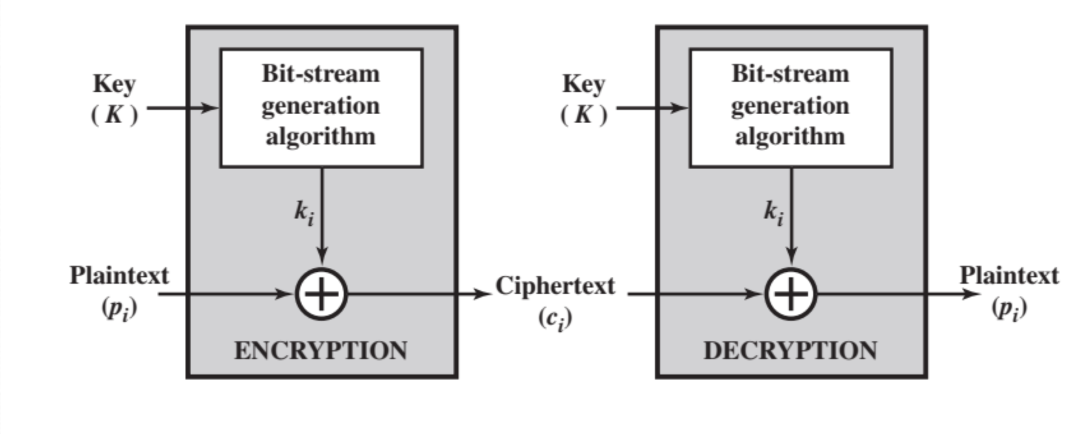
Symmetric key Encryption

Stream Cipher

- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time.
- Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher.
- The keystream (*ki*) is as long as the plaintext bit stream (*pi*) in one-time pad version of the Vernam cipher.
- If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream.

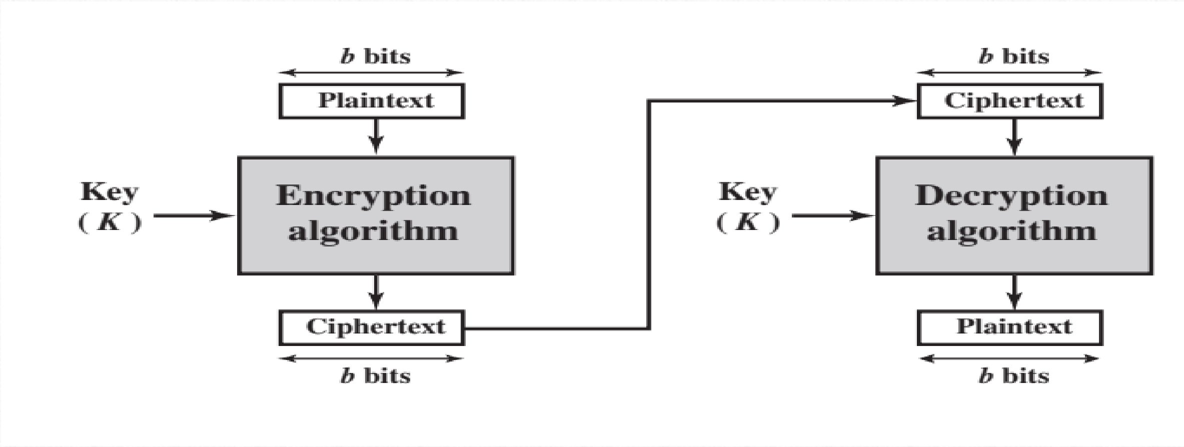
Stream Cipher



Block Cipher

- A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- A block size of 64 or 128 bits is used.
- Using some of the modes of operations, a block cipher can be used to achieve the same effect as a stream cipher.

Block Cipher



Fiestel Cipher

- Feistel proposed the use of a cipher that alternates substitutions and permutations.
- **Substitution:** Each plaintext element or group of elements are uniquely replaced by a corresponding ciphertext element or group of elements.
- **Permutation:** A sequence of plaintext elements are replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

Confusion and Diffusion

- Introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system.
- **Diffusion –** the statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext. Each plaintext digit affect the value of many ciphertext digits.
- **Confusion-** makes the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key.

Feistel Cipher

- **Feistel cipher** refers to a type of block cipher design, not a specific cipher
- Split plaintext block into left and right halves: Plaintext = (L_0,R_0)
- For each round $i=1,2,\ldots,n$, compute

  $L_i = R_{i-1}$

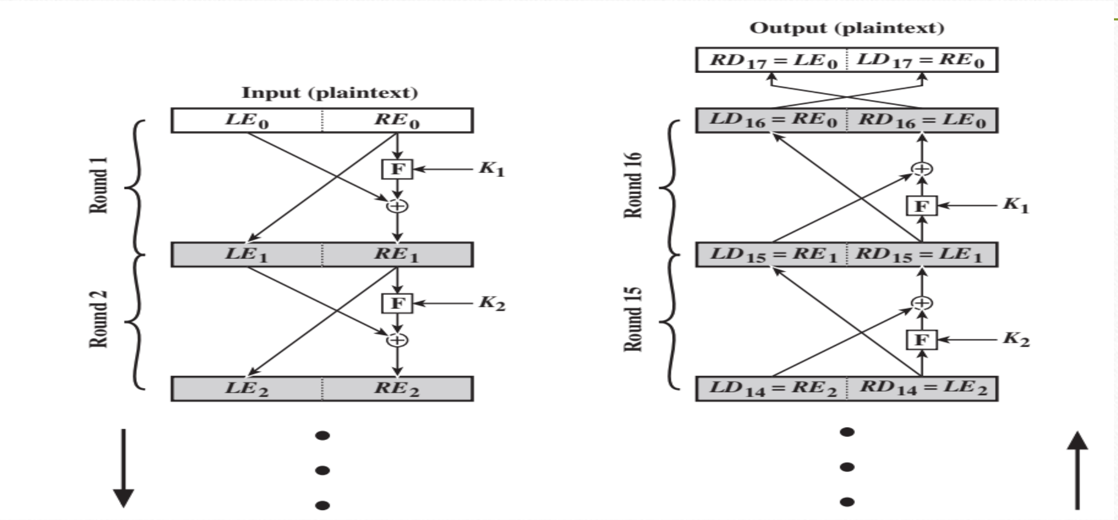
  $R_i = L_{i-1} \oplus F(R_{i-1},K_i)$

  where F is **round function** and K_i is **subkey**
- Ciphertext = (L_n,R_n)

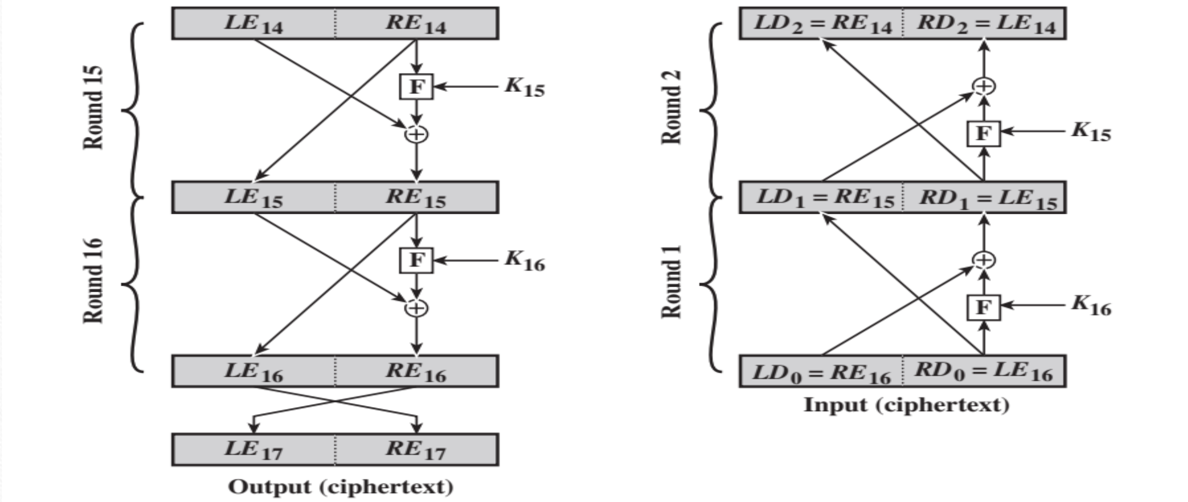
Feistel Cipher

- Decryption: Ciphertext = (L_n,R_n)
- For each round $i=n,n-1,\ldots,1$, compute
  $R_{i-1} = L_i$
  $L_{i-1} = R_i \oplus F(R_{i-1},K_i)$
  where F is round function and K_i is subkey
- Plaintext = (L_0,R_0)
- Formula "works" for any function F
- But only secure for certain functions F

Fiestel Cipher Encryption and Decryption



Fiestel Cipher Encryption and Decryption



Fiestel Encryption and Decryption

- Encryption

$$
\begin{aligned}
LE_{16} &= RE_{15} \\
RE_{16} &= LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})
\end{aligned}
$$

- Decryption

$$
\begin{aligned}
LD_1 &= RD_0 \; = LE_{16} = RE_{15} \\
RD_1 &= LD_0 \oplus \mathrm{F}(RD_0, K_{16}) \\
&= RE_{16} \oplus \mathrm{F}(RE_{15}, K_{16}) \\
&= [LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})] \oplus \mathrm{F}(RE_{15}, K_{16})
\end{aligned}
$$

Fiestel Encryption and Decryption

- *i*th iteration of the encryption algorithm

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus \mathrm{F}(RE_{i-1}, K_i)$$

- Rearranging terms

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus \mathrm{F}(RE_{i-1}, K_i) = RE_i \oplus \mathrm{F}(LE_i, K_i)$$