Block Ciphers

Data Encryption Standard

The Data Encryption Standard, or DES, is one of the most important examples of a Feistel cryptosystem. DES was the result of a contest set by the U.S. National Bureau of Standards (now called the NIST) in 1973, and adopted as a standard for unclassified applications in 1977.

The winning standard was developed at IBM, as a modification of the previous system called LUCIFER. The DES is widely used for encryption of PIN numbers, bank transactions, and the like. DES is also specified as an Australian banking standard.

The DES is an example of a Feistel cipher, which operates on blocks of 64 bits at a time, with an input key of 64 bits. Every 8th bit in the input key is a parity check bit which means that in fact the key size is effectively reduced to 56 bits.

Advanced Encryption Standard

In 1997, the NIST called for submissions for a new standard to replace the aging DES. The contest terminated in November 2000 with the selection of the Rijndael cryptosystem as the Advanced Encryption Standard (AES).

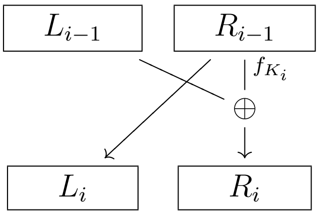
Product ciphers and Feistel ciphers

As a precursor to the description of DES, we make the following definitions, which describe various aspects of the constructions, specific properties, and design components of DES.

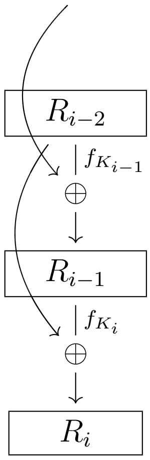
A *product cipher* is a composite of two or more elementary ciphers with the goal of producing a cipher which is more secure that any of the individual components. A *substitution-permutation network* is a product cipher composed of stages, each involving substitutions and permutations, in which the blocks can be partitioned into smaller blocks for substitutions and recombined with permutations. An *iterated block cipher* is a block cipher involving the repetition of an internal *round function*, which may involve a key as input. Each of the sequential steps is termed a *round.*

We now describe in more detail an example of an iterated block cipher, called a *Feistel cipher.* In a Feistel the input block is of even length $2t$, of the form L_0R_0, and outputs ciphertext of the form R_rL_r. For each i such that $1 \le i \le r$, the round map takes $L_{i-1}R_{i-1}$ to L_iR_i, where $L_i = R_{i-1}$ and $R_i = L_{i-1} \oplus f_{K_i}(R_{i-1})$, where f_{K_i} is a cipher which depends only on an input subkey K_i, which is derived from the cipher key K.

The flow of the Feistel cipher therefore looks something like:



We can eliminate the L_i by defining $R_{-1} = L_0$, so that the input is $R_{-1}R_0$, and the round operations are of the form $R_i = R_{i-2} \oplus f_{K_i}(R_{i-1})$, in which case the flow diagram looks like:



The final output of the Feistel cipher is the inverted pair $R_rL_r = R_rR_{r-1}$, which allows the Feistel cipher to be inverted by running through the same algorithm with the key sequence reversed.

Exercise. Verify that reversing the internal key sequence gives the inverse cipher.

Proof. We prove this exercise for the convenience of the reader, by a comparison of the enciphering and deciphering sequences $\{R_i\}$ and $\{R'_j\}$.

Enciphering. A message $M = L_0R_0 = R_{-1}R_0$, is enciphered via the iteration:

$$R_{i+1} = R_{i-1} \oplus f_{K_{i+1}}(R_i), \tag{1}$$

with respect to a key sequence $K_1, K_2, \ldots, K_r$.

Deciphering. Suppose we begin with $C = R_rR_{r-1} = R'_{-1}R'_0$, and a reversed key sequence $K'_1, K'_2 \ldots, K'_r = K_r, K_{r-1} \ldots, K_1$. The deciphering follows the same algorithm as enciphering with respect to this key sequence:

$$R'_{j+1} = R'_{j-1} \oplus f_{K'_{j+1}}(R'_j). \tag{2}$$

Setting $j = r - i - 1$, we have $K'_{j+1} = K'_{r-i} = K_{i+1}$. We moreover want to show the relations

$$R'_{-1} = R_r,\ R'_0 = R_{r-1}, \ldots, R'_{r-1} = R_0,\ R'_r = R_{-1}.$$

In other words, we want to show that $R'_j = R_i$ whenever $i + j = r - 1$.

Clearly this relation holds for $(i, j) = (r, -1)$ and $(i, j) = (r-1, 0)$. Assuming it holds for $j - 1$ and j we prove that it holds for $j + 1$. The deciphering sequence (2) can be replaced by

$$R'_{j+1} = R'_{j-1} \oplus f_{K'_{j+1}}(R'_j) = R'_{r-i-2} \oplus f_{K'_{r-i}}(R'_{r-i-1}) = R_{i+1} \oplus f_{K_{i+1}}(R_i)$$

The expression $R_{i+1} = R_{i-1} \oplus f_{K_{i+1}}(R_i)$ in (2) can be rearranged by adding (= substracting) $f_{K_{i+1}}(R_i)$ to both sides to get $R_{i+1} \oplus f_{K_{i+1}}(R_i) = R_{i-1}$. We conclude that $R'_{j+1} = R_{i-1}$, so the equality holds by induction.

Example of a Feistel Cipher

Let f_{K_i} be the block cipher, of block length 4, which is the composition of the following maps:

1. The transposition cipher $T = [4, 2, 1, 3]$; followed by
2. A bit-sum with the 4-bit key K_i; followed by
3. A substitution cipher S applied to the 2-bit blocks

$$S(00) = 10, \quad S(10) = 01, \quad S(01) = 11, \quad S(11) = 00,$$

i.e. $b_1b_2b_3b_4 \mapsto S(b_1b_2)S(b_3b_4)$.

Let C be the 3-round Feistel cryptosystem of key length 12, where the three internal keys K_1, K_2, K_3 are the first, second, and third parts of the input key K, and the round function is f_{K_i}.

Exercise. Compute the enciphering of the text $M = 11010100$, using the key $K = 001011110011$.

Overview of the Digital Encryption Standard

The DES is a 16-round Feistel cipher, which is preceeded and followed by an initial permutation IP and its inverse IP^{-1}. That is, we start with a message M, and take $L_0R_0 = IP(M)$ as input to the Feistel cipher, with output $IP^{-1}(R_{16}L_{16}$. The 64-bits of the key are used to generate 16 internal keys, each of 48 bits. The steps of the round function f_K is given by the following sequence, taking on 32-bit strings, expanding them to 48-bit strings, and applying a 48-bit block function.

1. Apply a fixed *expansion permutation* E — this function is a permutation the 32 bits with repetitions to generate a 48-bit block $E(R_i)$.

2. Compute the bit-sum of $E(R_i)$ with the 48-bit key K_i, and write this as 8 blocks $B_1, \dots, B_8$ of 6 bits each.

3. Apply to each block $B_j = b_1b_2b_3b_4b_5b_6$ a substitution S_j. These substitutions are specified by *S-boxes*, which describe the substitution as a look-up table. The output of the substitution cipher is a 4-bit string C_j, which results in the 32-bit string $C_1C_2C_3C_4C_5C_6C_7C_8$.

4. Apply a fixed 32-bit permutation P to $C_1C_2C_3C_4C_5C_6C_7C_8$, and output the result as $f_{K_i}(R)$.

This completes the description of the round function f_{K_i}.

Overview of the Advanced Encryption Standard

In 1997, the NIST called for submissions for a new standard to replace the aging DES. The contest terminated in November 2000 with the selection of the Rijndael cryptosystem as the Advanced Encryption Standard (AES).

The Rijndael cryptosystem operates on 128-bit blocks, arranged as 4×4 matrices with 8-bit entries. The algorithm consists of multiple iterations of a round cipher, each of which is the composition of the following four basic steps:

- *ByteSub* transformation. This step is a nonlinear substition, given by a S-box (look-up table), designed to resist linear and differential cryptanalysis.
- *ShiftRow* transformation. Provides a linear mixing for diffusion of plaintext bits.
- *MixColumn* transformation. Provides a similar mixing as in the ShiftRow step.
- *AddRoundKey* transformation. Bitwise `XOR` with the round key.

The Advanced Encryption Standard allows Rhijndael with key lengths 128, 192, or 256 bits.

The eight-bit byte blocks which form the matrix entries are interpretted as elements of the finite field of $2^8 = 256$ elements. The finite field is represented by the quotient ring

$$\mathbb{F}_{2^8} = \mathbb{F}_2[X]/(X^8 + X^4 + X^3 + X + 1),$$

whose elements are polynomials $c_7X^7 + c_6X^6 + c_5X^5 + c_4X^4 + c_3X^3 + c_2X^2 + c_1X + c_0$.

We denote by BS, SR, MC, and ARK these four basic steps. There exist corresponding inverse operations IBS, ISR, IMC, IARK. The flow of the algorithms for enciphering and deciphering are as follows:

1. ARK	**1.** ARK
2. BS, SR, MC, ARK	**2.** IBS, ISR, IMC, IARK
⋮	⋮
3. BS, SR, MC, ARK	**3.** IBS, ISR, IMC, IARK
4. BS, SR, ARK	**4.** IBS, ISR, ARK

ByteSub. The ByteSub operation is given by the S-box look-up table. Alternatively the S-box has a description in terms of the structure of the finite fields and linear algebra. Let x' be the inverse of x in $\mathbb{F}_{2^8}$ if $x \neq 0$ and set $x' = x = 0$ otherwise. Then the ByteSub step is given by $x \mapsto X^6 + X^5 + X + 1 + x'A$ where A is the matrix:

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