

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO BÀI TẬP LỚN

MÔN HỌC: Cơ Sở An Toàn Thông Tin

Đề tài: Giải thuật mã hóa khóa bí mật AES

Thành viên:

Nguyễn Đức Trí- B22DCAT302 (Nhóm Trưởng)

Nguyễn Việt Hà – B22DCVT172

Nguyễn Hải Lâm– B22DCVT303

Nguyễn Đại Phát– B22DCVT393

Nguyễn Việt Hoàng Hải – B22DCAT108

Nhóm 6 lớp E22CQCN05-B

Hà Nội, Tháng 10/2024

MỤC LỤC

LỜI NÓI ĐẦU	3
PHẦN 1: Tổng quan về thuật toán	4
1.Quá trình phát triển	4
2.Mô tả thuật toán	5
3.Phân công tìm hiểu.....	5
Phần 2 : Giải thuật mã hóa AES.....	6
1.Giới thiệu.....	6
2.Quá trình mã hóa	6
3.Mở rộng khóa	9
4.Các hàm xử lý chính	12
5.Quá trình giải mã.....	17
Phần 3 : Điểm yếu của AES.....	18
1.Khóa Ngắn	18
2.Sai sót trong cài đặt	18
3.Phương pháp sinh khóa yếu	18
Phần 4 : Các dạng tấn công vào AES.....	19
1.Tấn Công Brute-Force.....	19
2.Tấn công phân tích dấu hiệu	19
3. Tấn công dựa trên cấu trúc	19
4.Tấn công chọn plaintext.....	19
Phần 5 : Biện pháp phòng chống	20
Sử dụng khóa đủ dài.....	20
Thực hiện cài đặt đúng	20
Quản lý khóa chặt chẽ	20
Giảm thiểu rò rỉ thông tin.....	20
KẾT LUẬN	21
TÀI LIỆU THAM KHẢO.....	21

LỜI NÓI ĐẦU

Từ trước công nguyên con người đã phải quan tâm tới việc làm thế nào để đảm bảo an toàn bí mật cho các tài liệu, văn bản quan trọng, đặc biệt là trong lĩnh vực quân sự, ngoại giao. Ngày nay với sự xuất hiện của máy tính, các tài liệu văn bản giấy tờ và các thông tin quan trọng đều được số hóa và xử lý trên máy tính, được truyền đi trong môi trường mạng - một môi trường mà mặc định là không an toàn. Do đó yêu cầu về việc có một cơ chế, giải pháp để bảo vệ sự an toàn và bí mật của các thông tin nhạy cảm, quan trọng ngày càng trở nên cấp thiết. An toàn bảo mật thông tin là mối bận tâm bảo đảm mục đích này. Khó có thể thấy một ứng dụng Tin học có ích nào lại không sử dụng các thuật toán mã hóa thông tin.

Trong thời gian học tập tại trường Học viện Công nghệ Bưu chính viễn thông , được sự giúp đỡ tận tình của giảng viên Đinh Trường Duy , chúng em đã có thêm nhiều kiến thức về môn học cũng như ứng dụng của An toàn bảo mật thông tin trong thực tế. Trong phạm vi bài tập lớn, chúng em sẽ tìm hiểu về hệ mã hóa AES (Advanced Encryption Standard). Chúng em xin chân thành cảm ơn giảng viên Đinh Trường Duy đã giúp đỡ chúng em hoàn thành bài tập này!

Chúng em xin chân thành cảm ơn!

Hà Nội, ngày 4 tháng 10 năm 2024
Nhóm thực hiện : Nhóm 6
Lớp E22CQC�05-B

Phần 1: Tổng quan về thuật toán

Trong mật mã học, AES (viết tắt của từ tiếng anh: Advanced Encryption Standard hay Tiêu chuẩn mã hóa tiên tiến) là một chuẩn mã hóa dữ liệu được Viện Tiêu chuẩn và Công nghệ Mỹ (NIST) công nhận năm 2001. Giống như tiêu chuẩn tiền nhiệm DES, AES được kỳ vọng áp dụng trên phạm vi thế giới và đã được nghiên cứu rất kỹ lưỡng. AES được chấp thuận làm tiêu chuẩn liên bang bởi Viện công nghệ và tiêu chuẩn quốc gia Hoa Kỳ (NIST) sau một quá trình tiêu chuẩn hóa kéo dài 5 năm.

1. Quá trình phát triển

Năm 1997, NIST phát động một cuộc thi quốc tế để phát triển một thuật toán mã hóa mới nhằm thay thế DES. Mục tiêu của cuộc thi là tìm kiếm một thuật toán có khả năng mã hóa mạnh, hiệu quả về hiệu suất, linh hoạt về chiều dài khóa, và có thể được triển khai trong các thiết bị phần cứng và phần mềm đa dạng.

Năm 1998, NIST nhận được 15 đề xuất thuật toán từ nhiều nhà mật mã học trên toàn thế giới. Sau đó, danh sách này được rút ngắn xuống còn 5 thuật toán cuối cùng: MARS, RC6, Rijndael, Serpent, và Twofish.

Sau quá trình đánh giá và thảo luận kỹ lưỡng, thuật toán Rijndael, do hai nhà mật mã học người Bỉ là Joan Daemen và Vincent Rijmen phát triển, đã được chọn làm thuật toán AES.

Lý do chọn Rijndael:

Rijndael có tính linh hoạt về kích thước khối và kích thước khóa, hỗ trợ chiều dài khóa 128, 192, và 256 bit.

Thuật toán có hiệu suất cao trên cả phần cứng lẫn phần mềm.

Thiết kế đơn giản nhưng hiệu quả trong việc chống lại các tấn công mật mã học.

Năm 2000, NIST chính thức công bố Rijndael là người chiến thắng cuộc thi và trở thành thuật toán AES.

Năm 2001, NIST chuẩn hóa Rijndael thành AES qua tiêu chuẩn FIPS PUB 197 (Federal Information Processing Standards Publication). Từ đó, AES trở thành tiêu chuẩn mã hóa chính thức của Hoa Kỳ và được sử dụng rộng rãi trên toàn thế giới.

Sau khi được công bố, AES nhanh chóng được tích hợp vào các tiêu chuẩn bảo mật như SSL/TLS, IPsec, và nhiều giao thức mã hóa khác. Các chính phủ, tổ chức tài chính, và các tập đoàn công nghệ đã áp dụng AES trong việc bảo vệ dữ liệu nhạy cảm.

AES hiện nay được xem là một trong những thuật toán mã hóa mạnh mẽ và hiệu quả nhất, với chưa có lỗ hổng lớn nào bị phát hiện trong quá trình sử dụng.

2. Mô tả thuật toán

AES là dạng mã hóa khối, với khối dữ liệu vào có kích thước là 128 bit và khóa bí mật với kích thước có thể là 128, 192, hoặc 256 bit. AES được thiết kế dựa trên mạng hoán vị-thay thế (Substitution-permutation network) và nó có thể cho tốc độ thực thi cao khi cài đặt bằng cả phần mềm và phần cứng. Đặc biệt, giải thuật AES đã được tích hợp vào các bộ vi xử lý gần đây của hãng Intel dưới dạng tập lệnh AES-NI, giúp tăng đáng kể tốc độ thực thi các thao tác mã hóa và giải mã dựa trên AES.

3. Phân công tìm hiểu

Thuật toán AES là 1 trong những thuật toán tiên tiến và đang được sử dụng nhiều trên thế giới vì tính an toàn . Vì thế trong quá trình tìm hiểu nhóm em sẽ phân công công việc cho các thành viên như sau

Nguyễn Đức Trí : Tìm hiểu chính phần giải mã, mã hóa , sinh khóa.

Nguyễn Đại Phát : Tìm hiểu chính điểm yếu, các dạng tấn công , biện pháp phòng chống .

Nguyễn Hải Lâm :Viết demo thuật toán.

Nguyễn Việt Hà :Tổng hợp , căn chỉnh báo cáo .

Nguyễn Việt Hoàng Hải : Trình bày MixColumn.

Phần 2 : Giải thuật mã hóa AES

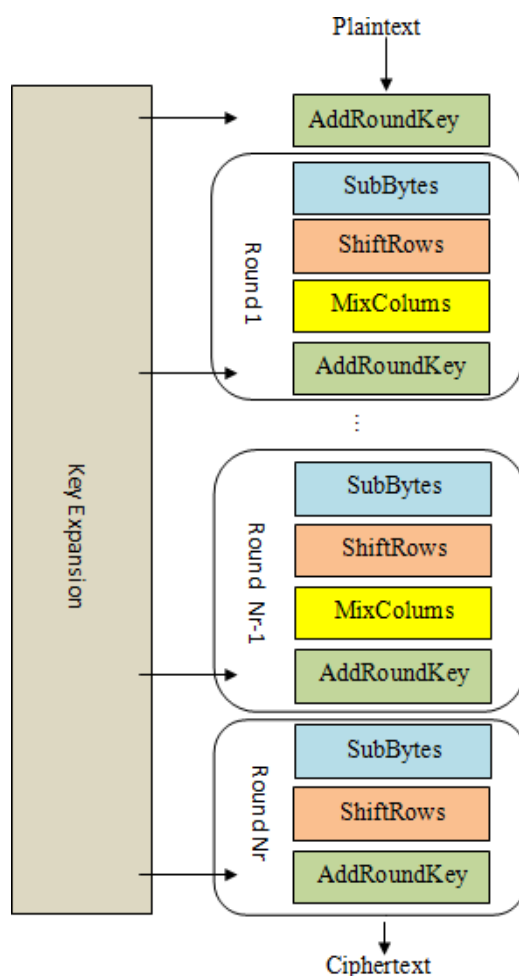
1. Giới thiệu

- AES vận hành dựa trên một ma trận vuông 4x4, được gọi là *state* (trạng thái). Ma trận này gồm 16 phần tử, mỗi phần tử là 1 byte dữ liệu. State được khởi trị là khối 128 bit bản rõ và qua quá trình biến đổi sẽ chứa khối 128 bit bản mã ở đầu ra. Như đã đề cập, AES hỗ trợ 3 kích thước khóa và kích thước của khóa quyết định số vòng lặp chuyển đổi cần thực hiện để chuyển bản rõ thành bản mã như sau:
- 10 vòng lặp với khóa 128 bit
- 12 vòng lặp với khóa 192 bit;
- 14 vòng lặp với khóa 256 bit.

2. Quá trình mã hóa

Giải thuật AES cho mã hóa dữ liệu, như minh họa trên hình dưới, gồm các bước xử lý

chính như sau: Mở rộng khóa (Key Expansion): các khóa vòng (Round key) dùng trong các vòng lặp được sinh ra từ khóa chính AES sử dụng thủ tục sinh khóa Rijndael.




Các bước xử lý mã hóa dữ liệu của AES

- Vòng khởi tạo (Initial Round): Thực hiện hàm AddRoundKey, trong đó mỗi byte trong *state* được kết hợp với khóa vòng sử dụng phép XOR.

Phép XOR:

Giá trị text	Giá trị nhị phân
CAT dưới dạng bit	0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 0
VVV là khóa	0 1 0 1 0 1 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0 1 1 0
Bản mã	0 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 0

Ví dụ mã hóa bằng phương pháp XOR:

Symbol	A	B	Q
	0	0	0
	1	0	1
	0	1	1
	1	1	0
$Q = A \oplus B$			

Phương pháp mã hóa XOR sử dụng phép toán logic XOR để tạo bản mã, trong đó từng bit của bản rõ được XOR với bit tương ứng của khóa. Để giải mã, ta thực hiện XOR từng bit của bản mã với bit tương ứng của khóa.

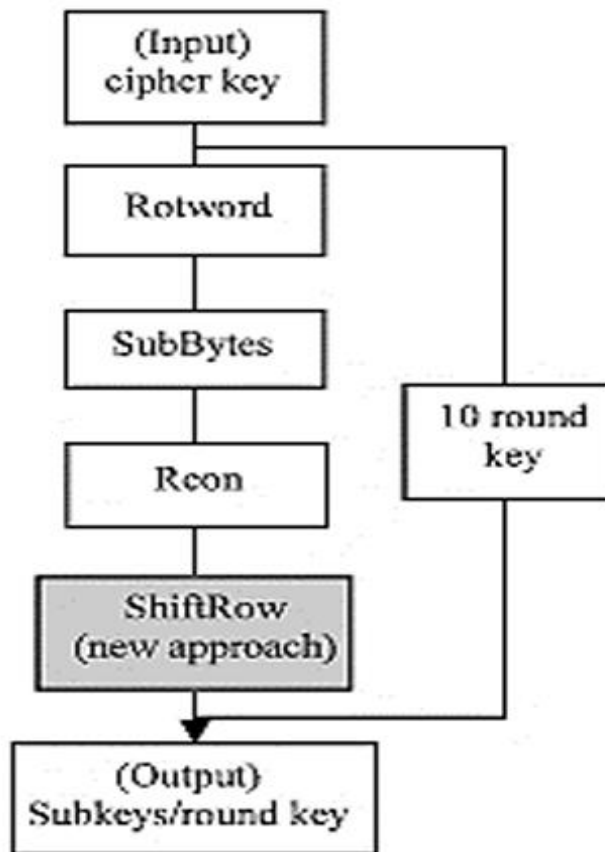
- Các vòng lặp chính (Rounds): Có 4 hàm biến đổi dữ liệu được thực hiện trong mỗi vòng, gồm:
 - + SubBytes: hàm thay thế phi tuyến tính, trong đó mỗi byte trong *state* được thay thế bằng một byte khác sử dụng bảng tham chiếu S-box;
 - + ShiftRows: hàm đổi chỗ, trong đó mỗi dòng trong *state* được dịch một số bước theo chu kỳ;
 - + MixColumns: trộn các cột trong *state*, kết hợp 4 bytes trong mỗi cột.
 - + AddRoundKey.
- Vòng cuối (Final Round): Tương tự các vòng lặp chính, nhưng chỉ thực hiện 3 hàm biến đổi dữ liệu, gồm:
 - + SubBytes;
 - + ShiftRows;
 - + AddRoundKey.

Lý do vòng lặp cuối chỉ có 3 hàm biến đổi và không có hàm MixColumns :

- Mục đích của **MixColumns**: Bước **MixColumns** trong các vòng trước nhằm tăng cường sự phụ thuộc giữa các byte và làm cho dữ liệu trở nên khó giải mã hơn. Nó thực hiện phép toán trên mỗi cột của ma trận trạng thái, làm cho mỗi byte trong cột bị ảnh hưởng bởi các byte khác trong cùng một cột.
- Bản chất của vòng lặp cuối: Trong vòng lặp cuối, mục tiêu chính là hoàn tất quá trình mã hóa và đưa ra kết quả cuối cùng, vì vậy **MixColumns** không cần thiết. Nếu thực hiện **MixColumns** trong vòng cuối, dữ liệu sẽ bị biến đổi thêm một lần nữa, và điều đó sẽ khiến cho dữ liệu không thể được khôi phục chính xác trong quá trình giải mã.
- Đồng bộ với quá trình giải mã: Trong quá trình giải mã AES, một bước "giải **MixColumns**" (Inverse MixColumns) được thực hiện ở tất cả các vòng trừ vòng cuối. Điều này giúp cho quá trình mã hóa và giải mã trở nên đối xứng. Bỏ qua **MixColumns** trong vòng cuối cùng giúp duy trì tính đồng bộ giữa quá trình mã hóa và giải mã.

Tóm lại, **MixColumns** được bỏ qua trong vòng cuối cùng để đảm bảo tính chính xác và đối xứng giữa quá trình mã hóa và giải mã, đồng thời không làm mất đi tính bảo mật của thuật toán.

3. Mở rộng khóa



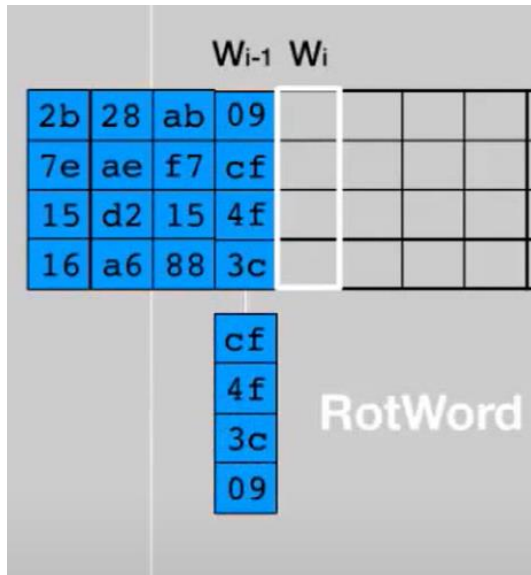
Khâu mở rộng khóa AES sử dụng thủ tục sinh khóa Rijndael để sinh các khóa vòng (Round key) cho các vòng lặp xử lý như biểu diễn trên Hình . Thủ tục Rijndael nhận đầu vào là khóa chính AES (cipher key) và xuất ra một khóa vòng (Subkey/Round key) sau mỗi vòng lặp. Một vòng lặp của thủ tục Rijndael gồm các khâu:

- Rotword: quay trái 8 bit từng từ 32 bit từ khóa gốc;
- SubBytes: thực hiện phép thay thế sử dụng bảng tham chiếu S-box.
- Rcon: tính toán giá trị $Rcon(i) = x^{(i-1)} \bmod x^8 + x^4 + x^3 + x + 1$
- ShiftRow: thực hiện đổi chỗ tương tự hàm ShiftRows của AES.

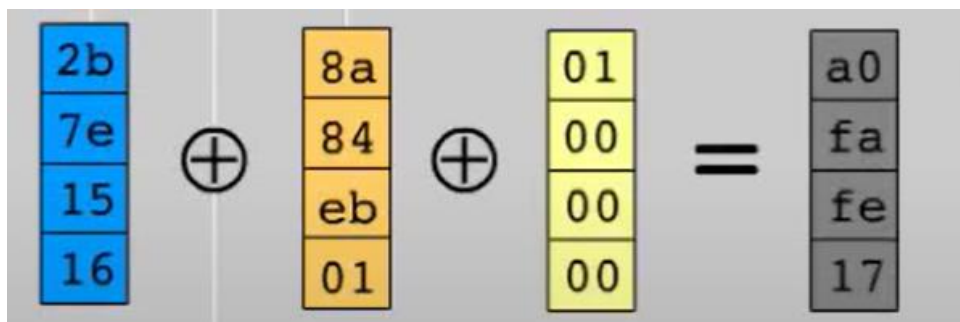
Ví dụ với khoá thứ i, sử dụng aes 128:

RotWord: Với cột đầu tiên

Lấy ra và dịch cột ngoài cùng key i-1 lên 1 byte;

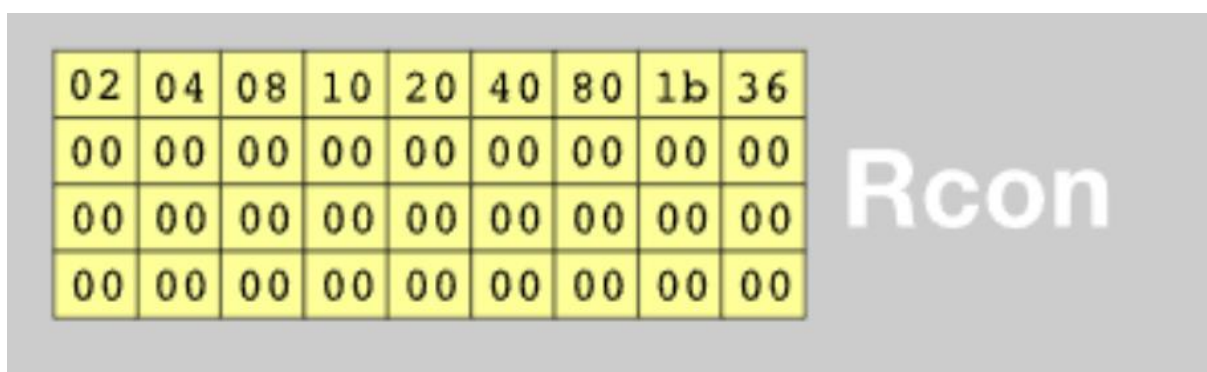


Sau đó SubBytes và xor nó với cột 1 của khoá trc và rcon i;



Sau đây là Rcon(1) đến Rcon(10) :

Rcon: tính toán giá trị $Rcon(i) = x^{(i-1)} \bmod x^8 + x^4 + x^3 + x + 1$



Các cột sau, ta gọi là cột j, thì giá trị cột j bằng xor cột j của key i-1 và cột j-1 của key i.

W_{i-4}				W_{i-1}	W_i				
2b	28	ab	09	a0					
7e	ae	f7	cf	fa					
15	d2	15	4f	fe					
16	a6	88	3c	17					

28	a0	88
ae	fa	54
d2	fe	2c
a6	17	b1

$\oplus =$

Sau đó **Shift row**

Tiếp tục làm như vậy cho đến khi đến key 10.

2b	28	ab	09	a0	88	23	2a	f2	7a	59	73	3d	47	1e	6d
7e	ae	f7	cf	fa	54	a3	6c	c2	96	35	59	80	16	23	7a
15	d2	15	4f	fe	2c	39	76	95	b9	80	f6	47	fe	7e	88
16	a6	88	3c	17	b1	39	05	f2	43	7a	7f	7d	3e	44	3b
Cipher key				Round key 1				Round key 2				Round key 3			

...

d0	c9	e1	b6
14	ee	3f	63
f9	25	0c	0c
a8	89	c8	a6
Round key 10			

4. Các hàm xử lý chính

Hàm SubBytes: Mỗi byte trong ma trận *state* được thay thế bởi 1 byte trong Rijndael S-box, hay $b_{ij} = S(a_{ij})$. S-box là một bảng tham chiếu phi tuyến tính, được tạo ra bằng phép nhân nghịch đảo một số cho trước trong trường $GF(2^8)$. Nếu như trong khâu mã hóa S-box được sử dụng thì bảng S-box *đảo* được sử dụng trong khâu giải mã.

S-Box:

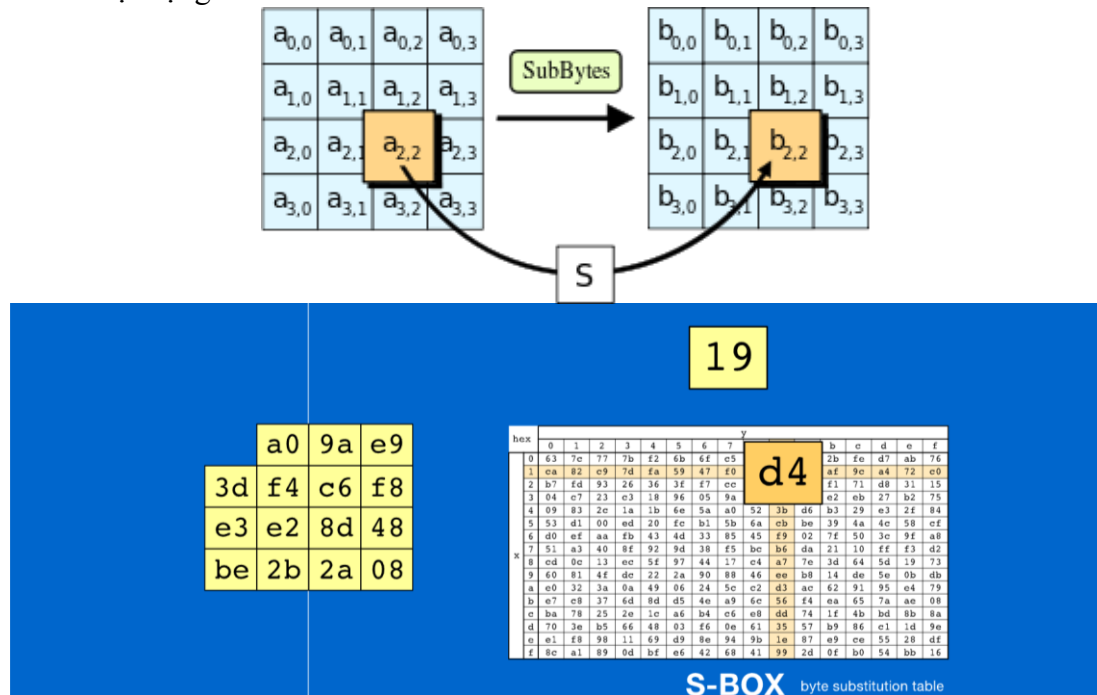
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
10	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
20	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
30	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
40	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
50	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
60	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
70	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
80	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
90	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	D8
A0	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B0	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C0	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D0	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E0	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F0	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Invert S-box:

AES inverse S-box

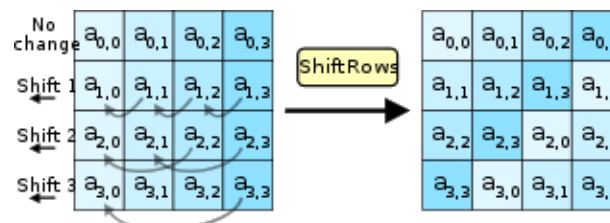
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
10	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
20	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
30	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
40	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
50	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
60	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
70	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
80	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
90	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A0	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B0	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C0	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D0	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E0	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F0	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Cách hoạt động :

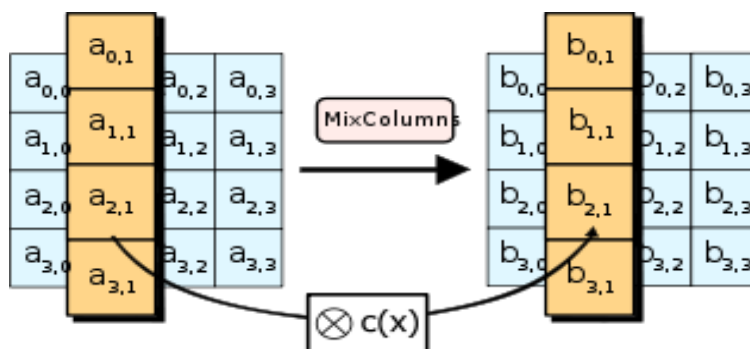


VD : Giá trị 19 được lấy từ hàng 1 cột 9 trong bảng Sbox ta được giá trị d4

Hàm ShiftRows: Các dòng của ma trận *state* được dịch theo chu kỳ sang trái theo nguyên tắc: hàng số 0 giữ nguyên, hàng số 1 dịch 1 byte sang trái, hàng số 2 dịch 2 byte và hàng số 3 dịch 3 byte, như minh họa sau:



Hàm MixColumns: Mỗi cột của ma trận *state* được nhân với một đa thức $c(x)$, như hình minh họa dưới. Đa thức $c(x) = 3x^3 + x^2 + x + 2$.



Mô tả bằng ma trận như sau :

$$S'(x) = a(x) * S(x)$$

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

Phép nhân trong trường GF(2⁸):

- Một số nguyên $a \in [0, 2^8 - 1]$, $a = a_7a_6a_5a_4a_3a_2a_1a_0$, sẽ tương đương với một đa thức có dạng:

$$f(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

- ĐN Phép cộng:** $(f + g)(x) = \sum_{i=0}^7 ((a_i + b_i) \bmod 2) x^i$

- ĐN Phép nhân:**

$$(f \times g)(x) = \left[\sum_{i=0}^7 \sum_{j=0}^7 ((a_i \times b_j) \bmod 2) x^{i+j} \right] [\bmod m(x)]$$

- đa thức tối giản $m(x) = x^8 + x^4 + x^3 + x + 1$

Ví dụ:

e0	b8	1e
b4	41	27
52	11	98
ae	f1	e5

\cdot

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

\cdot

d4
bf
5d
30

$=$

04
66
81
e5

The four numbers of one column are modulo multiplied in Rijndael's Galois Field by a given matrix.

Với hàng 1 cột 1:

-) {02}. {d4}

{02} = 0000 0010 tương đương $f(x) = x$

{d4} = 1101 0100 tương đương $g(x) = x^7 + x^6 + x^4 + x^2$

Ta có : $f(x) \cdot g(x) = x^8 + x^7 + x^5 + x^3$

Thực hiện phép $f(x) \cdot g(x) \bmod m(x)$:

$(x^8 + x^7 + x^5 + x^3) \bmod (x^8 + x^4 + x^3 + x + 1)$

$= x^7 + x^5 + x^4 + x + 1$

-) {03}. {bf}

{03} = 0000 0011 tương đương $f(x) = x + 1$

{bf} = 1011 1111 tương đương $g(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$

Ta có : $f(x) \cdot g(x) = x^8 + x^7 + x^6 + 1$

Thực hiện phép $f(x) \cdot g(x) \bmod m(x)$:

$(x^8 + x^7 + x^6 + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$

$= x^7 + x^6 + x^4 + x^3 + x$

-) {01}. {5d}

{01} = 0000 0001 tương đương $f(x) = 1$

{5d} = 0101 1101 tương đương $g(x) = x^6 + x^4 + x^3 + x^2 + 1$

Ta có : $f(x) \cdot g(x) = (x^6 + x^4 + x^3 + x^2 + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$

$= x^6 + x^4 + x^3 + x^2 + 1$

-) {01}. {30}

{01} = 0000 0001 tương đương $f(x) = 1$

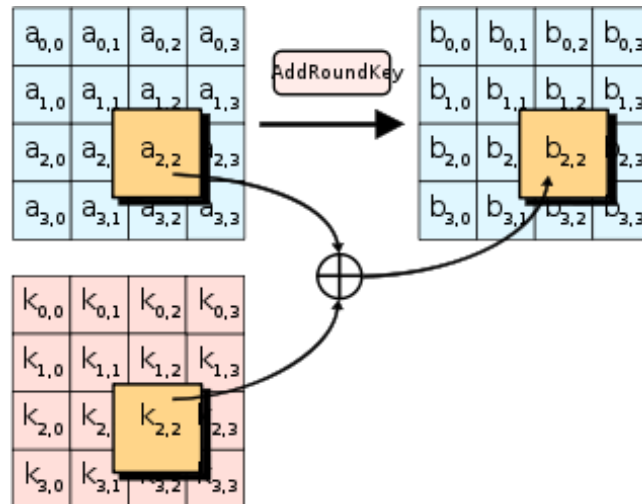
{30} = 0011 0000 tương đương $g(x) = x^5 + x^4$

Ta có : $f(x) \cdot g(x) = (x^5 + x^4) \bmod (x^8 + x^4 + x^3 + x + 1)$

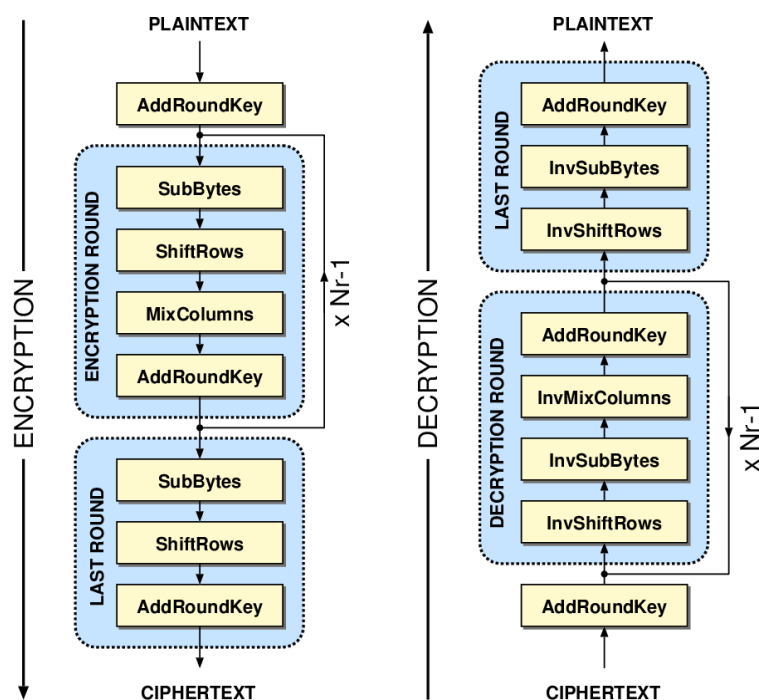
$= x^5 + x^4$

- Vậy ta có $(\{02\}.\{d4\}) \oplus (\{03\}.\{bf\}) \oplus (\{01\}.\{5d\}) \oplus (\{01\}.\{30\}) = x^2$ tương đương 0000 0100 tương đương với {04} .

Hàm *AddRoundKey*: Mỗi byte của ma trận *state* được kết hợp với một byte tương ứng của khóa vòng sử dụng phép (XOR), như minh họa như hình dưới



5. Quá trình mã hóa và giải mã trong AES



Khâu giải mã trong AES cũng gồm các bước xử lý tương tự như khâu mã hóa. Hình trên biểu diễn quá trình mã hóa và giải mã trong AES. Theo đó, ngoài bước Mở rộng khóa, quá trình giải mã gồm Vòng khởi tạo (AddRoundKey), Các vòng lặp chính (Decryption round) và Vòng cuối (Last round) để chuyển khối mã thành khối rõ. Điểm khác biệt chính của khâu giải mã so với khâu mã hóa là các hàm đảo được sử dụng, như các hàm đảo InvSubBytes (tra bảng), InvShiftRows (ví dụ nếu shiftrow hàng đó bị dịch sang trái i bước thì giờ chỉ cần dịch thêm $4-i$ bước) và InvMixColumns tương ứng thay cho các hàm SubBytes, ShiftRows và MixColumns.

InvMixColumns:

InvMixColumns là hàm nghịch đảo của MixColumn(). InvMixColumn xử lý từng cột như một đa thức bốn hạng tử. Các cột được coi là đa thức trên $GF(2^8)$ và nhân modulo $x^2 + 1$ với một đa thức cố định được đưa ra bởi:

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}.$$

Hay nó có thể viết như một phép nhân ma trận:

$$s'(x) = a^{-1}(x) * s(x)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Phần 3: Điểm Yếu của AES

1. Khóa Ngắn

- Mô tả: Sử dụng khóa ngắn hơn 128 bit (như 64 bit) làm giảm khả năng chống lại tấn công brute-force. AES được thiết kế để hoạt động với khóa 128, 192 và 256 bit, và việc sử dụng khóa ngắn hơn sẽ dễ dàng bị bẻ khóa bởi các kẻ tấn công.
- NIST khuyến nghị sử dụng khóa dài tối thiểu 128 bit để đảm bảo độ an toàn cho các ứng dụng mã hóa, và trong nhiều trường hợp, việc sử dụng khóa 192 hoặc 256 bit được xem là cần thiết để tăng cường bảo mật.

2. Sai Sót Trong Cài Đặt

- Mô tả: Việc triển khai không chính xác AES có thể dẫn đến các lỗ hổng bảo mật nghiêm trọng. Điều này có thể bao gồm việc không sử dụng chế độ hoạt động đúng cách, hoặc không quản lý khóa đúng cách, chẳng hạn như việc không bảo vệ khóa khỏi việc truy cập trái phép.
- Ví dụ cụ thể: Nếu một ứng dụng sử dụng chế độ ECB (Electronic Codebook Mode), mỗi khối plaintext sẽ được mã hóa độc lập, dẫn đến việc có thể dễ dàng phân tích và xác định được cấu trúc của dữ liệu. NIST khuyến nghị sử dụng các chế độ an toàn hơn như CBC (Cipher Block Chaining) hoặc GCM (Galois/Counter Mode) để cải thiện tính bảo mật.

3. Phương Pháp Sinh Khóa Yếu

- Mô tả: Nếu phương pháp tạo khóa không đủ ngẫu nhiên, có thể dẫn đến việc sinh ra các khóa dễ đoán. Điều này làm cho việc bẻ khóa trở nên khả thi hơn, đặc biệt là với các tấn công brute-force.
- NIST khuyến nghị sử dụng các phương pháp sinh khóa mạnh mẽ và an toàn, chẳng hạn như sử dụng các trình tạo số ngẫu nhiên (Random Number Generators - RNG) theo tiêu chuẩn FIPS 140-2. Việc sinh ra khóa phải dựa trên nguồn ngẫu nhiên chất lượng cao để đảm bảo rằng không có mẫu hoặc sự lặp lại nào trong các khóa được sinh ra.

Phần 4: Các Dạng Tấn Công vào AES

1. Tấn Công Brute-Force

- Mô tả: Tấn công brute-force thử tất cả các khóa khả thi cho đến khi tìm ra khóa đúng.
- Ví dụ: Đối với AES-128, có 2^{128} khóa khả thi, điều này khiến việc thực hiện tấn công brute-force trở nên không khả thi trong thực tế. Tuy nhiên, việc sử dụng các khóa ngắn hơn (như 64 bit) khiến chúng dễ bị tấn công brute-force.

2. Tấn Công Phân Tích Dấu Hiệu (Side-Channel Attacks)

- Mô tả: Tấn công này khai thác thông tin rò rỉ trong quá trình thực hiện mã hóa, như thời gian thực thi, điện năng tiêu thụ hoặc tiếng ồn điện từ.
- Ví dụ: Tấn công dựa vào thời gian để phân tích thời gian mà một hệ thống cần để thực hiện AES với các khóa khác nhau, từ đó suy đoán được giá trị của khóa. Một cuộc tấn công phổ biến là Differential Power Analysis (DPA), nơi kẻ tấn công ghi lại tiêu thụ điện năng của một thiết bị khi thực hiện mã hóa và sử dụng thông tin đó để lấy thông tin về khóa.

3. Tấn Công Dựa Trên Cấu Trúc

- Mô tả: Tấn công này tận dụng các thiếu sót trong thiết kế hoặc cấu trúc của AES để tìm ra khóa.
- Ví dụ: Một ví dụ là tấn công chiếm lĩnh (cryptanalysis attack), nơi kẻ tấn công sử dụng các thuộc tính của mạng lưới AES (sự sắp xếp của các khối và phép toán) để xây dựng các cuộc tấn công có thể phá vỡ mã hóa. Các tấn công này có thể bao gồm linear cryptanalysis và differential cryptanalysis, nơi kẻ tấn công so sánh các đầu vào và đầu ra của một số khối mã hóa để tìm ra mối quan hệ và suy luận về khóa.

4. Tấn Công Chọn Plaintext (Chosen-Plaintext Attacks)

- Mô tả: Kẻ tấn công có khả năng chọn một plaintext cụ thể để mã hóa và sau đó phân tích kết quả.
- Ví dụ: Kẻ tấn công có thể chọn một plaintext mà họ biết trước và mã hóa nó để xem cách AES mã hóa đầu vào đó, từ đó so sánh với các plaintext khác đã được mã hóa để suy luận ra khóa. Tấn công này rất hiệu quả trong các hệ thống mà kẻ tấn công có thể kiểm soát hoặc quan sát đầu vào và đầu ra.

Phần 5: Biện Pháp Phòng Chống

1. Sử Dụng Khóa Đủ Dài

- Mô tả: Đảm bảo rằng khóa mã hóa được sử dụng có độ dài tối thiểu là 128 bit. Khóa dài hơn (192 bit hoặc 256 bit) có thể cung cấp mức độ bảo mật cao hơn.
- Lý do: Khóa ngắn hơn 128 bit dễ bị tấn công brute-force, trong khi khóa dài hơn cung cấp nhiều khả năng hơn trong việc ngăn chặn các cuộc tấn công tìm kiếm khóa.
- NIST nhấn mạnh tầm quan trọng của việc sử dụng các khóa đủ dài trong tài liệu của họ, vì điều này có thể làm giảm đáng kể khả năng bị bẻ khóa và nâng cao tính bảo mật tổng thể.

2. Thực Hiện Cài Đặt Đúng

- Mô tả: Tuân thủ các hướng dẫn từ NIST để triển khai AES chính xác là rất quan trọng, bao gồm việc chọn chế độ hoạt động an toàn cho mã hóa.
- Chế độ Hoạt động: Các chế độ như CBC (Cipher Block Chaining), GCM (Galois/Counter Mode) hoặc CTR (Counter Mode) được khuyến nghị hơn so với ECB (Electronic Codebook Mode) vì chúng cung cấp sự bảo mật tốt hơn bằng cách tăng cường sự ngẫu nhiên trong mã hóa và giảm khả năng bị phân tích dữ liệu.
- NIST SP 800-38A cung cấp hướng dẫn chi tiết về các chế độ hoạt động khác nhau, yêu cầu cài đặt chính xác, và tầm quan trọng của việc sử dụng các yếu tố ngẫu nhiên trong mã hóa để ngăn chặn các cuộc tấn công phân tích.

3. Quản Lý Khóa Chặt Chẽ

- Mô tả: Việc quản lý khóa đúng cách là yếu tố quan trọng trong bảo mật AES. Điều này bao gồm việc lưu trữ, phân phối và xóa các khóa mã hóa một cách an toàn.
- Phương pháp bảo vệ: Sử dụng các thiết bị như HSM (Hardware Security Module) để lưu trữ khóa một cách an toàn. HSM cung cấp một môi trường an toàn để thực hiện các hoạt động mã hóa và giải mã mà không tiết lộ khóa cho bên ngoài.
- NIST nhấn mạnh sự cần thiết phải thực hiện quản lý khóa theo tiêu chuẩn FIPS 140-2 để đảm bảo an toàn cho khóa mã hóa.

4. Giảm Thiểu Rò Rỉ Thông Tin

- Mô tả: Các cuộc tấn công side-channel có thể khai thác thông tin rò rỉ từ quá trình mã hóa, chẳng hạn như thời gian thực thi hoặc tiêu thụ năng lượng.
- Biện pháp bảo vệ: Triển khai các biện pháp bảo vệ như "randomized timing" (thêm độ ngẫu nhiên vào thời gian thực hiện) để làm khó cho kẻ tấn công trong việc phân tích thông tin bị rò rỉ.
- NIST khuyến cáo việc thiết kế hệ thống sao cho giảm thiểu thông tin rò rỉ và bảo vệ chống lại các loại tấn công này bằng cách sử dụng các phương pháp bảo vệ bổ sung.

KẾT LUẬN

Bài báo cáo tìm hiểu về **AES (Advanced Encryption Standard)** – một thuật toán mã hóa đối xứng được công nhận và sử dụng rộng rãi trong lĩnh vực bảo mật thông tin. Báo cáo được chia thành các phần như sau:

1. Tổng quan về thuật toán:

AES là thuật toán mã hóa khối, với các kích thước khóa 128, 192 và 256 bit. Nó được phát triển để thay thế DES và đã trở thành tiêu chuẩn mã hóa toàn cầu sau khi được NIST công nhận năm 2001.

2. Giải thuật mã hóa AES:

Quá trình mã hóa AES gồm các bước chính như: mở rộng khóa, vòng khởi tạo, các vòng lặp chính và vòng cuối cùng. Các phép toán cơ bản bao gồm SubBytes (thay thế byte), ShiftRows (dịch dòng), MixColumns (trộn cột) và AddRoundKey (XOR với khóa vòng).

3. Điểm yếu của AES:

AES vẫn có một số điểm yếu tiềm tàng như sử dụng khóa ngắn, lỗi trong cài đặt và phương pháp sinh khóa yếu.

4. Các dạng tấn công vào AES:

Các hình thức tấn công bao gồm tấn công brute-force, tấn công phân tích dấu hiệu (side-channel attacks), và tấn công chọn plaintext.

5. Biện pháp phòng chống:

Biện pháp phòng chống bao gồm sử dụng khóa đủ dài, cài đặt chính xác, quản lý khóa chặt chẽ và giảm thiểu rò rỉ thông tin.

Kết luận AES là một trong những thuật toán mã hóa mạnh mẽ và phổ biến nhất hiện nay, với tính an toàn cao khi được cài đặt và quản lý đúng cách. Tuy nhiên, để đảm bảo tính bảo mật tối đa, người dùng cần chú ý tới việc sử dụng khóa đủ dài, triển khai các biện pháp bảo vệ khỏi các cuộc tấn công như brute-force và phân tích dấu hiệu. Với việc áp dụng đúng các khuyến nghị, AES vẫn là một lựa chọn lý tưởng cho việc mã hóa thông tin trong các hệ thống bảo mật.

Tài Liệu Tham Khảo

[1] Bài Giảng An Toàn Và Bảo Mật Hệ Thống Thông Tin – Hoàng Xuân Dậu 2021.

[2] NIST Special Publication 800-38A,

<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-38a.pdf>

NIST Special Publication 800-57 Part 1, “Recommendation for Key Management”

<https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-57pt1r4.pdf>