

Lecture 7: Block Cipher Principle

Yuan Xue

I. SYMMETRIC ENCRYPTION PRINCIPLES

This lecture discusses the principles of all known contemporary symmetric key cryptosystems. All these systems have evolved from early classical ciphers discussed in the previous lectures. As we have seen, these classical ciphers may operate in the following two ways.

- *Stream cipher*, such as Vigenère cipher, encrypts one letter at a time.
- *Block cipher*, such as Hill cipher, treats a n -letter block of plaintext as a whole and produce a ciphertext block of equal length.

A. Block Cipher Principles

As block cipher have different modes of operation (we will discuss this topic later in this lecture) and applies to a broader range of applications than stream cipher, we will focus on its design principles in this lecture.

A block cipher transform a plaintext block of n letters into an encrypted block. For the alphabet with 26 letters, there are 26^n possible different plaintext blocks. The most general way of encrypting a n -letter block is to take each of the plaintext blocks and map it to a cipher block (arbitrary n -letter substitution cipher). For decryption to be possible, such mapping needs to be one-to-one (*i.e.*, each plaintext block must be mapped to a unique ciphertext block). The number of different one-to-one mappings among n -letter blocks is $(26^n)!$.

The length of block n can not be too short in order to secure the cryptographic scheme. For example, $n = 1$ gives a monoalphabetic cipher. Such schemes, as we have seen, are vulnerable to frequency analysis and brute-force attacks. However, an arbitrary reversible substitution cipher for a large block size n is not practical. Let's consider the problem of specifying a mapping of all possible n -letter blocks. In a cipher, each key specifies such a mapping. Let's assume the key consists of a block of k letters. Then the number of all possible keys is 26^k . Then for a n -letter arbitrary substitution block cipher, the key size needs to satisfy $26^k \geq (26^n)!$, *i.e.*, $k \geq n \times 26^n!$.

So the major challenge to design a symmetric key cryptographic scheme is to provide enough security (e.g., using a reasonable large block size) with a reasonable small size key¹.

¹It is fairly obvious that the key length can not be too short either. Otherwise the cryptographic scheme would also be vulnerable to brute-force attack where the attackers may search through all possible keys.

However, how do we know that a cryptographic system is secure enough? To answer this question, Claude Shannon theoretically deduced the following principles that should be followed to design secure cryptographic systems. These principles aim at thwarting cryptanalysis based on known statistical properties of the plaintext.

- *Confusion.* In Shannon's original definitions, confusion makes the relation between the key and the ciphertext as complex as possible. Ideally, every letter in the key influences every letter of the ciphertext block. Replacing every letter with the one next to it on the typewriter keyboard is a simple example of confusion by substitution. However, good confusion can only be achieved when each character of the ciphertext depends on several parts of the key, and this dependence appears to be random to the observer. Ciphers that do not offer much confusion (such as Vigenère cipher) are vulnerable to frequency analysis.
- *Diffusion.* Diffusion refers to the property that the statistics structure of the plaintext is dissipated into long range statistics of the ciphertext. In contrast to confusion, diffusion spreads the influence of a single plaintext letter over many ciphertext letters. In terms of the frequency statistics of letters, digrams, etc in the plaintext, diffusion randomly spreads them across several characters in the ciphertext. This means that much more ciphertexts are needed to do a meaningful statistical attack on the cipher.

B. The Feistel Network

Product ciphers use the two classical encryption forms: substitution and transposition, alternatively in multiple rounds to achieve both confusion and diffusion respectively. Shannon was the first to investigate the product cryptosystem (so called substitution-permutation network) and show that some sophisticated heuristic ciphers were nothing other than products of some simpler ciphers. Most importantly, Shannon identified the necessary condition of the cipher strength increases as a result of cascading simple ciphers.

One possible way to build a secret key algorithm using substitution-permutation-network is to break the input into manageable-sized chunks, do a substitution on each small chunk, and then take the outputs of all the substitutions and run them through a permuter that is as big as the input, which shuffles the letters around. Then the process is repeated, so that each letter winds up as an input to each of the substitutions.

Since modern cryptosystems are all computer-based, from now on we will assume that both plain and cipher text are strings of bits ($\{0, 1\}$), instead of strings of letters ($\{a, b, c, \dots, z\}$).

The Feistel network shown in Fig. 1 is a particular form of the substitution-permutation network. The input to a Feistel network is a plaintext block of n bits, and a key K . The plaintext block is divided into

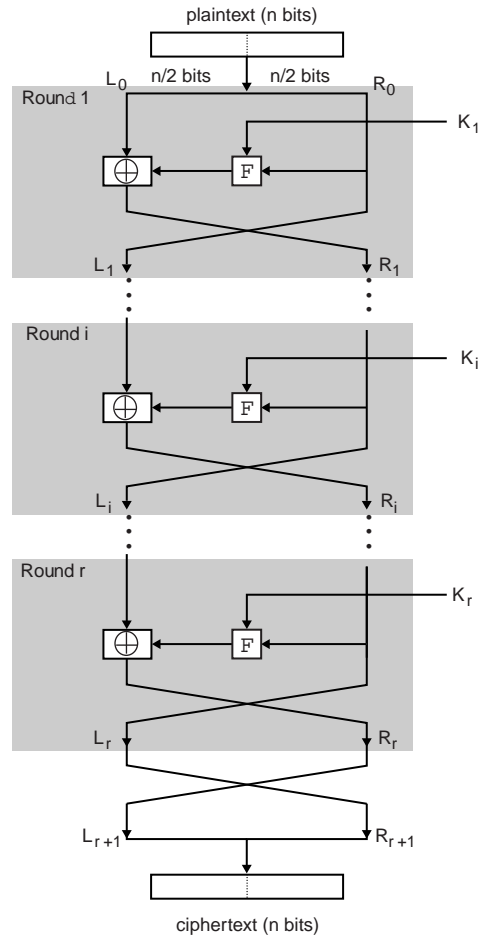


Fig. 1. Feistel Network.

two halves, L_0 and R_0 . The two halves of the data pass through r rounds of processing and then combine to produce the ciphertext block. Each round i has as input L_{i-1} and R_{i-1} , derived from the previous round, as well as a subkey K_i , derived from the overall key K . In general, the subkey K_i are different from K and from each other. In this structure, a substitution is performed via the round function F , and permutation is performed that interchanges the two halves of the data.

The exact realization of a Feistel network depends on the choices of the following parameters and design features.

- Block size: Larger block size means greater security, but reduces encryption/decryption speed.
- Key size: Larger key size means greater security but may decrease encryption/decryption speed.
- Number of rounds: Multiple rounds offer increasing security.
- Subkey generation algorithm: Greater complexity in subkey generation leads to greater security.

- Round function: Greater complexity in round function means greater difficulty of cryptanalysis.

We will illustrate these design choices using DES (Data Encryption Standard) as an example in the next lecture.

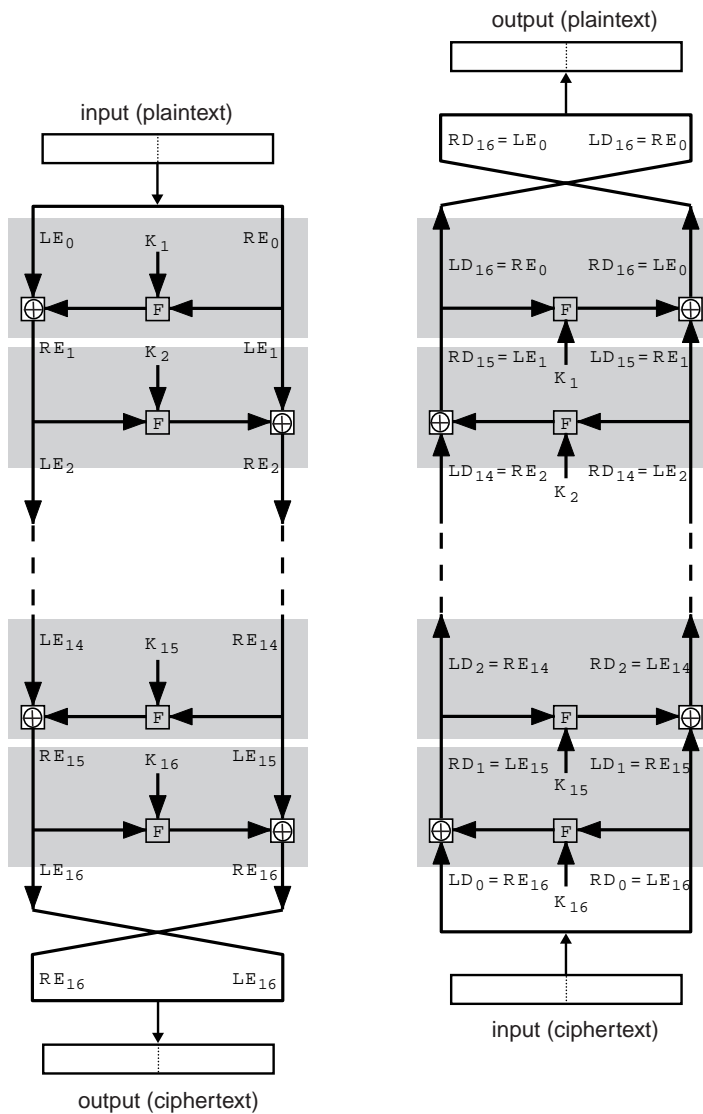


Fig. 2. The encryption and decryption of Feistel network

It is worth noting that the process of decryption with a Feistel network is essentially the same as the encryption process by using the ciphertext as input to the network, but using the subkey K_i in reverse order, as shown in Fig 2. The reason is explained as follows. Let's consider the last step in encryption, which gives,

$$LE_{16} = RE_{15} \quad (1)$$

$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16}) \quad (2)$$

On the decryption side,

$$LD_1 = RD_0 = LE_{16} = RE_{15} \quad (3)$$

$$RD_1 = LD_0 \oplus F(RD_0, K_{16}) \quad (4)$$

$$= RE_{16} \oplus F(RE_{15}, K_{16}) \quad (5)$$

$$= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16}) \quad (6)$$

$$= LE_{15} \quad (7)$$

The process can be done iteratively. Finally, we will see that the output of the decryption is the same as the input to the encryption (*i.e.*, original plaintext).