บทที่ 2

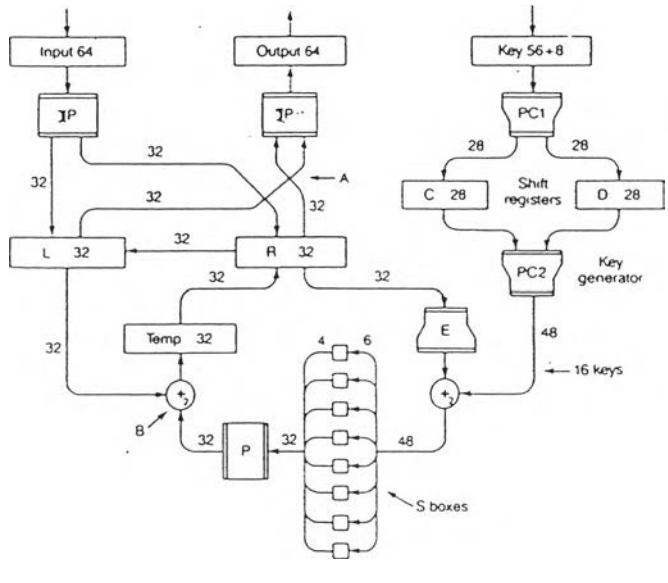
มาตรฐานการเข้ารหัสข้อมูล(Data Encryption Standard)

คำนำบท

ในบทนี้จะกล่าวถึงอัลกอริธึ่มและเทคนิคการเข้ารหัสตามมาตรฐาน ของ DES ที่ออกแบบโดย NBS โดยจะลงในรายละเอียดของทุกฟังก์ชันบล็อคที่ใช้ในอัลกอริธึ่ม

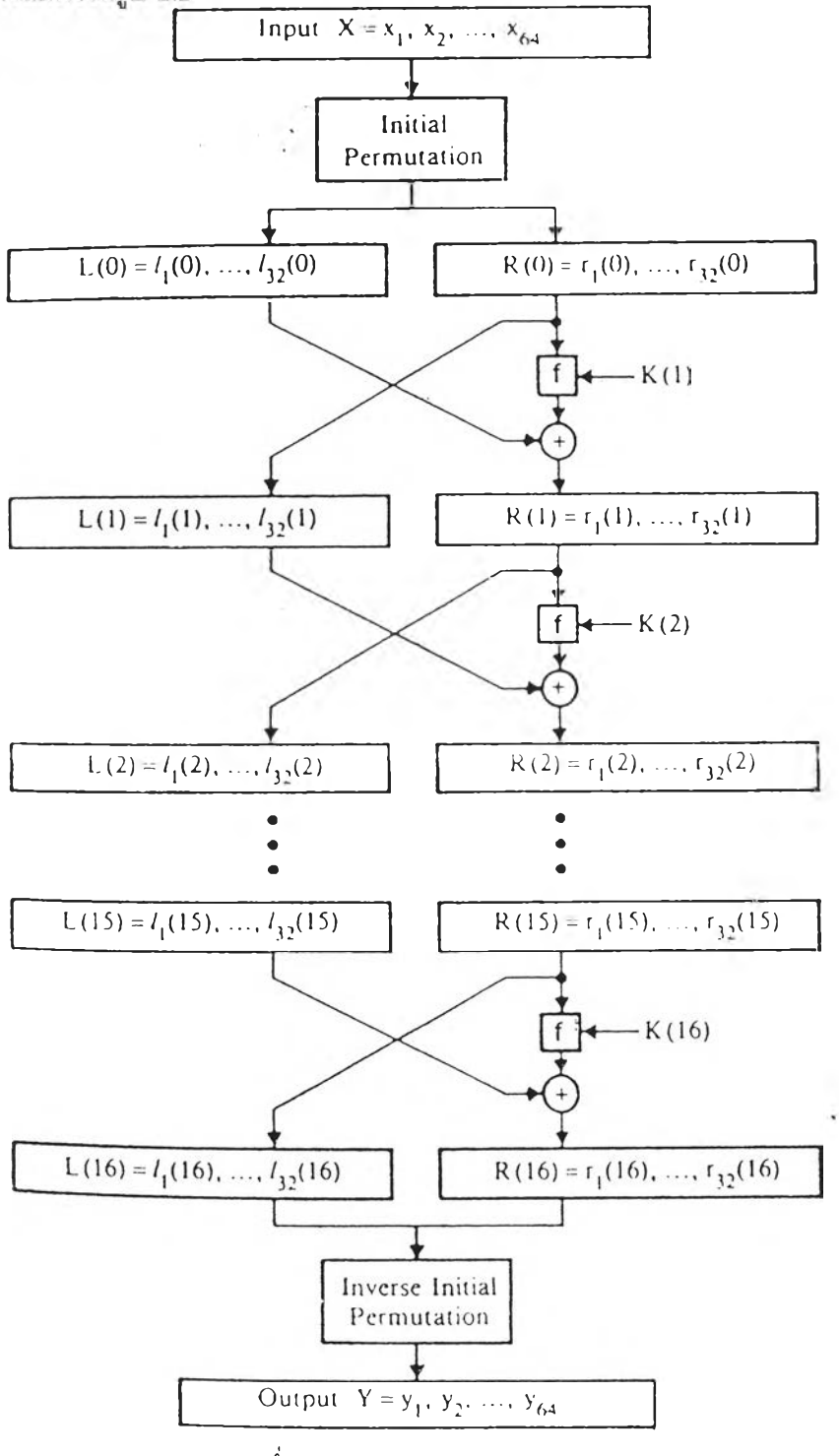
มาตรฐานการเข้ารหัสข้อมูลหรือดาต้าเอนคริปชันสแดนดาร์ด [Meyer and Matyas,1982]

มาตรฐานที่กำหนดอัลกอริธึ่มที่ใช้สำหรับเข้ารหัสข้อมูลที่ออกแบบโดย NBS เพื่อป้องกันความลับรั่วไหลโดยทั่วไปแล้วจะแบ่งลำดับชั้นของไซเฟอร์เท็กซ์ (Class of Cipher) ได้เป็น 3 ชั้นคือ ทรานส์โพซิชันไซเฟอร์ (Transposition Cipher) ซัพสทิทิวชันไซเฟอร์ (Substitution Cipher) และโพรดักท์ไซเฟอร์ (Product Cipher) ซึ่งเป็นการผสมผสานระหว่างทรานซ์โพซิชันไซเฟอร์และซัพสทิทิวชันไซเฟอร์อัลกอริธึ่มนี้เป็นโพรดัคท์ไซเฟอร์ที่จะแบ่งกลุ่มของข้อมูลอินพุท เป็นบล็อก ๆ ละ 64 บิทและใช้คีย์ขนาด 64 บิทเช่นกันการไหลผ่านของข้อมูลใน DES ที่จะแสดงได้ดังรูป 2.1



รูป 2.1 การไหลของข้อมูลในมาตรฐานการเข้ารหัสข้อมูล [Caelli,Longley and Shain,1989]

ข้อมูลที่ถูกป้อนเข้ามาทางด้านอินพุทจะถูกสลับตำแหน่งและแทนที่ด้วยข้อมูลใหม่ในPและS ตามลำดับและคีย์ก็จะถูกสลับตำแหน่งของบิทในPC1และPC2 และชิพรีจิสเตอร์ C และ Dการสลับตำแหน่งและการแทนที่ข้อมูลจะกระทำทั้งหมด 16 รอบโดยในแต่ละรอบคีย์ก็จะเปลี่ยนแปลงไปดังแสดงในรูป 2.2



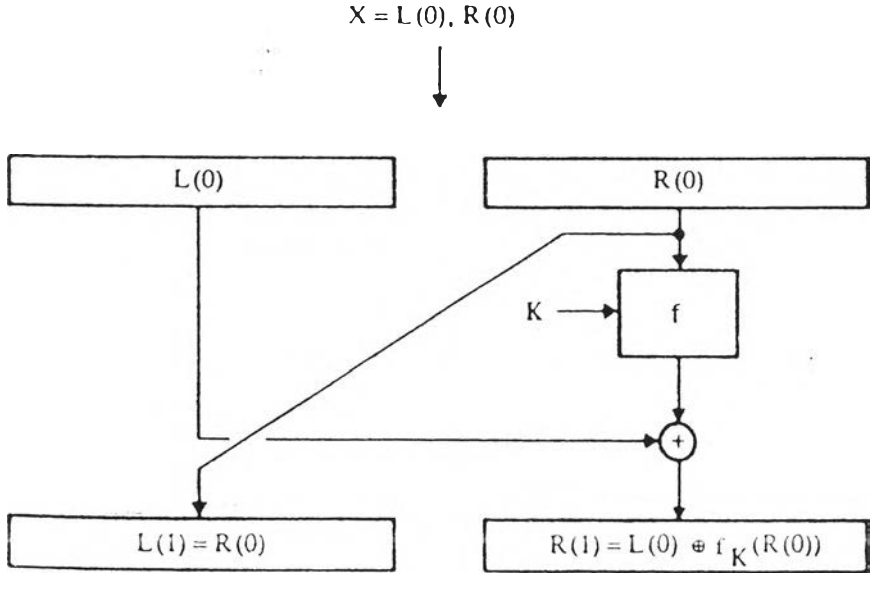
รูป 2.2 อัลกอริธึ่มของมาตรฐานการเข้ารหัสข้อมูล [Meyer and Matyas,1982]

ขั้นตอนการทำงานของ DES ในแต่ละรอบ

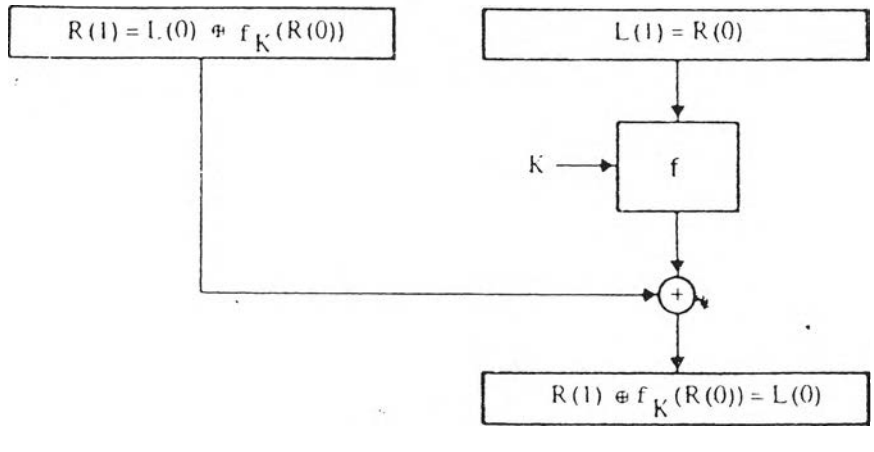
ขั้นตอนในการทำงานแต่ละรอบพอจะสรุปได้ดังนี้

1. บล็อกของข้อมูลอินพุทจะถูกแยกออกเป็น2ส่วนทางด้านซ้ายและด้านขวาส่วนละเท่าๆ กัน

2. ส่วนทางด้านขวาจะถูกป้อนเข้าสู่ไซเฟอร์ฟังก์ชัน f ดังรูป 2.3



รูป 2.3 การแปลงของบล็อคข้อมูลขาเข้า L(0) และ R(0) [Meyer and Matyas,1982]

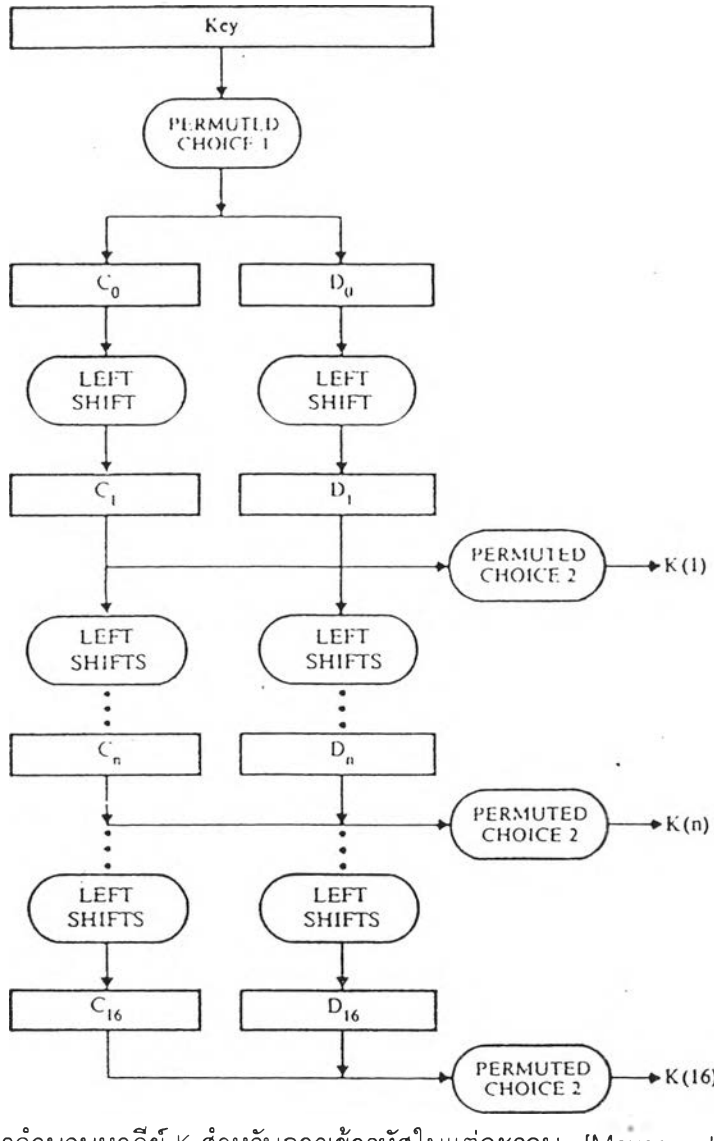


รูป 2. - การแปลงเพื่อให้ได้ L(0) [Meyer and Matyas,1982]

3. เอาท์พุทที่ได้จากไซเฟอร์ฟังก์ชันจะถูกเอ็กคลูซีพออร์กับส่วนที่อยู่ในด้านซ้ายในข้อ1โดยที่ฟังก์ชัน f จะต้องถูกออกแบบให้สามารถทำงานได้ไม่ว่าในกรณีของการเข้ารหัสหรือถอดรหัส เช่น ในกรณีการเข้ารหัส คีย์ในแต่ละรอบจะถูกเปลี่ยนแปลงไปจาก K(1) ถึง K(16) ในการถอดรหัสคีย์จะเป็น K(16) ถึง K(1) อัลกอริธึ่มนี้ก็ยังทำงานถูกต้อง

การกำเนิดคีย์ที่ใช้สำหรับแต่ละรอบของ DES

คีย์ K(1) ถึง K(16) นั้นจะมีขนาด 48 บิทและเพื่อความปลอดภัยของข้อมูลทั้ง 16 คีย์นี้จะต้องแตกต่างกันโดยจะเลือก 48 บิท จาก 56 บิทในแต่ละรอบของการทำงานซึ่งวิธีการเลือกคีย์ในแต่ละรอบแสดงอยู่ในรูป 2.5



รูป 2.5 การคำนวนหาคีย์ K สำหรับการเข้ารหัสในแต่ละรอบ [Meyer and Matyas,1982]

จากรูป2.5แสดงถึงการคำนวณหาคีย์ในแต่ละรอบของการเข้ารหัสสำหรับการถอดรหัสก็มีลักษณะเดียวกันแต่เป็นจากชิพทางซ้ายเป็นชิพไปทางขวายกเว้นการชิพระหว่าง (Co,Do) และ(C_1 , D_1) จะไม่มีการเลื่อน

การคำนวณหาค่าคีย์สำหรับการเข้ารหัสเริ่มจากการทำงานเริ่มต้นเลือก (Initialpermutation) โดย PC-1 (Permuted Choice-1) ซึ่ง PC-1 นี้จะเหมือนกันทั้งการเข้ารหัสและถอดรหัส โดยเลือก 56 บิทจาก 64 บิท (8 บิทที่ขาดไปคือแพริตี้บิท ได้แก่บิทที่8,16,24...64) แล้ว ส่งให้ชิพรีจิสเตอร์ C และ D ตัวละ 28 บิทในระหว่างการ เข้ารหัสข้อมูลในรีจิสเตอร์ C_{i-1}และD_{i-1} จะถูกชิพทางซ้าย 1 หรือ 2 บิทตาม ตารางที่ 2.1

ตาราง 2.1 แสดงการเลื่อนของข้อมูลที่ใช้ในการคำนวนหาค่าคีย์ของการเข้าและถอดรหัส

Iteration Number I	Number of left shifts (Encipherment)	Number of right shifts (Decipherment)
1	1	0
2	1	1
3	2	2
4	2	2
5	2	2
6	2	2
7	2	2
8	2	2
9	1	1
10	2	2
11	2	2
12	2	2
13	2	2
14	2	2
15	2	2
16	1	1

และเมื่อนำ C_i และ D_i มาผ่าน PC-2 ก็จะได้ K_i และเมื่อครบ 16 รอบแล้ว จะทำให้ $C_{16} = C_0$ และ $D_{16} = D_0$

สำหรับการถอดรหัสคีย์ที่จะถูกป้อนเข้าสู่อัลกอริธึ่ม คีย์แรกคือ K(16) ต่อมาคือ K(15) และลดลงตามลำดับ ดังนั้นในการคำนวณหาคีย์ก็จะทำตามรูปที่ 2.5 โดย K(16) จะถูก สร้างขึ้นมาโดยไม่มีการชิพในครั้งแรก (จาก C_0 , D_0 ไป C_1 , D_1) K(15)จะได้โดยการชิพ C_0 (C_{16}) และ D_0 (D_{16}) ไปทางขวา 1 บิทและคีย์ที่เหลือก็ชิพไปทางขวาตามตาราง 2.1

เพื่อให้เห็นค่าที่ชัดเจนจะแสดงรายละเอียดการสร้างคีย์ดังต่อไปนี้สมมติให้คีย์

$$K = k_1, k_2, k_3, \ldots, k_{64} \qquad (2.1)$$

เมื่อผ่าน PC-1 จะเหลือเพียง 56 บิต และถูกโหลดไปยังรีจิสเตอร์ขนาด 28 บิต 2 ชุดคือ C และ D ดังนี้

$$\begin{aligned} C_0 = {} & k_{57},k_{49},k_{41},k_{33},k_{25},k_{17},k_{09}, \\ & k_{01},k_{58},k_{50},k_{42},k_{34},k_{26},k_{18}, \\ & k_{10},k_{02},k_{59},k_{51},k_{43},k_{35},k_{27}, \\ & k_{19},k_{11},k_{03},k_{60},k_{52},k_{44},k_{36}, \end{aligned} \qquad (2.2)$$

$$\begin{aligned} D_0 = {} & k_{63},k_{55},k_{47},k_{39},k_{31},k_{23},k_{15}, \\ & k_{07},k_{62},k_{54},k_{46},k_{38},k_{30},k_{22}, \\ & k_{14},k_{06},k_{61},k_{53},k_{45},k_{37},k_{29}, \\ & k_{21},k_{13},k_{05},k_{28},k_{20},k_{12},k_{04} \end{aligned} \qquad (2.3)$$

จะเห็นว่า บิตที่ 57 , 49 และ 41 ของ k จะเป็นบิตแรก บิตที่ 2 และที่ 3 ของ รีจิสเตอร์ C_0 ตามลำดับในขณะที่ บิตที่ 63 , 55 และ 47 ของ k จะเป็นบิตแรก บิตที่ 2 และที่ 3 ของรีจิสเตอร์ D_0 ตามลำดับและจะสังเกตได้ว่า จาก 64 บิตในตอนแรกจะลดลงเหลือ 56 บิต เพราะ พาริตี้บิต k_8 , k_{16} , k_{24} , k_{32} , k_{40} , k_{56} และ k_{64} จะถูกขจัดออกไป เมื่อได้ C_0 และ D_0 แล้วจะได้ C_1 และ D_1 โดยการชิพท์ไปทางซ้าย (ในกรณีของการเข้ารหัส และทางขวาในกรณีของการถอดรหัส) การชิพท์ในที่นี้เป็นการชิพท์แบบวนรอบ คือ นำค่าที่ชิพท์ออกไปมาต่อท้ายตัวมันเอง ซึ่งจำนวนบิตที่ชิพท์ในแต่ละรอบ

แสดงไว้ในตาราง 2.1 และ 3.7.1 ดังนั้นค่าที่อยู่ในรีจิสเตอร์ C และ D ในรอบต่าง ๆ จะเป็นดัง ตาราง 2.2 และ 2.3

ตาราง 2.2 ตำแหน่งบิตของข้อมูลที่ใช้ในการคำนวนคีย์ที่ถูกเก็บไว้ในรีจิสเตอร์ C [Meyer and Matyas,1982]

Round (i)	Index of Elements in Vector C_i																												Round (i)
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
1	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	57	1
2	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	57	49	2
3	25	17	9	1	58	50	42	34	26	18	10	2	59	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	3
4	9	1	58	50	42	34	26	18	10	2	59	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	4
5	58	50	42	34	26	18	10	2	59	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	5
6	42	34	26	18	10	2	59	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	6
7	26	18	10	2	59	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	42	34	7
8	10	2	59	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	42	34	26	18	8
9	2	59	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	42	34	26	18	10	9
10	51	43	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	42	34	26	18	10	2	59	10
11	35	27	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	42	34	26	18	10	2	59	51	43	11
12	19	11	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	42	34	26	18	10	2	59	51	43	35	27	12
13	3	60	52	44	36	57	49	41	33	25	17	9	1	58	50	42	34	26	18	10	2	59	51	43	35	27	19	11	13
14	52	44	36	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	14
15	36	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	15
16	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	16

k_{29}, k_{41}, k_{11}, ..., etc. are the 1st, 2nd, 3rd, ..., etc. key bits in C_1, i.e., in register C during the 1st round.

ตาราง 2.3 ตำแหน่งบิตของข้อมูลที่ใช้ในการคำนวนคีย์ที่ถูกเก็บไว้ในรีจิสเตอร์ D [Meyer and Matyas,1982]

Round (i)	Index of Elements in Vector D_i																												Round (i)
	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	
1	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	63	1
2	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	63	55	2
3	31	23	15	7	62	54	46	38	30	22	14	6	61	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	3
4	15	7	62	54	46	38	30	22	14	6	61	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	4
5	62	54	46	38	30	22	14	6	61	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	5
6	46	38	30	22	14	6	61	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	6
7	30	22	14	6	61	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	46	38	7
8	14	6	61	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	46	38	30	22	8
9	6	61	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	46	38	30	22	14	9
10	53	45	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	46	38	30	22	14	6	61	10
11	37	29	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	46	38	30	22	14	6	61	53	45	11
12	21	13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	46	38	30	22	14	6	61	53	45	37	29	12
13	5	28	20	12	4	63	55	47	39	31	23	15	7	62	54	46	38	30	22	14	6	61	53	45	37	29	21	13	13
14	20	12	4	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	14
15	4	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	15
16	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	16

PC-2 จะเป็นตัวแสดงว่าคีย์ K(1), K(2) ,....,K(16) ที่มีขนาด 48 บิตจาก (C_1 , D_1) , (C_2 , D_2),...,(C_{16} , D_{16}) ได้มาจากการเลือกตำแหน่งของบิตในรีจิสเตอร์ C และ D โดยเลือกบิตตำแหน่งที่

14 , 17 , 11 , 24 , 1 , 5 , 3, 28 , 15 , 6 , 21 , 10

23 , 19, 12 , 4 , 26 , 8 , 16 , 7 , 27 , 20 , 13 , 2

จากรีจิสเตอร์ C และเลือก จากรีจิสเตอร์ D ในบิตตำแหน่งที่

41 , 52 , 31 , 37 , 47 , 55 , 30 , 40 , 51 , 45 , 33 , 48

44 , 49 , 39 , 56 , 34 , 53 , 46 , 42 , 50 , 36 , 29 , 32

24 บิตแรกของ K(i) ที่ได้มาจาก C_i แสดงอยู่ในตาราง 2.4 และ 24 บิตหลังของ K(i) ที่ได้มาจาก D_i แสดงไว้ในตาราง 2.5 ดังนั้นสามารถแสดงคีย์ต่าง ๆ ได้ดังนี้

K(1) = k_{10} , k_{51} ,..., k_{41} , k_{22} , k_{28} ,..., k_{31}

K(2) = k_2 , k_{43} ,..., k_{33} , k_{14} , k_{20} ,..., k_{23}

K(16) = k_{18} , k_{59} ,..., k_{49} , k_{30} , k_5 ,..., k_{39}

ตาราง 2.4 24 บิตแรกของคีย์ K(i) ในแต่ละรอบ [Meyer and Matyas,1982]

Round (i)	Index of Elements in Vector K (i)																						Round (i)
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
	Index of Selected Elements in Vector C_i (Obtained from PC-2)																								
1	10	51	34	60	49	17	33	57	2	9	19	42	3	35	26	25	44	58	59	1	36	27	18	41	1
2	2	43	26	52	34	6	25	49	59	1	11	34	60	27	18	17	36	50	51	58	57	19	10	33	2
3	26	8	10	36	25	58	9	33	43	50	60	18	44	11	2	1	49	34	35	42	41	3	59	17	3
4	34	11	59	39	9	43	58	17	27	34	44	2	57	60	51	50	33	18	19	26	25	52	43	1	4
5	18	60	4	33	58	26	42	7	17	36	57	51	41	44	35	34	17	2	3	10	9	36	27	50	5
6	7	44	27	3	42	11	26	50	60	2	41	35	25	57	19	18	1	51	52	59	58	49	11	34	6
7	28	5	19	1	26	59	10	15	33	51	25	19	9	41	3	2	50	35	36	43	42	33	60	9	7
8	36	41	60	50	10	63	59	23	53	32	9	3	58	25	52	51	34	19	49	27	26	17	44	2	8
9	17	33	52	42	1	55	51	10	49	27	1	60	50	17	44	43	26	11	41	19	18	9	36	59	9
10	31	17	36	26	5	19	35	55	32	11	59	44	34	1	57	27	10	60	25	3	2	58	49	43	10
11	25	1	49	10	33	7	19	42	23	60	34	57	18	50	41	11	59	44	9	52	51	42	33	23	11
12	9	50	33	59	19	52	3	25	13	44	18	49	1	34	55	60	43	57	58	36	35	26	17	7	12
13	58	34	17	43	3	36	52	14	0	5	2	25	51	18	9	44	27	41	42	49	19	10	1	60	13
14	47	18	1	57	32	49	36	60	34	41	51	9	35	2	58	57	11	25	26	33	13	59	50	44	14
15	15	2	50	11	36	63	49	42	18	23	35	58	19	51	42	41	60	9	10	17	52	43	34	55	15
16	12	59	42	25	57	20	41	26	50	17	27	50	11	43	34	33	52	1	2	9	44	35	26	39	16

ตาราง 2.5 24 บิตหลังของคีย์ K(i) ในแต่ละรอบ [Meyer and Matyas,1982]

Round (i)	Index of Elements in Vector K (i)																								Round (i)
	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	
	Index of Selected Element in Vector D_i (Obtained from PC-2)																								
	41	52	31	37	47	55	30	40	51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32	
1	22	28	39	54	37	4	47	30	5	53	23	29	61	21	38	63	15	20	45	14	13	62	55	31	1
2	14	20	31	46	29	63	39	22	28	45	15	21	53	13	30	55	7	12	37	6	5	54	47	23	2
3	61	4	15	30	13	47	23	6	12	29	62	5	37	28	14	39	54	63	21	53	20	38	31	7	3
4	45	55	62	14	28	31	7	53	63	13	46	20	21	12	61	23	38	47	5	37	4	22	15	54	4
5	29	39	46	61	12	15	54	37	47	28	30	4	5	63	45	7	22	31	20	21	55	6	62	38	5
6	13	23	30	45	63	62	38	21	31	12	14	55	20	47	29	54	6	15	4	5	39	53	46	22	6
7	28	7	14	29	47	46	22	5	15	63	61	39	4	31	13	38	53	62	55	20	23	37	30	6	7
8	12	54	61	13	31	30	6	20	62	47	45	23	55	15	28	22	37	46	39	4	7	21	14	53	8
9	4	46	53	5	23	22	61	12	54	39	37	15	47	7	20	14	29	38	31	63	62	13	6	45	9
10	55	30	37	20	7	6	45	63	38	23	21	62	31	54	4	61	13	22	15	47	46	28	53	29	10
11	39	14	21	4	54	53	29	47	22	7	5	46	15	38	55	45	28	6	62	31	30	12	37	13	11
12	23	61	5	55	38	37	13	31	6	54	20	30	62	22	39	29	12	53	46	15	14	63	21	28	12
13	7	45	20	39	22	21	28	15	53	38	4	14	46	6	23	13	63	37	30	62	61	47	5	12	13
14	54	29	4	23	6	5	12	62	37	22	55	61	30	53	7	28	47	21	14	46	45	31	20	63	14
15	38	13	55	7	53	20	63	46	21	6	39	45	14	37	54	12	31	5	61	30	29	15	4	47	15
16	30	5	47	62	45	12	55	38	13	61	31	37	6	29	46	4	23	28	53	22	21	7	63	39	16

<u>รายละเอียดของ อัลกอริธึ่ม DES</u> [Elizabeth and Denning,1982]

วิธีการเข้ารหัสโดยDESได้แสดงอยู่แล้วในรูปที่ 2.2 ในหัวข้อนี้จะอธิบายรายละเอียดของ Initial Permutation (IP) และฟังก์ชัน f โดยที่จะกล่าวถึง IP ก่อน

สมมติว่าข้อมูลขนาด 64 บิตที่จะถูกเข้ารหัส คือ

$$X = x_1 , x_2 , \ldots , x_{64}$$

เมื่อผ่าน IP แล้ว จะได้ผลดังต่อไปนี้

$$\begin{aligned} L(0) = {} & x_{58},x_{50},x_{42},x_{34},x_{26},x_{28},x_{10},x_{02}, \\ & x_{60},x_{52},x_{44},x_{36},x_{28},x_{20},x_{12},x_{04}, \\ & x_{62},x_{54},x_{46},x_{38},x_{30},x_{22},x_{14},x_{06}, \\ & x_{64},x_{56},x_{48},x_{40},x_{32},x_{24},x_{16},x_{08} \qquad (2.5) \\ R(0) = {} & x_{57},x_{49},x_{41},x_{33},x_{25},x_{17},x_{09},x_{01}, \\ & x_{59},x_{51},x_{43},x_{35},x_{27},x_{19},x_{11},x_{03}, \\ & x_{61},x_{53},x_{45},x_{37},x_{29},x_{21},x_{13},x_{05}, \\ & x_{63},x_{55},x_{47},x_{39},x_{31},x_{23},x_{15},x_{07} \qquad (2.6) \end{aligned}$$

L(0) และ R(0) จะถูกใช้ร่วมกับคีย์ K(1) ถึง K(16) ตามรูป 2.2 เพื่อสร้างให้เกิด L(16) และ R(16)

$$L(16),R(16) = l_1(16),l_2(16),\ldots,l_{32}(16),$$
$$r1(16),r2(16),\ldots,r32(16) \qquad (2.7)$$

ต่อจากนั้น L(16) และ R(16) จะถูกป้อนเข้าสู่ Inverse Initial Permutation(IP^{-1}) จะทำให้ได้เอาท์พุท

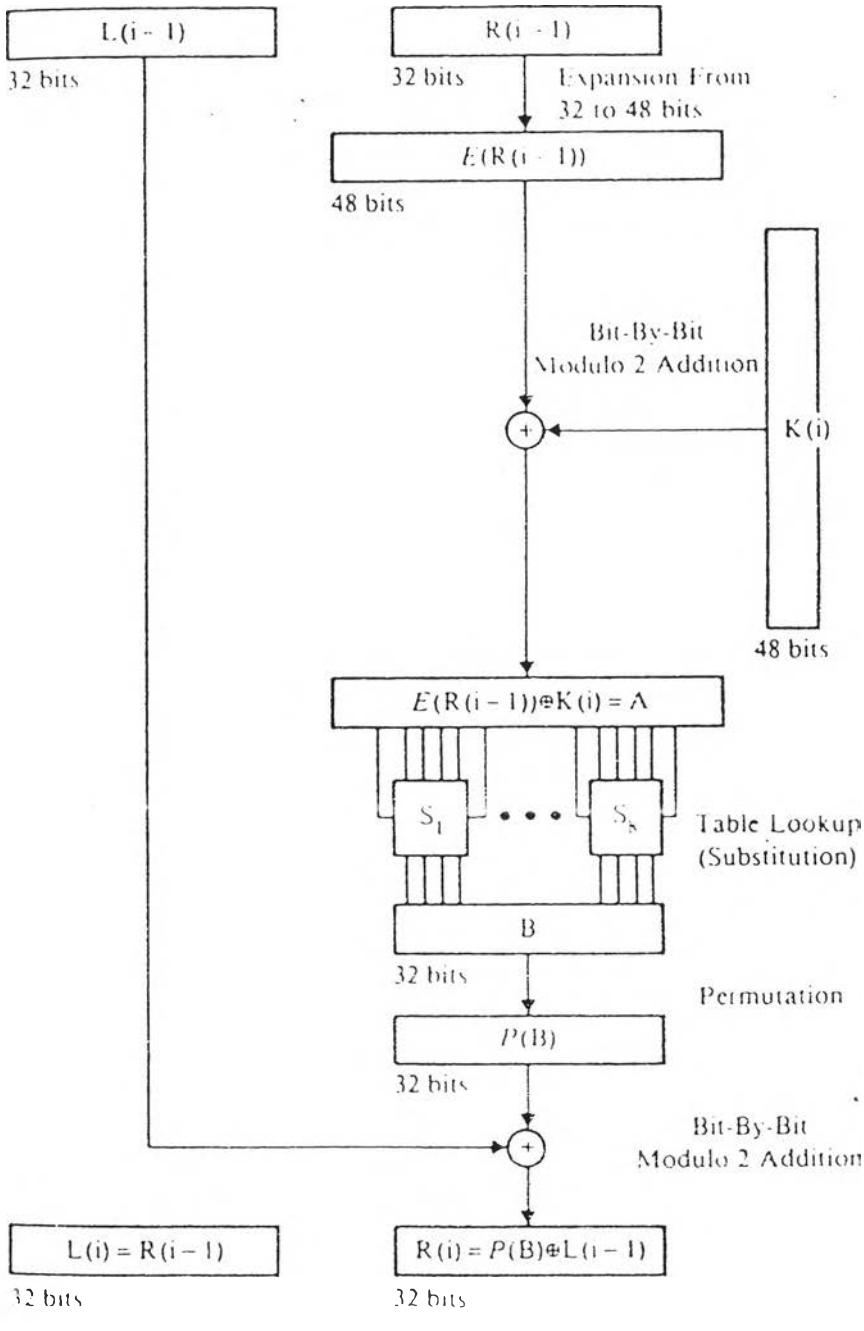
$$\begin{aligned} Y &= y_{01},y_{02},\ldots,y_{64} \\ &= l_{08},r_{08},l_{16},r_{16},l_{24},r_{24},l_{32},r_{32}, \\ &\quad l_{07},r_{07},l_{15},r_{15},l_{23},r_{23},l_{31},r_{31}, \\ &\quad l_{06},r_{06},l_{14},r_{14},l_{22},r_{22},l_{30},r_{30}, \\ &\quad l_{05},r_{05},l_{13},r_{13},l_{21},r_{21},l_{29},r_{29}, \\ &\quad l_{04},r_{04},l_{12},r_{12},l_{20},r_{20},l_{28},r_{28}, \\ &\quad l_{03},r_{03},l_{11},r_{11},l_{19},r_{19},l_{27},r_{27}, \\ &\quad l_{02},r_{02},l_{10},r_{10},l_{18},r_{18},l_{26},r_{26}, \\ &\quad l_{01},r_{01},l_{09},r_{09},l_{17},r_{17},l_{25},r_{25} \end{aligned} \qquad (2.8)$$

สำหรับรายละเอียดของฟังก์ชัน f ได้แสดงไว้ในรูปที่ 2.6 จากรูปข้อมูลทางด้านขวาของอินพุทรอบที่ i คือ

$$R(i-1) = r1(i-1),r1(i-1),\ldots,r32(i-1)$$

จะถูกขยายจาก 32 บิต เป็น 48 บิต ทำให้ได้

$$\begin{aligned} E(R(i-1)) &= r_{32},r_{01},r_{02},r_{03},r_{04},r_{05}, \\ &\quad r_{04},r_{05},r_{06},r_{07},r_{08},r_{09}, \\ &\quad r_{08},r_{09},r_{10},r_{11},r_{12},r_{13}, \\ &\quad r_{12},r_{13},r_{14},r_{15},r_{16},r_{17}, \\ &\quad r_{16},r_{17},r_{18},r_{19},r_{20},r_{21}, \\ &\quad r_{20},r_{21},r_{22},r_{23},r_{24},r_{25}, \\ &\quad r_{24},r_{25},r_{26},r_{27},r_{28},r_{29}, \\ &\quad r_{28},r_{29},r_{30},r_{31},r_{32},r_{01} \end{aligned} \qquad (2.9)$$



รูป 2.6 รายละเอียดของฟังก์ชัน f [Meyer and Matyas,1982]

ตาราง 2.6 รายละเอียดใน S - BOX [Caelli et al.,1989]

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
	1	00	15	07	04	14	02	13	01	10	06	02	11	09	05	03	08
	2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
	3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13
S2	0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
	1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
	2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
	3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09
S3	0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
	1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
	2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
	3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12
S4	0	07	13	14	03	00	06	09	10	01	02	08	05	11	12	04	15
	1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
	2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
	3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14
S5	0	02	12	04	01	07	10	11	06	08	05	03	15	13	00	14	09
	1	14	11	02	12	04	07	13	01	05	00	15	00	03	09	08	06
	2	04	02	01	11	10	13	07	08	15	09	12	05	06	03	00	14
	3	11	08	12	07	01	14	02	13	06	15	00	09	10	04	05	03
S6	0	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
	1	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
	2	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
	3	04	03	02	12	09	05	15	10	11	14	01	07	06	00	08	13
S7	0	04	11	02	14	15	00	08	13	03	12	09	07	05	10	06	01
	1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
	2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
	3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12
S8	0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
	1	01	15	13	08	10	03	07	04	12	05	06	11	00	14	09	02
	2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
	3	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

จุดประสงค์สำหรับการขยายจาก 32 บิตเป็น 48 บิตคือเพื่อใช้แต่ละบิตของไซเฟอร์ เท็กซ์ขึ้นอยู่กับ (ไวต่อการเปลี่ยนแปลง) เพลนเท็กซ์และคีย์มากที่สุดเมื่อ E(R(i-1)) ถูกสร้างขึ้น และ ถูกบวกแบบโมดูโล-2 เข้ากับคีย์ K(i) ผลที่ได้คือ เวกเตอร์ A ที่มีขนาด 48 บิต

$$A = ER(i-1) \oplus K(i)$$
$$= a_1 , a_2 , ..., a_{48} \qquad (2.10)$$

จากรูป 2.6 เวกเตอร์ A จะถูกใช้เป็นอาร์กิวเมนต์ในซับสติติวชันโอเปอเรชัน(Substitution Operation) หรือ S-Box จาก S_1 ถึง S_8

รายละเอียดของS-Boxได้แสดงไว้ในตาราง 2.6 โดยแต่ละ Box จะประกอบด้วย 4 แถว (00,01,10 และ 11) 16 คอลัมน์ (0000,0001,...,1111) ซึ่งจะทำให้มีตัวประกอบย่อยทั้งหมด 64 ตัว S-Box จะทำให้การลดขนาดของ A จาก 48 เป็น 32 บิตโดยมีวิธีการคือจะแบ่ง A ออกเป็นกลุ่ม กลุ่มละ 6 บิต ดังนั้นจะได้ข้อมูลทั้งหมด 8 กลุ่ม ในแต่ละกลุ่มบิตทางซ้ายสุด กับทางขวาสุด จะเป็นตัวกำหนดคอลัมน์ส่วนที่เหลือจะเป็นตัวกำหนดแถวใน S-Box ดังนั้น ใน S-Box ที่ 1 จะมีอินพุต คือ a_1 , a_2, a_3 , a_4, a_5 , a_6 โดยจะได้เอาท์พุต จาก S-Box คือ

$$S_1^{a1,a6}(a_2,a_3,a_4,a_5)$$

ซึ่งจะมีขนาด 4 บิตสมมติว่าคือ b_1 , b_2, b_3 และ b_4 ดังนั้น 4 บิตสุดท้ายที่ได้จาก S-Box ที่ 8คือ

$$S_8^{a43,a48}(a_{44},a_{45},a_{46},a_{47})$$

หรือ b_{29} , b_{30}, b_{31} และ b_{32} เอาท์พุตทั้งหมดที่ได้จาก S_1 ถึง S_8 คือ เวคเตอร์ B ตามรูปที่ 2.6และโดยการทำการสลับตำแหน่ง (Permutation) ของค่าในเวคเตอร์ B ตามสมการที่ 2.11

$$P(B) = b_{16},b_7,b_{20},b_{21},b_{29},b_{12},b_{28},b_{17},$$
$$b_1,b_{15},b_{23},b_{26},b_5,b_{18},b_{31},b_{10},$$
$$b_2,b_8,b_{24},b_{14},b_{32},b_{27},b_3,b_9,$$
$$b_{19},,b_{23},b_{30},b_6,b_{22},b_{11},b_4,b_{25} \qquad (2.11)$$

เอาท์พุตที่ได้คือ

$$L(i), R(i) = R(i-1) , P(B) \oplus L(i-1)$$

สรุปขั้นตอนในการทำ DES

1. กำหนดคีย์ ขนาด 64 บิต k_1 , k_2 ,..., k_{64}

2. จาก k_1 , k_2 ,..., k_{64} จะได้คีย์เวกเตอร์ 16 คีย์ คือ K(1) ถึง K(16) ซึ่งแต่ละคีย์มีขนาด 48 บิต ซึ่งใช้สำหรับการเข้ารหัสรอบที่ 1 ถึง 16 ตามลำดับ

3. รับเพลนเท็กซ์ขนาด 64 บิต x_1, x_2,..., x_{64} เข้ามา

4. จาก x_1, x_2,..., x_{64} สร้างเวกเตอร์ขนาด 32 บิต L(0) และ R(0) ตาม สมการที่ 2.5 และ 2.6

5. เซ็ทค่าตัวนับรอบของการเข้ารหัสเป็น i = 1

6. สร้างER(i-1)จาก R(i-1) โดยการใช้ฟังก์ชันในการขยาย E ที่กำหนดโดยสมการ 2.9

7. ใช้ K(i) ถ้าเป็นการเข้ารหัส แต่ถ้าเป็นการถอดรหัสใช้ K(17-i)

8. นำผลที่ได้จากขั้นที่ 6 และ 7 บวกกันแบบโมดูโล-2 และกำหนดให้ผลลัพธ์ที่ได้เป็นเวกเตอร์ A ที่มีขนาด 48 บิต A = a_1, a_2,..., a_{48}

9. สร้าง $S_1^{a1,a6}$ (a_2,a_3,a_4,a_5) ขึ้นมา และกำหนดผลที่ได้เป็น b_1,b_2,b_3,b_4 ทำซ้ำขั้นตอนนี้ จาก $S_2^{a1,a6}$ (a_8,a_9,a_{10},a_{11}) ถึง $S_8^{a43,a48}$ (a_{44},a_{45},a_{46},a_{47})โดยจะได้ผลลัพธ์ที่ได้ออกมาจะเป็นเวกเตอร์ B ขนาด 32 บิต คือ b_1,b_2,..,b_{32}

10. หาค่า P(B) ตามสมการ 2.11

11. นำ P(B) บวกแบบโมดูโล-2 กับ L(i-1) และให้ผลลัพธ์ที่ได้เป็น R(i)

12. กำหนด L(i) = R(i-1)

13. เพิ่มค่าตัวนับจำนวนรอบ i ขึ้น 1

14. ถ้าตัวนับจำนวนรอบเท่ากับ 16 หรือน้อยกว่า ให้ทำซ้ำขั้นที่ 6 ถึงขั้นที่ 14 ถ้าไม่ให้หาค่าเอาท์พุทสุดท้ายจาก L(16) และ R(16) ตามสมการ 2.7 และ 2.8

จะยกตัวอย่างที่แสดงให้เห็นถึงการเข้ารหัสใน 1 รอบ (ของทั้งหมด 16 รอบ) เพื่อให้ได้มาซึ่ง L(1) และ R(1) ดังนี้ โดยสมมติให้ X และ K เป็นเลขฐานสิบหก

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

หรือถ้าเป็นเลขฐานสองก็จะได้

X = K = 0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

ดังนั้นจากตารางที่ 2.4 และ 2.5 จะได้ว่า

K(1) = 0000 1011 0000 0010 0110 0111

1001 1011 0100 1001 1010 0101 (ฐานสอง)

= 0 B 0 2 6 7 9 B 4 9 A 5 (ฐานสิบหก)

จากสมการที่ 2.5 และ 2.6 จะได้ว่า

L(0) = 1100 1100 0000 0000 1100 1100 1111 1111

= C C 0 0 C C F F

และ

R(0) = 1111 0000 1010 1010 1111 0000 1010 1010

= F 0 A A F 0 A A

ต่อจากนี้ก็จะขยาย R(0) บวกแบบโมดูโล-2 เข้ากับ K(1) ทำให้ได้

$A = E\,R(0) \oplus K(1)$

= 011100 010001 011100 110010

111000 010101 110011 110000

จากข้อมูลทั้ง 48 บิตนี้ นำไปลดขนาดให้เหลือ 32 บิต โดยผ่านกระบวนการที่อธิบาย แล้วข้างต้นจะได้

$S_1^{00}(1110) = S_1^{0}(14) = 0$ (ฐานสิบ) = 0000 (ฐานสอง)

$S_2^{01}(1000) = S_2^{1}(8) = 12 = 1100$

$S_3^{00}(1110) = S_3^{0}(14) = 2 = 0010$

$S_4^{10}(1001) = S_4^{2}(9) = 1 = 0001$

$S_5^{10}(1100) = S_5^{2}(12) = 6 = 0110$

$S_6^{01}(1010) = S_6^{1}(10) = 13 = 1101$

$S_7^{11}(1001) = S_7^{3}(9) = 5 = 0101$

$S_8^{10}(1000) = S_8^{2}(8) = 0 = 0000$

นำค่า S_1 - S_8 มาเรียงลำดับและให้เท่ากับ B จะได้

B = 0000 1100 0010 0001 0110 1101 0101 0000

= 0 C 2 1 6 D 5 0

นำค่า B ไปผ่านสมการสลับตำแหน่งตามสมการ 2.11

P(B) = 1001 0010 0001 1100 0010 0000 1001 1100

= 9 2 1 C 2 0 9 C

นำ P(B) บวกแบบโมดูโล-2 เข้ากับ L(0) เพื่อให้ได้ R(1)

R(1) = 0100 1110 0001 1100 1110 1100 0110 0011

= 5 E 1 C E C 6 3

และสุดท้าย ข้อมูลทางด้านครึ่งซ้ายหรือ L(1) จะมีค่าเท่ากับ R(0) ดังนั้น

L(1) = R(0) = F 0 A A F 0 A A

ซึ่งเมื่อได้ L(1) และ R(1) ก็จะเป็นการสิ้นสุดกระบวนการใน 1 รอบ และใช้ L(1) และ R(1) นี้เป็นข้อมูลในรอบต่อไปจนครบ 16 รอบ