GT.0000026899

ĐẠI HỌC THÁI NGUYÊN
CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TRẦN ĐỨC SỰ (Chủ biên) - NGUYỄN VĂN TẢO, TRẦN THỊ LƯỢNG

GIÁO TRÌNH
AN TOÀN BẢO MẬT DỮ LIỆU

NGUYÊN
ỌC LIỆU

NHÀ XUẤT BẢN ĐẠI HỌC THÁI NGUYÊN

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TRẦN ĐỨC SỰ (Chủ biên)
NGUYỄN VĂN TẢO, TRẦN THỊ LƯỢNG

GIÁO TRÌNH
AN TOÀN BẢO MẬT DỮ LIỆU

ĐẠI HỌC THÁI NGUYÊN
TRUNG TÂM HỌC LIỆU

NHÀ XUẤT BẢN ĐẠI HỌC THÁI NGUYÊN
NĂM 2015

MÃ SỐ: 02 - 35
ĐHTN - 2015

Biên mục trên xuất bản phẩm của Trung tâm Học liệu – Đại học Thái Nguyên

Trần, Đức Sự (chủ biên)

Giáo trình an toàn và bảo mật dữ liệu / Trần Đức Sự (chủ biên), Nguyễn Văn Tảo, Trần Thị Lượng. - Thái Nguyên: Đại học Thái Nguyên , 2015. - 236 tr. ; 24 cm.

ISBN: 978-604-915-250-4

1.An toàn thông tin – Giáo trình. 2. An toàn dữ liệu – Giáo trình. 3. Mật mã khoá bí mật – Thuật toán. 4. Mật mã khóa công khai – Thuật toán. I. Nguyễn, Văn Tảo. II. Trần, Thị Lượng.

005.8 – dc14

MỤC LỤC

DANH MỤC TỪ NGỮ VIẾT TẮT 7
DANH MỤC BẢNG 8
DANH MỤC HÌNH VẼ 8
LỜI NÓI ĐẦU 10
Chương 1. **GIỚI THIỆU CHUNG** 12
1.1. Hệ thống thông tin và các hình thức tấn công hệ thống thông tin 12
1.1.1. Thông tin và hệ thống thông tin 12
1.1.2. Ba thuộc tính cơ bản của thông tin 13
1.1.3. Các hình thức tấn công vào hệ thống thông tin 14
1.2. Mật mã và an toàn thông tin 19
1.2.1. Các ứng dụng của mật mã 19
1.2.2. Vai trò của mật mã trong bảo đảm an toàn thông tin 21
1.3. Sơ lược về mật mã học 22
1.3.1. Các khái niệm cơ bản 23
1.3.2. Các kiểu tấn công vào hệ mật mã 25
1.3.3. Phân loại các thuật toán mật mã 26
1.4. Cơ sở toán học của lý thuyết mật mã 28
1.4.1. Kiến thức về độ phức tạp tính toán 28
1.4.2. Kiến thức về lý thuyết số 33
1.5. Bài tập 52
Chương 2. **HỆ MẬT MÃ KHÓA BÍ MẬT** 55
2.1. Giới thiệu 55
2.2. Mật mã cổ điển 57
2.2.1. Mã dịch chuyển 57
2.2.2. Mã thay thế 58

2.2.3. Mã hoán vị 59
2.2.4. Mã Affine 61
2.2.5. Mã Vigenère 66
2.2.6. Hệ mật Hill 68
2.2.7. Hệ mật mã Playfair 73
2.3. Mã dòng 76
2.4. Mã khối 78
2.4.1. Giới thiệu chung 78
2.4.2. Các khái niệm cơ bản 79
2.4.3. Các chế độ hoạt động của mã khối (Modes of operation) 83
2.4.4. Chuẩn mã dữ liệu (DES) 93
2.4.5. Chuẩn mã dữ liệu tiên tiến (AES) 123
2.5. Bài tập 128
Chương 3. **MẬT MÃ KHÓA CÔNG KHAI** 132
3.1. Giới thiệu chung 132
3.2. Hệ mật RSA 135
3.2.1. Thuật toán mã hóa, giải mã RSA 138
3.2.2. Kiểm tra qui tắc giải mã 139
3.2.3. Độ an toàn của hệ RSA 140
3.2.4. Thực hiện RSA 141
3.2.5. Vấn đề điểm bất động trong RSA 141
3.3. Hệ mật Rabin 142
3.3.1. Tạo khóa 142
3.3.2. Mã hóa và giải mã của hệ mật Rabin 143
3.3.3. Ví dụ 143
3.3.4. Đánh giá hiệu quả 144
3.4. Hệ mật Elgamal 144
3.4.1. Bài toán logarit rời rạc 144

3.4.2. Mã hóa, giải mã Elgamal 155
3.4.3. Tham số của hệ mật 156
3.5. Một số hệ mã khóa công khai khác 158
3.5.1. Bài toán xếp ba lô và hệ mật Merkle - Hellman 158
3.5.2. Hệ mật Chor - Rivest (CR) 161
3.5.3. Bài toán mã sửa sai và hệ mật McElice 166
3.5.4. Hệ mật trên đường cong elliptic 172
3.6. Ưu, nhược điểm của hệ mật khóa công khai 181
3.7. Bài tập 181
***Chương 4.* HÀM BĂM VÀ CHỮ KÍ SỐ** 184
4.1. Giới thiệu về hàm băm 184
4.1.1. Khái niệm và phân loại hàm băm 185
4.1.2. Các tính chất cơ bản 187
4.2. Các hàm băm không có khóa 191
4.2.1. MDC độ dài đơn 193
4.2.2. MDC độ dài kép: MDC -2 và MDC - 4 194
4.3. Các hàm băm có khóa (MAC) 196
4.3.1. MAC dựa trên các mật mã khối 197
4.3.2. Xây dựng MAC từ MDC 198
4.4. Chữ kí số 200
4.4.1. Khái niệm chữ ký số 200
4.4.2. Phân loại chữ ký số 202
4.4.3. Xác thực giữa những người sử dụng 206
4.4.4. Kết hợp chữ ký số và mã hoá 206
4.5. Các lược đồ chữ ký số thông dụng 207
4.5.1. Lược đồ RSA 207
4.5.2. Lược đồ Elgamal 208
4.5.3. Lược đồ chữ ký số chuẩn DSS 209

4.5.4. Lược đồ chữ ký số trên EC 210
4.6. Một số lược đồ chữ ký khác 213
4.6.1. Sơ đồ Shamir 213
4.6.2. Sơ đồ Ong – Schnorr – Shamir 219
4.6.3. Các chữ ký số có nén 222
4.7. Ứng dụng của chữ ký số 226
4.7.1. Ứng dụng của chữ ký số 226
4.7.2. Luật về chữ ký số của một số nước trên thế giới 226
4.7.3. Chữ ký số tại Việt Nam 228
4.8. Bài tập 229
TÀI LIỆU THAM KHẢO 234

DANH MỤC TỪ NGỮ VIẾT TẮT

ATTT		An toàn thông tin
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
EC	Elliptic Curve	Đường cong elliptic

DANH MỤC BẢNG

Bảng 1.1. Thuật toán Euclide mở rộng và các giá trị vào a = 4864, b = 3458 38
Bảng 1.2. Cấp của các phần tử trong Z^*_{21} 41
Bảng 1.3. Các lũy thừa của 6 42
Bảng 1.4. Tính 5^{596} mod 1234 44
Bảng 1.5. Độ phức tạp bit của các phép toán cơ bản trong Z_n 45
Bảng 1.6. Các ký hiệu Jacobi của các phần tử trong Z^*_{21} 49
Bảng 2.1. Số các vòng mã hóa của AES 124
Bảng 3.1. Kết quả tính bước 3 của thuật toán Pollard 136
Bảng 3.2. Giải lôgarit rời rạc bằng thuật toán p-pollard. 148
Bảng 3.3. Một số số nguyên tố dạng p=2q+1 157
Bảng 3.4. Giá trị y tương ứng với x trên Z_{23} 174
Bảng 3.5. Bảng tính kP 177

DANH MỤC HÌNH VẼ

Hình 1.1. Mối quan hệ giữa ba tính chất cơ bản của TT 14
Hình 1.2. Sơ đồ tổng quát hệ thống thông tin viễn thông và các hiểm hoạ ATTT đi kèm 15
Hình 1.3. Các hình thức tấn công đối với thông tin trên mạng 16
Hình 1.4. Các tấn công bị động và chủ động 17
Hình 1.5. Sơ đồ khối của một hệ thống thông tin số 22
Hình 1.6. Sơ đồ hệ thống thông tin mật 24
Hình 1.7. Lược đồ các thành phần mật mã cơ bản 27
Hình 2.1. Sơ đồ khối của hệ truyền tin mật 55
Hình 2.2. Mã dịch vòng 57
Hình 2.3. Mã Affine 65
Hình 2.4. Mã Vigenère 66
Hình 2.5. Bảng mã Vigenère 67
Hình 2.6. Mật mã Hill 73

Hình 2.7. Bốn kiểu hoạt động của mã khối 85
Hình 2.8. Một vòng của DES 94
Hình 2.9. Hàm f của DES 96
Hình 2.10. Tính bảng khóa DES 100
Hình 2.11. Chế độ ECB 115
Hình 2.12. Chế độ CBC 116
Hình 2.13. Chế độ CFB 117
Hình 2.14. Chế độ OFB 117
Hình 2.15. DES bội hai 119
Hình 2.16. Mã hóa và giải mã TDES với hai khóa 120
Hình 2.17. Thuật toán mã hóa GDES 122
Hình 3.1. Hệ mật Mc Elice 170
Hình 3.2. Các đường cong $y^2 = x^3 + 2x + 5$ và $y^2 = x^3 - 2x + 1$ 172
Hình 3.3. Nhóm E_{23} (1, 1) 175
Hình 4.1. Phân loại các hàm băm mật mã và ứng dụng 186
Hình 4.2. MDC độ dài đơn 193
Hình 4.3. Thuật toán MDC – 2 195
Hình 4.4. Thuật toán MDC – 4 196
Hình 4.5. Thuật toán MAC dùng CBC 198
Hình 4.6. Lược đồ chữ ký số với phần đính kèm 204
Hình 4.7. Lược đồ chữ ký số khôi phục thông điệp 204
Hình 4.8. Xác thực thông báo dùng sơ đồ chữ kí 214
Hình 4.9. Vòng nén chữ kí 222
Hình 4.10. Sơ đồ chữ kí D – L (đầu phát) 224
Hình 4.11. Kiểm tra chữ kí D – L (đầu thu) 225

LỜI NÓI ĐẦU

Trong thế giới hiện đại, vai trò của máy tính và hệ thống thông tin điện tử ngày càng quan trọng, càng ngày càng có nhiều nhu cầu truyền dẫn, lưu trữ và thậm chí là thực hiện các giao dịch nghiệp vụ trên các hệ thống thông tin điện tử. Trong xã hội bùng nổ thông tin, khi mà thông tin có vai trò và giá trị vượt trội quyết định đến sự thành bại của công tác nghiệp vụ, từ các doanh nghiệp vừa và nhỏ đến các tập đoàn lớn xuyên quốc gia, các cơ quan an ninh, các tổ chức chính trị, xã hội cho đến các trường học, viện nghiên cứu thì vấn đề đảm bảo được an ninh thông tin là một vấn đề được đặt lên hàng đầu. Do vậy, một ứng dụng công nghệ thông tin ngoài việc đáp ứng đầy đủ các yêu cầu nghiệp vụ còn đòi hỏi phải đảm bảo được tính an toàn cho thông tin và dữ liệu trong quá trình xử lý và lưu trữ, tức là phải đảm bảo được các đặc tính:

- Tính bí mật (Confidential)
- Tính xác thực (Authentication)
- Tính toàn vẹn (Intergrity) của thông tin.

Để đảm bảo được các đặc tính này của thông tin, hệ thống thông tin và người quản trị hệ thống cần thực hiện rất nhiều quy tắc và phương pháp khác nhau, từ đảm bảo an toàn vật lý cho đến đảm bảo an toàn người dùng..., và đặc biệt quan trọng nhất là đảm bảo an toàn dữ liệu khi lưu trữ và truyền dẫn. Vấn đề an toàn và bảo mật thông tin cũng liên quan rất nhiều đến các ngành khoa học khác đặc biệt là Toán học, do vậy việc trình bày đầy đủ mọi khía cạnh của nó trong khuôn khổ một giáo trình là một điều khó có thể làm được. Chính vì lý do đó, trong Giáo trình ***An toàn bảo mật dữ liệu*** này các vấn đề về đảm bảo an toàn vật lý và người dùng cũng như các vấn đề liên quan đến kỹ thuật và quy tắc sẽ không được nhắc đến nhiều. Nội dung chính trong giáo trình chỉ chủ yếu đề cập đến vấn đề bảo đảm an toàn thông tin bằng các giao thức và thuật

toán mật mã – một công cụ vốn đã xuất hiện và được sử dụng từ rất sớm để bảo đảm tính bí mật cho thông tin.

Giáo trình ***An toàn bảo mật dữ liệu*** được biên soạn phục vụ cho sinh viên đại học, cao học các ngành Công nghệ thông tin hoặc Khoa học máy tính như 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 trong việc đảm bảo an toàn và bảo mật dữ liệu.

Nội dung giáo trình bao gồm 4 chương:

Chương 1. Giới thiệu chung: Trình bày một số khái niệm, định nghĩa cơ bản và cơ sở lý thuyết thông tin áp dụng cho các hệ mật.

Chương 2. 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 3. 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 4. Hàm băm và chữ ký số: Trình bày khái niệm hàm băm và 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 vững, hiểu cụ thể và sâu sắc hơn các vấn đề lý thuyết được trình bày.

Việc biên soạn giáo trình không thể tránh khỏi các thiếu sót nhất định. Nhóm tác giả rất mong nhận được các ý kiến đóng góp quý báu của các quý đồng nghiệp, quý độc giả và các em học viên, sinh viên để cho lần tái bản sau của giáo trình được hoàn thiện hơn.

CÁC TÁC GIẢ

TRẦN ĐỨC SỰ
NGUYỄN VĂN TẢO
TRẦN THỊ LƯỢNG

Chương 1

GIỚI THIỆU CHUNG

1.1. Hệ thống thông tin và các hình thức tấn công hệ thống thông tin

1.1.1. Thông tin và hệ thống thông tin

Trên quan điểm an toàn thông tin người ta định nghĩa thông tin như sau:

Định nghĩa 1.1. *Thông tin là tập hợp các dữ liệu (các tin tức) về thế giới xung quanh chúng ta (các sự kiện, các cá nhân, các hiện tượng, các quá trình, các nhân tố và các mối liên hệ giữa chúng), được thể hiện trong dạng thức phù hợp cho việc truyền đi bởi những người này và tiếp nhận bởi những người kia và được sử dụng với mục đích thu nhận kiến thức (các tri thức) và đưa ra những quyết định.*

Ngày nay thông tin được hình thành, tồn tại và vận động trong các hệ thống thông tin – viễn thông. Chúng ta cần định nghĩa rõ về khái niệm hệ thống thông tin – viễn thông.

Định nghĩa 1.2. *Hệ thống thông tin – viễn thông là tập hợp các thiết bị kỹ thuật và bảo đảm phần mềm, liên hệ với nhau bằng các kênh truyền và nhận thông tin. Từ các yếu tố ngăn cách nhau về vị trí địa lý, chúng liên kết chặt chẽ với nhau thành một thể thống nhất nhằm mục đích bảo đảm chu trình công nghệ xử lý thông tin (tìm kiếm, lưu trữ, bảo vệ, xử lý, hiệu đính) và cung cấp cho người dùng kết quả của sự xử lý này ở dạng đòi hỏi (yêu cầu). Tóm lại, hệ thống thông tin – viễn thông bao gồm các mạng máy tính, các bảo đảm toán học (các phần mềm) và hệ thống liên lạc.*

Như vậy, ta thấy thông tin là các tri thức trong ý nghĩa rộng nhất của từ này. Vì rằng thông tin phản ánh các thuộc tính của các đối tượng vật chất và mối quan hệ giữa chúng, nên theo các khái niệm cơ bản của triết học, thông tin có thể coi là đối tượng của nhận thức.

Suy cho cùng, bảo đảm thông tin là cơ sở cho bất kỳ hoạt động nào của con người. Thông tin trở thành một trong những phương tiện cơ bản để giải quyết các vấn đề và các nhiệm vụ của một quốc gia, của các đảng chính trị và các nhà lãnh đạo của các cơ quan thương mại khác nhau và của cá nhân con người.

Ngày nay, kinh tế thế giới phát triển ở mức độ cao, khoa học công nghệ đã đưa tới sự ra đời của nền kinh tế tri thức. Lượng thông tin tích luỹ được về mọi khía cạnh của cuộc sống xã hội hiện đại trở nên khổng lồ. Các thông tin mới được sáng tạo ra với tốc độ ngày càng cao. Nhưng mặt khác, việc thu nhận thông tin bằng con đường nghiên cứu, khảo sát riêng (của cá nhân hoặc của tập thể) ngày càng trở nên đắt giá, tốn kém và khó khăn. Cho nên việc thu lượm thông tin bằng con đường rẻ hơn nhưng bất hợp pháp (tức là lấy cắp thông tin) ngày càng trở nên thường xuyên và mở rộng hơn bao giờ hết.

Trong bối cảnh đó, nhiệm vụ bảo vệ an ninh thông tin trong tất cả các lĩnh vực hoạt động của con người đang ngày càng trở nên cấp thiết: trong phục vụ các cơ quan Nhà nước (lãnh đạo, chỉ huy, an ninh, quốc phòng, đối ngoại); trong thương mại, kinh doanh; trong hoạt động khoa học công nghệ, trong sản xuất và thậm chí trong đời sống riêng tư của các cá nhân. Sự cạnh tranh thường xuyên giữa các phương pháp lấy cắp thông tin (và các phương tiện thực hiện chúng) với các phương pháp (phương tiện) bảo vệ thông tin đã dẫn đến sự xuất hiện trên thị trường rất nhiều chủng loại thiết bị bảo vệ thông tin, và cũng đã xuất hiện vấn đề lựa chọn chúng sao cho tối ưu và sử dụng cho hiệu quả trong những điều kiện cụ thể. Để làm rõ vấn đề bảo vệ an ninh thông tin, ta đi vào ba thuộc tính cơ bản của thông tin dưới đây.

1.1.2. Ba thuộc tính cơ bản của thông tin

Chúng ta định nghĩa ba thuộc tính cơ bản của thông tin như đối tượng cần bảo vệ. Đó là *tính bí mật, tính toàn vẹn* và *tính sẵn sàng dịch vụ* của thông tin. Trên thực tế khó phân biệt ranh giới giữa chúng. Ba phạm trù này có những miền giao nhau. Dễ thấy rằng, có những thông tin

mật dành riêng cho một đối tượng dùng mà việc đáp ứng tính bí mật đã bao hàm cả sự toàn vẹn và sẵn sàng phục vụ rồi. Có thể miêu tả quan hệ giữa ba tính chất cơ bản của thông tin trong sơ đồ sau:

Hình 1.1. Mối quan hệ giữa ba tính chất cơ bản của TT

➢ Đảm bảo *tính bí mật* (*Confidentiality*): có nghĩa là ngăn chặn/phát hiện/cản trở những truy nhập thông tin trái phép. Nói chung, tính bí mật được sử dụng để bảo vệ dữ liệu trong những môi trường bảo mật cao như các trung tâm quân sự hay kinh tế quan trọng, bảo vệ tính riêng tư của dữ liệu.

➢ Đảm bảo *tính toàn vẹn (Integrity):* có nghĩa là ngăn chặn/phát hiện/cản trở các sửa đổi thông tin trái phép.

➢ Đảm bảo *tính sẵn sàng (Availability):* có nghĩa là ngăn chặn/phát hiện/cản trở sự từ chối trái phép các truy nhập hợp pháp đến dịch vụ trong hệ thống.

Như vậy vấn đề an toàn thông tin có thể được hiểu là vấn đề đảm bảo ba thuộc tính cơ bản của thông tin là: tính toàn vẹn, tính bí mật và tính sẵn sàng. Ba thuộc tính này của thông tin có thể bị tác động và ảnh hưởng bởi các hình thức tấn công hệ thống thông tin mà ta quan tâm dưới đây.

1.1.3. Các hình thức tấn công vào hệ thống thông tin

An toàn thông tin (ATTT) là một nhu cầu rất quan trọng đối với các cá nhân cũng như các tổ chức xã hội và các quốc gia trên thế giới. Trước

khi sử dụng máy tính và mạng máy tính, an toàn thông tin được tiến hành thông qua các phương pháp vật lý và hành chính. Từ khi ra đời cho đến nay mạng máy tính đã đem lại hiệu quả vô cùng to lớn trong tất cả các lĩnh vực của đời sống kinh tế, chính trị, xã hội. Bên cạnh đó người sử dụng mạng phải đối mặt với các hiểm họa do thông tin trên mạng của họ bị tấn công. An toàn thông tin trên mạng máy tính bao gồm các phương pháp nhằm bảo vệ thông tin được lưu giữ và truyền trên mạng. An toàn thông tin trên mạng máy tính là một lĩnh vực đang được đặc biệt quan tâm đồng thời cũng là một công việc hết sức khó khăn và phức tạp.

Có rất nhiều các sự kiện thực tế để chứng tỏ rằng có một tình trạng rất đáng lo ngại về các tấn công thông tin trong quá trình xử lý, truyền và lưu giữ thông tin. Những tác động bất hợp pháp lên thông tin với mục đích làm tổn thất, sai lạc, lấy cắp các tệp lưu giữ tin, sao chép các thông tin mật, giả mạo người được phép sử dụng thông tin trong các mạng máy tính.

Hiện nay, có nhiều phương pháp khác nhau được sử dụng để có thể đưa ra các hiểm hoạ ATTT đối với một hệ thốngthông tin – viễn thông, ví dụ như: phương pháp liệt kê, phương pháp cây hiểm hoạ, phương pháp phân loại học... Các phương pháp này đều sử dụng các sơ đồ, bảng biểu... Dưới đây là một sơ đồ tổng quát của hệ thống thông tin và các hiểm họa ATTT đi kèm với nó.

Hình 1.2. Sơ đồ tổng quát hệ thống thông tin viễn thông và các hiểm hoạ ATTT đi kèm

Trên mạng máy tính, thông tin bao gồm nhiều loại khác nhau như: văn bản, hình ảnh, âm thanh. Chúng được lưu giữ trong các thiết bị như: ổ đĩa, băng từ... hoặc được truyền qua kênh công khai. Những thông tin có giá trị luôn luôn gặp những mối đe dọa của những người không có thẩm quyền biết nội dung thông tin. Họ có thể là những người dùng bất hợp pháp hoặc thậm chí là những người dùng trong nội bộ của cơ quan, tổ chức.

Trên đường truyền công khai, thông tin có thể bị tấn công bởi những người không được uỷ quyền nhận tin, ta gọi là kẻ tấn công.

Các tấn công đối với thông tin trên mạng bao gồm:

1.1.3.1. Ngăn chặn thông tin (Interruption)

Tài nguyên thông tin bị phá huỷ, không sẵn sàng phục vụ hoặc không sử dụng được. Đây là hình thức tấn công làm mất khả năng sẵn sàng phục vụ của thông tin. Những ví dụ về kiểu tấn công này là phá huỷ đĩa cứng, cắt đứt đường truyền tin, vô hiệu hoá hệ thống quản lý tệp.

1.1.3.2. Chặn bắt thông tin (Interception)

Kẻ tấn công có thể truy nhập tới tài nguyên thông tin. Đây là hình thức tấn công vào tính bí mật của thông tin. Trong một số tình huống kẻ tấn công được thay thế bởi một chương trình hoặc một máy tính. Việc chặn bắt thông tin có thể là nghe trộm để thu tin trên mạng và sao chép bất hợp pháp các tệp hoặc các chương trình.

Hình 1.3. Các hình thức tấn công đối với thông tin trên mạng

'.1.3.3. Sửa đổi thông tin (Modification)

Kẻ tấn công truy nhập, chỉnh sửa thông tin trên mạng. Đây là hình thức tấn công lên tính toàn vẹn của thông tin. Kẻ tấn công có thể thay đổi giá tr trong tệp dữ liệu, sửa đổi một chương trình để nó vận hành khác đi và sửa đổi nội dung các thông báo truyền trên mạng.

'.1.3.4. Chèn thông tin giả (Fabrication)

Kẻ tấn công chèn các thông tin và dữ liệu giả vào hệ thống. Đây là hình thức tấn công lên tính xác thực của thông tin. Nó có thể là việc chèn các thông báo giả mạo vào mạng hay thêm các bản ghi vào tệp.

Các kiểu tấn công trên được phân chia thành hai lớp cơ bản là tấn công chủ động và bị động. Hình 1.4 chỉ ra các các kiểu tấn công thuộc các lớp tấn công chủ động, tấn công bị động tương ứng.

- *Tấn công bị động*

Là kiểu tấn công chặn bắt thông tin như: nghe trộm và quan sát truyền tin. Mục đích của kẻ tấn công là biết được thông tin truyền trên mạng. Có hai kiểu tấn công bị động là khám phá nội dung thông báo và phân tích luồng thông tin.

Hình 1.4. Các tấn công bị động và chủ động

ĐẠI HỌC THÁI NGUYÊN
TRUNG TÂM HỌC LIỆU

Việc khám phá nội dung có thể được thực hiện bằng cách nghe trộm các cuộc nói chuyện điện thoại, đọc trộm thư điện tử hoặc xem trộm nội dung tệp tin rõ .

Trong kiểu phân tích luồng thông tin, kẻ tấn công thu các thông báo được truyền trên mạng và tìm cách khám phá thông tin. Nếu nội dung các thông báo bị mã hoá thì đối phương có thể quan sát các mẫu thông báo để xác định vị trí và định danh của máy tính liên lạc và có thể quan sát tần số và độ dài thông báo được trao đổi, từ đó đoán ra bản chất của các cuộc liên lạc.

Tấn công bị động rất khó bị phát hiện vì nó không làm thay đổi số liệu và không để lại dấu vết rõ ràng. Biện pháp hữu hiệu để chống lại kiểu tấn công này là ngăn chặn chứ không phải là phát hiện.

- *Tấn công chủ động*

Là các tấn công sửa đổi luồng dữ liệu hay tạo ra luồng dữ liệu giả và có thể được chia làm bốn loại nhỏ sau :

- *Đóng giả (Masquerade)*: Một thực thể (người dùng, máy tính, chương trình, ...) đóng giả thực thể khác.

- *Dùng lại (Replay)*: Thụ động bắt các thông báo và sau đó truyền lại nó nhằm đạt được mục đích bất hợp pháp.

- *Sửa đổi thông báo (Modification of messages)*: Một bộ phận của thông báo hợp lệ bị sửa đổi hoặc các thông báo bị làm trễ và thay đổi trật tự để đạt được mục đích bất hợp pháp.

- *Từ chối cung cấp dịch vụ (Denial of service)*: Ngăn hoặc cấm việc sử dụng bình thường hoặc quản lý các tiện ích truyền thông.

Tấn công này có thể có chủ ý cụ thể, ví dụ một kẻ tấn công có thể ngăn cản tất cả các thông báo được chuyển tới một đích nào đó (như dịch vụ kiểm tra an toàn chẳng hạn), vô hiệu hoá một mạng hoặc tạo ra tình trạng quá tải với các thông báo của họ làm giảm hiệu năng mạng.

Chúng ta thấy rằng hai kiểu tấn công chủ động và bị động có những đặc trưng khác nhau. Kiểu tấn công bị động khó phát hiện nhưng có biện

pháp để ngăn chặn thành công. Ngược lại, kiểu tấn công chủ động dễ phát hiện nhưng lại rất khó ngăn chặn tuyệt đối, nó cũng đòi hỏi phải bảo vệ vật lý đối với tất cả các phương tiện truyền thông ở mọi lúc, mọi nơi. Giải pháp để chống lại kiểu tấn công này là phát hiện chúng và khôi phục mạng sau khi mạng bị tấn công và thông tin bị trễ.

1.2. Mật mã và an toàn thông tin

1.2.1. Các ứng dụng của mật mã

1.1.2.1. Ứng dụng trong đời sống thông tin, kinh tế, xã hội

Sự phát triển lớn mạnh của công nghệ thông tin trong những năm vừa qua, đặc biệt là sự bùng nổ của mạng Internet đã dẫn đến việc sử dụng rộng rãi hệ thống máy tính trong mọi tổ chức, cá nhân và công cộng. Các hoạt động thông tin, kinh tế, xã hội cũng đang được áp dụng, triển khai rộng rãi qua mạng Internet. Từ đó cũng đã làm xuất hiện một nền kinh tế mới, nền kinh tế thương mại điện tử, nơi mà các hoạt động mua bán và dịch vụ đều dựa trên hệ thống mạng Internet.

Hệ thống World Wide Web trước kia sử dụng giao thức HTTP để đảm bảo cho việc truyền nhận thông tin tới các đối tượng, nhưng lại không thể đảm bảo bí mật cho các thông tin đó khi truyền đi, thì ngày nay đã được thay thế bằng giao thức HTTPS, ngoài việc đảm bảo truyền nhận thông thường thì nội dung thông tin cũng được đảm bảo giữ bí mật.

Khi các hàm băm chưa được sử dụng, các ngân hàng lưu trữ thông tin thẻ tín dụng ở dạng clear-text (gồm: tên chủ thẻ, số tài khoản, mã PIN, ngày hết hạn, v.v...). Điều này tạo ra nguy cơ bị lộ toàn bộ thông tin thẻ tín dụng, khi kẻ tấn công có thể truy cập và đọc được nội dung những trường hoặc file lưu trữ của cơ sở dữ liệu đó (thông qua một số kiểu tấn công như SQL Injection chẳng hạn). Ngày nay mối lo ngại này đã được loại bỏ, các thông tin bí mật sẽ không được lưu trữ một cách trực tiếp, mà được thay thế bằng giá trị băm của thông tin đó, nên cho dù kẻ tấn công có thể đọc được giá trị băm, thì cũng rất khó để tìm ra thông tin bí mật trước khi bị băm là thông tin gì.

Các hoạt động xã hội trước kia, như: nộp thuế, kê khai thuế, vốn yêu cầu những văn bản có giá trị pháp lý cao, bắt buộc phải có chữ ký (hoặc dấu vân tay) của người nộp, kê khai thuế. Ngày nay, hình thức đó đã dần được chuyển sang bằng một phương pháp mới, đó là sử dụng chữ ký số để thay thế. Chữ ký số đã được chứng minh là an toàn về mặt tính toán, tức thời gian để có thể tạo ra chữ ký số giả một cách hợp lệ với kẻ tấn công có năng lực tính toán hạn chế sẽ là rất lớn. Ví dụ, để phá RSA với độ dài khóa 1024 bít, độ phức tạp tính toán là 3.10^{11} MIPS, còn với RSA 2048 bít, độ phức tạp tính toán là 3.10^{20} MIPS, với ECC độ dài khóa 234 bít, độ phức tạp tính toán là $1.6.10^{28}$ MIPS (1 MIPS = 1 triệu tập lệnh trên một giây).

1.1.2.2. Ứng dụng trong an ninh, quốc phòng

Ngay từ khi mới ra đời, đối tượng chủ yếu của mật mã là những người có liên quan đến quân đội, ngoại giao và chính phủ nói chung. Từ đó cho đến nay, mật mã đã được sử dụng như một loại công cụ, vũ khí để bảo vệ các chiến lược và bí mật của quốc gia.

Trong suốt thời kỳ trước và trong chiến tranh Thế giới lần thứ II, nhằm bảo vệ bí mật quân sự, không để lộ ý đồ tác chiến, Quân đội Đức đã mã hóa hầu hết các chỉ thị và mệnh lệnh của họ. Chiếc máy mã Enigma có cơ chế mã hóa phức tạp hơn bất cứ loại mật mã nào từng biết đến trong lịch sử trước đó, và trong suốt quá trình sử dụng nó cũng liên tục được cải tiến, ngày càng phức tạp hơn, khiến cho người Đức luôn tin rằng "*Enigma là bất khả xâm phạm*". Thế nhưng, từ đầu những năm 1933, các nhà toán học của Cục mật mã Ba Lan đã có thể giải được toàn bộ điện mật của Đức.Và với sự giúp đỡ của Ba Lan, người Anh và Pháp đã đọc được các bức điện mật của Đức. Nhiều nhà sử học đã đánh giá rằng, nhờ công trình giải mã bằng máy mã Enigma mà Thế chiến thứ II đã "*ngắn đi đến hai năm*".

Sau chiến tranh thế giới thứ II, công nghệ mật mã tiếp tục được giới quân sự các nước đầu tư nghiên cứu và phát triển. Một số quốc gia có những cơ quan chuyên nghiên cứu về những công nghệ này, ví dụ như

Cơ quan An ninh Quốc gia Hoa Kỳ/Cục An ninh Trung ương (NSA/CSS - National Security Agency/Central Security Service). NSA có liên quan rất nhiều đến tranh cãi xung quanh quá trình hình thành Chuẩn mã hóa dữ liệu (DES), một chuẩn mã khối dùng cho chính phủ. Trong suốt quá trình thiết kế DES tại IBM vào thập kỷ 1970, NSA đã đề xuất những thay đổi trong thuật toán. Vì thế, nhiều người nghi ngờ NSA đã cố tình làm yếu thuật toán để có thể phá vỡ khi cần thiết. Nghi ngờ tập trung chủ yếu vào một thành phần quan trọng của thuật toán, S-box. Đây có thể là một cổng sau để NSA có thể dễ dàng đọc được các thông tin đã được mã hóa. Ngoài ra độ dài khóa cũng bị rút ngắn, tạo điều kiện cho NSA có thể phá vỡ hệ thống với hệ thống siêu máy tính của mình. Khi kỹ thuật phân tích mã lượng sai được tìm ra thì những nghi ngờ này có phần được giảm bớt. S-box đã được sửa đổi có khả năng chống lại dạng tấn công này. Vì thế nhiều khả năng NSA đã biết đến phân tích mã lượng sai vào thời điểm thiết kế DES, trước khi kỹ thuật này được độc lập phát hiện vài thập kỷ sau đó. Tuy nhiên việc can thiệp để giảm độ dài khóa DES từ 128 (theo đề nghị của IBM) xuống còn 56 bít thì chỉ có thể giải thích rằng đây là cố gắng làm yếu thuật toán để NSA với công suất tính toán vượt trội của mình có thể tấn công duyệt toàn bộ để giải mã trong trường hợp cần thiết.

NSA cũng là yếu tố quan trọng trong những tranh cãi hồi cuối thập kỷ 1990 trong vấn đề xuất khẩu công nghệ mật mã. Từ lâu, phần cứng và phần mềm mật mã được xếp cùng hạng với máy bay chiến đấu, xe tăng, pháo và bom nguyên tử. Tại nhiều thời điểm, NSA/CSS đã cố gắng hạn chế việc xuất bản các tài liệu nghiên cứu về mật mã học.

1.2.2. Vai trò của mật mã trong bảo đảm an toàn thông tin

Ở trên ta đã xem xét một cách tổng quan về vai trò và ứng dụng của mật mã nói chung. Nhưng vai trò này tác động lên khâu nào của hệ thống thông tin và tác động như thế nào. Vấn đề này sẽ được xem xét ở mục này.

Trước hết, ta xem xét mô hình chung nhất của một hệ thống thông tin và vị trí, vai trò của mật mã trong hệ thống này, dưới đây là sơ đồ khối của một hệ thống thông tin số:

Hình 1.5. 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 sửa đổi, đánh cắp và giả mạo thông tin. Trong khuôn khổ của cuốn giáo trình này, tác giả sẽ tập trung trình bày về việc đảm bảo an toàn và bảo mật dữ liệu này bằng các thuật toán mật mã.

1.3. Sơ lược về mật mã học

Khoa học mật mã đã ra đời từ hàng nghìn năm. Tuy nhiên, trong suốt nhiều thế kỷ, các kết quả của lĩnh vực này hầu như không được ứng dụng trong các lĩnh vực dân sự thông thường của đời sống – xã hội mà chủ yếu được sử dụng trong lĩnh vực quân sự, chính trị, ngoại giao... Ngày nay, các ứng dụng mã hóa và bảo mật thông tin đang được sử dụng ngày càng phổ biến trong các lĩnh vực khác nhau trên thế giới, từ các lĩnh vực an ninh, quốc phòng..., cho đến các lĩnh vực dân sự như thương mại điện tử, ngân hàng...

Từ thời xa xưa, để tỏ lòng tôn kính những người đã chết, người Ai Cập đã khắc những mã tượng hình lên các ngôi mộ. Qua nhiều thế kỷ, phương pháp mật mã cũng đã có nhiều biến đổi. Chúng ta tạm phân mật mã làm hai phần, mật mã cổ điển và mật mã "*hiện đại*". Mật mã hiện đại gồm mật mã đối xứng và mật mã bất đối xứng. Mật mã hiện nay đang phát triển rất mạnh với rất nhiều thuật toán mã nổi bật như: DES, 3DES,

IDEA, Feal, AES, RSA... Còn mật mã cổ điển là mật mã được mã hoá và giải mã bằng thủ công. Mật mã loại này ra đời sớm nhất, nó được sử dụng lâu đời và là cơ sở, nền tảng để phát triển mật mã hiện đại.

1.3.1. Các khái niệm cơ bản

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) hay mã thám*: là khoa học nghiên cứu cách phá các hệ mật nhằm phục hồi bản rõ ban đầu từ bản mã.

Các bản tin rõ và bản tin mã được định nghĩa như sau:

Bản tin rõ (Plain text): Bản tin rõ tức là một bản tin có mang nội dung thông tin mà người đọc có thể hiểu được nó nói cái gì hoặc là nó có ý nghĩa rõ ràng. Bản tin rõ có thể tồn tại dưới dạng chữ viết, tiếng nói, hình vẽ, biểu bảng... tương ứng ta sẽ có khái niệm mã ký tự, mã thoại, mã fax, mã dữ liệu, ...

Bản mã mật (Cipher text): Bản mã mật thường được biểu diễn dưới dạng một dãy các ký hiệu hoặc có thể cũng thuộc bảng chữ cái những không theo một quy tắc cú pháp nào cả. Có thể xem đó là dãy ngẫu nhiên.

Mục tiêu cơ bản của mật mã là cho phép hai người, thường được đề cập tới như Alice và Bob, liên lạc trên kênh không an toàn theo cách mà đối thủ Oscar không thể hiểu cái gì đang được nói. Kênh này có thể là đường điện thoại hoặc máy tính chẳng hạn. Thông tin mà Alice muốn gửi tới cho Bob sẽ được gọi là "*thông báo rõ*". Nó có thể là bài tiếng Anh, dữ liệu số v.v... Cấu trúc của nó hoàn toàn tuỳ ý. Alice mã thông báo rõ bằng cách dùng khoá xác định trước, và gửi thông báo mã thu được trên kênh không an toàn. Oscar, dù thấy thông báo mã này trên kênh không an toàn bằng cách nghe trộm, cũng không xác định được thông báo rõ là gì; nhưng Bob, người biết khoá mã, có thể giải thông báo mã và thiết lập thông báo rõ.

Dùng quan niệm toán học ta sẽ mô tả khái niệm này hình thức hơn.

Định nghĩa 1.3. (Hệ mật mã)

Hệ mật hay hệ mật mã là một bộ 5 thành phần (P, C, K, E, D) thoả mãn các điều kiện sau đây:

1/ P là tập hữu hạn các bản rõ có thể.

2/ C là tập hữu hạn các bản mã có thể.

3/ K là tập hữu hạn các khoá có thể (không gian khoá).

4/ Với mỗi $k \in K$, tồn tại một quy tắc mã $e_k \in E$ và một quy tắc giải mã tương ứng $d_k \in D$. Mỗi e_k: $P \rightarrow C$ và d_k: $C \rightarrow P$ thoả mãn:

$d_k(e_k(x)) = x$ với mỗi bản rõ $x \in P$.

Alice và Bob sẽ thực hiện giao thức sau đây để sử dụng một hệ mật. Trước hết, họ chọn khoá ngẫu nhiên k thuộc K. Họ làm điều này theo một cách an toàn, chẳng hạn khi họ ở cùng một chỗ và không bị Oscar quan sát hoặc họ dùng một kênh an toàn khi ở xa nhau để trao đổi và thỏa thuận khóa mật k.

Sau đó, giả sử Alice muốn gửi một thông báo tới Bob trên kênh không an toàn. Thông báo đó là dòng:

$$x = x_1 x_2 \ldots x_n \qquad n \geq 1, x_i \in P, \ 1 \leq i \leq n$$

Alice tính: $\quad y_i = e_k(x_i), \ 1 \leq i \leq n$

và mã thu được là: $\quad y = y_1 y_2 \ldots y_n$

Alice sẽ gửi y trên kênh không an toàn. Khi nhận được $y_1 y_2 \ldots y_n$, Bob sẽ dùng d_k để phục hồi thông báo ban đầu $x_1 x_2 \ldots x_n$.

Hình 1.6. Sơ đồ hệ thống thông tin mật

Rõ ràng phải có: $e_k(x_1) \neq e_k(x_2)$ khi $x_1 \neq x_2$. Nếu $P = C$ thì e_k là một phép hoán vị.

1.3.2. Các kiểu tấn công vào hệ mật mã

Nhìn chung, mục tiêu của nhà mã thám là biết khoá của hệ mã đang dùng, hoặc nếu không được thì tìm cách hiểu nội dung bản rõ tương ứng, hoặc một phần của bản rõ.

Các tấn công được phân loại dựa trên sự sở hữu của đối phương đối với bản rõ và bản mã và xếp hạng tăng dần theo hiệu quả tấn công của họ đối với hệ mã.

- *Tấn công chỉ biết bản mã (Ciphertext only attack)*: trong loại tấn công bị động này, đối phương cố gắng để rút ra một số thông tin về khóa (hoặc về bản rõ) chỉ bằng cách quan sát một số lượng nào đó các bản mã. Thông thường, chúng ta thừa nhận đối phương biết một số thuộc tính về bản rõ hoặc khóa, ví dụ đối phương có thể biết rằng bản rõ bao gồm các ký tự mã ASCII. Các hệ mã có thể bị tổn thương đối với các tấn công chỉ biết bản mã được coi là bị bẻ gãy hoàn toàn.

- *Tấn công bản rõ đã biết (Known plaintext attack)*: trong trường hợp này, giả sử rằng đối phương biết một lượng cặp bản rõ – mã nào đó, mục đích của loại tấn công bị động này là tìm ra khóa. Điển hình chúng ta thấy xuất hiện tấn công bản rõ đã biết trong trường hợp ở đó đối phương có thể thu được các phiên bản bản mã (encrypted version) khác nhau của dữ liệu đã biết, như dữ liệu trao đổi trong các pha cài đặt của một giao thức. Ví dụ điển hình của tấn công bản rõ đã biết là thám mã tuyến tính.

- *Tấn công bản rõ lựa chọn không thích ứng (Non adaptive chosen-plaintext attack):* khi thực hiện tấn công chủ động này, đối phương có thể lựa chọn bản rõ và thu được bản mã tương ứng, bản rõ phải không phụ thuộc vào bản mã đã thu được: đầu tiên có thể xem xét bản rõ và bản mã một cách tương đương. Sau đó, đối phương sử dụng những thông tin suy diễn hợp lệ để lấy hoặc là khóa hoặc là bản rõ tương ứng với các bản mã chưa biết trước đó. Có thể bắt gặp kịch bản như thế trong trường hợp khi module chống trộm (temper-proof) (bản quyền) ứng dụng mã khối với một khóa cố định rơi vào tay đối phương nhưng đối phương không thể khôi

phục trực tiếp được khóa (ví dụ theo nghĩa vật lý). Một ví dụ điển hình của tấn công lựa chọn bản rõ không thích ứng là thám mã lượng sai.

- *Tấn công bản rõ lựa chọn thích ứng (Adaptive chosen-plaintext attack):* tấn công này như tấn công lựa chọn bản rõ trong đó việc lựa chọn bản rõ phụ thuộc vào bản mã nhận được từ các yêu cầu trước đó.

- *Tấn công bản mã lựa chọn thích ứng (hoặc không thích ứng) ((No) Adaptive chosen-ciphertext attack):* giả sử đối phương có thể giải mã được bất kỳ bản mã nào (theo cách thích ứng hoặc không) và thu được bản rõ tương ứng với mục đích khôi phục khóa hoặc để mã hóa các bản rõ (trước đó chưa quan sát được). Trong trường hợp các mã khối, loại tấn công này rất giống với tấn công bản rõ lựa chọn.

- *Kết hợp tấn công bản mã lựa chọn và tấn công bản rõ lựa chọn (Combined chosen-plaintext and chosen-ciphertext attack):* đây là loại tấn công thích ứng rất mạnh giả thiết rằng đối phương có thể mã hóa hoặc giải mã bất kỳ văn bản mong muốn. Ví dụ tiêu biểu của tấn công này là tấn công boomerang của Wagner.

- *Tấn công khóa quan hệ (Related-key attack):* mô hình tấn công này giả sử đối phương biết (hoặc có thể chọn) thêm quan hệ toán học nào đó giữa các khóa sử dụng để mã hóa và giải mã nhưng không phải giá trị của khóa.

Ngoài các kiểu tấn công trên, một dạng tấn công kinh điển nhất vào các hệ mật mã là tấn công vét cạn. Trong đó, thám mã cố gắng duyệt tất cả các khả năng của khóa, nếu như ứng viên nào của khóa có thể giải mã được bản mã để tạo ra một bản rõ có nghĩa thì ứng viên này được giả định là khóa thật của hệ mã. Với các hệ mã có độ dài khóa lớn thì dạng tấn công này không hiệu quả.

1.3.3. Phân loại các thuật toán mật mã

Dưới đây là mô hình phân loại các thành phần mật mã cơ bản, được sử dụng trong hầu hết các ứng dụng mật mã.

Khi sử dụng các thành phần mật mã trên, ta cần quan tâm tới các tính chất sau:

• *Độ an toàn*: đây thường là vấn đề khó đánh giá. Thường được xác định bằng số lượng công việc cần thiết để phá vỡ đối tượng, được tính bằng số phép toán yêu cầu (theo phương pháp tốt nhất được biết hiện tại) để có thể phá vỡ đối tượng đó, hay còn gọi là độ an toàn tính toán.

• *Chức năng:* khi sử dụng thực tế, các thành phần bảo mật cơ bản cần kết hợp lại với nhau để đáp ứng các mục tiêu an toàn thông tin khác nhau. Tùy đặc tính của từng thành phần mà nó sẽ có hiệu quả cao nhất với một mục tiêu an toàn cụ thể.

• *Phương pháp vận hành:* các thành phần bảo mật cơ bản khi được áp dụng bằng các cách khác nhau, với những đầu vào khác nhau, thì sẽ có những đặc điểm khác nhau. Do đó, một thành phần bảo mật cơ bản có thể cung cấp những chức năng rất khác nhau, tùy thuộc vào chế độ hoạt động hoặc cách sử dụng.

Hình 1.7. Lược đồ các thành phần mật mã cơ bản

• *Hiệu suất:* tính chất này dùng để chỉ hiệu quả của một thành phần bảo mật trong một chế độ vận hành riêng. (Ví dụ, một thuật toán mã hóa có thể được đánh giá bằng số bit mà nó có thể mã hóa trong một giây).

• *Khả năng cài đặt:* liên quan đến những khó khăn khi cài đặt, triển khai trong thực tế, có thể trong phần cứng hoặc phần mềm.

Tầm quan trọng của các tiêu chí khác nhau phụ thuộc rất nhiều vào ứng dụng và nguồn tài nguyên hiện có. Ví dụ, trong một môi trường mà sức mạnh tính toán bị hạn chế, ta có thể phải đánh đổi một mức độ bảo mật rất cao cho hiệu năng của hệ thống.

1.4. Cơ sở toán học của lý thuyết mật mã

Phần này sẽ trình bày về độ phức tạp tính toán và kiến thức về lý thuyết số nhằm phục vụ cho các thuật toán mật mã.

1.4.1. Kiến thức về độ phức tạp tính toán

1.4.1.1. 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 trình 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 A trên bảng kí tự Σ (tức có đầu vào là các từ trong Σ). Độ phức tạp tính toán của thuật toán 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 A cần để thực hiện tiến trình tính toán của mình trên các dữ liệu có độ dài $\leq n$. Ta nói thuật toán 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 A.

Một lớp bài toán vô cùng quan trọng đó là lớp các *bài toán quyết định* và từ nay 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 Σ 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. Ở phần này chúng ta đã tìm hiểu sơ qua về độ phức tạp tính toán như một hàm $f_A(n)$ phụ thuộc vào kích thước dữ liệu vào n, dưới đây ta sẽ đưa ra các thức đánh giá độ phức tạp tính toán này.

1.4.1.2. Độ phức tạp tính toán

Định nghĩa 1.4. *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 1.1.

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

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

3. 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 1.5. *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 l O (k^d), tức là tương đương với một đa thức của k.

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

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

Cũng có những thuật toán có độ phức tạp trung gian giữa đa thức và mũ. Ta thường gọi đó là thuật toán dưới mũ. Chẳng hạn thuật toán nhanh

nhất được biết hiện nay để phân tích một số nguyên n ra thừa số là thuật toán có độ phức tạp:

$$\exp(\sqrt{\text{lognloglogn}})$$

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.

1.4.1.3. 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 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 Σ_1 và Σ_2. Một thuật toán f: $\Sigma_1^* \rightarrow \Sigma_2^*$ được gọi là một *phép quy 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 A *chấp nhận* dữ liệu x, nếu với dữ liệu vào x thuật toán 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 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 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 $P \in P$, thì P = NP

Từ các tính chất đó ta có thể xem rằng trong lớp NP, 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 P thì lập tức suy ra 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 [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!

1.4.2. Kiến thức về lý thuyết số

1.4.2.1. Số học modulo và đồng dư

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

Định nghĩa 1.6. Ước số

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

Định nghĩa 1.7. (Phép chia 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

$$\mathrm{a\ div\ b} = \left[\frac{a}{b}\right], \qquad \mathrm{a\ mod\ b} = a - b\left[\frac{a}{b}\right]$$

Ví dụ 1.2. a = 73, b = 17.

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

Định nghĩa 1.8. Ước chung.

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

Định nghĩa 1.9. 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 1.3. Các ước chung của 12 và 18 là $\{\pm 1, \pm 2, \ \pm 3, \pm 6\}$

$$(12,18) = 6$$

Để xác định ƯCLN của hai số ta thường sử dụng thuật toán Euclide dưới đây

Thuật toán Euclide

Tính ƯCLN 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 ← a mod b, a ← b, b ← r

(2) Return (a).

Định lý 1.1.

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

Ví dụ 1.4. 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 = 2.114 + 38

76 = 3.38 + 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 mà ta sẽ xem xét ở phần dưới (thuật toán Euclide mở rộng).

Định nghĩa 1.10. Bội chung nhỏ nhất (BCNN)

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

(i) $a|d$, $b|d$.

(ii) Nếu có $a|c$, $b|c$ thì $d|c$.

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

Tính chất

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

Ví dụ 1.5.

$(12,18) = 6 \quad \Longrightarrow \quad \text{BCNN}(12,18) = \frac{12.18}{6} = 36$

Định nghĩa 1.11.

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

Định nghĩa 1.12.

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 1.2. (Định lý cơ bản của số học)

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

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

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 1.13.

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 1.3.

Với $\forall\, n \geq 5$*:*

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

Định nghĩa 1.14.

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* $n|(a - b)$.

Số nguyên n được gọi là modulo đồng dư.

Ví dụ 1.6. $24 \equiv 9 \bmod 5$ 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) $a \equiv b \pmod n$ nếu và chỉ nếu a và b cũng có phần dư khi chia cho n.

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

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

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

thì $a \equiv c \pmod n$

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

$a + b \equiv a_1 + b_1 \pmod n$ và $a.b \equiv a_1.b_1 \pmod n$

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 $a = q.n + r$ với $0 \leq r \leq n$ thì $a \equiv r \pmod n$.

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 mod n. Như vậy a và r có thể được dùng để biểu thị cho lớp tương đương này.

Định nghĩa 1.15.

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

Ví dụ 1.7. $Z_{25} = \{0, 1, 2, \dots, 24\}$. Trong Z_{25} ta có:

$$13 + 16 = 4 \text{ vì } 13 + 16 = 29 \equiv 4 \pmod{25}$$

Tương tự $13 . 16 = 8$ trong Z_{25}.

Định nghĩa 1.16. (Phần tử nghịch đảo).

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

$$ax \equiv 1 \pmod{n}$$

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 1.17.

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

Định lý 1.4.

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

Ví dụ 1.8. Các phần tử khả nghịch trong Z_9 là 1, 2, 4, 5, 7 và 8.

Chẳng hạn $4^{-1} = 7$ vì $4.7 \equiv 1 \pmod{9}$

Để tìm nghịch đảo của một phần tử bất kỳ ta sử dụng thuật toán Euclide mở rộng

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. $q \leftarrow \lfloor a/b \rfloor$, $r \leftarrow a - qb$,

$$x \leftarrow x_2 - qx_1 \ , \qquad y \leftarrow y_2 - qy_1$$

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 1.5.

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

Ví dụ 1.9. Bảng 1.2 sau chỉ ra các bước của thuật toán trên với các giá trị vào $a = 4864$ và $b = 3458$

Bảng 1.1. 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

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).

Bây giờ ta xét các phương trình đồng dư tuyến tính.

$$ax=b \pmod n \qquad (1.1)$$

trong đó a,b,n là các số nguyên, n>0, và x là ẩn số.

Định lý 1.6.

Cho $d = (a, n)$. Phương trình đồng dư $ax \equiv b \pmod n$ có nghiệm x nếu và chỉ nếu $d|b$, 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 $n|d$.

Định lý 1.7. (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:

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\cdots\cdots\cdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

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 được 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 1.10.

Cặp phương trình đồng dư $x \equiv 3 \pmod 7, x \equiv 7 \pmod{13}$

có nghiệm duy nhất $x \equiv 59 \pmod{91}$

Định lý 1.8.

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

$$x \equiv a \pmod{n_1}, x \equiv a \pmod{n_2}$$

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

Định nghĩa 1.18.

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

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

Định nghĩa 1.19.

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 1.9.

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

(2) Nếu n là tích của các số nguyên khác nhau và nếu $r \equiv s \pmod{\Phi(n)}$ thì $a^r \equiv a^s \pmod n$ 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 1.10. *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 1.20.

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

Định nghĩa 1.21.

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

Ví dụ 1.11.

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

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

Bảng 1.2. Cấp của các phần tử trong Z^*_{21}

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

Định nghĩa 1.22.

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:

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

(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 $\alpha^{\Phi(n)/p} \neq 1 \pmod n$ đối với mỗi nguyên tố p của $\Phi(n)$

Ví dụ 1.12. 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 1.23.

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 $x^2 \equiv a \pmod n$. Nếu không tồn tại x như vậy thì a được gọi là thặng dư không bậc hai mod 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 1.11.

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 $|Q_n| = \frac{(p-1)}{2}$ và $|\overline{Q_n}| = \frac{(p-1)}{2}$, 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 1.13. $\alpha = 6$ là một phần tử sinh của Z_{13}^*. Các luỹ thừa của α được liệt kê ở bảng sau:

Bảng 1.3. Các lũy thừa của 6

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 1.12.

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* $a \in Q_p$. *Điều đó dẫn tới*

$$|Q_n| = |Q_p|.|Q_q| = \frac{(p-1).(q-1)}{4}$$

Và $|\overline{Q_n}| = \frac{3(p-1)(q-1)}{4}$

Ví dụ 1.14. Cho n=21.

Khi đó $Q_{21} = \{1,4,16\}$ $\quad \overline{Q_{21}} = \{2,5,8,10,11,13,17,19,20\}$

Định nghĩa 1.24.

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

Định lý 1.13. (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 1.15. Các căn bậc 2 của 12 mod 37 là 7 và 30. Các căn bậc 2 của 121 mod 315 là 11, 74, 101, 151, 164, 214, 241 và 304.

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

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

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

VÀO: $a \in Z_n$ và số nguyên k, $(0 \le k \le n)$ 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 1.16. Bảng 1.4 sau chỉ ra các bước tính toán

$5^{596} \bmod 1234 = 1013$

Bảng 1.4. Tính 5^{596} *mod* 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 1.5.

Bảng 1.5. 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	$a^k \bmod n, k < n$	$0((\lg n)^3)$

1.4.2.2. Các kí hiệu Legendre và Jacobi

Định nghĩa 1.25.

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:*

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

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

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

(4) $\frac{2}{p} \equiv (-1)^{\frac{p^2-1}{8}}$. 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 (mod 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.

Định nghĩa 1.26.

Cho $n \geq 3$ *là các số nguyên lẻ có phân tích*

$n = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$. *Khi đó ký hiệu Jacobi* $\left(\frac{a}{n}\right)$ *được định nghĩa là*

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

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

Các tính chất của kí hiệu Jacobi

Cho $n \geq 3$ là các số nguyên 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 $\text{UCLN}(a, n) \neq 1$.

(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 $a \equiv b \bmod n$ 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 $n \equiv 1 (\bmod 4)$

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

(7) $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$.

Bởi vậy $\left(\frac{2}{n}\right) = 1$ nếu $n \equiv 1$ hoặc $7 (\bmod 8)$

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

(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 $3 (\bmod 4)$, 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.

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 mod 8

(5) Nếu $n \equiv 3 \pmod 4$ và $a_1 \equiv 3 \pmod 4$ thì đặt $s \leftarrow -s$

(6) Đặt $n_1 \leftarrow n \bmod a_1$

(7) Return (s.JACBI(n_1, a_1))

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

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.

Ví dụ 1.17. ***Tính toán kí hiệu Jacobi:***

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

$$\left(\frac{158}{235}\right)=\left(\frac{2}{235}\right)\left(\frac{79}{235}\right)=(-1)\left(\frac{235}{79}\right)(-1)^{78.234/4}=\left(\frac{77}{79}\right)$$

$$=\left(\frac{77}{79}\right)(-1)^{76.78/4}=\left(\frac{2}{77}\right)=-1$$

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 $a \in Q_n$.

Ví dụ 1.18. ***(các thặng dư bậc 2 và không thặng dư bậc 2):***

Bảng 1.6. 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 1.6 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 $5 \notin Q_{21}$.

Định nghĩa 1.27.

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* $J_n - Q_n$.

Định lí 1.14:

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.*

1.4.2.3. Căn nguyên thủy

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

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

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, $1 \leq b \leq p - 1$ 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. $c \leftarrow b^t$ mod p và $r \leftarrow a^{(t+1)/2}$ mod p

6. For i from 1 to s – 1 do

7. Tính $d = (r^2.a^{-1})^{2^{s-i-1}} \bmod p$

8. Nếu d ≡ -1 mod p thì đặt r ← r.c mod p

9. Đặt c ← c^2 mod p

10. Return (r, -r)

- **Thuật toán tính căn bậc hai modulo p khi p ≡ 3 mod 4**

VÀO: số nguyên tố lẻ p với p ≡ 3 mod 4 và a ∈ Q_p

RA: hai căn bậc hai của a modulo p

1. Tính r = $a^{(p+1)/4}$ mod 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 ≡ 5 mod 8 và a ∈ Q_p

RA: hai căn bậc hai của a modulo p

1. Tính d = $a^{(p+1)/4}$ mod p

2. Nếu d = 1 thì r = $a^{(p+3)/8}$ mod p

3. Nếu d = -1 thì r = $2a(4a)^{(p-5)/8}$ mod 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 ≡ 3 mod 4, q ≡ 3 mod 4), c∈ 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 = $c^{(p+1)/4}$mod p

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

x = (aps + bqr) mod n

y = (aps – bqr) mod n

3. Return (±x, ±y)

1.4.2.3. Các số nguyên Blum

Định nghĩa 1.28.

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 1.15:

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 1.29.

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 mod n.

Ví dụ 1.19. ***(Số nguyên Blum).***

Đối với số nguyên Blum n= 21. Ta có $J_n = \{1,4,5,16,17,20\}$ và $\widetilde{Q_n}$={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 1.16:

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

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

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

1.5. Bài tập

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

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

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

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

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

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

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.

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

8. Giải hệ phương trình đồng dư sau:

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

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

a. 17^{-1} mod 101

b. 357^{-1} mod 1234

c. 3125^{-1} mod 9987

10. 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 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 modulo p khi và chỉ khi $x^{(p-1)/p_i} \neq 1 \bmod p$ với $1 \leq i \leq n$.

Chương 2
HỆ MẬT MÃ KHÓA BÍ MẬT

2.1. Giới thiệu

Mật mã khoá bí mật hay còn gọi là mật mã đối xứng ra đời từ rất sớm. Từ khi máy tính chưa ra đời, mật mã khoá bí mật đã đóng vai trò quan trọng trong việc mã hoá thông tin. Mật mã khóa bí mật yêu cầu người gửi và người nhận phải thỏa thuận một khóa trước khi thông báo được gửi, và khóa này phải được cất giữ bí mật. Độ an toàn của thuật toán này vẫn phụ thuộc vào khóa, nếu để lộ ra khóa này nghĩa là bất kì người nào cũng có thể mã hóa và giải mã hệ thống mật mã.

Hình 2.1. Sơ đồ khối của hệ truyền tin mật

Tại nơi gửi (nguồn thông báo) có một bản rõ R được sinh ra. Để mã R cần có một khoá K. Nếu K được sinh tại nguồn thông báo thì nó phải được chuyển tới đích thông báo theo một kênh an toàn. Hoặc một bên thứ ba có thể sinh khoá và chuyển một cách an toàn tới cả nguồn và đích.

Với thông báo R và khoá mã K, thuật toán mã E sẽ tạo ra bản mã

$$M = E_K(R).$$

Tại nơi nhận (đích thông báo) với bản mã M và khoá mã K, thuật toán giải mã D sẽ tạo ra bản rõ $R = D_K(M)$.

Một kẻ tấn công thu được M nhưng không có khoá K, anh ta phải cố gắng khôi phục R hoặc khoá K. Thừa nhận rằng kẻ tấn công biết thuật toán mã E và thuật toán giải mã D. Nếu kẻ tấn công chỉ quan tâm đến nội dung thông báo, họ cố khôi phục R bằng việc sinh ra một ước lượng R' của R. Tuy nhiên thường kẻ tấn công mong muốn tìm ra khoá K để giải mã các thông báo tiếp theo, bằng cách sinh ra một khoá ước lượng K' của K. Độ bảo mật của mật mã khóa bí mật là thước đo mức độ khó khăn của việc tìm ra thông báo rõ hoặc khoá khi biết bản mã.

Ưu điểm của mật mã khóa bí mật là tốc độ mã hóa và giải mã nhanh. Tuy nhiên, mật mã khóa bí mật lại có khá nhiều nhược điểm:

- Trong phương pháp mã này, cả người mã hóa và người giải mã phải cùng chung một khóa mật. Khóa này phải được gửi đi trên kênh an toàn. Do đó nhất thiết phải duy trì một kênh an toàn để truyền khóa. Trên thực tế, công việc này rất khó khăn và tốn kém.

- Hệ mã hóa đối xứng không bảo vệ được sự an toàn nếu có xác suất cao khóa người gửi bị lộ.

- Vấn đề quản lý và phân phối khóa là khó khăn và phức tạp khi sử dụng. Người gửi và người nhận luôn luôn phải thống nhất với nhau về vấn đề khóa. Việc thay đổi khóa là rất khó và rất dễ bị lộ.

- Khuynh hướng cung cấp khóa dài mà nó phải được thay đổi thường xuyên cho mọi người trong khi vẫn duy trì cả tính an toàn lẫn hiệu quả, chi phí sẽ cản trở rất nhiều tới sự phát triển của hệ mật mã này.

- Nếu có N thực thể muốn liên lạc theo cặp thì cần N(N-1)/2 khoá bí mật cần truyền trên nhiều kênh an toàn. Số lượng này là không nhỏ để có thể quản lý và kiểm soát an toàn.

Trong mật mã khóa bí mật hay mật mã khóa đối xứng, chúng ta xét đến mật mã cổ điển, mã khối và mã dòng. Sau đây, chúng ta sẽ lần lượt tìm hiểu các loại mã này.

2.2. Mật mã cổ điển

Trong suốt một thời gian lịch sử dài từ thời cổ đại cho đến vài ba thập niên gần đây, các phương pháp mật mã được sử dụng trong thực tế đều là mật mã khóa đối xứng, từ hệ mật mã Ceasar đã được dùng hơn nghìn năm trước cho đến các hệ mật mã hiện đại ngày nay. Trong phần này, ta sẽ tìm hiểu một số hệ mật mã cổ điển và cách thám các hệ mã này.

2.2.1. Mã dịch chuyển

Để sử dụng mật mã dịch chuyển hay còn gọi là mã dịch vòng (MDV) cho văn bản tiếng Anh, người ta quy ước bảng 26 chữ cái tiếng Anh với các mã tương ứng 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

Quy tắc của MDV được cho trong hình dưới đây:

Giả sử $\mathcal{P} = \mathcal{C} = \mathcal{K} = Z_{26}$ 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

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 theo quy tắc mã sau:

$e_k(x) = x + 5 \bmod 26$

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 và 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) theo quy tắc giải mã sau:

$d_k(x) = y - 5 \bmod 26$

Và cuối cùng, biến đổi lại dãy số nhận được này thành các ký tự và thu được bản rõ ban đầu: *meetmeatsunset.*

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 khoád_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.

2.2.2. Mã thay thế

Mật mã này đã được sử dụng trong hàng trăm năm trước, trong đó: P = C = Z_{26}, vàK gồm tất cả các hoán vị trên tập 26 phần tử 0, 1, ... , 25.

Với mỗi $k \in K$: $k=\alpha_0\alpha_1 \ldots \alpha_{25}$ xác định phép thế sau:

$$\pi = \begin{pmatrix} 0 & 1 & \cdot & \cdot & 25 \\ \alpha_0 & \alpha_1 & \cdot & \cdot & \alpha_{25} \end{pmatrix}$$

và đặt

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

với π^{-1} là phép thế ngược của π.

Chú ý rằng, đối với mật mã thay thế, ta không cần sử dụng các phép tính nên có thể coi tập P như tập 26 chữ cái.

Ví dụ 2.2.

$$\pi = \begin{bmatrix} abcdefghij & klmnopqrst & uvwxyz \\ xnyahpogzq & wbtsflrcvm & uekjdi \end{bmatrix}$$

$e_\pi(a) = \pi(a) = x$; $e_\pi(b) = \pi(b) = n$ ...

$$\pi^{-1} = \begin{bmatrix} abcdefghij & klmnopqrst & uvwxyz \\ dlryyohezx & uptbgfjqmn & uskaci \end{bmatrix}$$

Giả sử ta có thông báo rõ:

hafno oij

áp dụng π, ta được thông báo mã:

GXPSF FZQ

Để giải mã, ta dùng π^{-1} (π^{-1} cũng là một phép thay thế).

2.2.3. Mã hoán vị

Ý tưởng của mật mã hoán vị là giữ các ký tự trên bản rõ không thay đổi, nhưng thay đổi vị trí của chúng bằng cách sắp xếp lại. Loại này cũng đã được sử dụng hàng trăm năm nay.

Cho m là số nguyên dương cố định, xác định $P = C = (Z_{26})^m$; K – tất cả các hoán vị của $\{1, 2, \ldots, m\}$.

Với mỗi $k \in K$: $k=\alpha_1\alpha_2\ldots\alpha_m$,xác định phép thế như sau:

$$\pi = \begin{pmatrix} 1 & 2 & \cdot & \cdot & m \\ \alpha_0 & \alpha_1 & \cdot & \cdot & \alpha_m \end{pmatrix}$$

Khi đó xác định:

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

$d_\pi(y_1, \dots, y_m) = \left(y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \dots, y_{\pi^{-1}(m)}\right)$

Với π^{-1} là phép thế ngược của π.

Ví dụ 2.3.

$m = 6$ và

$$\pi = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 1 & 6 & 4 & 2 \end{bmatrix}$$

Cần mã thông báo: hoofchisminh

Trước tiên, ta gom thành nhóm 6 phần tử;

hoofch isminh

$x_1 = h, x_2 = o, x_3 = o, x_4 = f, x_5 = c, x_6 = h$

Khi đó nhóm thứ nhất được mã thành

$x_3x_5x_1x_6x_4x_2$ = ochhfo

Tương tự, bản mã của nhóm thứ hai là: mnihis

Vậy thông báo mã toàn bộ là:

ochhfo mnihis

Ta có:

$$\pi^{-1} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 6 & 1 & 5 & 4 & 2 \end{bmatrix}$$

Áp dụng π^{-1} cho thông báo mã toàn bộ "ochhfo mnihis", sẽ thu lại được thông báo rõ ban đầu.

Thật ra, ta có thể mã nhanh hơn như sau: đặt dòng thứ hai của π^{-1} dưới bản rõ:

hoofch isminh

361524 361524

Sau đó ta sắp xếp lại trong từng nhóm theo thứ tự số tăng dần và được:

ochhfo mnihis

Bây giờ việc dịch chỉ đặt hàng thứ hai của π dưới bản mã:

ochhfo mnihis

351642 351642

rồi sắp theo thứ tự tăng dần:

hoofch isminh

2.2.4. 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:

$$ax \equiv y \pmod{26} \quad y \in Z_{26}$$

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 mod 26 không phải là một hàm đơn ánh và bởi vậy nó không thể là hàm mã hoá hợp lệ.

Ví dụ 2.4. Do UCLN(4,26)=2 nên 4x + 7 không là hàm mã hoá hợp lệ: x và 4x + 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 UCLN(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(mod 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 $ax \equiv b \pmod{m}$ *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 $a \in Z_m$ 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 2.1.

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. 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.

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:

$$7 \times 5 = 105 \equiv 1 \pmod{26} \text{ bởi vậy } 7^{-1} = 15$$

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 $\mathcal{P} = \mathcal{C} = Z_{26}$ 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 2.4.

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.

2.2.5. Mã Vigenère

Trong hai hệ mã dịch vòng và mã thay thế, 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 Vigenère nổi tiếng(Blaide Vigenere – Thế kỷ XVI).

Sử dụng phép tương ứng $A \leftrightarrow 0, B \leftrightarrow 1, \dots, Z \leftrightarrow 25$ 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.

Ta có thể mô tả mật mã Vigenère như sau:

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

Với khóa $k = (k_1, k_2, \dots, k_m)$ ta xác định được:

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

và

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

Trong đó tất cả các phép toán được thực hiện trong Z_{26}

Hình 2.4. Mã Vigenère

Ví dụ 2.5.

m = 6, từ khoá k = CIPHER = (2, 8, 15, 7, 4, 17)

Thông báo rõ: thiscryptosystemisnotsecure

Ta hãy chuyển thành số:

19	7	8	18	2	17	24	15	19	14	18
2	8	15	7	4	17	2	8	15	7	4
21	15	23	25	6	8	0	23	8	21	22

18	19	4	12	8	18	13	14	19	18	4	2	20	17	4
2	8	15	7	4	17	2	8	15	7	4	17	2	8	15
20	1	19	19	12	9	15	22	8	25	8	9	22	25	19

Lại đưa về chữ:

VPXZGI AXIVWP UBTTMJ PWIZIT WZT

Thông thường, người ta lập bảng để giúp cho việc mã hoá, giải mã được dễ dàng. Bảng này được gọi là bảng Vigenere (xem bảng dưới đây). Khi đó quá trình mã và dịch không cần số hoá nữa.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Hình 2.5. Bảng mã Vigenère

Lưu 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 Vigenère là26^m. Bởi vậy, thậm chí với *m* khá nhỏ, phương pháp tìm kiếm vét cạn cũng yêu cầu thời gian khá lớn. Ví dụ, với $m = 6$ thì không gian khoá cũng có kích thước lớn hơn 3.10^8 khoá.

2.2.6. 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 $\mathcal{P} = \mathcal{C} = (Z_{26})^m$. 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 $x = (x_1, x_2, \ldots, x_m) \in P$

và $k \in K$, ta tính $y = e_k(x) = (y_1, y_2, \ldots, y_m)$ như sau:

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

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 $1 \times m$ và $B = (b_{1,k})$ là một ma trận cấp m.n thì tích ma trận $AB = (c_{1,k})$ 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 A.B=B.A , 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 2x2 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 $BI_m = B$ 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

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

(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:

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

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

Bởivậy, 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)=detA x detB.

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 UCLN(detk, 26)=1.

Sau đây sẽ chứng minh ngắn gọn kết quả này.

Trước tiên, giả sử rằng UCLN(detk, 26)=1. Khi đó detk 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 detk 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 $\mathcal{P} = \mathcal{C} = (Z_{26})^m$ 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

2.2.7. Hệ mật mã 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 2.7.

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.

2.3. Mã dòng

Trong các hệ mật trên, các phần tử rõ được mã hoá bằng cách dùng *cùng một khoá*k:

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

Ở đây x_i có thể là một hoặc một dãy ký tự.

Hệ mật loại này được gọi là mật mã khối, hay đơn giản là mã khối.Bây giờ ta nghiên cứu mật mã dòng. Ý tưởng cơ bản là sinh dòng khoá:

$z = z_1\, z_2 \ldots$ và mã hóa dòng rõ $x = x_1\, x_2 \ldots$ theo cách:

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

Mật mã dòng hoạt động như sau:

+ Giả sử k là khoá và $x_1 x_2 ...$ là dòng rõ, f_i là hàm của k và i – 1 đặc trưng rõ:

$z_i = f_i(k, x_1, ..., x_{i+1})$, x_1được chọn trước bởi hai bên.

$y_i = e_{z_i}(x_i)$, i = 2, 3,...

Do đó để mã hóa dòng rõ $x_1 x_2 ...$, ta tính liên tiếp:

$z_1, y_1, z_2, y_2 ...$

Việc giải mã được làm tương tự:

$z_1, x_1, z_2, x_2 ...$

Nếu $z_i = k$ với mọi i, thì ta có thể nghĩ mật mã khối như trường hợp đặc biệt của mật mã dòng. Sau đây là một số trường hợp đặc biệt nhưng quan trọng của mật mã dòng:

- Mật mã đồng bộ: $z_i = f_i(k)$ i = 1, 2, ...

- Mật mã tuần hoàn với chu kỳ d: $z_{i+d} = z_i$, với mọi i >= 1

Mật mã dòng được chú ý nhiều là trường hợp $P = C = Z_2$. Khi đó phép mã hoá và giải mã là cộng theo modulo 2:

$e_z(x) = x + z \bmod 2$

$d_z(x) = y + z \bmod 2$

Khoá được sinh theo phương pháp ghi dịch phản hồi.

- Mật mã khoá tự động:

$P = C = K = Z_{26}$

$Z_1 = k$, $Z_i = x_{i-1}$ (i >= 2)

$e_z(x) = x + z \bmod 2$

$d_z(x) = y - z \bmod 2$; với $x, y \in Z_{26}$

Ví dụ:

K = 8, thông báo cần mã là: hairphongf

Trước tiên, chuyển thông báo rõ thành dãy số nguyên:

7 0 8 17 15 7 14 13 6 5

Dòng khoá như sau:

8 7 0 8 17 15 7 14 13 6 5

Cộng dãy khoá và dãy rõ theo qui tắc: $y_i = x_i + z_i \bmod 26$ i = 1, 2, ...

ta được:

15 7 8 25 6 22 21 1 19 11

và chuyển thành chữ:

p h i z g w v b t l

Với bản mã này và k = 8, ta giải mã như sau:

- Chuyển dãy mã thành số và trừ lần lượt

15	**7**	**8**	**25**	**6**	**22**	**21**	**1**	**19**	**11**
8	**7**	**0**	**8**	**17**	**15**	**7**	**14**	**13**	**6**
7	**0**	**8**	**17**	**15**	**7**	**14**	**13**	**6**	**3**

- Chuyển dãy số thành dãy chữ: h a i r p h o n g f

2.4. Mã khối

2.4.1. Giới thiệu chung

Các hệ mật khóa bí mật thường được chia thành các hệ mã khối và hệ mã dòng. Đối với mã khối bản rõ có dạng các khối "lớn" (chẳng hạn 128-bit) và dãy các khối đều được mã bởi cùng một hàm mã hóa, tức là bộ mã hóa là một hàm không nhớ. Trong mã dòng, bản rõ thường là dãy các khối "nhỏ" (thường là 1-bit) và được biến đổi bởi một bộ mã hóa có nhớ.

Các hệ mã khối có ưu điểm là chúng có thể được chuẩn hóa một cách dễ dàng, bởi vì các đơn vị xử lý thông tin hiện này thường có dạng block như bytes hoặc words.

Nhược điểm lớn nhất của mã khối là phép mã hóa không che giấu được các mẫu dữ liệu: các khối mã giống nhau sẽ suy ra các khối rõ cũng giống nhau. Tuy nhiên nhược điểm này có thể được khắc phục bằng cách

đưa vào một lượng nhỏ có nhớ trong quá trình mã hóa, tức là bằng cách sử dụng cách thức móc xích khối mã (CBC-Cipher Block Channing mode) trong đó hàm mã hóa không nhớ được áp vào tổng XOR của block rõ và block mã trước đó. Phép mã lúc này có kiểu cách kỹ thuật như mã dòng áp dụng đối với các khối "lớn".

2.4.2. Các khái niệm cơ bản

2.4.2.1. Các định nghĩa về mã khối và độ an toàn của mã khối

Giả sử F_2 là trường Galois hai phần tử. Ký hiệu F_2^m là không gian véc tơ các bộ m-khối các phần tử của F_2. Trong phần này chúng ta giả thiết không mất tổng quát rằng, bản rõ X, bản mã Y lấy các giá trị trong không gian véc tơ F_2^m, còn khóa Z lấy giá trị trong không gian véc tơ F_2^k. Như vậy m- là độ dài bít của các khối rõ và mã, còn k- là độ dài bit của khóa bí mật.

Định nghĩa 2.4. *Hệ mã khối khóa bí mật là một ánh xạ $E: F_2^m \times S_z \rightarrow F_2^m$, sao cho với mỗi $z \in S_z$, $E(., z)$ là một ánh xạ có ngược từ F_2^m vào F_2^m.*

Hàm có ngược E(., z) được gọi là hàm mã hóa tương ứng với khóa z. Ánh xạ nghịch đảo của E(., z) được gọi là hàm giải mã tương ứng với khóa z và sẽ được ký hiệu là D(., z). Chúng ta viết Y = E(X, Z) đối với một mã khối có nghĩa là bản mã Y được xác định bởi bản rõ X và khóa bí mật Z theo ánh xạ E. Tham số m được gọi là độ dài khối còn tham số k được gọi là độ dài khóa của hệ mã khối đó. Cỡ khóa đúng của hệ mã khối được xác định bởi số $k_t = \log_2 (\#(S_z))$ bit. Như vậy độ dài khóa sẽ bằng cỡ khóa đúng nếu và chỉ nếu $S_z = F_2^k$, tức là mọi bộ k-bit nhị phân đều là một khóa có hiệu lực. Chẳng hạn đối với chuẩn mã dữ liệu DES, độ dài khóa là k = 64 bit, trong khi cỡ khóa đúng của nó là k_t = 56 bit. Chú ý rằng ở đây ta xem xét các mã khối có độ dài khối mã bằng độ dài khối rõ.

Độ an toàn của các hệ mã khối

Như đã nói ở trên, một mã khối được sử dụng nhằm bảo vệ chống sự rò rỉ không mong muốn của bản rõ. Nhiệm vụ của thám mã đối phương là phá hệ mã này theo nghĩa anh ta có thể mở ra được các bản rõ từ các

bản mã chặn bắt được. *Một hệ mã là bị phá hoàn toàn nếu như thám mã có thể xác định được khóa bí mật đang sử dụng và từ đó anh ta có thể đọc được tất cả các thông báo một cách dễ dàng như là một người dùng hợp pháp. Một hệ mã là bị phá thực tế nếu thám mã có thể thường xuyên mở ra được các bản rõ từ các bản mã nhận được, nhưng vẫn chưa tìm ra được khóa.*

Độ an toàn luôn gắn với các đe dọa tấn công. Như đã nói ở trên, chúng ta giả sử rằng kẻ tấn công luôn có thể truy nhập tới mọi thứ được truyền thông qua kênh không an toàn. Tuy nhiên, có thể có các thông tin khác đối với thám mã. Khả năng tính toán của thám mã phải luôn được xem xét trước khi xem xét độ an toàn của một mã có thể bị truy nhập.

Độ an toàn tính toán

Trong thực tế không kẻ tấn công nào có khả năng tính toán vô hạn. Độ an toàn của một hệ mật thực tế phụ thuộc vào tính không thể phá hệ mã đó về mặt lý thuyết mà đúng hơn là phụ thuộc độ khó thực tế của các tấn công. Một hệ mật được gọi là an toàn tính toán nếu độ khó của tấn công tối ưu vượt quá khả năng tính toán của thám mã. Shannon đã mô tả độ khó của tấn công như thế (tấn công chỉ biết bản mã) bởi đặc trưng W(n) xem như là khối lượng công việc đòi hỏi để xác định khóa khi n-bản mã là được biết. Ta cũng có thể xem xét W(n) đối với các kiểu tấn công khác. Trong suốt phần này, chúng ta sử dụng từ "*độ phức tạp*" để mô tả độ khó như thế. Độ phức tạp của một tấn công hiểu một cách chung chung là số trung bình các phép toán (thao tác) dùng trong tấn công đó. Chú ý rằng một hệ mã là an toàn tính toán có nghĩa là độ phức tạp của tấn công tối ưu vượt quá khả năng tính toán của thám mã đối phương. Để chứng minh một hệ mật là an toàn tính toán cần phải chỉ ra được cận dưới hữu ích về độ phức tạp của việc giải quyết một bài toán tính toán nào đó. Hiện tại, điều này là không thể đối với tất cả các bài toán tính toán.

Do vậy, trong thực tế, việc đánh giá độ an toàn của một hệ mật phụ thuộc vào độ phức tạp của tấn công tốt nhất cho tới hiện tại. Một mã

khối thực tế được xem là an toàn tính toán nếu không có tấn công đã biết nào có thể làm tốt hơn so với tấn công vét cạn khóa. Trong tấn công vét cạn khóa chỉ biết bản mã trên một mã khối, mỗi một khóa có thể đều được thử để giải mã của một hoặc hiều hơn các khối mã chặn bắt được cho tới khi nào một khóa cho kết quả khối rõ có thể đọc được. Độ phức tạp của tấn công này, xem như là số các phép giải mã thử, về mặt trung bình sẽ bằng 2^{k_t-1} đối với một hệ mã khối có cỡ khóa đúng là k_t. Tấn công vét cạn khóa là một tấn công "brute-force" nó có thể áp vào hệ mã khối bất kỳ. Như vậy một hệ mã khối muốn an toàn thì cỡ khóa đúng của nó là phải đủ lớn để tạo cho tấn công vét cạn khóa là không thể thực hiện được.

Độ phức tạp xử lý và độ phức tạp dữ liệu của một tấn công cụ thể

Độ phức tạp của một tấn công được chia ra làm hai phần: *độ phức tạp dữ liệu và độ phức tạp xử lý*. Độ phức tạp dữ liệu là lượng dữ liệu đầu vào cần cho tấn công đó trong khi độ phức tạp xử lý là lượng các tính toán cần để xử lý dữ liệu như thế. Thành phần dominant - trội hơn thường được mô tả như là độ phức tạp của tấn công này. Chẳng hạn, trong tấn công vét cạn khóa, lượng dữ liệu đầu vào cần cho tấn công này là số các khối mã chặn bắt được (hoặc số các cặp rõ/mã trong tấn công bản rõ đã biết), nói chung đó là một số lượng rất nhỏ so với số các phép toán (trung bình cần 2^{k_t-1} phép giải mã với các khóa khác nhau trong việc tìm ra khóa đúng) cần thiết của tấn công này. Do vậy độ phức tạp của tấn công duyệt khóa thường chính là độ phức tạp xử lý. Ví dụ khác là tấn công vi sai của Biham và Shamir, đó là kiểu tấn công bản rõ lựa chọn. Đối với tấn công vi sai độ phức tạp vượt trội lên bởi số các cặp rõ/mã cần trong tấn công đó, trong khi số các tính toán sử dụng trong tấn công này lại tương đối nhỏ. Do đó độ phức tạp của tấn công vi sai thực chất là độ phức tạp dữ liệu.

Nói chung đối với một mã khối độ dài khối m-bit và cỡ khóa đúng là k_t-bit, độ phức tạp dữ liệu của tấn công bản rõ đã biết (hoặc bản rõ lựa chọn) có thể được đo bởi số các cặp rõ/mã đã biết (hay lựa chọn) cần cho tấn công này, nhiều nhất là 2^m là số toàn bộ các cặp như thế đối với một

khóa cố định. Độ phức tạp xử lý có thể bị chặn trên bởi số 2^{k_t}phép mã hóa do đặc tính của tấn công vét cạn khóa và do nói chung thao tác mã hóa là được tính toán nhanh, hiệu quả. Như vậy chúng ta có thể nói rằng một hệ mật là an toàn tính toán nếu như không có tấn công nào trên hệ mật đó có độ phức tạp dữ liệu nhỏ hơn đáng kể 2^m phép mã và độ phức tạp xử lý nhỏ hơn đáng kể 2^{k_t} phép mã hóa. Một hệ mật được gọi là an toàn thực tế chống lại một tấn công cụ thể nếu với tấn công này, độ phức tạp dữ liệu vào khoảng 2^m cặp rõ/mã hoặc độ phức tạp xử lý là vào khoảng 2^{k_t} phép mã hóa. Đối với thám mã, độ phức tạp dữ liệu là loại độ phức tạp bị động, anh ta phải chờ người sử dụng tạo ra các cặp rõ /mã cho anh ta. Mặt khác, độ phức tạp xử lý lại là kiểu độ phức tạp chủ động và có thể khắc phục nói chung bằng cách sử dụng nhiều máy tính mạnh.

2.4.2.2. Các tham số của mã khối

Độ dài khối m

Để một hệ mã khối là an toàn, độ dài khối m của nó phải đủ lớn ngăn cản các tấn công phân tích thống kê, tức là để không cho đối phương thu được thông tin có ích nào về khối rõ nào đó thường xuất hiện nhiều hơn các khối rõ khác. Ngoài ra độ dài khối m cũng phải được chọn sao cho số các cặp rõ/mã mà đối phương có thể thu nhận được trong thực tế phải nhỏ hơn rất nhiều so với 2^m.

Khi độ dài khối của hệ mã trở nên lớn thì độ phức tạp của ứng dụng cũng tăng theo. Dù rằng độ phức tạp trong ứng dụng chọn ngẫu nhiên hàm có ngược là tăng theo cỡ mũ so với độ dài khối, nhưng chỉ có hàm đơn giản mới xuất hiện ngẫu nhiên, điều này tạo cơ hội phục vụ hàm mã hóa thực tế khi độ dài khối m là lớn. Tuy nhiên, Shannon đã chỉ ra rằng sự dễ dàng trong tính toán các hàm mã hóa E(., z) và hàm giải mã D(., z) với mọi z không suy ra được việc giải tìm khóa z từ các phương trình y = E(x, z) và x = D(y, z) sẽ là dễ dàng khi biết x và y.

Độ dài khóa k và cỡ khóa đúng k_t

Để hệ mã khối an toàn chống lại tấn công vét cạn khóa, cỡ khóa đúng cần phải đủ lớn sao cho 2^{k_t-1} phép mã hóa cần cho tấn công này là

vượt xa khả năng của thám mã. Mặt khác, độ dài khóa k cũng cần nhỏ ở mức nào đó sao cho việc tạo, phân phối và lưu trữ khóa có thể thực hiện được hiệu quả và an toàn. Chẳng hạn, DES có độ dài khóa là 64 bit, còn cỡ khóa đúng là 56 bit. Tấn công vét cạn khóa là không thể nhưng cũng không là quá xa vời. Nhiều gợi ý muốn tăng cỡ khóa đúng của DES. Chẳng hạn, mở rộng cỡ khóa đúng của DES tới 128 bit bằng phép mã bội ba dùng hai khóa xem là một cách thức chuẩn để sử dụng DES.

2.4.3. Các chế độ hoạt động của mã khối (Modes of operation)

Đã có nhiều kiểu hoạt động khác nhau cho mã khối. Những kiểu hoạt động này nói chung cung cấp một vài tính chất mong muốn tới các khối mã, như thêm tính ngẫu nhiên vào thuật toán mã khối, đệm các thông báo rõ cho được độ dài tùy ý (sao cho độ dài bản mã không có liên quan với độ dài bản rõ tương ứng), điều khiển sự lan truyền sai số, sinh khóa dòng cho mật mã dòng,v.v...

Đối với chuẩn mã dữ liệu DES, người ta đã sử dụng bốn kiểu hoạt động của nó là kiểu từ điển điện tử (electronic codebook-ECB), kiểu xích khối mã (cipher block chaining-CBC), kiểu phản hồi đầu ra (output feedback-OFB), kiểu phản hồi mã (cipher feedback- CFB).

Sau khi AES ra đời, NIST định nghĩa 5 kiểu hoạt động của nó. Ngoài 4 kiểu trước đó là ECB, CBC, CFB và OFB còn thêm kiểu đếm (CTR - được đề xuất năm 1979 bởi Diffie và Hellman). Trong tương lai, NIST dự định công bố phiên bản mới trong đó phạm vi của kiểu CBC được mở rộng để bao gồm cả những bản rõ có độ dài bit không phải là bội của độ dài khối.

Bốn kiểu chung nhất là ECB, CBC, CFB và OFB. Chúng được tổng kết và được bàn luận ở dưới.

Ở dưới đây, E_K ký hiệu hàm mã của mã khối E được tham số hoá bởi khoá K, còn E_K^{-1} (hoặc D_K) ký hiệu phép giải mã. Thông báo rõ $x = x_1 \ldots x_t$ được giả thiết là chứa các khối n-bit cho các kiểu ECB và CBC và các khối r-bit cho các kiểu CFB và OFB với $r \leq n$ cố định thích hợp.

Sau đây, chúng ta sẽ trình bày 4 kiểu hoạt động cơ bản của mã khối là ECB, CBC, CFB, OFB.

(c)CFB, các ký tự r-bit, phản hồi r-bit

(d)OFB, các ký tự *r*-bit phản hồi *n*-bit

Hình 2.7. Bốn kiểu hoạtđộng của mã khối

2.4.3.1. Kiểu từ điển điện tử (ECB)

Kiểu hoạt động ECB được mô tả ở thuật toán sau

Mã hóa:

+ Đầu vào: khoá K có *k-bit*; các khối bản rõ $x_1, .., x_t$, mỗi khối có n bit.

+ Đầu ra: Các khối mã, mỗi khối có *n* bit

$$c_j \leftarrow E_K(x_j), \text{ với } 1 \le j \le t.$$

Giải mã:

+ Đầu vào: khoá K có *k-bit*; các khối bản mã $c_1, .., c_t$, mỗi khối có *n* bit.

+ Đầu ra: Các khối rõ, mỗi khối có *n* bit

$$x_j \leftarrow E_K^{-1}(c_j), \text{ với } 1 \le j \le t.$$

Các tính chất

+ Các khối rõ giống nhau (dưới cùng một khoá) mang lại cùng một bản mã.

+ Các phụ thuộc móc xích: các khối được mã một cách độc lập với các khối khác. Việc thay đổi thứ tự các khối mã dẫn đến việc thay đổi thứ tự tương ứng của các khối rõ.

+ Lan sai: một hay nhiều lỗi bit trong duy nhất một khối mã chỉ ảnh hưởng đến kết quả giải mã của chỉ một khối. Đối với các mã pháp E điển hình, kết quả giải mã của một khối như vậy là ngẫu nhiên (với khoảng 50% các bit bản rõ được giải mã là bị lỗi).

Chú ý

Về sử dụng của kiểu ECB: Vì các khối bản mã là độc lập, việc thay thế ác ý của các khối ECB (ví dụ, chèn vào một khối) không ảnh hưởng đến việc giải mã của các khối lân cận. Hơn nữa, các mã khối không giấu được các kiểu mẫu dữ liệu - các khối bản mã giống nhau kéo theo các khối bản rõ giống nhau. Vì lý do này, kiểu ECB không được khuyến cáo cho những thông báo dài hơn 1 khối, hoặc khi các khoá được sử dụng lại cho nhiều hơn 1 thông báo duy nhất có 1 khối. Độ an toàn có thể được cải tiến bằng cách đưa vào các bit đệm ngẫu nhiên trong mỗi khối.

2.4.3.2. Kiểu xích khối mã (CBC)

Kiểu hoạt động xích khối mã được chỉ ra trong thuật toán sau, nó sử dụng một vectơ khởi tạo *n-bit* được ký hiệu là IV.

Mã hóa:

+ Đầu vào: khoá K có k-bit; các khối bản rõ $x_1, .., x_t$, mỗi khối có *n* bit.

+ Đầu ra: Các khối mã, mỗi khối có *n* bit

$$c_0 \leftarrow IV,\ c_j \leftarrow E_K(c_{j-1} \oplus x_j), \text{ với } 1 \le j \le t$$

Giải mã:

+ Đầu vào: khoá *K* có *k-bit*; các khối bản mã $c_1, .., c_t$, mỗi khối có *n* bit.

+ Đầu ra: Các khối rõ, mỗi khối có *n* bit

$$c_0 \leftarrow IV.\ x_j \leftarrow c_{j-1} \oplus E_K^{-1}(c_j), \text{ với } 1 \le j \le t$$

Các tính chất

+ Với các bản rõ giống nhau: các khối bản mã giống nhau thu được khi cùng một bản rõ được mã dưới cùng một khoá và IV. Việc thay đổi IV, khoá hoặc khối rõ thứ nhất mang lại các bản mã khác nhau.

+ Những phụ thuộc móc xích: cơ chế móc xích gây cho bản mã c_j phụ thuộc vào x_j và tất cả các khối rõ đứng trước Do đó, việc đổi chỗ của các khối mã ảnh hưởng tới việc giải mã. Việc giải mã đúng của một khối bản mã đúng đòi hỏi khối bản mã phía trước đúng.

+ Lan sai: lỗi một bit duy nhất trong khối mã c_j ảnh hưởng tới việc giải mã của các khối c_j và c_{j+1} (vì x_j phụ thuộc vào c_j và c_{j-1}). Khối x'_j được khôi phục từ c_j thông thường là ngẫu nhiên toàn bộ, trong khi bản rõ được khôi phục x'_{j+1} có các lỗi bit chính xác như c_j có. Tức là kẻ thù địch có thể gây ra các thay đổi bit dự đoán được trong x_{j+1} bằng cách thay đổi các bit tương ứng của c_j.

+ Khắc phục lỗi: kiểu CBC là tự đồng bộ theo nghĩa là nếu lỗi (bao gồm việc bị mất một hay nhiều khối nguyên) xảy ra trong khối c_j nhưng không xảy ra ở c_{j+1} thì c_{j+2} vẫn được giải mã một cách chính xác thành x_{j+2}.

Chú ý

Lan sai trong khi mã: những sửa đổi đối với khối rõ x_j trong quá trình mã thay đổi toàn bộ các khối bản mã sau đó. Điều này ảnh hưởng tới tính hữu dụng của các kiểu móc xích đối với các ứng dụng đòi hỏi truy cập đọc/ghi ngẫu nhiên tới dữ liệu đã được mã. Kiểu ECB là cái thay thế.

Tự đồng bộ và các lỗi khung: Mặc dù có khả năng tự đồng bộ theo nghĩa khôi phục lại được từ các lỗi bit, việc khôi phục từ các bit "bị mất" gây ra các lỗi trong các ranh giới khối (các lỗi toàn vẹn khung) là không thể trong kiểu CBC và các kiểu khác.

Tính toàn vẹn của IV trong CBC: Trong khi IV trong kiểu CBC không cần phải bí mật, tính toàn vẹn của nó cần phải được bảo vệ, vì sửa đổi ác ý cho phép kẻ thù địch làm các thay đổi bit dự đoán được đối với khối bản rõ đầu tiên được giải mã. Việc sử dụng IV bí mật là một phương pháp để ngăn chặn điều này. Tuy nhiên, nếu tính toàn vẹn của thông báo được yêu cầu, một kỹ thuật thích hợp cần được sử dụng; các kỹ thuật mã thông thường chỉ đảm bảo tính bí mật.

Kiểu CBC giúp ngẫu nhiên hóa bản mã. Tuy nhiên, để bảo đảm an toàn cần phải luôn luôn sử dụng IV ngẫu nhiên cho mỗi lần mã hóa. Nó phải kết hợp với một số kỹ thuật mật mã nữa mới có thể giúp bảo vệ tính trung thực dữ liệu.

2.4.3.3. Kiểu phản hồi mã (CFB)

Kiểu CBC xử lý cùng một lúc *n* bit của bản rõ (sử dụng mã khối *n-bit*), nhưng một số ứng dụng lại đòi hỏi rằng các đơn vị *r-bit* của bản rõ được mã và được truyền đi không có độ trễ, với r nào đó nhỏ hơn *n* (thông thường r = 1 hoặc r = 8). Trong trường hợp này, kiểu phản hồi mã (CFB) có thể được sử dụng như được chỉ ra ở thuật toán sau.

Thuật toán (kiểu hoạt động CFB-r)

Mã hóa:

+ Đầu vào: khoá K có *k-bit*; *n-bit* IV; các khối bản rõ $x_1, .., x_u$, mỗi khối có *r* bit $(1 \le r \le n)$.

+ Đầu ra: Các khối mã, mỗi khối có *r* bit

$I_j \leftarrow$ IV; (I_j là giá trị đầu vào trong thanh ghi dịch). Với $1 \le j \le u$:

$O_j \leftarrow E_K(I_j)$. (Tính đầu ra của mã khối)

$t_j \leftarrow r$ bit bên trái nhất của O_j (giả sử bit bên trái nhất được định danh như là bit thứ nhất)

$c_j \leftarrow x_j \oplus t_j$. (Tạo khối mã *r-bit* c_j)

$I_{j+1} \leftarrow 2^r \cdot I_j + c_j \bmod 2^n$. (Dịch c_j về đầu bên phải của thanh ghi)

Giải mã:

+ Đầu vào: khoá K có *k-bit*; n-bit IV; các khối bản mã $c_1, .., c_u$, mỗi khối có *r* bit.

+ Đầu ra: Các khối rõ, mỗi khối có *r* bit

$I_1 \leftarrow$ IV. Với $1 \le j \le u$ khi nhận được c_j:

$x_j \leftarrow c_j \oplus t_j$ trong đó t_j, O_j và I_j được tính như ở trên.

Các tính chất

+ Các bản rõ giống nhau: cũng giống như phép mã CBC, việc thay đổi IV làm cho cùng một bản rõ đầu vào sẽ được mã thành đầu ra khác nhau. IV không cần phải bí mật (mặc dù IV không dự đoán được có thể được mong muốn trong một số ứng dụng).

+ Độ phụ thuộc móc xích: tương tự với phép mã CBC, kỹ thuật móc xích làm cho khối mã c_j phụ thuộc cả vào x_j và các khối rõ phía trước; do đó việc thay đổi thứ tự của các khối mã ảnh hưởng đến việc giải mã. Việc giải mã đúng của khối bản mã đúng đòi hỏi $\lceil n/r \rceil$ khối bản mã phía trước là đúng (kết quả là thanh ghi chứa giá trị đúng).

+ Lan sai: các lỗi 1 hay nhiều bit trong một khối bản mã duy nhất r-bit c_j ảnh hưởng tới việc giải mã của khối đó và $\lceil n/r \rceil$ khối bản mã sau đó (tức là, cho đến khi n bit của bản mã được xử lý, sau đó khối bị lỗi c_j được dịch hoàn toàn ra khỏi thanh ghi). Bản rõ được khôi phục x'_j sẽ khác với x_j một cách chính xác tại các vị trí bit mà c_j đã bị lỗi; các khối bản rõ được khôi phục một cách không đúng khác thông thường sẽ là các vectơ ngẫu nhiên. Cho nên kẻ thù địch có thể gây ra các thay đổi bit dự đoán được trong x_j bằng cách thay đổi các bit tương ứng của c_j.

+ Khôi phục sau lỗi: kiểu CFB là tự đồng bộ tương tự với CBC, nhưng đòi hỏi $\lceil n/r \rceil$ khối bản mã để khôi phục được.

+ Thông lượng: với $r < n$, thông lượng bị giảm đi bởi tỷ lệ r/n so với kiểu CBC do mỗi lần thực hiện của E chỉ dẫn tới r bit của bản mã đầu ra.

Chú ý

CFB chỉ sử dụng phép mã E: vì hàm mã E được sử dụng cho cả phép mã và giải mã CFB, nên không được sử dụng kiểu CFB nếu mã khối E là thuật toán khoá công khai; thay vào đó, cần dùng kiểu CBC.

Phiên bản ISO/IEC của CFB: kiểu CFB của thuật toán trên có thể được sửa đổi như sau để cho phép xử lý các khối bản rõ (các ký tự) mà độ dài bit s của nó là nhỏ hơn so với độ dài bit r của biến phản hồi (ví dụ, các ký tự 7-bit sử dụng phản hồi 8-bit; $s < r$). s bit bên trái nhất của O_j

(chứ không phải r bit) được gán cho t_j; ký tự bản mã s-$bitc_j$ được tính; biến phản hồi được tính từ c_j bằng cách gán vào phía trước (về bên trái) r-s bit 1; biến phản hồi thu được r-bit được dịch về phía có ý nghĩa ít nhất của thanh ghi như ở trên. Chú ý rằng trong mô tả của ISO/IEC 10116 còn có một tham số nữa, đó là kích thước của bộ đệm phản hồi (lớn hơn hoặc bằng độ dài khối và nhỏ hơn hoặc bằng 2 lần độ dài khối).

2.4.3.4. Kiểu phản hồi đầu ra (OFB)

Kiểu phản hồi đầu ra OFB có thể được sử dụng cho những ứng dụng trong đó tất cả lan sai cần phải tránh. Nó tương tự như CFB, và cho phép mã hoá các kích thước khối khác nhau (các ký tự), nhưng khác ở chỗ đầu ra của hàm khối mã E (chứ không phải bản mã) được phản hồi

Có hai phiên bản của OFB sử dụng mã khối n-bit như sau

Thuật toán 1 (kiểu OFB với phản hồi đầy đủ)

+ Đầu vào: khoá K có *k-bit*; n-bit IV; các khối bản rõ *r-bit* $x_1, ..., x_u$ $(1 \le r \le n)$

+ Đầu ra: Các khối mã $c_1, ..., c_u$, mỗi khối có n bit

$I_1 \leftarrow$ IV. Với $1 \le j \le u$:

$O_j \leftarrow E_K(I_j)$. (Tính đầu ra của mã khối)

$t_j \leftarrow r$ bit bên trái nhất của O_j (giả sử bit bên trái nhất được định danh như là bit thứ nhất)

$c_j \leftarrow x_j \oplus t_j$. (Truyền khối mã r-bit c_j)

$I_{j+1} \leftarrow O_j$ (Cập nhật đầu vào của mã khối cho khối tiếp sau)

Giải mã:

+ Đầu vào: khoá K có k-bit; n-bit IV; các khối bản mã $c_1, .., c_u$, mỗi khối có r bit.

+ Đầu ra: Các khối rõ, mỗi khối có r bit

$I_1 \leftarrow$ IV. Với $1 \le j \le u$:

$x_j \leftarrow c_j \oplus t_j$ trong đó t_j, O_j và I_j được tính như ở trên

Thuật toán 2 (kiểu OFB với phản hồi r-bit)

+ Đầu vào: khoá K có k-bit; n-bit IV; các khối bản rõ r-*bit* $x_1, ..., x_u$ $(1 \leq r \leq n)$

+ Đầu ra: các khối bản mã $c_1, ..., c_u$;

Giống như Thuật toán 1 ở trên, nhưng bước d) "$I_{j+1} \leftarrow O_j$" được thay bởi: $I_{j+1} \leftarrow 2^r \cdot I_j + t_j \bmod 2^n$ (dịch đầu ra t_j về đầu cuối bên phải của thanh ghi) .

Các tính chất

+ Các bản rõ giống nhau: cũng như các kiểu CBC và CFB, việc thay đổi IV làm cho cùng một bản rõ được mã thành đầu ra khác nhau.

+ Những phụ thuộc móc xích: dòng khoá là không phụ thuộc vào bản rõ.

+ Lan sai: một hay nhiều lỗi bit trong ký tự bản mã bất kỳ c_j ảnh hưởng tới việc giải mã chỉ của ký tự đó, tại chính xác (những) vị trí bit mà c_j có lỗi, làm cho (các) bit bản rõ được khôi phục tương ứng bị đảo ngược.

+ Khôi phục sau lỗi: kiểu OFB khôi phục lại được từ các lỗi bit bản mã, nhưng không thể tự đồng bộ sau khi mất các bit bản mã, nó tiêu huỷ sự sắp đúng hàng của dòng khoá và dòng mã khi giải mã.

+ Thông lượng: với $r < n$, thông lượng là giảm so với kiểu CFB. Tuy nhiên, trong tất cả các trường hợp, vì dòng khoá là độc lập với bản rõ hoặc bản mã, nó có thể được tính lại (khi đã cho khoá và IV).

Chú ý

Thay đổi IV trong OFB: đại lượng IV, vốn không cần phải bí mật, nhưng cần phải được thay đổi nếu khoá K của OFB được dùng lại. Trong trường hợp ngược lại, sẽ thu được các dòng khoá giống nhau và bằng cách XOR các bản mã tương ứng, kẻ thù địch có thể đưa việc thám mã về việc thám một mã pháp có khoá chạy với một bản rõ như khoá chạy.

OFB chỉ sử dụng phép mã E.

Kiểu bộ đếm: Một đơn giản hoá của OFB là việc cập nhật các khối đầu vào như một bộ đếm, $I_{j+1} = I_j + 1$, thay cho việc sử dụng phản hồi. Điều này tránh được vấn đề chu kỳ ngắn, đồng thời cho phép khôi phục lại khi bị lỗi trong việc tính toán E. Hơn thế nữa, nó cung cấp tính chất truy cập ngẫu nhiên: khối bản mã thứ i không cần phải được giải mã thì mới giải mã được khối thứ *i+1*.

Các kiểu như mã dòng: Rõ ràng là cả hai kiểu OFB với phản hồi đầy đủ và kiểu bộ đếm (vừa nói tới ở trên) khai thác mã khối như một bộ tạo khoá dòng cho mã dòng. Một cách tương tự, kiểu CFB mã dòng ký tự nhờ mã khối như một bộ tạo dòng khoá (phụ thuộc vào bản rõ). Kiểu CBC cũng có thể được xem như một mã dòng với các khối *n-bit* đóng vai của ký tự rất lớn. Những kiểu hoạt động này cho phép người ta định nghĩa các mã dòng từ các mã khối.

2.4.3.5. Kiểu đếm (CTR)

Kiểu CTR có đặc điểm là gắn thuật toán mã khối cơ sở với giá trị đếm và giá trị đếm này bắt đầu từ giá trị khởi tạo. Với việc bộ đếm tăng lên, thuật toán mã khối cơ sở cho ra các khối liên tiếp để lập nên một dòng bít. Dòng bít này được dùng như dòng khóa của mật mã Vernam; nghĩa là, dòng khóa được cộng mô đun 2 (XOR) theo từng bít với các khối rõ. Kiểu CTR hoạt động như sau

Mã hóa:

+ Đầu vào: Khóa K, Ctr_1 và các khối bản rõ $x_1, .., x_m$, mỗi khối có n bit

+ Đầu ra: Ctr_1 và các khối mã $c_1, c_2, ..., c_m$

$$c_i = x_i \oplus E_K(Ctr_i),\ Ctr_{i+1} = E_K(Ctr_i),\ 1 \le i \le m$$

Giải mã:

+ Đầu vào: Khóa K, Ctr_1 và các khối bản mã $c_1, c_2, ..., c_m$, mỗi khối có *n* bit

+ Đầu ra: Ctr_1 và các khối rõ $x_1, x_2, ..., x_m$

$$x_i = c_i \oplus E_K(Ctr_i),\ Ctr_{i+1} = E_K(Ctr_i),\ 1 \le i \le m$$

Các tính chất

Do không có phản hồi nên phép giải mã cũng như phép mã hóa kiểu CTR có thể được thực hiện song song, do đó nhanh. Đây là ưu thế của kiểu CTR so với các kiểu CFB, OFB.

Các tính chất của kiểu CTR có thể được phát biểu tương tự như tính chất của các kiểu CFB, OFB

Đối với kiểu CTR, việc thực thi là đơn giản

Đối với mã khối, bản rõ có dạng các khối "lớn" (chẳng hạn 128 bit hoặc 256 bit), trong mã dòng, bản rõ là dãy các khối "nhỏ" (thường là 1-bit).

2.4.4. Chuẩn mã dữ liệu (DES)

2.4.4.1. Giới thiệu

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.

2.4.4.2. 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 2.8.

Hình 2.8. 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

$$B_1B_2B_3B_4B_5B_6B_7B_8$$

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 $B_i = b_1b_2b_3b_4b_5b_6b_7b_8$), 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 2.9. Hàm f của DES

Hàm f được mô tả trong hình 2.9. 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 $IP(x)$; bit thứ 50 của x là bit thứ hai của $IP(x)$, .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 định 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:

$$PC\text{-}1(k) = C_0 D_0$$

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 2.10

Hình 2.10. Tính bảng khóa DES

Các phép dịch vòng trái L_i:

Chỉ số *i*	1,2	3÷8	9	10÷15	16
Số bit dịch vòng trái	1	2	1	2	1

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.

2.4.4.3. 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
11100001010101011110000101010101010

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

$E(R_0)$ = 011110100001010101010101011110100001010101010101

K_1 =
000110110000001011101111111111000111000001110010 $E(R_0) \oplus K_1$ = 011000010001011110111010100001100110010100100111

S-box outputs 010111001000001010110101100101 11

$f(R_0,K_1)$ = 00100011010010101010100110111011

$L_2 = R_1$ = 11101111010010100110010101000100

$E(R_1)$= 011101011110101001010100001100001010101000001001

K_2= 011110011010111011011001110110111100100111100101

$E(R_1) \oplus K_2$ =
000011000100010010001101111010110110001111101100

S-box outputs 111110001101000000111010101011 10

$f(R_1,K_2)$ = 00111100101010111000011110100011

$L_3 = R_2$ = 11001100000000001011101110000 1001

$E(R_2)$ =
111001011000000000000010101110101110100001010011

K_3 = 010101011111110010001010010000101100111110011001

$E(R_2) \oplus K_3$ =
101100000111110010001000111110000010011111001010

S-box outputs 001001110001000011100001011011 11

$f(R_2,K_3)$ = 01001101000101100110111010110000

$L_4 = R_3$ = 10100010010111000000101111110100

$E(R_3) =$
010100000100001011111000000001010111111111010100

$K_4 =$ 011100101010110111010110110110110011010100011101

$E(R_3) \oplus K_4 =$
001000101110111100101110110111100100101010110100

S-box outputs 001000011110110110011111100111010

$f(R_3,K_4) =$ 101110110010001101110111010011 00

$L_5 = R_4 =$ 011101110010001000000000001000101

$E(R_4) =$
101110101110100100000100000000000000000001000001010

$K_5 =$ 011111001110110000000111111010110101001110101000

$E(R_4) \oplus K_5 =$
110001100000010100000011111010110101000110100010

S-box outputs 010100001100100000110001111 01011

$f(R_4,K_5) =$ 001010000001001110101101110000 11

$L_6 = R_5 =$ 100010100100111110100110001101 11

$E(R_5) =$
110001010100001001011111110100001100000110101111

$K_6 =$ 011000111010010100111110010100000111101100101111

$E(R_5) \oplus K_6 =$
101001101110011101100001100000000101110101000000 0

S-box outputs 010000011111001101001100001111 01

$f(R_5,K_6) =$ 100111100100010111001101001 01100

$L_7 = R_6 =$ 111010010110011111001101011 01001

$E(R_6) =$
111101010010101100001111111001011010101101010011

$K_7 = 111011001000010010110111111101100001100010111100$

$E(R_6) \oplus K_7 =$
000110011010111110111000000100111011001111101111

S- box outputs 00010000011101010100000010101101

$f(R_6,K_7) = 10001100000001010001110000100111$

$L_8 = R_7 = 00000110010010101011101000010000$

$E(R_7) =$
000000001100001001010101010111110100000010100000

$K_8 = 111101111000101000111010110000010011101111111011$

$E(R_7) \oplus K_8 =$
111101110100100001101111100111100111101101011011

S-box outputs 01101100000110000111110010101110

$f(R_7,K_8) = 00111100000011101000011011111001$

$L_9 = R_8 = 11010101011010010100101110010000$

$E(R_8) =$
011010101010101101010010101001010111110010100001

$K_9 = 111000001101101111101011111011011110011110000001$

$E(R_8) \oplus K_9 =$
100010100111000010111001010010001001101100100000

S-box outputs 00010001000011000101011101110111

$f(R_8,K_9) = 00100010001101100111110001101010$

$L_{10} = R_9 = 00100100011111001100011001111010$

$E(R_9) =$
000100001000001111111001011000001100001111110100

$K_{10} =$
101100011111001101000111101110100100011001001111

$E(R_9) \oplus K_{10} =$
101000010111000010111110110110101000010110111011

S-box outputs 110110100000010001010010011101 01

$f(R_9,K_{10}) =$ 011000101011110010011110000100010

$L_{11} = R_{10} =$ 101101111101010111010111110110010

$E(R_{10}) =$
010110101111111010101011111010101111110110100101

$K_{11} =$ 001000010101111111010011110111110110100111000011 0

$E(R_{10}) \oplus K_{11} =$
011110111010000101111000001101000010111000100011

S-box outputs 011100110000010111010001000000 01

$f(R_{10},K_{11}) =$ 111000010000010011111010000000010

$L_{12} = R_{11} =$ 110001010111100000111100011111000

$E(R_{11}) =$
011000001010101111110000000111111000001111110001

$K_{12} =$ 011101010111000111110101100101000110011111101001

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

S-box outputs 011100110000010111010001000000 01

$f(R_{11},K_{12}) =$ 110000100110100011001111111101010

$L_{13} = R_{12} =$ 011101011011110100011000010110 00

$E(R_{12}) =$
001110101011110111111010100011110000001011110000

$K_{13} =$
100101111100010111010001111110101011101001000001

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

Sbox outputs 1001101011010001100010110100111 1

$f(R_{12},K_{13}) =$ 1101110110111011001010010010001 0

$L_{14} = R_{13} =$ 0001100011000011000101010101101 0

$E(R_{13}) =$
000011110001011000000110100010101010101011110100

$K_{13} =$ 010111110100001110110111111100101110011100111010

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

S-box outputs 0110010001111001100110101111 0001

$f(R_{13},K_{14}) =$ 1011011100110001100011100101010 1

$L_{15} = R_{14} =$ 1100001010001100100101100000110 1

$E(R_{14}) =$
111000000101010001011001010010101100000001011011

$K_{15} =$ 101111111001000110001101001111010011111100001010

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

S-box outputs 1011001011101000100011010011110 0

$f(R_{14},K_{15}) =$ 0101101110000001001001110110111 0

$R_{15} =$ 0100001101000010001100100011010 0

$E(R_{15}) =$
001000000110101000000100000110100100000110101000

$K_{16} = 110010110011110110001011000011100001011111110101$

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

S-box outputs 101001111000001100100100000101001

$f(R_{15},K_{16}) = 110010001100000000100111110011000$

$R_{16} = 000010100100110011011001100100101$

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

2.4.4.4. Độ an toàn của 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 giấu 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

$$e, f \in \{0, 1\}: S(x) \neq S(x \oplus 1\text{lef}00).$$

- 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 $K_1, \ldots, K_{16}$ 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

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 $x_1, x_2, \cdots$ (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 $y_1, y_2, \cdots$ 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 2.12.

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 $z_1, z_2, \cdots$ theo quy tắc $z_i = e_k(z_{i-1}), i \geq 1$. Dãy bản rõ $x_1, x_2, \cdots$ sau đó sẽ được mã hoá bằng cách tính $y_i = x_i \oplus z_i, i \geq 1$.

Hình 2.11. Chế độ ECB

Hình 2.12. Chế độ CBC

Hình 2.13. Chế độ CFB

Hình 2.14. 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 2.13 (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 $y_1, \ldots, y_n$.

Một số biến thể của DES

DES bội hai (Double DES)

a. Mã hóa DES bội hai

b. Giải mã DES bội hai

Hình 2.15. DES bội hai

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

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

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.

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

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

Hình 2.16. 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.

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.

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 2.17

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 $M_i = m_{i1}, m_{i2}, \ldots m_{i32}$

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

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

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:

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

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 2.17. Thuật toán mã hóa GDES

2.4.5. Chuẩn mã dữ liệu tiên tiến (AES)

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

2.4.5.1. Cơ sở toán học của AES

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

<u>Phép cộng:</u>

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

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

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

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 nhân:

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 $C=AE_H=10101110$

2.4.5.2. 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

Bảng 2.1. 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 Afine 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

<u>Dịch các hàng của State Array; Phép biến đổi ShiftRows( )</u>

Phép biến đổi tiếp theo của AES là dịch các hàng của State Array. Lượng dịch $\mathrm{Shift}\left(r, N_b\right)$ 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:

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

trong đó $0 \le c \le N_b$

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

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

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

<u>Trộn dữ liệu trong một cột State Array: Phép biến đổi Mixcolumns( )</u>

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

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

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

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

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

Ở đây $0 \le c < N_b$

Mở rộng khóa AES: KeyExpansion( )

Thuật toán AES sẽ tạo từ khóa mã 128 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 $N_b(N_r+1)$ 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 :

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

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

Begin

$i=0$

while $(i<N_k)$

$w[i]=\text{word}[\text{key}[4_*i],\text{key}[4_*i+1],\text{key}[4_*i+2],\text{key}[4_*i+3]]$

$i=i+1$

end while

$i \simeq N_k$

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

word $\text{temp}=w[i-1]$

if $(i \bmod N_k=0)$

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

else if $(N_k = 8 \text{ and } i \bmod N_k = 4)$

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

end if

$$w[i] \simeq w[i - N_k] = \text{xor temp}$$

$$i = i + 1$$

end while

end

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

Chương trình giải mã của AES

Cipher

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

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

2.5. BÀI TẬP

1. Hãy thiết lập hệ mật thay thế trên bảng chữ cái la tinh, sau đó mã hoá thông báo sau:

good wine needs no bush

Giả sử ta dùng hệ mật thay thế với khoá Π như sau:

$$\Pi = \begin{bmatrix} a & b & c & d & e & f & g & h & i & j & k & l & m & n & o & p & q & r & s & t & u & v & w & x & y & z \\ d & e & f & g & h & i & j & k & o & l & n & m & p & q & s & r & w & t & v & u & y & x & b & a & z & c \end{bmatrix}$$

hãy giải thông báo mã sau đây:

qdbqj vpybd mdifk yzhhq lfydt utsbo iacza

3. Cho m=5 hãy thiết lập mật mã hoán vị có độ dài m=5 và mã hoá thông báo rõ sau:

follow this road until you reach the river.

4. Biết rằng bản mã sau đây thu được từ mật mã hoán vị với m=6 và phép chuyển vị như sau:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 1 & 6 & 4 & 2 \end{pmatrix}$$

Hãy giải thông báo mã sau đây:

wfvqau tuuwra nfodgw ihdoja dacnaj

ajlppa wwhnou dognod dxglad feaelm

ynxirk mneaje nwwmam ahtfnh ajltpa wwrnou xkgzyf

5. Biết rằng thông báo mã sau đây thu được từ mật mã Vigenère :

VPXZG IAXIV WPUBT TMJPW IZITW ZT

với khoá CIPHER. Hãy giải thông báo mã đó.

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

7. Hệ mã Affine - Hill là hệ mã Hill được sửa đổi như sau: Giả sử mlà một số nguyên dương và $\mathcal{P}=C=(Z_{26})^m$. 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 $b \in (Z_{26})^m$ theo công thức $y = xL + b$. Bởi vậy, nếu $L = (l_{ij})$ và $b = (b_1,\ldots,b_m)$ thì:

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

Giả sử Oscar đã biết bản rõ là *"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.

8. 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:

LMQETXYEAGTXCTUIEWNCTXLZEWUAISPZYVAPEWLM GQWVAXFTGMSQCADAGTXLMDXNXSNPJQSYVAPRIQSMHNO CVAXFV.

9. 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):

MYAMRARUYIQTENCTORAHROYWĐSOYEOUARRGĐERN OGW

10. 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.

Chương 3
MẬT MÃ KHÓA CÔNG KHAI

3.1. Giới thiệu chung

Trong mô hình mật mã chúng ta nghiên cứu cho đến nay (mật mã khóa bí mật), Alice và Bob thoả thuận chọn một cách bí mật khoá k. Từ k người ta suy ra qui tắc mã hoá e_k và qui tắc giải mã d_k.Trong các hệ mật này, chúng ta thấy d_k hoặc trùng với e_k, hoặc dễ dàng rút ra từ e_k (ví dụ phép giải mã DES nói chung đồng nhất với phép mã hoá, chỉ khác là lược đồ khoá thì đảo ngược). Các hệ mật loại này được gọi là hệ mật khoá bí mật (hoặc riêng, hoặc đối xứng), vì việc tiết lộ e_k sẽ làm cho hệ thống không an toàn.

Một đặc điểm của hệ mật khoá bí mật là ở chỗ nó yêu cầu thoả thuận về khoá giữa Alice và Bob bằng sử dụng kênh an toàn, trước khi bản mã bất kì được truyền.Trong thực tế thực hiện điều này là rất khó.

Ý tưởng nằm sau hệ mật khoá công khai là ở chỗ người ta có thể tìm ra một hệ mật trong đó không thể tính toán để xác định d_k khi biết e_k. Nếu thế thì qui tắc mã e_k có thể cho công khai bằng cách công bố nó trong một thư mục (vì thế mới có thuật ngữ hệ mật khoá công khai). Ưu điểm của hệ mật khoá công khai là ở chỗ Alice (hoặc người khác bất kỳ) có thể gửi thông báo đã mã tới Bob (mà không cần liên lạc trước về khoá bí mật) bằng cách dùng qui tắc mã hoá công khai e_k. Bob sẽ là người duy nhất có thể giải bản mã này bằng cách sử dụng qui tắc giải mã bí mật d_k của anh ta.

Ta có thể hình dung như sau: Alice đặt một vật vào hộp sắt sau đó khoá nó với cái khoá bấm do Bob để lại. Bob là người duy nhất có thể mở hộp vì chỉ anh ta có chìa.

Một nhận xét rất quan trọng là hệ mật khoá công khai có thể không bao giờ cung cấp độ mật vô điều kiện. Đó là vì bằng quan sát bản mã y, đối phương có thể mã hoá mỗi bản rõ có thể nhờ e_k cho đến khi tìm thấy

x duy nhất thoả mãn $y=e_k(x)$. Nghiệm x này là giải mã của y. Như vậy độ an toàn của các hệ mật khoá công khai là độ an toàn tính toán.

Hàm mã hoá công khai e_k của Bob phải dễ dàng tính toán. Chúng ta chú ý rằng việc tính hàm ngược, nghĩa là việc giải mã, phải khó đối với bất kỳ người nào ngoài Bob. Tính chất dễ tính toán và khó đảo ngược này thường được gọi là tính chất một chiều (tựa như bản dẫn). Chúng ta mong muốn rằng e_k là hàm một chiều.

Các hàm một chiều đóng vai trò trung tâm trong mật mã, chúng quan trọng đối với việc thiết lập các hệ mật khoá công khai và trong các nội dung khác. Đáng tiếc là, mặc dù có nhiều hàm được người ta tin là hàm một chiều, nhưng hiện nay vẫn chưa có hàm nào được chứng minh là hàm một chiều.

Nếu ta định thiết lập hệ mật khoá công khai thì việc tìm hàm một chiều là chưa đủ. Bob muốn có thể giải mã các thông báo nhận được một cách có hiệu quả. Như vậy Bob cần có một cửa sập (trap door), nó chứa thông tin bí mật cho phép dễ dàng đảo ngược e_k. Nghĩa là Bob có thể giải mã hiệu quả vì anh ta có tri thức bí mật đặc biệt về k. Do đó ta nói rằng: f(x) là hàm một chiều cửa sập nếu đó là hàm một chiều, nhưng nó trở nên dễ đảo ngược khi có tri thức về cửa sập xác định. Nói chung, có những cách để tìm cửa sập của 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).

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 mật Rabin:*

Độ bảo mật của hệ Rabin cũng dựa trên độ khó của việc phân tích ra thừa số nguyên lớn.

- *Hệ mật ElGamal:*

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

- *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.

- *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 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

3.2. Hệ mật RSA

Bài toán phân tích thừa số

Bài toán phân tích một số nguyên n >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 toán 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 mod 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 $\leftarrow a^{q^l}$ mod 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 3.1. 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 $\leq$ 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 môdulo.

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 ($1 \leq v \leq n - 1$). 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 $v_1, \ldots, v_m$, thì sẽ có thể đạt được xác suất cho kết quả không đúng là $<1/2^m$!

3.2.1. Thuật toán mã hóa, giải mã RSA

Giả sử n=p.q,trong đó p,q là hai số nguyên tố lẻ khác nhau và $\Phi(n)$ là hàm Ơle. Hệ mật RSA được định nghĩa như sau:

Cho $P=C=Z_n$; $K=\{(n,p,q,a,b) : ab\equiv 1 \bmod \Phi(n)\}$

Với mỗi k=(n,p,q,a,b), quy tắc mã hóa và giải mã của hệ mật RSA được xác định như sau:

$$e_k(x)=x^b \bmod n$$

và $\quad d_k(y)=y^a \bmod n \qquad (x,y\in Z_n).$

3.2.2. Kiểm tra qui tắc giải mã

Do $ab\equiv 1 \bmod \Phi(n)$, $\Phi(n)=(p-1)(q-1)= \Phi(p)\ \Phi(q)$ nên $ab=1+t\Phi(n)$,với t là số nguyên khác 0. Chú ý rằng $0\leq x <n$.

(*) Giả sử $(x,n)=1$, ta có

$$y^a \bmod n \equiv (x^b)^a \bmod n \equiv x^{1+t\Phi(n)} \bmod n \equiv x[x^{\Phi(n)} \bmod n] \bmod n$$

$$\equiv x.1 \bmod n \quad (\text{ vì } (x,n)=1 \text{ nên } x^{\Phi(n)} \bmod n=1)$$

$$= x \ (\text{ do } x<n\ ).$$

(**) Nếu $(x,n) = d > 1$ thì d=p hoặc d=q hoặc d=n.

Nếu d=n thì x = 0 và đương nhiên y = 0. Do đó $y^a \bmod n = 0 = x$.

Giả sử d=p khi đó do $0\leq x <n$ nên x= p. Ta có:

$$y^a \bmod n \equiv x^{ab} \bmod n \equiv p^{ab} \bmod n$$

Ký hiệu

$$u = p^{ab} \bmod n;$$

Thế thì,

$u +kn = p^{ab}$, $0\leq u <n$, hay $u+kpq = p^{ab}$

Do đó

$$u = p(p^{ab-1}-kq) = p(p^{t\Phi(n)}-kq).$$

Vế phải chia hết cho p nên vế trái phải chia hết cho p, nghĩa là u phải chia hết cho p. Nhưng $0\leq u <n$ nên hoặc u=0 hoặc u=p. Nếu u=0 thì p^{ab-1} chia hết cho q. Suy ra p chia hết cho q. Vô lý vì p,q là hai số nguyên tố khác nhau. Thế thì u=p=x, tức là $y^a \bmod n = x$.

Vậy $(x^b)^a \bmod n = x$, $\forall x\in[1,n-1]$.

Ví dụ 3.2.

Giả sử Bob chọn p = 101 và q = 113. Thế thì n=11413 và $\Phi(n)=100*112=11200=2^6.5^2.7$.

Bob chọn b sao cho $(\Phi(n),b)=1$. Giả sử b=3533. Dùng thuật toán Ơclit mở rộng sẽ tìm được $b^{-1}=6597 \bmod 11200$. Vì thế số mũ bí mật của Bob là a=6597.

Bob công bố n =11413 và b =3533 trong thư mục khoá công khai.Bây giờ giả sử Alice muốn gửi bản rõ 9726 cho Bob. Cô sẽ tính:

$$9726^{3533} \bmod 11413=5761$$

và gửi bản mã 5761 trên kênh.

Khi Bob nhận được bản mã 5761, anh sẽ dùng số mũ bí mật của mình để giải

$$5761^{6597} \bmod 11413=9726$$

Do $ab\equiv 1 \bmod \Phi(n)$, $\Phi(n)= (p-1)(q-1)= \Phi(p)\ \Phi(q)$ nên $ab=1+t\ \Phi(n)$,với t là số nguyên khác 0. Chú ý rằng $0\leq x <n$.

(*) Giả sử $(x,n) =1$, ta có

$$y^a \bmod n \equiv (x^b)^a \bmod n \equiv x^{1+t\Phi(n)} \bmod n \equiv x[x^{\Phi(n)} \bmod n] \bmod n$$

$$\equiv x.1 \bmod n \quad (\text{ vì } (x,n) =1 \text{ nên } x^{\Phi(n)} \bmod n=1)$$

$$=x \ (\text{ do } x<n\ ).$$

3.2.3. Độ an toàn của hệ RSA

Độ an toàn của hệ RSA dựa trên hy vọng rằng hàm mã hoá $e_k(x)=x^b \bmod n$ là một chiều, từ đó đối phương không thể tính toán để giải bản mã được. Cái cửa sập cho phép Bob giải mã là kiến thức về phân tích n=p.q. Vì Bob biết p và q nên có thể tính được $\Phi(n)=(p-1)(q-1)$ và sau đó tính số mũ giải mã a nhờ thuật toán Ơclit mở rộng.

Muốn biết p và q thì Oscar phải phân tích được n, vì vậy nếu bài toán phân tích số là khó thì Oscar sẽ không phân tích được n. Cho đến nay, người ta thấy rằng bài toán phân tích số là khó và đoán rằng việc phá vỡ hệ RSA là tương đương với việc phân tích số nhưng đáng tiếc là

chưa chứng minh được điều đó. Một cách tổng quát, chưa có phương pháp nào phá được hệ RSA. Nghĩa là hệ RSA vẫn được coi là an toàn.

3.2.4. Thực hiện RSA

Việc thiết lập hệ RSA được Bob tiến hành theo các bước sau :

1/ Sinh ra hai số nguyên tố lớn p và q

2/ Tính n=p.q và $\Phi(n)=(p-1)(q-1)$

3/ Chọn ngẫu nhiên b $(0 < b < \Phi(n))$ sao cho $(b, \Phi(n))=1$

4/ Tính $a=b^{-1} \bmod \Phi(n)$ nhờ thuật toán Ơclit mở rộng.

5/ Công bố n và b trong thư mục như khoá công khai của mình.

Như đã phân tích ở trên, muốn cho hệ RSA an toàn thì n=p.q phải lớn để không thể phân tích được nó về mặt tính toán.

3.2.5. 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 3.1.

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 $_{Zu3}\, M^{e-1} \equiv 0 \bmod p$ chỉ có một nghiệm tương tự với q ta có được kết quả của định lý

Ví dụ 3.3. $n = 35$

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:

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

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

Nếu $\text{UCLN}(e-1, 2p') = p'$ 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 (bằng $1/p'$)

3.3. Hệ mật Rabin

3.3.1. 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).

3.3.2. Mã hóa và giải mã của hệ mật 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.

3.3.3. 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 $\overline{m} = 1001111001$.

3.3.4. 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.

3.4. Hệ mật Elgamal

3.4.1. Bài toán logarit rời rạc

Định nghĩa 3.1.

Cho G là nhóm cylic hữu hạn bậc n, α là phần tử sinh của G, β là phần tử thuộc G. Lôgarit rời rạc của β theo cơ số α được ký hiệu là $\log_\alpha \beta$, là một số nguyên x, $0 \le x \le n-1$, thỏa mãn $\beta = \alpha^x$.

Ví dụ 3.4.

Cho p=101, Z^*_{101} là nhóm cyclic có bậc n=100, α =2 là phần tử sinh của nhóm Z^*_{101}, ta có $2^{88} = 92 \pmod{101} \Rightarrow \log_2 92 = 88$ trên Z^*_{101}.

Bổ đề 3.1.

Cho α là phần tử sinh của nhóm cyclic G bậc n, $\beta, \gamma \in G$, s là một số nguyên, ta có:

$$\log_\alpha(\beta\gamma) = (\log_\alpha \beta + \log_\alpha \gamma)\ mod\ n$$

$$\log_\alpha \beta^s = s.(\log_\alpha \beta\ mod\ n$$

3.4.1.1. Bài toán lôgarit rời rạc tổng quát (GDLP)

Bài toán lôgarit rời rạc tổng quát (*Generalized discrete logarithm problem - GDLP*): cho G là nhóm cyclic hữu hạn bậc n, α là phần tử sinh của G, phần tử β∈G, tìm số nguyên x, $0 \le x \le n-1$, sao cho $\alpha^x = \beta$.

Độ khó của bài toán lôgarit rời rạc tổng quát (GDLP) độc lập với phần tử sinh

Chứng minh:

Cho α và γ là 2 phần tử sinh của nhóm cyclic G bậc n, và β∈G.

Đặt: $\begin{cases} x = \log_\alpha \beta \\ y = \log_\gamma \beta \\ z = \log_\alpha \gamma \end{cases} \Rightarrow \alpha^x = \beta = \gamma^y = (\alpha^z)^y.$

Vậy ta có: $\begin{cases} x = zy mod n \\ \log_\gamma \beta = (\log_\alpha \beta)(\log_\alpha \gamma)^{-1} mod n \end{cases}$

Điều này có nghĩa rằng bất kỳ thuật toán nào được dùng để tính lôgarit theo cơ số α cũng có thể dùng để tính lôgarit theo cơ số γ bất kỳ, với γ cũng là phần tử sinh của G.

3.4.1.2. Bài toán lôgarit rời rạc (DLP)

Bài toán lôgarit rời rạc (*Discrete logarithm problem -DLP*): cho p là số nguyên tố, α là phần tử sinh của nhóm Z_p^*, và phần tử $\beta \in Z_p^*$, tìm số nguyên x, $0 \le x \le p-2$, sao cho $\alpha^x = \beta \ (mod\ p)$.

Ví dụ 3.5. 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 rạ trên trường hữu hạn GF(p) có độ phức tạp lớn hơn so với trên $GF(2^m)$.

3.4.1.3. Một số thuật toán giải bài toán logarit rời rạc

Thuật toán vét cạn

Vét cạn là một thuật toán giải bài toán tìm 1 phương án đúng tron không gian n phương án.

Thuật toán vét cạn tìm phương án đúng bằng cách lựa chọn lần lượ từng phương án trong tập hợp tất cả các phương án của bài toán để tìm r phương án tối ưu.

Trong nhiều bài toán, không gian các phương án quá lớn. Do vậy khi áp dụng thuật toán vét cạn không đảm bảo về thời gian cũng nh kỹ thuật.

Thuật toán bước lớn bước nhỏ

Tính $\log_x \beta$ =? trên Z_p^*

- Tính m=[$\sqrt{n}$], với n là cấp của x, x là phần tử sinh.
- Tính bảng giá trị (j, x^j) với j= 0,...,m-1
- Tính bảng giá trị $(i, \beta.x^{-m.i})$ với i= 0,...,m-1
- Tính bảng giá trị x^j cho đến khi thỏa mãn $x^j = \beta.x^{-m.i}$

Khi đó $\log_x \beta = im + j$

Ví dụ 3.6. Tìm $\log_{31} 45$ =? trên Z_{61}^*.

Ø(61)=60 => m=[$\sqrt{60}$]=8

j								
$31^j mod 61$		1	6	3	2	1	1	1

Ta có

$$x^{-m} \, mod \, p = 31^{-8} \, mod \, 61 = (31^{-1})^8 \, mod \, 61 = 2^8 \, mod \, 61 = 12$$

$$\beta.x^{-m.i} = 45.12^i \bmod 61$$

i	0	1	2	3	4	5	6	7
$45.12^i mod 61$	45	52	14	46	3	36	5	60

Ta thấy tại $i = 2, j = 3$ thì $31^j \bmod 61 = 45.12^i \bmod 61 = 46$

Vậy $\log_{31} 45 = 8.3 + 2 = 26$

Thuật toán ρ- pollard

Nhóm G được chia thành 3 nhóm con có kích thước gần bằng nhau dựa vào một số tính chất dễ kiểm tra. Định nghĩa dãy các phần tử nhóm $x_0, x_1, x_2, \ldots$ bởi $x_0=1$ và:

$$x_{i+1} = f(x_i) \overset{def}{=} \begin{cases} \beta.x_i, & if\ x_i \in S_1 \\ x_i^2, & if\ x_i \in S_2 \\ \alpha.x_i, & if\ x_i \in S_3 \end{cases}$$

Với $i \geq 0$. Dùng dãy các phần tử nhóm này để định nghĩa 2 dãy khác của các số nguyên a_0, a_1, a_2 và $b_0, b_1, b_2 \ldots$ thỏa mãn $x_i = \alpha^{a_i} \beta^{b_i}$.

Với $a_0 = b_0 = 0$ thì:

$$a_{i+1} = \begin{cases} a_i, & if\ x_i \in S_1 \\ 2a_i \bmod n, & if\ x_i \in S_2 \\ a_i + 1, & if\ x_i \in S_3 \end{cases}$$

$$b_{i+1} = \begin{cases} b_i + 1, & if\ x_i \in S_1 \\ 2b_i \bmod n, & if\ x_i \in S_2 \\ b_i, & if\ x_i \in S_3 \end{cases}$$

Thuật toán tìm chu kỳ Floyd có thể được sử dụng để tìm 2 phần tử của nhóm x_i và x_{2i} sao cho $x_i = x_{2i}$. Khi đó $\alpha^{a_i}\beta^{b_i} = \alpha^{a_{2i}}\beta^{b_{2i}}$ và do đó $\beta^{b_i - b_{2i}} = \alpha^{a_i - a_{2i}}$. Lấy logarit theo cơ số α của cả 2 vế của đẳng thức cuối ta có: $(b_i - b_{2i}).\log_\alpha \beta \equiv (a_{2i} - a_i)(\bmod n)$

Nếu $b_i \neq b_{2i}(modn)$ (trường hợp $b_i = b_{2i}(mod)$ xảy ra với xác suất nhỏ), phương trình này có thể giải hiệu quả để xác định $\log_\alpha \beta$.

Thuật toán

INPUT: phần tử sinh α của nhóm tuần hoàn G có bận n nguyên tố, phần tử β∈G.

OUTPUT: logarith rời rạc $x = \log_\alpha \beta$

Set $x_0 \leftarrow 1$, $a_0 \leftarrow 0$, $b_0 \leftarrow 0$.

For i=1,2,... do

Dùng các đại lượng $x_{i-1}, a_{i-1}, b_{i-1}$và $x_{2i-2}, a_{2i-2}, b_{2i-2}$đã được tính trước để tính các đại lượng x_i, a_i, b_ivà x_{2i}, a_{2i}, b_{2i} theo các công thức trên.

Nếu $x_i = x_{2i}$thì làm:

Set $r \leftarrow b_i - b_{2i}$ mod n

Nếu r=0 thì dừng thuật toán và Output('fail')

Ngược lại, tính $x = r^{-1}(a_{2i} - a_i) mod n$và trả về (x).

Ví dụ 3.7. (Tính logarith trong nhóm con của Z_{383}^*), phần tử α =2 là phần tử sinh của nhóm con G trong Z_{383}^* có bậc n=191. Giả sử β = 228. Phân nhóm các phần tử của G thành 3 nhóm con theo quy tắc x ∈ S_1 nếu x≡1 (mod 3), x ∈ S_2 nếu x≡0(mod 3) và x ∈ S_3 nếu x≡2(mod 3). Bảng sau chỉ ra các giá trị của $x_i, a_i, b_i, x_{2i}, a_{2i}, b_{2i}$tại cuối mỗi vòng lặp. Cuối cùng tính:

$r = b_{14} - b_{28}\ mod\ 191 = 125,\ r^{-1} = 125^{-1}\ mod\ 191 = 136,\ r^{-1}(a_{28} - a_{14})\ mod\ 191 = 110$ Cho nên $\log_2 228 = 110$.

Bảng 3.2. Giải lôgarit rời rạc bằng thuật toán p-pollard.

i	x_i	a_i	b_i	x_{2i}	a_{2i}	b_{2i}
1	228	0	1	279	0	2
2	279	0	2	184	1	4
3	92	0	4	14	1	6
4	184	1	4	256	2	7

5	205	1	5	304	3	8
6	14	1	6	121	6	18
7	28	2	6	144	12	38
8	256	2	7	235	48	152
9	152	2	8	72	48	154
10	304	3	8	14	96	118
11	372	3	9	256	97	119
12	121	6	18	304	98	120
13	12	6	19	121	5	51
14	144	12	38	144	10	104

Thuật toán Pohlig-Hellman

Giả sử $n = p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r}$ là phân tích của n. Nếu $x = \log_\alpha \beta$ thì phương pháp ở đây là xác định $x_i = x \bmod p_i^{e_i}, 1 \le i \le t$ và sau đó sử dụng thuật toán Gauss rồi tìm $x \bmod n$. Mỗi số nguyên x_i được xác định bằng cách tính từng chữ số của nó là $l_0, l_1, \ldots, l_{e_i - 1}$ trong biểu diễn của x_i theo cơ số p_i : $x_i = l_0 + l_1 p_i + \ldots + l_{e_i - 1} p_i^{e_i - 1}, 0 \le l_j \le p_i - 1, 0 \le j \le e_i - 1$.

Thuật toán

INPUT:phần tử sinh α của nhóm tuần hoàn G có bậc n và phần tử β∈ G.

OUTPUT: logarithm rời rạc $x=\log_\alpha \beta$

Tìm phân tích của n: $n = p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r}$, với $e_i \ge 1$.

For i =1 to n do

Set $q \leftarrow p_i$, $e \leftarrow e_i$

Set $\gamma \leftarrow 1$ and $l_{-1} \leftarrow 0$

tính $\overline{\alpha} \leftarrow \alpha^{n/q}$

(tính l_j) For j=0 to e-1 do

Compute $\gamma \leftarrow \gamma \alpha^{l_{j-1} q^{j-1}}$ and $\overline{\beta} \leftarrow (\beta \gamma^{-1})^{n/q^{j+1}}$

Compute $l_j \leftarrow \log_{\overline{\alpha}} \overline{\beta}$

Set $x_i \leftarrow l_0 + l_1 q + \ldots + l_{e-1} q^{e-1}$

Dùng thuật toán Gauss rồi tìm số nguyên x, $0 \le x \le n-1$sao cho $x \equiv x_i (\text{mod } p_i^{e_i}), 1 \le i \le r$

Return(x)

Ví dụ 3.8. Tính logarith trong Z_{251}^*. Giả sử p=251. Phần tử α=71 là phần tử sinh của Z_{251}^* bậc n=250. Lấy β=210. Thế thì $x=\log_{71}210$ được tính như sau:

Phân tích của n là $250=2.5^3$

(a) tính x_1=x mod 2

có $\overline{\alpha} = \alpha^{n/2} \text{ mod } p = 250$ và $\overline{\beta} = \beta^{n/2} \text{ mod } p = 250$

.Vìvậy $x_1=\log_{250}250=1$

(b) tính x_2= x mod $5^3= l_0+l_1 5+l_2 5^2$

tính $\overline{\alpha} = \alpha^{n/5} \text{ mod } p = 20$

tính γ=1 và $\overline{\beta} = (\beta\gamma^{-1})^{n/5} \text{ mod } p = 149$. Dùng phép vét cạn tính được $l_0=\log_{20}149=2$

tính $\gamma = \gamma.\alpha^2 mod p = 21$

và $\overline{\beta} = (\beta\gamma^{-1})^{n/25} \text{ mod } p = 113$.

Dùng phương pháp vét cạn tính được $l_1 = \log_{20} 113 = 4$

tính $\gamma = \gamma.\alpha^{4.5} mod p = 115$

và $\overline{\beta} = (\beta\gamma^{-1})^{n/125} \text{ mod } p = 149$. Dùng phương pháp vét cạn tính được $l_2 = \log_{20} 149 = 2$cho nên $x_2 = 2 + 4.5 + 2.5^2 = 72$

Cuối cùng, giải cặp phương trình đồng dư $x \equiv 1 (mod\ 2)$, $x \equiv 72 mod 125$ để nhận được $x = \log_{71} 210 = 197$

Nhận xét: Thuật toán phân tích số ở bước 1 cần phải tìm được ước số nhỏ trước; nếu bậc n không là số nguyên mịn thì thuật toán này không hiệu quả.

Thuật toán tính chỉ số

Thuật toán

INPUT: phần tử sinh α của nhóm tuần hoàn G bậc n, phần tử β∈ G

OUTPUT: logarith rời rạc $y = \log_\alpha \beta$

(Chọn cơ sở nhân tử S). Chọn tập con $S = \{p_1, p_2, \dots, p_t\}$ của G sao cho một phần đáng kể các phần tử con của G có thể biểu diễn như là tích của các phần tử trong G.

(Chỉnh các quan hệ tuyến tính gồm logarith của các phần tử trong trong S)

Chọn ngẫu nhiên số nguyên k, $0 \le k \le n-1$và tính α^k

Cố gắng viết α^k như là tích của các phần tử trong S:

$$\alpha^k = \prod_{i=1}^{t} p_i^{c_i}, c_i \ge 0.$$

Nếu thành công, lấy logarith của cả 2 vế để được quan hệ tuyến tính

$$k \equiv \sum_{i=1}^{t} c_i \log_\alpha p_i \pmod n$$

Lặp lại cho đến khi nhận được t+c quan hệ tuyến tính với c là số nguyên dương nhỏ (ví dụ c=10) sao cho hệ phương trình được cho bởi t+c quan hệ có lời giải duy nhất với xác suất cao.

Tìm logarith của các phần tử trong S.

Giải hệ t+c phương trình tuyến tính theo modulo n (với t ẩn số) đã thu được ở bước 2 để nhận được các giá trị của $\log_\alpha p_i$, $1 \le i \le t$

Tính y.

Chọn số nguyên ngẫu nhiên k, $0 \le k \le n-1$ và tính $\beta.\alpha^k$

Cố gắng viết $\beta.\alpha^k$ như là tích của các phần tử trong S: $\beta.\alpha^k = \prod_{i=1}^{t} p_i^{d_i}, d_i \geq 0$. Nếu không thành công thì quay lại bước 4.1. ngược lại, lấy logarith của cả 2 vế phương trình trên và được $\log_\alpha \beta = \left(\sum_{i=1}^{t} d_i \log_\alpha p_i - k\right) \bmod n$;

cho nên tính $y = \left(\sum_{i=1}^{t} d_i \log_\alpha p_i - k\right) \bmod n$ và trả về (y).

Ví dụ thuật toán trong Z_p^*

Trong trường Z_p với p nguyên tố, cơ sở nhân tử S có thể chọn như là t số nguyên tố đầu tiên. Quan hệ ở bước 2.2 của thuật toán trên có thể sinh ra bằng cách tính $\alpha^k mod\ p$, sau đó bằng cách chia thử để kiểm tra xem số nguyên này có phải là tích của các số nguyên tố trong S hay không.

Sau đây chúng ta xét một ví dụ trong Z_{229}^*, tức là $p = 229$. Phần tử $\alpha = 6$ là phân tử sinh của Z_{229}^* có bậc $n = 228$. Xét $\beta = 13$. Khi đó $\log_{16} 13$ được tính như sau bằng kỹ thuật tính chỉ số.

Cơ sở nhân tử được chọn là 5 số nguyên tố đầu tiên:

$$S = \{2,3,5,7,11\}$$

Nhận được 6 quan hệ sau chứa các phần tử của cơ sở nhân tử (những phép thử không thành công được bỏ qua):

$$6^{100} mod 229 = 180 = 2^2.3^2.5$$

$$6^{18} mod 229 = 176 = 2^4.11$$

$$6^{12} mod 229 = 165 = 3.5.11$$

$$6^{62} mod 229 = 154 = 2.7.11$$

$$6^{143} mod 229 = 210 = 2.3.5.7$$

$$6^{206} mod 229 = 210 = 2.3.5.7$$

Các quan hệ này dẫn đến 6 phương trình sau cho logarith của các phần tử trong cơ sở nhân tử:

$$100 \equiv 2.\log_6 2 + 2.\log_6 3 + \log_6 5\,(mod228)$$
$$18 \equiv 4\log_6 2 + \log_6 11(mod228)$$
$$12 \equiv \log_6 3 + \log_6 5 + \log_6 11\,(mod228)$$
$$62 \equiv \log_6 2 + \log_6 7 + \log_6 11(mod228)$$
$$143 \equiv \log_6 2 + 2\log_6 3 + \log_6 11(mod228)$$
$$206 \equiv \log_6 2 + \log_6 3 + \log_6 5 + \log_6 7(mod228)$$

Giải hệ gồm 6 phương trình tuyến tính có 5 ẩn số (đó là logarith $x_i = \log_6 p_i$) sẽ cho lời giải $\log_6 2 = 21$, $\log_6 3 = 208$, $\log_6 5 = 98$, $\log_6 7 = 107$, $\log_6 11 = 162$.

Giả sử rằng chọn số nguyên $k = 77$. Vì $\beta.\alpha^k = 13.6^{77} mod229 = 147 = 3.7^2$, từ đó suy ra rằng

$$\log_6 13 = (\log_6 3 + 2\log_6 7 - 77)mod228 = 117.$$

Ví dụ thuật toán tính chỉ số trong trường $F_{2^m}^*$

Các phần tử của trường hữu hạn F_{2^m} được biểu diễn như là các đa thức trong $Z_2[x]$ có bậc nhiều nhất (m-1), với phép nhân được thực hiện theo modulo một đa thức bất khả quy cố định f(x) bậc m trong $Z_2[x]$. Cơ sở nhân tử S có thể chọn là tập tất cả các đa thức bất khả quy trong $Z_2[x]$ có bậc không vượt quá một số b nào đó. Quan hệ ở bước 2.2 được sinh ra bằng cách tính $x^k\ mod\ f(x)$và sử dụng phép chia thử để kiểm tra xem đa thức này có phải là tích của các đa thức trong S hay không. Ví dụ sau được tính trong $F_{2^7}^*$.

Đa thức $f(x) = x^7 + x + 1$ là bất khả quy trên Z_2. Các phần tử của trường hữu hạn F_{2^7} có bậc 128 có thể biểu diễn như là tập tất cả các đa thức trong $Z_2[x]$ có bậc nhiều nhất bằng 6, với phép nhân thực hiện theo modulo $f(x)$. Bậc của $F_{2^7}^*$ là $n = 2^7 - 1 = 127$và x là phần tử sinh của

$F_{2^7}^*$ Giả sử $\beta = x^4 + x^3 + x^2 + x + 1$. Khi đó $y = \log_\alpha \beta$ có thể tính sau theo thuật toán tính chỉ số.

Cơ sở nhân tử được chọn là tập tất cả các đa thức bất khả quy trong $Z_2[x]$ có bậc không quá 3:

$$S = \{x, x+1, x^2+x+1, x^3+x+1, x^3+x^2+1\}$$

Có 5 quan hệ sau giữa các phần tử của cơ sở nhân tử:

$$x^{18} \bmod f(x) \equiv x^6 + 4 = x^4(x+1)^2$$

$$x^{105} \bmod f(x) \equiv x^6 + x^5 + x^4 + x$$

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

$$x^{18} \bmod f(x) \equiv x^6 + x^4 = x^2(x+1)^2(x^2+x+1)$$

$$x^{18} \bmod f(x) \equiv x^6 + x^4 = (x+1)^2(x^3+x+1)$$

$$x^{121} \bmod f(x) \equiv x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

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

Các quan hệ này dẫn đến 5 phương trình đối với các logarith của các phần tử trong cơ sở nhân tử, ta đặt:

$p_1 = \log_x x, p_2 = \log_x(x+1), p_3 = \log_x(x^2+x+1), p_4 = \log_x(x^3+x+1), p_5 = \log_x(x^3+x^2+1)$ $18 \equiv 4p_1 + 2p_2 (mod 127)$

$$105 \equiv p_1 + 2p_2 + p_5 (mod 127)$$

$$72 \equiv 2p_1 + 2p_2 + p_3 (mod 127)$$

$$45 \equiv 2p_2 + p_4 (mod 127)$$

$$121 \equiv p_4 + p_5 (mod 127)$$

Giải hệ 5 phương trình tuyến tính theo 5 ẩn p_i cho kết quả $p_1 = 1, p_2 = 7, p_3 = 56, p_4 = 31, p_5 = 90$. Giả sử $k = 66$được chọn. Vì $\beta.\alpha^k = (x^4+x^3+x^2+x+1).x^{66} \bmod f(x) = x^5 + x^3 + x = x(x^2+x+1)^2$

nên $\log_x(x^4+x^3+x^2+x+1) = (p_1 + 2p_3 - 66) \bmod 127 = 47$.

Nhận xét:

Thuật toán tính chỉ số khó triển khai vì:

- Kỹ thuật để chọn cơ sở nhân tử chưa được chỉ ra.

- Phương pháp hiệu quả để chỉ ra một số quan hệ cần thiết cũng chưa được chỉ ra.

- Kỹ thuật này cũng không được áp dụng cho mọi nhóm.

3.4.2. Mã hóa, giải mã Elgamal

3.4.2.1. 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.

3.4.2.2. Thuật toán mã hóa, giải mã

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 $c = (\gamma, \delta)$ 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 $\left(\gamma^{-a}\right)\delta \bmod p$.

Chứng minh hoạt động giải mã:

Thuật toán trên cho phép A thu được bản rõ vì:

$$\gamma^{-a}\,\delta \equiv \alpha^{-ak}.m\,\alpha^{ak} \equiv m \bmod p$$

3.4.2.3. 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 $\left(p = 2357, \alpha = 2, \alpha^a = 1185\right)$

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:

$m = 872.697 \bmod 2357 = 2035$.

3.4.3. Tham số của hệ mật

Để chống lại các thuật toán tấn công P-Pollard, Pollig-Hellman, số nguyên tố p được chọn phải thỏa mãn một số điều kiện sau:

$$\begin{cases} p - 1 \text{ có ước số nguyên tố lớn (cỡ 100 bit trở lên)} \\ p \text{ có độ lớn cỡ 1024 bit trở lên} \end{cases}$$

Trong thực tế, thường sử dụng các số nguyên tố p có dạng: $p = 2q + 1$ (q là số nguyên tố), p khi đó được gọi là số nguyên tố siêu mạnh.

Bảng sau liệt kê một số số nguyên tố p có dạng như vậy và phần tử sinh α nhỏ nhất của nhóm.

Bảng 3.3. Một số số nguyên tố dạng p=2q+1

STT	P	Số chữ số của p	Độ dài bits của p	α
1	107	3	7	2
2	1000000007	10	30	5
3	4000000559	10	31	7
4	4294967387	10	32	2
5	18446744073709554719	20	64	7
6	340282366920938463463374607431768223907	39	128	2
7	115792089237316195423570985008687907853269984665640564039457584007913129870127	78	256	5
8	13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006370963	155	512	2
9	17976931348623159077293051907890247336179769789423065727343008115773267580550096313270847732240753602112011387987139335765878976881441662249284743063947412437776789342486548527630221960124609411945308295208500576883815068234246288147391311054082723716335051068458629823994724593847971630483535632962422579508 3	309	1024	2
10	32317006071311007300714876688669951960444102669715484032130345427524655138867890893197201411522913463688717960921898019494119559150490921095088152386448283120630877	617	2048	2

	367300996091750197750389652106796057638384067568276792218642619756161838094338476170470581645852036305042887575891541065808607552399123930385521914333389668342420684974786564569494856176035326322058077805659331026192708460314150258592864177116725943603718461857357598351152301645904403697613233287231227125684710820209725157101726931323469678542580656697935045997268352998638215525166389437335543602135433229604645318478604952148193555853611059606260467			

3.5. Một số hệ mã khóa công khai khác

3.5.1. Bài toán xếp ba lô và hệ mật Merkle - Hellman

3.5.1.1. Định nghĩa dãy siêu tăng

Định nghĩa 3.2. *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* $\forall i, 2 \le i \le n$

3.5.1.2. 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:

$$S = b_1 M_1 + b_2 M_2 + \ldots + b_n M_n$$

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.

3.5.1.3. 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.

3.5.1.4. 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

3.5.1.5. 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: $(W^{-1} = -224 = 513)$

$$d = W^{-1}c \bmod M = 136$$

và giải bài toán xếp ba lô trong trường hợp dãy siêu tăng sau:

$$136 = 12r_1 + 17r_2 + 33r_3 + 74r_4 + 157r_5 + 316\,r_6$$

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.

3.5.2. 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.

3.5.2.1. 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, $0 \leq d \leq p^h - 2$

(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 $(f(x), g(x), \pi, d)$.

3.5.2.2. 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 $\binom{p}{h} = \dfrac{p!}{h!(p-h)!}$.

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

$$M_{i-1} \leftarrow 0 \left(CY : \binom{n}{0} = 1 \qquad n \ge 0 \right.$$

$$\left. \binom{0}{l} = 0 \qquad l \ge 1 \right)$$

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.

$s(x) = \prod_{j-1}^{h} (x + t_j)$ 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

3.5.2.3. 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:

$a_0=\log_{g(x)}(x) \quad =1028$

$a_1=\log_{g(x)}(x+1)=1935$

$a_2=\log_{g(x)}(x+2)=2054$

$a_3=\log_{g(x)}(x+3)=1008$

$a_4=\log_{g(x)}(x+4)=379$

$a_5=\log_{g(x)}(x+5)=1780$

$a_6=\log_{g(x)}(x+6)=223$

(5) Chọn phép hoán vị ngẫu nhiên trên $\{0,1,2,3,4,5,6\}$ như sau:

$\pi(0)=6$	$\pi(3)=2$	$\pi(5)=5$
$\pi(1)=4$	$\pi(4)=1$	$\pi(6)=3$
$\pi(2)=0$		

(6) Chọn số nguyên ngẫu nhiên $d = 1702$

(7) Tính

$$C_0 = (a_6 + d) \bmod 2400 = 1925$$
$$C_1 = (a_4 + d) \bmod 2400 = 2081$$
$$C_2 = (a_0 + d) \bmod 2400 = 330$$
$$C_3 = (a_2 + d) \bmod 2400 = 1356$$
$$C_4 = (a_1 + d) \bmod 2400 = 1237$$
$$C_5 = (a_5 + d) \bmod 2400 = 1082$$
$$C_6 = (a_3 + d) \bmod 2400 = 310$$

(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

$\pi^{-1}(0) = 2 \qquad \pi^{-1}(2) = 3 \qquad \pi^{-1}(3) = 6 \qquad \pi^{-1}(6) = 0$

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.

3.5.3. Bài toán mã sửa sai và hệ mật McElice

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 ngụy 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.

Định nghĩa 3.3.

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.

Định lý 3.2.

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* $Hx^T = [00\ldots0]^T$.

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 giềng 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 $x_0 G = x_1$.

Hình 3.1. Hệ mật Mc Elice

Ví dụ 3.9. 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.

3.5.4. Hệ mật trên đường cong elliptic

3.5.4.1. 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 $y^2 = x^3 - 2x + 1$

3.5.4.2. 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: $4a^3 + 27b^2 \bmod p \ne 0$. 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 3.10. (cấu trúc của một nhóm E)

Giả sử p = 23, a = 1 và b = 1

Trước tiên ta kiểm tra lại:

$$\begin{aligned} 4a^3 + 27b^2 \bmod p &= 4.1^3 + 27.1^2 \bmod 23 \\ &= 4 + 27, \text{od}23 = 31 \bmod 23 \\ &= 8 \ne 0 \end{aligned}$$

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 3.3 dưới đây:

Bảng 3.4. Giá trị y tương ứng với x trên Z_{23}

x	x^3+x+1 mod 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:

$$E_{23}(1,1) = \begin{Bmatrix} (0,1) & (0,22) & (1,7) & (1,16) & (3,10) & (3,13) & (4,0) \\ (5,4) & (5,19) & (6,4) & (6,19) & (7,11) & (7,12) & (9,7) \\ (9,16) & (11,3) & (11,20) & (12,4) & (12,19) & (13,7) & (13,16) \\ (17,3) & (17,20) & (18,3) & (18,20) & (19,5) & (19,18) & \end{Bmatrix}$$

Hình 3.3. Nhóm E_{23} (1, 1)

3.5.4.3. 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 3.11. 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 3.5. Bảng tính kP

k	$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ (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)

3.5.4.4. 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:

$$P_C = [(kG),(P_M + kP_B)]$$

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 $(P_M + kP_B)$);

$$(P_M + kP_B) - n_B(kG) = (P_M + kn_B G) - n_B(kG) = P_M$$

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 3.12.

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: $P_B = n_B G = 85(0,\ 376)$

$$P_B = (671, 558)$$

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:

$$(P_M + kP_B) - n_B(kG) = (217, 606) - [85(34, 633)]$$
$$= (217, 606) - [(47,416)]$$
$$= (217, 606) + (47,4-16)$$

(vì $-P = (x_1, -y_1)$)

$$= (217, 606) + (47,335)$$

(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.

3.5.4.5. 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: 2.10^8MIPS - 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.

3.6. 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 công khai 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.

3.7. Bài tập

1. Giả sử p=13, q=17. Thế thì n=221; $\phi(n)=192$. Cho b=11 là số công khai của Bob.

a. Hãy tìm số mũ bí mật a của Bob.

b. Giả sử Alice muốn gửi bản rõ x=123 cho Bob. Hãy tính bản mã mà Alice sẽ gửi cho Bob.

2. Với số liệu ở bài 1, giả sử Alice gửi cho Bob bản mã: y=45. Hãy giải để giúp Bob tìm ra bản rõ mà Alice đã gửi cho Bob.

3. Biết rằng Bob dùng hệ RSA với n=667. Bob để lộ $\phi(667)=616$. Hãy tìm p,q của Bob.

4. Cho p=23;

a. Hãy tìm phần tử nguyên thuỷ α trong Z_{23}

b. Hãy chọn khoá bí mật a và khoá công khai $\beta=\alpha^a \bmod 23$ của Bob.

5. 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.

6. 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?

7. 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.

8. 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.

d. Hãy xác định khoá bí mật.

e. 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.

9. 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).

10. 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 nhận được sau bằng phương pháp giải mã theo syndrom:

a. $r = (1,\ 1.\ 0,\ 0,\ 0.\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,)$

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 VÀ CHỮ KÍ SỐ

4.1. Giới thiệu về hàm băm

Các hàm băm mật mã đóng một vai trò quan trọng trong mật mã hiện đại, chúng có thể được dùng để xác thực tính nguyên vẹn dữ liệu cũng như được dùng để tạo chữ ký số trong các giao dịch điện tử.

Các hàm băm lấy một thông báo làm đầu vào và tạo ra một đầu ra được xem như là mã băm (hash code), kết quả băm (hash result), hoặc giá trị băm (hash value). Chính xác hơn, một hàm băm h ánh xạ các xâu bit có độ dài hữu hạn tuỳ ý thành các xâu có độ dài cố định (n-bit). Với một miền xác định D và khoảng xác định R với h: $D \rightarrow R$ và $|D| > |R|$, hàm là nhiều-sang-một (many-to-one), nghĩa là tồn tại các va chạm (các cặp đầu vào với cùng đầu ra) là điều không thể tránh khỏi. Vai trò cơ bản của các hàm băm mật mã là một giá trị băm coi như ảnh đại diện thu gọn (compact representative image) (đôi khi gọi là một *dấu vết* (imprint), *vân tay số* (digital fingerprint), hoặc *tóm lược thông báo* (message digest)) của một xâu đầu vào, và có thể được dùng như là một định danh duy nhất với xâu đó.

Các hàm băm thường được dùng cho toàn vẹn dữ liệu kết hợp với các lược đồ chữ ký số. Một lớp các hàm băm riêng được gọi là mã xác thực thông báo (MAC) cho phép xác thực thông báo bằng các kỹ thuật mã đối xứng. Các thuật toán MAC có thể được xem như các hàm băm lấy hai đầu vào riêng biệt: một thông báo và một khóa bí mật, và tạo ra một đầu ra có kích cỡ cố định (n bit), với mục đích thiết kế sao cho nó trên thực tế không thể tạo ra cùng đầu ra mà không biết thông tin về khóa. MAC có thể được dùng để cung cấp cho toàn vẹn dữ liệu và xác thực dữ liệu gốc, cũng như các lược đồ định danh khóa đối xứng.

Có thể sử dụng các hàm băm (không khóa) cho toàn vẹn dữ liệu như sau. Giá trị băm tương ứng với một thông báo cụ thể x được tính ở thời điểm T_1. Tính toàn vẹn của giá trị băm này (mà không phải bản thân

thông báo) được bảo vệ ở một số dạng. Ở một thời điểm T_2 sau đó, kiểm tra sau được thực hiện để quyết định xem thông báo đã bị thay đổi chưa, cụ thể là xem thông báo x′ có giống với thông báo gốc hay không. Giá trị băm của x′ được tính toán và so sánh với giá trị băm được bảo vệ; nếu chúng bằng nhau, người ta chấp nhận rằng các đầu vào là cũng bằng nhau, và do đó thông báo chưa bị thay đổi. Bài toán đảm bảo tính toàn vẹn của một thông báo lớn do đó được rút gọn về việc đảm bảo tính toàn vẹn của một giá trị băm có kích cỡ cố định. Vì sự tồn tại của các va chạm là chắc chắn trong các ánh xạ nhiều-vào-một, sự kết hợp tính duy nhất giữa các đầu vào và các giá trị băm chỉ theo nghĩa về mặt tính toán. Một giá trị băm có thể là định danh duy nhất với một đầu vào về mặt thực hành, và các va chạm phải là khó tìm về mặt tính toán.

4.1.1. Khái niệm và phân loại hàm băm

Nói chung, các hàm băm có thể được chia thành hai lớp: các hàm băm không có khóa để chỉ các hàm bị tác động một tham số đầu vào duy nhất (một thông báo); và các hàm băm có khóa để chỉ các hàm bị tác động bởi hai tham số đầu vào (một thông báo và một khóa bí mật).

Định nghĩa 4.1. *Một hàm băm h (theo nghĩa không hạn chế) là một hàm có tối thiểu hai tính chất sau:*

(i) Nén – h ánh xạ một đầu vào x có độ dài bit tùy ý thành một đầu ra h(x) có độ dài bit cố định n.

(ii) Dễ tính toán – cho trước h và một đầu vào x, h(x) là dễ tính toán.

Trong sử dụng thực tế, việc phân loại hàm băm theo các hướng (có khóa và không có khóa) là cần thiết. Phân loại theo chức năng của hàm băm, ta có hai kiểu hàm băm sẽ được xem xét cụ thể trong chương này.

1. Mã phát hiện tác động (manipulation detection codes – MDC) cũng được xem như các mã phát hiện sửa đổi (modification detection code). Mục đích của MDC là để cung cấp một biểu diễn ảnh hoặc băm của thông báo, thỏa mãn thêm các tính chất được cải tiến ở dưới. MDC là các lớp con của các hàm băm không có khóa, và chúng có thể được phân

loại thêm nữa; các lớp đặc biệt của MDC được tập trung chính trong chương này là:

(i) *Các hàm băm một chiều (OWHF)*: là các hàm băm mà việc tìm một đầu vào để băm thành một giá trị băm được xác định trước là khó.

(ii) *Các hàm băm kháng va chạm (CRHF)*: là các hàm băm mà việc tìm hai đầu vào có cùng giá trị băm là khó.

2. Các mã xác thực thông báo (MAC)

Mục đích của một MAC là phải thuận tiện mà không sử dụng bất kỳ kỹ thuật bổ sung nào, bảo đảm cả tài nguyên của thông báo và tính toàn vẹn của nó. MAC có hai tham số khác nhau về chức năng, một thông báo đầu vào và một khóa bí mật; chúng là một lớp con của các hàm băm có khóa.

Hình 4.1. Phân loại các hàm băm mật mã và ứng dụng

Hình 4.1 chỉ ra cách phân loại hàm băm. Giả thiết rằng việc mô tả một thuật toán hàm băm là công khai, do đó trong trường hợp MDC, cho trước một thông báo làm đầu vào, một người bất kỳ đều có thể tính kết quả băm; và trong trường hợp MAC, cho trước một thông báo làm đầu vào, một người bất kỳ biết khóa bí mật có thể tính được giá trị băm.

4.1.2. Các tính chất cơ bản

Để thuận tiện cho các định nghĩa sau này, ta đưa thêm 3 tính chất sau (bổ sung thêm cho tính *dễ tính toán* và *nén* như định nghĩa 4.1) với một hàm băm không có khóa h với các đầu vào x, x′ và các đầu ra là y, y′.

1. *Kháng tiền ảnh* – với hầu hết toàn bộ các đầu ra xác định trước, ta không thể tính toán để tìm đầu vào mà băm thành đầu ra đó; tức là không thể tìm một tiền ảnh x′ sao cho h(x′) = y khi cho trước y bất kỳ mà với chúng đầu vào tương ứng là chưa được biết*.

2. *Kháng tiền ảnh thứ hai* – ta không thể tính toán để tìm một đầu vào thứ hai bất kỳ mà có cùng đầu ra như với đầu vào đã được xác định trước, tức là, cho trước x, không thể tìm được nghịch ảnh thứ hai x′≠ x sao cho h(x) = h(x′).

3. *Kháng va chạm* – ta không thể tính toán được hai đầu vào phân biệt x, x′ mà băm thành cùng đầu ra (h(x) = h(x′)). (Chú ý rằng ở đây được tự do chọn cả hai đầu vào.)

Ở đây thuật ngữ "*dễ*" và "*không thể tính toán*" (hoặc "*khó*") là những khái niệm không có định nghĩa hình thức nào; chúng ta sẽ giải thích chúng sơ bộ để có thể hiểu trong phạm vi của chương. "Dễ" có thể theo nghĩa không gian và thời gian đa thức; hoặc trong một số phép tính của máy hoặc các đơn vị thời gian cụ thể (có thể theo giây hoặc mili giây). Còn về khái niệm "*không thể tính toán*" có thể bao gồm khối lượng tính toán siêu đa thức, hoặc vượt xa tài nguyên hiện có, hoặc vượt quá một cận dưới với số các phép tính hoặc bộ nhớ đòi hỏi theo nghĩa một tham số an toàn cụ thể...

Xét một lược đồ chữ ký số trong đó chữ ký được áp dụng cho giá trị băm h(x) thay vì thông báo x. Ở đây h là một MDC với tính chất kháng

---

* Tính chất này là: một đối phương có thể dễ dàng tính trước các đầu ra cho tập nhỏ các đầu vào bất kỳ, và do đó lấy ngược hàm băm chỉ đơn thuần cho những đầu ra đó.

tiền ảnh thứ hai, nếu không thì đối phương C có thể thu được chữ ký của bên A nào đó trên h(x), sau đó tìm một thông báo x′ sao cho h(x) = h(x′), và khẳng định rằng A đã ký vào thông báo x′. Nếu C có thể chọn được thông báo mà A ký, thì C chỉ cần tìm một cặp va chạm (x, x′) dễ hơn việc tìm một tiền ảnh thứ hai của x; trong trường hợp này, kháng va chạm cũng là cần thiết. Ít rõ ràng hơn là yêu cầu về kháng tiền ảnh cho một số lược đồ chữ ký khóa công khai; xét RSA, mà bên A có khóa công khai (e, n). C có thể chọn một giá trị ngẫu nhiên y, tính $z = y^e \bmod n$, và (phụ thuộc vào quá trình kiểm tra chữ ký RSA cụ thể được sử dụng) khẳng định là y là chữ ký của A lên z. Sự giả mạo này (có thể xảy ra) gây nguy hiểm nếu C có thể tìm được một tiền ảnh x sao cho h(x) = z và với nó x là sử dụng thực tế.

Định nghĩa 4.2. *Hàm băm một chiều* (OWHF) *là hàm băm h như định nghĩa 4.1 (cụ thể có tính chất dễ tính toán và nén) với các tính chất bổ sung sau, như đã định nghĩa ở trên: kháng tiền ảnh, kháng tiền ảnh thứ hai.*

Định nghĩa 4.3. *Hàm băm kháng va chạm* (CRHF) *là hàm băm h như định nghĩa 4.1 (cụ thể là dễ tính toán và nén) với các tính chất bổ sung sau, như đã định nghĩa ở trên: kháng tiền ảnh thứ hai và kháng va chạm.*

Ví dụ 4.1.

Một kiểm tra tổng modulo-32 đơn giản (tổng 32-bit của tất cả các từ 32-bit của xâu dữ liệu) là một hàm được tính toán dễ dàng, chúng có tính nén, nhưng không kháng tiền ảnh.

Các mục tiêu của đối phương với các thuật toán MDC

Mục tiêu của đối phương muốn tấn công một MDC là như sau:

(a) Để tấn công một OWHF: cho trước giá trị băm y, tìm một tiền ảnh x sao cho y = h(x) hoặc cho một cặp (x, h(x)), tìm một tiền ảnh thứ hai x′ sao cho h(x′) = h(x).

(b) Để tấn công một CRHF: tìm hai đầu vào bất kỳ x, x′ sao cho $h(x') = h(x)$.

Một CRHF phải được thiết kế để chống lại các tấn công ngày sinh chuẩn

Định nghĩa 4.4 *Một thuật toán mã xác thực thông báo (MAC) là một họ các hàm h_k được tham số hóa bởi một khóa bí mật k, với các tính chất sau:*

1. Trị k và một đầu vào x, $h_k(x)$ là dễ tính toán. Kết quả này gọi là giá trị MAC hoặc MAC.

2. Nén – h_k ánh xạ một đầu vào x có độ dài bit tùy ý hữu hạn thành một đầu ra $h_k(x)$ có độ dài bit cố định n.

Hơn nữa, khi cho trước mô tả về họ hàm h, với mỗi giá trị có thể được phép cố định của k (không biết với đối phương), tính chất sau phải xảy ra:

3. Kháng tính toán – cho trước không hoặc nhiều cặp text-MAC $(x_i, h_k(x_i))$, ta không thể tính toán để tìm một cặp text-MAC bất kỳ $(x, h_k(x))$ cho một đầu vào mới $x \neq x_i$ (bao gồm cả khả năng $h_k(x) = h_k(x_i)$ với i nào đó).

Nếu không có tính kháng tính toán, thuật toán MAC có thể bị giả mạo. Chú ý, tính kháng tính toán suy ra tính không thể khôi phục khóa, còn tính không thể khôi phục khóa không suy ra được tính kháng tính toán.

<u>*Nhận xét 4.1.*</u> (*tính kháng của MAC khi khóa đã biết*) Định nghĩa 4.4 không nói rằng các thuật toán MAC có cần phải là kháng va chạm và kháng tiền ảnh cho các thành viên đã biết khóa k hay không.

Các mục tiêu của đối phương với các thuật toán MAC

Mục tiêu của đối phương tương ứng với một MAC là như sau:

Để tấn công MAC: không biết trước về khóa k, tính một cặp text-MAC mới $(x, h_k(x))$ với $x \neq x_i$, khi biết trước một cặp hoặc nhiều cặp $(x_i, h_k(x_i))$.

Kháng tính toán của MAC ở đây cần phải đúng khi các văn bản x_i có thể được cho bởi đối phương, hoặc có thể cho đối phương tự do lựa chọn. Tương tự như tình huống như các lược đồ chữ ký, các tình huống tấn công sau tồn tại với các thuật toán MAC được xếp theo lợi thế tăng dần:

1) *Tấn công văn bản đã biết* (known-text attack): Một hoặc nhiều cặp $(x_i, h_k(x_i))$ là có giá trị.

2) *Tấn công văn bản lựa chọn* (chosen-text attack): một hoặc nhiều cặp $(x_i, h_k(x_i))$ là có giá trị với x_i được chọn bởi đối phương.

3) *Tấn công văn bản chọn lọc thích ứng* (adaptive chosen-text attack): x_i có thể được chọn bởi đối phương như ở trên, bây giờ cho phép lựa chọn thành công dựa trên các kết quả truy vấn có ưu tiên.

Các thuật toán MAC phải chống lại được tấn công văn bản chọn lọc thích ứng mà không quan tâm xem một tấn công như vậy có thể thực hiện trong một môi trường đặc biệt. Một số ứng dụng thực tế có thể hạn chế một số tương tác được phép trên một khoảng thời gian cố định, hoặc có thể được thiết kế sao cho để tính toán các MAC chỉ cho các đầu vào được tạo trong bản thân ứng dụng; các ứng dụng khác có thể cho phép truy cập tới một số không hạn chế các cặp text-MAC, hoặc cho phép kiểm tra MAC của một số không hạn chế các văn bản và chấp nhận bất kỳ xử lý thêm với một MAC đúng.

Các kiểu giả mạo *(có lựa chọn, tồn tại)*

Khi có thể giả mạo MAC (nghĩa là thuật toán MAC đã bị phá vỡ), tính nghiêm trọng của hậu quả giả mạo phụ thuộc vào trình độ điều khiển đối phương thực hiện trên giá trị x với MAC có thể bị giả mạo đó. Trình độ này là khác nhau bởi cách phân loại các giả mạo sau:

1. Giả mạo có lựa chọn – là các tấn công mà nhờ đó một đối phương có thể tạo ra một cặp text-MAC mới cho văn bản mà hắn lựa chọn (hoặc có thể từng phần dưới điều khiển của hắn). Chú ý rằng ở đây giá trị được lựa chọn là văn bản cho một MAC bị giả mạo, ngược lại trong tấn công văn bản chọn lọc giá trị chọn lọc là văn bản của cặp text-MAC được

dùng cho các mục đích phân tích (ví dụ: để giả mạo một MAC trên một văn bản phân biệt).

2. Tồn tại giả mạo – là các tấn công mà nhờ đó đối phương có thể tạo một cặp text-MAC mới, nhưng không điều khiển trên giá trị của văn bản đó.

Khôi phục khóa của bản thân khóa MAC là tấn công nguy hiểm nhất, và thường cho phép giả mạo có lựa chọn. Giả mạo MAC cho phép đối phương có một văn bản giả mạo được chấp nhận như văn bản tin cậy. Hậu quả có thể nghiêm trọng ngay trong trường hợp tồn tại giả mạo. Một ví dụ kinh điển là sự thay thế số lượng tiền đã biết là nhỏ bằng một số ngẫu nhiên được phân bố trong khoảng từ 0 đến $2^{32} - 1$. Với lý do này, các thông báo mà tính toàn vẹn hoặc tính xác thực được kiểm tra thường bị ràng buộc có cấu trúc xác định trước hoặc độ dư thừa có thể kiểm tra được cao, để cố gắng ngăn ngừa các tấn công có ý nghĩa.

Tương tự như với các thuật toán MAC, các tấn công lên các lược đồ MDC (chủ yếu là các tấn công va chạm và tiền ảnh thứ hai) có thể được phân loại là có lựa chọn hoặc tồn tại. Nếu văn bản có thể được điều khiển từng phần, thì tấn công có thể được phân loại là có lựa chọn từng phần.

4.2. Các hàm băm không có khóa

Từ quan điểm về cấu trúc, các loại hàm băm không có khóa ở đây được phân loại dựa trên bản chất của các phép tính nằm trong các hàm nén bên trong. Do đó, có ba loại hàm băm lặp phổ biến nhất được nghiên cứu cho tới nay là các hàm băm dựa trên mã khối, các hàm băm chuyên dụng và các hàm băm dựa trên số học modulo. Các hàm chuyên dụng là các hàm băm được thiết kế đặc biệt để băm với tốc độ cao và độc lập với các thành phần hệ thống khác. Bảng dưới đây tổng quát hóa độ an toàn ước lượng của tập con các thuật toán MDC được thảo luận trong phần này.

Bảng 4.1. Ước lượng độ an toàn của các thuật toán MDC

↓ Hàm băm	n	m	Tiền ảnh	Va chạm	Ghi chú
Matyas-Meyer-Oseas[a]	n	n	2^n	$2^{n/2}$	cho độ dài khóa =
MDC-2 (với DES)[b]	64	128	2.2^{82}	2.2^{54}	n
MDC-4(với DES)	64	128	2^{109}	4.2^{54}	hạng 0.5
Merkle (với DES)	106	128	2^{112}	2^{56}	hạng 0.25
MD4	512	128	2^{128}	2^{20}	hạng 0.276
MD5	512	128	2^{128}	2^{64}	Nhận xét 13
RIPEMD-128	512	128	2^{128}	2^{64}	Nhận xét 14
SHA-1, RIPEMD-160	512	160	2^{160}	2^{80}	-

[a] Giống với độ an toàn được phỏng đoán cho các hàm băm Davies-Meyer và Miyaguchi-Preneel.

[b] Độ an toàn có thể tăng nếu sử dụng mã pháp với độ dài khóa bằng với độ dài khối.

Bảng 4.1. Các cận trên về độ an toàn của các hàm băm chọn lọc, các khối thông báo n-bit được xử lý để tạo ra các giá trị băm m-bit. Số phép tính mã pháp hoặc hàm nén hiện được tin là cần thiết để tìm các tiền ảnh và các va chạm được xác định, với giả thiết không có các yếu điểm tiềm ẩn với các mã khối (các con số cho MDC-2 và MDC-4 ước tính cho DES và các tính chất khóa yếu).

Định nghĩa 4.5.

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.*

Định nghĩa 4.6.

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.

4.2.1. 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.

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:

$$H_0 = IV\ ,\quad H_i = E_{g(H_{i-1})}(x_i) \oplus x_i\ ,\quad 1 \le i \le t$$

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 $\bmod 2$ 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. 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 $\widetilde{IV}$ 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)

$IV = 0x5252525252525252$

$I\widetilde{V} = 0x2525252525252525$

(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 $h(x) = H_t \| \widetilde{H}_t$ được xác định như sau: (với $1 \le i \le t$)

$$H_0 = IV\ ,\ \ k_i = g(H_{i-1})\ ,\ C_i = E_{k_i}(x_i) \oplus x_i\ \ ,\ \ H_i = C_i^L \| \widetilde{C}^R$$
$$\widetilde{H}_0 = I\widetilde{V}\ ,\ \ \widetilde{k}_i = \widetilde{g}(\widetilde{H}_{i-1})\ ,\ \widetilde{C}_i = E_{\widetilde{k}_i}(x_i) \oplus x_i\ \ ,\ \ \widetilde{H}_i = \widetilde{C}_i^L \| C^R$$

Thuật toán MDC - 4 có thể được mô tả theo sơ đồ sau:

Hình 4.4. Thuật toán MDC – 4

4.3. Các hàm băm có khóa (MAC)

Hàm băm có khóa là hàm băm mà mục đích của chúng là xác thực thông báo, được gọi là các thuật toán mã xác thực thông báo (MAC). Nhiều thuật toán trong số này dựa trên mã khối. Những thuật toán này có

độ dài bit MAC khá ngắn (ví dụ 32 bit với MAA) hoặc khóa ngắn (ví dụ: 56 bit với các thuật toán MAC dựa trên DES-CBC) có thể vẫn đáp ứng đủ độ an toàn, phụ thuộc vào tài nguyên tính toán có thể của đối phương và môi trường ứng dụng cụ thể.

Nhiều thuật toán MAC lặp có thể được mô tả như hàm băm lặp. Trong trường hợp này, khóa MAC là phần chung của biến đổi đầu ra g; nó cũng có thể là đầu vào cho hàm nén trong lặp đầu tiên, và được bao gồm trong hàm nén *f* ở mỗi bước.

Thực tế sau đưa ra cận trên về độ an toàn chung của các thuật toán MAC.

Thực tế về tấn công ngày sinh trên các thuật toán MAC: Giả sử h là một thuật toán MAC dựa trên hàm nén lặp, có các biến chuỗi n-bit, và là tất định (ví dụ, kết quả m-bit được quyết định đầy đủ bởi thông báo). Khi đó, việc giả mạo MAC có thể sử dụng $O(2^{n/2})$ cặp text-MAC đã biết cộng với v cặp text-MAC chọn lọc (phụ thuộc vào h) là giữa 1 và khoảng 2^{n-m}.

4.3.1. MAC dựa trên các mật mã khối

Thuật toán MAC dựa trên CBC:

Thuật toán MAC được dùng thông dụng nhất là dựa trên mã khối sử dụng chế độ móc xích khối mã. Khi DES được dùng làm mã khối E, có nghĩa là n = 64, và khóa MAC là 56-bit khóa của DES.

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

Nhận xét (*làm mạnh thêm cho CBC-MAC*): Bước xử lý tùy chọn làm giảm nguy cơ tìm vét cạn khóa, và ngăn chặn giả mạo văn bản chọn lọc (ví dụ 8), mà không ảnh hưởng tới tính hiệu quả của các bước trung gian như sử dụng hai-khóa ba-lần-mã. Thay vì phải chống lại giả mạo như vậy ta thêm trước vào đầu vào độ dài khối trước khi tính toán MAC: hoặc sử dụng khóa K để mã hóa độ dài m thu được $K' = E_K(m)$, trước khi sử dụng K′ làm khóa cho thông báo MAC.

4.3.2. Xây dựng MAC từ MDC

Một gợi ý chung là xây dựng thuật toán MAC từ thuật toán MDC, bằng cách đơn giản là gộp khóa bí mật k làm một phần đầu vào của MDC. Tiếp cận này là rõ ràng nhưng các giả thiết chưa được kiểm tra chứng thường được tạo bởi các tính chất mà MDC có; đặc biệt, trong khi hầu hết các MDC được thiết kế để cung cấp tính một chiều hoặc kháng va chạm, thì các yêu cầu của thuật toán MAC lại khác. Ngay trong trường hợp hàm băm một chiều ngăn ngừa khôi phục khóa bí mật được dùng làm một phần đầu vào thông báo, nó cũng không bảo đảm tính

không thể tạo MAC cho các đầu vào mới. Các ví dụ sau gợi ý rằng xây dựng thuật toán MAC từ hàm băm yêu cầu cần phải phân tích một cách cẩn thận.

Ví dụ 4.2. (*phương pháp bí mật tiền tố*): Xét một thông báo $x = x_1x_2...x_t$ và một MDC lặp h với hàm nén f, định nghĩa: $H_0 = IV$, $H_i = f(H_{i-1}, x_i)$; $h(x) = H_t$.

(1) Giả sử ta cố gắng sử dụng h làm thuật toán MAC chưa quyết định trước một khóa bí mật k, sao cho MAC đề xuất lên x là $M = h(k||x)$. Khi đó, mở rộng thông báo x bởi một khối tùy ý y, ta có thể suy ra $M' = h(k||x||y)$ như là $f(M, y)$ mà không biết khóa bí mật k (MAC nguyên thủy M coi như biến chuỗi). Điều này đúng cả với các hàm băm mà xử lý trước bằng pad các đầu vào với số chỉ về độ dài (ví dụ: MD5); trong trường hợp này, padding/khối-độ-dài z cho thông báo gốc x có thể xuất hiện như một phần của thông báo mở rộng, $x||z||y$, nhưng MAC giả mạo trên trường hợp sau có thể được suy luận.

(2) Với các lý do tương tự, sẽ không an toàn khi sử dụng một MDC để xây dựng một thuật toán MAC bằng sử dụng khóa MAC bí mật k làm IV. Nếu k gồm toàn bộ khối đầu tiên, thì với $f(IV, k)$ hiệu quả có thể được tính toán trước, chỉ ra rằng đối phương chỉ cần tìm k′ (không nhất thiết là k) sao cho $f(IV, k) = f(IV, k')$; điều này tương đương với sử dụng IV bí mật.

Ví dụ 4.3. (*phương pháp bí mật hậu tố*) Một đề xuất khác là sử dụng khóa bí mật làm hậu tố, cụ thể MAC n-bit trên x là $M = h(x||k)$. Trong trường hợp này, tấn công ngày sinh áp dụng được. Đối phương tự do chọn thông báo x (hoặc cố định trước) có thể tìm được một cặp thông báo x, x′ sao cho $h(x) = h(x')$ trong $O(2^{n/2})$ phép tính. (Việc này có thể thực hiện off-line, và không yêu cầu biết về k; giả thiết ở đây là kích cỡ của cả biến chuỗi và đầu ra cuối cùng). Việc thu được MAC M trên x theo nghĩa chính xác thì cho phép đối phương tạo ra cặp text-MAC (x′, M) cho thông báo mới x′. Chú ý rằng phương pháp này cần phải băm và sau đó mã hóa giá trị băm trong lần lặp cuối cùng; trong phương pháp này dạng

yếu của MAC, các giá trị MAC chỉ phụ thuộc vào giá trị chuỗi cuối cùng và khóa chỉ được dùng trong một bước.

Ví dụ 4.4. (*Phương pháp bọc với padding*). Với khóa k và MDC h, tính MAC trên thông báo x là: $h_k(x) = h(k||p||x||k)$. Ở đây p là một xâu được dùng để pad k vào độ dài của một khối, để đảm bảo rằng tính toán bên trong gồm ít nhất hai phép lặp. Ví dụ nếu h là MD5 và k là 128 bit, p là một xâu 384-bit.

Ví dụ 4.5. (*MAC dựa trên băm*) Với khóa k và MDC k, tính MAC trên thông báo x là $HMAC(x) = h(k \| p_1 \| h(k \| p_2 \| x))$, trong đó p_1, p_2 là các xâu khác nhau độ dài đủ để pad k thành khối đầy đủ cho hàm nén. Xây dựng tổng là khá hiệu quả mặc dù hai lần gọi h, vì chỉ xử lý trên đầu ra (ví dụ nếu h là MD5) một đầu vào hai khối, độc lập với độ dài của x.

4.4. Chữ kí số

4.4.1. Khái niệm chữ ký số

Chữ ký viết tay truyền thống gắn với tài liêu được dùng để chỉ ra cá nhân tương ứng với nó. Chữ ký được dùng hàng ngày như khi viết thư, rút tiền ở bank, ký hợp đồng,...

Lược đồ chữ ký số là phương pháp ký thông báo được lưu dưới dạng điện tử và thông báo được ký có thể truyền trên mạng máy tính. Tuy có nhiệm vụ của chữ ký, song có sự khác nhau cơ bản giữa chữ ký truyền thống và chữ ký số.

Về việc ký tài liệu: Với chữ ký truyền thống, chữ ký là bộ phận vật lý của tài liệu được ký. Tuy nhiên, chữ ký số không được gắn một cách vật lý với thông báo được ký, do đó thuật toán được dùng phải "trói" chữ ký với thông báo theo một cách nào đó.

Về việc kiểm tra: Chữ ký truyền thống được kiểm tra bằng cách so sánh nó với những chữ ký đã xác thực. Tất nhiên, phương pháp này không an toàn lắm vì nó tương đối dễ đánh lừa bởi chữ ký của người khác. Mặt khác, chữ ký số có thể được kiểm tra bằng cách dùng thuật toán kiểm tra đã biết công khai. Như vậy, "*người bất kì*" có thể kiểm tra chữ ký số. Việc sử dụng lược đồ ký an toàn sẽ ngăn chặn khả năng đánh lừa.

Điều khác nhau cơ bản khác là ở chỗ "*bản sao*" thông báo số được ký là đồng nhất với bản gốc. Mặt khác, bản sao chép tài liệu giấy đã ký thường là khác với bản gốc. Đặc điểm này có nghĩa là phải cẩn thận để ngăn chặn một thông báo đã ký số bị sử dụng lại. Chẳng hạn, nếu Bob ký thông báo số cho quyền Alice rút $100 từ tài khoản ở nhà bank của mình, anh ta chỉ muốn Alice làm việc đó một lần. Do đó, thông báo tự nó phải chứa thông tin để ngăn chặn Alice làm lại việc đó nhiều lần.

Một lược đồ chữ ký số gồm 2 thành phần: một thuật toán ký và một thuật toán kiểm tra. Bob có thể ký thông báo x nhờ thuật toán ký (bí mật) ***Sig***. Chữ ký thu được sig(x) sau đó có thể được kiểm tra nhờ thuật toán kiểm tra công cộng ***Ver***. Khi cho cặp (x,y) thuật toán kiểm tra sẽ trả lời "đúng" hoặc "sai" phụ thuộc vào việc chữ ký có đích thực không?.

Định nghĩa 4.3. *Lược đồ chữ ký là bộ năm $S = (\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, thoả mãn các điều kiện sau:*

1. *$\mathcal{P}$-tập hữu hạn các thông báo*
2. *A-tập hữu hạn các chữ ký có thể*
3. *K-tập hữu hạn các khoá(không gian khoá)*
4. *với $k \in K$, $\exists sig_k \in \mathcal{S}$ và $ver_k \in \mathcal{V}$.*

Mỗi $sig_k: P \to A$, $ver_k: P \times A \to \{true, false\}$

sao cho:

$$\mathrm{ver}(x,y) = \begin{cases} \text{true, nếu } y = \mathrm{sig}_k(x) \\ \text{false, nếu } y \neq \mathrm{sig}_k(x) \end{cases}$$

Yêu cầu:

1. $\forall k \in K$, sig_k và ver_k là các hàm thời gian đa thức.

2. ver_k: hàm công cộng; sig_k: hàm bí mật

Với mọi x, duy nhất Bob tính được chữ ký y sao cho ver(x,y)=true.

Oscar có thể thử lần lượt các y cho đến khi đạt yêu cầu đó. Vì vậy mục đích của ta là tìm các lược đồ chữ ký sao cho Oscar không đủ thời gian thực tế để thử như thế (an toàn tính toán).

4.4.2. Phân loại chữ ký số

Có nhiều cách khác nhau để phân loại chữ ký số, sau đây ta tìm hiểu các cách phân loại này:

4.4.2.1. Phân loại dựa vào các thành phần tham gia ký

a. Chữ ký số trực tiếp

Chữ ký số trực tiếp là chữ ký có các đặc điểm sau:

- Chỉ liên quan đến bên gửi và bên nhận

- Với mật mã khóa công khai

+ Dùng khóa riêng ký toàn bộ thông báo hoặc giá trị băm

+ Có thể mã hóa sử dụng khóa công khai của bên nhận

+ Quan trọng là ký trước mã hóa sau

- Chỉ có tác dụng khi khóa riêng của bên gửi được đảm bảo an ninh

+ Bên gửi có thể giả vờ mất khóa riêng. Do đó cần bổ sung thông tin thời gian và báo mất khóa kịp thời

+ Khóa riêng có thể bị mất thật. Kẻ cắp có thể gửi thông báo với thông tin thời gian sai lệch

b. Chữ ký số gián tiếp

Là chữ ký số mà :

- Có sự tham gia của một bên trọng tài

+ Nhận thông báo có chữ ký số từ bên gửi, kiểm tra tính hợp lệ của nó

+ Bổ sung thông tin thời gian và gửi đến bên nhận

- An ninh phụ thuộc chủ yếu vào bên trọng tài

+ Cần được bên gửi và bên nhận tin tưởng

- Có thể cài đặt với mã hóa đối xứng hoặc mã hóa khóa công khai

- Bên trọng tài có thể được phép nhìn thấy hoặc không nội dung thông báo

- Các kỹ thuật chữ ký số gián tiếp

(a) Mã hóa đối xứng, trọng tài thấy thông báo

(1) $X \rightarrow A$: $M \parallel E_{K_{XA}}[ID_X \parallel H(M)]$

(2) $A \rightarrow Y$: $E_{K_{AY}}[ID_X \parallel M \parallel E_{K_{XA}}[ID_X \parallel H(M)] \parallel T]$

(b) Mã hóa đối xứng, trọng tài không thấy thông báo

(1) $X \rightarrow A$: $ID_X \parallel E_{K_{XY}}[M] \parallel E_{K_{XA}}[ID_X \parallel H(E_{K_{XY}}[M])]$

(2) $A \rightarrow Y$: $E_{K_{AY}}[ID_X \parallel E_{K_{XY}}[M] \parallel E_{K_{XA}}[ID_X \parallel H(E_{K_{XY}}[M])] \parallel T]$

(c) Mã hóa khóa công khai, trọng tài không thấy thông báo

(1) $X \rightarrow A$: $ID_X \parallel E_{KR_X}[ID_X \parallel E_{KU_Y}[E_{KR_X}[M]]]$

(2) $A \rightarrow Y$: $E_{KR_A}[ID_X \parallel E_{KU_Y}[E_{KR_X}[M]] \parallel T]$

Ký hiệu: X = Bên gửi; M = Thông báo; Y = Bên nhận ; T = Nhãn thời gian ; A = Trọng tài

4.4.2.2. Phân loại dựa vào phương pháp ký

Phân loại theo phương pháp này có: Chữ ký với phần đính kèm, chữ ký khôi phục thông điệp.

a. Chữ ký số với phần đính kèm

- Yêu cầu thông điệp gốc là mộtthành phần đầu vào của quá trình thẩm định chữ ký.

- Dựa vào hàm băm mã hoá

Hình 4.6. Lược đồ chữ ký số với phần đính kèm

b. <u>Chữ ký số khôi phục thông điệp</u>

- Thông điệp gốc được khôi phục từ chính chữ ký số. Do đó nó chỉ phù hợp với những thông điệp ngắn. Việc biết trước về thông điệp là không cần thiết đối với quá trình thẩm định chữ ký.

Hình 4.7. Lược đồ chữ ký số khôi phục thông điệp

4.4.2.3. Phân loại dựa vào đặc điểm an ninh của các hệ mã hoá

Khi dựa vào đặc điểm an ninh của các hệ mã hoá được áp dụng trong các lượcđồ chữ ký số, ta có thể phân thành ba dạng như sau:

a. Lược đồ chữ ký số dựa vào độ khó của việc phân tích thừa số nguyên (IFP - Integer Factorization Problem).

b. Lược đồ chữ ký số dựa vào độ khó của vấn đề Logarit rời rạc (DLP – Discrete Logarithm Problem).

c. Lược đồ chữ ký số dựa trên độ khó của vấn đề Logarit trên đường cong Elip (ECC – Elliptic Curve Cryptography).

4.4.2.4. Phân loại dựa trên tính ứng dụng của các lược đồ ký

Thông thường, một lược đồ chữ ký số cơ bản gồm 5 thành phần chính là: Bên ký, bên thẩm định chữ ký, thông điệp, khoá công khai và khoá bí mật. Có các cách phân loại sau:

a. Phân loại dựa vào bên ký

- Chữ ký do 1 người tạo ra
- Chữ ký do 1 nhóm người tạo ra
- Chữ ký do 1 người được uỷ quyền tạo ra
- Chữ ký do 1 nhóm người được uỷ quyền tạo ra
- v.v.v.

b. Phân loại dựa vào bên thẩm định chữ ký

- Bất kỳ ai cũng có thể thực hiện thẩm định chữ ký
- Chỉ những người được chỉ định mới có thể thực hiện thẩm định chữ ký
- Việc thẩm định chữ ký được thực hiện nếu có sự giúp đỡ của người ký
- v.v.v.

c. Phân loại dựa vào nội dung thông điệp

- Người ký biết nội dung thông điệp
- Người ký không biết nội dung thông điệp

d. Phân loại dựa vào phương pháp tạo khoá công khai

- Khoá công khai của người ký được chứng thực và được đưa ra công khai bởi một trung tâm chứng thực.
- Khoá công khai được đưa ra bởi chính người ký.

e. Phân loại dựa vào phương pháp cập nhật khoá bí mật

4.4.3. Xác thực giữa những người sử dụng

Ở trên ta đã trình bày lược đồ xác thực giữa Alice và Bob. Tuy nhiên, lược đồ đó có hai nhược điểm lớn:

Một là, nếu Bob tự nghĩ ra một thông báo, tính vết nhờ khoá k (đã thoả thuận với Alice) rồi nói rằng thông báo đó do Alice gửi cho mình thì toà án không thể phân xử được vì hai người đều có khoá k.

Hai là, trong mạng nhiều người sử dụng, nếu mỗi cặp có một khoá thoả thuận như vậy thì mỗi người phải lưu giữ n-1 khoá bí mật. Khi n đủ lớn, đó là một việc phiền phức, phức tạp.

Vì vậy, người ta nghiêng về việc sử dụng các lược đồ chữ ký số.

Alice sẽ chọn thuật toán ký bí mật sig_A của mình, công bố thuật toán kiểm tra chữ ký công cộng ver_A ở thư mục công cộng. Mỗi người sử dụng cũng đều làm như vậy.

Khi gửi thông báo m cho Bob, Alice tính $s = sig_A(m)$, rồi gửi cặp (m,s) cho Bob.

Nhận được (m,s), Bob tính $ver_A(s)$. Nếu $m = ver_A(s)$ thì Bob chấp nhận m là thông báo do Alice gửi cho mình. Nếu có tranh cãi, Bob sẽ trình cặp (m,s) cùng thuật toán ver_A cho toà án. Toà án cũng tính $ver_A(s)$. Nếu đúng, toà sẽ tuyên bố m là thông báo do Alice gửi cho Bob.

Sở dĩ toà án làm như vậy là vì hai điều:

Một là: từ ver_A không ai có thể tính được sig_A ngoài Alice. Do đó chỉ một mình Alice biết sig_A.

Hai là: chỉ có Alice mới tính được s thoả mãn: $m = ver_A(s)$.

4.4.4. Kết hợp chữ ký số và mã hoá

Người ta có thể kết hợp mã hoá với xác thực. Cho bản rõ x, Alice tính chữ ký số $y=sig_A(x)$ rồi mã hoá cả x cả y bởi thuật toán mã hoá công khai e_B của Bob để được $z=e_B(x,y)$ và truyền z cho Bob.

Với z, trước tiên Bob giải mã nó nhờ thuật toán giải mã d_B để được (x,y). Sau đó kiểm tra xem có xảy ra $ver_A(y)=x$ hay không ?.

Điều gì xảy ra nếu Alice làm như sau:

- Tính $z=e_B(x)$.
- Ký $y=sig_A(z)$.
- Gửi (y,z) cho Bob.

Tất nhiên, Bob làm như sau:

- Giải $x=d_B(z)$.
- Kiểm tra $ver_A(y)=z$.

Nhưng Oscar sẽ tiến hành:

- Thu nhận (y,z).
- Ký $y'=sig_{Oscar}(z)$.
- Gửi (y',z) cho Bob.

Khi đó Bob hiểu (y',z) do Oscar gửi cho mình và kiểm tra bằng ver_{Oscar}. Tiếp đến, Bob tin rằng x là của Oscar gửi cho mình. Tất nhiên, nếu x là "Tôi nợ Bob $1000" thì Oscar thiệt to! Song mấy khi có bức điện như thế!

Do khó khăn này, hầu hết chúng tađều được nhắc nhở rằng: ký trước, mã sau.

4.5. Các lược đồ chữ ký số thông dụng

4.5.1. Lược đồ RSA

Lược đồ chữ ký RSA được cho bởi bộ năm

$$S = (\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V}),$$

trong đó, $\mathcal{P} = \mathcal{A} = Zn$, với $n = p.q$ là tích của hai số nguyên tố lớn p, q, $\mathcal{K}$ là tập các cặp khoá $k = (k', k'')$, với $k' = a$ và $k'' = (n,b)$, a và b là hai số thuộc Z_n^* thoả mãn $a.b \equiv 1(\text{mod}\,\phi(n))$. Các hàm $sig_{k'}$ và $ver_{k''}$ được xác định như sau:

$$sig_{k'}(x) = x^a \text{ modn},$$

$$ver_{k''}(x,y) = \text{đúng} \Leftrightarrow x \equiv y^b \text{ (modn)}.$$

Dễ chứng minh được rằng lược đồ được định nghĩa như vậy là hợp thức, tức là với mọi $x \in \mathcal{P}$ và mọi chữ ký $y \in \mathcal{A}$:

$$ver_{K''}(x, y) = \text{đúng} \Leftrightarrow y = sig_{K'}(x).$$

Hai vấn đề xác nhận và bảo mật theo lược đồ RSA có bề ngoài giống nhau, nhưng nội dung của chúng là hoàn toàn khác nhau: Khi A gửi thông báo x cho B, để B có căn cứ xác nhận đó đúng thực là thông báo do A gửi, A phải gửi kèm theo chữ ký $sig_{K'}(x)$, tức là A gửi cho B (x, $sig_{K'}(x)$), trong các thông tin gửi đi đó, thông báo x hoàn toàn không được giữ bí mật. Cũng tương tự như vậy, nếu dùng sơ đồ mật mã RSA, khi một chủ thể A nhận được một bản mật mã $e_{K'}(x)$ từ B thì A chỉ biết rằng thông báo x được bảo mật, chứ không có gì để xác nhận x là của B.

Nếu ta muốn vừa đảm bảo tính bảo mật vừa có tính xác thực, thì ta phải sử dụng đồng thời cả hai hệ mật mã và xác nhận (bằng chữ ký).

4.5.2. Lược đồ Elgamal

Lược đồ chữ ký ElGamal được đề xuất năm 1985, gần như đồng thời với sơ đồ hệ mật mã ElGamal, cũng dựa trên độ khó của bài toán lôgarit rời rạc.

$$S = (\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V}),$$

Trong đó $\mathcal{P} = Z_p^*$, $\mathcal{A} = Z_p^* \times Z_{p-1}$, với p là một số nguyên tố sao cho bài toán tính lôgarit rời rạc trong Z_p^* là rất khó. Tập hợp $\mathcal{K}$ gồm các cặp khoá $k = (k', k'')$, với $k' = a$ là một số thuộc Z_p^*, $k'' = (p, \alpha, \beta)$, α là một phần tử nguyên thuỷ của Z_p^*, và $\beta = \alpha^a \bmod p$. k' là khoá bí mật dùng để ký, và k'' là khoá công khai dùng để kiểm thử chữ ký. Các thuật toán ký và kiểm thử chữ ký được xác định như sau: Với mỗi thông báo x, để tạo chữ ký trên x ta chọn thêm một số ngẫu nhiên $d \in Z_{p-1}^*$, rồi tính

$sig_{K'}(x,d) = (\gamma, \delta)$, với

$$y = \alpha^d \bmod p,$$

$$\delta = (x-ay).d^{-1} \bmod (p-1).$$

Thuật toán kiểm thử được định nghĩa bởi:

$ver_{K'}(x, (\gamma, \delta))$ = đúng $\Leftrightarrow \beta^{\gamma}.\gamma^{\delta} \equiv \alpha^x \pmod p$.

Dễ thấy rằng lược đồ chữ ký được định nghĩa như trên là hợp thức. Thực vậy, nếu $sig_{K'}(x,d) = (\gamma, \delta)$, thì ta có :

$$\beta^{\gamma}.\gamma^{\delta} \equiv \alpha^{a\gamma}.\alpha^{d\delta} \bmod p$$

$$\equiv \alpha^x \bmod p,$$

vì $d\delta + a\gamma \equiv x \bmod (p-1)$. Do đó, $ver_{K'}(x, (\gamma, \delta))$ = đúng.

4.5.3. Lược đồ chữ ký số chuẩn DSS

Chuẩn chữ ký số (DSS) được đề xuất từ năm 1991 và được chấp nhận vào cuối năm 1994 để sử dụng trong một số lĩnh vực giao dịch điện tử tại Hoa kỳ. DSS dựa vào lược đồ chữ ký ElGamal, với một vài sửa đổi. Để bảo đảm an toàn, số nguyên tố p cần phải đủ lớn, biểu diễn nhị phân của p phải có từ 512 bit trở lên. Tuy nhiên, độ dài chữ ký theo lược đồ ElGamal là gấp đôi số bit của p, mà trong nhiều ứng dụng người ta lại mong muốn có chữ ký độ dài ngắn, nên giải pháp sửa đổi được đề xuất là: trong khi vẫn dùng p lớn với độ dài biểu diễn 512 bit trở lên, thì sẽ hạn chế độ dài của γ và δ trong chữ ký (γ,δ) vào khoảng 160 bit (như vậy cả chữ ký sẽ có độ dài khoảng 320 bit); điều này được thực hiện bằng cách dùng một nhóm con cyclic Z_q^* của Z_p^* thay cho chính bản thân Z_p^*, do đó mọi tính toán vẫn được thực hiện như trong Z_p^* nhưng các dữ liệu và thành phần chữ ký lại thuộc Z_q^*. Ta được *lược đồ chuẩn chữ ký số DSS* như mô tả sau đây:

Chọn p là một số nguyên tố lớn có độ dài biểu diễn ≥ 512 bit sao cho bài toán tính logarit rời rạc trong Z_p là khó, q là một ước số nguyên tố của

p -1, có độ dài biểu diễn cỡ 160 bit. Gọi $\alpha \in Z_p^*$ là một căn bậc q của 1 theo mod p.

Đặt $\mathcal{P} = Z_p^*$, $\mathcal{A} = Z_q^* \times Z_q^*$. Chọn $a \in Z_q^*$ và tính $\beta \equiv \alpha^a \bmod p$.

Xác định khoá $k = (k', k'')$, trong đó khoá bí mật $k' = a$, và khoá công khai $k'' = (p, q, \alpha, \beta)$. Thuật toán ký và thuật toán kiểm thử được định nghĩa như sau: Với $x \in \mathcal{P} = Z_p^*$, ta chọn thêm một số ngẫu nhiên d ($0 \le d \le q$ -1), và định nghĩa chữ ký

$$sig_{K'}(x, d) = (y, \delta),$$

Trong đó $y = \alpha^d (\bmod p) \bmod q$,

$\delta = (x + ay).d^{-1} \bmod q$.

Thuật toán kiểm thử được định nghĩa bởi:

$$ver_{K'}(x, (y, \delta)) = \text{đúng} \Leftrightarrow (\alpha^{e1} . \beta^{e2} \bmod p) \bmod q = \gamma,$$

trong đó $e_1 = x.\delta^{-1} \bmod q$ và $e_2 = y.\delta^{-1} \bmod q$.

Chú ý rằng ta phải có $\delta \neq 0 \bmod q$ để có thể tính được $\delta^{-1} \bmod q$ dùng trong thuật toán kiểm thử, vì vậy nếu chọn d mà được $\delta \equiv 0 \bmod q$ thì phải chọn lại số d khác để có được $\delta \neq 0 \bmod q$.

4.5.4. Lược đồ chữ ký số trên EC

4.5.4.1. Lược đồ chữ ký ECDSA

Để thiết lập lược đồ chữ ký ECDSA, cần xác định các tham số: lựa chọn đường cong E trên trường hữu hạn F_q với đặc số p phù hợp, điểm cơ sở $G \in E(F_q)$.

Một số khuyến nghị khi lựa chọn các tham số:

1. Kích thước q của trường, hoặc $q = p$ ($p > 2$) hoặc $q = 2^m$.

2. Hai phần tử a, b thuộc F_q xác định phương trình đường cong elliptic:

$y^2 = x^3 + ax + b$ ($p > 2$) hoặc $y^2 + xy = x^3 + ax^2 + b$ ($p = 2$)

3. Hai phần tử x_G và y_G thuộc F_q xác định điểm cơ sở $G = (x_G, y_G)$

4. Bậc n của điểm G với n>max($n > 4\sqrt{q}$, 2^{160})

Sinh khóa

1. Chọn số ngẫu nhiên d trong khoảng [2, n – 1] làm khóa bí mật.

2. Tính $Q = dG$ làm khóa công khai.

Ký trên bản rõ *m*

1. Chọn một số ngẫu nhiên k, $2 \le k \le n-1$

2. Tính $kG = (x_1, y_1)$.

3. Tính $r = x_1 \bmod n$. Nếu $r = 0$, quay lại bước 1.

4. Tính $k^{-1} \bmod n$.

5. Tính $s = k^{-1}(m + dr) \bmod n$. Nếu $s = 0$ quay lại bước 1.

6. Chữ ký trên thông điệp m là (r, s)

Kiểm tra chữ ký

1. Kiểm tra r và s có là các số tự nhiên trong khoảng [2, n – 1] không.

2. Tính $w = s^{-1} \bmod n$

3. Tính $u_1 = mw \bmod n$ và $u_2 = rw \bmod n$

4. Tính $X = u_1G + u_2Q = (x_X, y_X)$

5. Nếu $X = O$ thì phủ nhận chữ ký. Ngược lại tính $v = x_X \bmod n$.

6. Chữ ký chỉ được chấp nhận nếu $v = r$.

Chứng minh

Nếu chữ ký (r, s) trên m là đúng thì $s = k^{-1}(m + dr) \bmod n$.

$$k \equiv s^{-1}(m + dr) \equiv s^{-1}e + s^{-1}rd \equiv wm + wrd \equiv u_1 + u_2d \pmod n.$$

Vì vậy, $u_1G + u_2Q = (u_1 + u_2d)G = kG$, và vì vậy $v = r$.

4.5.4.2. Lược đồ ký mù Harn trên EC

Năm 1994, Harn đã công bố một lược đồ chữ ký mù tựa ECDSA. Chữ ký mù là chữ ký thực hiện trên một văn bản mà người ký hoàn toàn không biết nội dung. Điều này thực hiện được vì người trình ký đã sử dụng một phương pháp nào đó để che dấu nội dung của văn bản gốc để

người ký không biết. Để người ký yên tâm, người xin cấp chữ ký phải chứng minh tính hợp lệ của nội dung đã bị che giấu.

Lược đồ chữ ký mù Harn trên EC

Sinh khóa

Chọn các tham số cho đường cong Elliptic

(1) Chọn số nguyên tố p và số nguyên n.

(2) Với 2 phần tử a_1, a_2 của $GF(p^n)$, xác định phương trình của E trên $GF(p^n)$ ($y^2 = x^3 + a_1x + a_2$ trong trường hợp $p>3$) với $4a_1^3 + 27a_2^2 \neq 0$

(3) Với 2 phần tử x_G và y_G trong $GF(p^n)$ xác định một điểm $G = (x_G, y_G)$ trên $E(GF(p^n))$ ($G \neq O$ với O là điểm gốc).

(4) Giả sử điểm G có bậc q

Việc sinh khóa bao gồm:

(1) Chọn một khóa bí mật d là số nguyên ngẫu nhiên trong $[2, q-1]$

(2) Tính khóa công khai Q, là một điểm trên E sao cho $Q = dG$.

Ký mù

Giả sử Bob yêu cầu Alice ký lên một văn bản m_0 mà m là đại diện của văn bản này ($m = H(m_0)$ với H là một hàm băm nào đó). Giao thức ký được thực hiện như sau:

(1) Alice sinh ra cặp khóa $(\bar{k}, \bar{R})$ theo cách sau: chọn ngẫu nhiên $\bar{k} \in [2, q-1]$ và tính $\bar{R} = \bar{k}G = (x_{\bar{k}}, y_{\bar{k}})$. Đặt $\bar{r} = x_{\bar{k}}$, rồi gửi $\bar{r}$ và $\bar{R}$ cho Bob

(2) Bob chọn các tham số làm mù $a, b \in [1, q-1]$, tính R trên E sao cho $R = a\bar{R} + bG = (x_k, y_k)$ và tính $r = c(x_k)$ và $\bar{m} = (m + r)a^{-1} - \bar{r}$. Sau đó gửi $\bar{m}$ cho Alice ($\bar{m}$ là m sau khi đã bị làm mù).

(3) Alice tính $\bar{s} = d(\bar{m} + \bar{r}) + \bar{k} \bmod q$, rồi gửi $\bar{s}$ cho Bob.

(4) Bob nhận được $\bar{s}$, xóa mù để có được chữ ký s trên m bằng cách tính $s = a\bar{s} + b$

Cặp (r, s) là một chữ ký trên m.

Chứng minh

Cặp (r, s) là một chữ ký Harn của thông điệp m và lược đồ ký trên là một lược đồ chữ ký mù trên đường cong elliptic.

Việc xác minh tính hợp lệ của chữ ký Harn được thực hiện như sau:

(1)Tìm một điểm V trên E sao cho $sG - (m + r)Q = (x_v, y_v)$.

(2)Kiểm tra $r \overset{?}{=} x_v \pmod q$. Nếu đúng thì (r, s) là chữ ký hợp lệ.

Để chứng minh giao thức trên thực sự tạo ra chữ ký có tính chất "mù", chúng ta chỉ ra rằng mỗi người ký có một cặp duy nhất (a, b) là tham số làm mù, với $a, b \in [1, q-1]$. Với $\overline{R}, \overline{k}, \overline{r}, \overline{m}, \overline{s}$ và một chữ ký hợp lệ (r, s) của m ta có:

$$a = (m + r)(\overline{m} + \overline{r})^{-1} \pmod q \quad b = s - a\overline{s} \pmod q$$

Ta phải chứng minh: $R = a.\overline{R} + bG$. Thực vậy,

$$a\overline{R} + bG = a\overline{k}G + sG - a\overline{s}G = a\overline{k}G + sG - aG(d\overline{m} + d\overline{r} + \overline{k})$$

$$= sG - adG((m + r)a^{-1} - \overline{r}) - ad\overline{r}G$$

$$= sG - dmG - drG = sG - (m + r)Q = R$$

4.6. Một số lược đồ chữ ký khác

4.6.1. 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 – 2k bit (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 4.8. Xác thực thông báo dùng sơ đồ chữ kí

4.6.1.1. 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:

$$s_i = \sum_{i=1}^{k} m_i h_{ij} \quad \text{với } 1 \le j \le 2k$$

$$s_i \in [0, k]$$

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 $\{h_{i,j}\}$

4.6.1.2. 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 4.3. 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.

$$S' = M'^{r} \times H + R$$

$$S' = \left(m'_k, \ldots, m'_1\right)\begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,2k-1} & h_{1,2k} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,2k-1} & h_{2,2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{k,1} & h_{k,2} & \cdots & h_{k,2k-1} & h_{k,2k} \end{pmatrix} + \left(r_1, \ldots, r_{2k}\right)$$

$$S' = \left(s_1, \ldots, s_{2k}\right) + \left(r_1, \ldots, r_{2k}\right)$$

$$S' = \left(s'_1, \ldots, s'_{2k}\right)$$

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 4.4. 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 2k bít 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:

$$M = S' \times A = (2,1,0,0,1,2) \times \begin{pmatrix} 1 \\ 3 \\ 4 \\ 4 \\ 1 \\ 2 \end{pmatrix} \bmod 7 = 10 \bmod 7 = 3$$

4.6.2. 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:

$$\begin{aligned}
S_1^2 + \left(K_A S_2^2\right) \bmod n_A &= \left[2^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]\right]^2 + K_A\left[2^{-1}k_A\left[\left(MR_A^{-1}\right) - R_A\right]\right]^2 \bmod n_A \\
&= 4^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]^2 + 4^{-1}K_A k_A^2\left[\left(MR_A^{-1}\right) - R_A\right]^2 \bmod n_A \\
&= 4^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]^2 - 4^{-1}k_A^2 k_A^{-2}\left[\left(MR_A^{-1}\right) - R_A\right]^2 \bmod n_A \\
&= 4^{-1}\left[\left(MR_A^{-1}\right) + R_A\right]^2 - 4^{-1}\left[\left(MR_A^{-1}\right) - R_A\right]^2 \bmod n_A \\
&= 4^{-1}\left[M^2R_A^{-2} + 2MR_A^{-1}R_A + R_A^{-2}\right] - \left[M^2R_A^{-2} - 2MR_A^{-1}R_A + R_A^{-2}\right] \bmod n_A \\
&= 4^{-1}\left(M^2R_A^{-2} + 2M + R_A^{-2} - M^2R_A^{-2} + 2M - R_A^{-2}\right) \bmod n_A \\
&= 4^{-1}(2M + 2M) \bmod n_A \\
&= M
\end{aligned}$$

Ví dụ 4.5. 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 :

$$S_1^2 + \left(K_A S_2^2\right) \bmod n_A = 22^2 + \left(14.18^2\right) \bmod 27$$
$$= 484 + 14.324 \bmod 27$$
$$= 5020 \bmod 27$$
$$= 25 = M$$

4.6.3. Các chữ ký 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.

4.6.3.1. Nén chữ kí

Hình 4.8. 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 4.9. Vòng nén chữ kí

4.6.3.2. 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:

$$i = 1 \Rightarrow (K_{1,0}, K_{1,1})$$
$$i = 2 \Rightarrow (K_{2,0}, K_{2,1})$$
$$\vdots$$
$$i = n \Rightarrow (K_{n,0}, K_{n,1})$$

(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 4.6. Nếu thông báo M là :

$$M = m_1 \quad m_2 \quad m_3 \quad m_4 \quad \ldots \quad m_{n-1} \quad m_n$$
$$M = 1 \quad 0 \quad 0 \quad 1 \quad \ldots \quad 1 \quad 1$$

Thì chữ ký SG(M) là:

$$SG(M) = K_{1,i_1} \quad K_{2,i_2} \quad K_{3,i_3} \quad K_{4,i_4} \quad \ldots \quad K_{n-1,i_{n-1}} \quad K_{n,i_n}$$
$$SG(M) = K_{1,1} \quad K_{2,0} \quad K_{3,0} \quad K_{4,1} \quad \ldots \quad K_{n-1,1} \quad K_{n,1}$$

Sơ đồ chữ ký Diffie-Lamport được mô tả trên hình sau:

Hình 4.10. 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.

$$E_{K_{1,i_1}}\left(S_{1,i_1}\right) \stackrel{?}{=} R_{1,i_1}$$

$$E_{K_{2,i_2}}\left(S_{2,i_2}\right) \stackrel{?}{=} R_{2,i_2}$$

$$\vdots$$

$$E_{K_{n,i_n}}\left(S_{n,i_n}\right) \stackrel{?}{=} R_{n,i_n}$$

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 4.10 chỉ ra quá trình kiểm tra chữ ký.

Hình 4.11. 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.

4.7. Ứng dụng của chữ ký số

4.7.1. Ứng dụng của chữ ký số

Với chữ ký tay trên văn bản, người nhận rất khó có thể kiểm tra được độ chính xác, tính xác thực của chữ ký. Tình trạng sử dụng chữ ký giả rất dễ xảy ra bởi không cần phải làm đăng ký cho loại chữ ký này (trừ chữ ký của những nhân vật cao cấp). Tuy nhiên, với chữ ký số, người sử dụng phải đăng ký, vừa được đảm bảo độ an toàn, chính xác bằng công nghệ hiện đại, vừa được xác thực bởi các tổ chức chứng thực... Do đó, độ an toàn của chữ ký số cao hơn rất nhiều so với chữ ký tay truyền thống.

Chính bởi những ưu điểm đó, cùng với sự phát triển nhanh chóng của môi trường giao dịch điện tử, chữ ký điện tử đã được công nhận và sử dụng rộng rãi tại nhiều nước trên thế giới. Ngoài các nước phát triển như Mỹ, EU, Singapore, Nhật Bản, Hàn Quốc..., chữ ký điện tử cũng đã được các nước Trung Quốc, Ấn Độ, Brazil... công nhận và sử dụng. Ở Việt Nam, chữ ký điện tử cũng đã được sử dụng chính thức trong các giao dịch của ngành tài chính, ngân hàng.

4.7.2. Luật về chữ ký số của một số nước trên thế giới

Để hỗ trợ các hoạt động thương mại điện tử, nhiều nước trên thế giới đều đã xây dựng khung pháp lý riêng, dựa trên những khái niệm và những nguyên tắc cơ bản của bộ luật mẫu về Thương mại điện tử của Uỷ Ban Pháp luật thương mại quốc tế - Liên hợp quốc (UN Commision on International Trade Law - UNCITRAL) soạn thảo năm 1996. Bộ luật mẫu này cung cấp các nguyên tắc có tính quốc tế, giải quyết một số trở ngại, nhằm tạo ra môi trường an toàn về pháp lý cho các hoạt động thương mại điện tử.

- **Australia:** Luật giao dịch điện tử năm 1999 (căn cứ trên luật mẫu về TMĐT của UNCITRAL) quy định các nghĩa vụ pháp lý với việc phát hành đối với phương tiện điện tử

- **Nhật Bản:** Hàng loạt luật liên quan đến công nghệ thông tin ban hành trong năm 2000 công nhận tính hiệu lực của việc chuyển các văn

bản bằng phương tiện điện tử. Luật về chữ ký điện tử và tổ chức chứng thực điện tử của Nhật Bản cũng được ban hành ngày 25/5/2000.

- **Trung Quốc:** Luật hợp đồng thừa nhận tính hiệu lực của các hợp đồng điện tử.

- **Đặc khu Hongkong:** Ngày 7/1/2000, Hồng Kông đã ban hành pháp lệnh giao dịch điện tử. Văn bản này có quy định về chữ ký điện tử, bản ghi điện tử và được áp dụng rộng rãi cho mọi hoạt động truyền thông, công nhận tính pháp lý của các giao dịch điện tử.

- **Hàn Quốc:** Hàn Quốc có Luật Chữ ký điện tử vào năm 1999 và sửa đổi vào năm 2001.

- **Mehico:** Nghị định về Thương mại điện tử được thông qua năm 2000.

- **New Zealand:** Luật Giao dịch điện tử ban hành năm 1998, xác định quyền và nghĩa vụ của các bên tham gia vào một giao dịch điện tử.

- **Thái Lan:** Luật Giao dịch điện tử của Thái Lan được thông qua vào tháng 10/2000 đã bao quát cả chữ ký điện tử.

- **Mỹ:** Áp dụng Luật thương mại chung; Áp dụng Luật Chuyển tiền điện tử đối với các sản phẩm lưu trữ giá trị dưới sự kiểm soát của Cục Dự trữ Liên bang; Luật Giao dịch điện tử thống nhất thông qua năm 1999 thừa nhận tính bình đẳng của chữ ký điện tử và chữ ký viết tay. Các bang ban hành luật riêng dựa trên luật giao dịch điện tử thống nhất.

- **Malaysia:** Ngày 1/10/1998, Luật về chữ ký điện tử của Malaysia đã có hiệu lực.

- **Singapore:** Ngày 29/6/1998, Luật giao dịch điện tử của Singapore đã ra đời quy định về chữ ký điện tử, chữ ký số cũng như bản ghi điện tử.

- **Philipines:** Luật Thương mại điện tử của Philipines ban hành ngày 14/6/2000 đã điều chỉnh về chữ ký điện tử, giao dịch điện tử.

- **Brunei:** Luật Giao dịch điện tử của Brunei được ban hành tháng 11/2000 bao quát đến vấn đề hợp đồng điện tử cũng như chữ ký điện tử và chữ ký số.

- **Ấn Độ:** Luật về công nghệ thông tin của Ấn Độ được thi hành từ tháng 10/2000 quy định về chữ ký số và bản ghi điện tử.

- **Áo:** Luật chữ ký, 2000

- **Anh, Scotland và Wales:**Luật thông tin điện tử, 2000

- **Đức:** Luật chữ ký, 2001

- **Nauy:** Luật chữ ký điện tử, 2001

- **Tây ban nha:** Luật chữ ký điện tử, 2003

- **Thụy điển:** Luật chữ ký điện, tử 2000

- **Thụy sỹ:** Luật liên bang về dịch vụ chứng thực liên quan tới chữ ký điện tử, 2003.

4.7.3. Chữ ký số tại Việt Nam

Việt Nam có Luật giao dịch điện tử số 51/2005/QH11, được Quốc hội khoá XI thông qua ngày 29/11/2005 tại kỳ họp thứ 8, chính thức có hiệu lực từ ngày 01/03/2006.

Luật gồm 8 chương, với 54 điều bao gồm hầu hết các yếu tố, bên liên quan đến giao dịch điện tử như: Chữ ký điện tử, tổ chức cung cấp dịch vụ chứng thực chữ ký điện tử, giá trị pháp lý chữ ký điện tử, giá trị pháp lý của hợp đồng ký bằng chữ ký điện tử, trách nhiệm các bên liên quan đến bảo mật thông tin, giải quyết tranh chấp liên quan đến giao dịch điện tử cũng như quy định về giao dịch điện tử trong hoạt động của các cơ quan nhà nước; lĩnh vực dân sự, kinh doanh, thương mại và các lĩnh vực khác do pháp luật quy định.

Các quy định khác

- Nghị định của Chính phủ số 26/2007/NĐ-CP ngày 15 tháng 02 năm 2007 Quy định chi tiết thi hành luật Giao dịch điện tử về chữ ký số và Dịch vụ chứng thực chữ ký số

- Nghị định của Chính phủ số 27/2007/NĐ-CP ngày 23 tháng 02 năm 2007 về Giao dịch điện tử trong hoạt động tài chính.

- Ngày 27/07/2006, Bộ Thương Mại ban hành Quyết định số 25/2006/QĐ-BTM về Quy chế sử dụng chữ ký số của Bộ Thương mại

- Ngày 30/07/2007, Bộ Thương Mại ban hành Quyết định số 018/2007/QĐ-BTM về Quy chế cấp chứng nhận xuất xứ điện tử.

- Ngày 31/12/2008, Bộ thông tin và Truyền thông ban hành **Danh mục tiêu chuẩn** bắt buộc áp dụng về chữ ký số và dịch vụ chứng thực chữ ký số. (*Tiêu chuẩn được lựa chọn cho giải thuật chữ ký số: RSA; lựa chọn cho giải thuật băm an toàn: SHA, MD5*)

Tình hình ứng dụng chữ ký số ở Việt Nam

Khả năng ứng dụng của chữ ký số khá lớn, do có tác dụng tương tự như chữ ký tay, nhưng dùng cho môi trường điện tử. Thường chữ ký số được sử dụng trong giao dịch cần an toàn qua mạng Internet, như giao dịch thương mại điện tử, tài chính, ngân hàng. Ngoài ra còn dùng để ký lên email, văn bản tài liệu Soft-Copy, phần mềm... module phần mềm và việc chuyển chúng thông qua Internet hay mạng công cộng.

Theo quyết định số 25/2006/QĐ-BTM về quy chế sử dụng chữ ký số của bộ Thương Mại, mọi văn bản điện tử được ký bằng chữ ký số có giá trị pháp lý tương đương văn bản giấy được ký và đóng dấu. Ngoài ra, nghị định 26 về chữ ký số và dịch vụ chứng thực chữ ký số đã được Chính Phủ ban hành ngày 15/2/2007, qua đó công nhận chữ ký số và chứng thực số có giá trị pháp lý trong giao dịch điện tử, bước đầu thúc đẩy sự phát triển của thương mại điện tử tại Việt Nam.

Hiện nay nhiều ngân hàng Việt Nam đã ứng dụng chữ ký số trong các hệ thống như Internet Banking, Home Banking hay hệ thống bảo mật nội bộ. Ngoài ra các website của các ngân hàng, công ty cần bảo mật giao dịch trên đường truyền, mạng riêng ảo VPN đã áp dụng chữ ký số. Có thể nói, càng ngày càng nhiều sự hiện diện của chữ ký số trong các hệ thống, ứng dụng Công nghệ thong tin bảo mật của doanh nghiệp, tổ chức ở Việt Nam.

4.8. Bài tập

1. Giả sử Alice và Bob dùng chung lược đồ chữ ký số RSA với n=143

(p=11, q=13); $\phi(n)=120$.

a. Nếu b=7 là khoá ký công khai của Bob, hãy tính khoá ký bí mật a của Bob.

b. Nếu thông báo cần ký của Bob x=110 thì chữ ký số của Bob là gì?

c. Biết rằng chữ ký của Bob trên x=85 là 6. Hãy kiểm tra xem chữ ký đó có đáng tin cậy không (Tức là chứng minh 6 là chữ ký của Bob trên x=85).

2. Giả sử Alice và Bob dùng lược đồ chữ ký số ElGamal với p=467, α=2 và khoá bí mật a=127 ;

a. Với thông báo x=50, Bob chọn ngẫu nhiên r =3. Hãy tính chữ ký số của Bob trên x=50.

b. Alice nhận được thông báo của Bob trên x=200 là (32;66), (Vẫn với khoá ký a=127).

Hãy chứng minh rằng đó là chữ ký của Bob trên thông báo x=200.

3. Giả sử Bob sử dụng hệ mật ElGamal với p=31847, α=5 và a=7899, β=18074. Hãy giải các bản mã sau của Alice gửi cho Bob: (3781,14409), (31552, 3930).

4. Giả sử Alice dùng hệ RSA với p=101, q=113, khi đó n=11413,

ϕ (n)=11200. Alice chọn a=7467 do đó b=3. Hãy giải bản mã c=8165 do Bob gửi cho Alice.

5. Giả sử Bob đang dùng lược đồ chữ ký ElGamal với p=31847, α=5 và β=25703. Cho 2 chữ ký của Bob là (23972,31396), đối với thông báo x=8990 và (23972, 20841) đối với thông báo x=31415. Hãy tính số a và r của Bob đã dùng để ký 2 thông báo trên.

6. Giả sử Bob đang dùng hệ ElGamal với p=467, α=2 và β=450. Biết rằng Bob ký thông báo x=100 với số r=31 và cho chữ ký là (26, 216). Hãy tìm số mũ bí mật a của Bob.

7. Cho e là đường cong Elip:

$y^2 = x^3 + x + 28$, xác định trên Z_{71}.

a. Hãy xác định số các điểm trên E

b. Bậc cao nhất của các phần tử trong E là bao nhiêu. Hãy tìm phần tử có bậc đó.

8. Cho E là đường cong Elip:

$y^2 = x^3 + x + 13$, xác định trên Z_{31}.

Có thể chỉ ra rằng #E=34 và $\alpha = (9,10)$, là phần tử bậc 34 trong E. Hệ mật Menezen-Vanstone được xác định trên E sẽ có không gian rõ là $Z_{34}^* \times Z_{34}^*$. Giả sử số mũ bí mật của Bob là a=25.

a. Tính $\beta = a\alpha$

b. Hãy giải dòng mã sau:

((4,9), 28 ,27), ((19,28), 9, 13), ((5,22), 20, 17), ((25,16), 12, 27)

c. Giả sử mỗi bản rõ biểu diễn hai chữ cái, hãy chuyển bản rõ sang từ tiếng Anh (ở đây ta sử dụng phép tương ứng A↔1,...,Z↔26, vì 0 không được phép có trong cặp rõ có thứ tự. Ví dụ: DOG = $4 \times 26^2 + 15 \times 26 + 7 = 3101$).

9. Giả sử Alice và Bob dùng lược đồ chữ ký số ElGamal với p = 467. Đối với thông báo x=100, chữ ký của Bob là (29,51). Bob lại ký trên thông báo x=125 với chữ ký là (29,108).

a. Có nhận xét gì về việc dùng chữ ký của Bob?

b. Hãy tìm khoá ký bí mật a của Bob.

10. Alice dùng hệ mật mã khoá công khai Knapsack với các tham số như sau:

- Dãy siêu tăng là s = (2, 3, 7, 13, 29, 57); số nguyên tố p = 113 và số bí mật a = 61.

- Dãy số công khai tương ứng của Alice là : t = (9, 70, 88, 2, 74, 87).

Alice nhận được bản mã sau đây do Bob gửi cho mình: y = 186. Hãy giúp Alice giải bản mã này.

11. Mỗi bản mã sau là kết quả việc dùng hệ mã RSA mã từng chữ cái một với quy ước a=0, b=1,..., z=25 và n=18721, b=25 :

365, 0, 4845, 14930, 2608, 2608, 0. Chẳng hạn:

$$x^{25} \bmod 18721 = 365, \text{ với } 0 <= x <= 25.$$

Hãy tìm các bản rõ tương ứng mà không cần phân tích n ra thừa số.

12. Giả sử Bob và Charlie sử dụng hệ RSA với cùng n=18721. Số công khai của Bob là $b_1=13$, của Charlie là $b_2=7$.

Alice mã bản rõ x để gửi cho cả Bob và Charlie bằng cách dùng b_1, b_2 với các bản mã tương ứng sau:

$$y_1 = x^{b1} \bmod n = x^{13} \bmod 18721 = 6992;$$

$$y_2 = x^{b2} \bmod n = x^{7} \bmod 18721 = 4877;$$

Hãy tính giá trị x do Alice gửi cho Bob và Charlie mà không cần phân tích số n=18721.

13. Bob, Bart, Bert dùng hệ RSA với các modun riêng của mình lần lượt là $n_1=319$, $n_2=299$, $n_3=323$. Tuy nhiên, họ lại dùng số mũ chung b=3.

Alice mã cùng bản rõ x để gửi cho cả 3 người nói trên:

$$y_1 = x^3 \bmod n_1 = x^3 \bmod 319 = 60;$$

$$y_2 = x^3 \bmod n_2 = x^3 \bmod 299 = 222;$$

$$y_3 = x^3 \bmod n_3 = x^3 \bmod 323 = 56;$$

oscar biết b=3, biết $n_1=319$, $n_2=299$, $n_3=323$ và y_1, y_2, y_3 như trên. Không cần phân tích n_1, n_2, n_3 mà oscar vẫn tính được x. Vậy oscar đã tính x như thế nào và giá trị cụ thể của x là bao nhiêu?

14. 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.

15. 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 $h : \{1, \ldots, n^2\} \rightarrow Z_n^*$ 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.

TÀI LIỆU THAM KHẢO

[1] TS. Trần Văn Trường, ThS. Trần Quang Kỳ, *Giáo trình mật mã học nâng cao"*, Học viện Kỹ thuật Mật mã, 2007.

[2] Nguyễn Binh,*Giáo trình mật mã học*, NXB Bưu điện, 2004

[3] A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone,*Handbook of applied cryptography*. CRC Press 1998.

[4] B. Schneier, *Applied Cryptography*. John Wiley Press 1996.

[5] D. R. Stinson, *Cryptography. Theory and Practice*. CRC Press 1995.

[6] 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

[7] M. R. A. Huth,*Secure Communicating Systems*. Cambridge University Press 2001.

[8] C. Pfleeger, *Security in Computing*. Prentice Hall. 1997.

[9] 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 YvesChouirard. University of Ottawa. April 2002.

NHÀ XUẤT BẢN ĐẠI HỌC THÁI NGUYÊN
Phường Tân Thịnh - thành phố Thái Nguyên - tỉnh Thái Nguyên
Điện thoại: 0280 3840023; Fax: 0280 3840017
Website: nxb.tnu.edu.vn * E-mail: nxb.dhtn@gmail.com

TRẦN ĐỨC SỰ (Chủ biên)
NGUYỄN VĂN TẢO, TRẦN THỊ LƯỢNG

GIÁO TRÌNH
AN TOÀN BẢO MẬT DỮ LIỆU

Chịu trách nhiệm xuất bản:
PGS.TS. NGUYỄN ĐỨC HẠNH
Chịu trách nhiệm nội dung:
TBT - PGS.TS. TRẦN THỊ VIỆT TRUNG

Biên tập kỹ thuật:	TRẦN THỊ VÂN TRUNG
	NGUYỄN THỊ THÙY DƯƠNG
Thiết kế bìa:	LÊ THÀNH NGUYÊN
Trình bày:	LÊ THÀNH NGUYÊN
Sửa bản in:	DƯƠNG VĂN HOÀNH

ISBN: 978-604-915-250-4

In 500 cuốn, khổ 16 x 24 cm, tại Doanh nghiệp Tư nhân Tiến Dậu (Địa chỉ: thành phố Thái Nguyên). Giấy phép xuất bản số: 1329-2015/CXBIPH/02-35ĐHTN. Quyết định xuất bản số: 58/QĐ-NXBĐHTN. In xong và nộp lưu chểu quí IV năm 2015.