实验一 DES 加密算法编程实现

华南理工大学 软件学院
陈春华 (博士)
chunhuachen@scut.edu.cn

一、实验目的

通过使用DES 算法对实验数据进行加密和解密，掌握现代分组密码算法基本原理，熟练掌握DES 算法各部件的运算原理和具体运算过程。

二、实验原理

现在密码算法可分为对称密码（Symmetric Cryptology）和非对称密码（Asymmetric Cryptology）；其区分依据主要是所采用的密钥间的关系。在对称密码中，加密密钥和解密密钥是完全相同的，或彼此之间容易互相推导。在非对称密码算法，或称为公钥密码（Public Key Cryptology）中，加密密钥和解密密钥是不同的，从加密密钥推导出解密密钥在计算上是不可行的（Computationally infeasible）。

根据对明文的处理方式不同，密码算法又可分为流密码（Stream Cipher）和分组密码（Block Cipher）。一次只对明文中的单个比特（有时对字节）运算的密码称为流密码。对明文的一组比特进行运算，这些比特组称为分组（如64位比特为一组），相应的密码称为分组密码。

1973 年，美国国家标准局（NBS）开始征集一种标准的数据加密标准算法（DES），以用于非机密性政府机构、商业部门和民间的对非机密的敏感数据进行加密。IBM 公司在1971年完成的LUCIFER 密码(64 比特分组，128 比特密钥)的基础上，改进成为建议的DES。改进后的DES算法仅使用56比特密钥，同时对S盒的修改被列入官方机密，曾广受批评。1975年3 月17 日，NBS 公布了这个算法，并说明要以它作为联邦信息处理标准，征求各方意见。1977 年1 月15 日，建议被批准为联邦标准—FIPSPUB 46，并设计推出了DES 芯片。1981年，ANSI 将DES 作为标准，即DEA[ANSI X3.92]。1983 年，ISO 采用DES 作为标准，即DEA-1。DES（Data Encryption Standard）是一个优秀的对称分组密码算法，直到2000 年10月2 日NIST 宣布AES 算法前，其一直是业界的标准。

DES加密：

图1表明了DES加密的整个机制。对任意加密方案，总有两个输入：明文和密钥。DES的明文长为64位，密钥长为56位。

从图1的左半部分，可见明文的处理经过三个阶段。首先，64位的明文经过**初始置换（IP）**而重新排列。然后进行16轮相同函数的作用（又称**迭代**），每轮都进行置换和替代的操作。这16轮迭代操作可以视为一个函数，其输入包括64位明文和16个轮密钥（图1中的K_i等，详细见下文），其输出为64位比特流（即为最后一轮迭代输出）。该输出左半部分（左32位）和右半部分（右32位）**互换**（即图中32位互换）产生预输出。最后该预输出再通过一个初始置换（IP）互逆的置换（IP^{-1}，又称**逆初始置换**）的作用产生64位的密文。

图1的右半部分给出了使用56位密钥的过程。密钥经过初始置换（即图中**置换选择1**）后，经过**循环左移**和置换（即图中**置换选择2**）分别得到子密钥K_i用于每轮的迭代（又称**轮密钥**）。每轮的置换函数（置换选择2）都一样，但是由于密钥的循环位移使得轮密钥互不相同。

初始置换：

表1和表2给分别定义了初始置换和逆初始置换，其解释如下。表的输入标记为1到64，共64位。置换表中64个元素代表从1到64这些数的一个置换。置换表中的每一个元素表明了某个输入位在64位输出中的位置。

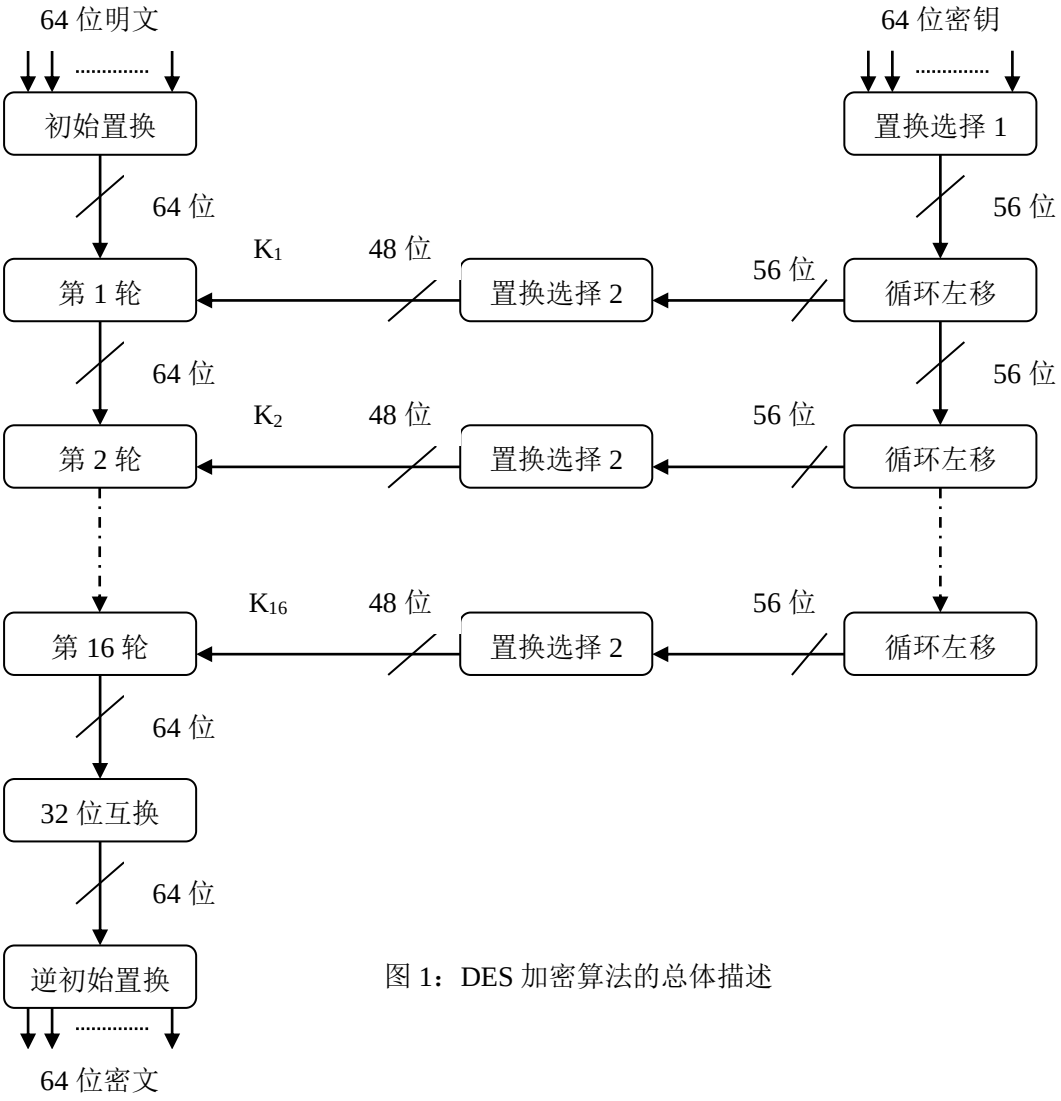

图 1：DES 加密算法的总体描述

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

表 1 初始置换（IP）

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

表 2 逆初始置换（IP^{-1}）

每轮变换的详细过程：

图2给出了一轮变换的内部结构。我们同样先看看图的左半部分。64位中间数据的左右两部分分作独立的32位数据，分别记为L（左）和R（右）。在任何古典Feistel密码中，每轮变换的整个过程可以写成如下的公式：

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

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

子密钥K_i长48位。R为32位。首先将R用表3定义的置换（E）扩展为48位（见表3扩充置换），其中16位是重复的。输出的48位与Ki异或，再用一个代换函数（即S盒）作用产生32位的输出，再用表4定义的置换（P）进行作用后输出。

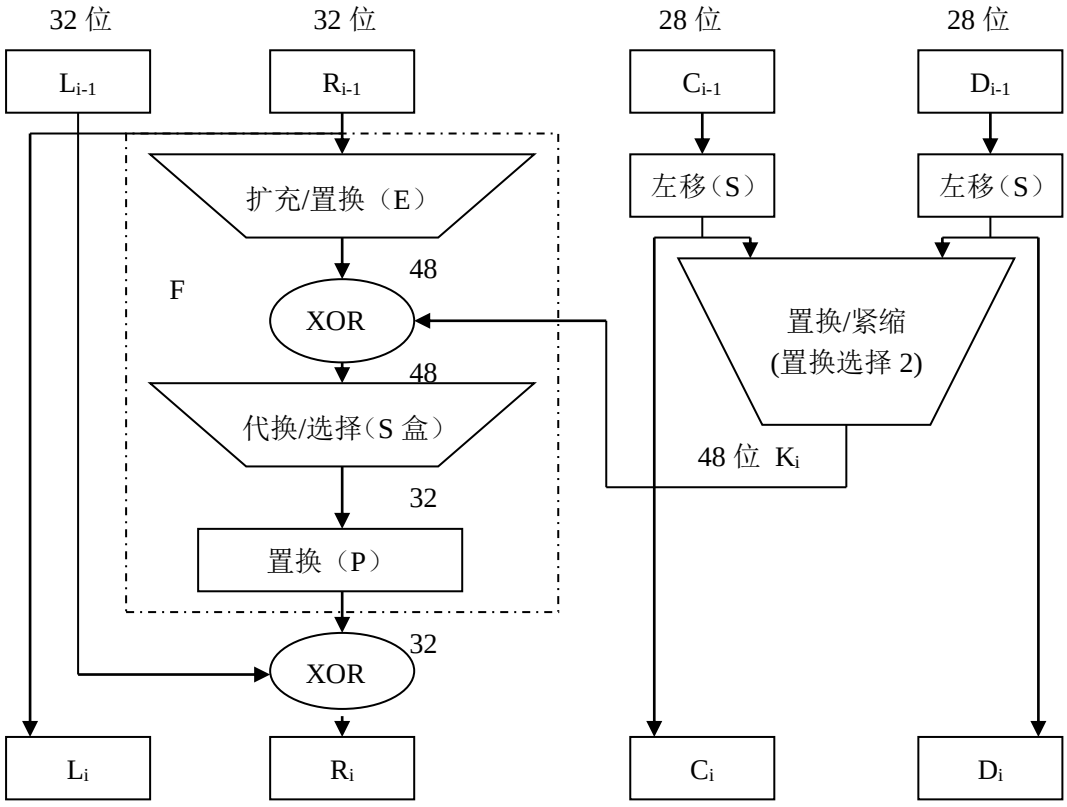

图 2 DES 算法一轮迭代的过程

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

表 3 扩充置换（E）

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

表 4 置换函数（P）

图3解释了S盒在函数F中的作用。代换函数有8个S盒来组成（见下列表），每个S盒都输入6位，输出4位。这些变换参见表5，其中解释如下：S_i盒输入的第一位和最后一位组成一个2位的二进制数来选择S_i盒4行（编号0~3）代换值中的一行，中间4位用来选择16列（编号0~15）中的一列。行列交叉处的十进制数转换为二进制后可得到输出的4位二进制数。例如，在S1中，如输入位011001，则行是1（01），列是12（1100），该处的值是9，所以输出为1001。注意，S盒的每一行都定义了一个普通的可逆代换。

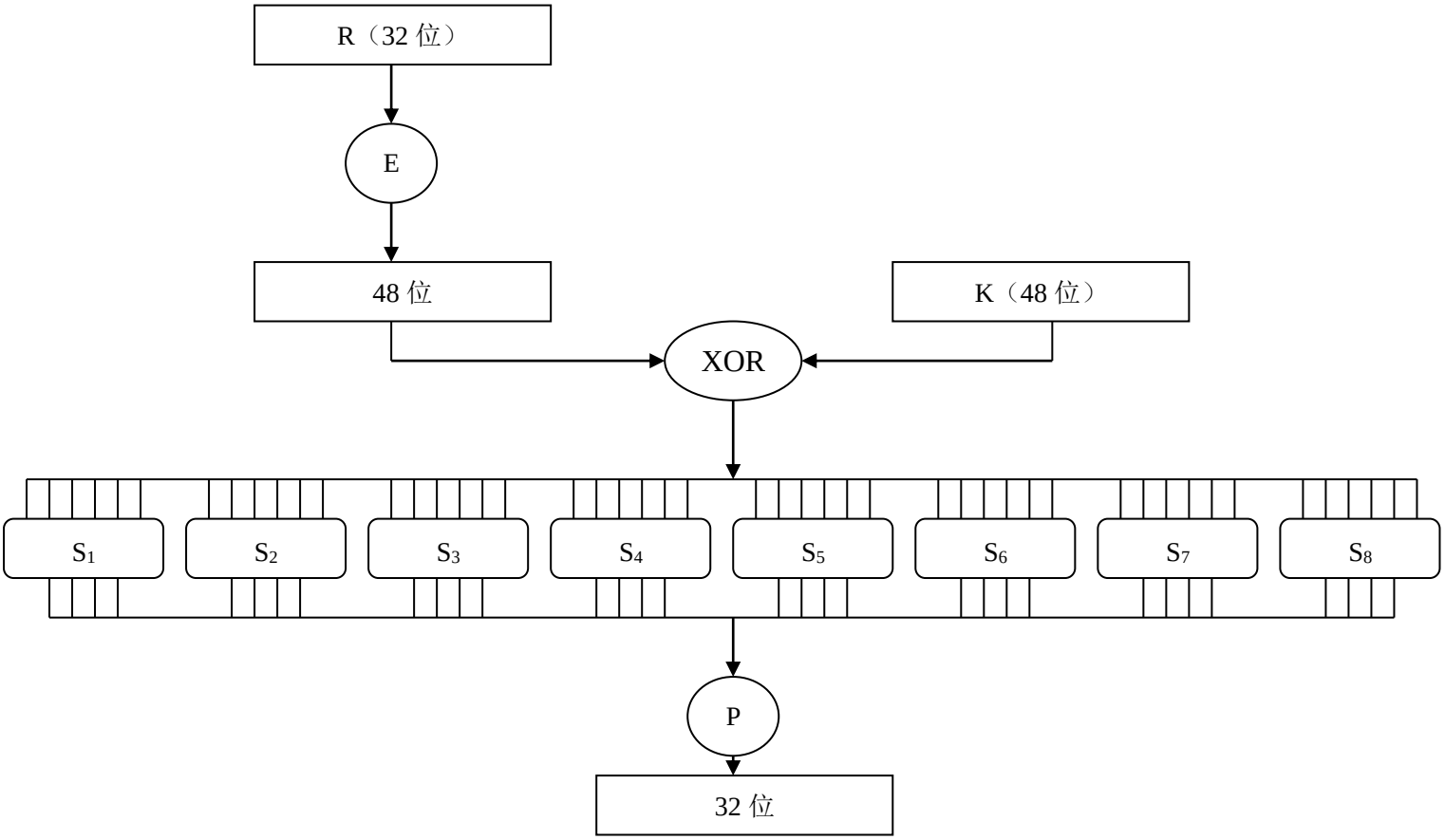

图 3　F（R, K）的计算

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4	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

S_5	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

S_6	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

S_7	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

S_8	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

表 5　S 盒表

密钥产生：

回顾图1和图2中，我们看到算法输入了64位的密钥，但是DES算法仅使用其中的56位。密钥各个比特分别标记为1到64，选取如表6中的无阴影部分，也就是每行的第8个比特位被忽略。对选取的56位密钥，首先进行**置换选择1**（表7）操作，所得到的56位密钥分为前后两个28位密钥数据C_0和D_0。每轮迭代中，C_{i-1}和D_{i-1}分别**循环左移**一位或两位，参加表8。移位后的值，通过**置换选择2**（表9）操作，产生一个48位的轮密钥作为函数$F(R_{i-1}, K_i)$的输入。

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	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
57	58	59	60	61	62	63	64

表 6 56 位密钥选取表

C_0	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
D_0	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

表 7 置换选择 1

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

表 8 置换选择 2

9	18	22	25	35	38	43	54

上述位被去除

迭代轮数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
移位次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

表 9 对循环左移位数的规定

DES解密：

Feistel密码的解密算法和加密算法是相同的，只是轮密钥的使用次序相反。

三、实验环境

运行Windows 或Linux 操作系统的PC 机，具有gcc(Linux)、VC（Windows）等C 语言编译环境。

四、实验内容

路线一

1. 分析和学习LibTomCrypt密码算法库，尤其是关于DES加解密算法的实现。（见/libtomcrypt-1.17/doc/crypt.pdf）

路线二

1. 使用一种编程语言（推荐用C或者C++，可使用其他语言，如JAVA等），实现DES加解密算法。

2. 利用自己编程实现的DES算法或者LibTomCryp库提供的DES算法，进行如下加密和解密操作。

a）使用同一密钥，对两组明文进行加密和解密。

64位密钥：

00000010 10010110 01001000 11000100 00111000 00110000 00111000 01100100

64位明文块1：

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

64位明文块2（与明文块1仅有一位的不同）：

10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

输出两个密文块的二进制流，统计两个密文块间不同数据位的数量。

b）对同一段明文，使用不同密钥进行加密和解密操作。

64位密钥1：

11100010 11110110 11011110 00110000 00111010 00001000 01100010 11011100

64位密钥2（与密钥1仅有一位的不同）：

01100010 11110110 11011110 00110000 00111010 00001000 01100010 11011100

明文：

01101000 10000101 00101111 01111010 00010011 01110110 11101011 10100100

输出两个密文块的二进制流，统计两个密文块间不同数据位的数量。

五、实验报告要求

1. 提交程序代码和执行结果（包括实验截图，和统计结果等）。
2. 什么是雪崩效益？根据以上结果，说明DES加密算法是否具有该性质。

参考文献

[1] 《密码编码学与网络安全》 William Stallings（著）,孟庆树等（译）, 电子工业出版社。