Cryptography and Network Security

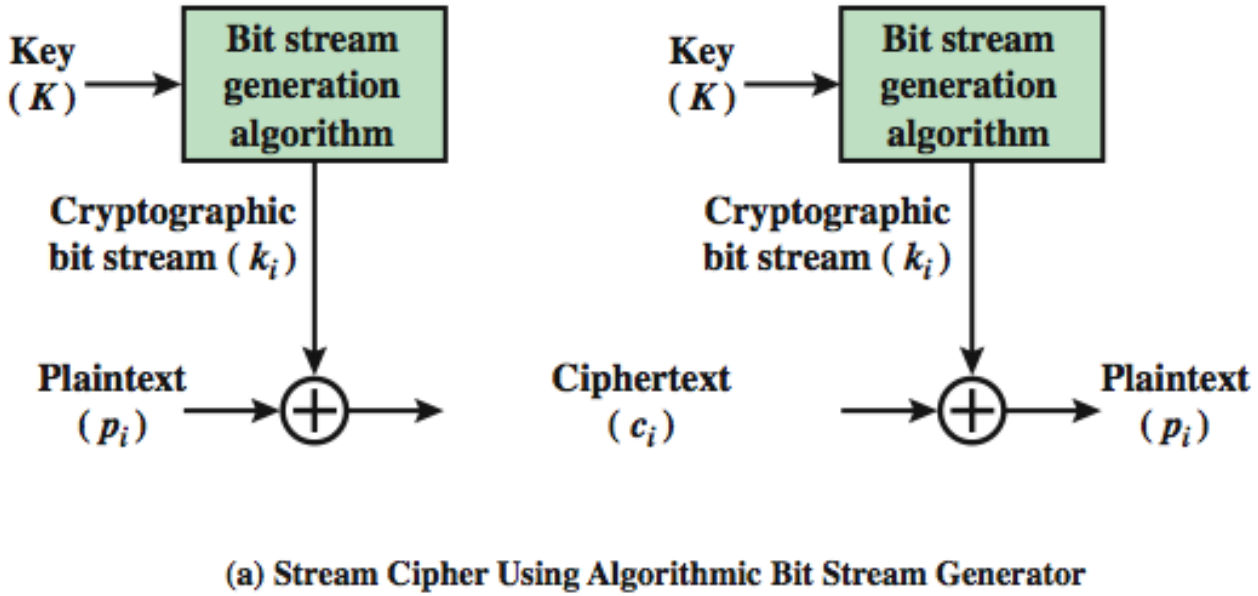
Lecture 03 – Block Ciphers

Ediz ŞAYKOL

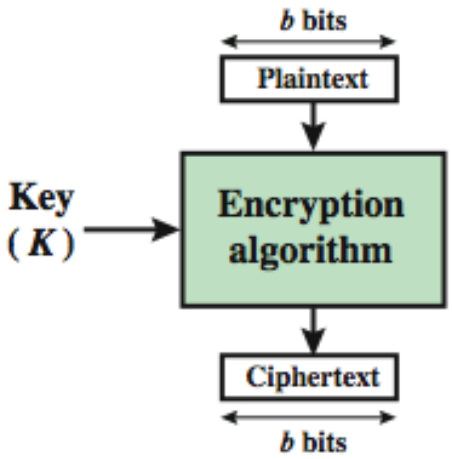
Block vs Stream Ciphers

- block ciphers process messages in blocks,
  - each of which is then en/decrypted
- like a substitution on very big characters
  - 64-bits or more
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers
  - better analysed
  - broader range of applications

Block vs Stream Ciphers



(a) Stream Cipher Using Algorithmic Bit Stream Generator

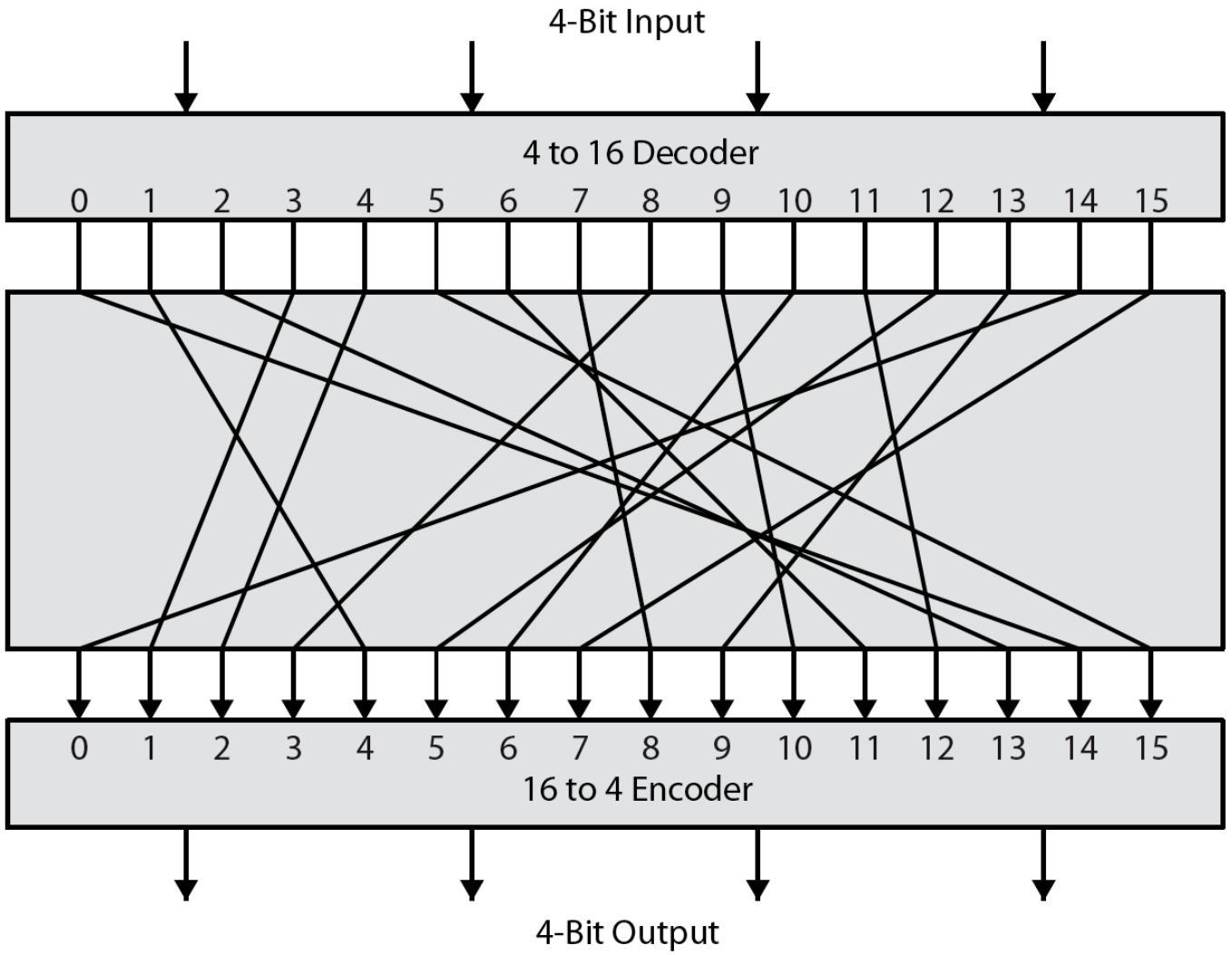


(b) Block Cipher

Block Cipher Principles

- most symmetric block ciphers are based on a **Feistel Cipher Structure**
  - A block cipher operates on a plaintext block of n bits to produce a ciphertext block of n bits.
- block ciphers have an extremely large substitution
  - for an *n*-bit general substitution block cipher, the size of the key is $n \times 2^n$
  - would need table of 2^{64} entries for a 64-bit block, the key size is 64 (bits) x 2^{64} (rows) = 2^{70} bits
- Feistel: instead create from smaller building blocks

Ideal Block Cipher



Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- form basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* & *diffusion* of message & key

Confusion and Diffusion

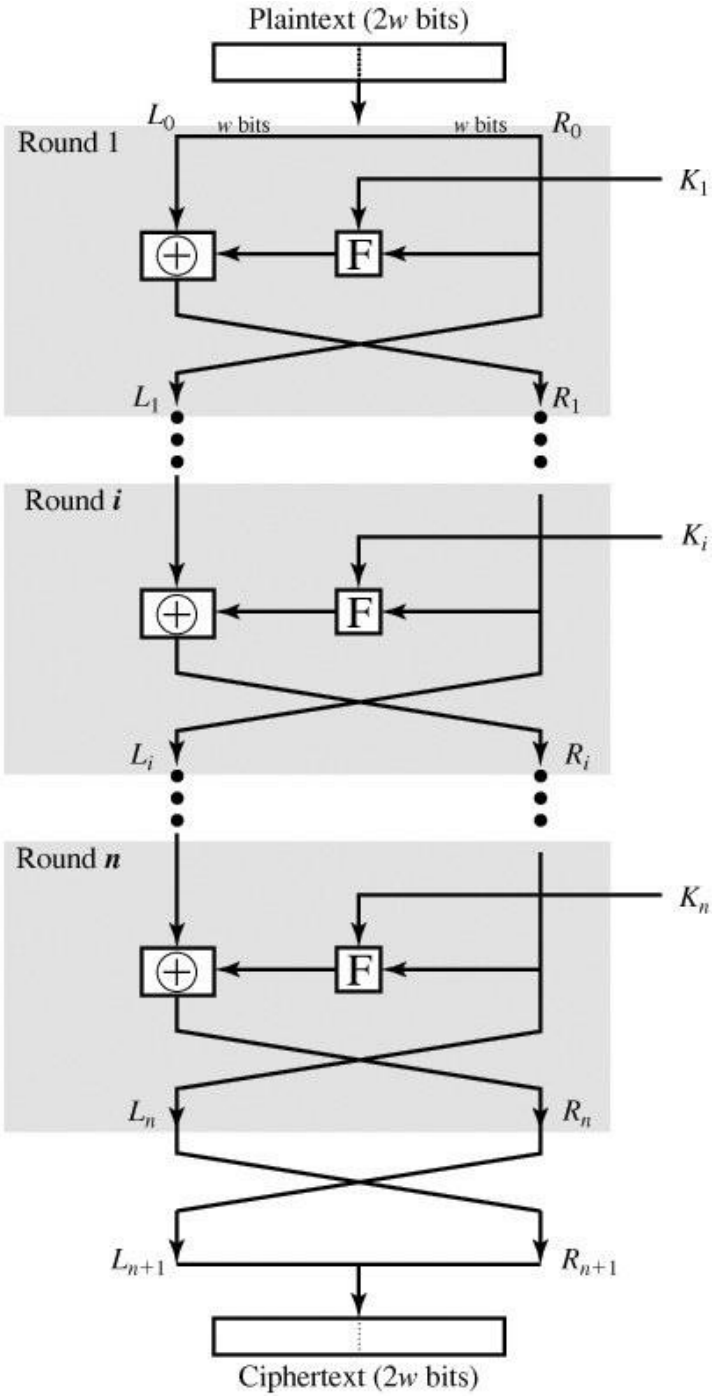
- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S & P elements to obtain:
  - **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
  - **confusion** – makes relationship between ciphertext and key as complex as possible

Feistel Cipher Structure

- Feistel devised the **feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into 2 halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon's S-P net concept

Feistel Cipher (Encryption)

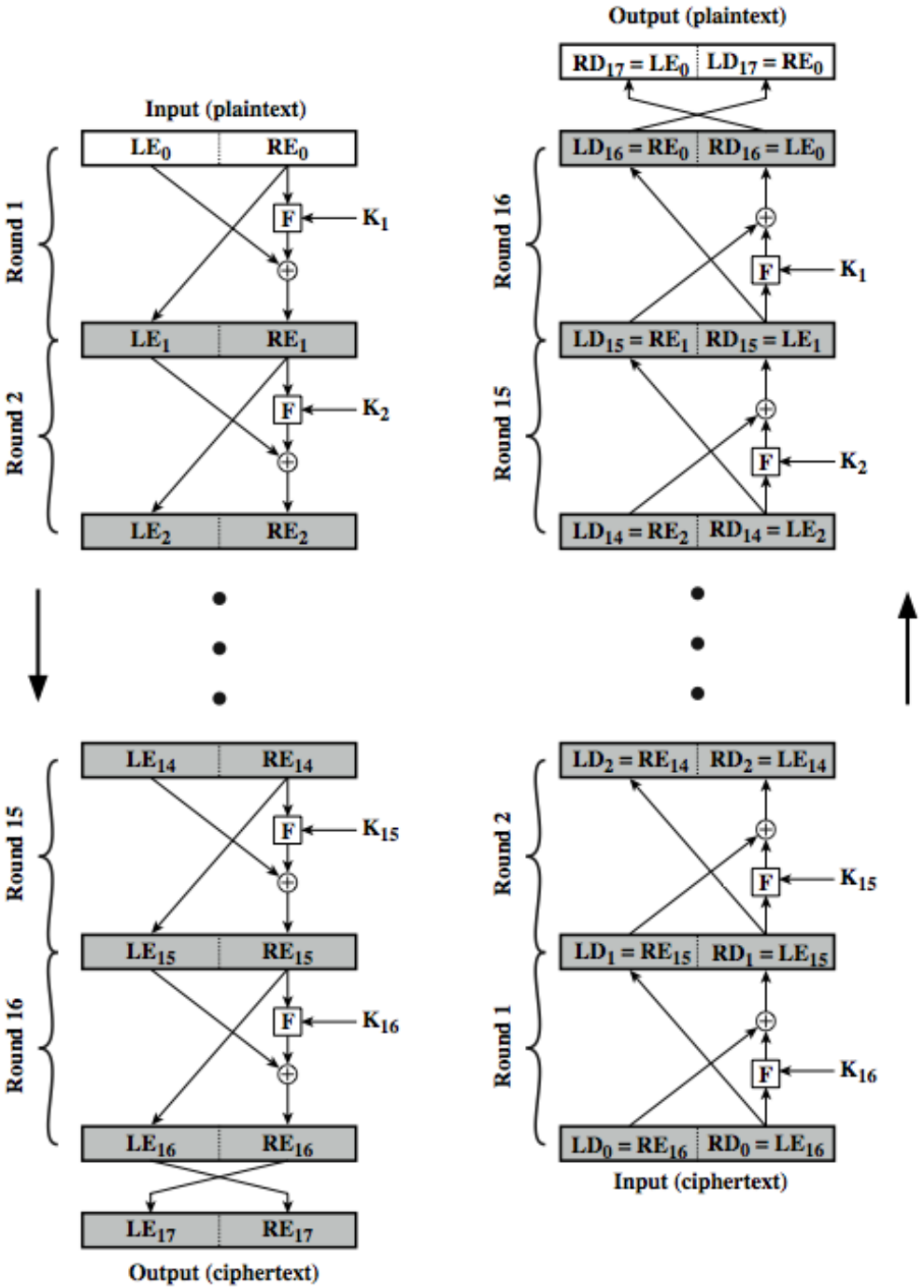
- The inputs to the encryption algorithm are a plaintext block of length 2w bits and a key K.
- The plaintext block is divided into two halves, L_0 and R_0.
- The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block.
- Each round i has as inputs L_{i-1} and R_{i-1}, derived from the previous round, as well as a subkey K_i, derived from the overall K.
- In general, the subkeys K are different from K and from each other.



Feistel Cipher (Enc & Dec)

- The rule is as follows: Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order.

- That is, use K_n in the first round, K_{n-1} in the second round, and so on until K_1 is used in the last round.



Feistel Cipher Design Elements

- block size - increasing size improves security, but slows cipher
- key size - increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- number of rounds - increasing number improves security, but slows cipher
- subkey generation algorithm - greater complexity can make analysis harder, but slows cipher
- round function - greater complexity can make analysis harder, but slows cipher
- fast software en/decryption - more recent concern for practical use
- ease of analysis - for easier validation & testing of strength