

TCVN

TIÊU CHUẨN QUỐC GIA

TCVN 11367-3:2016
ISO/IEC 18033-3:2010
Xuất bản lần 1

**CÔNG NGHỆ THÔNG TIN - CÁC KỸ THUẬT AN TOÀN -
THUẬT TOÁN MẬT MÃ - PHẦN 3: MÃ KHỐI**

Information technology -- Security techniques -- Encryption algorithms -- Part 3: Block ciphers

HÀ NỘI - 2016

Mục Lục

Lời nói đầu	4
1 Phạm vi áp dụng	5
2 Thuật ngữ và định nghĩa	5
3 Các ký hiệu.....	6
4 Mã khối 64 bit.....	8
4.1 Mở đầu.....	8
4.2 TDEA	8
4.3 MISTY1	9
4.4 CAST-128	16
4.5 HIGHT	26
5 Mã khối 128-bit.....	28
5.1 Mở đầu.....	28
5.2 AES.....	28
5.3 Camellia	37
5.4 SEED	51
Phụ lục A (Quy định) Mô tả DES.....	56
A.1 Mở đầu	56
A.2 Mã hóa DES	56
A.3 Phép giải mã DES	56
A.4 Các hàm DES.....	56
A.5 Lược đồ tạo khóa DES	62
Phụ lục B (Quy định) Định danh đối tượng.....	64
Phụ lục C (Quy định) Dạng đại số của các S-box trong Thuật toán MISTY1 và Camellia.....	66
C.1 Mở đầu	66
C.2 S-box trong MISTY1	66
C.3 Các S-box Camellia.....	67
Phụ lục D (Tham khảo) Véc tơ kiểm tra	68
D.1 Mở đầu	68
D.2 Véc tơ kiểm tra TDEA.....	68
D.3 Véc tơ kiểm tra MISTY1	71
D.4 Véc tơ kiểm tra CAST-128.....	71
D.5 Véc tơ kiểm tra HIGHT	72
D.6 Véc tơ kiểm tra AES	72
D.7 Véc tơ kiểm tra Camellia	77
D.8 Véc tơ kiểm tra SEED	80
Phụ lục E (Tham khảo) Bảng đặc tính.....	84
Thư mục tài liệu tham khảo.....	86

Lời nói đầu

TCVN 11367-3:2016 hoàn toàn tương đương với ISO/IEC 18033-3:2010.

TCVN 11367-3:2016 do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã biên soạn, Ban Cơ yếu Chính phủ đề nghị, Tổng cục Tiêu chuẩn Đo lường Chất lượng thẩm định, Bộ Khoa học và Công nghệ công bố.

Bộ tiêu chuẩn TCVN 11367 Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã gồm 04 phần:

- TCVN 11367-1:2016 (ISO/IEC 18033-1:2015) Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 1: Tổng quan.
- TCVN 11367-2:2016 (ISO/IEC 18033-2:2006) Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 2: Mật mã phi đối xứng.
- TCVN 11367-3:2016 (ISO/IEC 18033-3:2010) Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 3: Mã khối.
- TCVN 11367-4:2016 (ISO/IEC 18033-4:2011) Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 4: Mã dòng.

Công nghệ thông tin - Các kỹ thuật an toàn - thuật toán mật mã - Phần 3: Mã khối

Information technology - Security techniques - Encryption algorithms - Part 3: Block ciphers

1 Phạm vi áp dụng

Tiêu chuẩn này đặc tả các mã khối. Mã khối ánh xạ các khối n bit vào các khối n bit dưới sự kiểm soát của khóa có độ dài k -bit. Có tất cả bảy loại mã khối khác nhau được mô tả và chúng được phân loại trên bảng 1.

Bảng 1 - Đặc tả mã khối

Độ dài khối	Tên thuật toán	Độ dài khóa
64 bit	TDEA (4,2)	128 hoặc 192 bit
	MISTY1 (4,3)	128 bit
	CAST-128 (4,4)	
	HIGHT (4,5)	
128 bit	AES (5,2)	128, 192 hoặc 256 bit
	Camellia (5,3)	128, 192 hoặc 256 bit
	SEED (5,4)	128 bit

Các thuật toán được đặc tả trong tiêu chuẩn này của bộ TCVN 11367 (ISO/IEC 18033), được gán cho các định danh đối tượng phù hợp với tiêu chuẩn ISO/IEC 9834. Danh sách các định danh đối tượng được đưa ra tại Phụ lục B. Bất kì sự thay đổi về đặc tính của thuật toán dẫn đến thay đổi hành vi chức năng chức năng và sẽ dẫn đến thay đổi định danh đối tượng mà thuật toán nhằm tới.

2 Thuật ngữ và định nghĩa

Trong tiêu chuẩn này áp dụng các thuật ngữ và định nghĩa dưới đây:

TCVN 11367-3:2016

2.1

Khối (block)

Xâu bit có độ dài xác định.

CHÚ THÍCH Trong tiêu chuẩn này của bộ TCVN 11367 (ISO/IEC 18033), độ dài khối hoặc là 64 hoặc 128 bit.

[ISO/IEC 18033-1:2005].

2.2

Mã khối (block cipher)

Hệ thống mã khối đối xứng, với tính chất là thuật toán mã hóa thao tác trên các khối bản rõ, tức các xâu bit có độ dài xác định, cho ra khối bản mã.

[ISO/IEC 18033-1:2005]

2.3

Bản mã (ciphertext)

Dữ liệu đã được biến đổi để che giấu nội dung thông tin trong đó.

[ISO/IEC 9798-1:1997]

2.4

Khóa (key)

Dãy các kí hiệu điều khiển thao tác của phép biến đổi mật mã (ví dụ, phép mã hóa và phép giải mã)

CHÚ THÍCH Trong tất cả các hệ mật mã được mô tả trong tiêu chuẩn này của bộ TCVN 11367 (ISO/IEC-18033), khóa gồm một các dãy bit.

[ISO/IEC 11770-1:1996]

2.5

Mã khối n-bit (n-bit block cipher)

Mã khối với tính chất là các khối bản rõ và khối bản mã có độ dài n bit.

[ISO/IEC 10116:2006]

2.6

Bản rõ (plaintext)

Thông tin chưa được mã hóa

[ISO/IEC 9797-1:1999]

3 Các ký hiệu

n độ dài tính bằng bit của bản rõ/bản mã đối với mã khối

E_k	hàm mã hóa với khóa K
D_k	hàm giải mã với khóa K
Nr	số vòng của thuật toán AES, bằng 10, 12 hoặc 14 để chọn độ dài khóa tương ứng 128, 192, hoặc 256 bit .
\oplus	phép toán logic XOR trên chuỗi bit, nghĩa là nếu A và B là hai chuỗi cùng độ dài thì $A \oplus B$ là chuỗi bit bao gồm các bit là kết quả phép toán logic XOR của A và B
\otimes	phép nhân hai đa thức (mỗi đa thức có bậc bé hơn 4) theo mod $x^4 + 1$
\wedge	phép toán logic AND trên chuỗi bit, nghĩa là nếu A và B là các chuỗi bit cùng độ dài, thì $A \wedge B$ là chuỗi bit được tạo từ phép toán logic AND các bit tương ứng của A và B .
\vee	phép toán logic OR trên chuỗi bit, tức nếu A và B là hai chuỗi bit cùng độ dài, thì $A \vee B$ là chuỗi bit gồm các bit là kết quả của phép toán logic OR của A và B .
\parallel	phép ghép các chuỗi bit
\cdot	phép nhân trên trường hữu hạn
\ll_i	phép dịch vòng sang trái i bit.
\gg_i	phép dịch vòng sang phải i bit
\bar{x}	phép bù bit của x
$a \bmod n$	với các số nguyên a và n , $a \bmod n$ kí hiệu số dư (không âm) trong phép chia n cho a . Một cách tương đương, $b = a \bmod n$ nếu b là số nguyên duy nhất thỏa mãn các điều kiện sau: (i) $0 \leq b < n$ (ii) $(b-a)$ là bội số nguyên của n
\boxplus	phép cộng trong số học mô-đun, nghĩa là nếu A và B là hai chuỗi t -bit thì $A \boxplus B$ được xác định bằng $(A+B) \bmod 2^t$
\boxminus	phép cộng trong số học mô-đun, nghĩa là nếu A và B là hai chuỗi t -bit thì $A \boxminus B$ được xác định bằng $(A + B) \bmod 2^t$

4 Mã khối 64 bit

4.1 Tổng quan

Trong Điều này đặc tả bốn loại mã khối 64 bit: TDEA (hay “Ba DES”- “Triple DES”) ở Điều 4.2, MISTY I ở Điều 4.3, CAST-128 ở Điều 4.4 và HIGHT ở Điều 4.5.

Người dùng được cấp quyền để tiếp cận dữ liệu đã mã hóa, sẽ nhận được khóa dùng để mã dữ liệu để giải mã dữ liệu. Thuật toán của bất kì hệ mật nào trong điều này, được thiết kế để mã hóa và giải mã các khối dữ liệu 64 bit, theo sự điều khiển của khóa 128 (hoặc 192 bit). Phép giải mã sử dụng cùng khóa của phép mã hóa.

4.2 TDEA

4.2.1 Thuật toán mã dữ liệu bội ba

Thuật toán mã dữ liệu bội ba (TDEA-Triple Data Encryption) là một mã đối xứng, có thể xử lí các khối dữ liệu 64 bit, sử dụng khóa mật mã có độ dài 128 (hoặc 192) bit, trong đó 112 bit (hoặc 168 bit) có thể được chọn một cách tùy ý, các bit còn lại dùng để làm mã phát hiện sai. TDEA thường được biết đến như là DES bội ba (Triple-DES)

Phép mã hóa/giải mã TDEA là phép toán ghép các phép toán mã hoá và giải mã DES, ở đây thuật toán DES được đặc tả tại Phụ lục A. Khóa của TDEA gồm ba khóa DES.

4.2.2 Phép mã hóa/giải mã TDEA

4.2.2.1 Các định nghĩa mã hóa /giải mã

TDEA được xác định theo thuật ngữ của phép toán DES, ở đây E_k là phép toán mã hóa của DES với khóa K và D_k là phép giải mã của DES với khóa K .

4.2.2.2. Phép mã hóa của TDEA

Phép biến đổi khối P 64 bit thành khối C 64 bit được xác định như sau:

$$C = E_{K_3}(D_{K_2}(E_{K_1}(P))).$$

4.2.2.3 Phép giải mã TDEA

Phép biến đổi khối C 64 bit thành khối P 64 bit được xác định như sau:

$$P = D_{K_1}(E_{K_2}(D_{K_3}(C)))$$

4.2.3 Tùy chọn khóa trong TDEA

Tiêu chuẩn này của TCVN 11367 (ISO/IEC 18033) xác định các tùy chọn khóa cho TDEA. Khóa TDEA gồm ba khóa (K_1, K_2, K_3) .

1. Tùy chọn khóa 1: K_1, K_2 và K_3 là những khóa DES khác nhau;
2. Tùy chọn khóa 2: K_1 và K_2 là các khóa DES khác nhau, còn $K_3 = K_1$.

CHÚ THÍCH Tùy chọn khóa $K_1 = K_2 = K_3$ tương đương với DES-đơn không được khuyến cáo sử dụng. Ngoài ra nên sử dụng tùy chọn 1 hơn tùy chọn 2, vì tùy chọn 1 cung cấp mức an toàn bổ sung tại cùng hiệu năng thực thi (tham khảo thêm chi tiết ở [3]).

4.3 MISTY1

4.3.1 Thuật toán MISTY1

Thuật toán MISTY1 là mã khối đối xứng, có thể xử lý các khối 64 bit sử dụng khóa mật mã độ dài 128 bit.

4.3.2 Phép mã hóa MISTY1

Phép mã hóa được chỉ ra trên Hình 1. Phép biến đổi khối 64-bit P thành khối 64-bit C được xác định như sau (KL , KO , và KI là các khóa):

$$(1) P = L_0 \| R_0$$

$$KL = KL_1 \| KL_2 \| \dots \| KL_{10}$$

$$KO = KO_1 \| KO_2 \| \dots \| KO_8$$

$$KI = KI_1 \| KI_2 \| \dots \| KI_8$$

(2) Với $i = 1, 3, \dots, 7$ (độ chênh lệch các giá trị liên tiếp nhau của i bằng 2 vì có hai vòng lặp):

$$R_i = FL(L_{i-1}, KL_i)$$

$$L_i = FL(R_{i-1}, KL_{i+1}) \oplus FO(R_i, KO_i, KI_i)$$

$$L_{i+1} = R_i \oplus FO(L_i, KO_{i+1}, KI_{i+1})$$

$$R_{i+1} = L_i$$

với $i = 9$:

$$R_i = FL(L_{i-1}, KL_i)$$

$$L_i = FL(R_{i-1}, KL_{i+1})$$

$$(3) C = L_9 \| R_9$$

4.3.3 Phép giải mã MISTY1

Phép toán giải mã được chỉ ra trên Hình 2 và giống như phép mã hóa, trừ hai sửa đổi sau đây:

- (1) Tất cả hàm FL được thay bằng hàm ngược FL^{-1} của chúng.
- (2) Thứ tự áp dụng các khóa con được đảo lại.

4.3.4 Các hàm MISTRY1

4.3.4.1 Định nghĩa hàm MISTRY1

Thuật toán MISTRY1 sử dụng một số lượng hàm, đó là S_7 , S_9 , FI, FO, FL và FL^{-1} , các hàm này sẽ được định nghĩa dưới đây.

4.3.4.2 Hàm FL

TCVN 11367-3:2016

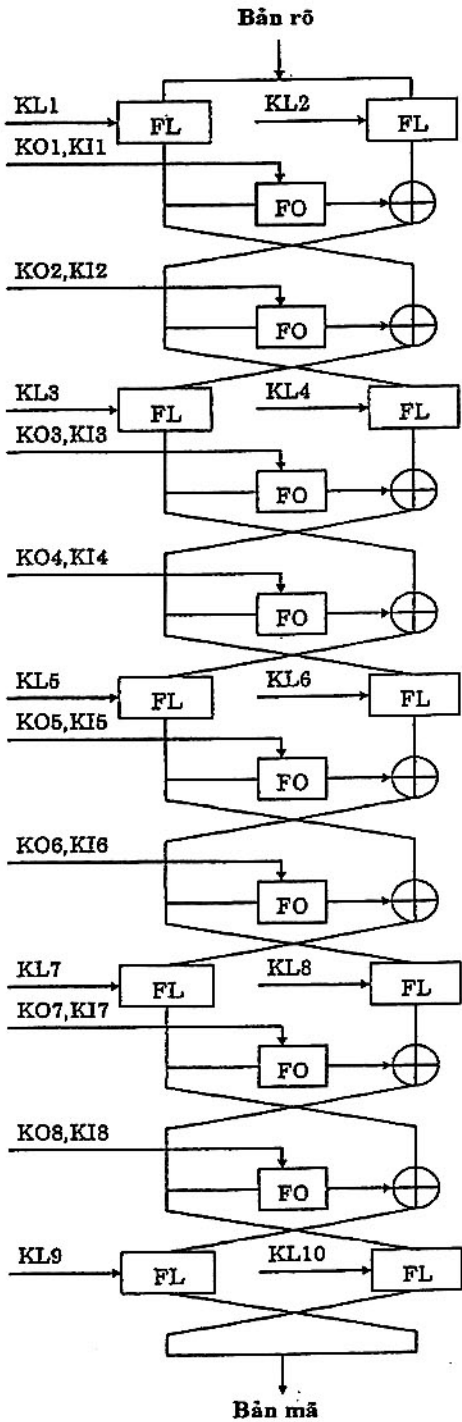
Hàm FL được chỉ ra trên Hình 2 và chỉ áp dụng trong phép mã hóa. Hàm FL được xác định như sau (X và Y là dữ liệu, KL là khóa):

$$(1) X_{32} = X_L \parallel X_R, KL_1 = KL_{iL} \parallel KL_{iR}$$

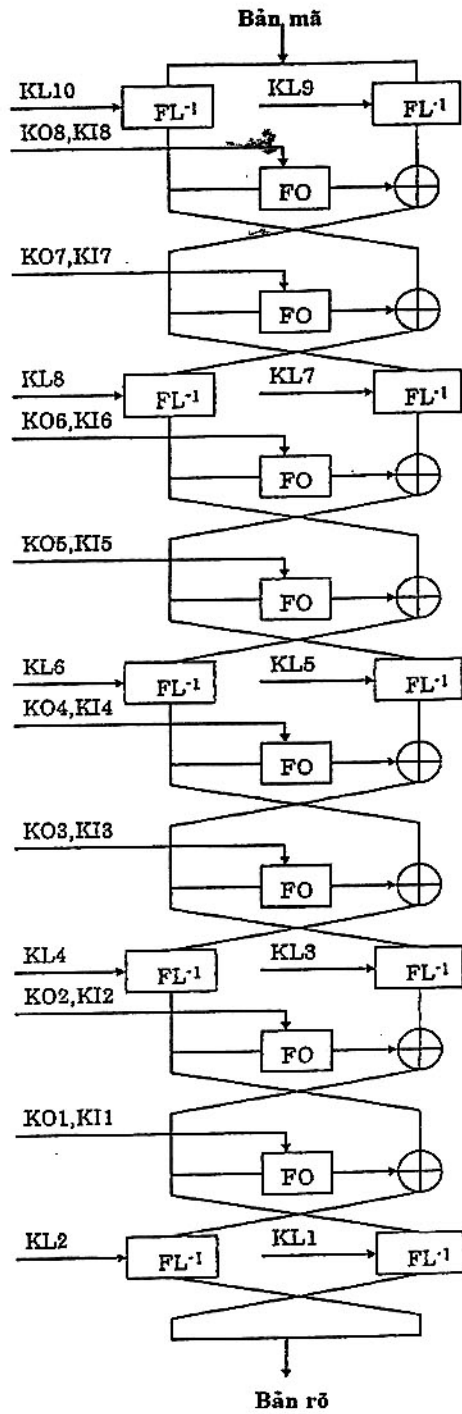
$$(2) Y_R = (X_L \wedge KL_{iL}) \oplus X_R$$

$$(3) Y_L = X_L \oplus (Y_R \vee KL_{iR})$$

$$(4) Y_{32} = Y_L \parallel Y_R$$



Hình 1 – Thủ tục mã hóa



Hình 2 – Thủ tục giải mã

4.3.4.3 Hàm FL⁻¹

Hàm FL⁻¹ là hàm ngược của hàm FL, chỉ được áp dụng vào phép giải mã và được chỉ ra trên Hình 4. Hàm FL⁻¹ được xác định như sau (X và Y là dữ liệu, KL là khóa):

$$(1) Y_{32} = Y_L \parallel Y_R, KL_i = KL_{iL} \parallel KL_{iR}$$

$$(2) X_L = Y_L \oplus (Y_R \vee KL_{iR})$$

$$(3) X_R = (X_L \wedge KL_{iL}) \oplus Y_R$$

$$(4) X_{32} = X_L \parallel X_R$$

4.3.4.4 Hàm FO

Hàm FO trình bày trên Hình 5, được sử dụng trong mã hóa và giải mã. Hàm FO được xác định như sau (X và Y là dữ liệu, KO và KI là khóa):

$$(1) X_{32} = L_0 \parallel R_0$$

$$KO_i = KO_{i1} \parallel KO_{i2} \parallel KO_{i3} \parallel KO_{i4}, KI_i = KI_{i1} \parallel KI_{i2} \parallel KI_{i3}$$

$$(2) \text{Với } j = 1, 2, 3$$

$$R_j = FI(L_{j-1} \oplus KO_{ij}, KI_{ij}) \oplus R_{j-1}$$

$$L_j = R_{j-1}$$

$$(3) Y_{32} = (L_3 \oplus KO_{i4}) \parallel R_3$$

4.3.4.5 Hàm FI

Hàm FI được sử dụng để mã hóa, giải mã và cho lược đồ khóa (schedule). Hàm FI được trình bày trên Hình 6, ở đây **Extnd** là phép toán mở rộng bằng các bit 0 gồm 7 đến 9 bit, bằng cách ghép 2 bit ở phía bên trái và **Trunc** là phép cắt hai bit phía bên trái. Hàm FI được xác định như sau (X và Y là dữ liệu, KI là khóa):

$$(1) X_{16} = L_0 (9 \text{ bit}) \parallel R_0 (7 \text{ bit}), KI_{ij} = KI_{ijL} \parallel KI_{ijR}$$

$$(2) R_1 = S_9(L_0) \oplus \text{Extnd}(R_0)$$

$$(3) L_1 = R_0$$

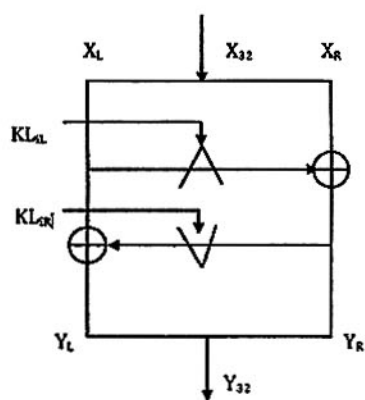
$$(4) R_2 = S7(L_1) \oplus \text{Trunc}(R_1) \oplus KI_{ijL}$$

$$(5) L_2 = R_1 \oplus KI_{ijR}$$

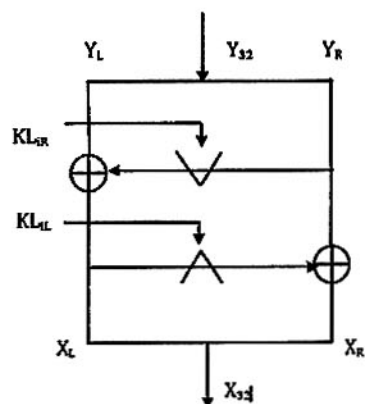
$$(6) R_3 = S_9(L_2) \oplus \text{Extnd}(R_2)$$

$$(7) L_3 = R_2$$

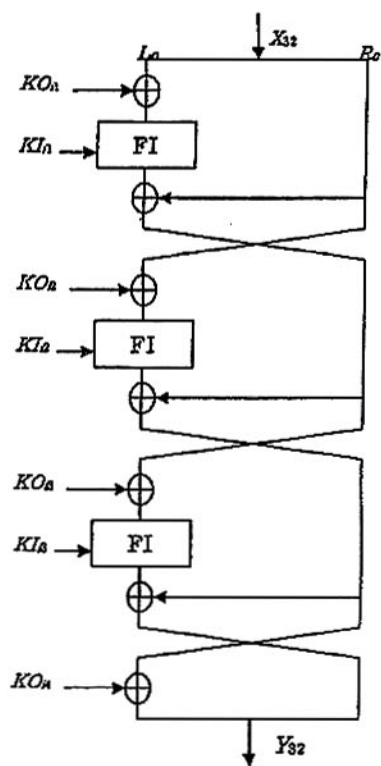
$$(8) Y_{16} = L_3 \parallel R_3$$



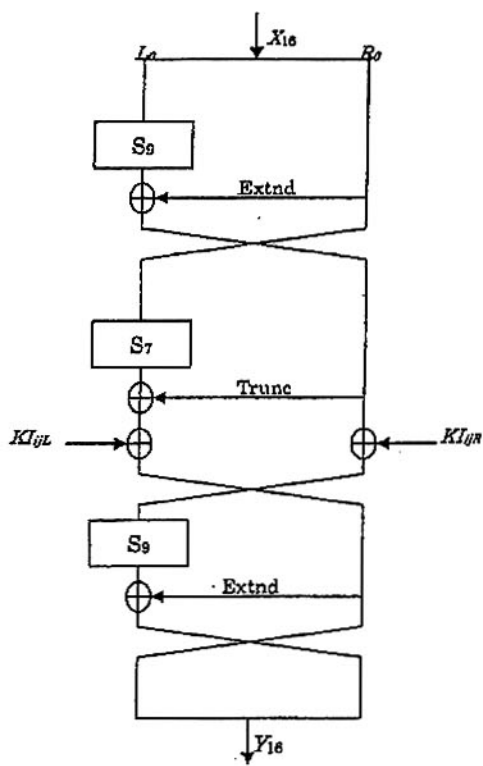
Hình 3 – Hàm FL



Hình 4 – Hàm FL⁻¹



Hình 5 – Hàm FO



Hình 6 – Hàm FI

4.3.4.6 Bảng tra cứu S_7 và S_9

S_7 là bảng tra cứu song ánh nhận đầu vào 7 bit và cho đầu ra 7 bit. S_9 là bảng tra cứu song ánh nhận đầu vào 9 bit và cho đầu ra 9 bit. Bảng 2 và 3 xác định các bảng tra cứu này ở dạng hệ đếm Hexa. S_7 và S_9 cũng có thể được mô tả dưới dạng đại số đơn giản trên trường $GF(2)$ được chỉ ra tại Điều C.2.

Ví dụ, nếu đầu vào của S_7 là {53}, thì giá trị thay thế được xác định bởi giao của hàng "5" và cột "3" với chỉ số "3" trên Bảng 2. Kết quả S_7 nhận được giá trị {57}.

Bảng 2 – S_7

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	1b	32	33	5a	3b	10	17	54	5b	1a	72	73	6b	2c	66	49
1	1f	24	13	6c	37	2e	3f	4a	5d	0f	40	56	25	51	1c	04
2	0b	46	20	0d	7b	35	44	42	2b	1e	41	14	4b	79	15	6f
3	0e	55	09	36	74	0c	67	53	28	0a	7e	38	02	07	60	29
4	19	12	65	2f	30	39	08	68	5f	78	2a	4c	64	45	75	3d
5	59	48	03	57	7c	4f	62	3c	1d	21	5e	27	6a	70	4d	3a
6	01	6d	6e	63	18	77	23	05	26	76	00	31	2d	7a	7f	61
7	50	22	11	06	47	16	52	4e	71	3e	69	43	34	5c	58	7d

Bảng 3 – S_9

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	1c3	0cb	153	19f	1e3	0e9	0fb	035	181	0b9	117	1eb	133	009	02d	0d3
01	0c7	14a	037	07e	0eb	164	193	1d8	0a3	11e	055	02c	01d	1a2	163	118
02	14b	152	1d2	00f	02b	030	13a	0e5	111	138	18e	063	0e3	0c8	1f4	01b
03	001	09d	0f8	1a0	16d	1f3	01c	146	07d	0d1	082	1ea	183	12d	0f4	19e
04	id3	0dd	1e2	128	1e0	0ec	059	091	0i1	12f	026	0de	0b0	18c	10f	1f7
05	0e7	16c	0b6	0f9	0d8	151	101	14c	103	0b8	154	12b	1ae	017	071	00c
06	047	058	07f	1a4	134	129	084	15d	19d	1b2	1a3	048	07c	051	1ea	023
07	13d	1a7	165	03b	042	0da	192	0ce	0c1	06b	09f	1f1	12c	184	0fa	196
08	1ei	169	17d	031	180	10a	094	1da	186	13e	11c	060	175	1ef	067	119
09	065	068	099	150	008	007	17c	0b7	024	019	0de	127	0db	0e4	1a9	052
0a	109	090	19c	1c1	028	1b3	135	16a	176	0df	1e5	188	0c5	16e	1de	1b1
0b	0c3	1df	036	0ee	1ee	0f0	093	049	09a	1b6	069	081	125	00b	05e	0b4
0c	149	ic7	174	03e	13b	1b7	08e	1c6	0ae	010	095	1ef	04e	0f2	1fd	085
0d	0fd	0f6	0a0	16f	083	08a	156	09b	13c	107	167	098	1d0	1e9	003	1fe
0e	0bd	122	089	0d2	18f	012	033	06a	142	0ed	170	11b	0e2	14f	158	131

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0f	147	05d	113	1ed	079	161	1a5	179	09e	1b4	0cc	022	132	01a	0e8	004
10	187	1ed	197	039	1bf	1d7	027	18b	0c6	09c	0d0	14e	06c	034	1f2	06e
11	0ca	r25	0ba	191	0fe	013	106	02f	1ad	172	1db	0c0	10b	1d6	0f5	1ec
12	10d	076	114	1ab	075	10c	1e4	159	054	11f	04b	0c4	1be	0f7	029	0a4
13	00e	1f0	077	04d	17a	086	08b	0b3	171	0bf	10e	104	097	15b	160	168
14	0d7	0bb	066	1ce	0fc	092	1c5	06f	016	04a	0a1	139	0af	0f1	190	00a
15	1aa	143	17b	056	18d	166	0d4	1fb	14d	194	19a	087	1f8	123	0a7	1b8
16	141	03c	1f9	140	02a	155	11a	1a1	198	0d5	126	1af	061	12e	157	1dc
17	072	18a	0aa	096	115	0ef	045	07b	08d	145	053	05f	178	0b2	02e	020
18	1d5	03f	1c9	1e7	1ac	044	038	014	0bi	16b	0ab	0b5	05a	182	1c8	1d4
19	018	177	064	0cf	06d	100	199	130	15a	005	120	1bb	1bd	0e0	04f	0d6
1a	13f	1c4	12a	015	006	0ff	19b	0a6	043	088	050	15f	1e8	121	073	17e
1b	0bc	0c2	0c9	173	189	1f5	074	1ee	1e6	1a8	195	01f	041	00d	1ba	032
1c	03d	1d1	080	0a8	057	1b9	162	148	0d9	105	062	07a	021	1ff	112	108
1d	1c0	0a9	1id	1b0	1a6	0cd	0f3	05c	102	05b	1d9	144	1f6	0ad	0a5	03a
1e	1cb	136	17f	046	0e1	10e	1dd	0e6	137	1fa	185	08c	08f	040	1b5	0be
1f	078	000	0ac	110	15e	124	002	1bc	0a2	0ea	070	1fc	116	15c	04c	1c2

4.3.5 Lược đồ khóa MISTY1

Lược đồ khóa nhận khóa 128 bit K và đưa ra khóa con 128 bit K' , như được chỉ ra ở hình dưới đây. Lược đồ khóa được cho trên Hình 7.

Phép lập lược đồ khóa được định nghĩa như sau:

$$(1) K = K_1 \| K_2 \| K_3 \| K_4 \| K_5 \| K_6 \| K_7 \| K_8$$

(2) với $i = 1, \dots, 7$:

$$K'_i = FI(K_i, K_{i+1})$$

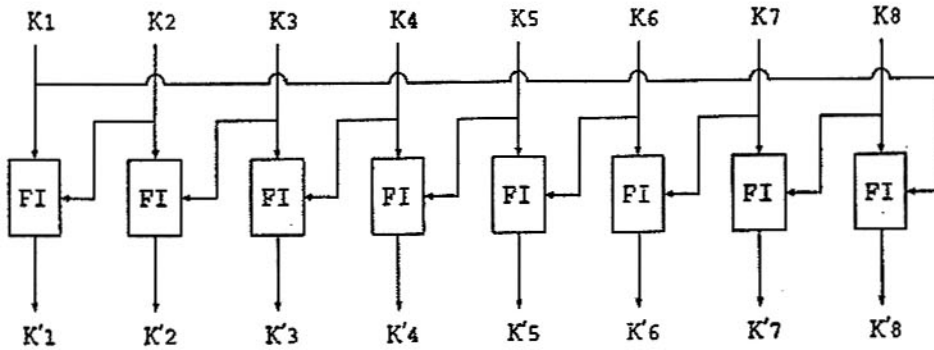
$$(3) K'_8 = FI(K_8, K_1)$$

$$(4) K' = K'_1 \| K'_2 \| K'_3 \| K'_4 \| K'_5 \| K'_6 \| K'_7 \| K'_8$$

$$(5) KO_{i1} = K_i, KO_{i2} = K_{i+2}, KO_{i3} = K_{i+7}, KO_{i4} = K_{i+4}, Kl_{i1} = K'_{i+5}, Kl_{i2} = K'_{i+1}, Kl_{i3} = K'_{i+3}, (i = 1, \dots, 8)$$

$$KL_{iL} = K_{(i+1)/2} \text{ (i lẻ) hoặc } K'_{(i/2)+2} \text{ (i chẵn)}, KL_{iR} = K'_{(i+2)/2+6} \text{ (i lẻ) hoặc } K_{(i/2)+4} \text{ (i chẵn)} \text{ (} i = 1, \dots, 10)$$

CHÚ THÍCH Khi giá trị hậu tố lớn hơn 8, thì trừ đi 8.



Hình 7- Lược đồ khóa MISTY1

4.4 CAST-128

4.4.1 Thuật toán CAST-128

Thuật toán CAST-128 là mã khối đối xứng, có thể xử lí các khối 64 bit, sử dụng khóa mật mã độ dài 128 bit với 16 vòng.

CHÚ THÍCH Độ dài khóa của phiên bản CAST-128 gốc thay đổi từ 40 đến 128 bit. Tuy nhiên trong tiêu chuẩn này của bộ TCVN 11367 (ISO/IEC-18033) chỉ sử dụng phương án khóa 128 bit.

4.4.2 Mã hóa theo CAST-128

Phép biến đổi khối 64 -bit P thành khối 64-bit C được xác định như sau (K_m và K_r là khóa):

$$(1) P = L_0 \| R_0$$

$$(2) \text{ với } j = 1, \dots, 16$$

$$L_j = R_{j-1}$$

$$R_j = L_{j-1} \oplus f(R_{j-1}, K_{m_j}, K_{r_j})$$

$$(3) C = R_{16} \| L_{16}$$

4.4.3 Giải mã CAST-128

Phép giải mã cũng tương tự như phép mã hóa đã mô tả ở trên, chỉ khác là các vòng (và các cặp khóa con) được sử dụng theo thứ tự ngược lại để tính (L_0, R_0) từ (R_{16}, L_{16}) .

4.4.4 Các hàm CAST-128

4.4.4.1 Các cặp khóa vòng

CAST-128 sử dụng một cặp khóa con cho mỗi vòng: 32 bit K_m được sử dụng như một khóa "mặt nạ" và 5-bit K_r được sử dụng như một khóa "hoán vị vòng".

4.4.4.2 Các f-hàm

Ba hàm vòng khác nhau được sử dụng trong phép mã hóa và phép giải mã, phụ thuộc vào số thứ tự của vòng. Các vòng như sau (" D " là đầu vào dữ liệu của f-hàm, và tương ứng " Ia " - " Id " tương ứng là các byte có có nghĩa nhất qua byte ít có nghĩa nhất của I).

$$\text{Type 1: } I = ((Km_1 + D) \lll_{Kr1})$$

$$f = ((S1[Ia] \oplus S2[Ib]) - S3[Ic]) + S4[Id]$$

$$\text{Type 2: } I = ((Km_1 \oplus D) \lll_{Kr1})$$

$$f = ((S1[Ia] - S2[Ib]) + S3[Ic]) \oplus S4[Id]$$

$$\text{Type 3: } I = ((Km_1 - D) \lll_{Kr1})$$

$$f = ((S1[Ia] + S2[Ib]) \oplus S3[Ic]) - S4[Id]$$

Các vòng 1, 4, 7, 10, 13 và 16 sử dụng hàm f kiểu 1 (Type1).

Các vòng 2, 5, 8, 11 và 14 sử dụng hàm f kiểu 2 (Type2).

Các vòng 3, 6, 9, 12 và 15 sử dụng hàm f kiểu 3 (Type3).

4.4.4.3 Các hộp thế

4.4.4.3.1 Các S-box S1 đến S8

Tám hộp thế được sử dụng: các S-box S1, S2, S3 và S4 là những S-box hàm vòng, S5, S6, S7 và S8 là những S-box lược đồ khóa.

4.4.4.3.2 S-Box S1

30fb40d4	9fa0ff0b	6beccd2f	3f258c7a	1e213f2f	9c004dd3	6003e540	cf9fc949
bfd4af27	88bbbdb5	e2034090	98d09675	6e63a0e0	15c361d2	c2e7661d	22d4ff8e
28683b6f	c07fd059	ff2379c8	775f50e2	43c340d3	df2f8656	887ca41a	a2d2bd2d
a1c9e0d6	346c4819	61b76d87	22540f2f	2abe32e1	aa54166b	22568e3a	a2d341d0
66db40c8	a784392f	004dff2f	2ab9d2de	97943fac	4a97c1d8	527644b7	b5f437a7
b82cbaef	d751d159	6ff7f0ed	5a097a1f	827b68d0	90ecf52e	22b0c054	bc8e5935
4b6d2f7f	50bb64a2	d2664910	bee5812d	b7332290	e93b159f	b48ee411	4bff345d
fd45c240	ad31973f	c4f6d02e	55fc8165	d5b1caaa	a1ac2dae	a2d4b76d	c19b0c50
882240f2	0c6e4f38	a4e4bfd7	4f5ba272	564c1d2f	c59c5319	b949e354	b04669fe
b1b6ab8a	c71358dd	6385c545	110f935d	57538ad5	6a390493	e63d37e0	2a54f6b3
3a787d5f	6276a0b5	19a6fcd6	7a42206a	29f9d4d5	f61b1891	bb72275e	aa508167
38901091	c6b505eb	84c7cb8c	2ad75a0f	874a1427	a2d1936b	2ad286af	aa56d291
d7894360	425c750d	93b39e26	187184c9	6c00b32d	73e2bb14	a0bec3c	54623779
64459eab	3f328b82	7718cf82	59a2cea6	04ee002e	89fe78e6	3fab0950	325ff6c2
81383f05	6963c5c8	76cb5ado	a49974c9	ca180dcf	380782d5	c7fa5cf6	8ac31511
35e79e13	47da91d0	f40f9086	a7e2419e	31366241	051ef495	aa573b04	4a805d8d
548300d0	00322a3c	bf64cddf	ba57a68e	75c6372b	50afd341	a7c13275	915a0bf5
6b54bfab	2b0b1426	ab4cc9d7	449ccd82	f7fbf265	ab85c5f3	1b55db94	aad4e324
cfa4bd3f	2deaa3e2	9e204d02	c8bd25ac	eadf55b3	d5bd9e98	e31231b2	2ad5ad6c
954329de	adbe4528	d8710f69	aa51c90f	aa786bf6	22513f1e	aa51a79b	2ad344cc

TCVN 11367-3:2016

7b5a41f0	d37cfbad	1b069505	41ece491	b4c332e6	032268d4	c9600acc	ce387e6d
bf6bb16c	6a70fb78	0d03d9c9	d4df39de	e01063da	4736f464	5ad328d8	b347cc96
75bb0fc3	98511bfb	4ffbcc35	b58bcf6a	e11f0abc	bfc5fe4a	a70aec10	ac39570a
3f04442f	6188b153	e0397a2e	5727cb79	9ceb418f	1cacd68d	2ad37c96	0175cb9d
c69dff09	c75b65f0	d9db40d8	ec0e7779	4744ead4	b11c3274	dd24cb9e	7e1c54bd
f01144f9	d2240eb1	9675b3fd	a3ac3755	d47c27af	51c85f4d	56907596	a5bb15e6
580304f0	ca042cf1	011a37ea	8dbfaadb	35ba3e4a	3526ffa0	c37b4d09	bc306ed9
98a52666	5648f725	ff5e569d	0cea63d0	7c63b2cf	700b45e1	d5ea50f1	85a92872
af1fbda7	d4234870	a7870bf3	2d3b4d79	42e04198	0cd0ede7	26470db8	f881814c
474d6ad7	7c0c5e5c	d1231959	381b7298	f5d2f4db	ab838653	6e2f1e23	83719c9e
bd91e046	9a56456e	dc39200c	20c8c571	962bda1c	e1e696ff	b141ab08	7cca89b9
1a69e783	02cc4843	a2f7c579	429ef47d	427b169c	5ac9f049	dd8f0f00	5c8165bf

4.4.4.3.3 S-Box S2

1f201094	ef0ba75b	69e3cf7e	393f4380	fe61cf7a	eec5207a	55889c94	72fc0651
ada7ef79	4e1d7235	d55a63ce	de0436ba	99c430ef	5f0c0794	18dcab7d	ald6eff3
a0b52f7b	59e83605	ee15b094	e9ffd909	dc440086	ef944459	ba83ccb3	e0c3cdfb
d1da4181	3b092ab1	f997f1c1	a5e6cf7b	01420ddb	e4e7ef5b	25a1ff41	e180f806
1fc41080	179bee7a	d37ac6a9	fe5830a4	98de8b7f	77e83f4e	79929269	24fa9f7b
e113c85b	acc40083	d7503525	f7ea615f	62143154	0d554b63	5d681121	c866c359
3d63cf73	cee234c0	d4d87e87	5c672b21	071f6181	39f7627f	361e3084	e4eb573b
602f64a4	d63acd9c	1bbc4635	9e81032d	2701f50c	99847ab4	a0e3df79	ba6cf38c
10843094	2537a95e	f46f6ffe	alff3b1f	208cfb6a	8f458c74	d9e0a227	4ec73a34
fc884f69	3e4de8df	ef0e0088	3559648d	8a45388c	1d804366	721d9bfd	a58684bb
e8256333	844e8212	128d8098	fed33fb4	ce280ae1	27e19ba5	d5a6c252	e49754bd
c5d655dd	eb667064	77840b4d	a1b6a801	84db26a9	e0b56714	21f043b7	e5d05860
54f03084	066ff472	a31aa153	dadc4755	b5625dbf	68561be6	83ca6b94	2d6ed23b
eccf01db	a6d3d0ba	b6803d5c	af77a709	33b4a34c	397bc8d6	5ee22b95	5f0e5304
81ed6f61	20e74364	b45e1378	de18639b	881ca122	b96726d1	8049a7e8	22b7da7b
5e552d25	5272d237	79d2951c	c60d894c	488cb402	1ba4fe5b	a4b09f6b	1ca815cf
a20c3005	8871df63	b9ae2fcb	ccc6c9e9	0beeff53	e3214517	b4542835	9f63293c
ee41e729	6e1d2a7c	50045286	1e6685f3	f33401c6	30a22c95	31a70850	60930f13
73f98417	a1269859	ec645c44	52c877a9	cdff33a6	a02b1741	7cbad9a2	2180036f
50d99c08	cb3f4861	c26bd765	64a3f6ab	80342676	25a75e7b	e4e6d1fc	20c710e6
cdf0b680	17844d3b	31eef84d	7e0824e4	2ccb49eb	846a3bae	8ff77888	ee5d60f6
7af75673	2fdd5cdb	a11631c1	30f66f43	b3faec54	157fd7fa	ef8579cc	d152de58
db2ffd5e	8f32ce19	306af97a	02f03ef8	99319ad5	c242fa0f	a7e3ebb0	c68e4906
b8da230c	80823028	dcdef3c8	d35fb171	088a1bc8	bec0c560	61a3c9e8	bca8f54d
c72feffa	22822e99	82c570b4	d8d94e89	8b1c34bc	301e16e6	273be979	b0ffea6
61d9b8c6	00b24869	b7ffce3f	08dc283b	43daf65a	f7e19798	7619b72f	8f1c9ba4
dc8637a0	16a7d3b1	9fc393b7	a7136eeb	c6bcc63e	1a513742	ef6828bc	520365d6
2d6a77ab	3527ed4b	821fd216	095c6e2e	db92f2fb	5eea29cb	145892f5	91584f7f
5483697b	2667a8cc	85196048	8c4bacea	833860d4	0d23e0f9	6c387e8a	0ae6d249
b284600c	d835731d	dcblc647	ac4c56ea	3ebd81b3	230eabb0	6438bc87	f0b5b1fa
8f5ea2b3	fc184642	0a036b7a	4fb089bd	649da589	a345415e	5c038323	3e5d3bb9

43d79572 7e6dd07c 06dfdf1e 6c6cc4ef 7160a539 73bfbef0 83877605 4523ecf1

4.4.4.3.4 S-Box S3

8edfc240	25fa5d9f	eb903dbf	e810c907	47607fff	369fe44b	8c1fc644	aececa90
beb1f9bf	eefbcaea	e8cf1950	51df07ae	920e8806	f0ad0548	e13c8d83	927010d5
11107d9f	07647db9	b2e3e4d4	3d4f285e	b9afa820	fade82e0	a067268b	8272792e
553fb2c0	489ae22b	d4ef9794	125e3fbc	21fffcee	825b1bfd	92 55c5ed	1257a240
4e1a8302	bae07fff	528246e7	8e57140e	3373f7bf	8c9f8188	a6fc4ee8	c982b5;,
a8c01db7	579fc264	67094f31	f2bd3f5f	40fff7c1	1fb78dfc	8e6bd2c1	437be59b
99b03dbf	b5dbc64b	638dc0e6	55819d99	a197c81c	4a012d6e	c5884a28	ccc36f71
b843c213	6c0743f1	8309893c	0feddd5f	2f7fe850	d7c07f7e	02507fbf	5afb9a04
a747d2d0	1651192e	af70bf3e	58c31380	5f98302e	727cc3c4	0a0fb402	0f7fef82
8c96fdad	5d2c2aae	8ee99a49	50da88b8	8427f4a0	1eac5790	796fb449	8252dc15
efbd7d9b	a672597d	ada840d8	45f54504	fa5d7403	e83ec305	4f91751a	925669c2
23efe941	a903f12e	60270df2	0276e4b6	94fd6574	927985b2	8276dbcb	02778176
f8af918d	4e48f79e	8f616ddf	e29d840e	842f7d83	340ce5c8	96bbb682	93b4b148
ef303cab	984faf28	779faf9b	92dc560d	224d1e20	8437aa88	7d29dc96	2756d3dc
8b907cee	b51fd240	e7c07ce3	e566b4a1	c3e9615e	3cf8209d	6094d1e3	cd9ca341
b843c213	6c0743f1	8309893c	0feddd5f	2f7fe850	d7c07f7e	02507fbf	5afb9a04
a747d2d0	1651192e	af70bf3e	58c31380	5f98302e	727cc3c4	0a0fb402	0f7fef82
8c96fdad	5d2c2aae	8ee99a49	50da88b8	8427f4a0	1eac5790	796fb449	8252dc15
efbd7d9b	a672597d	ada840d8	45f54504	fa5d7403	e83ec305	4f91751a	925669c2
23efe941	a903f12e	60270df2	0276e4b6	94fd6574	927985b2	8276dbcb	02778176
f8af918d	4e48f79e	8f616ddf	e29d840e	842f7d83	340ce5c8	96bbb682	93b4b148
ef303cab	984faf28	779faf9b	92dc560d	224d1e20	8437aa88	7d29dc96	2756d3dc
8b907cee	b51fd240	e7c07ce3	e566b4a1	c3e9615e	3cf8209d	6094d1e3	cd9ca341
5c76460e	00ea983b	d4d67881	fd47572c	f76cedd9	bda8229c	127dadaa	438a074e
1f97c090	081bdb8a	93a07ebe	b938ca15	97b03cff	3dc2c0f8	8d1ab2ec	64380e51
68cc7bfb	d90f2788	12490181	5de5ffd4	dd7ef86a	76a2e214	b9a40368	925d958f
4b39fffa	ba39aee9	a4ffd30b	faf7933b	6d498623	193cbcfa	27627545	825cf47a
61bd8ba0	d11e42d1	cead04f4	127ea392	10428db7	8272a972	9270c4a8	127de50b
285ba1c8	3c62f44f	35c0eaa5	e805d231	428929fb	b4fcdcf82	4fb66a53	0e7dc15b
1f081fab	108618ae	fcfd086d	f9ff2889	694bcc11	236a5cae	12deca4d	2c3f8cc5
d2d02dfe	f8ef5896	e4cf52aa	95155b67	494a488c	b9b6a80c	5c8f82bc	89d36b45
3a609437	ec00c9a9	44715253	0a874b49	d773bc40	7c34671c	02717ef6	4feb5536
a2d02fff	d2bf60c4	d43f03c0	50b4ef6d	07478cd1	006e1888	a2e53f55	b9e6d4bc
a2048016	97573833	d7207d67	de0f8f3d	72f87b33	abcc4f33	7688c55d	7b00a6b0
947b0001	570075d2	f9bb88f8	8942019e	4264a5ff	856302e0	72dbd92b	ee971b69
6ea22fde	5f08ae2b	af7616d	e5c98767	cf1febd2	61efc8c2	flac2571	cc8239c2
67214cb8	b1e583d1	b7dc3e62	7f10bdce	f90a5c38	0ff0443d	606e6dc6	60543a49
5727c148	2be98a1d	8ab41733	20e1be24	af96daCf	68458425	99833be5	600d457d
282f9350	8334b362	d91d1120	2b6d8da0	642b1e31	9c305a00	52bce688	1b03588a
f7baefd5	4142ed9c	a4315c11	83323ec5	dfef4636	a133c501	e9d3531c	ee353783

4.4.4.3.5 S-Box S4

9db30420	1fb6e9de	a7be7bef	d273a298	4a4f7bdb	64ad8c57	85510443	fa020ed1
7e287aff	e60fb663	095f35a1	79ebf120	fd059d43	6497b7b1	f3641f63	241e4adf
28147f5f	4fa2b8cd	c9430040	0cc32220	fdd30b30	c0a5374f	1d2d00d9	24147b15
ee4d111a	0fca5167	71ff904c	2d195ffe	1a05645f	0c13fefe	081b08ca	05170121

TCVN 11367-3:2016

80530100	e83e5efe	ac9af4f8	7fe72701	d2b8ee5f	06df4261	bb9e9b8a	7293ea25
ce84ffdf	f5718801	3dd64b04	a26f263b	7ed48400	547eebe6	446d4ca0	6cf3d6f5
2649abdf	aea0c7f5	36338cc1	503f7e93	d3772061	11b638e1	72500e03	f80eb2bb
abe0502e	ec8d77de	57971e81	e14f6746	c9335400	6920318f	081dbb99	ffc304a5
4d351805	7f3d5ce3	a6c866c6	5d5bccca9	daec6fea	9f926f91	9f46222f	3991467d
a5bf6d8e	1143c44f	43958302	d0214eeb	022083b8	3fb6180c	18f8931e	281658e6
26486e3e	8bd78a70	7477e4c1	b506e07c	f32d0a25	79098b02	e4eabb81	28123b23
69dead38	1574ca16	df871b62	211c40b7	a51a9ef9	0014377b	041e8ac8	09114003
bd59e4d2	e3d156d5	4fe876d5	2f91a340	557be8de	00eae4a7	0ce5c2ec	4db4bba6
e756bdf	dd3369ac	ec17b035	06572327	99afc8b0	56c8c391	6b65811c	5e146119
6e85cb75	be07c002	c2325577	893ff4ec	5bbfc92d	d0ec3b25	b7801ab7	8d6d3b24
20c763ef	c366a5fc	9c382880	0ace3205	aac9548a	ecald7c7	041afa32	1d16625a
6701902c	9b757a54	31d477f7	9126b031	36cc6fdb	c70b8b46	d9e66a48	56e55a79
026a4ceb	52437eff	2f8f76b4	0df980a5	8674cde3	edda04eb	17a9be04	2c18f4df
b7747f9d	ab2af7b4	efc34d20	2e096b7c	1741a254	e5b6a035	213d42f6	2c1c7c26
61c2f50f	6552daf9	d2c231f8	25130f69	d8167fa2	0418f2c8	001a96a6	0d1526ab
63315c21	5e0a72ec	49bafefd	187908d9	8d0dbd86	311170a7	3e9b640c	cc3e10d7
d5cad3b6	0caec388	f73001e1	6c728aff	71eae2a1	If9af36e	cfcbd12f	c1de8417
ac07be6b	cb44a1d8	8b9b0f56	013988c3	b1c52fca	b4be31cd	d8782806	12a3a4e2
6f7de532	58fd7eb6	d01ee900	24adffc2	f4990fc5	9711aac5	001d7b95	82e5e7d2
109873f6	6.13E+05	c32d9521	ada121ff	29908415	7fbb977f	af9eb3db	29c9ed2a
5ce2a465	a730f32c	d0aa3fe8	8a5cc091	d49e2ce7	0ce454a9	d60acd86	015f1919
77079103	dea03af6	78a8565e	dee356df	21f05cbe	8b75e387	b3c50651	b8a5c3ef
d8eeb6a2	e523be77	c2154529	2f69efdf	afe67afb	470c4b22	f3e0eb5b	d6cc9876
39e4460c	1fda8538	1987832f	ca007367	a99144f8	96b299ef	492fc295	9266beab
b5676e69	9bd3ddda	df7e052f	db25701c	1b5e51ee	65324e6	6afce36c	0316cc04
8644213e	b7dc59d0	7965291f	ccd6fd43	41823979	932bcdff6	b657c34d	4edfd282
7ae5290c	3cb9536b	851e20fe	9833557e	13ecf0b0	d3ffb372	3f85c5c1	0aef7ed2

4.4.4.3.6 S-Box S5

7ec90c04	2c6e74b9	9b0e66df	a6337911	b86a7fff	1dd358f5	44dd9d44	1731167f
08fbf1fa	e7f511cc	d2051b00	735aba00	2ab722d8	376381cb	acf6243a	69befd7a
e6a2e77f	f0c720cd	c4494816	ccf5c180	38851640	15b0a848	e68b18cb	4caadef
5f480a01	0412b2aa	259814fc	41d0efe2	4e40b48d	248eb6fb	8dba1cfe	41a99b02
1a550a04	ba8f65cb	7251f4e7	95a51725	c106ecd7	97a5980a	c539b9aa	4d79fe6a
f2f3f763	68af8040	ed0c9e56	11b4958b	e1eb5a88	8709e6b0	d7e07156	4e29fea7
6366e52d	02d1c000	c4ac8e05	9377f571	0c05372a	578535f2	2261be02	d642a0c9
dfla280	74b55bd2	682199c0	d421e5ec	53fb3ce8	c8adedb3	28a87fc9	3d959981
5c1ff900	fe38d399	0c4eff0b	062407ea	aa2f4fb1	4fb96976	90c79505	b0a8a774

ef55a1ff	e59ca2c2	a6b62d27	e66a4263	df65001f	0ec50966	dfdd55bc	29de0655
911e739a	17af8975	32c7911c	89f89468	0d01e980	524755f4	03b63cc9	0cc844b2
bcf3f0aa	87ac36e9	e53a7426	01b3d82b	1a9e7449	64ee2d7e	cddbb1da	01c94910
b868bf80	0d26f3fd	9342ede7	04a5c284	636737b6	50f5b616	f24766e3	8eca36c1
136e05db	fef18391	fb887a37	d6e7f7d4	c7fb7dc9	3063fcdf	b6f589de	ec2941da
26e46695	b7566419	f654efc5	d08d58b7	48925401	clbacb7f	e5ff550f	b6083049
5bb5d0e8	87d72e5a	ab6a6ee1	223a66ce	c62bf3cd	9e0885f9	68cb3e47	086c010f
a21de820	d18b69de	f3f65777	fa02c3f6	407edac3	ccb3d550	1793084d	b0d70eba
0ab378d5	d951fb0c	ded7da56	4124bbe4	94ca0b56	0f5755d1	e0e1e56e	6184b5be
580a249f	94f74bc0	e327888e	9f7b5561	c3dc0280	05687715	646c6bd7	44904db3
66b4f0a3	c0f1648a	697ed5af	49e92ff6	309e374f	2cb6356a	85808573	4991f840
76f0ae02	083be84d	28421c9a	44489406	736e4cb8	c1092910	8bc95fc6	7d869cf4
134f616f	2e77118d	b31b2be1	aa90b472	3ca5d717	7d161bba	9cad9010	af462ba2
9fe459d2	45d34559	d9f2da13	dbc65487	f3e4f94e	176d486f	097c13ea	631da5c7
445f7382	175683f4	cdc66a97	70be0288	b3cdcf72	6e5dd2f3	20936079	459b80a5
be60e2db	a9c23101	eba5315c	224e42f2	1c5c1572	f6721b2c	1ad2fff3	8c25404e
324ed72f	4067b7fd	0523138e	5ca3cb78	dc0fd66e	75922283	784d6b17	58ebb16e
44094f85	3f481d87	fcfeae7b	77b5ff76	8c2302bf	aaf47556	5f46b02a	2b092801
3d38f5f7	0ca81f36	52af4a8a	66d5e7c0	df3b0874	95055110	1b5ad7a8	f61ed5ad
6cf6e479	20758184	d0cef.65	88f7be58	4a046826	0ff6f8f3	a09c7f70	5346aba0
5ce96c28	e176eda3	6bac307f	376829d2	85360fa9	17e3fe2a	24b79767	f5a96b20
d6cd2595	68ff1ebf	7555442c	f19f06be	f9e0659a	eeb9491d	34010718	bb30cab8
e822fe15	88570993	750e6249	da627e55	5e76ffa8	b1534546	6d47de08	efe9e7d4

4.4.4.3.7 S-Box S6

f6fa8f9d	2cac6ce1	4ca34867	e2337f7c	95db08e7	016843b4	eced5cbc	325553ac
bf9f0960	dfa1e2ed	83f0579d	63ed86b9	1ab6a6b8	de5ebe39	f38ff732	8989b138
33f14961	c01937bd	f506c6da	e4625e7e	a308ea99	4e23e33c	79cbd7cc	48a14367
a314 9619	fec94bd5	a114174a	ea01866	a084db2d	09a0486f	a888614a	2900af98
01665991	e1992863	c8f30c60	2e78ef3c	d0d51932	cf0fec14	f7ca07d2	d0a82072
fd41197e	9305a6b0	e86be3da	74bed3cd	372da53c	4c7f4448	dab5d440	6dba0ec3
083919a7	9fbaeed9	49dbcfb0	4e670c53	5c3d9c01	64bdb941	2c0e636a	ba7dd9cd
ea6f7388	e70bc762	35f29adb	5c4cdd8d	f0d48d8c	b88153e2	08a19866	Iae2eac8
284caf89	aa928223	9334be53	3b3a21bf	16434be3	9aea3906	efe8c36e	f890cdd9
80226dae	c340a4a3	df7e9c09	a694a807	5b7c5ecc	221db3a6	9a69a02f	68818a54
ceb2296f	53c0843a	fe893655	25bfe68a	b4628abc	cf222ebf	25ac6f48	a9a99387
53bddb65	e10ffbe7	e967fd78	0ba93563	8e342bc1	e8a11be9	4980740d	c8087dfc
8de4bf99	a11101a0	7fd37975	da5a26c0	e81f994f	9528cd89	fd339fed	b87834bf
5f04456d	22258698	c9c4c83b	2dc156be	4f628daa	57f55ec5	e2220abe	d2916ebf
4ec75b95	24f2c3c0	42d15d99	cd0d7fa0	7b6e27ff	a8dc8af0	7345c106	f41e232f
35162386	e6ea8926	3333b094	157ec6f2	372b74af	692573e4	e9a9d848	f3160289
3a62ef1d	a787e238	f3a5f676	74364853	20951063	4576698d	b6fad407	592af950

TCVN 11367-3:2016

36f73523	4cfb6e87	7da4cec0	6c152daa	cb0396a8	c50dfe5d	fcd707ab	0921c42f
89dff0bb	5fe2be78	448f4f33	754613c9	2b05d08d	48b9d585	dc049441	c8098f9b
7dede786	c39a3373	42410005	6a091751	0ef3c8a6	890072d6	28207682	a9a9f7be
bf32679d	d45b5b75	b353fd00	cbb0e358	830f220a	1f8fb214	d372cf08	cc3c4a13
8cf63166	061c87be	88c98f88	6062e397	47cf8e7a	b6c85283	3cc2acfb	3fc06976
4e8f0252	64d8314d	da3870e3	1e665459	c10908f0	513021a5	6c5b68b7	822f8aa0
3007cd3e	74719eef	dc872681	073340d4	7e432fd9	0c5ec241	8809286c	f592d891
08a930f6	957ef305	b7fbffbd	c266e96f	6fe4ac98	b173ecc0	bc60b42a	953498aa
fbalae12	2d4bd736	0f25faab	a4f3fceb	e2969123	257f0c3d	9348af49	361400bc
e8816f4a	3814f200	a3f94043	9c7a54c2	bc704f57	da41e7f9	c25ad33a	54f4a084
b17f5505	59357cbe	edbd15c8	7f97c5ab	ba5ac7b5	b6f6deaf	3a479c3a	5302da25
653d7e6a	54268d49	51a477ea	5017d55b	d7d25d88	44136c76	0404a8c8	b8e5a121
b81a928a	60ed5869	97c55b96	eaec991b	29935913	01fdb7f1	088e8dfa	9ab6f6f5
3b4cbf9f	4a5de3ab	e6051d35	a0e1d855	d36b4cf1	f544edeb	b0e93524	bebb8fbd
a2d762cf	49c92f54	38b5f331	7128a454	48392905	a65b1db8	851c97bd	d675cf2f

4.4.4.3.8 S-Box S7

85e04019	332bf567	662dbfff	cfc65693	2a8d7f6f	ab9bc912	de6008a1	2028da1f
0227bce7	4d642916	18fac300	50f18b82	2cb2cb11	b232e75c	4b3695f2	b28707de
a05fbcf6	cd4181e9	e150210c	e24ef1bd	b168c381	fde4e789	5c79b0d8	1e8bfd43
4d495001	38be4341	913cee1d	92a79c3f	089766be	baeoadf4	1286becf	b6eacb19
2660c200	7565bde4	64241f7a	8248dca9	c3b3ad66	28136086	0bd8dfa8	356d1cf2
107789be	b3b2e9ce	0502aa8f	0bc0351e	166bf52a	eb12ff82	e3486911	d34d7516
4e7b3aff	5f43671b	9cf6e037	4981ac83	334266ce	8c9341b7	d0d854c0	cb3a6c88
47bc2829	4725ba37	a66ad22b	7ad61f1e	0c5cbafa	4437f107	b6e79962	42d2d816
0a961288	e1a5c06e	13749e67	72fc081a	b1d139f7	f9583745	cf19df58	bec3f756
c06eba30	07211b24	45c28829	c95e317f	bc8ec511	38bc46e9	c6e6fa14	bae8584a
ad4ebc46	468f508b	7829435f	f124183b	821dba9f	aff60ff4	ea2c4e6d	16e39264
92544a8b	009b4fc3	aba68ced	9ac96f78	06a5b79a	b2856e6e	1aec3ca9	be838688
0e0804e9	55f1be56	e7e5363b	b3a1f25d	f7debb85	61fe033c	16746233	3c034c28
da6d0c74	79aac56c	3ce4e1ad	51f0c802	98 f8f35a	1626a49f	eed82b29	1d382*3
0c4fb99a	bb325778	3ec6d97b	6e77a6a9	cb658b5c	d45230c7	2bd1408b	60c03eb7
b9068d78	a33754f4	f430c87d	c8a71302	b96d8c32	ebd4e7be	be8b9d2d	7979fb06
e7225308	8b75cf77	11ef8da4	e083c858	8d6b786f	5a6317a6	fa5cf7a0	5ada0033
f28ebfb0	f5b9c310	a0eac280	08b9767a	a3d9d2b0	79d34217	021a718d	0921c42f
2711fd60	438050e3	069908a8	3d7fedc4	826d2bef	4eeb8476	488dcf25	c8098f9b
28e74e41	c2610aca	3d49a9cf	bae3b9df	b65f8de6	92aeaf64	3ac7d5e6	a9a9f7be
f22b017d	a4173f70	dd1e16c3	15e0d7f9	50b1b887	2b9f4fd5	625aba82	cc3c4a13
2ec01b9c	15488aa9	d716e740	40055a2c	93d29a22	e32dbf9a	058745b9	3fc06976
d699296e	496cff6f	1c9f4986	dfe2ed07	b87242d1	19de7eae	053e561a	822f8aa0
66626c1c	7154c24c	ea082b2a	93eb2939	17dcb0f0	58d4f2ae	9ea294fb	f592d891
9883fe66	2ec40581	763953c3	01d6692e	d3a0c108	a1e7160e	e4f2dfa6	953498aa
74904698	4c2b0edd	4f757656	5d393378	a132234f	3d321c5d	c3f5e194	361400bc
c79f022f	3c997e7e	5e4f9504	3ffaafbbd	76f7ad0e	296693f4	3d1fce6f	54f4a084
d3b5ab34	f72bf9b7	1b0434c0	4e72b567	5592a33d	b5229301	cfcd2a87f	5302da25

1814386b	30bcc33d	38a0c07d	fd1606f2	c363519b	589dd390	5479f8e6	b8e5a121
97fd61a9	ea7759f4	2d57539d	569a58cf	e84e63ad	462e1b78	6580f87e	9ab6f6f5
91da55f4	40a230f3	d1988f35	b6e318d2	3ffa50bc	3d40f021	c3c0bdae	bebb8fbd
518f36b2	84bd1370	0fedce83	878ddada	f2a279c7	94e01be8	90716f4b	d675cf2f

4.4.4.3.9 S-Box S8

e216300d	bbddfffc	a7ebdabd	35648095	7789f8b7	e6c1121b	0e241600	052ce8b5
11a9cfb0	e5952f11	ece7990a	9386d174	2a42931c	76e38111	b12def3a	37ddddfc
de9adeb1	0a0cc32c	be197029	84a00940	bb243a0f	b4d137cf	b44e79f0	049eedfd
0b15a15d	480d3168	8bbbde5a	669ded42	c7ece831	3f8f95e7	72df191b	7580330d
94074251	5c7dcdfa	abbe6d63	aa402164	b301d40a	02e7alca	53571dae	7a3182a2
12a8ddec	fdaa335d	176f43e8	71fb46d4	38129022	ce949ad4	b84769ad	965bd862
82f3d055	66fb9767	15b80b4e	1d5b47a0	4cfde06f	c28ec4b8	57e8726e	64 7a78fc
99865d44	608bd593	6c200e03	39dc5ff6	5d0b00a3	ae63aff2	7e8bd632	70108c0c
bbd35049	2998df04	980cf42a	9b6df491	9e7edd53	06918548	58cb7e07	3b74ef2e
522fffb1	d24708cc	1c7e27cd	a4eb215b	3cfl2e2e	19b47a38	424f7618	35856039
9d17dee7	27eb35e6	c9aff67b	36baf5b8	09c467cd	c18910b1	e11dbf7b	06cd1af8
7170c608	2d5e3354	d4de495a	64c6d006	bcc0c62c	3dd00db3	708f8f34	77d51b42
264f620f	24b8d2bf	15c1b79e	46a52564	f8d7e54e	3e378160	7895cda5	859c15a5
e6459788	c37bc75f	db07ba0c	0676a3ab	7f229b1e	31842e7b	24259fd7	f8bef472
835ffcb8	6df4c1f2	96f5b195	fd0af0fc	b0fe134c	e2506d3d	4f9b12ea	f215f225
a223736f	9fb4c428	25d04979	34c713f8	c4618187	ea7a6e98	7cd16efc	1436876c
f1544107	bedeee14	56e9af27	a04aa441	3cf7c899	92ecbae6	dd67016d	151682eb
a842eedf	faba60b4	f1907b75	20e3030f	24d8c29e	e139673b	efa63fb8	71873054
b6f2cf3b	9f326442	cb15a4cc	b01a4504	f1e47d8d	844a1be5	bae7dfdc	42cbda70
cd7dae0a	57e85b7a	d53f5af6	20cf4d8c	cea4d428	79d130a4	3486ebfb	33d3cddc
77853b53	37effcb5	c5068778	e580b3e6	4e68b8f4	c5c8b37e	0d809ea2	398feb7c
132a4f94	43b7950e	2fee7d1c	223613bd	dd06caa2	37df932b	c4248289	acf3ebc3
5715f6b7	ef3478dd	f267616f	c148cbe4	9052815e	5e410fab	b48a2465	2eda7fa4
e87b40e4	e98ea084	5889e9e1	efd390fc	dd07d35b	db485694	38d7e5b2	57720101
730edebc	5b643113	94917e4f	503c2fba	646f1282	7523d24a	e0779695	f9c17a8f
7a5b2121	d187b896	29263a4d	ba510cdf	81f47c9f	ad1163ed	ea7b5965	1a00726e
11403092	00da6d77	4a0cdd61	ad1f4603	605bdfb0	9eedc364	22ebe6a8	cee7d28a
a0e736a0	5564a6b9	10853209	c7eb8f37	2de705ca	8951570f	df09822b	bd691a6c
aa12e4f2	87451c0f	e0f6a27a	3ada4819	4cf1764f	0d771c2b	67cdb156	350d8384
5938fa0f	42399ef3	36997b07	0e84093d	4aa93e61	8360d87b	1fa98b0c	1149382c
e97625a5	0614d1b7	0e25244b	0c768347	589e8d82	0d2059d1	a466bb1e	f8da0a82
04f19130	ba6e4ec0	99265164	1ee7230d	50b2ad80	eae6801	8db2a283	ea8bf59e

4.4.5 Luợc đồ khóa CAST-128

Giả sử khóa 128-bit là $x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{Ax}x_{Bx}x_{Cx}x_{Dx}x_{Ex}x_{F}$, ở đây x_0 biểu diễn byte có trọng số lớn nhất và x_F biểu diễn byte có trọng số bé nhất, và z_0, \dots, z_F là các byte trung gian.

Các khóa con được tạo từ khóa $x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{Ax}x_{Bx}x_{Cx}x_{Dx}x_{Ex}x_{F}$ như sau:

$$z_0z_1z_2z_3 = x_0x_1x_2x_3 \oplus S5[x_D] \oplus S6[x_F] \oplus S7[x_C] \oplus S8[x_E] \oplus S7[x_8]$$

$$z_4z_5z_6z_7 = x_8x_9x_{Ax}x_B \oplus S5[z_0] \oplus S6[z_2] \oplus S7[z_1] \oplus S8[z_3] \oplus S8[x_A]$$

$$z_8z_9z_{Az}z_B = x_Cx_{Dx}x_{Ex}x_F \oplus S5[z_7] \oplus S6[z_6] \oplus S7[z_5] \oplus S8[z_4] \oplus S5[x_9]$$

TCVN 11367-3:2016

$$zCzDzEzF = x4x5x6x7 \oplus S5[zA] \oplus S6[z9] \oplus S7[zB] \oplus S8[z8] \oplus S6[xB]$$

$$K1 = S5[z8] \oplus S6[z9] \oplus S7[z7] \oplus S8[z6] \oplus S5[z2]$$

$$K2 = S5[zA] \oplus S6[zB] \oplus S7[z5] \oplus S8[z4] \oplus S6[z6]$$

$$K3 = S5[zC] \oplus S6[zD] \oplus S7[z3] \oplus S8[z2] \oplus S7[z9]$$

$$K4 = S5[zE] \oplus S6[zF] \oplus S7[z1] \oplus S8[z0] \oplus S8[zC]$$

$$x0x1x2x3 = z8z9zAzB \oplus S5[z5] \oplus S6[z7] \oplus S7[z4] \oplus S8[z6] \oplus S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \oplus S5[x0] \oplus S6[x2] \oplus S7[x1] \oplus S8[x3] \oplus S8[z2]$$

$$x8x9xAxB = z4z5z6z7 \oplus S5[x7] \oplus S6[x6] \oplus S7[x5] \oplus S8[x4] \oplus S5[z1]$$

$$xCxDxExF = zCzDzEzF \oplus S5[xA] \oplus S6[x9] \oplus S7[xB] \oplus S8[x8] \oplus S6[z3]$$

$$K5 = S5[x3] \oplus S6[x2] \oplus S7[xC] \oplus S8[xD] \oplus S5[x8]$$

$$K6 = S5[x1] \oplus S6[x0] \oplus S7[xE] \oplus S8[xF] \oplus S6[xD]$$

$$K7 = S5[x7] \oplus S6[x6] \oplus S7[x8] \oplus S8[x9] \oplus S7[x3]$$

$$K8 = S5[x5] \oplus S6[x4] \oplus S7[xA] \oplus S8[xB] \oplus S8[x7]$$

$$z0z1z2z3 = x0x1x2x3 \oplus S5[xD] \oplus S6[xF] \oplus S7[xC] \oplus S8[xE] \oplus S7[x8]$$

$$z4z5z6z7 = x8x9xAxB \oplus S5[z0] \oplus S6[z2] \oplus S7[z1] \oplus S8[z3] \oplus S8[xA]$$

$$z8z9zAzB = xCxDxExF \oplus S5[z7] \oplus S6[z6] \oplus S7[z5] \oplus S8[z4] \oplus S5[x9]$$

$$zCzDzEzF = x4x5x6x7 \oplus S5[zA] \oplus S6[z9] \oplus S7[zB] \oplus S8[z8] \oplus S6[xB]$$

$$K9 = S5[z3] \oplus S6[z2] \oplus S7[zC] \oplus S8[zD] \oplus S5[z9]$$

$$K10 = S5[z1] \oplus S6[z0] \oplus S7[zE] \oplus S8[zF] \oplus S6[zC]$$

$$K11 = S5[z7] \oplus S6[z6] \oplus S7[z8] \oplus S8[z9] \oplus S7[z2]$$

$$K12 = S5[z5] \oplus S6[z4] \oplus S7[zA] \oplus S8[zB] \oplus S8[z6]$$

$$x0x1x2x3 = z8z9zAzB \oplus S5[z5] \oplus S6[z7] \oplus S7[z4] \oplus S8[z6] \oplus S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \oplus S5[xQ] \oplus S6[x2] \oplus S7[x1] \oplus S8[x3] \oplus S8[z2]$$

$$x8x9xAxB = z4z5z6z7 \oplus S5[x7] \oplus S6[x6] \oplus S7[x5] \oplus S8[x4] \oplus S5[z1]$$

$$xCxDxExF = zCzDzEzF \oplus S5[xA] \oplus S6[x9] \oplus S7[xB] \oplus S8[x8] \oplus S6[z3]$$

$$K13 = S5[x8] \oplus S6[x9] \oplus S7[x7] \oplus S8[x6] \oplus S5[x3]$$

$$K14 = S5[xA] \oplus S6[xB] \oplus S7[x5] \oplus S8[x4] \oplus S6[x7]$$

$$K15 = S5[xC] \oplus S6[xD] \oplus S7[x3] \oplus S8[x2] \oplus S7[x8]$$

$$K16 = S5[xE] \oplus S6[xF] \oplus S7[x1] \oplus S8[x0] \oplus S8[xD]$$

CHÚ THÍCH Nếu còn lại giống với những gì được đưa ra ở trên, tiếp tục từ chuỗi cuối cùng được tạo ra x0...xF để sinh ra khóa K17 - K32.

$$z0z1z2z3 = x0x1x2x3 \oplus S5[xD] \oplus S6[xF] \oplus S7[xC] \oplus S8[xE] \oplus S7[x8]$$

$$z4z5z6z7 = x8x9xAxB \oplus S5[z0] \oplus S6[z2] \oplus S7[z1] \oplus S8[z3] \oplus S8[xA]$$

$$z8z9zAzB = xCxDxExF \oplus S5[z7] \oplus S6[z6] \oplus S7[z5] \oplus S8[z4] \oplus S5[x9]$$

$$zCzDzEzF = x4x5x6x7 \oplus S5[zA] \oplus S6[z9] \oplus S7[zB] \oplus S8[z8] \oplus S6[xB]$$

$$K17 = S5[z8] \oplus S6[z9] \oplus S7[z7] \oplus S8[z6] \oplus S5[z2]$$

$$K18 = S5[zA] \oplus S6[zB] \oplus S7[z5] \oplus S8[z4] \oplus S6[z6]$$

$$K19 = S5[zC] \oplus S6[zD] \oplus S7[z3] \oplus S8[z2] \oplus S7[z9]$$

$$K20 = S5[zE] \oplus S6[zF] \oplus S7[z1] \oplus S8[z0] \oplus S8[zC]$$

$$x0x1x2x3 = z8z9zAzB \oplus S5[z5] \oplus S6[z7] \oplus S7[z4] \oplus S8[z6] \oplus S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \oplus S5[x0] \oplus S6[x2] \oplus S7[x1] \oplus S8[x3] \oplus S8[z2]$$

$$x8x9xAxB = z4z5z6z7 \oplus S5[x7] \oplus S6[x6] \oplus S7[x5] \oplus S8[x4] \oplus S5[z1]$$

$$xCxDxEzF = zCzDzEzF \oplus S5[xA] \oplus S6[x9] \oplus S7[xB] \oplus S8[x8] \oplus S6[z3]$$

$$K21 = S5[x3] \oplus S6[x2] \oplus S7[xC] \oplus S8[xD] \oplus S5[x8]$$

$$K22 = S5[x1] \oplus S6[x0] \oplus S7[xE] \oplus S8[xF] \oplus S6[xD]$$

$$K23 = S5[x7] \oplus S6[x6] \oplus S7[x8] \oplus S8[x9] \oplus S7[x3]$$

$$K24 = S5[x5] \oplus S6[x4] \oplus S7[xA] \oplus S8[xB] \oplus S8[x7]$$

$$z0z1z2z3 = x0x1x2x3 \oplus S5[xD] \oplus S6[xF] \oplus S7[xC] \oplus S8[xE] \oplus S7[x8]$$

$$z4z5z6z7 = x8x9xAxB \oplus S5[z0] \oplus S6[z2] \oplus S7[z1] \oplus S8[z3] \oplus S8[xA]$$

$$z8z9zAzB = xCxDxEzF \oplus S5[z7] \oplus S6[z6] \oplus S7[z5] \oplus S8[z4] \oplus S5[x9]$$

$$zCzDzEzF = x4x5x6x7 \oplus S5[zA] \oplus S6[z9] \oplus S7[zB] \oplus S8[z8] \oplus S6[xB]$$

$$K25 = S5[z3] \oplus S6[z2] \oplus S7[zC] \oplus S8[zD] \oplus S5[z9]$$

$$K26 = S5[z1] \oplus S6[z0] \oplus S7[zE] \oplus S8[zF] \oplus S6[zC]$$

$$K27 = S5[z7] \oplus S6[z6] \oplus S7[z8] \oplus S8[z9] \oplus S7[z2]$$

$$K28 = S5[z5] \oplus S6[z4] \oplus S7[zA] \oplus S8[zB] \oplus S8[z6]$$

$$x0x1x2x3 = z8z9zAzB \oplus S5[z5] \oplus S6[z7] \oplus S7[z4] \oplus S8[z6] \oplus S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \oplus S5[x0] \oplus S6[x2] \oplus S7[x1] \oplus S8[x3] \oplus S8[z2]$$

$$x8x9xAxB = z4z5z6z7 \oplus S5[x7] \oplus S6[x6] \oplus S7[x5] \oplus S8[x4] \oplus S5[z1]$$

$$xCxDxEzF = zCzDzEzF \oplus S5[xA] \oplus S6[x9] \oplus S7[xB] \oplus S8[x8] \oplus S6[z3]$$

$$K29 = S5[x8] \oplus S6[x9] \oplus S7[x7] \oplus S8[x6] \oplus S5[x3]$$

$$K30 = S5[xA] \oplus S6[xB] \oplus S7[x5] \oplus S8[x4] \oplus S6[x7]$$

$$K31 = S5[xC] \oplus S6[xD] \oplus S7[x3] \oplus S8[x2] \oplus S7[x8]$$

$$K32 = S5[xE] \oplus S6[xF] \oplus S7[x1] \oplus S8[x0] \oplus S8[xD]$$

Bước tiếp theo được sử dụng để tạo ra mặt nạ khóa con và các khóa con hoán vị vòng.

Giả sử Km_1, \dots, Km_{16} là các mặt nạ khóa con 32-bit (mỗi khóa một vòng), và Kr_1, \dots, Kr_{16} là 32 bit các khóa con hoán vị (mỗi khóa một vòng); chỉ có 5 bit cuối cùng có trọng số thấp nhất được dùng trong mỗi vòng.

với $i = 1$ đến 16

$$Km_i = K_i$$

$$Kr_i = K_{16+i}$$

4.5 HIGHT

4.5.1 Thuật toán HIGHT

Thuật toán HIGHT là mã khối đối xứng, có thể xử lý các khối 64-bit sử dụng khóa mật mã độ dài 128-bit.

4.5.2 Phép mã hóa theo HIGHT

Phép mã hóa được chỉ ra trên Hình 8. Phép biến đổi khối 64-bit P thành khối 64-bit C được xác định như sau (WK_i và SK_i là các byte khóa trắng và các byte khóa con tương ứng được xác định trong 4.5.5):

$$(1) P = P_7 \parallel P_6 \parallel P_5 \parallel P_4 \parallel P_3 \parallel P_2 \parallel P_1 \parallel P_0 \quad (P_i \text{ là các bản rõ})$$

(2)

$$\begin{aligned} X_{0,0} &= P_0 \boxplus WK_0, & X_{0,1} &= P_1 \\ X_{0,2} &= P_2 \oplus WK_1, & X_{0,3} &= P_3 \\ X_{0,4} &= P_4 \boxplus WK_2, & X_{0,5} &= P_5 \\ X_{0,6} &= P_6 \oplus WK_3, & X_{0,7} &= P_7 \end{aligned}$$

(3) với $i = 0$ tới 30

$$X_{i+1,0} = X_{i,7} \oplus (F_0(X_{i,6}) \boxplus SK_{4i+3}), \quad X_{i+1,1} = X_{i,0}$$

$$X_{i+1,2} = X_{i,1} \boxplus (F_1(X_{i,0}) \oplus SK_{4i}), \quad X_{i+1,3} = X_{i,2}$$

$$X_{i+1,4} = X_{i,3} \oplus (F_0(X_{i,2}) \boxplus SK_{4i+1}), \quad X_{i+1,5} = X_{i,4}$$

$$X_{i+1,6} = X_{i,5} \boxplus (F_1(X_{i,4}) \oplus SK_{4i+2}), \quad X_{i+1,7} = X_{i,6}$$

Với $i = 31$

$$X_{i+1,0} = X_{i,0}, \quad X_{i+1,1} = X_{i,1} \boxplus (F_1(X_{i,0}) \oplus SK_{124})$$

$$X_{i+1,2} = X_{i,2}, \quad X_{i+1,3} = X_{i,3} \oplus (F_0(X_{i,2}) \boxplus SK_{125})$$

$$X_{i+1,4} = X_{i,4}, \quad X_{i+1,5} = X_{i,5} \boxplus (F_1(X_{i,4}) \oplus SK_{126})$$

$$X_{i+1,6} = X_{i,6}, \quad X_{i+1,7} = X_{i,7} \oplus (F_0(X_{i,6}) \boxplus SK_{127})$$

(4)

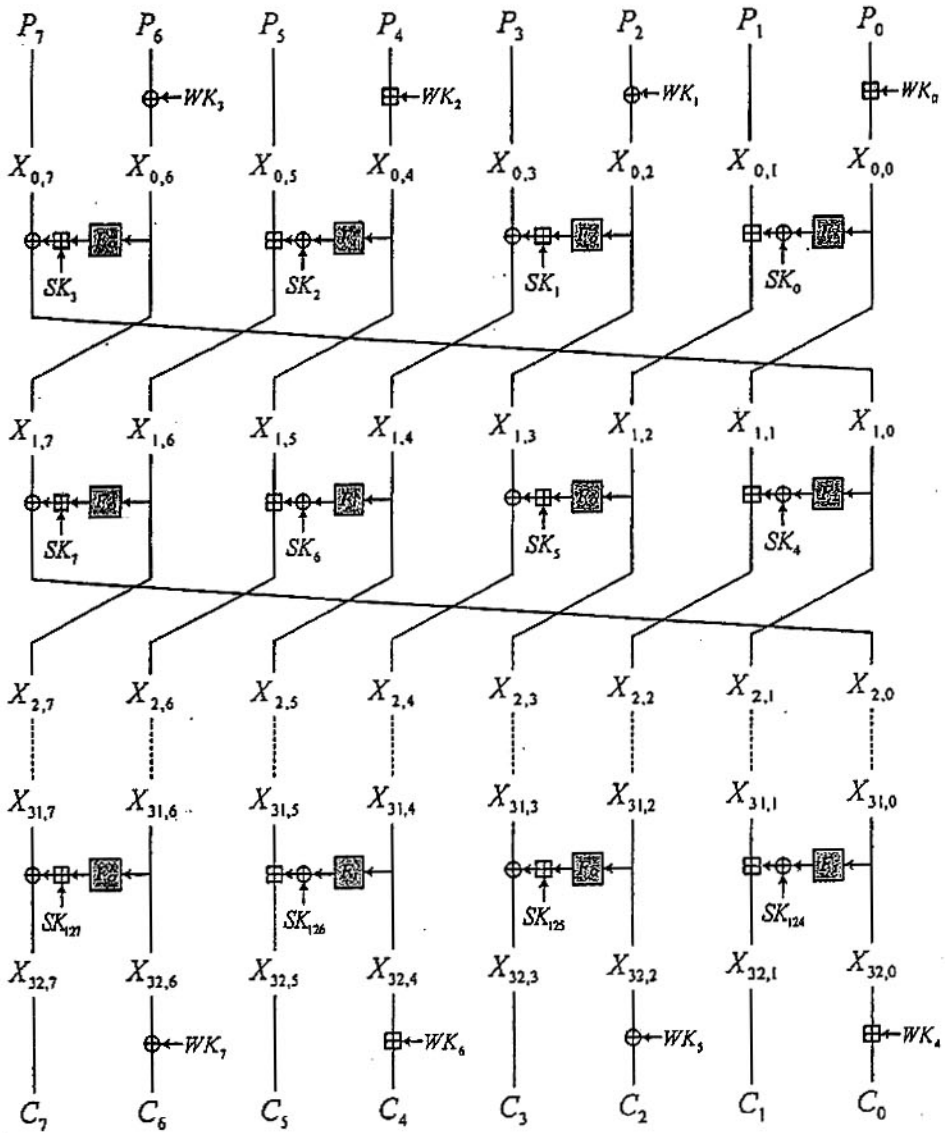
$$C_0 = X_{32,0} \boxplus WK_4, \quad C_1 = X_{32,1}$$

$$C_2 = X_{32,2} \oplus WK_5, \quad C_3 = X_{32,3}$$

$$C_4 = X_{32,4} \boxplus WK_6, \quad C_5 = X_{32,5}$$

$$C_6 = X_{32,6} \oplus WK_7, \quad C_7 = X_{32,7}$$

(5) $C = C_7 \parallel C_6 \parallel C_5 \parallel C_4 \parallel C_3 \parallel C_2 \parallel C_1 \parallel C_0$ (C_i - các byte bản mã)



Hình 8 – Thủ tục mã hóa theo HIGHT

4.5.3 Giải mã theo HIGHT

Phép toán giải mã cũng giống như phép toán mã hóa, trừ hai thay đổi sau đây:

- (1) Tất cả phép toán \boxplus được thay thế bằng phép \boxminus trừ phép toán \boxplus nối SK_i và đầu ra của F_0 .
- (2) Thứ tự áp dụng WK_i và SK_i ngược với thứ tự trong mã hóa.

4.5.4 Các hàm HIGHT

4.5.4.1 Các hàm F_0 và F_1

Thuật toán HIGHT sử dụng hai hàm F_0 và F_1 được định nghĩa dưới đây.

4.5.4.2 Hàm F_0

Hàm F_0 được sử dụng để mã hóa và giải mã. Hàm F_0 được xác định như sau:

$$F_0(x) = (x \lll 4) \oplus (x \lll 2) \oplus (x \lll 7)$$

4.5.4.3 Hàm F_1

Hàm F_1 được sử dụng để mã hóa và giải mã. Hàm F_1 được xác định như sau:

$$F_1(x) = (x \lll 3) \oplus (x \lll 4) \oplus (x \lll 6)$$

4.5.5 Lược đồ khóa HIGHT

Phần lược đồ khóa nhận khóa chủ 128-bit $K = K_{15} \| K_{14} \| \dots \| K_0$ và đưa ra 8 byte khóa trắng WK_i và 128 byte khóa con SK_i và xác định như sau.

với $i = 0, 1, 2, 3$:

$$WK_i = K_{i+12}$$

với $i = 4, 5, 6, 7$

$$WK_i = K_{i-4}$$

128 khóa con được sử dụng để mã hóa và giải mã, mỗi vòng có bốn khóa con. Việc tạo các khóa con được xác định như sau:

$$(1) \quad s_0 = 0, s_1 = 1, s_2 = 0, s_3 = 1, s_4 = 1, s_5 = 0, s_6 = 1$$

$$\delta_0 = s_6 \| s_5 \| s_4 \| s_3 \| s_2 \| s_1 \| s_0$$

$$(2) \quad \text{với } i = 1 \text{ đến } 127$$

$$s_{i+6} = s_{i+2} \oplus s_{i-1}$$

$$\delta_i = s_{i+6} \| s_{i+5} \| s_{i+4} \| s_{i+3} \| s_{i+2} \| s_{i+1} \| s_i$$

$$(3) \quad \text{với } i = 0 \text{ đến } 7$$

với $j = 0$ đến 7

$$SK_{16,i+j} = K_{j-l \bmod 8} \boxplus \delta_{16,i+j}$$

với $j = 0$ đến 7

$$SK_{16,i+j+8} = K_{(j-l \bmod 8)+8} \boxplus \delta_{16,i+j+8}$$

5 Mã khối 128-bit

5.1 Mở đầu

Điều khoản này đặc tả ba mã khối: AES ở Điều 5.2, Camellia ở Điều 5.3, và SEED ở Điều 5.4

5.2 AES

5.2.1 Thuật toán AES

Thuật toán AES là mã khối đối xứng, có thể xử lý các khối dữ liệu 128 bit, sử dụng khóa mật mã độ dài 128, 192 và 256 bit. Thuật toán AES còn được biết đến là Thuật toán Rijndael. Thuật toán AES có thể được sử dụng với ba độ dài khóa khác nhau như đã chỉ ra ở trên và do đó có ba tham chiếu là "AES-128", "AES-192", và "AES-256".

Trong thuật toán AES, độ dài của khối đầu vào và đầu ra là 128 bit (4 từ). Độ dài của khóa K là 128, 192 và 256 bit. Số vòng, Nr , là 10, 12 và 14, tùy thuộc độ dài khóa, được mô tả trên Bảng 4.

Bảng 4 – Số vòng trong AES

Độ dài khóa	Số vòng (Nr)
AES-128	10
AES-192	12
AES-256	14

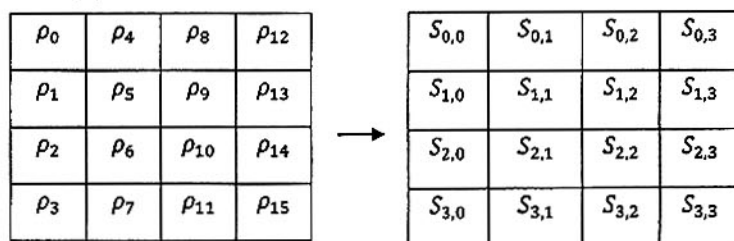
Đối với cả hai phép mã hóa và giải mã, thuật toán AES sử dụng hàm vòng bao gồm bốn phép biến đổi khác nhau trên byte: 1) phép thay thế byte sử dụng bảng thay thế (S-box), 2) dịch hàng của mảng Trạng thái bằng các phép bù khác nhau, 3) trộn dữ liệu trong mỗi cột của mảng Trạng thái, 4) bổ sung khóa vòng vào Trạng thái. Các phép biến đổi này (và các phép nghịch đảo của chúng) được mô tả trong Điều 5.2.4

Phép mã hóa và giải mã được mô tả tương ứng trong 5.2.2 và 5.2.3, lược đồ khóa tương ứng được mô tả trong 5.2.5

5.2.2 Phép mã hóa theo AES

Thuật toán AES gồm một dãy các phép toán được thực hiện trên mảng của các byte hai chiều, được gọi là Trạng thái. Trạng thái gồm bốn dòng byte, mỗi dòng chứa 4 byte. Trong mảng Trạng thái được kí hiệu bằng s , mỗi byte riêng biệt có hai chỉ số với số dòng r , $0 \leq r < 4$ và số cột c , với $0 \leq c < 4$. Trạng thái được kí hiệu là $S = (s_{r,c})$

Khi bắt đầu quá trình mã hóa, 16 byte của Trạng thái được khởi tạo với p_i byte bản rõ, tính từ trên xuống dưới và từ trái sang phải và được minh họa trên Hình 9.



Hình 9 – Khởi đầu Trạng thái

Sau khi cộng khóa vòng ban đầu, Trạng thái được biến đổi bằng cách thực thi hàm vòng Nr lần, với vòng cuối khác một ít với $Nr-1$ vòng đầu. Nội dung cuối cùng của Trạng thái chính là bản mã đầu ra.

Phép mã hóa đầy đủ có thể được mô tả như sau:

$$(1) \quad S = \text{AddRoundKey}(P, W_0)$$

(2) với $j = 1$ đến $Nr-1$

$S = \text{SubBytes}(S)$
 $S = \text{ShiftRow}(S)$
 $S = \text{MixColumns}(S)$
 $S = \text{AddRoundKey}(S, W_1)$

(3) $S = \text{SubBytes}(S), S = \text{ShiftRow}(S)$

(4) $C = \text{AddRoundKey}(S, W_{Nr})$

Các phép biến đổi riêng biệt $\text{SubBytes}()$, $\text{ShiftRows}()$, $\text{MixColumns}()$, $\text{AddRoundKey}()$ xử lý Trạng thái và được mô tả tại Điều 5.2.4. Tất cả Nr vòng đều giống nhau, trừ vòng cuối cùng không chứa phép biến đổi $\text{MixColumns}()$. Trong phép toán ở trên, mảng W chứa các khóa vòng được mô tả tại Điều 5.2.5

5.2.3 Phép giải mã AES

Tất cả các phép biến đổi sử dụng trong các phép mã hóa đều khả nghịch. Khi thực thi phép giải mã, đây các phép biến đổi được sử dụng trong phép mã hóa vẫn được duy trì, nhưng thay bằng các phép biến đổi ngược như sau.

Phép giải mã đầy đủ có thể mô tả như sau:

(1) $S = \text{AddRoundKey}(C, W_{Nr})$

(2) với $i = Nr-1$ xuống 1

$S = \text{ShiftRows}^{-1}(S)$
 $S = \text{SubBytes}^{-1}(S)$
 $S = \text{AddRoundKey}(S, W_i)$
 $S = \text{MixColumns}^{-1}(S)$

(3) $S = \text{ShiftRows}^{-1}(S)$

$S = \text{SubBytes}^{-1}(S)$

(4) $P = \text{AddRoundKey}(S, W_0)$

Các phép biến đổi $\text{SubBytes}^{-1}()$, $\text{ShiftRows}^{-1}()$, $\text{MixColumns}^{-1}()$ thực hiện xử lý Trạng thái và được mô tả tại Điều 5.2.4. Tất cả Nr vòng đều giống nhau, trừ vòng cuối không chứa phép biến đổi $\text{MixColumns}^{-1}()$. Việc tính các khóa vòng W được mô tả tại Điều 5.2.5

5.2.4 Các phép biến đổi AES

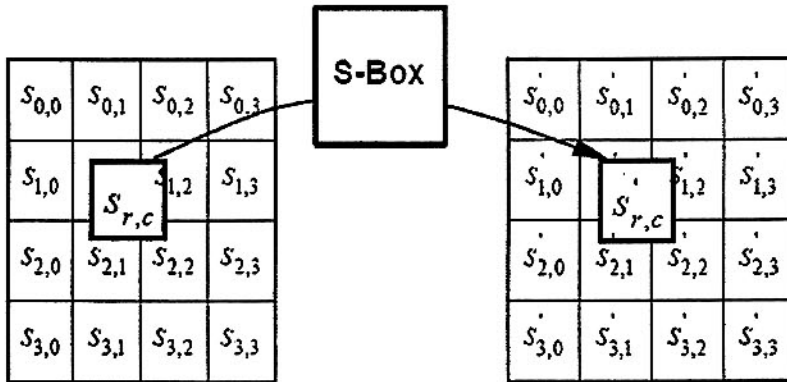
5.2.4.1 Các phép biến đổi xác định cho AES

Thuật toán AES sử dụng các phép biến đổi $\text{SubBytes}()$, $\text{SubBytes}^{-1}()$, $\text{ShiftRows}()$, $\text{ShiftRows}^{-1}()$, $\text{MixColumns}()$, $\text{MixColumns}^{-1}()$, và $\text{AddRoundKey}()$, được mô tả dưới đây.

5.2.4.1 Phép biến đổi $\text{subBytes}()$

Phép biến đổi $\text{SubBytes}()$ thực hiện thay thế mỗi byte Trạng thái $s_{i,j}$ bởi giá trị mới $s'_{i,j}$, bằng cách sử dụng bảng thay thế (S-box) khả nghịch.

Hình 10 minh họa tác động của phép biến đổi $\text{SubBytes}()$ lên bảng Trạng thái



Hình 10 – $\text{SubBytes}()$ áp dụng S-box cho từng byte của Trạng thái.

S-box được sử dụng trong phép biến đổi $\text{SubBytes}()$ và được trình bày theo hệ Hexa trên Bảng 5

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0	
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c1	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	1f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	be	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	ab
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e1	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Bảng 5 – Các S-box của AES

Ví dụ, nếu $s_{1,1} = \{53\}$ thì giá trị thay thế là giá trị nằm trên giao của hàng có chỉ số '5' và cột có chỉ số '3' của Bảng 5, điều này cho kết quả là $s'_{1,1}$ có giá trị $\{ed\}$.

5.2.4.3 Phép biến đổi $\text{SubBytes}^{-1}()$

$\text{SubBytes}^{-1}()$ là phép biến đổi nghịch đảo của phép biến đổi $\text{SubBytes}()$, trong đó S-box nghịch đảo được áp dụng cho từng byte của Trạng thái. Điều này đạt được bằng cách áp dụng phép biến đổi nghịch đảo được mô tả tại Điều 5.2.4.2

S-box nghịch đảo được sử dụng trong phép biến đổi $\text{SubBytes}^{-1}()$ được mô tả trong Bảng 6.

Bảng 6 – S-box nghịch đảo AES

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	sd	9d	84
6	90	ds	ab	00	8c	be	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	7	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e1	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	cb	d2	79	20	9a	db	c0	f	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	30	ec	5f
d	60	51	7	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

5.2.4.4 Phép biến đổi ShiftRows ()

Trong phép biến đổi ShiftRows (), các byte ở ba dòng cuối của Trạng thái được dịch vòng lên một số lượng bytes khác nhau (offsets). Hàng thứ nhất, hàng 0 được giữ nguyên (không dịch chuyển).

Cụ thể, phép biến đổi ShiftRows () được thực hiện như sau:

$$S'_{r,c} = S_{r,(c+r) \bmod 4} \quad \text{với } 0 < r < 4, \text{ và } 0 \leq c < 4, \text{ ở đây } r \text{ là số thứ tự của hàng.}$$

Theo đó sẽ các byte dịch chuyển sang trái (nghĩa là các giá trị thấp hơn của c trong một hàng cho trước), trong khi các byte phía trái ngoài cùng dịch vòng sang các vị trí phía phải ngoài cùng của hàng (tức là những giá trị cao hơn của c trong hàng cho trước)

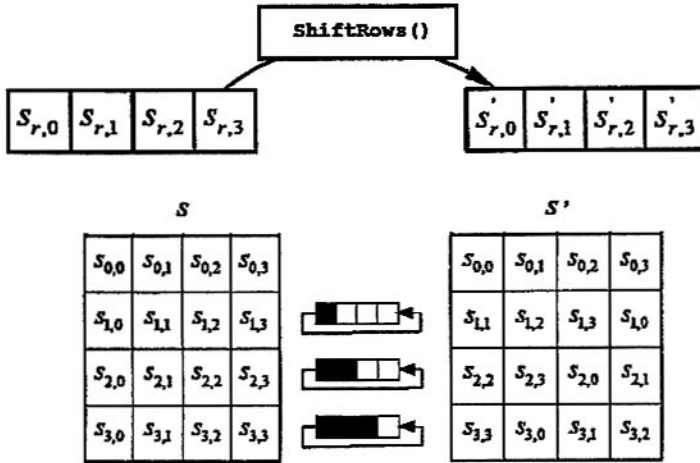
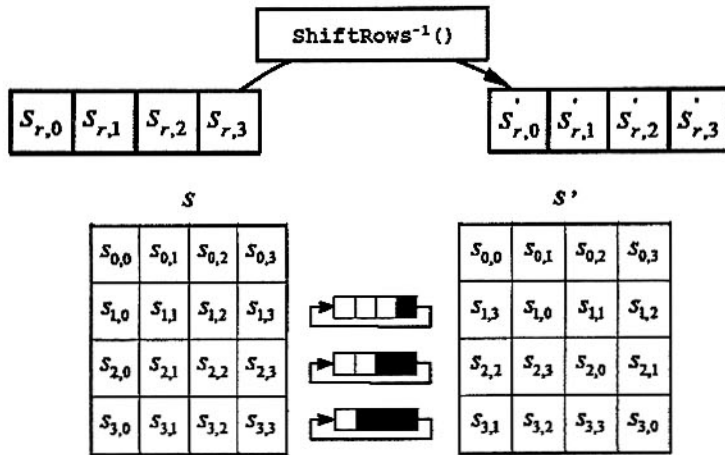
Hình 11 mô tả phép biến đổi ShiftRows (), trong đó các byte được dịch vòng sang trái.

5.2.4.5 Phép biến đổi ShiftRows⁻¹ ()

ShiftRows⁻¹ () là phép nghịch đảo của phép biến đổi ShiftRows (). Các bytes trong ba dòng cuối của Trạng thái được dịch vòng lên một số lượng bytes khác nhau. Dòng thứ nhất, dòng 0 không dịch chuyển. Ba dòng dưới cùng được dịch vòng lên $4-r$ bytes, ở đây r là số thứ tự vòng.

Cụ thể, phép biến đổi ShiftRows⁻¹ () được thực hiện như sau:

$$S'_{r,c} = S_{r,(c+r) \bmod 4} \quad \text{với } 0 < r < 4, \text{ và } 0 \leq c < 4.$$

Hình 12 mô tả phép biến đổi $\text{ShiftRows}^{-1}()$ Hình 11 – $\text{ShiftRows}()$ dịch vòng ba dòng cuối của Trạng tháiHình 12 – $\text{ShiftRows}^{-1}()$ dịch vòng ba dòng cuối của Trạng thái.

5.2.4.6 Phép biến đổi $\text{MixColumns}()$

Phép biến đổi $\text{MixColumns}()$ thao tác trên Trạng thái, thay mỗi cột bằng cột khác. Các cột của Trạng thái được xem như những đa thức trên trường $GF(2^8)$ và được nhân modulo $x+1$ với đa thức cố định $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. Phép nhân này có thể viết dưới dạng phép nhân ma trận:

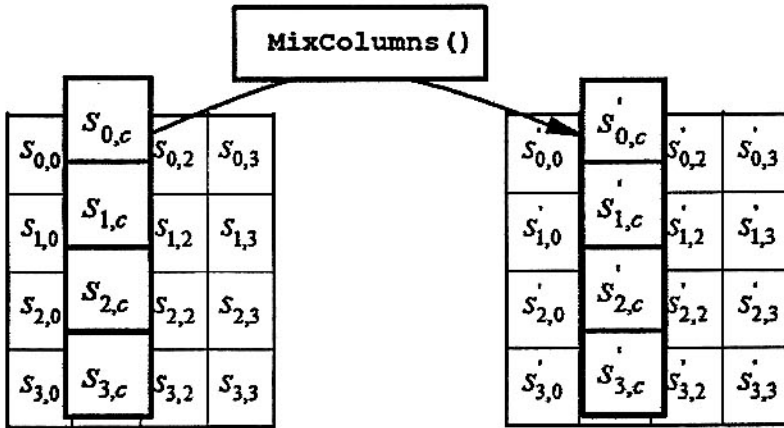
$$s'(x) = a(x) \otimes s(x): \quad \begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{với } 0 \leq c < 4.$$

Kết quả của phép nhân trên là bốn byte trong cột được thay thế như sau:

$$\begin{aligned} s'_{0,c} &= (\{02\} \cdot s_{0,c}) \oplus (\{03\} \cdot s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (\{02\} \cdot s_{1,c}) \oplus (\{03\} \cdot s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (\{02\} \cdot s_{2,c}) \oplus (\{03\} \cdot s_{3,c}) \\ s'_{3,c} &= (\{03\} \cdot s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \cdot s_{3,c}) \end{aligned}$$

Toán tử \oplus trong các biểu thức trên biểu thị phép cộng trong trường $GF(2^8)$, với phép cộng bit XOR. Phép nhân được thực hiện theo modulo của đa thức bất khả qui của trường. Trong trường hợp thuật toán AES đó là đa thức $x^8 + x^4 + x^3 + x + 1$.

Hình 13 mô tả phép biến đổi MixColumns ().



Hình 13 – MixColumns () thao tác trên Trạng thái, thay cột bằng cột khác

5.2.4.7 Phép biến đổi MixColumns⁻¹ ()

MixColumns⁻¹ () là phép biến đổi nghịch đảo của phép biến đổi MixColumns (). MixColumns⁻¹ () thao tác trên Trạng thái, thay mỗi cột bằng cột khác. Phép biến đổi này có thể biểu diễn dưới dạng phép nhân ma trận, ở đây mỗi byte được coi như một phần tử của trường hữu hạn $GF(2^8)$:

$$s^{-1}(x) = a^{-1}(x) \otimes s(x)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{với } 0 \leq c < 4.$$

Kết quả của của phép nhân là bốn byte trong mỗi cột được thay thế như sau:

$$\begin{aligned} s'_{0,c} &= (\{0e\} \cdot s_{0,c}) \oplus (\{0b\} \cdot s_{1,c}) \oplus (\{0d\} \cdot s_{2,c}) \oplus (\{09\} \cdot s_{3,c}) \\ s'_{1,c} &= (\{09\} \cdot s_{0,c}) \oplus (\{0d\} \cdot s_{1,c}) \oplus (\{0b\} \cdot s_{2,c}) \oplus (\{0e\} \cdot s_{3,c}) \\ s'_{2,c} &= (\{0d\} \cdot s_{0,c}) \oplus (\{0e\} \cdot s_{1,c}) \oplus (\{09\} \cdot s_{2,c}) \oplus (\{0b\} \cdot s_{3,c}) \\ s'_{3,c} &= (\{0b\} \cdot s_{0,c}) \oplus (\{09\} \cdot s_{1,c}) \oplus (\{0e\} \cdot s_{2,c}) \oplus (\{0d\} \cdot s_{3,c}) \end{aligned}$$

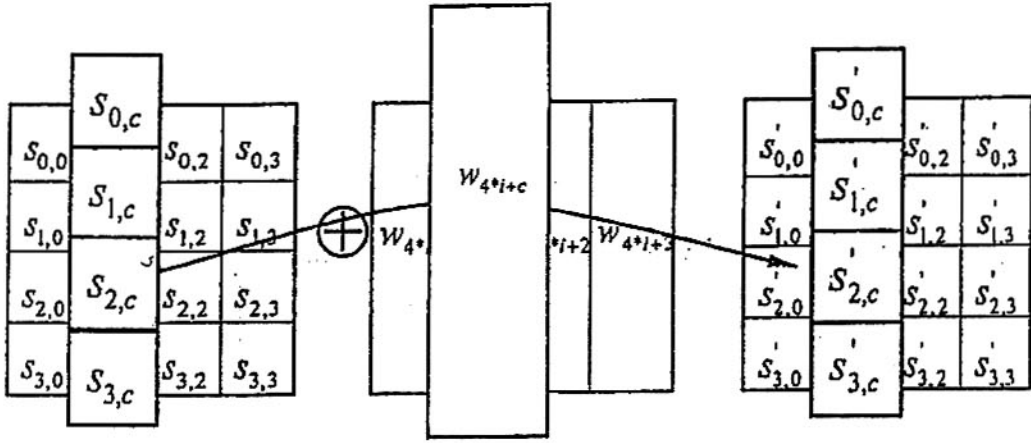
5.2.4.8 Phép biến đổi AddRoundKey ()

Trong phép biến đổi AddRoundKey (), khóa vòng được cộng vào Trạng thái bằng phép cộng bit đơn giản XOR. Mỗi khóa vòng gồm bốn từ (128 bit) lấy từ lược đồ khóa (được mô tả tại 5.2.5). Bốn từ này được cộng vào cột Trạng thái như sau:

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{(4+i+c)}] \quad \text{với } 0 \leq c < 4$$

Ở đây, $0 \leq c < 4$ và $w_{(4+i+c)}$ là các từ của lược đồ khóa thứ c của khóa vòng thứ i $W_i = [w_{(4+i)}, w_{(4+i+1)}, w_{(4+i+2)}, w_{(4+i+3)}]$ và i là giá trị thuộc khoảng $0 \leq i \leq Nr$. Trong phép mã hóa, phép cộng khóa vòng ban đầu xảy ra khi $i = 0$, trước ứng dụng thứ nhất của hàm vòng. Việc áp dụng phép biến đổi AddRoundKey () cho Nr vòng của phép mã hóa xảy ra khi $1 \leq i \leq Nr$.

Hoạt động của phép biến đổi $\text{AddRoundKey}()$, được minh họa trên Hình 14. Địa chỉ byte trong các từ của lược đồ khóa được mô tả trong 5.2.5



Hình 14 – $\text{AddRoundKey}()$ cộng bit XOR từng cột của Trạng thái với mỗi từ lấy từ lược đồ khóa

5.2.5 Lược đồ khóa AES

Thuật toán AES nhận khóa mật mã K và thực hiện thủ tục mở rộng khóa để tạo ra lược đồ khóa. Việc mở rộng khóa tạo ra tổng cộng $4(Nr + 1)$ từ. Thuật toán đòi hỏi tập khởi đầu gồm 4 từ, và mỗi vòng Nr vòng đòi hỏi 4 từ dữ liệu khóa. Lược đồ khóa nhận được là một mảng tuyến tính gồm các từ 4-byte, kí hiệu là w_j , với j nằm trong khoảng $0 \leq j < 4(Nr + 1)$.

Việc mở rộng khóa đầy đủ cho AES-128 và AES-192 có thể được mô tả như sau:

(1) $[w_0, w_1, w_2, w_3] = K$ và $Nk = 4$ với AES-128

$[w_0, w_1, w_2, w_3, w_4, w_5] = K$ và $Nk = 6$ với AES-192

(2) for $j = Nk$ đến $4(Nr + 1) - 1$:

if $(j \bmod Nk = 0)$ thì:

$$w_j = w_{j-Nk} \oplus \text{SubBytes} * (\text{ShiftColumn}(w_{j-1})) \oplus R^{i/Nk} c$$

else:

$$w_j = w_{j-Nk} \oplus w_{j-1}$$

(3) $W_i = [w_{(4*i)}, w_{(4*i+1)}, w_{(4*i+2)}, w_{(4*i+3)}]$ for $0 \leq i \leq Nr$

Trong phép toán trên, vòng khóa W_0 đầu tiên được sử dụng trong phép cộng khóa khởi đầu, trực tiếp kết hợp với 4 từ của khóa mật K . Các cột 32-bit w_j của các khóa vòng còn lại được tạo ra theo phương pháp truy hồi. Phép biến đổi SubBytes^* thay thế bytes của từng cột riêng lẻ, cũng giống như phép biến đổi $\text{subBytes}()$ được mô tả tại 5.2.4.2. Phép biến đổi $\text{ShiftColumn}()$ là dịch vòng lên phía trên một byte. Các hằng R_c^j là những cột 4-byte cố định, được xác định dưới dạng $(02^{j-1}, 00, 00, 00)^T$ với $\{02\}$ biểu diễn phần tử x trong trường $GF(2^8)$ (sử dụng cùng đa thức bất khả qui $x^8 + x^4 + x^3 + x + 1$).

Phép mở rộng khóa đầy đủ cho AES-256 có thể được mô tả như sau:

$$(1) [w_0, w_1, w_2, w_3] = K_0, [w_4, w_5, w_6, w_7] = K_1, Nk = 8$$

(2) for $j = Nk$ to $4(Nr + 1) - 1$:

if $(j \bmod Nk = 0)$ else

$$w_j = w_{j-Nk} \oplus \text{SubBytes} * (\text{ShiftColumn}(w_{j-1})) \oplus R^{i/Nk} C$$

Else if $(j \bmod Nk = 4)$ then:

$$w_j = w_{j-Nk} \oplus \text{SubBytes} * (w_{j-1})$$

else:

$$w_j = w_{j-Nk} \oplus w_{j-1}$$

$$(3) W_i = [w_{(4+i)}, w_{(4+i+1)}, w_{(4+i+2)}, w_{(4+i+3)}] \text{ for } 0 \leq i \leq Nr$$

Trong phép toán trên K_0 và K_1 biểu thị hai nửa của khóa mật mã K 256-bit.

5.3 Camellia

5.3.1 Thuật toán Camellia

Thuật toán Camellia là mã đối xứng có thể xử lý các khối 128 bit, sử dụng khóa mật mã độ dài 128, 192 và 256 bit. Giao diện cũng tương tự như của thuật toán AES.

5.3.2 Phép mã hóa Camellia

5.3.2.1 Khóa 128-bit

Quá trình mã hóa với khóa 128 bit làm việc trên 18 vòng, được chỉ ra trên Hình 15. Phép biến đổi khối 128-bit P vào khối 128-bit C được xác định như sau (L và R là các biến với độ dài 64-bit, $k_{\mathcal{W}}$, k và k' là các khóa vòng độ dài 64-bit):

$$(1) L_0 \| R_0 = P \oplus (k_{\mathcal{W}_1} \| k_{\mathcal{W}_2})$$

(2) for $i = 1$ to 18

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

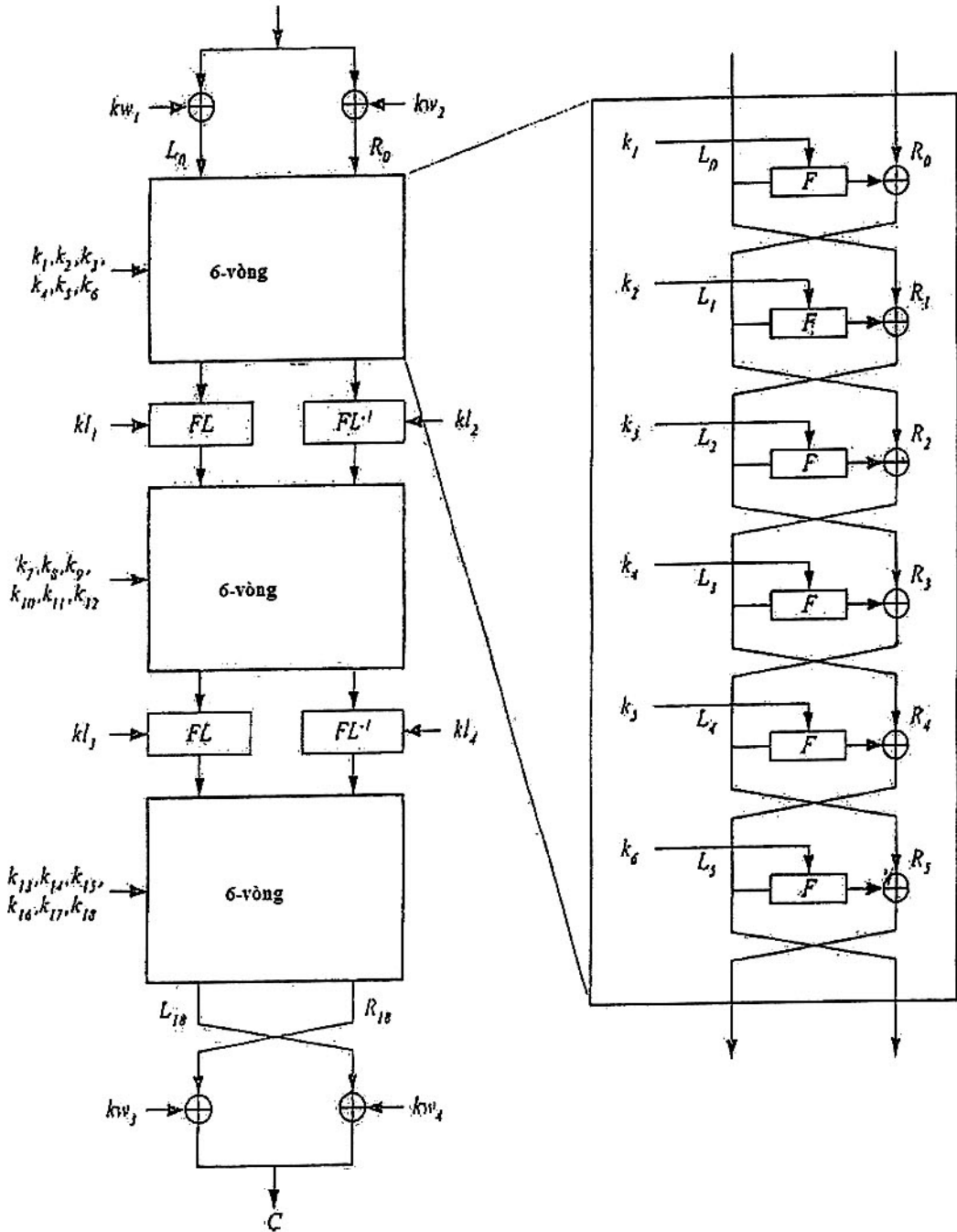
$$R_i = L_{i-1}$$

if $(i = 6 \text{ or } 12)$ else:

$$L_i = FL(L_i, k_{l_{i/3-1}})$$

$$R_i = FL^{-1}(R_i, k_{l_{i/3}})$$

$$(3) C = (R_{18} \oplus k_{\mathcal{W}_3}) \| (L_{18} \oplus k_{\mathcal{W}_4})$$



Hình 15 – Thủ tục mã hóa của Camellia cho khóa 128 bit

5.3.2.2 Khóa 192-bit và 256-bit

Quá trình mã hóa với khóa 192-bit hoặc khóa 256-bit làm việc trên 24 vòng, được chỉ ra trên Hình 16. Phép biến đổi khối 128 bit P vào khối 128-bit C được định nghĩa như sau (L và R là các biến độ dài 64 bit, kw , k và kl là các khóa vòng 64 bit):

$$(1) L_0 \| R_0 = P \oplus (kw_1 \| kw_2)$$

(2) for $i = 1$ to 24:

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

$$R_i = L_{i-1}$$

if ($i = 6$ or 12 or 18) then

$$L_i = FL(L_i, kl_{i/3-1})$$

$$R_i = FL^{-1}(R_i, kl_{i/3})$$

$$(3) C = (R_{24} \oplus kw_3) \| (L_{24} \oplus kw_4)$$

5.3.3 Phép giải mã Camellia

5.3.3.1 Khóa 128-bit

Quá trình giải mã cho khóa 128 bit được chỉ ra trên Hình 17, và cũng giống như trong phép giải mã, chỉ khác là vị trí và thứ tự các khóa vòng được đảo lại.

Phép giải mã được xác định như sau:

$$(1) R_{18} \| L_{18} = C \oplus (kw_3 \| kw_4)$$

(2) for $i = 18$ down to 1

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

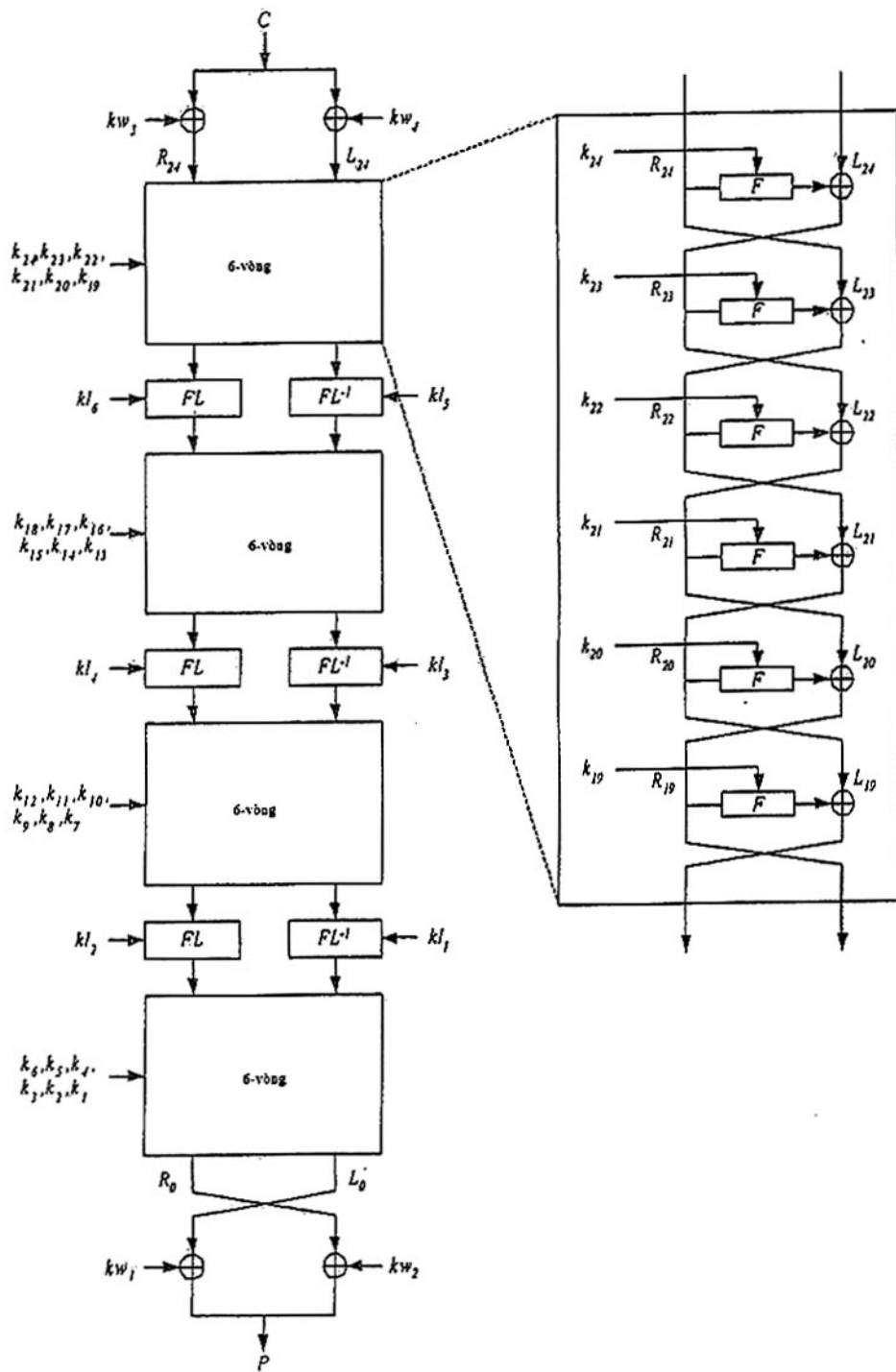
$$L_{i-1} = R_i$$

if ($i = 13$ or 7) then

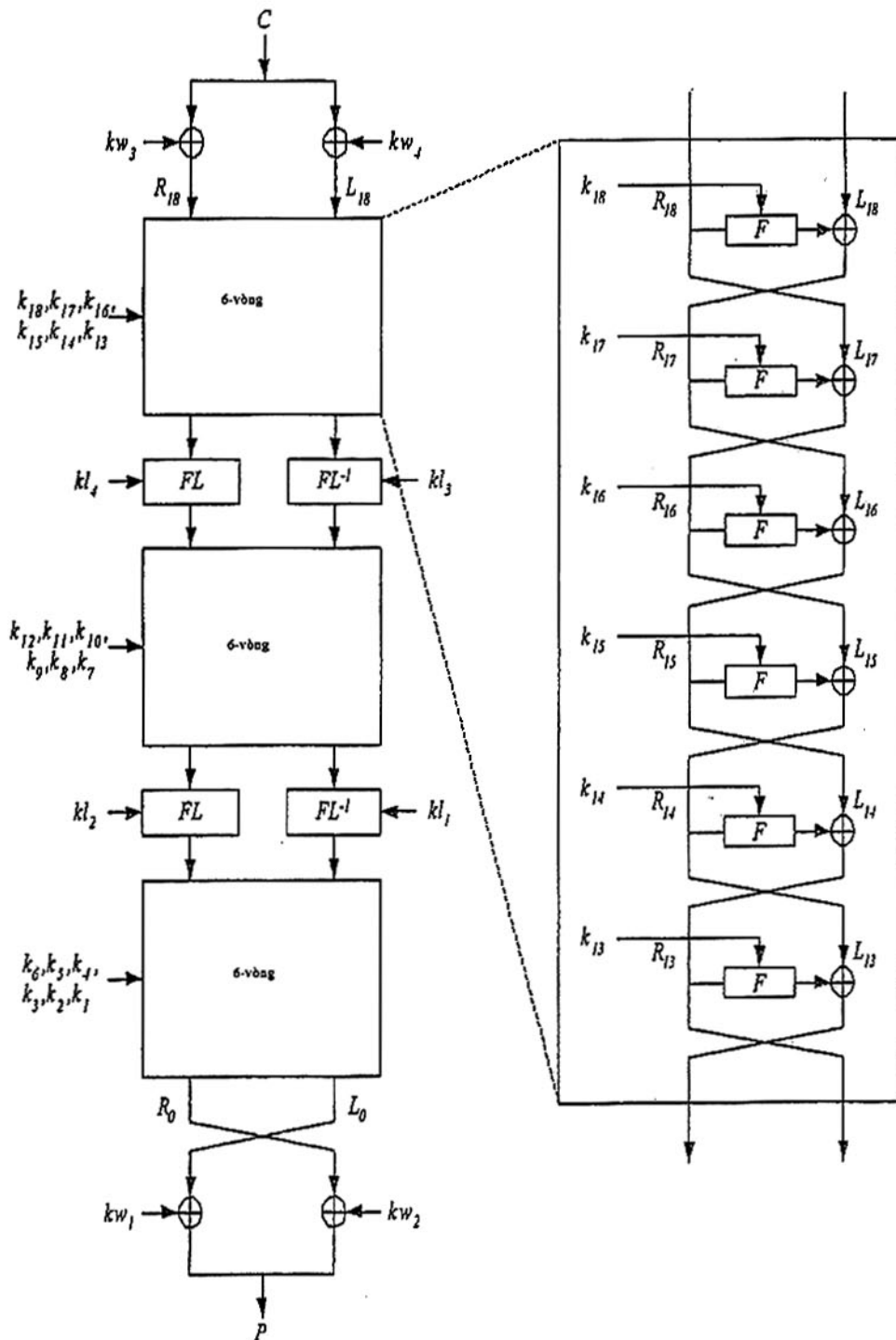
$$R_{i-1} = FL(R_{i-1}, kl_{(i-1)/3})$$

$$L_{i-1} = FL^{-1}(L_{i-1}, kl_{(i-1)/3-1})$$

$$(3) P = (L_0 \oplus kw_1) \| (R_0 \oplus kw_2)$$



Hình 16 – Thủ tục mã hóa Camellia cho khóa 192 và 256 bit



Hình 17 – Thủ tục giải mã Camellia cho khóa 128 bit

5.3.3.2 Khóa 192 và 256 bit

Quá trình giải mã cho khóa 192 bit và 256 bit được chỉ ra trên Hình 18, và là như nhau trong phép mã hóa, chỉ khác là vị trí và thứ tự các khóa vòng được đảo lại.

Phép giải mã được xác định như sau:

$$(1) R_{24} = L_{24} = C \oplus (kw_3 || kw_4)$$

(2) for $i = 24$ down to 1

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

$$L_{i-1} = R_i$$

if ($i = 19$ or 13 or 7) then

$$R_{i-1} = FL(R_{i-1}, kl_{(i-1)/3})$$

$$L_{i-1} = FL^{-1}(L_{i-1}, kl_{(i-1)/3-1})$$

$$(3) P = (L_0 \oplus kw_1) || (R_0 \oplus kw_2)$$

5.3.4 Các hàm Camellia

5.3.4.1 Các hàm xác định

Thuật toán Camellia sử dụng một số hàm, cụ thể là các hàm F, FL, FL⁻¹ và các S-box.

5.3.4.2 F-hàm

F-hàm được chỉ ra trên Hình 19. F-hàm bao gồm phép toán cộng bit XOR, tiếp đó áp dụng tám S-box song song kích thước 8x8 bit, tiếp theo tầng khuếch tán (P-hàm). x_j, y_j, z_j, z'_j là các biến, mỗi biến 8 bit; các biến L_i, k_i, L'_i là các biến 64 bit. Đầu vào 64 bit L_i trước hết được cộng XOR với khóa vòng 64-bit k_i , sau đó được chia thành 8 đoạn 8-bit y_j , như sau:

$$y_1 || y_2 || y_3 || y_4 || y_5 || y_6 || y_7 || y_8 = L_i \oplus k_i$$

ở đây,

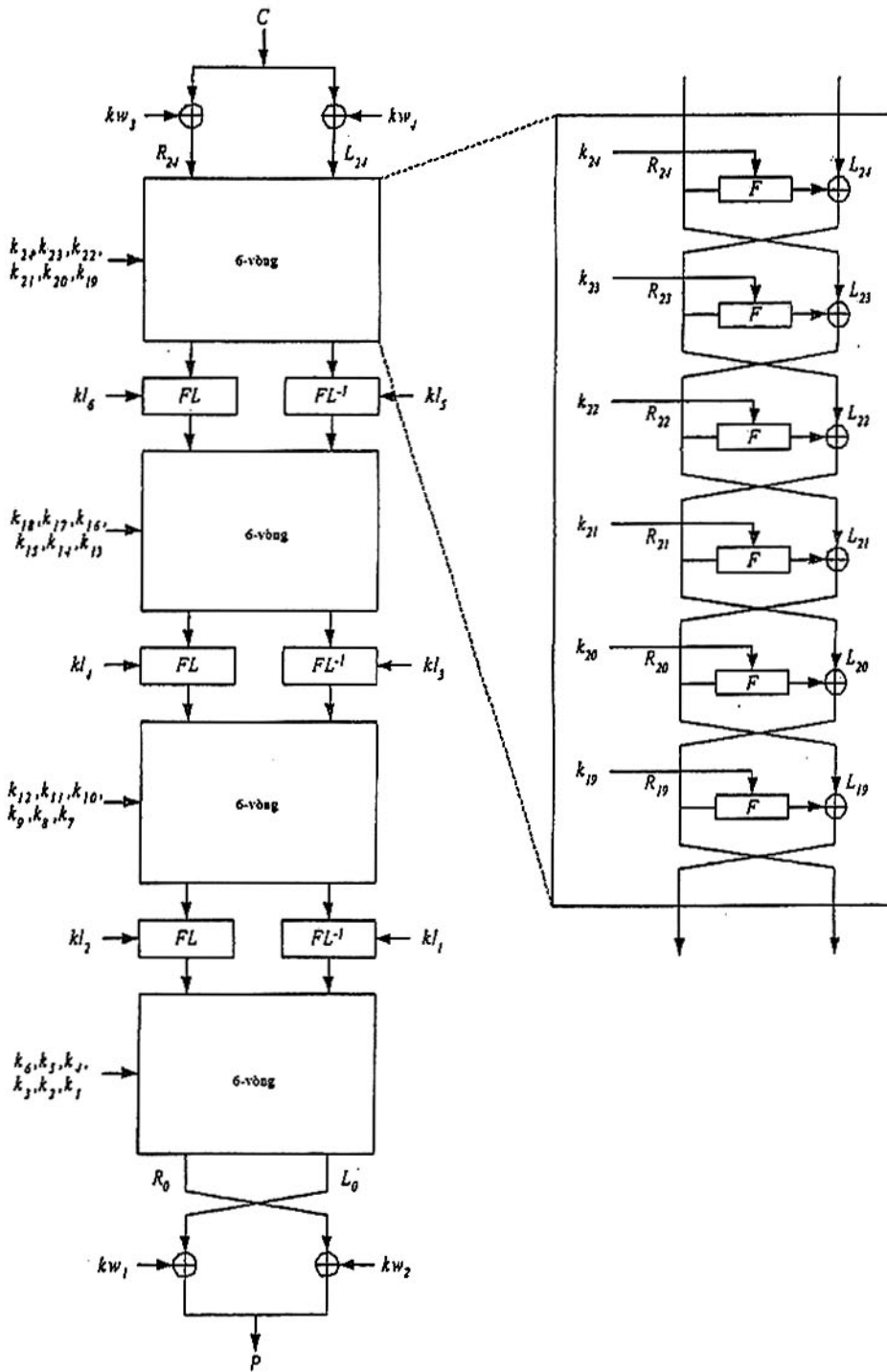
$$L_i = x_1 || x_2 || x_3 || x_4 || x_5 || x_6 || x_7 || x_8$$

Mỗi y_j sau đó được đi qua S-box kích thước 8x8 bit s_t để đưa ra 8 phân đoạn độ dài 8 bit z_j , ở đây

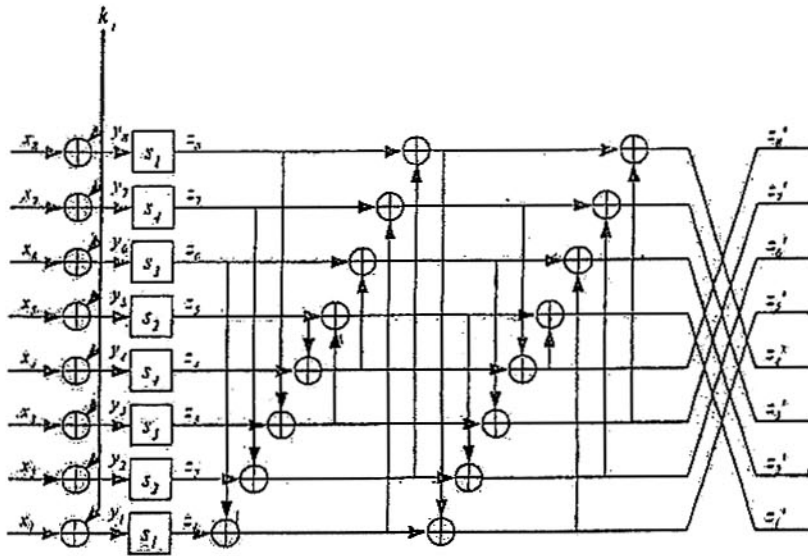
$$z_1 = s_1[y_1], z_2 = s_2[y_2], z_3 = s_3[y_3], z_4 = s_4[y_4], z_5 = s_2[y_5], z_6 = s_3[y_6], z_7 = s_4[y_7], z_8 = s_1[y_8].$$

8 phân đoạn 8-bit z_j được tác động bởi P-hàm, là lớp khuếch tán cho ra 8 phân đoạn 8-bit z'_j , trong đó

$$\begin{aligned} z'_1 &= z_1 \oplus z_3 \oplus z_4 \oplus z_6 \oplus z_7 \oplus z_8 & z'_5 &= z_1 \oplus z_2 \oplus z_6 \oplus z_7 \oplus z_8 \\ z'_2 &= z_1 \oplus z_2 \oplus z_4 \oplus z_5 \oplus z_7 \oplus z_8 & z'_6 &= z_2 \oplus z_3 \oplus z_5 \oplus z_7 \oplus z_8 \\ z'_3 &= z_1 \oplus z_2 \oplus z_3 \oplus z_5 \oplus z_6 \oplus z_8 & z'_7 &= z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_8 \\ z'_4 &= z_2 \oplus z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7 & z'_8 &= z_1 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7 \end{aligned}$$



Hình 18 – Thủ tục giải mã Camella cho khóa 192 và 256 bit



Hình 19 – F-hàm

P-hàm có thể được biểu diễn cách khác, dưới dạng véc tơ ma trận như sau:

$$\begin{pmatrix} z_8 \\ z_7 \\ \cdot \\ \cdot \\ \cdot \\ z_1 \end{pmatrix} \Rightarrow \begin{pmatrix} z'_8 \\ z'_7 \\ \cdot \\ \cdot \\ \cdot \\ z'_1 \end{pmatrix} = P \begin{pmatrix} z_8 \\ z_7 \\ \cdot \\ \cdot \\ \cdot \\ z_1 \end{pmatrix}$$

ở đây:

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

Sau đó đầu ra 64-bit của F-hàm L'_i được thiết lập bằng cách ghép các biến 8-bit z'_j :

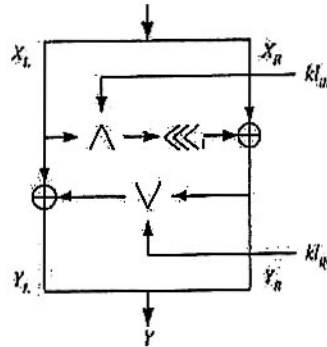
$$L'_i = z'_1 \| z'_2 \| z'_3 \| z'_4 \| z'_5 \| z'_6 \| z'_7 \| z'_8$$

5.3.4.3 FL-hàm

FL-hàm được chỉ ra trên Hình 20. FL-hàm được xác định như sau (X và Y là dữ liệu 64-bit, k_l là khóa vòng 64 bit, $X_L, X_R, Y_L, Y_R, kl_{iL}, kl_{iR}$ có độ dài 32-bit)

(1) $X_L \| X_R = X, kl_{iL} \| kl_{iR} = kl_i$

- (2) $Y_R = ((X_L \wedge kl_{iL}) \lll_1) \oplus X_R, Y_L = (Y_R \vee kl_{iR}) \oplus X_L$
 (3) $Y = Y_L \parallel Y_R$

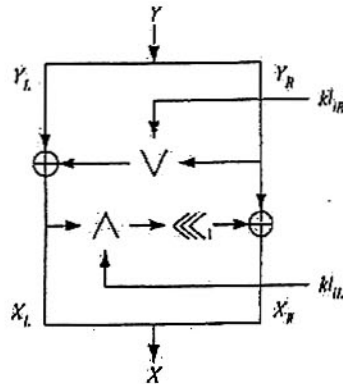


Hình 20 – Hàm FL

5.3.4.4 FL⁻¹-hàm

FL⁻¹-hàm được chỉ ra trên Hình 21. FL⁻¹-hàm được xác định như sau (X và Y là dữ liệu 64-bit, kl là khóa vòng, $X_L, X_R, Y_L, Y_R, kl_{iL}, kl_{iR}$ có độ dài 32-bit)

- (1) $Y_L \parallel Y_R = Y, kl_{iL} \parallel kl_{iR} = kl_i$
 (2) $X_L = (Y_R \vee kl_{iR}) \oplus Y_L, X_R = ((X_L \wedge kl_{iR}) \lll_1) \oplus Y_R$
 (3) $X = X_L \parallel X_R$

Hình 21 – FL⁻¹-hàm

5.3.4.5 Các S-box

5.3.4.5.1 Các S-box s_1 đến s_4 .

S-box s_1, s_2, s_3 và s_4 được đưa ra tại các Điều khoản con dưới đây. Các S-box này nhận đầu vào 8-bit và cho đầu ra 8-bit, có thể được mô tả dưới dạng đại số đơn giản. Dạng đại số của s_1 được cho tại Điều C.3.

5.3.4.5.2 S-box s_1

s_1 được đưa ra như sau:

$$s_1: y = h(g(f(c5 \oplus x))) \oplus 6e$$

ở đây các phép toán f , g và h nhận đầu vào 8-bit $a = a_1 || a_2 || a_3 || a_4 || a_5 || a_6 || a_7 || a_8$, và cho các giá trị đầu ra 8-bit $b = b_1 || b_2 || b_3 || b_4 || b_5 || b_6 || b_7 || b_8$, ở đây a_i và b_i là các giá trị 1-bit, f là hoán vị affine của đầu vào, g là phép nghịch đảo trên trường $GF(2^8)$ và h là phép biến đổi affine của đầu ra.

Ví dụ, đầu vào của s_1 là {53}, thì giá trị thay thế có thể được xác định bởi giao của dòng '5' và cột '3' trên Bảng 7, kết quả cho giá trị $s_1 = \{c_2\}$.

Bảng 7 – S-box s_1

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	70	82	2c	ec	b3	27	c0	e5	e4	85	57	35	ea	0c	ae	41
1	23	ef	6b	93	45	19	a5	21	ed	0e	4f	4e	1d	65	92	bd
0	86	b8	af	8f	7c	eb	1f	ce	3e	30	dc	5f	5e	c5	0b	1a
3	a6	e1	39	ca	d5	47	5d	3d	d9	01	5a	d6	51	56	6c	4d
4	8b	0d	9a	66	fb	cc	b0	2d	74	12	2b	20	f0	b1	84	99
5	df	4c	cb	c2	34	7e	76	05	6d	b7	a9	31	d1	17	04	d7
6	14	58	3a	61	de	1b	11	1c	32	0f	9c	16	53	18	f2	22
7	fe	44	cf	b2	c3	b5	7a	91	24	08	e8	a8	60	fc	69	50
8	aa	d0	a0	7d	a1	89	62	97	54	5b	1e	95	e0	ff	64	d2
9	10	c4	00	48	a3	f7	75	db	8a	03	e6	da	09	3f	dd	94
a	87	5c	83	02	cd	4a	90	33	73	67	f6	f3	9d	7f	bf	92
b	52	9b	d8	26	c8	37	c6	3b	81	96	6f	4b	13	be	63	2e
c	e9	79	a7	8c	9f	6e	be	8e	29	f5	f9	b6	2f	fd	b4	59
d	78	98	06	6a	e1	46	71	ba	d4	25	ab	42	88	a2	8d	fa
e	72	07	b9	55	f8	ee	ac	0a	36	49	2a	68	3c	38	f1	a4
f	40	28	d3	7b	bb	c9	43	c1	15	e3	ad	f4	77	c7	80	9e

5.3.4.5.3 S-box s_2

s_2 được chỉ ra trên Bảng 8 và có cấu trúc như sau:

$$s_2: y = s_1(x) \lll 1.$$

Bảng 8 – S-box s_2

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	e0	05	58	d9	67	4e	81	cb	c9	0b	ae	6a	d5	18	5d	82
1	46	df	d6	27	8a	32	4b	42	db	1c	9e	9c	3a	ca	25	7b
2	0d	71	5f	1f	f8	d7	3e	9d	7c	60	b9	be	be	8b	16	34
3	4d	c3	72	95	ab	8e	ba	7a	b3	02	b4	ad	a2	ac	d8	9a
4	17	1a	35	cc	f7	99	61	5a	e8	24	56	40	e1	63	09	33
5	bf	98	97	85	68	fc	ec	0a	da	6f	53	62	a3	2e	08	af
6	28	b0	74	c2	bd	36	22	38	64	1e	39	2c	a6	30	e5	44
7	fd	88	9f	65	87	6b	f4	23	48	10	d1	51	c0	f9	d2	a0
8	55	a1	41	fa	43	13	c4	2f	a8	b6	3c	2b	c1	ff	c8	a5
9	20	89	00	90	47	ef	ea	b7	15	06	cd	b5	12	1e	bb	29
a	0f	b8	07	04	9b	94	21	66	e6	ce	ed	e1	3b	fe	7f	c5
b	a4	37	b1	4c	91	6e	8d	76	03	2d	de	96	26	7d	c6	5c

c	d3	f2	4f	19	3f	dc	79	1d	52	eb	f3	6d	5e	fb	69	b2
d	f0	31	0c	d4	cf	8c	e2	75	a9	4a	57	84	11	45	1b	f5
e	e4	0e	73	aa	f1	dd	59	14	6c	92	54	d0	78	70	e3	49
f	80	50	a1	f6	77	93	86	83	2a	c1	5b	e9	ee	8f	01	3d

5.3.4.5.4 S-box s_3

s_3 được chỉ ra trên Bảng 9 và có cấu trúc như sau:

$$s_3: y = s_1(x) \lll\ll_7$$

Bảng 9 – S-box s_3

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	38	41	16	76	d9	93	60	f2	72	c2	ab	9a	75	06	57	a0
1	91	f7	b5	c9	a2	8c	d2	90	f6	07	a1	27	8e	b2	49	de
2	43	5c	d1	c1	3e	f5	8f	67	1f	18	6e	af	2f	e2	85	0d
3	53	f0	9c	65	ea	a3	ae	9e	ec	80	2d	6b	a8	2b	36	a6
4	c5	86	4	33	fd	66	58	96	3a	09	95	10	78	d8	42	cc
5	ef	26	e5	61	1a	3f	3b	82	b6	db	d4	98	e8	8b	02	eb
6	0a	2c	1d	b0	6f	8d	88	0e	19	87	4e	0b	a9	0c	79	11
7	7f	22	e1	59	e1	da	3d	c8	12	04	74	54	30	7e	b4	28
8	55	68	50	be	d0	c4	31	cb	2a	ad	0f	ca	70	ff	32	69
9	08	62	00	24	d1	fb	ba	ed	45	81	73	6d	84	9f	ee	4a
a	c3	2e	c1	01	e6	25	48	99	b9	b3	7b	f9	ce	bf	df	71
b	29	cd	6c	13	64	9b	63	9d	c0	4b	b1	a5	89	5f	b1	17
c	f4	be	d3	46	cf	37	5e	47	94	fa	fc	5b	97	fe	5a	ac
d	3c	4c	03	35	f3	23	b8	5d	6a	92	d5	21	44	51	c6	1d
e	39	83	dc	aa	7c	77	56	05	1b	a4	15	34	1e	1c	f8	52
f	20	14	e9	bd	dd	e4	a1	e0	8a	f1	d6	7a	bb	e3	40	4f

5.3.4.5.5 S-box s_4

s_4 được chỉ ra trên Bảng 10 và có cấu trúc như sau:

$$s_4: y = s_1(x \lll\ll_1)$$

Bảng 10 – S-box s_4

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	70	2c	b3	c0	e4	57	ea	ae	23	6b	45	af	ed	4f	1d	92
1	86	af	7c	1f	3e	dc	5e	0b	a6	39	d5	5d	d9	5a	51	6c
2	8b	9a	fb	b0	74	2b	f0	84	df	cb	34	76	6d	a9	d1	04
3	14	3a	de	11	32	9c	53	f2	fe	cf	c3	7a	24	e8	60	69
4	aa	a0	a1	62	54	1e	e0	64	10	00	a3	75	8a	e6	09	dd
5	87	83	cd	90	73	f6	9d	bf	52	d8	c8	c6	81	6f	13	63
6	e9	a7	9f	be	29	f9	2f	b4	78	06	e1	71	d4	ab	88	8d
7	72	b9	f8	ac	36	2a	3c	f1	40	d3	bb	43	15	ad	77	80
8	82	ec	27	e5	85	35	0c	41	ef	93	19	21	0e	4e	65	bd
9	b8	8f	eb	ce	30	5f	c5	1a	e1	ca	47	3d	01	d6	56	4d
a	0d	66	cc	2d	12	20	b1	99	4c	c2	7e	05	b7	31	17	d7
b	58	61	1b	1c	0f	16	18	22	44	b2	b5	91	08	a8	fc	50
c	d0	7d	89	97	5b	95	ff	d2	c4	48	f7	db	03	da	3f	94
d	5c	02	4a	33	67	f3	7f	e2	9b	26	37	3b	96	4b	be	2e
e	79	8c	6e	8e	f5	b6	fd	59	98	6a	46	ba	25	42	a2	fa
f	07	55	ee	0a	49	68	38	a4	28	7b	c9	c1	e3	f4	c7	9e

5.3.5 Lược đồ tạo khóa Camellia

Lược đồ tạo khóa được chỉ ra trên Hình 22 và các Bảng 12 và 13. Với phiên bản 128 bit khóa, khóa K là khóa 128-bit K_L , khóa 128-bit K_R được đặt tất cả các bit bằng 0, như vậy:

$$K = K_L \| K_R = 0$$

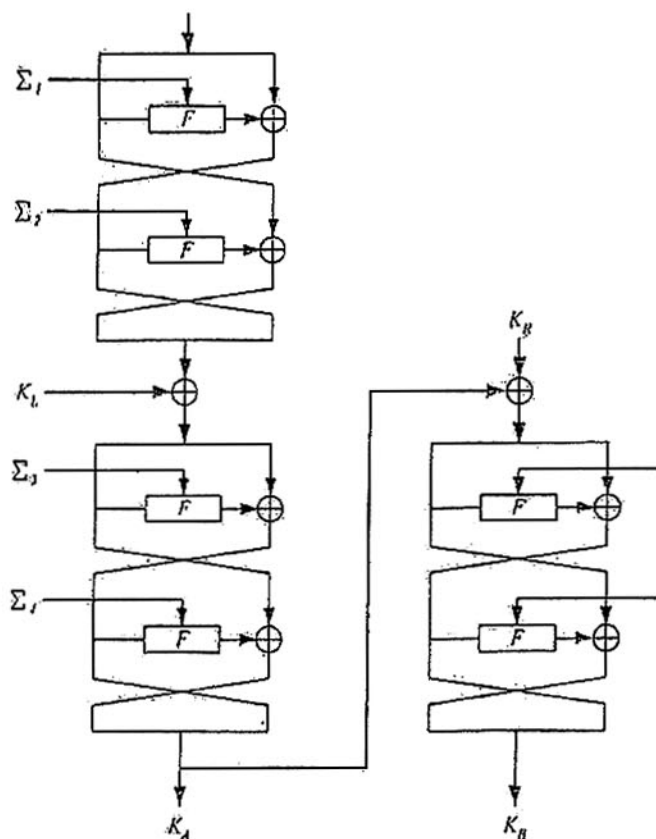
Với phiên bản 192 bit khóa, khóa K là khóa 128-bit K_L và 64 bit trái nhất K_R, K_{RL} , với 64 bit phải nhất K_R, K_{RR} được đặt phủ định theo từng bit của 64 bit trái nhất của K_R và K_{RL} . Như vậy,

$$K = K_L \| K_{RL}, K_{RR} = \overline{K_{RL}}, K_R = K_{RL} \| K_{RR}$$

Với phiên bản 256 bit, khóa K là khóa 128-bit K_L và khóa 128-bit K_R . Như vậy,

$$K = K_L \| K_R$$

Lược đồ tạo khóa sử dụng F-hàm của mô-đun mã hóa, và là giống nhau cho cả phép mã hóa và giải mã. Khóa K được mã bằng các phương tiện của F-hàm, sử dụng các hằng của lược đồ tạo khóa, ở đây các hằng Σ_i được xác định như những giá trị liên tục từ biểu diễn trong hệ Hexa của căn bậc hai số nguyên tố thứ i . Tiếp đó các khóa vòng được tạo, một phần từ những giá trị được dịch vòng của khóa K (ở đây K bằng $K_L, K_L \| K_{RL}$ or $K_L \| K_R$ tương ứng với phiên bản khóa 128-bit, 192-bit hoặc 256-bit) và phần còn lại từ các giá trị được dịch vòng của các khóa "được mã hóa" K_A và K_B (ở đây K_A, K_B có độ dài 128 bit).



Hình 22 – Phân chính của lược đồ khóa

Đối với phiên bản khóa 128-bit, đầu ra của phần chính của lược đồ tạo khóa là khóa con 128-bit K_A , tương ứng bỏ qua phía phải của hình 22 và khóa K_B không được tạo ra hoặc không được sử dụng. Đối với phiên bản khóa 192-bit và 256-bit, đầu ra của phần chính của lược đồ tạo khóa là khóa con 128 bit K_A và với khóa con 128-bit K_B . Lược đồ tạo khóa bao gồm hai hoặc ba phép toán 2-vòng cho các phiên bản khóa 128 bit hoặc 192/256 bit tương ứng. Mỗi phép toán 2 vòng được "khóa hóa" bằng một cặp hằng Σ_i .

Đầu vào 128-bit của phép toán 2-vòng thứ nhất nằm bên trái của Hình 22 là $K_L \oplus K_R$ và phép toán này được "khóa hóa" bởi hai hằng 64-bit Σ_1 và Σ_2 . Tiếp đó đầu ra 128-bit từ phép toán 2-vòng thứ nhất được cộng bit XOR với K_L trước khi là đầu vào của phép toán 2-vòng thứ hai ở bên trái Hình 22. Phép toán 2-vòng thứ hai này được "khóa hóa" bởi các hằng 64 bit là Σ_3 và Σ_4 . Đầu ra 128-bit của phép toán 2-vòng thứ hai là K_A . Với các phiên bản khóa 192-bit hoặc 256-bit thì sau đó K_A lại được XOR với khóa con 128-bit K_B trước khi kết quả thu được làm đầu vào của phép toán 2-vòng thứ ba được chỉ ra ở phía phải của Hình 22. Phép toán 2-vòng thứ ba này được "khóa hóa" bởi hai hằng 64 bit là Σ_5 và Σ_6 . Đầu ra 128 bit của phép toán 2-vòng thứ ba này là K_B .

Phép toán lập lược đồ tạo khóa đầy đủ được mô tả như sau (K_a , K_A và K_B có độ dài 128 bit):

- (1) $K_a = 2RoundFeistel(K_L \oplus K_R, \Sigma_1, \Sigma_2)$
- (2) $K_A = 2RoundFeistel(K_a \oplus K_L, \Sigma_3, \Sigma_4)$
- (3) $K_B = 2RoundFeistel(K_A \oplus K_R, \Sigma_5, \Sigma_6)$ (chỉ dùng cho khóa 192/256 bit)

ở đây đầu vào 128-bit cho 2RoundFeistel là tách thành hai phần 64-bit $L_0 \parallel R_0$, đầu ra 128-bit từ 2RoundFeistel cũng được tách thành hai phần 64 bit $L_2 \parallel R_2$ và cả hai đầu vào "khóa vòng" 64-bit của 2RoundFeistel là Σ_i và Σ_{i+1} .

2RoundFeistel được mô tả như sau:

- (1) với $j = 0, 1$:

$$L_{j+1} = F(L_j, \Sigma_{i+j}) \oplus R_j$$

$$R_{j+1} = L_j$$

Các hằng của lược đồ khóa 64-bit được xác định trên Bảng 11

Bảng 11 – Các hằng trong lược đồ khóa

	Hằng
Σ_1	a09e667f3bcc908b
Σ_2	b67ae8584caa73b2
Σ_3	c6ef372fe94f82be
Σ_4	54ff53a5f1d36f1c
Σ_5	10e527fade682d1d
Σ_6	b05688c2b3e6c1fd

Cuối cùng, các khóa vòng 64-bit, k , kw , và kl được dẫn xuất từ các khóa con 128-bit, K_L , K_R , K_A và K_B . Bảng 12 dành cho phiên bản 128-bit và Bảng 13-phiên bản 192 hoặc 256-bit

Bảng 12 – Khóa vòng cho khóa bí mật 128 bit

Hàm	Khóa vòng	Giá trị
	k_{W1}	$(K_L \lll 0)_L$
	k_{W1}	$(K_L \lll 0)_R$
F (vòng 1)	k_1	$(K_A \lll 0)_L$
F (vòng 2)	k_2	$(K_A \lll 0)_R$
F (vòng 3)	k_3	$(K_L \lll 15)_L$
F (vòng 4)	k_4	$(K_L \lll 15)_R$
F (vòng 5)	k_5	$(K_A \lll 15)_L$
F (vòng 6)	k_6	$(K_A \lll 15)_R$
FL	k_{d1}	$(K_A \lll 30)_L$
FL^{-1}	k_{d2}	$(K_A \lll 30)_R$
F (vòng 7)	k_7	$(K_L \lll 45)_L$
F (vòng 8)	k_8	$(K_L \lll 45)_R$
F (vòng 9)	k_9	$(K_A \lll 45)_L$
F (vòng 10)	k_{10}	$(K_L \lll 60)_R$
F (vòng 11)	k_{11}	$(K_A \lll 60)_L$
F (vòng 12)	k_{12}	$(K_A \lll 60)_R$
FL	K_{l3}	$(K_L \lll 77)_L$
FL^{-1}	K_{l4}	$(K_L \lll 77)_R$
F (vòng 13)	K_{13}	$(K_L \lll 90)_L$
F (vòng 14)	K_{14}	$(K_L \lll 90)_R$
F (vòng 15)	K_{15}	$(K_A \lll 90)_L$
F (vòng 16)	K_{16}	$(K_A \lll 90)_R$
F (vòng 17)	k_{17}	$(K_L \lll 111)_L$
F (vòng 18)	k_{18}	$(K_L \lll 111)_R$
	k_{W3}	$(K_A \lll 111)_L$
	k_{W4}	$(K_A \lll 111)_R$

Bảng 13 – Khóa vòng cho khóa bí mật 192/256 bit

Hàm	Khóa vòng	Giá trị
	k_{W1}	$(K_L \lll 0)_L$
	k_{W1}	$(K_L \lll 0)_R$
F (vòng 1)	k_1	$(K_B \lll 0)_L$
F (vòng 2)	k_2	$(K_B \lll 0)_R$
F (vòng 3)	k_3	$(K_R \lll 15)_L$
F (vòng 4)	k_4	$(K_R \lll 15)_R$
F (vòng 5)	k_5	$(K_A \lll 15)_L$
F (vòng 6)	k_6	$(K_A \lll 15)_R$
FL	k_{d1}	$(K_R \lll 30)_L$
FL^{-1}	k_{d2}	$(K_R \lll 30)_R$
F (vòng 7)	k_7	$(K_B \lll 30)_L$
F (vòng 8)	k_8	$(K_B \lll 30)_R$
F (vòng 9)	k_9	$(K_L \lll 45)_L$
F (vòng 10)	k_{10}	$(K_L \lll 45)_R$
F (vòng 11)	k_{11}	$(K_A \lll 45)_L$
F (vòng 12)	k_{12}	$(K_A \lll 45)_R$
FL	K_{l3}	$(K_L \lll 60)_L$
FL^{-1}	K_{l4}	$(K_L \lll 60)_R$
F (vòng 13)	K_{13}	$(K_R \lll 60)_L$
F (vòng 14)	K_{14}	$(K_R \lll 60)_R$
F (vòng 15)	K_{15}	$(K_B \lll 60)_L$
F (vòng 16)	k_{16}	$(K_B \lll 60)_R$
F (vòng 17)	k_{17}	$(K_L \lll 77)_L$
F (vòng 18)	k_{18}	$(K_L \lll 77)_R$
FL	K_{l5}	$(K_A \lll 77)_L$
FL^{-1}	K_{l6}	$(K_A \lll 77)_R$
F (vòng 19)	K_{19}	$(K_R \lll 90)_L$
F (vòng 20)	K_{20}	$(K_R \lll 90)_R$
F (vòng 21)	K_{21}	$(K_A \lll 90)_L$
F (vòng 22)	K_{22}	$(K_A \lll 90)_R$
F (vòng 23)	K_{23}	$(K_L \lll 111)_L$
F (vòng 24)	K_{24}	$(K_L \lll 111)_R$
	k_{W3}	$(K_B \lll 111)_L$
	k_{W4}	$(K_B \lll 111)_R$

5.4 SEED

5.4.1 Thuật toán SEED

Thuật toán SEED là mã khối đối xứng có thể xử lý khối dữ liệu 128 bit, sử dụng khóa mật mã độ dài 128 bit.

5.4.2 Phép mã hóa SEED

Phép toán mã hóa được chỉ ra trên Hình 23. Phép biến đổi khối dữ liệu 128-bit P thành khối dữ liệu 128-bit C được xác định như sau (K là khóa):

$$(1) P = L_0 \| R_0$$

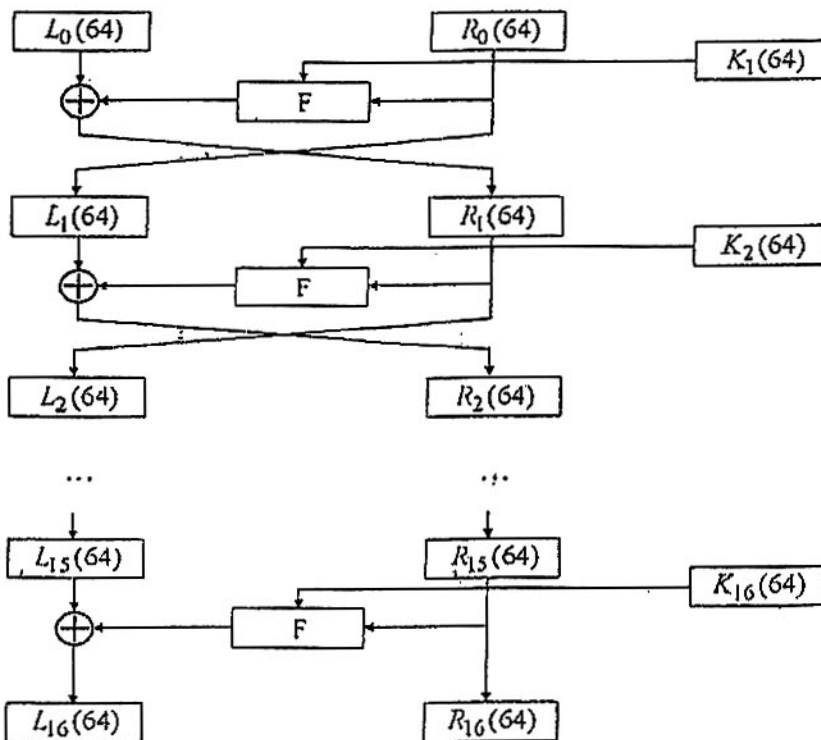
(2) với $i = 1$ đến 15:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

$$(3) L_{16} = L_{15} \oplus F(R_{15}, K_{16}); R_{16} = R_{15}$$

$$(4) C = L_{16} \| R_{16}$$



Hình 23 – Cấu trúc của SEED

5.4.3 Phép giải mã SEED

Giải mã cũng giống như phép mã hóa đã mô tả ở trên, chỉ khác là các vòng (và các khóa con) được sử dụng theo thứ tự ngược lại để tính (L_0, R_0) từ (R_{16}, L_{16})

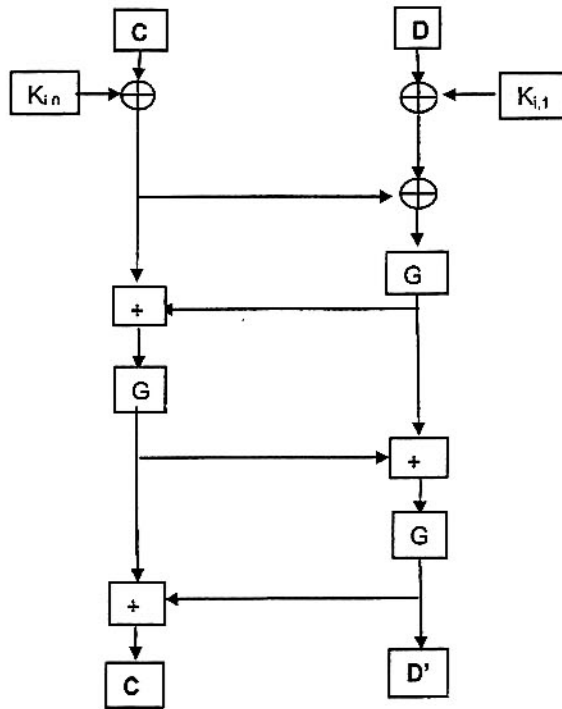
5.4.4 Các hàm của SEED

5.4.4.1 Hàm vòng F

Hàm vòng F chỉ ra trên Hình 24 và được định nghĩa như sau (C và D là dữ liệu, K là khóa)

$$C' = G[G\{(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})\} + (C \oplus k_{i,0}) + G\{(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})\} + G\{(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})\} + (C \oplus k_{i,0})]$$

$$D' = G[G\{(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})\} + (C \oplus k_{i,0}) + G\{(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})\}]$$



Hình 24 – Hàm vòng F

5.4.4.2 Hàm G

Hàm G chỉ ra trên Hình 25 có hai tầng: một tầng của hai S-box kích thước 8x8 và một tầng hoán vị khối của 16 khối con 8 bit.

Đầu ra a', b', c', d' của hàm G với bốn đầu vào 8-bit a, b, c, d được xác định như sau:

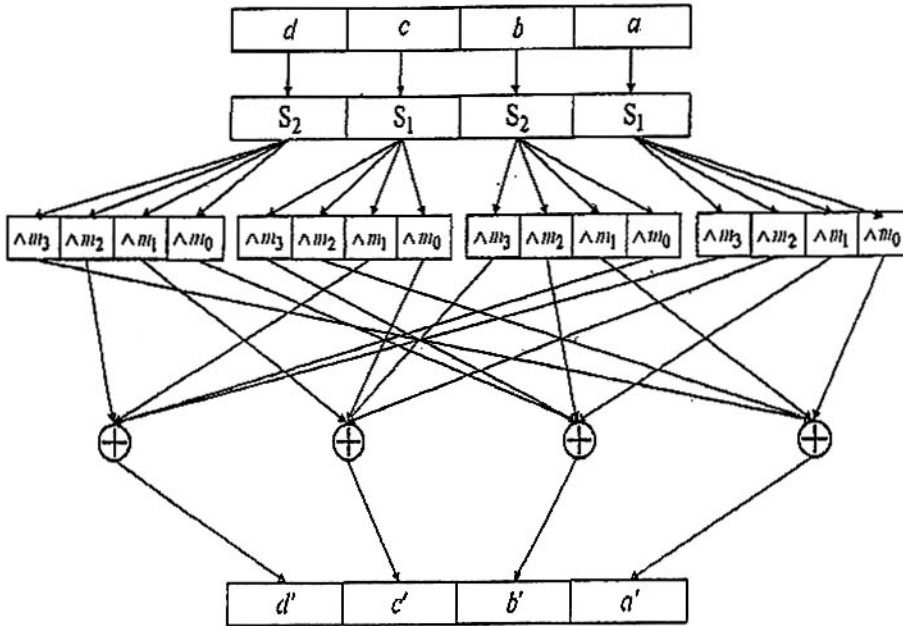
$$a' = (S_1(a)^{m_0}) \oplus (S_2(b)^{m_1}) \oplus (S_1(c)^{m_2}) \oplus (S_2(d)^{m_3})$$

$$b' = (S_1(a)^{m_1}) \oplus (S_2(b)^{m_2}) \oplus (S_1(c)^{m_3}) \oplus (S_2(d)^{m_0})$$

$$c' = (S_1(a) \wedge m_2) \oplus (S_2(b) \wedge m_3) \oplus (S_1(c) \wedge m_0) \oplus (S_2(d) \wedge m_1)$$

$$d' = (S_1(a) \wedge m_3) \oplus (S_2(b) \wedge m_0) \oplus (S_1(c) \wedge m_1) \oplus (S_2(d) \wedge m_2)$$

Ở đây, $m_0 = \{fc\}, m_1 = \{f3\}, m_2 = \{cf\}, m_3 = \{3f\}$



Hình 25 – Hàm G

5.4.4.3 S-box

Hai S-box S_1, S_2 là một phần của hàm G và được định nghĩa như sau:

$$S_i = Z_{2^{n_i}} \rightarrow Z_{2^{n_i}}, S_i(x) = A^{(i)} \cdot x^{n_i} \oplus b_i$$

Trong đó $n_1 = 247, n_2 = 251, b_1 = 169, b_2 = 56$ và

$$A^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}, A^{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Lưu ý rằng $A^{(i)} \bullet x^n \oplus b_i$ là phép biến đổi affine của x^n . Với x bất kì thuộc Z_{2^8} , x có thể biểu diễn dưới dạng véc tơ nhị phân $x = (x_7, \dots, x_0)$ (do đó $x = x_7 2^7 + x_6 2^6 + \dots + x_1 2 + x_0$). Chúng ta sử dụng đa thức nguyên thủy $p(x) = x^8 + x^6 + \dots + x + 1$ để biểu diễn lại x^n trong Z_{2^8} .

Các S-box S_1 và S_2 được mô tả tại Bảng 14 và 15.

Bảng 14 – S₁-hộp

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	a9	85	d6	d3	54	1d	ac	25	5d	43	18	1e	51	fc	ca	63
1	28	44	20	9d	e0	e2	c8	17	a5	8f	03	7b	bb	13	d2	ee
2	70	8c	3f	a8	32	dd	f6	74	ec	95	0b	57	5c	5b	bd	01
3	24	1c	73	98	10	cc	f2	d9	2c	e1	72	83	9b	d1	86	c9
4	60	50	a3	eb	0d	b6	9e	4	b7	5a	c6	78	a6	12	af	d5
5	61	c3	b4	41	52	7d	8d	08	1f	99	00	19	04	53	f7	e1
6	fd	76	2f	27	b0	8b	0e	ab	a2	6e	93	4d	69	1c	09	0a
7	bf	ef	f3	c5	87	14	fe	64	de	2e	4b	1a	06	21	6b	66
8	02	f	92	8a	0c	b3	1e	d0	7a	47	96	e5	26	80	ad	df
9	a1	30	37	ae	36	15	22	38	f4	a1	45	4c	81	e9	84	97
a	35	cb	ce	3c	71	11	c1	89	75	fb	da	f8	94	59	82	c4
b	ff	49	39	67	c0	cf	d1	b8	0f	8e	42	23	91	6c	db	a4
c	34	f1	48	c2	6f	3d	2d	40	be	3e	be	c1	aa	ba	4e	55
d	3b	dc	68	7f	9c	d8	4a	56	77	a0	ed	46	b5	2b	65	fa
e	e3	b9	b1	9f	5e	f9	e6	b2	31	ea	6d	5f	e4	f0	cd	88
f	16	3a	58	d4	62	29	07	33	e8	1b	05	79	90	6a	2a	9a

Bảng 15 – S₂-hộp

	0	1	2	3	4	5	6	7	8	9	a	b	c	a	e	f
0	38	e8	2d	a6	cf	de	b3	b8	af	60	55	c1	44	6f	6b	5b
1	c3	62	33	b5	29	a0	e2	a1	d3	91	11	06	1c	be	36	4b
2	ef	88	6c	a8	17	c4	16	f4	c2	45	e1	d6	3f	3d	8e	98
3	28	4e	f6	3e	a5	f9	0d	df	d8	2b	66	7a	27	2f	f1	72
4	42	d4	41	c0	73	67	ac	8b	f7	ad	80	1f	ca	2c	aa	34
5	d2	0b	ee	e9	5d	94	18	f8	57	ae	08	c5	13	cd	86	b9
6	ff	7d	c1	31	f5	8a	6a	b1	d1	20	d7	02	22	04	68	71
7	07	db	9d	99	61	be	e6	59	dd	51	90	dc	9a	a3	ab	d0
9	81	0f	47	1a	e3	ec	8d	bf	96	7b	5c	a2	a1	63	23	4d
9	c3	9e	9c	3a	0c	2e	ba	6e	9f	5a	f2	92	f3	49	78	cc
a	15	fb	70	75	1f	35	10	03	64	5d	c6	74	d5	b4	ea	09
b	76	19	fe	40	12	e0	bd	05	fa	01	f0	2a	5e	a9	56	43
c	85	14	89	9b	b0	e5	48	79	97	fc	1e	82	21	8c	1b	5f
d	77	54	b2	1d	25	4f	00	46	ed	58	52	eb	1e	da	c9	fd
e	30	95	65	3c	b6	e4	bb	7c	0e	50	39	26	32	84	69	93
f	37	e1	24	a4	cb	53	0a	87	d9	4c	83	8f	ce	3b	4c	b7

5.4.5 Lược đồ tạo khóa SEED

Lược đồ tạo khóa tạo các khóa con cho mỗi vòng. Lược đồ khóa sử dụng hàm G, các phép cộng/trừ và các phép dịch vòng (sang trái/ sang phải). Khóa đầu vào 128-bit được chia thành thành bốn khối 32-bit (a, b, c, d) và hai khóa con 32-bit của vòng thứ nhất $k_{1,0}$ và $k_{1,1}$ được tạo như sau:

$$k_{1,0} = G(a + c - KC_0), k_{1,1} = G(b + kC_0 - d)$$

Hai khóa con 32-bit của vòng thứ hai, $k_{2,0}$ và $k_{2,1}$ được tạo từ khóa đầu vào bằng cách dịch vòng 64 bit đầu tiên ($a||b$) sang phải 8-bit như sau :

$$a\|b \leftarrow (a\|b) \ggg_8$$

$$k_{2,0} = G(a + c - KC_1), k_{2,1} = G(b + kC_1 - d)$$

Hai khóa con của vòng thứ ba, $k_{3,0}$ và $k_{3,1}$ được tạo từ khóa đầu vào bằng cách dịch vòng 64-bit cuối cùng ($c\|d$) sang trái 8-bit như sau:

$$c\|d \leftarrow (c\|d) \lll_8$$

$$k_{3,0} = G(a + c - KC_2), k_{3,1} = G(b + kC_2 - d)$$

Các khóa con còn lại được tạo theo phương pháp lặp như sau:

(1) với $i=1$ đến 16

$$k_{i,0} = G(a + c - KC_{i-1})$$

$$k_{i,1} = G(b + KC_{i-1} - d)$$

$$i: \text{ chẵn} : a\|b = (a\|b) \ggg_8$$

$$i: \text{ lẻ} : c\|d = (c\|d) \lll_8$$

Ở đây các hằng KC_i (được mô tả ở Bảng 16) được tạo từ một phần của tỉ số vàng $\frac{\sqrt{5}-1}{2}$

Bảng 16 – Các hằng KC_i (ở dạng Hexa thập phân)

i	KC_i	i	KC_i
0	9e3779b9	8	3779b99e
1	3c6ef373	9	6ef3733c
2	78dde6e6	10	dde6e678
3	f1bbcdcc	11	bbcdccf1
4	e779b99	12	779b99e3
5	c6ef3733	13	ef3733c6
6	8dde6e67	14	de6e678d
7	1bbcdccf	15	bcdccf1b

Phụ lục A

(Quy định)

Mô tả DES

A.1 Mô đầu

Thuật toán DES là mã khối đối xứng có thể xử lý các khối dữ liệu 64 bit, sử dụng khóa bí mật độ dài 64 bit. Mỗi bit thứ tám của khóa mật mã thường được sử dụng để kiểm tra tính chẵn lẻ và được bỏ qua.

A.2 Phép mã hóa DES

Phép mã hóa được chỉ ra trên Hình A.1

Bản rõ 64-bit trước hết được biến đổi qua hoán vị ban đầu IP . Sau đó khối được chia thành hai nửa L_0 và R_0 , mỗi nửa gồm 32 bit. Tiếp đó thực hiện 16 vòng biến đổi giống nhau được gọi là hàm f , trong đó dữ liệu được kết hợp với khóa. Trong mỗi vòng, nửa phải là đầu vào của hàm f được khóa hóa, hàm này nhận đầu vào 32 bit và khóa con 48 bit K_i và cho đầu ra 32-bit. Đầu ra này tiếp đó được XOR với nửa trái để tạo ra nửa trái mới đã được biến đổi. Tại phần cuối của mỗi vòng, trừ vòng cuối cùng, hai nửa trái và phải đổi chỗ cho nhau để tạo ra L_i và R_i tương ứng. Sau khi thực hiện vòng cuối cùng, hai nửa trái và phải được ghép lại với nhau và khối 64-bit nhận được lại được biến đổi qua phép hoán vị cuối IP^{-1} là hoán vị nghịch đảo của hoán vị ban đầu IP . Đầu ra là bản mã 64-bit.

Phép mã hóa được xác định như sau (P và C là dữ liệu, K_i là khóa).

$$(1) IP(P) = L_0 \| R_0$$

$$(2) \text{ Với } i = 1, 2, \dots, 16:$$

$$L_i = R_{i-1}$$

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

$$(3) C = IP^{-1}(R_{16} \| L_{16})$$

A.3 Phép giải mã DES

Phép giải mã cũng giống như phép mã hóa. Sự khác nhau chỉ ở chỗ, các khóa con K_i được sử dụng theo thứ tự ngược lại.

A.4 Các hàm DES

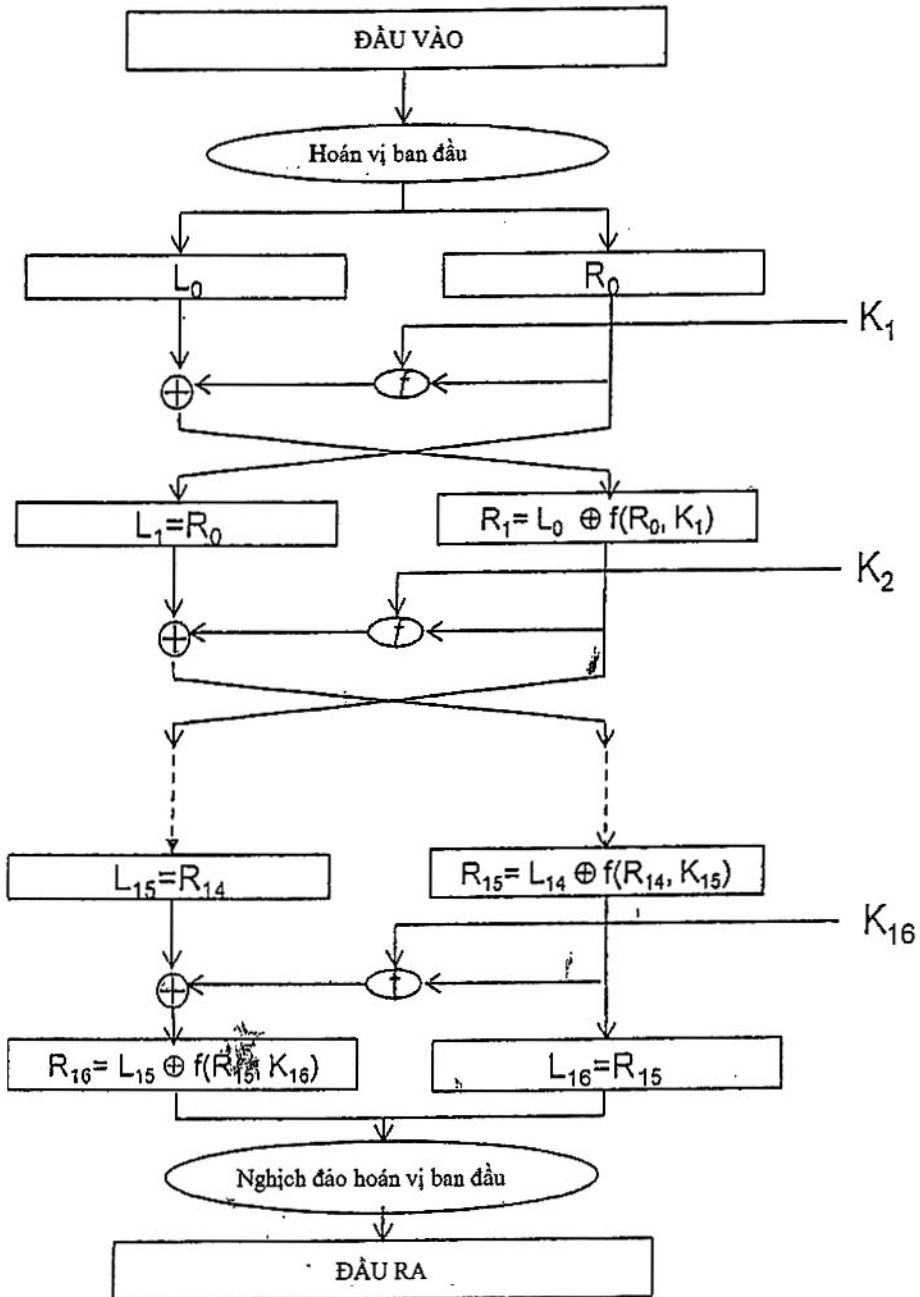
A.4.1 Phép hoán vị ban đầu IP

Phép hoán vị ban đầu IP được chỉ ra tại Bảng A.1. Hoán vị này nhận đầu vào 64-bit và cho đầu ra 64-bit, theo đó bit thứ nhất được hoán vị thành bit thứ 58, bit thứ hai thành bit thứ 50, v.v, bit cuối cùng thành bit thứ 7.

Bảng A.1 — Hoán vị khởi tạo

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



Hình A1 – Thủ tục mã hóa

A.4.2 Hoán vị nghịch đảo IP^{-1}

Hoán vị nghịch đảo IP^{-1} được chỉ ra trên Bảng A.2, nhận đầu vào 64-bit và cho đầu ra 64-bit. Đầu ra của Thuật toán nhận bit thứ 40 của khối đầu ra trước đó làm bit thứ nhất, bit thứ 8 làm bit thứ hai, v.v và bit thứ 25 của khối đầu ra trước đó làm bit cuối cùng.

Bảng A.2 – Hoán vị đầu nghịch đảo 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

A.4.3 Hàm f

Hàm f được chỉ ra trên Hình A.2

Hàm f nhận đầu vào 32 bit R và mở rộng thành R' có độ dài 48-bit bằng cách sử dụng phép hoán vị mở rộng E . Sau đó 48-bit R' được XOR với khóa con 48-bit K , thu được dữ liệu 48-bit, dữ liệu này được viết thành 8 khối, mỗi khối 6-bit, bằng cách chọn các bit đầu vào của nó theo một thứ tự được qui định trên bảng. Các hàm được chọn duy nhất đó được gọi là các S – hộp, S_1, S_2, \dots, S_8 , nhận đầu vào là các khối 6-bit r_i và cho đầu ra là các khối 4-bit $S_i(r_i)$. Hàm hoán vị P cho đầu ra 32-bit R''' từ đầu vào 32-bit R'' bằng cách hoán vị các bit của khối đầu vào. R''' là đầu ra của hàm f .

Hàm f do đó được xác định như sau (P và C là dữ liệu, K_i là khóa)

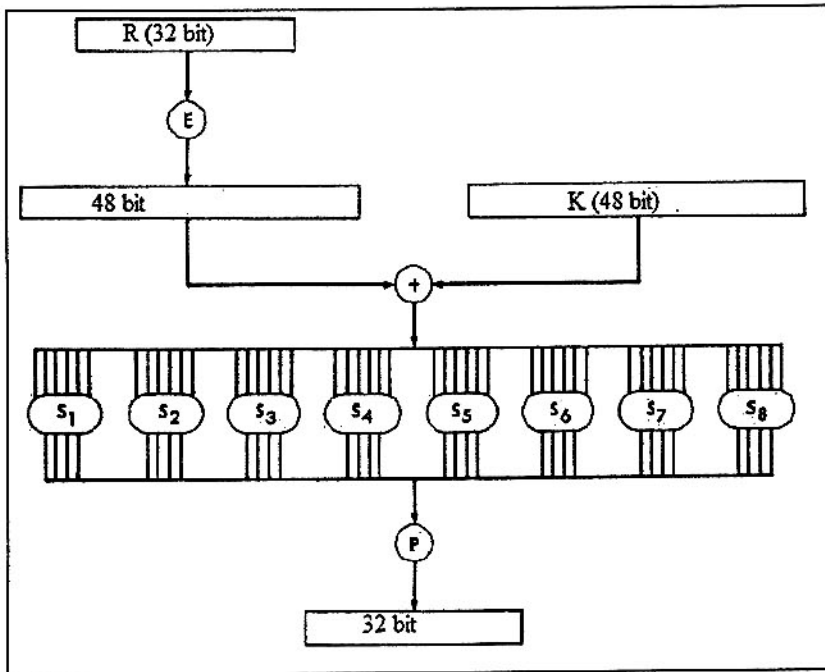
$$(1) R' = E(R)$$

$$(2) R'' = R' \oplus K$$

$$(3) R'' = r1 \parallel r2 \parallel r3 \parallel r4 \parallel r5 \parallel r6 \parallel r7 \parallel r8$$

$$R''' = S_1(r1) \parallel S_2(r2) \parallel S_3(r3) \parallel S_4(r4) \parallel S_5(r5) \parallel S_6(r6) \parallel S_7(r7) \parallel S_8(r8)$$

$$(4) R'''' = P(R''')$$

Hình A.2 – Tính $f(R, K)$

A.4.4 Hoán vị mở rộng E

Hoán vị mở rộng E được chỉ ra trên Bảng A.3. Hoán vị E nhận đầu vào 32-bit và cho đầu ra 48-bit. Ba bit đầu tiên của E là các bit ở các vị trí 32, 1 và 2, hai bit cuối ở các vị trí 32 và 1.

Bảng A.3 – Hoán vị mở rộng E

32	1	2	3	4	5	64	32
4	5	6	7	8	9	63	31
8	9	10	11	12	13	62	30
12	13	14	15	16	17	61	29
16	17	18	19	20	21	60	28
20	21	22	23	24	25	59	27
24	25	26	27	28	29	58	26
28	29	30	31	32	1	57	25

A.4.5 Hoán vị P

Hoán vị P được chỉ ra trên Bảng A.4. Hoán vị P nhận đầu vào 32-bit và cho đầu ra 32-bit. Đầu ra $P(L)$ của hàm P được xác định bởi Bảng A.4, thu được từ đầu vào L bằng cách lấy bit thứ 16 của L làm bit thứ nhất của $P(L)$, bit thứ bảy làm bit thứ hai của $P(L)$, v.v và bit thứ 25 của L làm bit thứ 32 của $P(L)$.

Bảng A.4 – Hoán vị 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

A.4.6 S-box

S-box được chỉ ra trên Bảng A.5. Mỗi S-box có 6-bit đầu vào và 4-bit đầu ra.

Nếu S_i là hàm được xác định trên bảng và B là khối 6-bit, thì $S_i(B)$ được xác định như sau: các bit đầu tiên và cuối cùng của B được biểu diễn theo cơ số 2 là một số nằm trong khoảng từ 0 đến 3. Giả sử số đó là i . Bốn bit giữa của B được biểu diễn theo cơ số 2 là số nằm trong khoảng từ 0 đến 15. Giả sử số đó là j . Trên bảng đó là số nằm trên hàng thứ i và cột thứ j . Đó là số nằm trong khoảng từ 0 đến 15, và được biểu diễn duy nhất bởi khối 4-bit. Khối này là đầu ra $S_i(B)$ của S_i ứng với đầu vào B . Ví dụ, với đầu vào 011011, hàng được biểu diễn bởi 01 là hàng 1, cột được xác định bởi 1101 là cột 13. Nằm trên giao của hàng 1 và cột 13 là số 5, do đó đầu ra là 0101.

Bảng A.5 – Các S-box

S1

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

S2

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

S3

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 5 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

A.5 Lược đồ tạo khóa DES

Phần lược đồ khóa DES được chỉ ra trên Hình A.3. Lược đồ này nhận khóa 64-bit khóa **KEY** và cho ra 16 khóa con 48-bit K_1, K_2, \dots, K_{16} .

Trong đó K_n , với $1 \leq n \leq 16$ là khối 48-bit trong bước (2) của thuật toán. Bởi vậy để mô tả **KS** chỉ cần mô tả việc tính toán của K_n từ **KEY** với $n = 1, 2, \dots, 16$. Việc tính toán này được mô tả trên Hình A.3. Do đó để xác định đầy đủ **KS**, chỉ cần mô tả hai lựa chọn hoán vị, cũng giống như lược đồ chuyển dịch sang trái. Một bit trong byte 8-bit của **KEY** có thể được khởi động để phát hiện lỗi trong tạo khóa, phân phối và lưu trữ khóa. Các bit 8, 16, ..., 64 được sử dụng để bảo đảm là mỗi byte có tính chất lẻ. Lựa chọn hoán vị 1 được xác định bởi Bảng A.6

Bảng A.6 – Hoán vị khóa PC-1

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

Bảng trên được chia thành hai phần, phần thứ nhất xác định cách chọn các bit trong C_0 , phần thứ hai xác định cách chọn các bit trong D_0 . Các bit của **KEY** được đánh số từ 1 đến 64. Các bit trong C_0 tương ứng là 57, 49, 41, ..., 44 và 36 của **KEY**, còn các bit trong D_0 là các bit 63, 55, 47, ..., 12 và 4 của **KEY**. Với C_0 và D_0 đã xác định, có thể xác định được C_n và D_n từ C_{n-1} và D_{n-1} , tương ứng với $n = 1, 2, \dots, 16$. Điều này đạt được bằng cách tuân thủ vào lược đồ phép dịch sang trái các khối riêng lẻ sau:

Bảng A.7 – Số các bit khóa được dịch chuyển của mỗi vòng

Số lần lặp	Số lần dịch trái
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
Số lần lặp	Số lần dịch trái
9	1
10	2
11	2

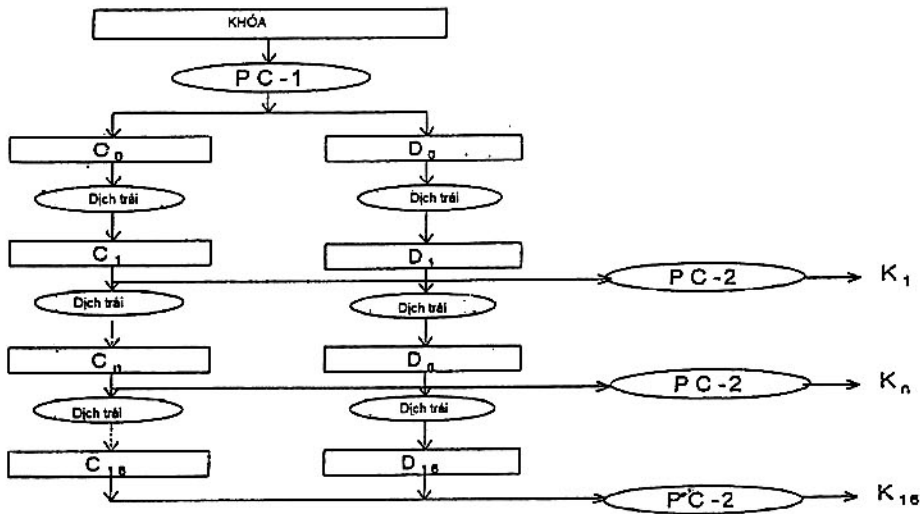
12	2
13	2
14	2
15	2
15	1

Ví dụ, C_3 và D_3 thu được từ C_2 và D_2 tương ứng bằng hai dịch chuyển sang trái, và C_{16} và D_{16} thu được từ C_{15} và D_{15} tương ứng bằng một lần dịch sang trái. Trong tất cả các trường hợp, việc dịch một lần sang trái được hiểu là dịch các bit sang trái một vị trí sao cho sau một lần dịch các bit ở 28 vị trí là các bit trước đó ở vị trí 2, 3, ..., 28, 1. Việc chọn hoán vị 2 được xác định trên Bảng A.8

Bảng A.8 – Hoán vị nén 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	10	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Bởi vậy bit thứ nhất của K_n là bit thứ 14 của $C_n D_n$, bit thứ hai là bit thứ 17, v.v, bit thứ 47 là bit thứ 29, bit thứ 48 là bit thứ 32.



Hình A.3 – Tính toán lược đồ khóa

Phụ lục B
(Quy định)
Định danh đối tượng

Phụ lục này liệt kê các định danh đối tượng được gán cho các Thuật toán đã được mô tả trong tiêu chuẩn này của bộ TCVN 11367 (ISO/IEC 18033).

CHÚ THÍCH

Trong các ứng dụng, nơi tổ hợp các thuật toán được sử dụng để cung cấp dịch vụ an toàn hoặc khi thuật toán được tham số hóa bằng sự lựa chọn tổ hợp các thuật toán khác thì tổ hợp này có thể được xác định như là dãy các định danh đối tượng được gán cho các thuật toán này hoặc bao gồm cả các định danh của các thuật toán ở tầng thấp hơn (ví dụ bằng việc xác định định danh đối tượng của cơ chế bọc khóa như một tham số trong cấu trúc định danh thuật toán xác định thuật toán mã hóa lai). Cấu trúc định danh thuật toán được xác định trong ISO/IEC 9594-8.

```
-- ISO/IEC 18033-3 ASN.1 Module
```

```
EncryptionAlgorithms-3 {
    iso(1) standard(0) encryption-algorithms(18033) part(3)
        asn1-module(0) algorithm-object-identifiers(0) version(1)}

    DEFINITIONS EXPLICIT TAGS ::= BEGIN

-- EXPORTS All;
-- IMPORTS None; --

OID ::= OBJECT IDENTIFIER -- Alias

-- Synonyms --
id18033-3    OID ::= {iso(1) standard(0) encryption-algorithms(18033) part(3)}
id-bc64     OID ::= { is18033-3 cipher-64-bit(1)      }
id-bc128    OID ::= { is18033-3 cipher-128-bit(2)    }

-- Assignments --
id-bc64-tdea      OID ::= {id-bc64 tdea(1)      }
id-bc64-mistyl    OID ::= {id-bc64 mistyl(2)}
id-bc64-cast128   OID ::= {id-bc64 cast128(3)}
id-bc64-hight     OID ::= {id-bc64 hight(4)}
id-bc128-aes      OID ::= {id-bc128 aes(1)}
id-bc128-camellia  OID ::= {id-bc128 camellia(2)}
id-bc128-seed     OID ::= {id-bc128 seed(3)}

EncryptionAlgorithmIdentifier{ALGORITHM:BlockAlgorithms} ::= SEQUENCE {
    algorithm ALGORITHM.&id({BlockAlgorithms}),
    parametersALGORITHM.SType({BlockAlgorithms}){@algorithm} OPTIONAL
}

BlockAlgorithms ALGORITHM ::= {
```

```

    { OID id-bc64-idea PARMS KeyLengthID } |
    { OID id-bc64-mistyl PARMS KeyLength } |
    { OID id-bc64-cast128 PARMS KeyLength } |
    { OID id-bc64-hight PARMS KeyLength } |
    { OID id-bc128-aes PARMS KeyLengthID } |
    { OID id-bc128-camellia PARMS KeyLengthID}
    { OID id-bc128-seed PARMS KeyLength}.
-- Except additional algorithms --
}
KeyLength ::= INTEGER
KeyLengthID ::= CHOICE {
    int KeyLength,
    oid OID
}
-- Cryptographic algorithm identification --
ALGORITHM ::= CLASS {
    &id OBJECT IDENTIFIER UNIQUE,
    &Type OPTIONAL
}
    WITH SYNTAX {OID &id [PARMS &Type] }
END -- EncryptionAlgorithms-3 --

```


Phụ lục C

(Quy định)

Dạng đại số của MISTY1 và các S-box Camellia

C.1 Mở đầu

Trong MISTY1 và các thuật toán Camellia, các S-box có thể được mô tả tương ứng bằng các dạng đại số sau đây.

C.2 S-box trong MISTY1

C.2.1 Các S-box S_7 và S_9

Điều này mô tả các dạng đại số trên trường GF(2) của S_7 và S_9 trong thuật toán MISTY1

Ví dụ, nếu đầu vào của S_7 là $\{53\}$, khi đó $X = (x_6, x_5, x_4, x_3, x_2, x_1, x_0) = (1, 0, 1, 0, 0, 1, 1)$. Giá trị thay thế có thể lấy bằng dạng đại số trên Hình C.1 và từ $Y = (y_6, y_5, y_4, y_3, y_2, y_1, y_0) = (1, 0, 1, 0, 1, 1, 1)$. Kết quả cho ta giá trị $S_7 = \{57\}$

C.2.2 MISTY1 S-box S_7

$$\begin{aligned}
 y_0 &= x_0 \oplus x_1 x_3 \oplus x_0 x_3 x_4 \oplus x_1 x_5 \oplus x_0 x_2 x_5 \oplus x_4 x_5 \oplus x_0 x_1 x_6 \oplus x_2 x_6 \oplus x_0 x_5 x_6 \oplus x_3 x_5 x_6 \oplus 1 \\
 y_1 &= x_0 x_2 \oplus x_0 x_4 \oplus x_3 x_4 \oplus x_1 x_5 \oplus x_2 x_4 x_5 \oplus x_6 \oplus x_0 x_6 \oplus x_3 x_6 \oplus x_2 x_3 x_6 \oplus x_1 x_4 x_6 \oplus x_0 x_5 x_6 \oplus 1 \\
 y_2 &= x_1 x_2 \oplus x_0 x_2 x_3 \oplus x_4 \oplus x_1 x_4 \oplus x_0 x_1 x_4 \oplus x_0 x_5 \oplus x_0 x_4 x_5 \oplus x_3 x_4 x_5 \oplus x_1 x_6 \oplus x_3 x_6 \oplus x_0 x_3 x_6 \oplus \\
 & x_4 x_6 \oplus x_2 x_4 x_6 \\
 y_3 &= x_0 \oplus x_1 \oplus x_0 x_1 x_2 \oplus x_0 x_3 \oplus x_2 x_4 \oplus x_1 x_4 x_5 \oplus x_2 x_6 \oplus x_1 x_3 x_6 \oplus x_0 x_4 x_6 \oplus x_5 x_6 \oplus 1 \\
 y_4 &= x_2 x_3 \oplus x_0 x_4 \oplus x_1 x_3 x_4 \oplus x_5 \oplus x_2 x_5 \oplus x_1 x_2 x_5 \oplus x_0 x_3 x_5 \oplus x_1 x_6 \oplus x_1 x_5 x_6 \oplus x_4 x_5 x_6 \oplus 1 \\
 y_5 &= x_0 \oplus x_1 \oplus x_2 \oplus x_0 x_1 x_2 \oplus x_0 x_3 \oplus x_1 x_2 x_3 \oplus x_1 x_4 \oplus x_0 x_2 x_4 \oplus x_0 x_5 \oplus x_0 x_1 x_5 \oplus x_3 x_5 \oplus x_0 x_6 \oplus \\
 & x_2 x_5 x_6 \\
 y_6 &= x_0 x_1 \oplus x_3 \oplus x_0 x_3 \oplus x_2 x_3 x_4 \oplus x_0 x_5 \oplus x_2 x_5 \oplus x_3 x_5 \oplus x_1 x_3 x_5 \oplus x_1 x_6 \oplus x_1 x_2 x_6 \oplus x_0 x_3 x_6 \oplus x_4 \\
 & x_6 \oplus x_2 x_5 x_6
 \end{aligned}$$

Bảng C.1 – Dạng đại số của S_7

$$(y_6 2^6 \oplus y_5 2^5 \oplus \dots \oplus y_1 2^1 \oplus y_0 2^0 = S_7(x_6 2^6 \oplus x_5 2^5 \oplus \dots \oplus x_1 2^1 \oplus x_0 2^0))$$

C.2.3 MISTY1 S-box S_9

$$\begin{aligned}
 y_0 &= x_0 x_4 \oplus x_0 x_5 \oplus x_1 x_5 \oplus x_1 x_6 \oplus x_2 x_6 \oplus x_2 x_7 \oplus x_3 x_7 \oplus x_3 x_8 \oplus x_4 x_8 \oplus 1 \\
 y_1 &= x_0 x_2 \oplus x_3 \oplus x_1 x_3 \oplus x_2 x_3 \oplus x_3 x_4 \oplus x_4 x_5 \oplus x_0 x_6 \oplus x_2 x_6 \oplus x_7 \oplus x_0 x_8 \oplus x_3 x_8 \oplus x_3 x_8 \oplus 1 \\
 y_2 &= x_0 x_1 \oplus x_1 x_3 \oplus x_4 \oplus x_0 x_4 \oplus x_2 x_4 \oplus x_3 x_4 \oplus x_4 x_5 \oplus x_0 x_6 \oplus x_5 x_6 \oplus x_1 x_7 \oplus x_3 x_7 \oplus x_8 \\
 y_3 &= x_0 \oplus x_1 x_2 \oplus x_2 x_4 \oplus x_5 \oplus x_1 x_5 \oplus x_3 x_5 \oplus x_4 x_5 \oplus x_5 x_6 \oplus x_1 x_7 \oplus x_6 x_7 \oplus x_2 x_8 \oplus x_4 x_8 \\
 y_4 &= x_1 \oplus x_0 x_3 \oplus x_2 x_3 \oplus x_0 x_5 \oplus x_3 x_5 \oplus x_6 \oplus x_2 x_6 \oplus x_4 x_6 \oplus x_5 x_6 \oplus x_6 x_7 \oplus x_2 x_8 \oplus x_7 x_8
 \end{aligned}$$

$$\begin{aligned}
 y_5 &= x_2 \oplus x_0 x_3 \oplus x_1 x_4 \oplus x_3 x_4 \oplus x_1 x_6 \oplus x_4 x_6 \oplus x_7 \oplus x_3 x_7 \oplus x_5 x_7 \oplus x_6 x_7 \oplus x_0 x_8 \oplus x_7 x_8 \\
 y_6 &= x_0 x_1 \oplus x_3 \oplus x_1 x_4 \oplus x_2 x_5 \oplus x_4 x_5 \oplus x_2 x_7 \oplus x_5 x_7 \oplus x_8 \oplus x_0 x_8 \oplus x_4 x_8 \oplus x_6 x_8 \oplus x_7 x_8 \oplus 1 \\
 y_7 &= x_1 \oplus x_0 x_1 \oplus x_1 x_2 \oplus x_2 x_3 \oplus x_0 x_4 \oplus x_5 \oplus x_1 x_6 \oplus x_3 x_6 \oplus x_0 x_7 \oplus x_4 x_8 \oplus x_6 x_7 \oplus x_1 x_8 \oplus 1 \\
 y_8 &= x_0 \oplus x_0 x_1 \oplus x_1 x_2 \oplus x_4 \oplus x_0 x_5 \oplus x_2 x_5 \oplus x_3 x_6 \oplus x_5 x_6 \oplus x_0 x_7 \oplus x_0 x_8 \oplus x_3 x_8 \oplus x_6 x_8 \oplus 1
 \end{aligned}$$

Bảng C.2 – Dạng đại số của S_9

$$(y_8 2^8 \oplus y_7 2^7 \oplus \dots \oplus y_1 2^1 \oplus y_0 2^0 = S_9(x_8 2^8 \oplus x_7 2^7 \oplus \dots \oplus x_1 2^1 \oplus x_0 2^0))$$

C.3 Các S-box Camellia

Điều này mô tả các dạng đại số trên trường $GF(2^8)$ của S-box trong Thuật toán Camellia.

s_1 được cho như sau (đầu vào là x và đầu ra là y):

$$s_1: y = h(g(f(c5 \oplus x))) \oplus 6e$$

Ở đây các phép toán f , g và h nhận đầu vào 8-bit $a = a_1 \| a_2 \| a_3 \| a_4 \| a_5 \| a_6 \| a_7 \| a_8$ và 8-bit đầu ra $b = b_1 \| b_2 \| b_3 \| b_4 \| b_5 \| b_6 \| b_7 \| b_8$, a_i và b_i là các giá trị 1-bit, f là hoán vị affine của đầu vào, g là phép nghịch đảo trên trường $GF(2^8)$ và h là biến đổi affine của đầu ra.

Trong phương trình trên, các phép toán f , g và h được mô tả dưới dạng đại số như sau:

f :

$$\begin{aligned}
 b_1 &= a_6 \oplus a_2, & b_5 &= a_7 \oplus a_6, \\
 b_2 &= a_7 \oplus a_1, & b_6 &= a_5 \oplus a_2, \\
 b_3 &= a_8 \oplus a_5 \oplus a_3, & b_7 &= a_8 \oplus a_1, \\
 b_4 &= a_8 \oplus a_3, & b_8 &= a_6 \oplus a_4,
 \end{aligned}$$

g :

$$\begin{aligned}
 &(b_8 + b_7 \alpha + b_6 \alpha^2 + b_5 \alpha^3) + (b_4 + b_3 \alpha + b_2 \alpha^2 + b_1 \alpha^3) \beta \\
 &= 1 / ((a_8 + a_7 \alpha + a_6 \alpha^2 + a_5 \alpha^3) + (a_4 + a_3 \alpha + a_2 \alpha^2 + a_1 \alpha^3) \beta)
 \end{aligned}$$

g là phép nghịch đảo trong trường $GF(2^8)$, giả thiết $\frac{1}{0} = 0$, ở đây β là phần tử trong trường $GF(2^8)$ thỏa mãn điều kiện $\beta^8 + \beta^6 + \beta^5 + \beta^3 + 1 = 0$ và $\alpha = \beta^{238} = \beta^6 + \beta^5 + \beta^3 + \beta^2$ là phần tử trong trường $GF(2^4)$ thỏa mã $\alpha^4 + \alpha + 1 = 0$

h :

$$\begin{aligned}
 b_1 &= a_5 \oplus a_6 \oplus a_2, & b_5 &= a_7 \oplus a_3, \\
 b_2 &= a_6 \oplus a_2, & b_6 &= a_8 \oplus a_1, \\
 b_3 &= a_7 \oplus a_4, & b_7 &= a_5 \oplus a_1, \\
 b_4 &= a_8 \oplus a_2, & b_8 &= a_6 \oplus a_3,
 \end{aligned}$$

Phụ lục D
(Tham khảo)
Véc tơ kiểm tra

D.1 Mở đầu

Phụ lục này cung cấp các véc tơ kiểm tra cho TDEA, MISTY1, CAST-128, HIGHT, AES, Camellia và SEED. Trong các ví dụ này, tất cả dữ liệu được biểu diễn dưới dạng Hexa.

D.2 Véc tơ kiểm tra TDEA**D.2.1 Mã hóa TDEA**

Với đầu vào đã cho (bản rõ và khóa), đầu ra (bản mã) được mô tả như sau:

Bảng D.1 – Các ví dụ của mã hóa TDEA

STT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	$K_1 = 0123456789abcdef$ $K_2 = 23456789abcdef01$ $K_3 = 456789abcdef0123$	-
	Mã hóa	$P = 4e6f772069732074$	$C = 314f8327fa7a09a8$
2	Khóa	$K_1 = 0123456789abcdef$ $K_2 = 23456789abcdef01$ $K_3 = 456789abcdef0123$	-
	Mã hóa	$P = 68652074696d6520$	$C = 4362760cc13ba7da$
3	Khóa	$K_1 = 0123456789abcdef$ $K_2 = 23456789abcdef01$ $K_3 = 456789abcdef0123$	-
	Mã hóa	$P = 666f7220616c6c20$	$C = ff55c5f80faaac45$
4	Khóa	$K_1 = 0123456789abcdef$ $K_2 = 23456789abcdef01$ $K_3 = 0123456789abcdef$	-
	Mã hóa	$P = 4e6f772069732074$	$C = 03e69f5bfa58eb42$

5	Khóa	$K_1 = 02c4da3d73f226ad$ $K_2 = 1cbce0f2bacd3b15$ $K_3 = 02c4da3d73f226ad$	-
	Mã hóa	$P = 03e69f5bfa58eb42$	$C = 262a60f9743e1fd8$
6	Khóa	$K_1 = 0123456789abcdef$ $K_2 = 23456789abcdef01$ $K_3 = 0123456789abcdef$	-
	Mã hóa	$P = 6996c8fa47a2abeb$	$C = 4e6f772069732074$
7	Khóa	$K_1 = 68b58c9dce086704$ $K_2 = 529dce3719e9e0da$ $K_3 = 68b58c9dce086704$	-
	Mã hóa	$P = 6b177e016e6ae12d$	$C = 6996c8fa47a2abeb$

D.2.2 Mã hóa và giải mã DES

D.2.2.1 Đầu vào và khóa đầu ra

Đầu vào và đầu ra của mỗi thuật toán DES được mô tả với các khóa sau:

$$K_1 = 0123456789abcdef$$

$$K_2 = 23456789abcdef01$$

$$K_3 = 456789abcdef0123$$

D.2.2.2 Phép mã hóa TDEA (DES mã hóa – giải mã – mã hóa)

Bảng D.2 – Ví dụ thứ nhất của phép mã hóa TDEA

Các biến đổi	Đầu vào	Đầu ra
$DES_1 - E_{K_1}$	4e6f772069732074	3fa40e8a984d4815
$DES_2 - D_{K_2}$	3fa40e8a984d4815	0663d1b37c48090c
$DES_3 - E_{K_3}$	0663d1b37c48090c	314f8327fa7a09a8

Bảng D.3 – Ví dụ thứ hai của phép mã hóa TDEA

Các biến đổi	Đầu vào	Đầu ra
$DES_1 - E_{K_1}$	68652074696d6520	6a271787ab8883f9

$DES_2 - D_{K2}$	6a271787ab8883f9	95cae06a9ff4d6d2
$DES_3 - E_{K3}$	95cae06a9ff4d6d2	4362760cc13ba7da

Bảng D.4 – Ví dụ thứ ba của phép mã hóa TDEA

Các biến đổi	Đầu vào	Đầu ra
$DES_1 - E_{K1}$	666f7220616c6c20	893d51ec4b563b53
$DES_2 - D_{K2}$	893d51ec4b563b53	5fed3eb05419f67c
$DES_3 - E_{K3}$	5fed3eb05419f67c	ff55c5f80aac45

D.2.2.3 Phép giải mã TDEA (DES giải mã – mã hóa – giải mã)

Bảng D.5 – Ví dụ thứ nhất của phép mã hóa TDEA

Các biến đổi	Đầu vào	Đầu ra
$DES_1 - D_{K3}$	314f8327fa7a09a8	0663d1b37c48090c
$DES_2 - E_{K2}$	0663d1b37c48090c	3fa40e8a984d4815
$DES_3 - D_{K1}$	3fa40e8a984d4815	4e6f772069732074

Bảng D.6 – Ví dụ thứ hai của phép mã hóa TDEA

Các biến đổi	Đầu vào	Đầu ra
$DES_1 - D_{K3}$	4362760cc13ba7da	95cae06a9ff4d6d2
$DES_2 - E_{K2}$	95cae06a9ff4d6d2	6a271787ab8883f9
$DES_3 - D_{K1}$	6a271787ab8883f9	68652074696d6520

Bảng D.7 – Ví dụ thứ ba của phép mã hóa TDEA

Các biến đổi	Đầu vào	Đầu ra
$DES_1 - D_{K3}$	ff55c5f80aac45	5fed3eb05419f67c
$DES_2 - E_{K2}$	5fed3eb05419f67c	893d51ec4b563b53

$DES_3 - D_{K1}$	893d51ec4b563b53	666f7220616c6c20
------------------	------------------	------------------

D.3 Véc tơ kiểm tra MISTY1

Với đầu vào đã cho (bản rõ và khóa), đầu ra (bản mã và khóa con) được mô tả như sau:

Bảng D.8 – Ví dụ của phép mã hóa MISTY1

STT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	K = 00112233445566778899aabbccddeeff	K' = cf518e7f5e29673acdbc07d6bf355e11
	Mã hóa	P = 0123456789abcdef	C = 8b1da5f56ab3d07c
2	Khóa	K = 414afd99bb577ee69df58cc8fb4e6888	K' = c7bd6e012268237a4389305a1b360b8c
	Mã hóa	P = 9fc302e281310e90	C = 15c270974b9b9163
3	Khóa	K = 3c54aed9a5389c947167db9d97c6967 a	K' = 7c8e13ebfe7648050c9097934205662b
	Mã hóa	P = 032c4a4a100ee807	C = 3346cb8c779cf2de
4	Khóa	K = d3f11 a6d25f1 b386fdada0b5e53fa17	K' = f011 d035ac920f832f69bcf7b860d4f0
	Mã hóa	P = db9e3218402023f3	C = b2dd1595a450bc9
5	Khóa	K = 5f87f88ec7641 d83af03fd8327821046	K' = 3736172d7421c91401596db29d3d5536
	Mã hóa	P = 6553de24c0dd900b	C = 60081 e65cb7c2b84

D.4 Véc tơ kiểm tra CAST-128

Với đầu vào đã cho (bản rõ và khóa), đầu ra (bản mã) được mô tả như sau:

Bảng D.9 – Ví dụ của phép mã hóa CAST -128

STT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	$K =$ 0123456712345678234567893456789a	
	Mã hóa	$P =$ 0123456789abcdef	$C = 238b4fe5847e44b2$

D.5 Véc tơ kiểm tra HIGHT

Bảng D.10 – Ví dụ của phép mã hóa HIGHT

STT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	$K =$ 00112233445566778899aabbccddeeff	
	Mã hóa	$P =$ 0000000000000000	$C = 00f418aed94f03f2$
2	Khóa	$K =$ ffeeddccbbaa99887766554433221100	
	Mã hóa	$P =$ 0011223344556677	$C = 23ce9f72e543e6d8$
3	Khóa	$K =$ 000102030405060708090a0b0c0d0e0f	
	Mã hóa	$P =$ 0123456789abcdef	$C = 7a6fb2a28d23f466$
4	Khóa	$K =$ 28dbc3bc49fd87dcfa509b11d422be7	
	Mã hóa	$P =$ b41e6be2eba84a14	$C = cc047a75209c1fc6$

D.6 Véc tơ kiểm tra AES

D.6.1 Mã hóa AES

D.6.1.1 Mở đầu

Với đầu vào đã cho (bản rõ và khóa), đầu ra (bản mã) được mô tả như sau:

D.6.1.2 Khóa 128 bit

Bảng D.11 – Ví dụ của phép mã hóa AES với 128 bit khóa

STT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	K = 000102030405060708090a0b0c0d0e0f	
	Mã hóa	P = 00112233445566778899aabbccddeeff	C = 69c4e0d86a7b0430d8cdb78070b4c55a

D.6.1.3 Khóa 192 bit

Bảng D.12 – Ví dụ của phép mã hóa với 192 bit khóa

STT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	K = 000102030405060708090a0b0c0d0e0f 1011121314151617	
	Mã hóa	P = 00112233445566778899aabbccddeeff	C = dda97ca4864cdf06eaf70a0ec0d7191

D.6.1.4 Khóa 256 bit

Bảng D.13 – Ví dụ của phép mã hóa với 256 bit khóa

STT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	K = 000102030405060708090a0b0c0d0e0f 101112131415161718191a1b1c1d1e1f	
	Mã hóa	P = 00112233445566778899aabbccddeeff	C = 8ea2b7ca516745bfeafc49904b496089

D.6.2 Ví dụ mở rộng khóa

Điều này chỉ ra sự phát triển của việc sử dụng lược đồ khóa

Cipher key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

Với $Nk = 4$, kết quả:

$$w_0 = 2b7e1516 \quad w_1 = 28aed2a6 \quad w_2 = abf71588 \quad w_3 = 09cf4f3c$$

Các giá trị trung gian của khóa được đưa sinh ra trong quá trình phát triển của lược đồ khóa như sau (tất cả các được định dạng Hexa, trừ cột chỉ số (i)).

Bảng D.14 – Ví dụ mở rộng của khóa

I (dec)	Byte tạm thời	Sau ShiftColumn()	Sau SubByte()	Rc[i/Nk]	Sau XOR Với Rcon	w[i-Nk]	w[i]= bytes tạm thời XOR w[i-Nk]
4	09cf4f3c	cf4f3c09	8a84eb01	01000000	8b84eb01	2b7e1516	a0fafe17
5	a0fafe17					28aed2a6	88542cb1
6	88542cb1					abf71588	23a33939
7	23a33939					09cf4f3c	2a6c7605
8	2a6c7605	6c76052a	50386be5	02000000	52386be5	a0fafe17	f2c295f2
9	f2c295f2					88542cb1	7a96b943
10	7a96b943					23a33939	5935807a
11	5935807a					2a6c7605	7359f67f
12	7359f67f	59f67f73	cb42d28f	04000000	cf42d28f	f2c295f2	3d80477d
13	3d80477d					7a96b943	4716fe3e
14	4716fe3e					5935807a	1e237e44
15	1e237e44					7359f67f	6d7a883b
16	6d7a883b	7a883b6d	dac4e23c	08000000	d2c4e23c	3d80477d	ef44a541
17	ef44a541					4716fe3e	a8525b7f
18	a8525b7f					1e237e44	b671253b
19	b671253b					6d7a883b	db0bad00
20	db0bad00	0bad00db	2b9563b9	10000000	3b9563b9	ef44a541	d4d1c6f8
21	d4d1c6f8					a8525b7f	7c839d87
22	7c839d87					b671253b	caf2b8bc
23	caf2b8bc					db0bad00	11f915bc
24	11f915bc	f915bc11	99596582	20000000	b9596582	d4d1c6f8	6d88a37a
25	6d88a37a					7c839d87	110b3efd
26	110b3efd					caf2b8bc	dbf98641
27	dbf98641					11 f915bc	ca0093fd
28	ca0093fd	0093fdca	63dc5474	40000000	23dc5474	6d88a37a	4e54f70e
29	4e54f70e					110b3efd	5f5fc9f3
30	5f5fc9f3					dbf98641	84a64fb2
31	84a64fb2					ca0093fd	4ea6dc4f
32	4ea6dc4f	a6dc4f4e	2486842f	80000000	a486842f	4e54f70e	ead27321
33	ead27321					5f5fc9f3	b58dbad2
34	b58dbad2					84a64fb2	312bf560

I (dec)	Byte tạm thời	Sau ShiftColumn()	Sau SubByte()	Rc[i/Nk]	Sau XOR Với Rcon	w[i-Nk]	w[i]= bytes tạm thời XOR w[i-Nk]
35	312bf560					4ea5dc4f	7f8d292f
36	7f8d292f	8d292f7f	5da515d2	1b000000	46a515d2	ead27321	ac7766f3
37	ac7766f3					b58dbad2	19fadc21
38	19fadc21					312bf560	28d12941
39	28d12941					7f8d292f	575c006e
40	575c006e	5c006e57	4a639f5b	36000000	7c639f5b	ac7766f3	d014f9a8
41	d014f9a8					19fadc21	c9ee2589
42	c9ee2589					28d12941	e13f0cc8
43	e13f0cc8					575c006e	b6630ca6

D.6.3 Ví dụ mật mã

Biểu đồ sau chỉ ra các giá trị mảng Trạng thái như một tiến trình mật mã cho mỗi khối, khóa mật mã 16 byte.

Input = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
 Cipher key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

Các giá trị vòng khóa được lấy từ ví dụ mở rộng khóa trong D.6.2

Bảng D.15 – Ví dụ mật mã

Số vòng	Bắt đầu vòng	Sau SubBytes	Sau ShiftRows	Sau MixColumns	Giá trị khóa vòng																																																																																	
input	<table border="1"> <tr><td>32</td><td>88</td><td>31</td><td>e0</td></tr> <tr><td>43</td><td>5a</td><td>31</td><td>37</td></tr> <tr><td>f6</td><td>30</td><td>98</td><td>07</td></tr> <tr><td>a8</td><td>8d</td><td>a2</td><td>34</td></tr> </table>	32	88	31	e0	43	5a	31	37	f6	30	98	07	a8	8d	a2	34	<table border="1"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>																	<table border="1"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>																	<table border="1"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>																	<table border="1"> <tr><td>2b</td><td>28</td><td>ab</td><td>09</td></tr> <tr><td>7e</td><td>ae</td><td>f7</td><td>cf</td></tr> <tr><td>15</td><td>d2</td><td>15</td><td>4f</td></tr> <tr><td>16</td><td>a6</td><td>88</td><td>3c</td></tr> </table>	2b	28	ab	09	7e	ae	f7	cf	15	d2	15	4f	16	a6	88	3c	=
32	88	31	e0																																																																																			
43	5a	31	37																																																																																			
f6	30	98	07																																																																																			
a8	8d	a2	34																																																																																			
2b	28	ab	09																																																																																			
7e	ae	f7	cf																																																																																			
15	d2	15	4f																																																																																			
16	a6	88	3c																																																																																			
1	<table border="1"> <tr><td>19</td><td>a0</td><td>9a</td><td>e9</td></tr> <tr><td>3d</td><td>f4</td><td>c6</td><td>f8</td></tr> <tr><td>e3</td><td>e2</td><td>8d</td><td>48</td></tr> <tr><td>be</td><td>2b</td><td>2a</td><td>08</td></tr> </table>	19	a0	9a	e9	3d	f4	c6	f8	e3	e2	8d	48	be	2b	2a	08	<table border="1"> <tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr> <tr><td>27</td><td>bf</td><td>b4</td><td>41</td></tr> <tr><td>11</td><td>98</td><td>5d</td><td>52</td></tr> <tr><td>ae</td><td>f1</td><td>e5</td><td>30</td></tr> </table>	d4	e0	b8	1e	27	bf	b4	41	11	98	5d	52	ae	f1	e5	30	<table border="1"> <tr><td>d4</td><td>e0</td><td>b8</td><td>1e</td></tr> <tr><td>bf</td><td>b4</td><td>41</td><td>27</td></tr> <tr><td>5d</td><td>52</td><td>11</td><td>98</td></tr> <tr><td>30</td><td>ae</td><td>f1</td><td>e5</td></tr> </table>	d4	e0	b8	1e	bf	b4	41	27	5d	52	11	98	30	ae	f1	e5	<table border="1"> <tr><td>04</td><td>e0</td><td>48</td><td>28</td></tr> <tr><td>66</td><td>cb</td><td>f8</td><td>06</td></tr> <tr><td>81</td><td>19</td><td>d3</td><td>26</td></tr> <tr><td>e5</td><td>9a</td><td>7a</td><td>4c</td></tr> </table>	04	e0	48	28	66	cb	f8	06	81	19	d3	26	e5	9a	7a	4c	<table border="1"> <tr><td>a0</td><td>88</td><td>23</td><td>2a</td></tr> <tr><td>fa</td><td>54</td><td>a3</td><td>6c</td></tr> <tr><td>fe</td><td>2c</td><td>29</td><td>76</td></tr> <tr><td>17</td><td>b1</td><td>39</td><td>05</td></tr> </table>	a0	88	23	2a	fa	54	a3	6c	fe	2c	29	76	17	b1	39	05	=
19	a0	9a	e9																																																																																			
3d	f4	c6	f8																																																																																			
e3	e2	8d	48																																																																																			
be	2b	2a	08																																																																																			
d4	e0	b8	1e																																																																																			
27	bf	b4	41																																																																																			
11	98	5d	52																																																																																			
ae	f1	e5	30																																																																																			
d4	e0	b8	1e																																																																																			
bf	b4	41	27																																																																																			
5d	52	11	98																																																																																			
30	ae	f1	e5																																																																																			
04	e0	48	28																																																																																			
66	cb	f8	06																																																																																			
81	19	d3	26																																																																																			
e5	9a	7a	4c																																																																																			
a0	88	23	2a																																																																																			
fa	54	a3	6c																																																																																			
fe	2c	29	76																																																																																			
17	b1	39	05																																																																																			

2	<table border="1"><tr><td>a4</td><td>68</td><td>6b</td><td>02</td></tr><tr><td>9c</td><td>9f</td><td>5b</td><td>6a</td></tr><tr><td>1f</td><td>35</td><td>ea</td><td>50</td></tr><tr><td>f2</td><td>2b</td><td>43</td><td>49</td></tr></table>	a4	68	6b	02	9c	9f	5b	6a	1f	35	ea	50	f2	2b	43	49	<table border="1"><tr><td>49</td><td>45</td><td>7f</td><td>77</td></tr><tr><td>de</td><td>db</td><td>39</td><td>02</td></tr><tr><td>d2</td><td>96</td><td>87</td><td>53</td></tr><tr><td>89</td><td>f1</td><td>1a</td><td>3b</td></tr></table>	49	45	7f	77	de	db	39	02	d2	96	87	53	89	f1	1a	3b	<table border="1"><tr><td>49</td><td>45</td><td>7f</td><td>77</td></tr><tr><td>db</td><td>39</td><td>02</td><td>de</td></tr><tr><td>87</td><td>53</td><td>d2</td><td>96</td></tr><tr><td>3b</td><td>89</td><td>f1</td><td>1a</td></tr></table>	49	45	7f	77	db	39	02	de	87	53	d2	96	3b	89	f1	1a	<table border="1"><tr><td>58</td><td>1b</td><td>db</td><td>1b</td></tr><tr><td>4d</td><td>4b</td><td>e7</td><td>6b</td></tr><tr><td>ca</td><td>5a</td><td>ca</td><td>b0</td></tr><tr><td>f1</td><td>ac</td><td>a8</td><td>e5</td></tr></table>	58	1b	db	1b	4d	4b	e7	6b	ca	5a	ca	b0	f1	ac	a8	e5	\oplus	<table border="1"><tr><td>f2</td><td>7a</td><td>59</td><td>73</td></tr><tr><td>c2</td><td>96</td><td>35</td><td>59</td></tr><tr><td>95</td><td>b9</td><td>80</td><td>f6</td></tr><tr><td>f2</td><td>43</td><td>7a</td><td>1f</td></tr></table>	f2	7a	59	73	c2	96	35	59	95	b9	80	f6	f2	43	7a	1f	=
	a4	68	6b	02																																																																																			
	9c	9f	5b	6a																																																																																			
	1f	35	ea	50																																																																																			
f2	2b	43	49																																																																																				
49	45	7f	77																																																																																				
de	db	39	02																																																																																				
d2	96	87	53																																																																																				
89	f1	1a	3b																																																																																				
49	45	7f	77																																																																																				
db	39	02	de																																																																																				
87	53	d2	96																																																																																				
3b	89	f1	1a																																																																																				
58	1b	db	1b																																																																																				
4d	4b	e7	6b																																																																																				
ca	5a	ca	b0																																																																																				
f1	ac	a8	e5																																																																																				
f2	7a	59	73																																																																																				
c2	96	35	59																																																																																				
95	b9	80	f6																																																																																				
f2	43	7a	1f																																																																																				

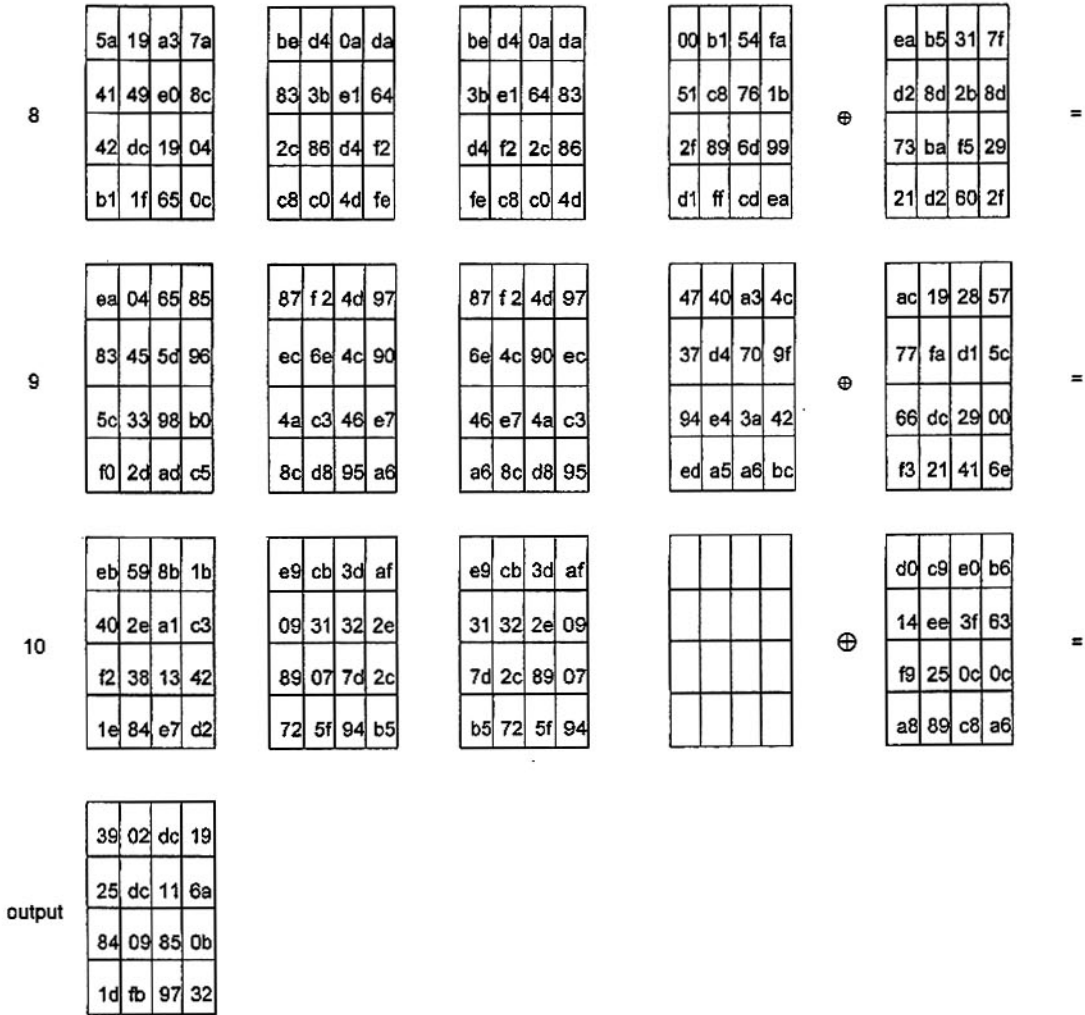
3	<table border="1"><tr><td>aa</td><td>61</td><td>82</td><td>68</td></tr><tr><td>8f</td><td>dd</td><td>d2</td><td>32</td></tr><tr><td>5f</td><td>e3</td><td>4a</td><td>46</td></tr><tr><td>03</td><td>ef</td><td>d2</td><td>9a</td></tr></table>	aa	61	82	68	8f	dd	d2	32	5f	e3	4a	46	03	ef	d2	9a	<table border="1"><tr><td>ac</td><td>ef</td><td>13</td><td>45</td></tr><tr><td>73</td><td>c1</td><td>b5</td><td>23</td></tr><tr><td>cf</td><td>11</td><td>d6</td><td>5a</td></tr><tr><td>7b</td><td>df</td><td>b5</td><td>b8</td></tr></table>	ac	ef	13	45	73	c1	b5	23	cf	11	d6	5a	7b	df	b5	b8	<table border="1"><tr><td>ac</td><td>ef</td><td>13</td><td>45</td></tr><tr><td>c1</td><td>b5</td><td>23</td><td>73</td></tr><tr><td>d6</td><td>5a</td><td>cf</td><td>11</td></tr><tr><td>b8</td><td>7b</td><td>df</td><td>b5</td></tr></table>	ac	ef	13	45	c1	b5	23	73	d6	5a	cf	11	b8	7b	df	b5	<table border="1"><tr><td>75</td><td>20</td><td>53</td><td>bb</td></tr><tr><td>ec</td><td>0b</td><td>c0</td><td>25</td></tr><tr><td>09</td><td>63</td><td>cf</td><td>d0</td></tr><tr><td>93</td><td>33</td><td>7c</td><td>dc</td></tr></table>	75	20	53	bb	ec	0b	c0	25	09	63	cf	d0	93	33	7c	dc	\oplus	<table border="1"><tr><td>3d</td><td>47</td><td>1e</td><td>6d</td></tr><tr><td>80</td><td>16</td><td>23</td><td>7a</td></tr><tr><td>47</td><td>fe</td><td>7e</td><td>88</td></tr><tr><td>7d</td><td>3e</td><td>44</td><td>3b</td></tr></table>	3d	47	1e	6d	80	16	23	7a	47	fe	7e	88	7d	3e	44	3b	=
	aa	61	82	68																																																																																			
	8f	dd	d2	32																																																																																			
	5f	e3	4a	46																																																																																			
03	ef	d2	9a																																																																																				
ac	ef	13	45																																																																																				
73	c1	b5	23																																																																																				
cf	11	d6	5a																																																																																				
7b	df	b5	b8																																																																																				
ac	ef	13	45																																																																																				
c1	b5	23	73																																																																																				
d6	5a	cf	11																																																																																				
b8	7b	df	b5																																																																																				
75	20	53	bb																																																																																				
ec	0b	c0	25																																																																																				
09	63	cf	d0																																																																																				
93	33	7c	dc																																																																																				
3d	47	1e	6d																																																																																				
80	16	23	7a																																																																																				
47	fe	7e	88																																																																																				
7d	3e	44	3b																																																																																				

4	<table border="1"><tr><td>48</td><td>67</td><td>4d</td><td>d6</td></tr><tr><td>6c</td><td>1d</td><td>e3</td><td>5f</td></tr><tr><td>4e</td><td>9d</td><td>b1</td><td>58</td></tr><tr><td>ee</td><td>0d</td><td>38</td><td>e1</td></tr></table>	48	67	4d	d6	6c	1d	e3	5f	4e	9d	b1	58	ee	0d	38	e1	<table border="1"><tr><td>52</td><td>85</td><td>e3</td><td>f6</td></tr><tr><td>50</td><td>a4</td><td>11</td><td>cf</td></tr><tr><td>2f</td><td>5e</td><td>c8</td><td>6a</td></tr><tr><td>28</td><td>d7</td><td>07</td><td>94</td></tr></table>	52	85	e3	f6	50	a4	11	cf	2f	5e	c8	6a	28	d7	07	94	<table border="1"><tr><td>52</td><td>85</td><td>e3</td><td>f6</td></tr><tr><td>a4</td><td>11</td><td>cf</td><td>50</td></tr><tr><td>c8</td><td>6a</td><td>2f</td><td>5e</td></tr><tr><td>94</td><td>28</td><td>d7</td><td>07</td></tr></table>	52	85	e3	f6	a4	11	cf	50	c8	6a	2f	5e	94	28	d7	07	<table border="1"><tr><td>0f</td><td>60</td><td>6f</td><td>5e</td></tr><tr><td>dc</td><td>31</td><td>c0</td><td>b3</td></tr><tr><td>da</td><td>38</td><td>10</td><td>13</td></tr><tr><td>a9</td><td>bf</td><td>6b</td><td>01</td></tr></table>	0f	60	6f	5e	dc	31	c0	b3	da	38	10	13	a9	bf	6b	01	\oplus	<table border="1"><tr><td>ef</td><td>a8</td><td>b6</td><td>db</td></tr><tr><td>44</td><td>52</td><td>71</td><td>0b</td></tr><tr><td>a5</td><td>5b</td><td>25</td><td>ad</td></tr><tr><td>41</td><td>7f</td><td>3b</td><td>00</td></tr></table>	ef	a8	b6	db	44	52	71	0b	a5	5b	25	ad	41	7f	3b	00	=
	48	67	4d	d6																																																																																			
	6c	1d	e3	5f																																																																																			
	4e	9d	b1	58																																																																																			
ee	0d	38	e1																																																																																				
52	85	e3	f6																																																																																				
50	a4	11	cf																																																																																				
2f	5e	c8	6a																																																																																				
28	d7	07	94																																																																																				
52	85	e3	f6																																																																																				
a4	11	cf	50																																																																																				
c8	6a	2f	5e																																																																																				
94	28	d7	07																																																																																				
0f	60	6f	5e																																																																																				
dc	31	c0	b3																																																																																				
da	38	10	13																																																																																				
a9	bf	6b	01																																																																																				
ef	a8	b6	db																																																																																				
44	52	71	0b																																																																																				
a5	5b	25	ad																																																																																				
41	7f	3b	00																																																																																				

5	<table border="1"><tr><td>e0</td><td>c8</td><td>d9</td><td>85</td></tr><tr><td>92</td><td>63</td><td>b1</td><td>b8</td></tr><tr><td>1f</td><td>63</td><td>35</td><td>be</td></tr><tr><td>e8</td><td>c0</td><td>50</td><td>01</td></tr></table>	e0	c8	d9	85	92	63	b1	b8	1f	63	35	be	e8	c0	50	01	<table border="1"><tr><td>e1</td><td>e8</td><td>35</td><td>97</td></tr><tr><td>4f</td><td>fb</td><td>c8</td><td>6c</td></tr><tr><td>d2</td><td>fb</td><td>96</td><td>ae</td></tr><tr><td>9b</td><td>ba</td><td>53</td><td>7c</td></tr></table>	e1	e8	35	97	4f	fb	c8	6c	d2	fb	96	ae	9b	ba	53	7c	<table border="1"><tr><td>e1</td><td>e8</td><td>35</td><td>97</td></tr><tr><td>fb</td><td>c8</td><td>6c</td><td>4f</td></tr><tr><td>96</td><td>ae</td><td>d2</td><td>fb</td></tr><tr><td>7c</td><td>9b</td><td>ba</td><td>53</td></tr></table>	e1	e8	35	97	fb	c8	6c	4f	96	ae	d2	fb	7c	9b	ba	53	<table border="1"><tr><td>25</td><td>bd</td><td>b6</td><td>4c</td></tr><tr><td>d1</td><td>11</td><td>3a</td><td>4c</td></tr><tr><td>a9</td><td>d1</td><td>33</td><td>c0</td></tr><tr><td>ad</td><td>68</td><td>8e</td><td>b0</td></tr></table>	25	bd	b6	4c	d1	11	3a	4c	a9	d1	33	c0	ad	68	8e	b0	\oplus	<table border="1"><tr><td>d4</td><td>7c</td><td>ca</td><td>11</td></tr><tr><td>d1</td><td>83</td><td>f2</td><td>f9</td></tr><tr><td>c6</td><td>9d</td><td>b8</td><td>15</td></tr><tr><td>f8</td><td>87</td><td>bc</td><td>bc</td></tr></table>	d4	7c	ca	11	d1	83	f2	f9	c6	9d	b8	15	f8	87	bc	bc	=
	e0	c8	d9	85																																																																																			
	92	63	b1	b8																																																																																			
	1f	63	35	be																																																																																			
e8	c0	50	01																																																																																				
e1	e8	35	97																																																																																				
4f	fb	c8	6c																																																																																				
d2	fb	96	ae																																																																																				
9b	ba	53	7c																																																																																				
e1	e8	35	97																																																																																				
fb	c8	6c	4f																																																																																				
96	ae	d2	fb																																																																																				
7c	9b	ba	53																																																																																				
25	bd	b6	4c																																																																																				
d1	11	3a	4c																																																																																				
a9	d1	33	c0																																																																																				
ad	68	8e	b0																																																																																				
d4	7c	ca	11																																																																																				
d1	83	f2	f9																																																																																				
c6	9d	b8	15																																																																																				
f8	87	bc	bc																																																																																				

6	<table border="1"><tr><td>f1</td><td>c1</td><td>7c</td><td>5d</td></tr><tr><td>00</td><td>92</td><td>c8</td><td>b5</td></tr><tr><td>6f</td><td>4c</td><td>0b</td><td>d5</td></tr><tr><td>55</td><td>ef</td><td>32</td><td>0c</td></tr></table>	f1	c1	7c	5d	00	92	c8	b5	6f	4c	0b	d5	55	ef	32	0c	<table border="1"><tr><td>a1</td><td>78</td><td>10</td><td>4c</td></tr><tr><td>63</td><td>4f</td><td>e8</td><td>d5</td></tr><tr><td>a8</td><td>29</td><td>3c</td><td>03</td></tr><tr><td>fc</td><td>df</td><td>23</td><td>fe</td></tr></table>	a1	78	10	4c	63	4f	e8	d5	a8	29	3c	03	fc	df	23	fe	<table border="1"><tr><td>a1</td><td>78</td><td>10</td><td>4c</td></tr><tr><td>4f</td><td>e8</td><td>d5</td><td>63</td></tr><tr><td>3d</td><td>03</td><td>a8</td><td>29</td></tr><tr><td>fe</td><td>fc</td><td>df</td><td>23</td></tr></table>	a1	78	10	4c	4f	e8	d5	63	3d	03	a8	29	fe	fc	df	23	<table border="1"><tr><td>4b</td><td>2c</td><td>33</td><td>37</td></tr><tr><td>86</td><td>4a</td><td>9d</td><td>d2</td></tr><tr><td>8d</td><td>89</td><td>f4</td><td>18</td></tr><tr><td>6d</td><td>80</td><td>e8</td><td>d8</td></tr></table>	4b	2c	33	37	86	4a	9d	d2	8d	89	f4	18	6d	80	e8	d8	\oplus	<table border="1"><tr><td>6d</td><td>11</td><td>db</td><td>ca</td></tr><tr><td>88</td><td>0b</td><td>f9</td><td>00</td></tr><tr><td>a3</td><td>3e</td><td>86</td><td>93</td></tr><tr><td>7a</td><td>fd</td><td>41</td><td>fd</td></tr></table>	6d	11	db	ca	88	0b	f9	00	a3	3e	86	93	7a	fd	41	fd	=
	f1	c1	7c	5d																																																																																			
	00	92	c8	b5																																																																																			
	6f	4c	0b	d5																																																																																			
55	ef	32	0c																																																																																				
a1	78	10	4c																																																																																				
63	4f	e8	d5																																																																																				
a8	29	3c	03																																																																																				
fc	df	23	fe																																																																																				
a1	78	10	4c																																																																																				
4f	e8	d5	63																																																																																				
3d	03	a8	29																																																																																				
fe	fc	df	23																																																																																				
4b	2c	33	37																																																																																				
86	4a	9d	d2																																																																																				
8d	89	f4	18																																																																																				
6d	80	e8	d8																																																																																				
6d	11	db	ca																																																																																				
88	0b	f9	00																																																																																				
a3	3e	86	93																																																																																				
7a	fd	41	fd																																																																																				

7	<table border="1"><tr><td>26</td><td>3d</td><td>e8</td><td>fd</td></tr><tr><td>0e</td><td>41</td><td>64</td><td>d2</td></tr><tr><td>2e</td><td>b7</td><td>72</td><td>8b</td></tr><tr><td>17</td><td>7d</td><td>a9</td><td>25</td></tr></table>	26	3d	e8	fd	0e	41	64	d2	2e	b7	72	8b	17	7d	a9	25	<table border="1"><tr><td>f7</td><td>27</td><td>9b</td><td>54</td></tr><tr><td>ab</td><td>83</td><td>43</td><td>b5</td></tr><tr><td>31</td><td>a9</td><td>40</td><td>3d</td></tr><tr><td>f0</td><td>ff</td><td>d3</td><td>3f</td></tr></table>	f7	27	9b	54	ab	83	43	b5	31	a9	40	3d	f0	ff	d3	3f	<table border="1"><tr><td>f7</td><td>27</td><td>9b</td><td>54</td></tr><tr><td>83</td><td>43</td><td>b5</td><td>ab</td></tr><tr><td>40</td><td>3d</td><td>31</td><td>a9</td></tr><tr><td>3f</td><td>f0</td><td>ff</td><td>d3</td></tr></table>	f7	27	9b	54	83	43	b5	ab	40	3d	31	a9	3f	f0	ff	d3	<table border="1"><tr><td>14</td><td>46</td><td>27</td><td>34</td></tr><tr><td>15</td><td>16</td><td>46</td><td>2a</td></tr><tr><td>b5</td><td>15</td><td>56</td><td>d8</td></tr><tr><td>bf</td><td>ec</td><td>d7</td><td>43</td></tr></table>	14	46	27	34	15	16	46	2a	b5	15	56	d8	bf	ec	d7	43	\oplus	<table border="1"><tr><td>4e</td><td>5f</td><td>84</td><td>4e</td></tr><tr><td>54</td><td>5f</td><td>a6</td><td>a6</td></tr><tr><td>f7</td><td>c9</td><td>4f</td><td>dc</td></tr><tr><td>0e</td><td>f3</td><td>b2</td><td>4f</td></tr></table>	4e	5f	84	4e	54	5f	a6	a6	f7	c9	4f	dc	0e	f3	b2	4f	=
	26	3d	e8	fd																																																																																			
	0e	41	64	d2																																																																																			
	2e	b7	72	8b																																																																																			
17	7d	a9	25																																																																																				
f7	27	9b	54																																																																																				
ab	83	43	b5																																																																																				
31	a9	40	3d																																																																																				
f0	ff	d3	3f																																																																																				
f7	27	9b	54																																																																																				
83	43	b5	ab																																																																																				
40	3d	31	a9																																																																																				
3f	f0	ff	d3																																																																																				
14	46	27	34																																																																																				
15	16	46	2a																																																																																				
b5	15	56	d8																																																																																				
bf	ec	d7	43																																																																																				
4e	5f	84	4e																																																																																				
54	5f	a6	a6																																																																																				
f7	c9	4f	dc																																																																																				
0e	f3	b2	4f																																																																																				



D.7 Véc tơ kiểm tra Camellia

D.7.1 Mở đầu

Với đầu vào đã cho (bản rõ và khóa), đầu ra (bản mã và khóa con) được mô tả như sau:

D.7.2 Phép mã hóa Camellia

D.7.2.1 Khóa 128 bit

Bảng D.16 – Ví dụ của phép mã hóa Camellia với 128 bit khóa

TT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	$K =$ 0123456789abcdeffedcba9876543210	$K_A =$ ae71c3d55ba6bf1d169240a795f89256

TT	Lựa chọn	Đầu vào	Đầu ra
	Mã hóa	$P =$ 0123456789abcdeffedcba9876543210	$C =$ 67673138549669730857065648eabe4
2	Khóa	$K =$ 4149d2aded9456681ec8b511d9e7ee0	$K_A =$ c501214a4e3ebde87c7cb2849487e0ab
	Mã hóa	$P =$ 2a9b0b74f4c5dc6239b7063a50a7946e	$C =$ db93bb9c0add5ab59ed94d467a6277f8
3	Khóa	$K =$ 47e8fb063dd4fe4ab430a73af7720206	$K_A =$ 0e295b7d3f360780e68144421220764f
	Mã hóa	$P =$ 0f9d74fc31ca654f921a606c024e7084	$C =$ 147375f650037166ca66c010cda255a5
4	Khóa	$K =$ 40bc8981241954a60a942b4a4334d1d	$K_A =$ 02596e63aa22ce0b724a216c366fea38
	Mã hóa	$P =$ 98048dd5d98b1f8dbbc6c7b238c9b948	$C =$ 3804e8e37a934cd71490c04ac34fd01e
5	Khóa	$K =$ 3da93f2679decbb104422e07332f7e3fe	$K_A =$ 83b4f6108365a66dd87f05f25fe79d4b
	Mã hóa	$P =$ cca0f0f0ba4596c4d9c10e1cf5dff82e	$C =$ 3ac304208199aa72ccddc42f5e6c7972

D.7.2.2 Khóa 192 bit

Bảng D.17 – Ví dụ của phép mã hóa Camellia với 192 bit khóa

TT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	$K =$ 0123456789abcdeffedcba987654321 00011223344556677	$K_A =$ 0766a2135c44e288cf62016a06babad3 $K_B =$ 8f3afac1cc974396c098a0b7e38b4df2
	Mã hóa	$P =$ 0123456789abcdeffedcba987654321 0	$C =$ b4993401b3e996f84ee5cee7d79b09b9

TT	Lựa chọn	Đầu vào	Đầu ra
2	Khóa	$K =$ 5e89b44b505c09f156bf78055f78a83c 24bfc19edd5c94ef	$K_A =$ ba0b31b670b34c26026dff7b74564087 $K_B =$ 3c23619d33dd5fcbec712953eeda757f
	Mã hóa	$P =$ dcac1785791e9ef611c7c7fc3bcdfe7	$C =$ 1e3bfb9b9a673ccb73c2c7da81d6e12d
3	Khóa	$K =$ d3e748b043dc9f66388b7d50567cc6a a2f884f3e53e4a3dd	$K_A =$ 49667a715ec02be735943e7a4b4acc0c $K_B =$ 264d533063503300aadf49fe61b451ce
	Mã hóa	$P =$ 54b3c1a40fcb95658a0d6bea861326a	$C =$ d01df1a0f3c44431a7d48ecabc94b25e
4	Khóa	$K =$ 1e1fe47104884ef696166eb80390add 8fb53ef43986dc268b	$K_A =$ d976805e61b8b9ec6f7df42b42c11239 $K_B =$ 50c525b03a7575a6f061f5780eb98d91
	Mã hóa	$P =$ d46ba51747457e7fd0fbcf267796d046	$C =$ 5d2092ed17143e2f01d7c8e50965720b
5	Khóa	$K =$ 43e6b5db547743bd09d19d312f2477a d902e2f8334a70d4f	$K_A =$ b4a87a3eaf5a4b27645242cf6cfede61 $K_B =$ 83adb334e4bf72262adb5e18fde7873
	Mã hóa	$P =$ 7e6bb782f305788a1be6421 f76f0f772	$C =$ 9c7c8c7148b6fae78fba6576d6808e92

D.7.2.3 Khóa 256 bit

Bảng D.18 – Ví dụ của phép mã hóa Camellia với 256 bit khóa

TT	Lựa chọn	Đầu vào	Đầu ra
1	Khóa	$K =$ 0123456789abcdeffedcba98765432100 0112233445566778899aabbccddeeff	$K_A =$ 17d1b5b046df07fac9bb914b7f1937ee $K_B =$ 3815214280a4d3c01848fd9ac7b1fe60

TT	Lựa chọn	Đầu vào	Đầu ra
	Mã hóa	$P =$ 0123456789abcdeffedcba9876543210	$C =$ 9acc237dff16d76c20ef7c919e3a7509
2	Khóa	$K =$ c940117c2eda1d1eea32c009d3c85421 b330d6547f0d36e7aa6a2b1e6d584636	$K_A =$ 5ac47afed9b4b6393b932b20a4c3a3a8 $K_B =$ cd96baa6ebd53714ceb77399047b20cf
	Mã hóa	$P =$ 4deadcb5a14f37e2679c344437032d64	$C =$ bcf8a9903df05b7fc6961a6bddfc492e
3	Khóa	$K =$ a8cd7528daab0f84153a668392acb92a 036cf1343dd64f3f7c7415eaec0c0b95	$K_A =$ 1013cf0907021375c6a47660ca372337 $K_B =$ 860596fdbac808d0b66e385332bc805d
	Mã hóa	$P =$ 9857b3c731d0e51b02a524d66e78f721	$C =$ 5559e464cf71c284c2279a6bdd8fa71
4	Khóa	$K =$ e9b481268ad16606457bf03188fbc6617 b8315a64f4ee755ecaaed3727b08411	$K_A =$ 8dfd62ace469c4c31944c934f6d819cd $K_B =$ e7eb9972ca9768a2f513e0e48cd147cd
	Mã hóa	$P =$ d980bdb42bcc3840069ec3984a7dc24d	$C =$ 8a9cd33a905a24a38eb0b4fbf2e7d68f
5	Khóa	$K =$ 971803e766ea3c52942a89bcda0ecd3e 14042ceba22107bb07545ce8685e4400	$K_A =$ a02f5b9a72d9ce7c7085a59258d8c8ed $K_B =$ 208a93b9dec678c522574c7dc203bca1
	Mã hóa	$P =$ 640637eed79df51 c19e54da1 e114025c	$C =$ b01ea3099f64847f8b0ad264841c64bd

D.8 Véc tơ kiểm tra SEED

Với đầu vào đã cho (bản rõ và khóa), đầu ra (bản mã và khóa con) được mô tả với $K_A = K_{1,0} \parallel K_{1,1} \parallel K_{2,0} \parallel K_{2,1}$, $K_B = K_{3,0} \parallel K_{3,1} \parallel K_{4,0} \parallel K_{4,1}$, $K_C = K_{5,0} \parallel K_{5,1} \parallel K_{6,0} \parallel K_{6,1}$, $K_D = K_{7,0} \parallel K_{7,1} \parallel K_{8,0} \parallel K_{8,1}$, $K_E = K_{9,0} \parallel K_{9,1}$

$K_{9,1} \parallel K_{10,0} \parallel K_{10,1}$, $K_F = K_{11,0} \parallel K_{11,1} \parallel K_{12,0} \parallel K_{12,1}$, $K_G = K_{13,0} \parallel K_{13,1} \parallel K_{14,0} \parallel K_{14,1}$ và $K_H = K_{15,0} \parallel K_{15,1} \parallel K_{16,0} \parallel K_{16,1}$.

Bảng D.19 – Ví dụ của phép mã hóa SEED

TT		Đầu vào	Đầu ra
1	Khóa	$K =$ 00000000000000000000000000000000	$K_A =$ 7c8f8c7e c737a22c ff276cdb a7ca684a $K_B =$ 2f9d01a1 70049e41 ae59b3c4 4245e90c $K_C =$ a1d6400f dbc1394e 85963508 0c5f1fcb $K_D =$ b684bda7 61a4aeae d17e0741 fee90aa1 $K_E =$ 76cc05d5 e97a7394 50ac6f92 1b2666e5 $K_F =$ 65b7904a 8ec3a7b3 2f7e2e22 a2b121b9 $K_G =$ 4d0bfde4 4e888d9b 631c8ddc 4378aSc4 $K_H =$ 216af65f 7878c031 71891150 98b255b0
	Mã hóa	$P =$ 00102030405060708090a0b0c0d0e0f	$C =$ 5ebac6e0054e166819aff1cc6d346cdb
2	Khóa	$K =$ 00102030405060708090a0b0c0d0e0f	$K_A =$ c119f584 5ae033a0 62947390 a600ad14 $K_B =$ f6f6544e 59Sc4b49 c1a3de02 ce483c49 $K_C =$ 0e742e6d 7e25163d 8299d2b4 790a46ce $K_D =$ ea67d836 55f354f2 c47329fb f50db634 $K_E =$ 2bd30235 51679ce6 fa8d6b76 a9f37e02 $K_F =$

TT		Đầu vào	Đầu ra
			8b99cc60 0f6092d4 bdaefcfa 489c2242 $K_G =$ f6357c14 cfccb126 a0aa6d85 f8c10774 $K_H =$ 47f4fec5 353ae1ba feccea48 a4ef9f9b
	Mã hóa	$P =$ 00000000000000000000000000000000	$C =$ c11f22f20140505084483597e4370f43
3	Khóa	$K =$ 123456789abcdef0123456789abcdef	$K_A =$ e47177f4 c737a22c 18752299J f462b107 $K_B =$ adc60ed9 6b39f7eb 869e8beb 1d01ae42 $K_C =$ 752bfe02 028f9991 a9671853 d12ac07f $K_D =$ f9786eb9 7e4b890c 5d6c61c2 1e556102 $K_E =$ 0caee621 e97a7394 56d2e637 d8c704dc $K_F =$ 67d24428 ef9d3283 ebaac63f d1b343ca $K_G =$ ebaac63f ebaac63f 286ba527 06e94437 $K_H =$ c52ed389 889f6217 25644913 1e0ff430
	Mã hóa	$P =$ 123456789abcdef0123456789abcdef	$C =$ 504ec8814d4f85eb81ec4bd210111425
4	Khóa	$K =$ 00102030405060708090a0b0c0d0e0f	$K_A =$ c119f584 5ae033a0 62947390 a600ad14 $K_B =$ f6f6544e 596c4b49 c1a3de02 ce483c49 $K_C =$ 0e742e5d 7e25163d 8299d2b4 790a46ce $K_D =$ ea67d836 55f354f2 jc47329fb f50db634

TT		Đầu vào	Đầu ra
			$K_E =$ 2bd30235 51679ce6 fa8d6b76 a9f37e02 $K_F =$ 8b99cc60 0f6092d4 bdaefcfa 489c2242 $K_G =$ f6357c14 cfccb126 a0aa6d85 f8c10774 $K_H =$ 47f4fec5 353ae1ba feccea48 a4ef9f9b
	Mã hóa	$P =$ 00102030405060708090a0b0c0d0e0f	$C =$ a6e8d7325bbe0998cf235c1 b57e64360
5	Khóa	$K =$ 123456789abcdeffedcba9876543210	$K_A =$ 6d4e5dae 4e297782 23a38bb1 5c412958 $K_B =$ d12712dc 42ad6022 808af8c0 f57bf0a3 $K_C =$ ab3bfd1 7edaa2f6 1fa555f6 8595ff92 $K_D =$ 0096103f 49ef98a7 50984c6f 145ab8e9 $K_E =$ 6f85548d 77682f20 f6d0417e 58448635 $K_F =$ 5b0f85d6 833fa7e3 88115657 aad9eb87 $K_G =$ 6f6fb9a3 3116727f 4b3f1382 b0881d6c $K_H =$ 3fb1117a cee6c40f f281cbdf 29b27cad
	Mã hóa	$P =$ 123456789abcdeffedcba9876543210	$C =$ caf1d16d5ec079a21ea4066794222c2a

Phụ lục E
(Tham khảo)
Bảng đặc tính

TT	Tên thuật toán	Đặc tính	Các tiêu chuẩn
1	TDEA [4]	<ul style="list-style-type: none"> • Tốc độ xử lý phần mềm chậm 	<ul style="list-style-type: none"> • Chuẩn công nghiệp ngân hàng • ANSI X9.52, FIPS46-3 • Thuật toán chính phủ điện tử Nhật bản (CRYPTREC)-chỉ dùng 3-khóa
2	MISTY1 [11]	<ul style="list-style-type: none"> • Phần cứng giới hạn • Thuật toán đa nền tảng (Hiệu năng phần mềm tốt trên các nền tảng khác nhau) 	<ul style="list-style-type: none"> • Chuẩn công nghiệp di động(3GPP&GSM:KASUMI dựa trên MISTY1) • IETF RFC 2994 • Thuật toán chọn NESSIE • Thuật toán chính phủ điện tử Nhật bản (CRYPTREC)-chỉ dùng 3-khóa
3	CAST-128 [8]	<ul style="list-style-type: none"> • Thuật toán đa nền tảng (Hiệu năng phần mềm tốt trên các platform khác nhau) • [Hiệu năng phần cứng không được bên thứ ba đánh giá] 	<ul style="list-style-type: none"> • IETF RFC 2144(xem RFCs 2984,2451, và 2400 sử dụng trong S/MIME, Ipsec và OpenPGP) • Mật mã chính phủ điện tử Canada
4	HIGHT [7],[10]	<ul style="list-style-type: none"> • Mã hóa tốc độ cao với phần cứng giới hạn 	<ul style="list-style-type: none"> • Chuẩn công nghiệp Hàn quốc (TTAS.KO-12.0040)
5	AES [5]	<ul style="list-style-type: none"> • Phần cứng giới hạn • Thuật toán đa nền tảng (Hiệu năng phần mềm tốt trên các nền tảng khác nhau) 	<ul style="list-style-type: none"> • FIPS 197 • Thuật toán chọn NESIE • Thuật toán chính phủ điện tử Nhật bản (CRYPTREC)
6	Camellia [9]	<ul style="list-style-type: none"> • Phần cứng giới hạn • Thuật toán đa nền tảng (hiệu năng phần mềm tốt trên các nền tảng khác nhau) 	<ul style="list-style-type: none"> • IETF RFC 3657,3713 • Thuật toán chọn NESIE • Thuật toán chính phủ điện tử Nhật bản

		<ul style="list-style-type: none"> • Giao diện giống như AES 	(CRYPTREC)
7	SEED [6]	<ul style="list-style-type: none"> • Tốc độ xử lý phần mềm modem chậm • [Hiệu năng phần cứng không được đánh giá bởi bên thứ ba] 	<ul style="list-style-type: none"> • IETF RFC 4269 (xem RFCs 4010, 4162, 2451, và 4196 sử dụng trong S/MIME, TLS và IPsec) • Thuật toán công nghiệp Hàn quốc (TTAS.KO.12-0004) • Thuật toán chính phủ điện tử Hàn quốc

Thư mục tài liệu tham khảo

- [1] ANSI X9.52-1998: Triple Data Encryption Algorithm Modes of Operation, 1999
 - [2] FIPS197: Advanced Encryption Standard (AES), November 26, 2001
 - [3] ISO/IEC JTC 1/SC 27 Standing Document No. 12 (SD12) on the Assessment of Cryptographic Algorithms and Key-Lengths
 - [4] ISO/IEC 9594-8, *Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks*
 - [5] ISO/IEC 9834 (all parts,), *Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities*
 - [6] ISO/IEC 18033-1, *Information technology — Security techniques — Encryption algorithms — Part 1: General*
 - [7] TTAS.KO-12.0004: A 128-bit Block Encryption Algorithm Standard, Telecommunication Technology Association, 1999
 - [8] TTAS.KO-12.0040/R1: 64-bit Block Cipher MIGHT, Telecommunication Technology Association, 2009
 - [9] C. Adams, "Constructing Symmetric Ciphers Using the CAST Design Procedure", *Designs, Codes and Cryptography*, vol.12, no.3, pp.283-316, November 1997
 - [10] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima and T. Tokita, "The 128-Bit Block Cipher Camellia", *IEICE Transaction*, vol.E85-A, no.1, pp.11-24, January 2002
 - [11] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim and S. Chee, "HIGHT: A New Block Cipher Suitable for Low-Resource Device", *Proceedings of the 8th Cryptographic Hardware and Embedded Systems Workshop*, Lecture Notes in Computer Science 4249, Springer-Verlag, pp.46-59, 2006
 - [12] M. Matsui, "New Block Encryption Algorithm MISTY", *Proceedings of the 4th Fast Software Encryption Workshop*, Lecture Notes in Computer Science 1267, pp.54-68, 1997
-