Project 4: Learning about DES

Cryptology 2005

Introduction

In this project we will learn about the construction and cryptanalysis of block ciphers. In particular, we will learn about the *Data Encryption Standard*, or *DES* for short, which is the most well-known block cipher. It is used in many cryptographic applications today.

The project is divided into two parts, one describing the DES algorithm, and one describing "linear cryptanalysis" which is one of the most famous general attack on block ciphers. You will need to do 6 exercises: 4 home (manually) and 2 laboratory (with a computer) exercises.

1 The DES Algorithm

1.1 Background

DES was developed at IBM, as a modification of an earlier cryptosystem known as LUCIFER. DES was first published in the Federal Register of March 17, 1975. After a considerable amount of public discussion, DES was adopted as a standard for "unclassified" applications on January 15, 1977. DES has been reviewed by the National Bureau of Standards every five years since its adoption.

1.2 Description of the DES Algorithm

A complete description of DES is given in the Federal Information Processing Standards Publication 46, dated January 15, 1977. DES encrypts a plaintext bitstring x of length 64, using a key K which is a bitstring of length 56, obtaining a ciphertext bitstring c which is again a bitstring of length 64. The algorithm can first be described in three steps.

1. From the plaintext x, a bitstring x_0 is constructed by applying a fixed *initial permutation*, called IP. The bitstring $x_0 = \mathrm{IP}(x)$ is divided in two parts, $x_0 = L_0R_0$, where L_0 is the first 32 (leftmost) bits and R_0 is the 32 last (rightmost) bits.

2. A certain function with start value x_0 is then iterated 16 times. If $x_i = L_iR_i$, we compute L_iR_i according to the following iteration:

$$\begin{aligned} L_i &= R_{i-1}, \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i), \end{aligned}$$

   where $\oplus$ denotes bitwise addition of the two bitstrings. The function f will be described later, and $K_1, K_2, \ldots, K_{16}$ are bitstrings of length 48, each a selection of bits from the key K. We describe later how these bits are selected. One such iteration is called a *round*, see Figure 2.

3. Finally, the inverse permutation IP^{-1} is applied to the **reversed** bitstring $R_{16}L_{16}$. The result is the ciphertext $c = \mathrm{IP}^{-1}(R_{16}L_{16})$. Note the reversed order of L_{16} and R_{16}.

Already now we can identify one of the essential features of DES, namely how the decryption algorithm will look like.

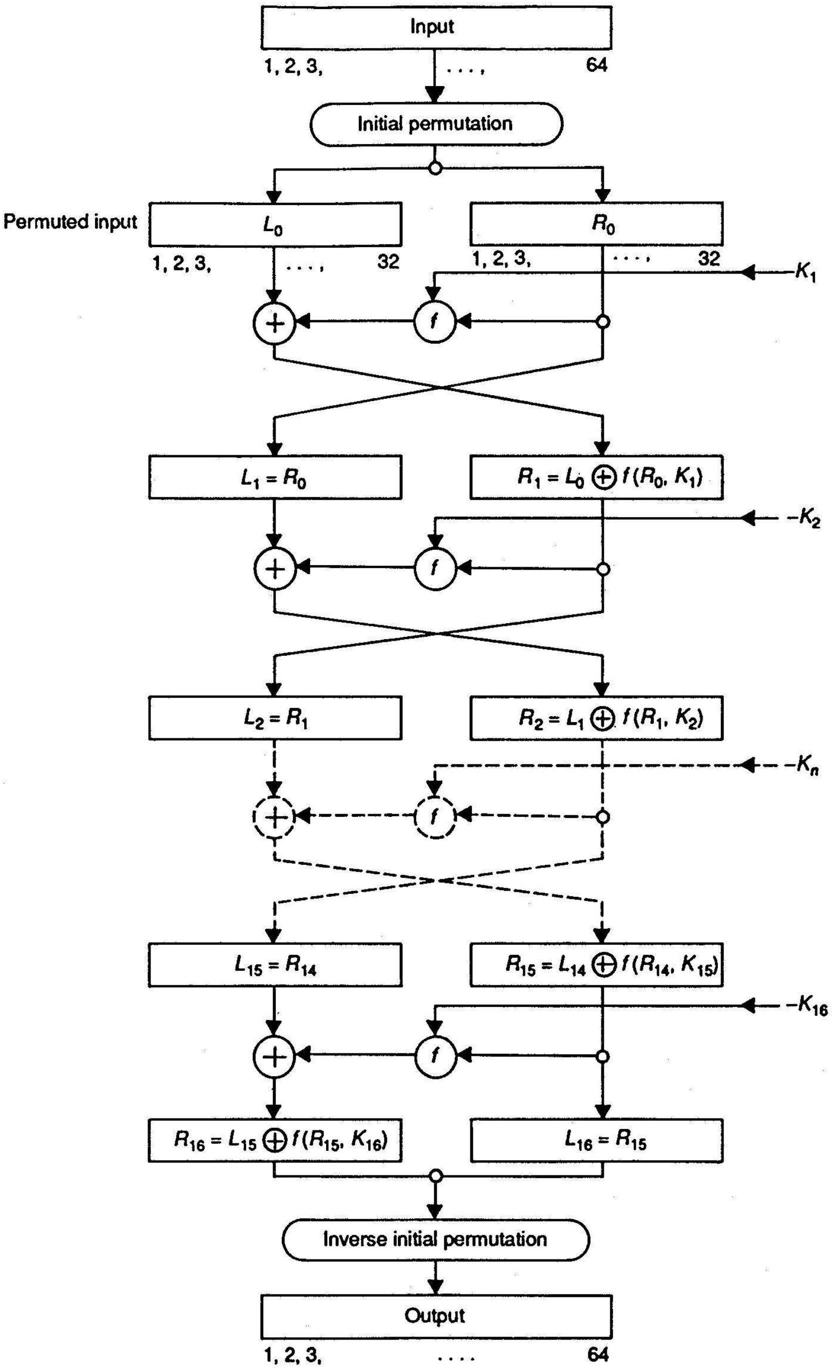

Figure 1: The structure of DES.

Home-Exercise 1 *Verify that decryption is done by using the same algorithm, starting with c as input, but using the keys K_i in reverse order, i.e., in the order $K_{16}, K_{15}, \ldots, K_1$.*

□

We now continue to describe the function f. If we write $f(R_x, K_x)$, then the first argument R_x is a bitstring of length 32, and the second argument K_x is a bitstring of length 48. The function $f(R_x, K_x)$ then returns a bitstring of length 32, which are obtained by executing the following steps:

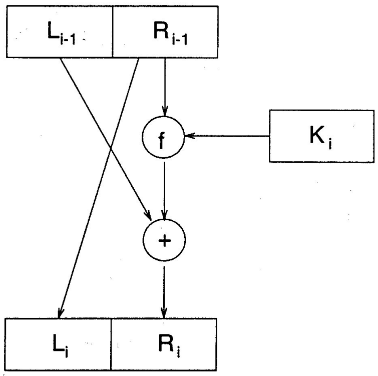

Figure 2: One round in DES.

1. The first argument R_x is expanded to a bitstring of length 48 according to a fixed expansion function E. $E(R_x)$ consists of 48 bits from R_x, some bits appearing twice.
2. Compute $B = E(R_x) \oplus K_x$ and write the result as a concatenation of eight 6-bit strings $B = B_1B_2B_3B_4B_5B_6B_7B_8$.
3. The next step uses eight *S-boxes* $S_1, S_2, \ldots, S_8$. Each S_i is a fixed 4×16 array whose entries are from the integers $0 - 15$. Given a 6-bit string $B_j = b_1b_2b_3b_4b_5b_6$, we compute $S_j(B_j)$ as follows. The two bits b_1b_6 determine the binary representation of a row r of S_j, $0 \leq r \leq 3$, and the four bits $b_2b_3b_4b_5$ determine the binary representation of a column c of S_j, $0 \leq c \leq 15$. Then $S_j(B_J)$ is defined to be the entry in row r and column c, written in a binary representation as a 4-bit string. In this fashion, we compute $C_j = S_j(B_j)$ for $1 \leq j \leq 8$.
4. The bitstring $C = C_1C_2C_3C_4C_5C_6C_7C_8$ obtained from the previous step is permuted according to a fixed permutation P. The resulting 32-bit string $P(C)$ is defined to be $f(R_x, K_x)$.

An overview of the f function is shown in Figure 3. The strength of the DES cipher lies entirely in the S-boxes.

Home-Exercise 2 *Find the value of $f(R_x, K_x)$ for input values R_x =115599DD and K_x =12345678ABCD.*

□

We now give the different predefined functions of DES. The initial permutation is as follows:

$$\mathrm{IP} = \begin{bmatrix} 58 & 50 & 42 & 34 & 26 & 18 & 10 & 2 \\ 60 & 52 & 44 & 36 & 28 & 20 & 12 & 4 \\ 62 & 54 & 46 & 38 & 30 & 22 & 14 & 6 \\ 64 & 56 & 48 & 40 & 32 & 24 & 16 & 8 \\ 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 \\ 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{bmatrix}$$

The notation means that the 58th bit of x is the first bit in $\mathrm{IP}(x)$, the 50th bit of x is the second bit in $\mathrm{IP}(x)$, and so on. The inverse permutation IP^{-1} is then the following:

$$\mathrm{IP}^{-1} = \begin{bmatrix} 40 & 8 & 48 & 16 & 56 & 24 & 64 & 32 \\ 39 & 7 & 47 & 15 & 55 & 23 & 63 & 31 \\ 38 & 6 & 46 & 14 & 54 & 22 & 62 & 30 \\ 37 & 5 & 45 & 13 & 53 & 21 & 61 & 29 \\ 36 & 4 & 44 & 12 & 52 & 20 & 60 & 28 \\ 35 & 3 & 43 & 11 & 51 & 19 & 59 & 27 \\ 34 & 2 & 42 & 10 & 50 & 18 & 58 & 26 \\ 33 & 1 & 41 & 9 & 49 & 17 & 57 & 25 \end{bmatrix}$$

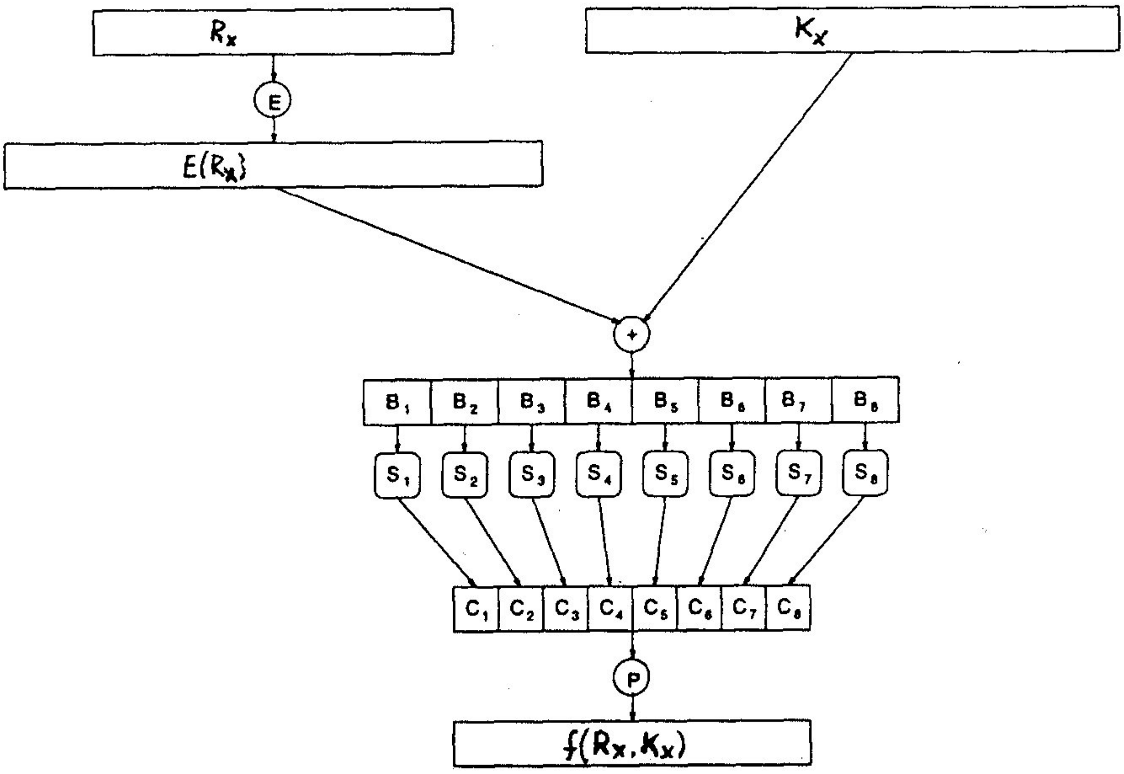

Figure 3: The f function in DES.

The expansion function E is specified by the following table:

$$\mathrm{E} = \begin{bmatrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{bmatrix}$$

The eight S-boxes are now as follows:

$$S_1 = \begin{bmatrix} 14 & 4 & 13 & 1 & 2 & 15 & 11 & 8 & 3 & 10 & 6 & 12 & 5 & 9 & 0 & 7 \\ 0 & 15 & 7 & 4 & 14 & 2 & 13 & 1 & 10 & 6 & 12 & 11 & 9 & 5 & 3 & 8 \\ 4 & 1 & 14 & 8 & 13 & 6 & 2 & 11 & 15 & 12 & 9 & 7 & 3 & 10 & 5 & 0 \\ 15 & 12 & 8 & 2 & 4 & 9 & 1 & 7 & 5 & 11 & 3 & 14 & 10 & 0 & 6 & 13 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \\ 3 & 13 & 4 & 7 & 15 & 2 & 8 & 14 & 12 & 0 & 1 & 10 & 6 & 9 & 11 & 5 \\ 0 & 14 & 7 & 11 & 10 & 4 & 13 & 1 & 5 & 8 & 12 & 6 & 9 & 3 & 2 & 15 \\ 13 & 8 & 10 & 1 & 3 & 15 & 4 & 2 & 11 & 6 & 7 & 12 & 0 & 5 & 14 & 9 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 10 & 0 & 9 & 14 & 6 & 3 & 15 & 5 & 1 & 13 & 12 & 7 & 11 & 4 & 2 & 8 \\ 13 & 7 & 0 & 9 & 3 & 4 & 6 & 10 & 2 & 8 & 5 & 14 & 12 & 11 & 15 & 1 \\ 13 & 6 & 4 & 9 & 8 & 15 & 3 & 0 & 11 & 1 & 2 & 12 & 5 & 10 & 14 & 7 \\ 1 & 10 & 13 & 0 & 6 & 9 & 8 & 7 & 4 & 15 & 14 & 3 & 11 & 5 & 2 & 12 \end{bmatrix}$$

$$S_4 = \begin{bmatrix} 7 & 13 & 14 & 3 & 0 & 6 & 9 & 10 & 1 & 2 & 8 & 5 & 11 & 12 & 4 & 15 \\ 13 & 8 & 11 & 5 & 6 & 15 & 0 & 3 & 4 & 7 & 2 & 12 & 1 & 10 & 14 & 9 \\ 10 & 6 & 9 & 0 & 12 & 11 & 7 & 13 & 15 & 1 & 3 & 14 & 5 & 2 & 8 & 4 \\ 3 & 15 & 0 & 6 & 10 & 1 & 13 & 8 & 9 & 4 & 5 & 11 & 12 & 7 & 2 & 14 \end{bmatrix}$$

$$S_5 = \begin{bmatrix} 2 & 12 & 4 & 1 & 7 & 10 & 11 & 6 & 8 & 5 & 3 & 15 & 13 & 0 & 14 & 9 \\ 14 & 11 & 2 & 12 & 4 & 7 & 13 & 1 & 5 & 0 & 15 & 10 & 3 & 9 & 8 & 6 \\ 4 & 2 & 1 & 11 & 10 & 13 & 7 & 8 & 15 & 9 & 12 & 5 & 6 & 3 & 0 & 14 \\ 11 & 8 & 12 & 7 & 1 & 14 & 2 & 13 & 6 & 15 & 0 & 9 & 10 & 4 & 5 & 3 \end{bmatrix}$$

$$S_6 = \begin{bmatrix} 12 & 1 & 10 & 15 & 9 & 2 & 6 & 8 & 0 & 13 & 3 & 4 & 14 & 7 & 5 & 11 \\ 10 & 15 & 4 & 2 & 7 & 12 & 9 & 5 & 6 & 1 & 13 & 14 & 0 & 11 & 3 & 8 \\ 9 & 14 & 15 & 5 & 2 & 8 & 12 & 3 & 7 & 0 & 4 & 10 & 1 & 13 & 11 & 6 \\ 4 & 3 & 2 & 12 & 9 & 5 & 15 & 10 & 11 & 14 & 1 & 7 & 6 & 0 & 8 & 13 \end{bmatrix}$$

$$S_7 = \begin{bmatrix} 4 & 11 & 2 & 14 & 15 & 0 & 8 & 13 & 3 & 12 & 9 & 7 & 5 & 10 & 6 & 1 \\ 13 & 0 & 11 & 7 & 4 & 9 & 1 & 10 & 14 & 3 & 5 & 12 & 2 & 15 & 8 & 6 \\ 1 & 4 & 11 & 13 & 12 & 3 & 7 & 14 & 10 & 15 & 6 & 8 & 0 & 5 & 9 & 2 \\ 6 & 11 & 13 & 8 & 1 & 4 & 10 & 7 & 9 & 5 & 0 & 15 & 14 & 2 & 3 & 12 \end{bmatrix}$$

$$S_8 = \begin{bmatrix} 13 & 2 & 8 & 4 & 6 & 15 & 11 & 1 & 10 & 9 & 3 & 14 & 5 & 0 & 12 & 7 \\ 1 & 15 & 13 & 8 & 10 & 3 & 7 & 4 & 12 & 5 & 6 & 11 & 0 & 14 & 9 & 2 \\ 7 & 11 & 4 & 1 & 9 & 12 & 14 & 2 & 0 & 6 & 10 & 13 & 15 & 3 & 5 & 8 \\ 2 & 1 & 14 & 7 & 4 & 10 & 8 & 13 & 15 & 12 & 9 & 0 & 3 & 5 & 6 & 11 \end{bmatrix}$$

Finally, the permutation P is described by

$$\mathrm{P} = \begin{bmatrix} 16 & 7 & 20 & 21 \\ 29 & 12 & 28 & 17 \\ 1 & 15 & 23 & 26 \\ 5 & 18 & 31 & 10 \\ 2 & 8 & 24 & 14 \\ 32 & 27 & 3 & 9 \\ 19 & 13 & 30 & 6 \\ 22 & 11 & 4 & 25 \end{bmatrix}$$

We now need to describe how the key K is used in the key schedule to give the partial keys $K_1, K_2, \ldots, K_{16}$. The key K is a 64 bit string which consists of 56 actual key bits and 8 parity check bits. The parity check bits are the bits in positions $8, 16, 24, \ldots, 64$. The parity check bits are defined in such a way that each byte contains an odd number of 1's. The parity check bits are ignored in the key schedule.

As mentioned before, each round uses as the key K_i 48 bits, that are selected bits from K. The entries in the array refer to the bits of K that are used in the various rounds.

$$K_1 = \begin{bmatrix} 10 & 51 & 34 & 60 & 49 & 17 & 33 & 57 & 2 & 9 & 19 & 42 \\ 3 & 35 & 26 & 25 & 44 & 58 & 59 & 1 & 36 & 27 & 18 & 41 \\ 22 & 28 & 39 & 54 & 37 & 4 & 47 & 30 & 5 & 53 & 23 & 29 \\ 61 & 21 & 38 & 63 & 15 & 20 & 45 & 14 & 13 & 62 & 55 & 31 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 2 & 43 & 26 & 52 & 41 & 9 & 25 & 49 & 59 & 1 & 11 & 34 \\ 60 & 27 & 18 & 17 & 36 & 50 & 51 & 58 & 57 & 19 & 10 & 33 \\ 14 & 20 & 31 & 46 & 29 & 63 & 39 & 22 & 28 & 45 & 15 & 21 \\ 53 & 13 & 30 & 55 & 7 & 12 & 37 & 6 & 5 & 54 & 47 & 23 \end{bmatrix}$$

$$K_3 = \begin{bmatrix} 51 & 27 & 10 & 36 & 25 & 58 & 9 & 33 & 43 & 50 & 60 & 18 \\ 44 & 11 & 2 & 1 & 49 & 34 & 35 & 42 & 41 & 3 & 59 & 17 \\ 61 & 4 & 15 & 30 & 13 & 47 & 23 & 6 & 12 & 29 & 62 & 5 \\ 37 & 28 & 14 & 39 & 54 & 63 & 21 & 53 & 20 & 38 & 31 & 7 \end{bmatrix}$$

$$K_4 = \begin{bmatrix} 35 & 11 & 59 & 49 & 9 & 42 & 58 & 17 & 27 & 34 & 44 & 2 \\ 57 & 60 & 51 & 50 & 33 & 18 & 19 & 26 & 25 & 52 & 43 & 1 \\ 45 & 55 & 62 & 14 & 28 & 31 & 7 & 53 & 63 & 13 & 46 & 20 \\ 21 & 12 & 61 & 23 & 38 & 47 & 5 & 37 & 4 & 22 & 15 & 54 \end{bmatrix}$$

$$K_5 = \begin{bmatrix} 19 & 60 & 43 & 33 & 58 & 26 & 42 & 1 & 11 & 18 & 57 & 51 \\ 41 & 44 & 35 & 34 & 17 & 2 & 3 & 10 & 9 & 36 & 27 & 50 \\ 29 & 39 & 46 & 61 & 12 & 15 & 54 & 37 & 47 & 28 & 30 & 4 \\ 5 & 63 & 45 & 7 & 22 & 31 & 20 & 21 & 55 & 6 & 62 & 38 \end{bmatrix}$$

$$K_6 = \begin{bmatrix} 3 & 44 & 27 & 17 & 42 & 10 & 26 & 50 & 60 & 2 & 41 & 35 \\ 25 & 57 & 19 & 18 & 1 & 51 & 52 & 59 & 58 & 49 & 11 & 34 \\ 13 & 23 & 30 & 45 & 63 & 62 & 38 & 21 & 31 & 12 & 14 & 55 \\ 20 & 47 & 29 & 54 & 6 & 15 & 4 & 5 & 39 & 53 & 46 & 22 \end{bmatrix}$$

$$K_7 = \begin{bmatrix} 52 & 57 & 11 & 1 & 26 & 59 & 10 & 34 & 44 & 51 & 25 & 19 \\ 9 & 41 & 3 & 2 & 50 & 35 & 36 & 43 & 42 & 33 & 60 & 18 \\ 28 & 7 & 14 & 29 & 47 & 46 & 22 & 5 & 15 & 63 & 61 & 39 \\ 4 & 31 & 13 & 38 & 53 & 62 & 55 & 20 & 23 & 37 & 30 & 6 \end{bmatrix}$$

$$K_8 = \begin{bmatrix} 36 & 41 & 60 & 50 & 10 & 43 & 59 & 18 & 57 & 35 & 9 & 3 \\ 58 & 25 & 52 & 51 & 34 & 19 & 49 & 27 & 26 & 17 & 44 & 2 \\ 12 & 54 & 61 & 13 & 31 & 30 & 6 & 20 & 62 & 47 & 45 & 23 \\ 55 & 15 & 28 & 22 & 37 & 46 & 39 & 4 & 7 & 21 & 14 & 53 \end{bmatrix}$$

$$K_9 = \begin{bmatrix} 57 & 33 & 52 & 42 & 2 & 35 & 51 & 10 & 49 & 27 & 1 & 60 \\ 50 & 17 & 44 & 43 & 26 & 11 & 41 & 19 & 18 & 9 & 36 & 59 \\ 4 & 46 & 53 & 5 & 23 & 22 & 61 & 12 & 54 & 39 & 37 & 15 \\ 47 & 7 & 20 & 14 & 29 & 38 & 31 & 63 & 62 & 13 & 6 & 45 \end{bmatrix}$$

$$K_{10} = \begin{bmatrix} 41 & 17 & 36 & 26 & 51 & 19 & 35 & 59 & 33 & 11 & 50 & 44 \\ 34 & 1 & 57 & 27 & 10 & 60 & 25 & 3 & 2 & 58 & 49 & 43 \\ 55 & 30 & 37 & 20 & 7 & 6 & 45 & 63 & 38 & 23 & 21 & 62 \\ 31 & 54 & 4 & 61 & 13 & 22 & 15 & 47 & 46 & 28 & 53 & 29 \end{bmatrix}$$

$$K_{11} = \begin{bmatrix} 25 & 1 & 49 & 10 & 35 & 3 & 19 & 43 & 17 & 60 & 34 & 57 \\ 18 & 50 & 41 & 11 & 59 & 44 & 9 & 52 & 51 & 42 & 33 & 27 \\ 39 & 14 & 21 & 4 & 54 & 53 & 29 & 47 & 22 & 7 & 5 & 46 \\ 15 & 38 & 55 & 45 & 28 & 6 & 62 & 31 & 30 & 12 & 37 & 13 \end{bmatrix}$$

$$K_{12} = \begin{bmatrix} 9 & 50 & 33 & 59 & 19 & 52 & 3 & 27 & 1 & 44 & 18 & 41 \\ 2 & 34 & 25 & 60 & 43 & 57 & 58 & 36 & 35 & 26 & 17 & 11 \\ 23 & 61 & 5 & 55 & 38 & 37 & 13 & 31 & 6 & 54 & 20 & 30 \\ 62 & 22 & 39 & 29 & 12 & 53 & 46 & 15 & 14 & 63 & 21 & 28 \end{bmatrix}$$

$$K_{13} = \begin{bmatrix} 58 & 34 & 17 & 43 & 3 & 36 & 52 & 11 & 50 & 57 & 2 & 25 \\ 51 & 18 & 9 & 44 & 27 & 41 & 42 & 49 & 19 & 10 & 1 & 60 \\ 7 & 45 & 20 & 39 & 22 & 21 & 28 & 15 & 53 & 38 & 4 & 14 \\ 46 & 6 & 23 & 13 & 63 & 37 & 30 & 62 & 61 & 47 & 5 & 12 \end{bmatrix}$$

$$K_{14} = \begin{bmatrix} 42 & 18 & 1 & 27 & 52 & 49 & 36 & 60 & 34 & 41 & 51 & 9 \\ 35 & 2 & 58 & 57 & 11 & 25 & 26 & 33 & 3 & 59 & 50 & 44 \\ 54 & 29 & 4 & 23 & 6 & 5 & 12 & 62 & 37 & 22 & 55 & 61 \\ 30 & 53 & 7 & 28 & 47 & 21 & 14 & 46 & 45 & 31 & 20 & 63 \end{bmatrix}$$

$$K_{15} = \begin{bmatrix} 26 & 2 & 50 & 11 & 36 & 33 & 49 & 44 & 18 & 25 & 35 & 58 \\ 19 & 51 & 42 & 41 & 60 & 9 & 10 & 17 & 52 & 43 & 34 & 57 \\ 38 & 13 & 55 & 7 & 53 & 20 & 63 & 46 & 21 & 6 & 39 & 45 \\ 14 & 37 & 54 & 12 & 31 & 5 & 61 & 30 & 29 & 15 & 4 & 47 \end{bmatrix}$$

$$K_{16} = \begin{bmatrix} 18 & 59 & 42 & 3 & 57 & 25 & 41 & 36 & 10 & 17 & 27 & 50 \\ 11 & 43 & 34 & 33 & 52 & 1 & 2 & 9 & 44 & 35 & 26 & 49 \\ 30 & 5 & 47 & 62 & 45 & 12 & 55 & 38 & 13 & 61 & 31 & 37 \\ 6 & 29 & 46 & 4 & 23 & 28 & 53 & 22 & 21 & 7 & 63 & 39 \end{bmatrix}$$

Home-Exercise 3 *Find the value of the key K_2 when $K = 01230123ABABEFEF$. Write the answer in hexadecimal notation.*

1.3 An example of a DES encryption

We here give an example showing how the encryption proceeds for a fixed key and a fixed plaintext. We encrypt the plaintext

0000000100100011010001010110011110001001101010111100110111101111

using the key (with parity check bits)

0001001**1**0011010**0**0101011**1**0111100**1**1001101**1**1011110**0**1101111**1**1111000**1**.

In hexadecimal notation this is written

Plaintext:	0123456789ABCDEF
Key:	133457799BBCDFF1

Applying IP we get L_0R_0 as

L_0R_0:	CC00CCFFF0AAF0AA

Then 16 rounds of encryption are performed, resulting in the following partial values.

K_1:	1B02EFFC7072	K_2:	79AED9DBC9E5
$E(R_0)$:	7A15557A1555	$E(R_1)$:	75EA5430AA09
$E(R_0)+K_0$:	6117BA866527	$E(R_1)+K_2$:	0C448DEB63EC
$f(R_0,K_1)$:	234AA9BB	$f(R_1,K_2)$:	3CAB87A3
L_1R_1:	F0AAF0AA - EF4A6544	L_2R_2:	EF4A6544 - CC017709
K_3:	55FC8A42CF99	K_4:	72ADD6DB351D
$E(R_2)$:	E58002BAE853	$E(R_3)$:	5042F8057FA9
$E(R_2)+K_3$:	B07C88F827CA	$E(R_3)+K_4$:	22EF2EDE4AB4
$f(R_2,K_3)$:	4D166EB0	$f(R_3,K_4)$:	BB23774C
L_3R_3:	C017709 - A25C0BF4	L_4R_4:	A25C0BF4 - 77220045
K_5:	7CEC07EB53A8	K_6:	63A53E507B2F
$E(R_4)$:	BAE90400020A	$E(R_5)$:	C5425FD0C1AF
$E(R_4)+K_5$:	C60503EB51A2	$E(R_5)+K_6$:	A6E76180BA80
$f(R_4,K_5)$:	2813ADC3	$f(R_5,K_6)$:	9E45CD2C
L_5R_5:	77220045 - 8A4FA637	L_6R_6:	8A4FA637 - E967CD69
K_7:	EC84B7F618BC	K_8:	F78A3AC13BFB
$E(R_6)$:	F52B0FE5AB53	$E(R_7)$:	00C2555F40A0
$E(R_6)+K_7$:	19AFB813B3EF	$E(R_7)+K_8$:	F7486F9E7B5B
$f(R_6,K_7)$:	8C051C27	$f(R_7,K_8)$:	3C0E86F9
L_7R_7:	E967CD69 - 064ABA10	L_8R_8:	064ABA10 - D5694B90
K_9:	E0DBEBEDE781	K_{10}:	B1F347BA464F
$E(R_8)$:	6AAB52A57CA1	$E(R_9)$:	1083F960C3F4
$E(R_8)+K_9$:	8A70B9489B20	$E(R_9)+K_{10}$:	A170BEDA85BB
$f(R_8,K_9)$:	22367C6A	$f(R_9,K_{10})$:	62BC9C22
L_9R_9:	D5694B90 - 247CC67A	$L_{10}R_{10}$:	247CC67A - B7D5D7B2
K_{11}:	215FD3DED386	K_{12}:	7571F59467E9
$E(R_{10})$:	5AFEABEAFDA5	$E(R_{11})$:	60ABF01F83F1
$E(R_{10})+K_{11}$:	7BA178342E23	$E(R_{11})+K_{12}$:	15DA058BE418
$f(R_{10},K_{11})$:	E104FA02	$f(R_{11},K_{12})$:	C268CFEA
$L_{11}R_{11}$:	B7D5D7B2 - C5783C78	$L_{12}R_{12}$:	C5783C78 - 75BD1858
K_{13}:	97C5D1FABA41	K_{14}:	5F43B7F2E73A
$E(R_{12})$:	3ABDFA8F02F0	$E(R_{13})$:	0F16068AAAF4
$E(R_{12})+K_{13}$:	AD782B75B8B1	$E(R_{13})+K_{14}$:	5055B1784DCE
$f(R_{12},K_{13})$:	DDBB2922	$f(R_{13},K_{14})$:	B7318E55
$L_{13}R_{13}$:	75BD1858 - 18C3155A	$L_{14}R_{14}$:	18C3155A - C28C960D
K_{15}:	BF918D3D3F0A	K_{16}:	CB3D8B0E17F5
$E(R_{14})$:	E054594AC05B	$E(R_{15})$:	206A041A41A8
$E(R_{14})+K_{15}$:	5FC5D477FF51	$E(R_{15})+K_{16}$:	EB578F14565D
$f(R_{14}K_{15})$:	5B81276E	$f(R_{15},K_{16})$:	C8C04F98
$L_{15}R_{15}$:	C28C960D - 43423234	$L_{16}R_{16}$:	43423234 - 0A4CD995

Applying IP^{-1} to the reversed bitstring $R_{16}L_{16}$ we finally obtain the ciphertext as

Ciphertext:	85E813540F0AB405

Lab-Exercise 1 *Implement a 16-round DES. Each group is given a pair (***Key**, **Plaintext***). Encrypt the* **Plaintext** *with the given* **Key***, and output the corresponding* **Ciphertext** *in hexadecimal notation.*

1.4 The controversy of DES

[1] When DES was adopted as a standard, there was considerable critisism. One objection concerned the design of the S-boxes. All computations in DES, with the exception of the S-boxes, are *linear*, for example the exclusive-or operation of two inputs, or a permutation of input bits. It is a well known fact that linear cryptosystems can easily be cryptanalyzed using a known plaintext attack. Hence, the S-boxes, being the only non-linear component of DES, are vital to its security. But the design criteria of the S-boxes are not completely known. Several people have suggested that they might contain *trapdoors*, i.e., hidden weaknesses that would allow the National Security Agency to decrypt messages while maintaining that DES is secure. It is of course impossible to disprove such an assertion, but no evidence has ever appeared that indicates that such trapdoors in DES do in fact exist.

In 1976 the National Security Agency asserted that some of the design criteria for the S-boxes was the following: each row of each S-box is a permutation of the integers 1–15; no S-box is a linear of affine function of it input; changing one input bit to an S-box causes at least two output bits to change; for any S-box and any input x, $S(x)$ and $S(x \oplus 001100)$ differs in at least two bits. Other properties of the S-boxes can be found, caused by the design.

The most serious critisism of DES is that the key space, 2^{56}, is too small. Many special purpose machines have been proposed to do a known plaintext attack, which would essentially do an exhaustive search for the key. Already in 1977, Diffie and Hellman suggested that one could build a VLSI chip which would test 10^6 keys per second, and estimated that a machine with 10^6 chips could find the key in a day and cost $20 000 000.

More recently, in 1993 M. Wiener gave a detailed description of a machine, based on a key search chip which is pipelined to perform 16 encryptions simultaneously. A machine costing $1 000 000 would require an average search time of about 3.5 hours.

Today, in 2004, DES is considered as broken and is no longer recommended as a standard. In 1998 the organization EFF built a special hardware machine doing keysearch on DES. It costed less than $250 000 and can find the key in a number of hours. Instead, a new standard has been developed, called AES (*Advanced Encryption Standard*).

1.5 DES in the real world

Even though the description of DES is quite lengthy, it can be implemented very efficiently, both in hardware and software. The only aritmetic operations to be performed are exclusive-ors of bitstrings. The other functions, the S-boxes, E, IP, P, and the partial keys $K_1, K_2, \ldots, K_{16}$ can all be done by table look-up in software, or by hard-wiring them into a circuit.

Recent hardware implementations can attain extremely fast encryption rates. Digital Equipment Corporation announced in 1992 that they have a chip with 50 000 transistors that can encrypt at the rate of 1 Gbit/second using a clock rate of 250 MHz. The cost of a chip was about $300.

A very important application of DES is in banking transactions, using standards developed by the American Bankers Association. DES is used to encrypt personal identification numbers (PINs) and account transactions carried out by automatic teller machines (ATMs). DES is also used by the Clearing House Interbank Payment System (CHIPS) to authenticate transactions involving over $150 000 000 000 per week. DES is also widely used in government organizations, such as the Department of Energy, the Justice Department and the Federal Reserve System.

It can also be noted that implementations of DES are restricted export merchandise, and may not be exported outside U.S.A. without permission.

[1] Nowadays, the information given in these subsections is more historical.

2 Linear Cryptanalysis of a 3-round DES

2.1 Background

Differential Cryptanalysis has been one of main topics in cryptology since the first paper by Biham and Shamir in 1990. They have broken the FEAL cipher in a subsequent paper, and then, afterwards, succeeded with breaking the full 16-round DES cipher in a chosen-plaintext attack.

2.2 Linear Approximations

In this project we consider a restricted version of DES where the number of rounds is only 3. Thus, the structure of the "tiny" 3-round DES to be investigated is illustrated in Figure 4.

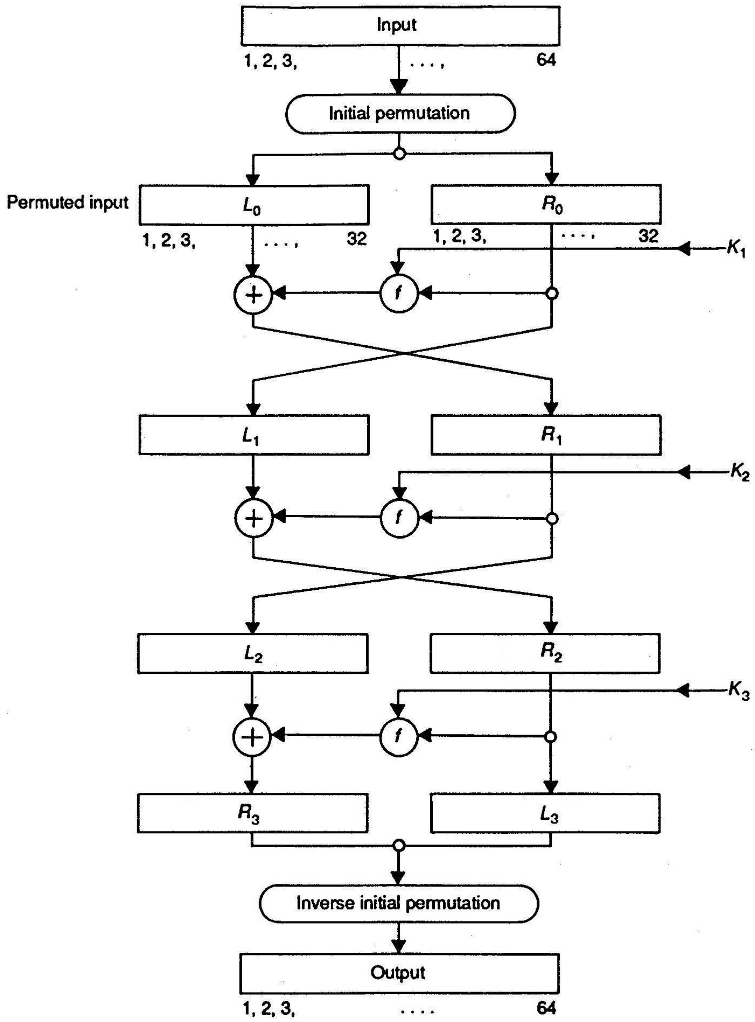

Figure 4: The structure of a 3-round DES.

For the notation purposes let us denote by $X[i_1, i_2, \ldots, i_n]$ the XOR-sum of bits $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ that belong to the binary vector X, i.e.,

$$X[i_1, i_2, \ldots, i_n] = \bigoplus_{t=1}^{n} x_{i_t}.$$

The purpose of linear cryptanalysis is to find the following "effective" linear expression for a given cipher algorithm:

$$P[i_1, i_2, \ldots, i_a] \oplus C[j_1, j_2, \ldots, j_b] = K[k_1, k_2, \ldots, k_c], \tag{1}$$

where $i_1, i_2, \ldots, i_a, j_1, j_2, \ldots, j_b$ and $k_1, k_2, \ldots, k_c$ denote fixed bit locations, and equation (1) holds with probability $p \neq 1/2$ for randomly given plaintext P and the corresponding ciphertext C. The magnitude of $|p - 1/2|$ represents the effectiveness of equation (1). Once we succeed in reaching an effective linear expression, it is possible to determine one key bit $K[k_1, k_2, \ldots, k_c]$ by a likelihood algorithm (out of scope in this project). We then can define a noise random variable to be:

$$N = P[i_1, i_2, \ldots, i_a] \oplus C[j_1, j_2, \ldots, j_b] \oplus K[k_1, k_2, \ldots, k_c], \tag{2}$$

which has some bias, i.e. $\Pr\{N = 0\} = \Pr\{Eq.\ (1)\ \ holds\} = p \neq 1/2$.

How do we find the best linear combination with maximum effectiveness?

In the equation for the noise the triple (P, C, K) is involved, according to equation (2). Note that in the 3-rounded DES only the f-function is a non-linear part, whereas all the other operations are linear. So, if we can find the best approximation for (*input*, *output*) of the f-function, then the corresponding approximation involving (P, C, K) of the "tiny" DES will be also the best.

Home-Exercise 4 *For the 3-round DES, find the best linear approximation for which the noise N has maximum bias (*$|\Pr\{N = 0\} - 0.5| \rightarrow$ *max).*

Hints:

1. The best linear approximation for the f-function (see Figure 3) is found to be:

   $$B[26] = f(R_x, K_x)[3, 8, 14, 25], \tag{3}$$

   i.e., the noise $N = B[26] \oplus f(R_x, K_x)[3, 8, 14, 25]$ has the maximum bias and the corresponding probability is $\Pr\{N = 0\} \approx 0.19$ (you may also check this by simulation yourself).

2. First find the relation

   $$R_1[3, 8, 14, 25] \oplus P[\ldots] = K[\ldots],$$

   and then

   $$R_1[3, 8, 14, 25] \oplus C[\ldots] = K[\ldots].$$

   The sum of these two equations gives us the required relation.

□

Lab-Exercise 2 *For the linear approximation that you have derived in the previous home exercise 4, find the bias of the noise by simulation. For this purpose, make a sufficient number (*$\approx 10^5$*) of random selections of the pair (P, K), then calculate the corresponding ciphertext C. Count the number of times $N = 0$ and calculate an estimate of* $\Pr\{N = 0\}$ *(which is the same as the probability that the derived linear combination holds). Give the bias of the noise in the form* $\text{bias}(N) = 1/2 \pm \epsilon$*. If everything is done correctly, the bias should be significantly different from* 0.5*! To check your work, as an intermediate simulation you may test the bias for equation* (3)*(try different inputs for the function f, when K is fixed. The bias should be 0.19).*

□