实验原理

一．对称密钥加密机制

对称密钥加密机制即对称密码体系，也称为单钥密码体系和传统密码体系。对称密码体系通常分为两大类，一类是分组密码（如DES、AES算法），另一类是序列密码（如RC4算法）。

对称密码体系加密和解密时所用的密钥是相同的或者是类似的，即由加密密钥可以很容易地推导出解密密钥，反之亦然。同时在一个密码系统中，我们不能假定加密算法和解密算法是保密的，因此密钥必须保密。发送信息的通道往往是不可靠的或者不安全的，所以在对称密码系统中，必须用不同于发送信息的另外一个安全信道来发送密钥。图6-1-1描述了对称密码(传统密码)系统原理框架，其中M表示明文；C表示密文；E表示加密算法；D表示解密算法；K表示密钥；I表示密码分析员进行密码分析时掌握的相关信息；B表示密码分析员对明文M的分析和猜测。

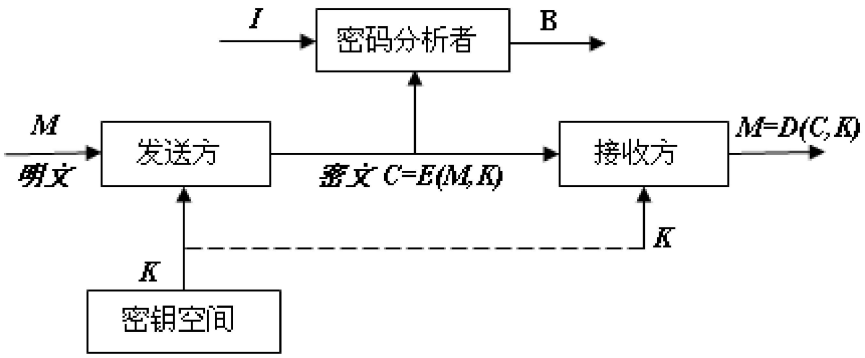

图6-1-1　传统密码系统原理框架图

对称密码体系的优点：

- 加密效率高，硬件实现可达每秒数百兆字节（软件实现略慢一些）。
- 密钥相对比较短。
- 可以用来构造各种密码机制。
- 可以用来建造安全性更强的密码。

对称密码体系的缺点：

- 通信双方都要保持密钥的秘密性。
- 在大型网络中，每个人需持有许多密钥。
- 为了安全，需要经常更换密钥。

二．DES加密算法简介

1973年5月15日，美国国家标准局在联邦注册报上发表一则启事，公开征集用来保护传输和静止存储的计算机数据的密码算法，这一举措最终导致了数据加密标准DES的出现。DES采用分组乘积密码体制，它是由IBM开发的，是对早期Lucifer密码体制的改进。DES在1975年3月17日首次在联邦记录中公布，而且声明对此算法征求意见。到1977年2月15日拟议中的DES被采纳为“非密级”应用的一个联邦标准。

最初预期DES作为一个标准只能使用10至15年。然而，出于种种原因，可能是DES还没有受到严重的威胁，事实证明了DES要长寿得多。在其被采用后，大约每隔5年被评审一次。DES的最后一次评审是在1999年1月。但是，随着计算机计算能力的提高，由于DES的密钥过短，仅有56位，对DES的成功攻击也屡见报导。例如：1999年1月，RSA数据安全公司宣布：该公司所发起的对56位DES的攻击已经由一个称为电子边境基金的组织，通过互联网上的100000台计算机合作在22小时15分钟内完成。

NIST（美国国家标准研究所）于1997年发布公告征集新的数据加密标准作为联邦信息处理标准以代替

DES。新的数据加密标准称为AES。尽管如此，DES的出现仍然是现代密码学历史上一个非常重要的事件。它对于我们分析掌握分组密码的基本理论与设计原理仍然具有重要的意义。

三．DES加密流程 动画演示

如图6-1-2所示，对于任意长度的明文，DES首先对其进行分组，使得每一组的长度为64位，然后分别对每个64位的明文分组进行加密。

对于每个64位长度的明文分组的加密过程如下：

（1）初始置换：输入分组按照初始置换表重排次序，进行初始置换。

（2）16轮循环：DES对经过初始置换的64位明文进行16轮类似的子加密过程。每一轮的子加密过程要经过DES的f函数，其过程如下：

- 将64位明文在中间分开，划分为2部分，每部分32位，左半部分记为L，右半部分记为R，以下的操作都是对右半部分数据进行的。
- 扩展置换：扩展置换将32位的输入数据根据扩展置换表扩展成为48位的输出数据。
- 异或运算：将48位的明文数据与48位的子密钥进行异或运算（48位子密钥的产生过程在实验原理八．子密钥产生过程中有详细讨论）。
- S盒置换：S盒置换是非线性的，48位输入数据根据S盒置换表置换成为32位输出数据。
- 直接置换：S盒置换后的32位输出数据根据直接置换表进行直接置换。
- 经过直接置换的32位输出数据与本轮的L部分进行异或操作，结果作为下一轮子加密过程的R部分。本轮的R部分直接作为下一轮子加密过程的L部分。然后进入下一轮子加密过程，直到16轮全部完成。

（3）终结置换：按照终结置换表进行终结置换，64位输出就是密文。

在每一轮的子加密过程中，48位的明文数据要与48位的子密钥进行异或运算，子密钥的产生过程如下：

- 循环左移：根据循环左移表对C和D进行循环左移。循环左移后的C和D部分作为下一轮子密钥的输入数据，直到16轮全部完成。
- 将C和D部分合并成为56位的数据。
- 压缩型换位2：56位的输入数据根据压缩型换位2表输出48位的子密钥，这48位的子密钥将与48位的明文数据进行异或操作。

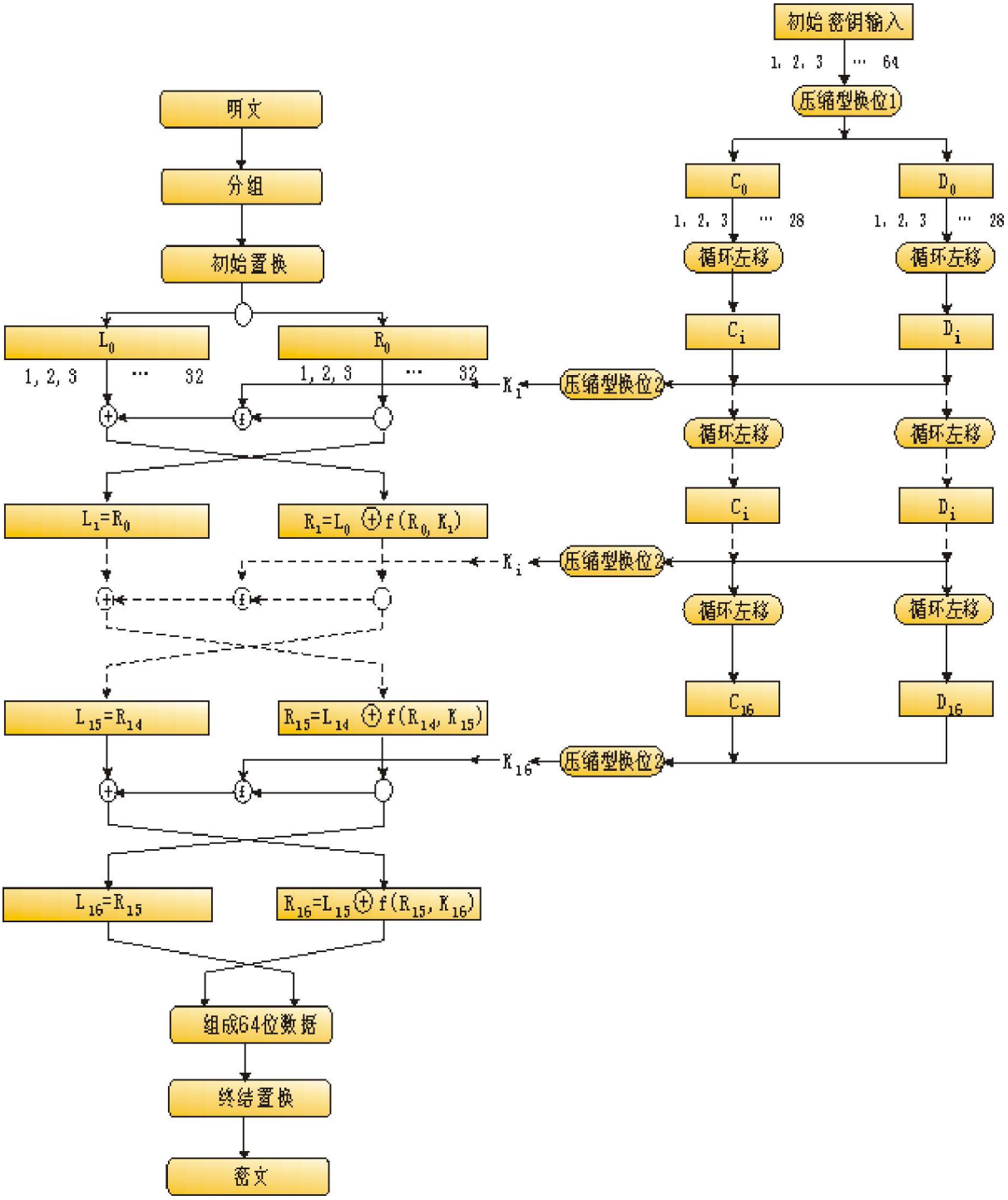

图6-1-2　DES加密流程

四．DES的分组过程

DES是一种分组加密算法，所谓分组加密算法就是对一定大小的明文或密文来做加密或解密动作。在DES加密系统中，每次加密或解密的分组大小均为64位，所以DES没有密文扩充的问题。对大于64位的明文只要按每64位一组进行切割，而对小于64位的明文只要在后面补“0”即可。

另一方面，DES所用的加密或解密密钥也是64位大小，但因其中有8个位是奇偶校验位，所以64位中真正起密钥作用的只有56位，密钥过短也是DES最大的缺点。

DES加密与解密所用的算法除了子密钥的顺序不同外，其他部分完全相同。

五．初始置换

经过分组后的64位明文分组将按照初始置换表重新排列次序，进行初始置换，置换方法如下：初始置换表从左到右，从上到下读取，如第一行第一列为58，意味着将原明文分组的第58位置换到第1位，初始置换表的下一个数为50，意味着将原明文分组的第50位置换到第2位，依次类推，将原明文分组的64位全部置换完成。

表6-1-1　初始置换表

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

六．16轮循环

经过了初始置换的64位明文数据在中间分成2部分，每部分32位，左半部分和右半部分分别记为L0和R0。然后，L0和R0进入第一轮子加密过程。R0经过一系列的置换得到32位输出，再与L0进行异或（XOR）运算。其结果成为下一轮的R1，R0则成为下一轮的L1，如此连续运作16轮。我们可以用下列两个式子来表示其运算过程：

Ri = Li-1 XOR f(Ri-1， Ki)

Li = Ri-1(i = 1， 2， ……， 16)

16轮循环过程如图6-1-3所示。

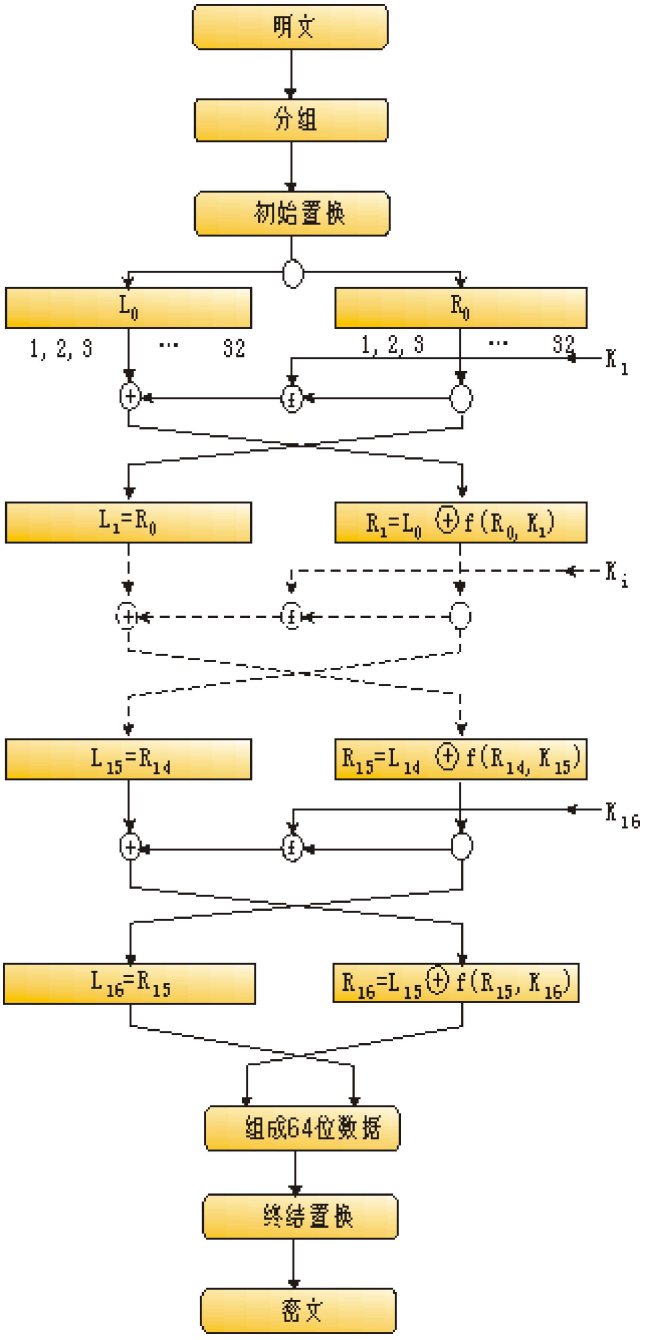

图6-1-3 DES 16轮循环

在每一轮的循环中，右半部分需要经过一系列的子加密过程，这个子加密过程也叫做f函数，子加密过程包括扩展置换、异或运算、S盒置换和直接置换，下面分别介绍这些过程。

1．扩展置换

32位的右半部分明文数据首先要进行扩展置换，扩展置换将32位的输入数据扩展成为48位的输出数据，它有三个目的：第一，它产生了与子密钥同长度的数据以进行异或运算；第二，它提供了更长的结果，使得在以后的子加密过程中能进行压缩；第三，它产生雪崩效应（avalanche effect），这也是扩展置换最主要的目的，使得输入的一位将影响两个替换，所以输出对输入的依赖性将传播的更快（雪崩效应）。扩展置换的置换方法与初始置换相同，只是置换表不同，扩展置换表如下所示。

表6-1-2 扩展置换表

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13

12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

2．异或运算

扩展置换的48位输出数据与相应的子密钥进行按位异或运算，关于子密钥的产生过程以后将详细讨论，按位异或运算的运算法则如下（其中⊕为异或运算符）：

$$0 \oplus 0=0$$

$$0 \oplus 1=1$$

$$1 \oplus 0=1$$

$$1 \oplus 1=0$$

异或以后的48位结果将继续进行S盒置换。

3．S盒置换

S盒置换是DES算法中最重要的部分，也是最关键的步骤，因为其他的运算都是线性的，易于分析，只有S盒代替是非线性的，它比DES中任何一步都提供了更好的安全性。

经过异或运算得到的48位输出数据要经过S盒置换，置换由8个盒完成，记为S盒。每个S盒都有6位输入，4位输出，如图6-1-4所示。

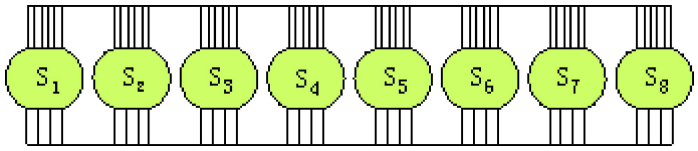

图6-1-4　S盒

这8个S盒是不同的，每个S盒的置换方法如表6-1-3所示。这个表的使用方法如下：48位的输入分成8组，每组6位，分别进入8个S盒。将每组的6位输入记为B0B1B2B3B4B5，那么表中的行号由B0 B5决定，而列号由B1 B2 B3 B4决定。例如，第一个分组111000要进入第一个S盒S1，那么行号为10（B0 B5）即第2行，列号为1100（B1 B2 B3 B4）即第12列，第2行第12列对应的数据为3，所以这个S盒的4位输出就是3的二进制表示0011。

表6-1-3　DES算法S盒置换表

S1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2																
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3																
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4																
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5																
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6																
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7																
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8																
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

4．直接置换

S盒置换后的32位输出数据将进行直接置换，该置换把每个输入位映射到输出位，任意一位不能被映射两次，也不能略去，表6-1-4为直接置换表，该表的使用方法与初始置换相同。

表6-1-4　直接置换表

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

七．终结置换

终结置换与初始置换相对应，它们都不影响DES的安全性，主要目的是为了更容易的将明文和密文数据以字节大小放入DES的f算法或者DES芯片中。表6-1-5为终结置换表，这个表的使用方法与初始置换表相同。

表6-1-5　终结置换表

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

对明文的每一个分组都做以上的操作，便得到了密文，明文和密文的位数是一致的。

八．子密钥产生过程

在每一轮的子加密过程中，48位的明文数据要与48位的子密钥进行异或运算，子密钥的产生过程如图6-1-5所示。

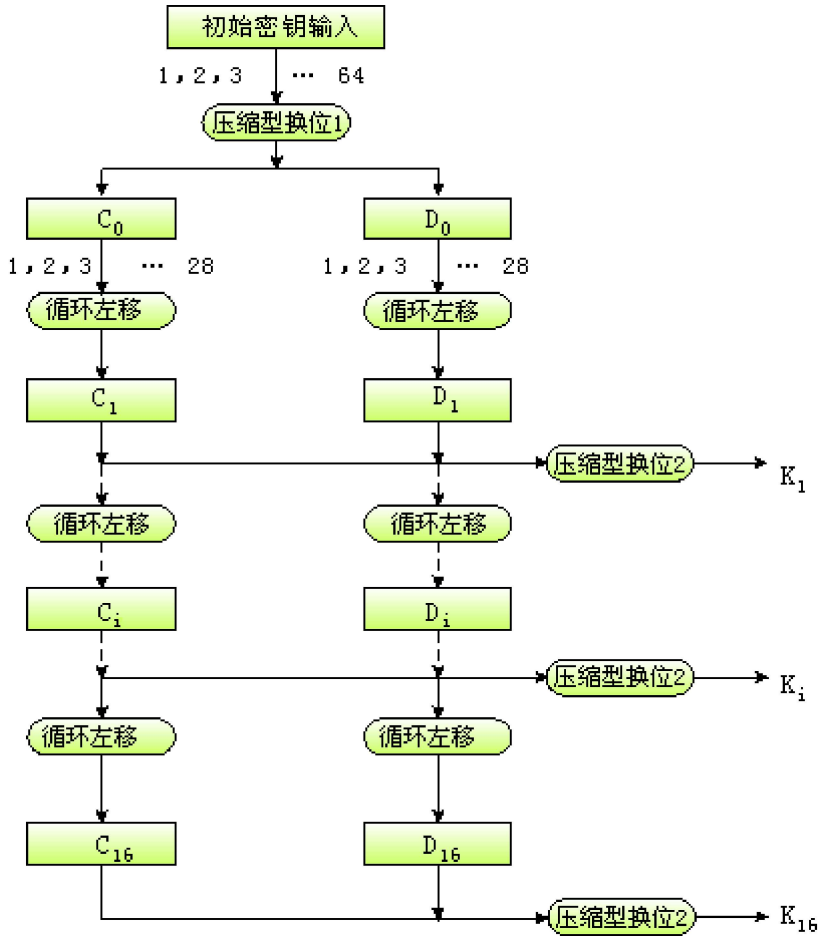

图6-1-5　子密钥产生过程

64位的初始密钥就是使用者所持有的64位密钥，首先初始密钥经过压缩型换位1，将初始密钥的8个奇偶校验位剔除，并且将留下的56位密钥顺序按位打乱。压缩型换位1的置换表如下：

表6-1-6　压缩型换位1置换表

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

经过压缩型换位1，64位密钥被压缩成为56位。这56位密钥在中间分开，每部分28位，左半部分记为C，右半部分记为D，然后进入子密钥生成的16轮循环，每一轮循环将产生一个子密钥。

九．子密钥的16轮循环

C和D要经过16轮类似的操作产生16份子密钥，每一轮子密钥的产生都要经过循环左移和压缩型换位2。

循环左移要求C部分和D部分要根据循环左移表进行循环左移，循环左移表给出了每一轮需要循环左移的位数，表6-1-7为循环左移表。循环左移后的C和D部分作为下一轮子密钥的输入数据，直到16轮全部完成。

表6-1-7　循环左移表

轮数	循环左移位数	轮数	循环左移位数	轮数	循环左移位数	轮数	循环左移位数
1	1	5	2	9	1	13	2
2	1	6	2	10	2	14	2
3	2	7	2	11	2	15	2
4	2	8	2	12	2	16	1

经过循环左移之后，C和D部分合并成为56位的数据。之后这56位数据要经过压缩型换位2生成最终的48位子密钥，这48位的子密钥将与48位的明文数据进行异或操作。表6-1-8为压缩型换位2的置换表。

表6-1-8　压缩型换位2置换表

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32