 94) \bmod 101 = 96$

Chữ kí (94, 97) trên bức điện 1234 được xác minh bằng các tính toán sau:

$$\delta^{-1} = 97^{-1} \bmod 101 = 25$$

$e_1 = 1234 \times 25 \bmod 101 = 45$

$e_2 = 94 \times 25 \bmod 101 = 27$

$(170^{45}\ 4567^{27} \bmod 7879) \bmod = 2518 \bmod 101 = 94$

vì thế chữ kí hợp lệ.

Giả sử p là số nguyên tố 512 bít sao cho bài toán logarithm rời rạc trong Z_p khong Giải được, cho p là số nguyên tố 160 bít là ước của (p-1). Giả thiết $\alpha \in Z_p$ là căn bậc q của 1modulo p: Cho P =Z_p . A = $Z_q \times Z_p$ và định nghĩa :

$$A = \{(p,q,\alpha,a,\beta) : \beta \equiv \alpha^a \pmod p\}$$

các số p, q, α và β là công khai, có a mật.

Với K = (p,q,α,a,β)và với một số ngẫu nhiên (mật) k ,$1 \le k \le q-1$, ta định nghĩa:

$$sig_k(x,k) = (\gamma,\delta)$$

trong đó $\gamma = (\alpha^k \bmod p) \bmod q$

và $\delta = (x + a\gamma)k^{-1} \bmod q$

Với $x \in Z_p$ và $\gamma, \delta \in Z_q$, qua trình xác minh sẽ hoàn toàn sau các tính toán :

$$e_1 = x\delta^{-1} \bmod q$$

$$e_2 = \gamma\delta^{-1} \bmod q$$

$$ver_k(x, \gamma, \delta) = true \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma$$

Hình 4-14. Chuẩn chữ kí số

Khi DSS được đề xuất năm 1991, đã có một vài chỉ trích đưa ra. Một ý kiến cho rằng, việc xử lý lựa chọn của NIST là không công khai. Tiêu chuẩn đã

được Cục An ninh Quốc gia (NSA) phát triển mà không có sự tham gia của khôi công nghiệp Mỹ. Bất chấp những ưu thế của sơ đồ, nhiều người đã đóng chặt cửa không tiếp nhận.

Còn những chỉ trích về mặt kĩ thuật thì chủ yếu là về kích thước modulo p bị cố định = 512 bít. Nhiều người muốn kích thước này có thể thay đổi được nếu cần, có thể dùng kích cỡ lớn hơn. Đáp ứng những đòi hỏi này, NIST đã chọn tiêu chuẩn cho phép có nhiều cở modulo, nghĩa là cỡ modulo bất kì chia hết cho 64 trong phạm vi từ 512 đến 1024 bít.

Một phàn nàn khác về DSS là chữ kí được tạo ra nhanh hơn việc xác minh nó. Trong khi đó, nếu dùng RSA làm sơ đồ chữ kí với số mũ xác minh công khai nhỏ hơn (chẳng hạn = 3) thì có thể xác minh nhanh hơn nhiều so với việc lập chữ kí. Điều này dẫn đến hai vấn đề liên quan đến những ứng dụng của sơ đồ chữ kí:

1.Bức điện chỉ được kí một lần, song nhiều khi lại cần xác minh chữ kí nhiều lần trong nhiều năm. Điều này lại gợi ý nhu cầu có thuật toán xác minh nhanh hơn.

2.Những kiểu máy tính nào có thể dùng để kí và xác minh ?. Nhiều ứng dụng, chẳng hạn các thẻ thông minh có khả năng xử lý hạn chế lại liên lạc với máy tính mạnh hơn. Vi thế có nhu cầu nhưng thiết kế một sơ đồ để có thực hiện trên thẻ một vài tính toán. Tuy nhiên, có những tình huống cần hệ thống mình tạo chữ kí, trong những tình huống khác lại cần thẻ thông minh xác minh chữ kí. Vì thế có thể đưa ra giải pháp xác định ở đây.

Sự đáp ứng của NIST đối với yêu cầu về số lần tạo xác minh chữ kí thực ra không có vấn đề gì ngoài yêu cầu về tốc độ, miễn là cả hai thể thực hiện đủ nhanh.

4.6. BÀI TẬP

1. Giả sử p =25307 còn $\alpha = 2$ là các tham số công khai dùng cho thủ tục thoả thuận khoá Diffie-Hellman.

Giả sử A chọn x = 3578 và B chọn y = 19956. Hãy tính khoá chung của A và B.

2. Giả sử $n = pq$, p và q là hai số nguyên tố riêng biệt lớn sao cho $p = 2p_1 + 1$ và $q = 2q_1 + 1$, với p_1, q_1 là các số nguyên tố. Giả sử α là phần tử có cấp $2p_1q_1$ trong Z_n^* (Đây là bậc lớn nhất của phần tử bất kỳ trong Z_n^*). Định nghĩa hàm băm $h: \{1,\ldots,n^2\} \to Z_n^*$ theo quy tắc $h(x) = \alpha^x \bmod n$.

Bây giờ giả sử n = 603241 và α = 11 được dùng để xác định hàm băm theo kiểu này và ta có ba va chạm đối với $h: h(1294755) = h(80115359) = h(52738737)$ Dùng thông tin này để phân tích nhân tử n

KẾT LUẬN

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. □

TÀI LIỆU THAM KHẢO

[1] Nguyễn Bình. *Giáo trình mật mã học*. NXB Bưu điện 2004

[2] A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone. *Handbook of applied cryptography*. CRC Press 1998.

[3] B. Schneier. *Applied Cryptography*. John Wiley Press 1996.

[4] D. R. Stinson. *Cryptography. Theory and Practic*e. CRC Press 1995.

[5] Nguyen Binh. *Crypto-system based on Cyclic Goemetric Progresssions over polynomial ring (Part 1). Circulant crypto-system over polynomial ring (Part 2)* 8th *VietNam Conference on Radio and Electronics*, 11-2002

[6] M. R. A. Huth. *Secure Communicating Systems*. Cambridge University Press 2001.

[7] C. Pfleeger. S*ecurity in Computing.* Prentice Hall. 1997.

[8] S. Bellovir, M. Merritt. *Encrypted Key Exchange*.
Proc. IEEE Symp. Security and Privacy
IEEE Comp Soc Press 1992.

[11] D. Denning, D. Branstad. *A Taxonomy of Key Escrow Ecryption Systems*. Comm ACM, v39 n3, Mar 1996.

[12] M. Blum. *Coin flipping by Telephone*. SIGACT News, 1981.

[13] S. Even. *A Randomizing Protocol for Signing Contracts*. Comm ACM, v28 n6, Jun 1985.

[14] R. Merkle, M. Hellman. *On the security of Multiple Encryption*. Comm ACM, v24 n7, July 1981.

[15] W. Tuchman, H*ellman Presents No Shortcut Solutions to the DES.* IEEE Spectrum, v16 n7, Jun 1979.

[16] A.Shamir. *Identity-based cryptorytions and signature schemes*.
Advanced in Cryptology - CRYPTO'84, LNCS196

Springer_Verlag, pp.47-53, 1985

[17] E.Okamoto, K.Tanaka. *Key distribution system based on indentification information*.

IEEE J.Selected Areas in communications, Vol.7,pp.481-485, 1989.

[18] *Secure Communications and Data Encryption*. Course notes

Jean Yves Chouirard. University of Ottawa. April 2002.