Unit VIII

Symmetric Encryption and Message Confidentiality

Agenda

- Encryption: Principles
- Symmetric Block Encryption Algorithms (Data Encryption Standard, Triple DES, Advanced Encryption Standard)
- Random and Pseudorandom Numbers
- Stream Ciphers and RC4
- Cipher Block Modes of Operation
- Public Key Cryptography: Approaches to Message Authentication,
- Secure Hash Functions
- Message Authentication Codes
- Public-Key Cryptography Principles
- Public-Key Cryptography Algorithms (RSA and Diffie-Hellman Exchange)
- Digital Signatures.

Some Basic Terminology

- Plaintext - original message
- Ciphertext - coded message
- Cipher - algorithm for transforming plaintext to ciphertext
- Key - info used in cipher known only to sender/receiver
- Encipher (encrypt) - converting plaintext to ciphertext
- Decipher (decrypt) - recovering ciphertext from plaintext
- Cryptography - study of encryption principles/methods
- Cryptanalysis (code breaking) - study of principles/methods of deciphering ciphertext without knowing key
- Cryptology - field of both cryptography and cryptanalysis
- Decryption:encryption algorithm run in reverse. It produce cipher text + secrete key and generate original text

Symmetric Encryption Principles

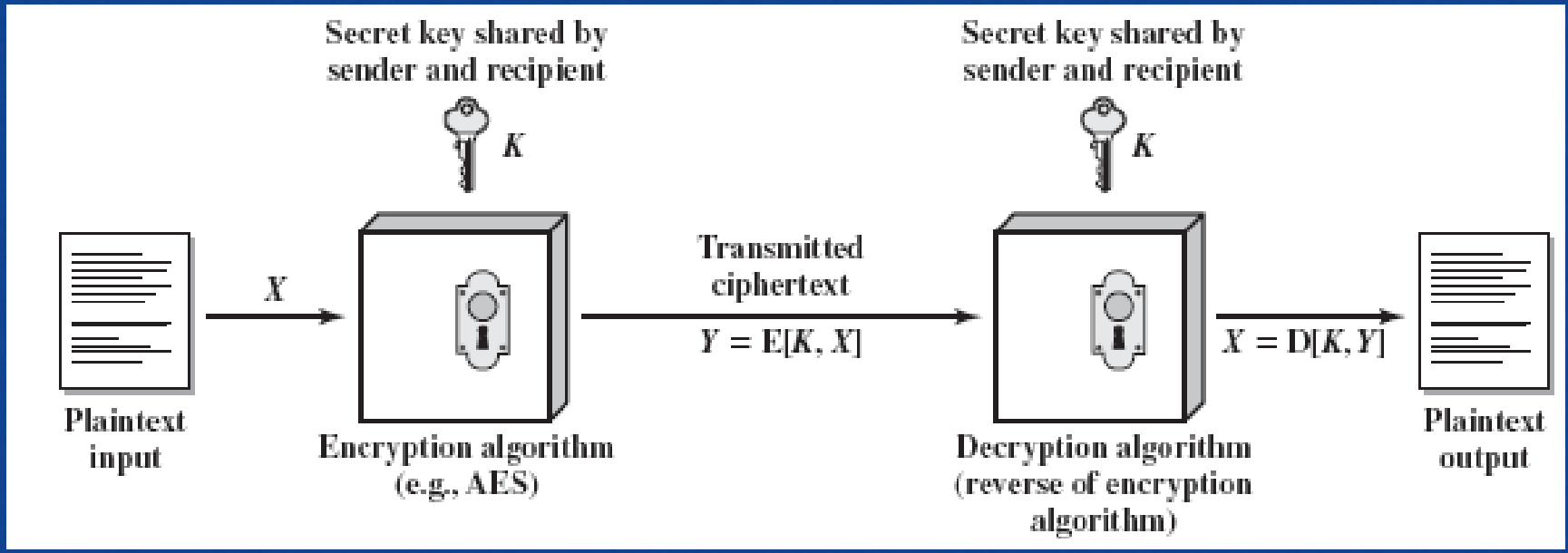
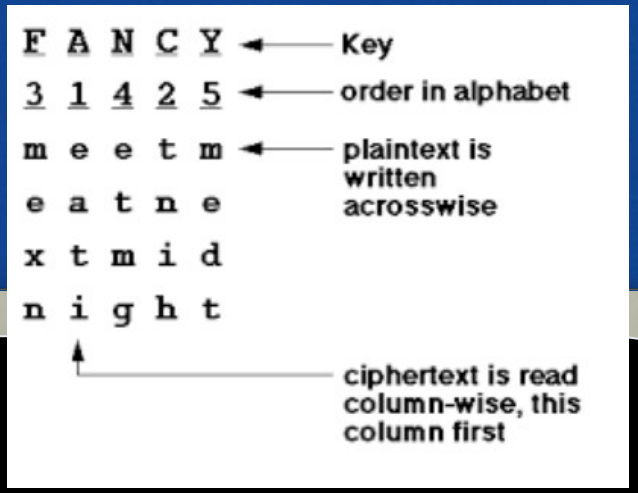


Figure 2.1 Simplified Model of Symmetric Encryption

Requirements

- There are two requirements for secure use of symmetric encryption:
  - A strong encryption algorithm
  - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure
  - That is same key is used for encryption and decryption process.
- The security of symmetric encryption depends on the secrecy of the key, not the secrecy of the algorithm
  - This makes it feasible for widespread use
  -

Cryptography



Cryptographic systems are generically classified along three independent dimensions:

- **The type of operations used for transforming plaintext to ciphertext**
  - Substitution
    - Each element in the plaintext is mapped into another element E.g A is replace by R.
  - Transposition(fig)
    - Elements in the plaintext are rearranged
    - Fundamental requirement is that no
    - information be lost
  - Product systems
    - Involve multiple stages of substitutions and transpositions

Cryptography

Cryptographic systems are generically classified along three independent dimensions:

- **The number of keys used**
  - Referred to as symmetric, single-key, secret-key, or conventional encryption if both sender and receiver use the same key
  - Referred to as asymmetric, two-key, or public-key encryption if the sender and receiver each use a different key
- **The way in which the plaintext is processed**
  - Block cipher processes the input one block of elements at a time, producing an output block for each input block
  - Stream cipher processes the input elements continuously, producing output one element at a time, as it goes along

Cryptanalysis

- Process of attempting to discover the plain text or key is known as cryptanalysis.
- By the assumption that attacker knows the algorithm and One possible attack under these circumstance is the brute-force approach of trying all possible keys. So opponent/attacker relay on analysis of cipher text using various statistical test.
- To use this approach, the opponent must have some general idea of the type of plaintext that is(e.g java file , exe file,english/french language)
- In many cases, however, the analyst has more information. The analyst may be able to capture one or more plain text messages as well as their encryptions.
- Or the analyst may know that certain plain-text patterns will appear in a message.

- For example, a file that is encoded in the Postscript format always begins with the same pattern,
- or there may be a standardized header or banner to an electronic funds transfer message, and so on.
- All of these are examples of known plaintext. With this knowledge, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed

Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext only	•Encryption algorithm •Ciphertext to be decoded
Known plaintext	•Encryption algorithm •Ciphertext to be decoded •One or more plaintext-ciphertext pairs formed with the secret key
Chosen plaintext	•Encryption algorithm •Ciphertext to be decoded •Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	•Encryption algorithm •Ciphertext to be decoded •Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	•Encryption algorithm •Ciphertext to be decoded •Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key •Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

- **Known plain text attack :**

cryptanalyst knows a chunk of plaintext, maybe only a single probable word, and then tries to determine some further chunks of plaintext—or the key and thereby the complete plaintext.
Attacker can analyze relationship between plain text and cipher text.
e.g Frequent words (specific context domain)

- **Chosen cipher -text attack** : attacker can choose some of the cipher message and can gain access to the achieve the plaintext.
- **Chosen plain text attack :** attacker can choose some plaintext and get corresponding cipher text and generate secrete key.

Cryptanalysis

- An encryption scheme is computationally secure if the ciphertext generated by the scheme meets one or both of the following criteria:
  - The cost of breaking the cipher exceeds the value of the encrypted information
  - The time required to break the cipher exceeds the useful lifetime of the information

Cryptanalysis

- Unfortunately, it is very difficult to estimate the amount of effort required to crypt analyze ciphertext successfully.
- However, assuming there are no inherent mathematical weaknesses in the algorithm, then a brute-force approach is indicated, and here we can make some reasonable estimates about costs and time.

Brute Force attack

Brute Force attack

- Involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained
- On average, half of all possible keys must be tried to achieve success
- Unless known plaintext is provided, the analyst must be able to recognize plaintext as plaintext
- To supplement the brute-force approach
  - Some degree of knowledge about the expected plaintext is needed
  - Some means of automatically distinguishing plaintext from garble is also needed

Brute Force attack

- Table 2.2 shows how much time is involved for various key sizes.
- The 56-bit key size is used with the DES (Data Encryption Standard) algorithm.
- For each key size, the results are shown assuming that it takes 1 μs to perform a single decryption, which is a reasonable order of magnitude for today's machines.
- With the use of massively parallel organizations of microprocessors, it may be possible to achieve processing rates many orders of magnitude greater. The final column of
- Table 2.2 considers the results for a system that can process 1 million keys per microsecond. As you can see, at this performance level, DES no longer can be considered computationally secure.

Brute Force attack

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu s$ = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu s$ = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Table 2.2 Average Time Required for Exhaustive Key Search

Feistel Encryption and Decryption

- Symmatric block encryption algorithms (like DES) have structure described by Feistel of IBM in 1973.
- Input is plaintext block of length 2w bits and Key is K
- The plaintext block is divided into two halves, LE0 and RE0 .
- 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 LEi-1 and REi-1 derived from the previous round, as well as a subkey Ki derived from the overall K .
- In general, the subkeys Ki are different from K and from each other and are generated from the key by a subkey generation algorithm.
- 16 rounds are used, although any number of rounds could be implemented.
- The right-hand side of Figure shows the decryption process.

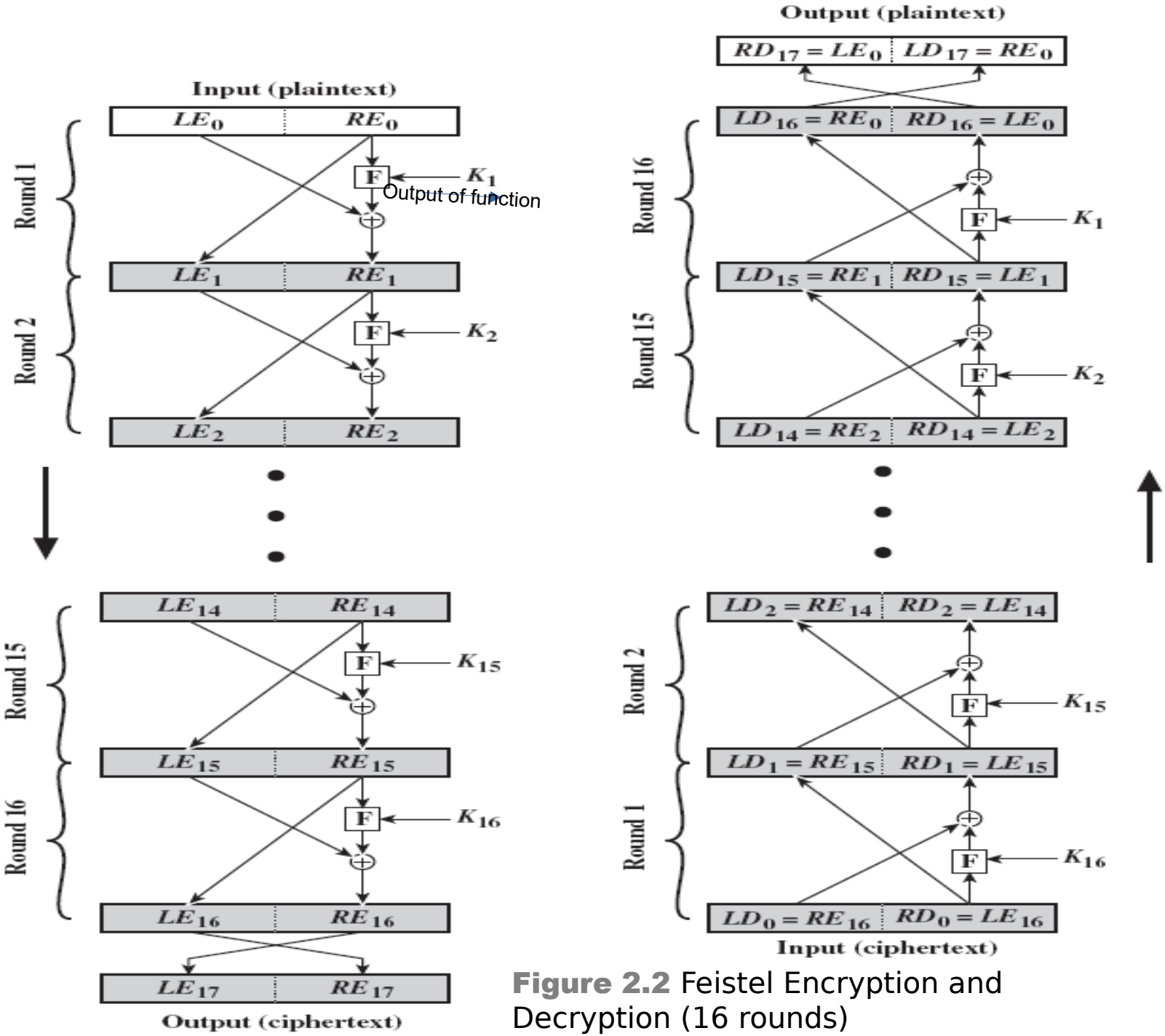


Figure 2.2 Feistel Encryption and Decryption (16 rounds)

Feistel Encryption and Decryption

- All rounds have the same structure. A substitution is performed on the left half of the data.
- This is done by applying a round function F to the right half of the data and then taking the exclusive-OR (XOR) of the output of that function and the left half of the data.
- The round function has the same general structure for each round but is parameterized by the round subkey Ki .
- Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.

Feistel Encryption and Decryption

- The Feistel structure is a particular example of the more general structure
- used by all symmetric block ciphers.
- In general, a symmetric block cipher consists of a sequence of rounds, with each round performing substitutions and permutations conditioned by a secret key value.
- The exact realization of a symmetric block cipher depends on the choice of the following parameters and design features.

Feistel Encryption and Decryption

- **Block size**: Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed. A block size of 128 bits is a reasonable trade-off and is nearly universal among recent block cipher designs.
- **Key size:** Larger key size means greater security but may decrease encryption/decryption speed. The most common key length in modern algorithms is 128 bits.
- **Number of rounds**: The essence of a symmetric block cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.
- **Subkey generation algorithm**: Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- **Round function**: Again, greater complexity generally means greater resistance to cryptanalysis.(encryption algorithm)

Feistel Cipher Design Elements

Block size

- Larger block sizes mean greater security but reduced encryption/decryption speed

Key size

- Larger key size means greater security but may decrease encryption/decryption speed

Number of rounds

- The essence of a symmetric block cipher is that a single round offers inadequate security but that multiple rounds offer increasing security

Subkey generation algorithm

- Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis

Round function

- Greater complexity generally means greater resistance to cryptanalysis

Fast software encryption/decryption

- In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the seed of execution of the algorithm becomes a concern

Ease of analysis

- If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

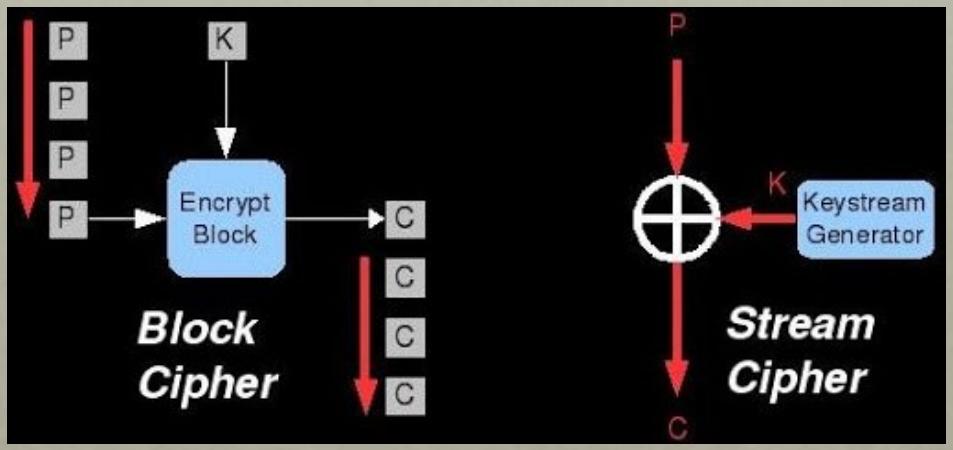
Feistel Encryption and Decryption

- There are two other considerations in the design of a symmetric block cipher:
- **Fast software encryption/decryption**: In many cases, encryption is embedded in applications or utility functions in such a way as to preclude(prevent to make) a hardware implementation. This is making the speed of execution of the algorithm becomes a concern.
- **Ease of analysis**: Although we would like to make our algorithm as difficult as possible to cryptanalyse, there is great benefit in making the algorithm easy to analyse .
- That is, if the algorithm can be concisely and clearly explained, it is easier to analyse that algorithm for cryptanalytic vulnerabilities.
- DES, for example, does not have an easily analysed functionality.

SYMMETRIC BLOCK ENCRYPTION ALGORITHMS

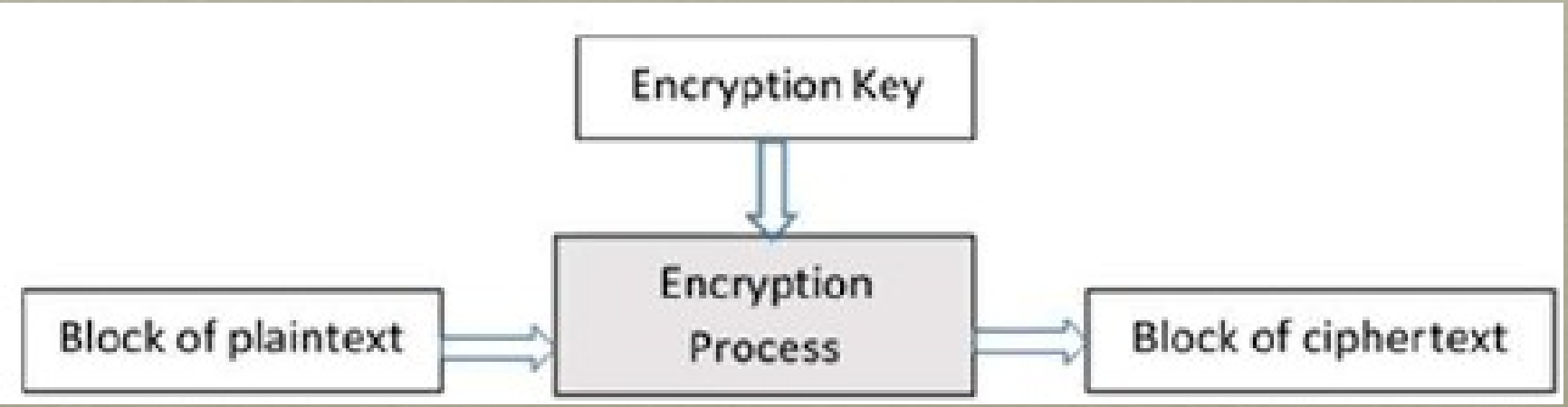
- The most commonly used symmetric encryption algorithms are block ciphers.
- A block cipher processes the plaintext input in fixed-sized blocks and produces a block of ciphertext of equal size for each plaintext block.
- This section focuses on the three most important symmetric block ciphers:
  - The Data Encryption Standard (DES),
  - Triple DES (3DES), and
  - The Advanced Encryption Standard (AES).

Block and stream cipher

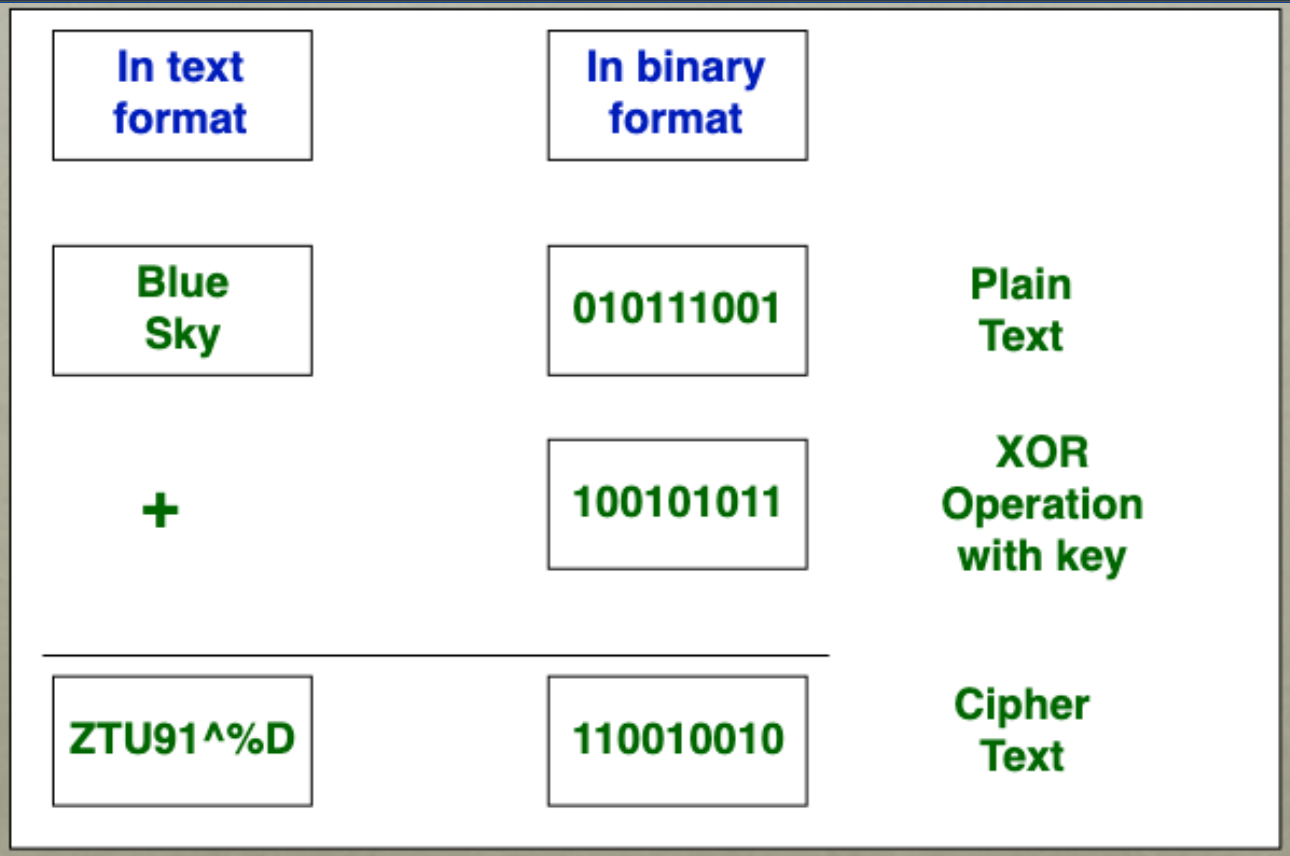


Block Cipher

Stream Cipher

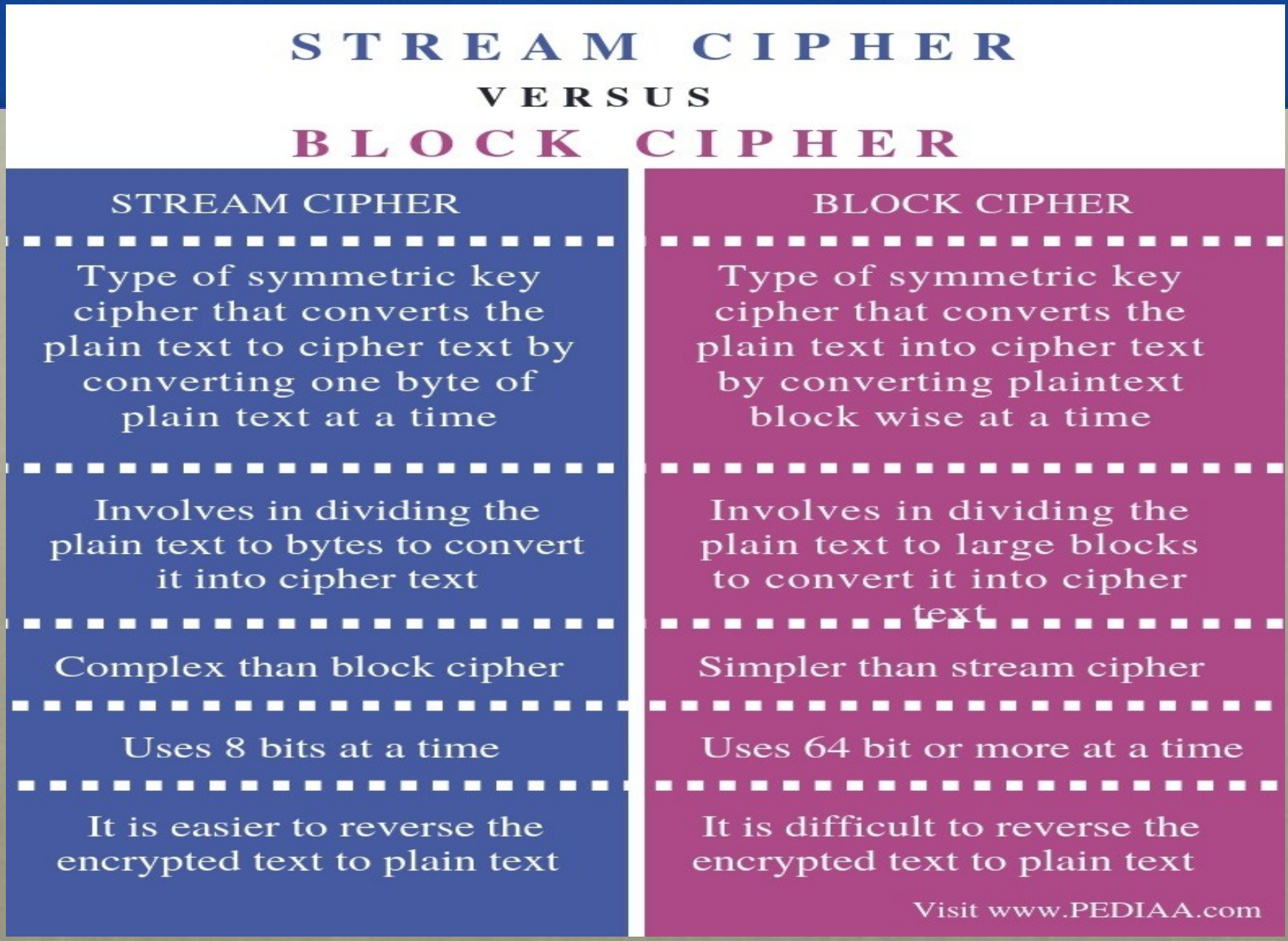

Encryption Key
Block of plaintext
Encryption Process
Block of ciphertext

Stream cipher



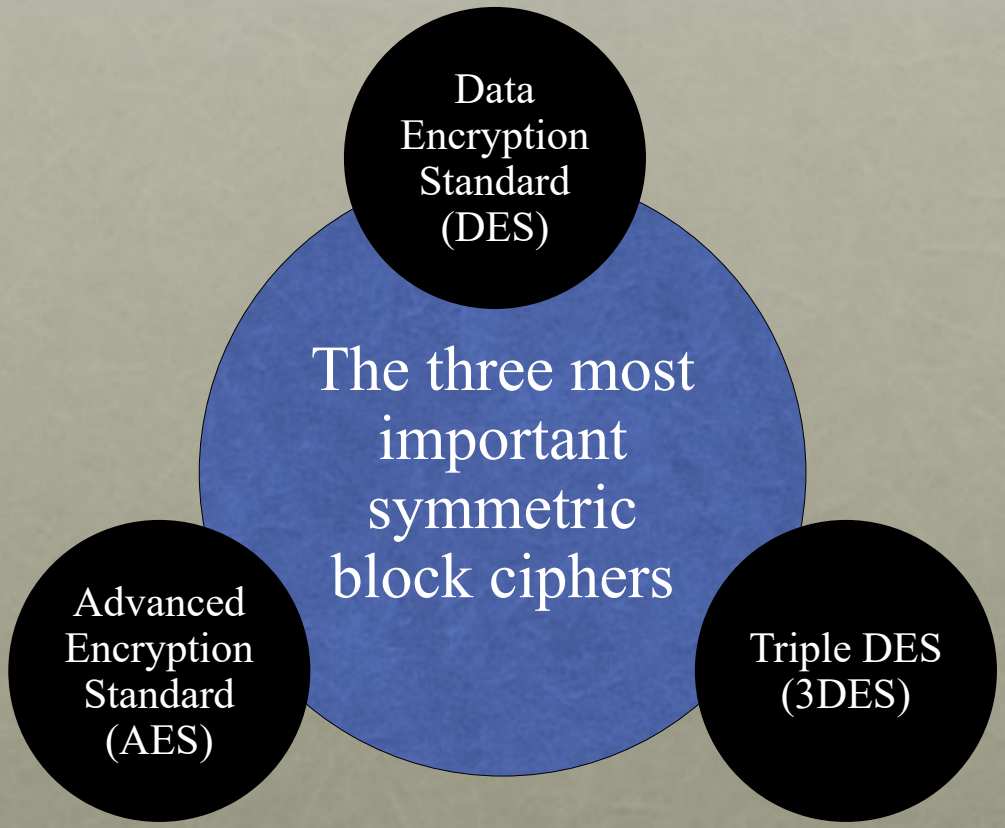
Stream Cipher

Block and stream cipher



Symmetric Block encryption algorithms

- Block cipher
  - The most commonly used symmetric encryption algorithms
  - Processes the plaintext input in fixed-sized blocks and produces a block of ciphertext of equal size for each plaintext block

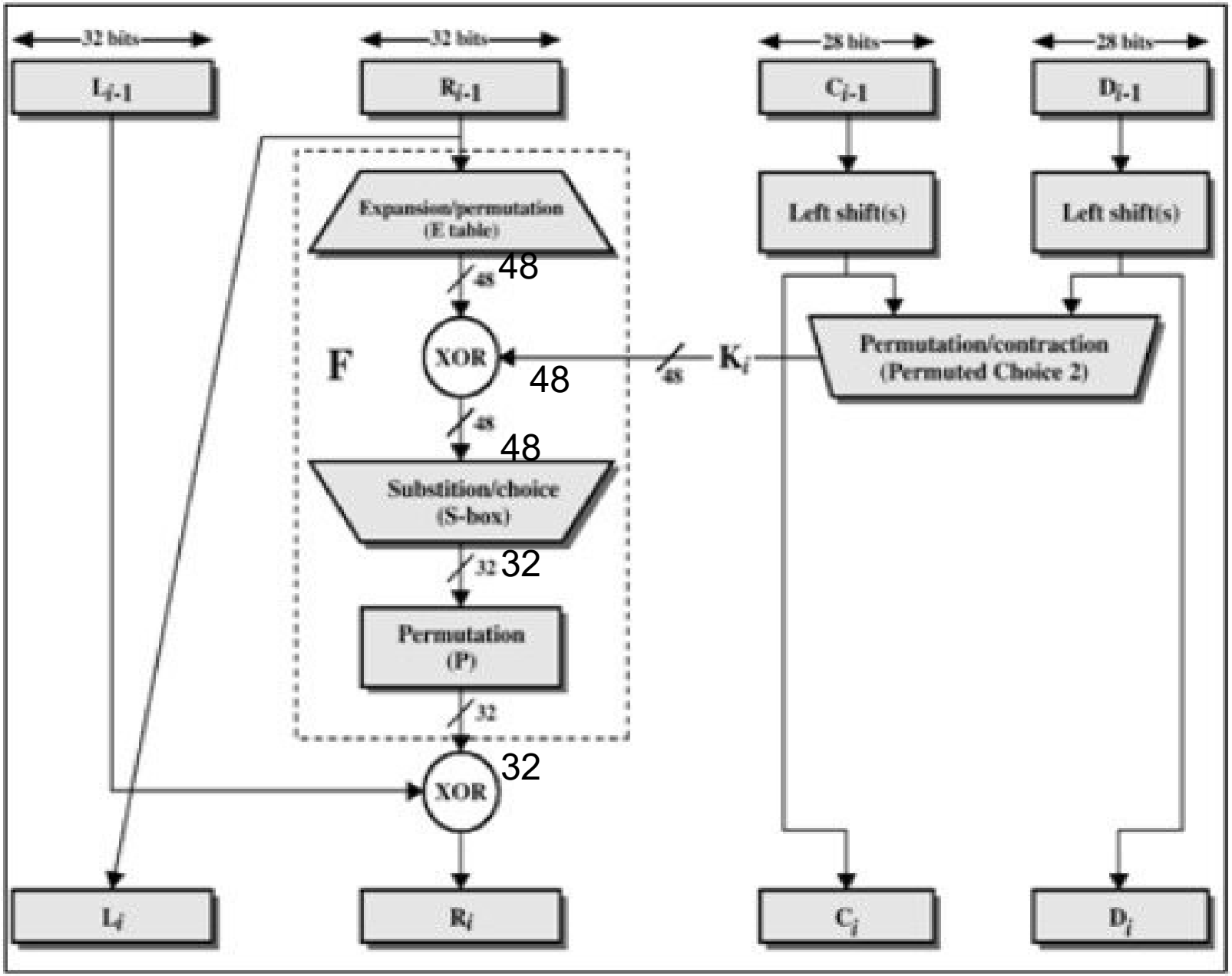


Data Encryption Standard (DES)

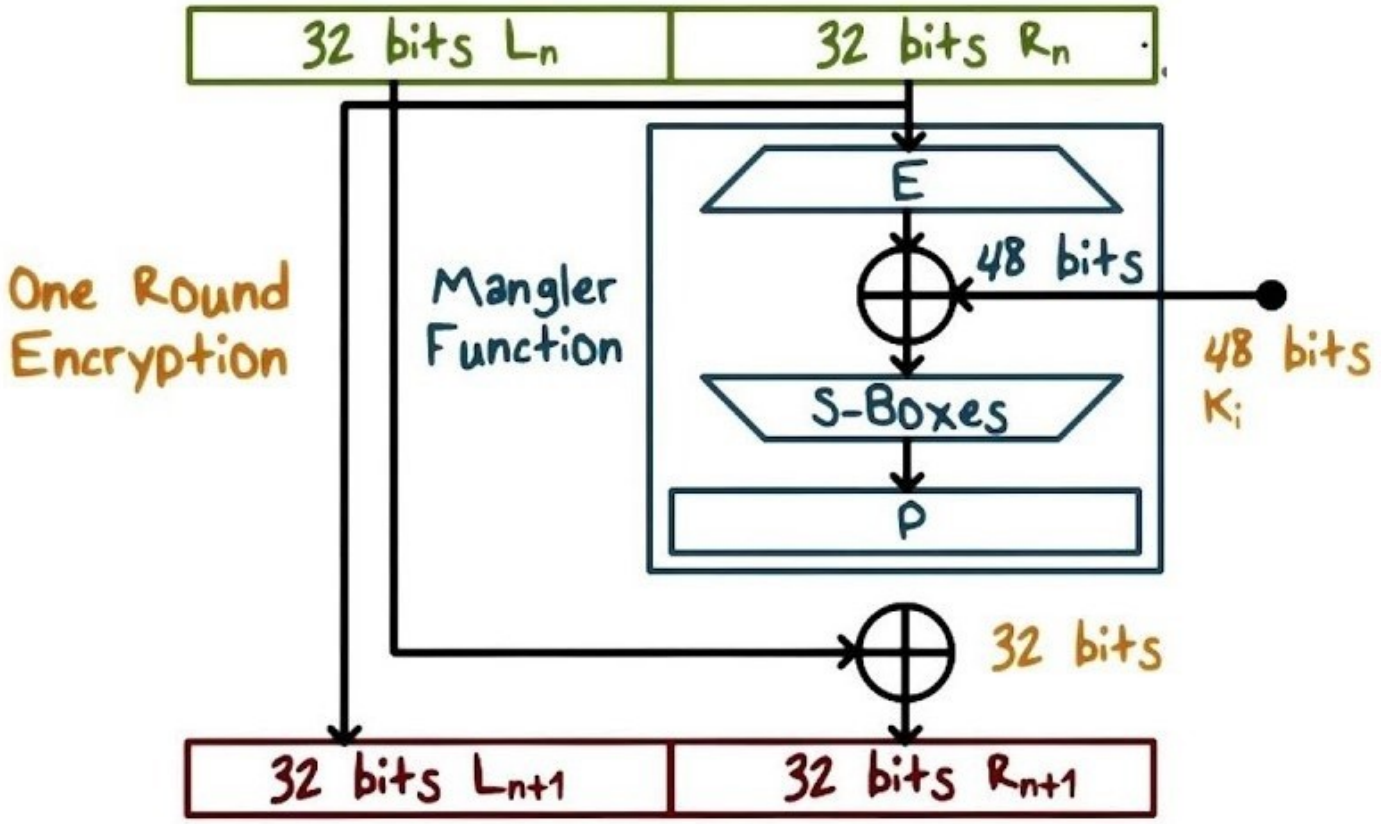
- Most widely used encryption scheme
- Issued in 1977 as Federal Information Processing Standard 46 (FIPS 46) by the National Institute of Standards and Technology (NIST)
- The algorithm itself is referred to as the Data Encryption Algorithm (DEA)

SYMMETRIC BLOCK ENCRYPTION ALGORITHMS DES

- **The Algorithm (DES)**
- The plaintext is 64 bits in length and the key is 56 bits in length;
- longer plaintext amounts are processed in 64-bit blocks.
- The DES structure is a minor variation of the Feistel network
- There are 16 rounds of processing. From the original 56-bit key, 16 subkeys are generated, one of which is used for each round.
- The process of decryption with DES is essentially the same as the encryption process.
- The rule is as follows: Use the ciphertext as input to the DES algorithm, but use the subkeys Ki in reverse order.
- That is, use K16 on the first iteration, K15 on the second iteration, and so on until K1 is used on the 16th and last iteration.


32 bits
32 bits
28 bits
28 bits
L_{i-1}
R_{i-1}
C_{i-1}
D_{i-1}
Expansion/permutation
(E table)
Left shift(s)
Left shift(s)
48
F
XOR
48
K_i
48
Permutation/contraction
(Permuted Choice 2)
48
Substition/choice
(S-box)
32
32
Permutation
(P)
32
XOR
32
L_i
R_i
C_i
D_i

A DES Round


32 bits L_n
32 bits R_n
E
48 bits
48 bits
K_i
S-Boxes
P
32 bits
32 bits L_{n+1}
32 bits R_{n+1}
One Round Encryption
Mangler Function

- S-box (substitution-box) is a basic component of symmetric key algorithms which performs substitution
- S-box from DES (S5), mapping 6-bit input into a 4-bit output:
- For input "011011" outer bit 01 is row and inner bit 1101 is colum

S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

SYMMETRIC BLOCK ENCRYPTION ALGORITHMS

- **THE STRENGTH (DES)**
- The first concern refers to the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm.
- Over the years, there have been numerous attempts to find weaknesses in the algorithm, making DES the most-studied encryption algorithm in existence.
- Despite numerous approaches, no one has so far succeeded in discovering a fatal weakness in DES.

SYMMETRIC BLOCK ENCRYPTION ALGORITHMS

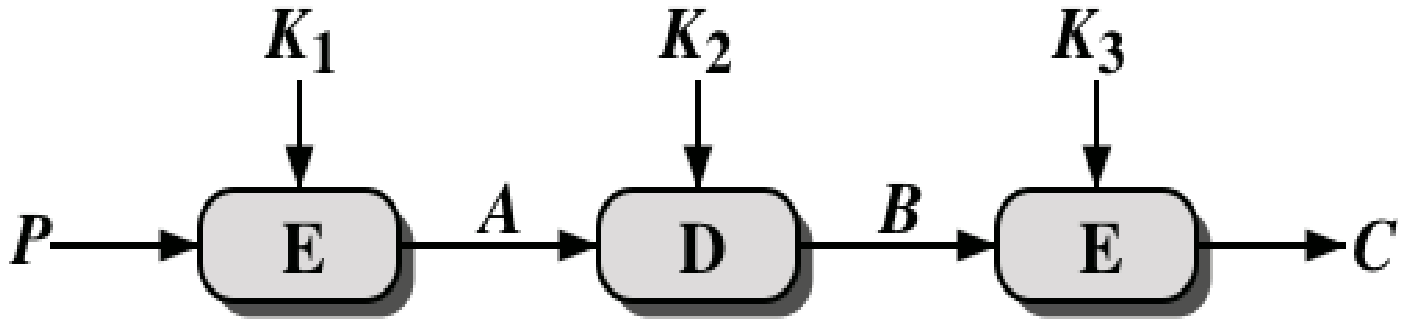
- **THE STRENGTH (DES)**
- A second and more serious concern is key length. With a key length of 56 bits, there are 2^n possible keys, which is approximately $7.2 * 10^{16}$ keys.
- Thus, on the face of it, a brute force attack appears impractical.
- Assuming that on average half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.

DES algorithm

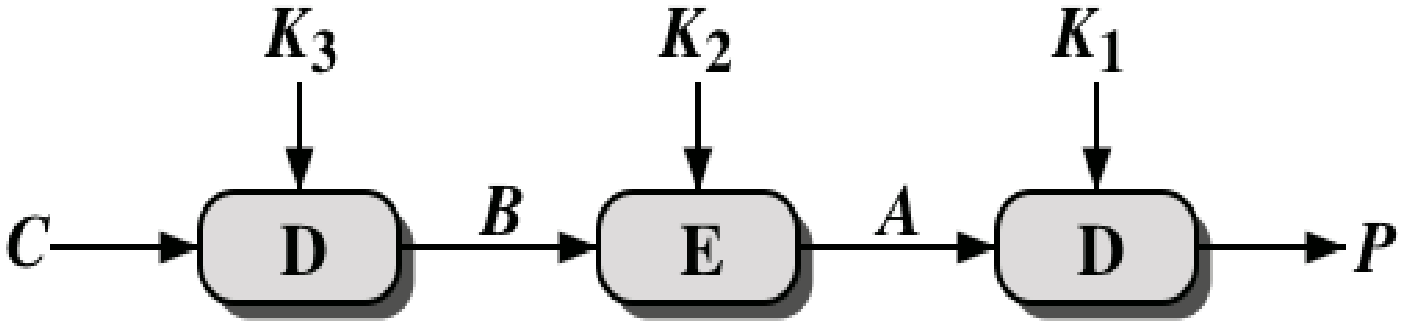
- Description of the algorithm:
  - Plaintext is 64 bits in length
  - Key is 56 bits in length (converted into 48 bit)
  - There are 16 rounds of processing
  - Process of decryption is essentially the same as the encryption process
- The strength of DES:
  - Concerns fall into two categories
    - The algorithm itself
      - Refers to the possibility that cryptanalysis is possible by exploiting the characteristics of the algorithm
    - The use of a 56-bit key
      - $2^{56} = 7.2 \times 10^{16}$ possible key
      - Speed of commercial, off-the-shelf processors threatens the security

Triple DES

- The speed of exhaustive key searches against DES after 1990 began to cause discomfort amongst users of DES.
- However, users did not want to replace DES as it takes an enormous amount of time and money to change encryption algorithms that are widely adopted and embedded in large security architectures.
- Before using 3TDES, user first generate and distribute a 3TDES key K, which consists of three different DES keys K1, K2 and K3.
- This means that the actual 3TDES key has length $3 \times 56 = 168$ bits.



(a) Encryption

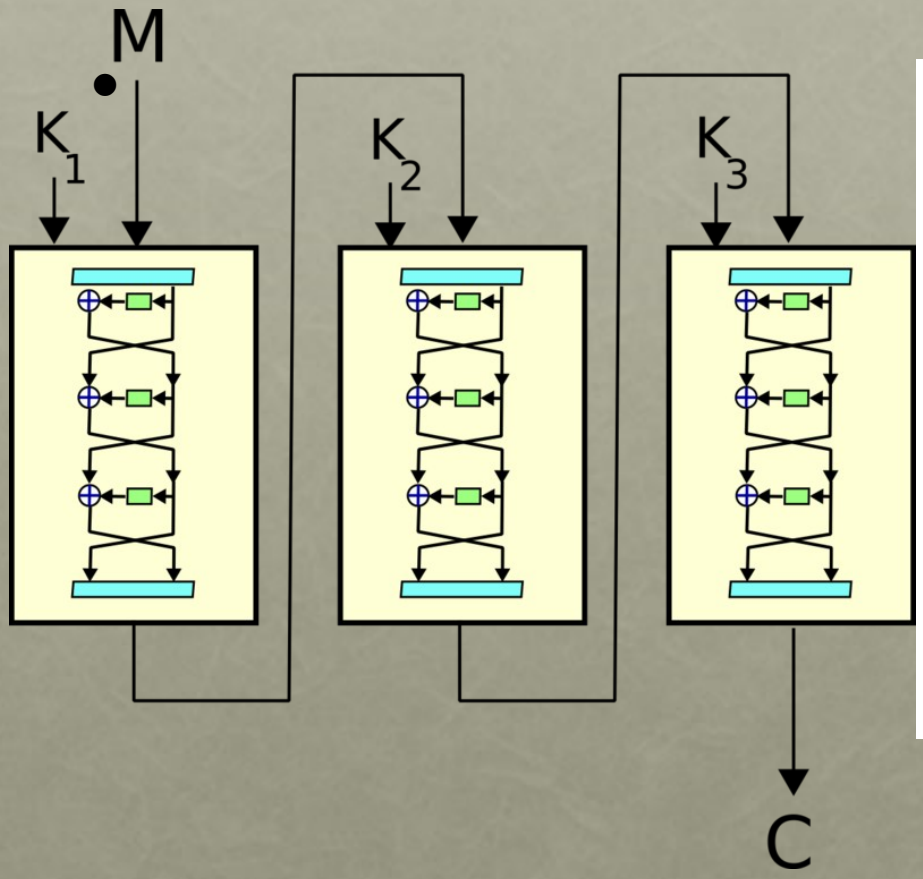


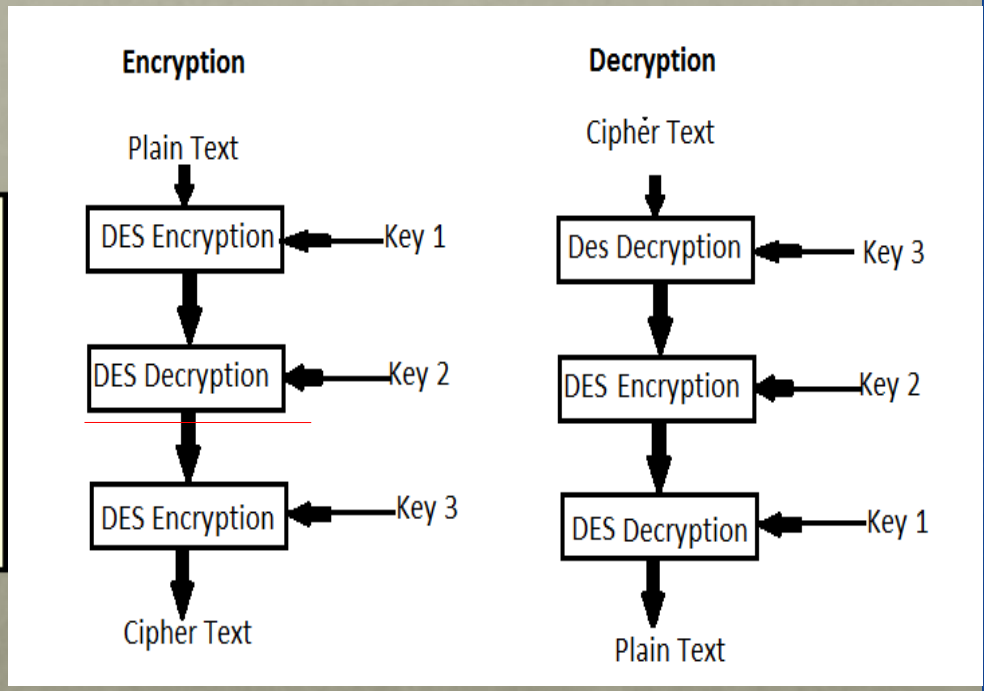
(b) Decryption

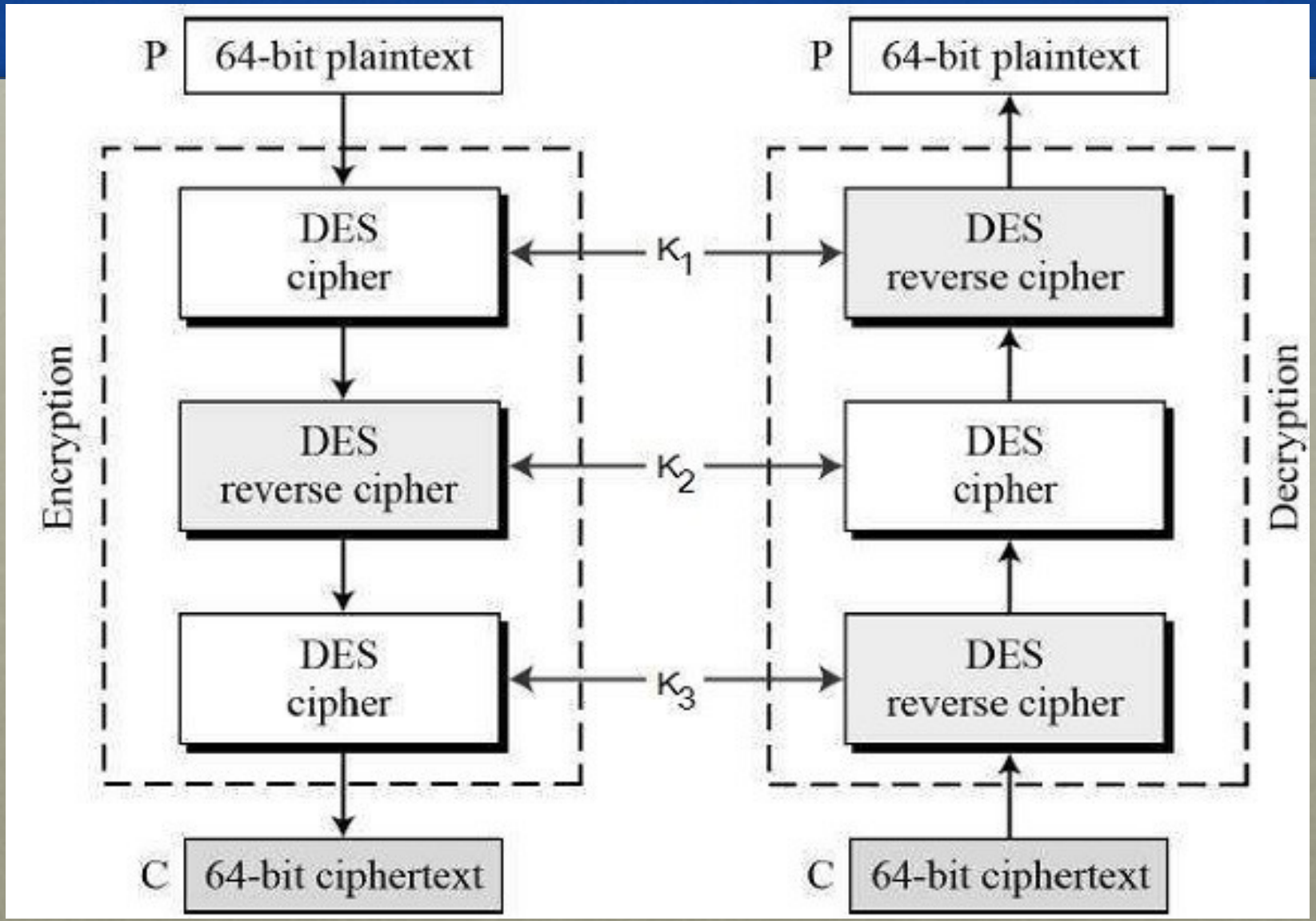
Figure 2.3 Triple DES

Triple DES

- M is plain text , K1,k2,k3 is keys and C is cypher text






P
64-bit plaintext
DES
cipher
DES
reverse cipher
DES
cipher
C
64-bit ciphertext
Encryption
K1
K2
K3
P
64-bit plaintext
DES
reverse cipher
DES
cipher
DES
reverse cipher
C
64-bit ciphertext
Decryption

Triple DES

- The encryption-decryption process is as follows −
- Encrypt the plaintext blocks using single DES with key K_1.
- Now decrypt the output of step 1 using single DES with key K_2.
- Finally, encrypt the output of step 2 using single DES with key K_3.
- The output of step 3 is the ciphertext.
- Decryption of a ciphertext is a reverse process. User first decrypt using K_3, then encrypt with K_2, and finally decrypt with K_1.

Triple DES

- Due to this design of Triple DES as an encrypt–decrypt–encrypt process, it is possible to use a 3TDES (hardware) implementation for single DES by setting $K_{1,}$ $K_{2,}$ and K_3 to be the same value.
- This provides backwards compatibility with DES.
- Second variant of Triple DES (2TDES) is identical to 3TDES except that K_3 is replaced by K_1.
- In other words, user encrypt plaintext blocks with key $K_{1,}$ then decrypt with key $K_{2,}$ and finally encrypt with K_1 again.
- Therefore, 2TDES has a key length of 112 bits.

3DES guidelines

Federal Information Processing Standards (FIPS) Publications are standards issued by NIST

- FIPS 46-3 includes the following guidelines for 3DES:
  - 3DES is the FIPS-approved symmetric encryption algorithm
  - The original DES, which uses a single 56-bit key, is permitted under the standard for legacy systems only; new procurements should support 3DES
  - Government organizations with legacy DES systems are encouraged to transition to 3DES
  - It is anticipated that 3DES and the Advanced Encryption Standard (AES) will coexist as FIPS-approved algorithms, allowing for a gradual transition to AES

Advanced encryption standard (AES)

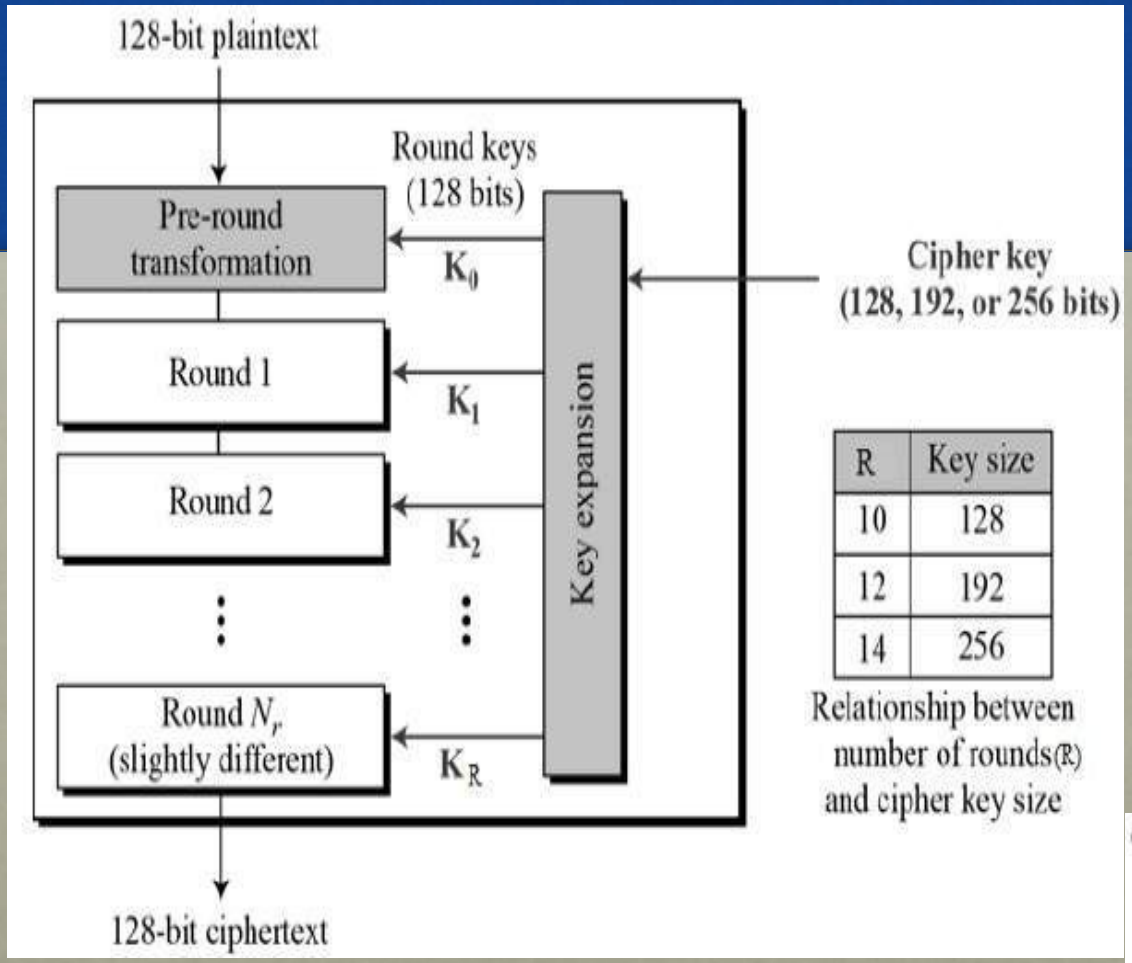
- Principal drawback of 3DEs is , it is relativally slower and it use 64 bit block size.
- In 1997 NIST issued a call for proposals for a new AES:
  - Should have a security strength better than 3DES and significantly improved efficiency
  - Must be a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits
  - Evaluation criteria included security, computational efficiency, memory requirements, hardware and software suitability, and flexibility.

Advanced encryption standard (AES)

- In a first round of evaluation, 15 proposed algorithms were accepted. A second round narrowed the field to five algorithms.
- NIST completed its evaluation process and published a final standard (FIPS PUB 197) in November of 2001Developers were two cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen.
- NIST selected Rijndael as the proposed AES algorithm

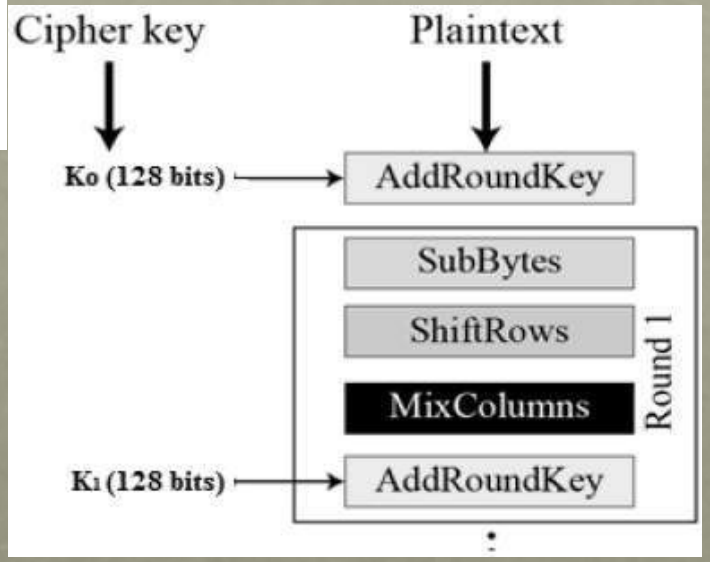
Features of AES

- The features of AES are as follows −
  - Symmetric key symmetric block cipher
  - 128-bit data, 128/192/256-bit keys
  - 10 rounds for 128-bit keys.
  - 12 rounds for 192-bit keys.
  - 14 rounds for 256-bit keys.
  - Stronger and faster than Triple-DES
  - Provide full specification and design details
  - Software implementable in C and Java



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size



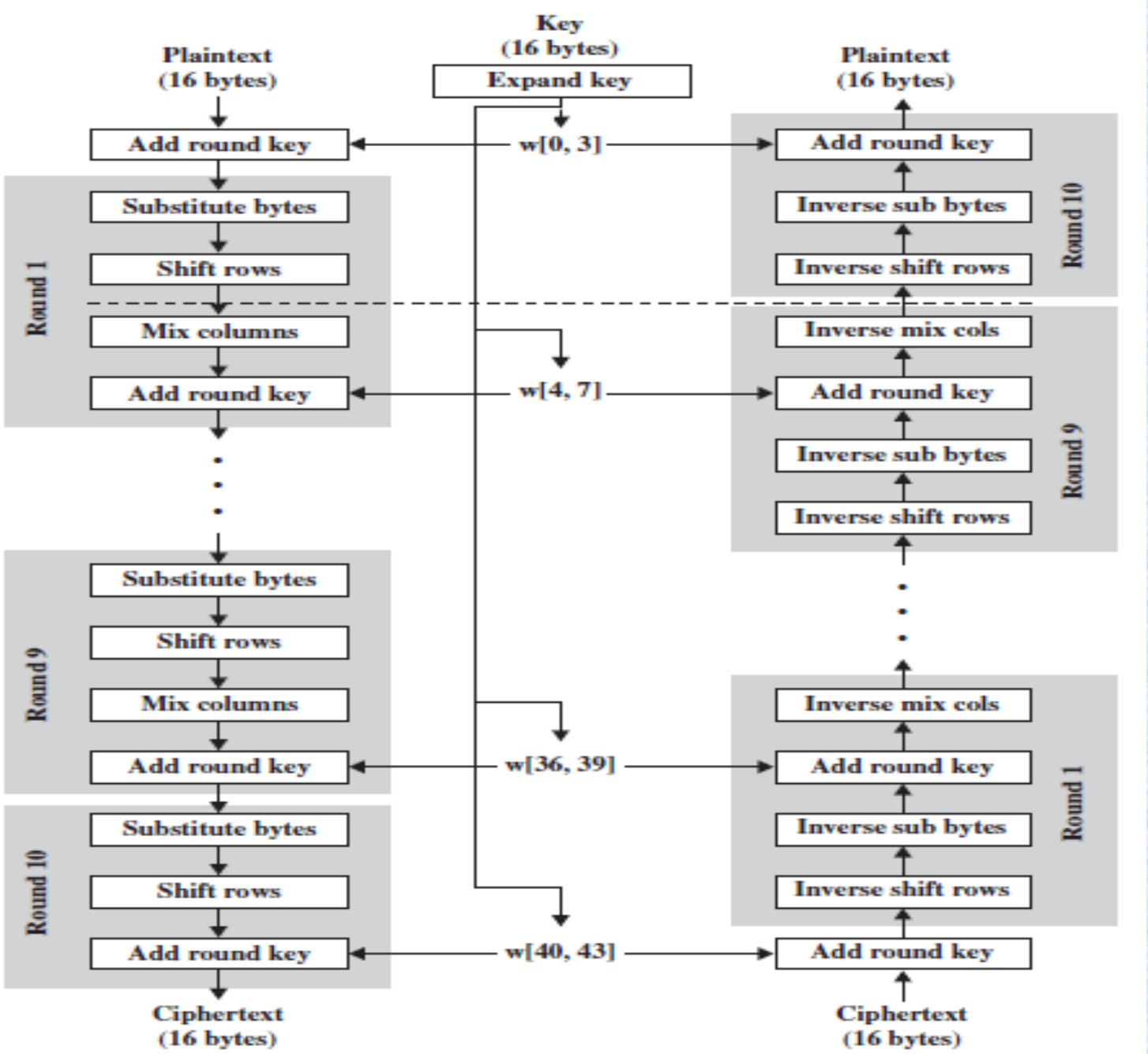


Figure 2.5 AES Encryption and Decryption

Encryption and Decryption

- Figure 2.5 shows the overall structure of AES. The input to algorithms is a single 128-bit block.
- In FIPS PUB 197, this block is depicted as a square matrix of bytes and copied into the State array, which is modified at each stage of encryption or decryption.
- After the final stage, State is copied to an output matrix. Similarly, the 128-bit key is depicted as a square matrix of bytes.

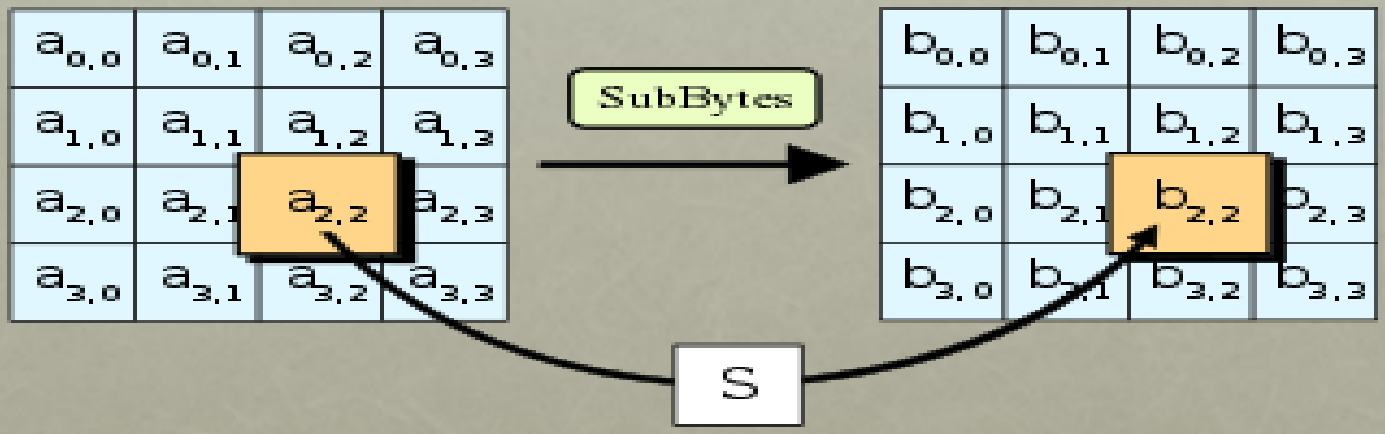
Encryption and Decryption

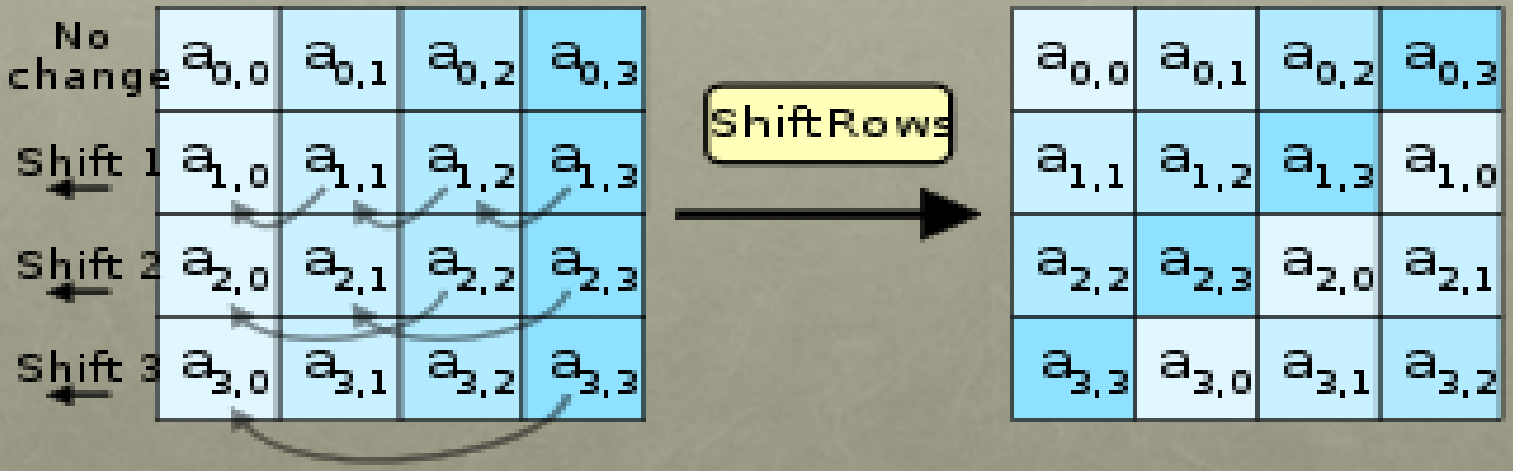
- This key is then expanded into an array of key schedule words: each word is four bytes and the total key schedule is 44 words for the 128-bit key.
- The ordering of bytes within a matrix is by column.
- So, for example, the first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the in matrix, the second four bytes occupy the second column, and so on.
- Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the w matrix

Steps

- Substitute bytes:
  - Uses a table, referred to as an S-box, to perform a byte-by-byte substitution of the block.
  - The result is in a matrix of four rows and four columns.
- Shift rows:
  - A simple permutation that is performed row by row.
  - First row is not shifted.
  - Second row is shifted one (byte) position to the left.
  - Third row is shifted two positions to the left.
  - Fourth row is shifted three positions to the left.
  - The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

Steps

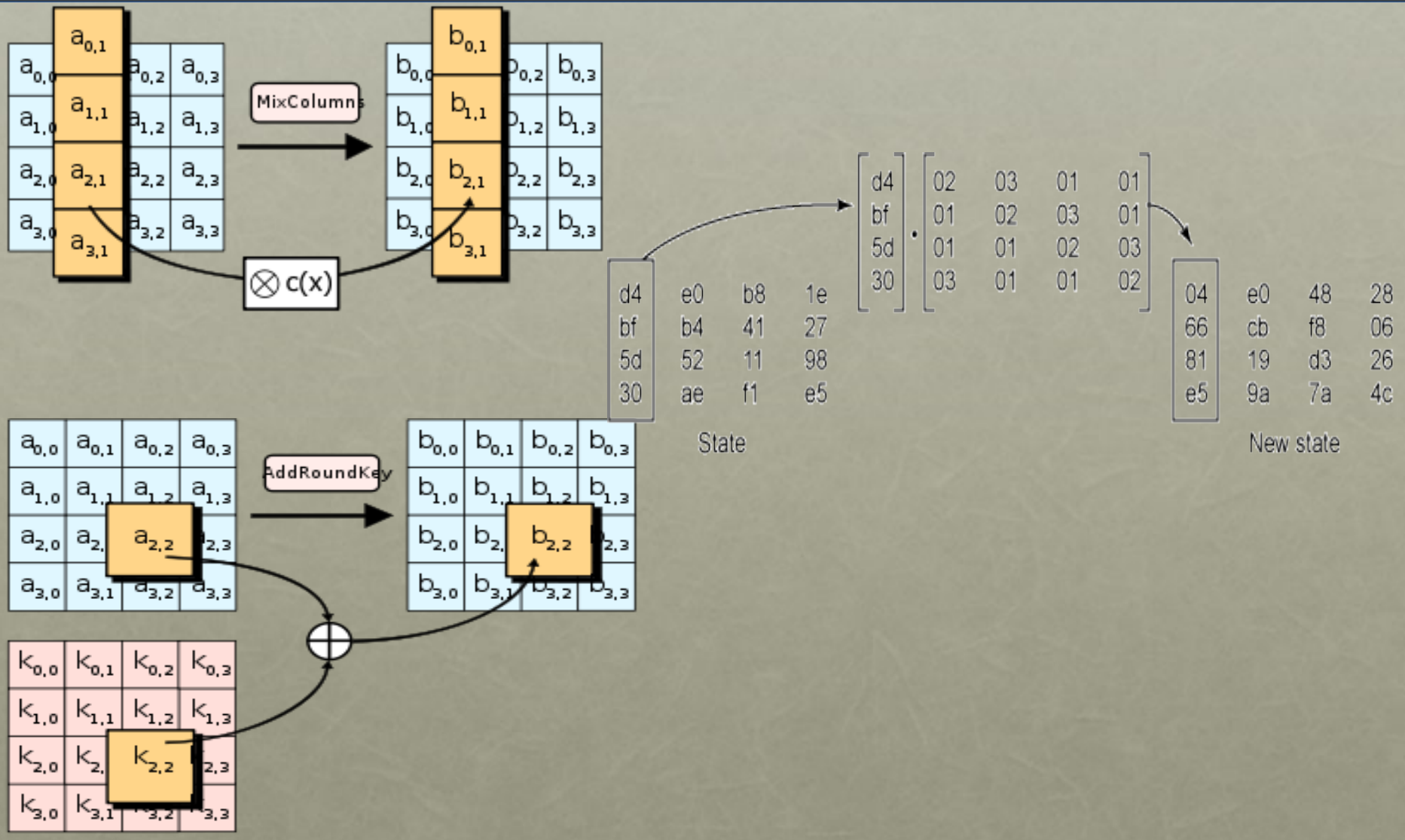

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S


No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

Steps

- Mix columns:
  - A substitution that alters each byte in a column as a function of all of the bytes in the column.
  - This step is not performed in the last round.
- Add round key:
  - The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key.
  - If this is the last round then the output is the ciphertext.
  - Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Steps



AES

- A replacement for DES was needed as its key size was too small
- The more popular and widely adopted symmetric encryption algorithm. In present day cryptography, AES is widely adopted and supported in both hardware and software.
- It is found at least six time faster than triple DES.
- Till date, no practical cryptanalytic attacks against AES has been discovered.
- Additionally, AES has built-in flexibility of key length, which allows a degree of ‘future-proofing’ against progress in the ability to perform exhaustive key searches.
- However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

Random and pseudorandom Numbers

- A number of network security algorithms based on cryptography make use of random numbers
- Examples:
  - Generation of keys for the RSA public-key encryption algorithm and other public-key algorithms
  - Generation of a symmetric key for use as a temporary session key; used in a number of networking applications such as Transport Layer Security, Wi-Fi, e-mail security, and IP security
  - In a number of key distribution scenarios, such as Kerberos, random numbers are used for handshaking to prevent replay attacks
- Two distinct and not necessarily compatible requirements for a sequence of random numbers are:
- Randomness
- Unpredictability

Randomness

- The following criteria are used to validate that a sequence of numbers is random:

Uniform Distribution

- The distribution of bits in the sequence should be uniform
- Frequency of occurrence of ones and zeros should be approximately the same

Independence

- No subsequence in the sequence can be inferred from the others

That is no value in the sequence should be dependent on the other value

- There is no test to "prove" independence
- The general strategy is to apply a number of tests until the confidence that independence exists is sufficiently strong

Unpredictability

- In applications such as reciprocal authentication and session key generation, the requirement is not so much that the sequence of numbers be statistically random but that the successive members of the sequence are unpredictable
- With “true” random sequences, each number is statistically independent of other numbers in the sequence and therefore unpredictable
- Care must be taken that an opponent not be able to predict future elements of the sequence on the basis of earlier elements

Random and pseudorandom Numbers

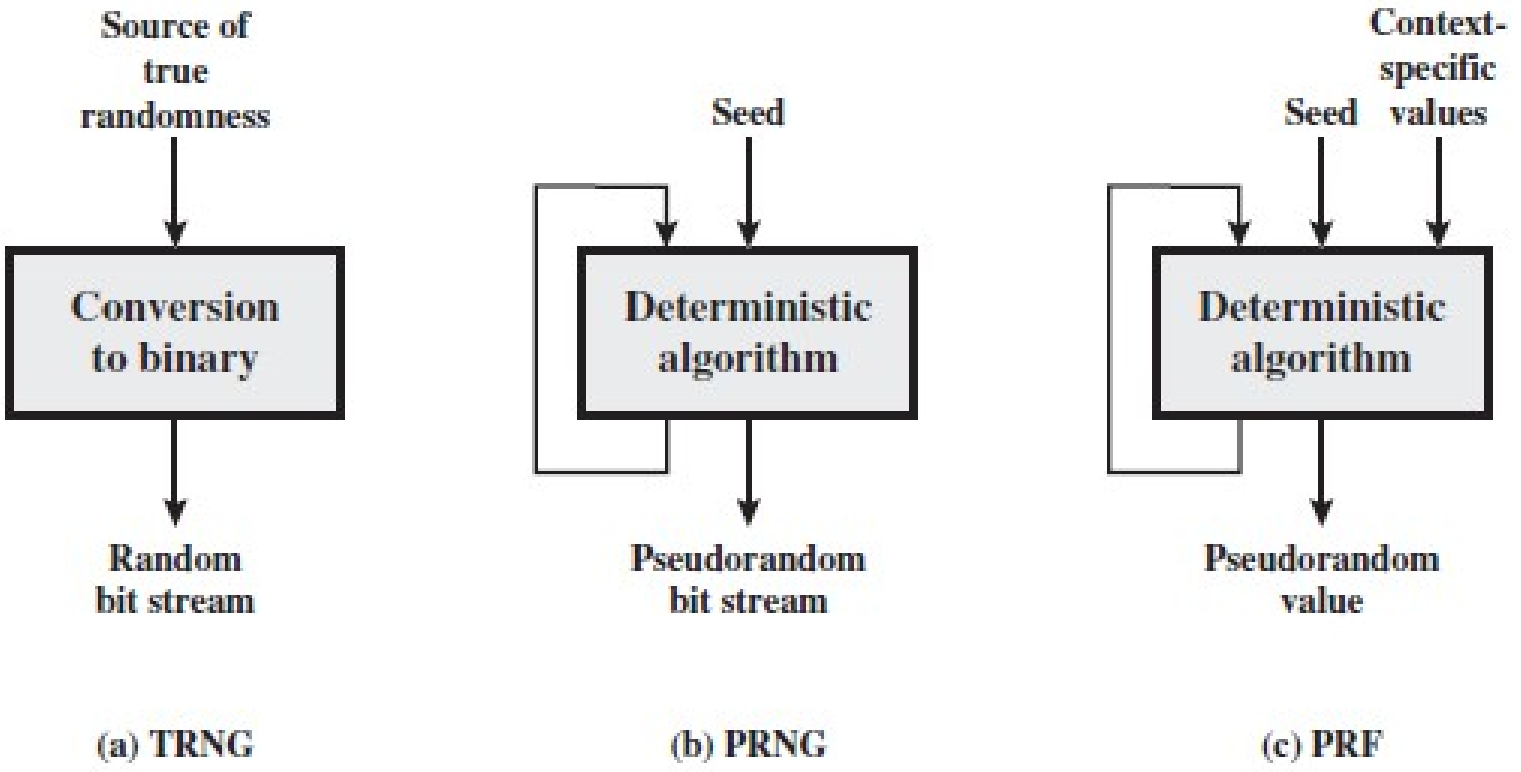


Figure 2.7 Random and Pseudorandom Number Generators

- TRNG : conversion in binary form
- PRNG takes input a fixed value and produces output bit using deterministic algorithm. There is some feedback path where Some of the result of algo. Feedback taken as an input as additional output bit are produced.
- PRF : produce a output string of bit of some fixed length . It takes the input a seed plus some context specification values such as user id /application ID .

Algorithm design

- If algorithm is good the resulting sequences will pass many reasonable test of randomness such numbers are reffered as pseudorandom numbers.

Purpose-built algorithms	• Designed specifically and solely for the of generating pseudorandom bit streams
Algorithms based on existing cryptographic algorithms	• Crypto graphic algorithms have the effect of randomizing input • Can serve as the core of PRNGs
Three broad categories of cryptographic algorithms are commonly used to create PRNGs:	• Symmetric block ciphers • Asymmetric ciphers • Hash functions and message authentication codes

Stream Ciphers and RC4

- A block cipher processes the input one block of elements at a time, producing an output block for each input block.
- A stream cipher processes the input elements continuously, producing output one element at a time as it goes along.
- Although block ciphers are far more common, there are certain applications in which a stream cipher is more appropriate.
- **Stream Cipher Structure**
- A typical stream cipher encrypts plaintext one byte at a time, although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time.
-

- For example, if the next byte generated by the generator is 01101100
- and the next plaintext byte is 11001100, then the resulting ciphertext byte is:

```
  11001100  plaintext
⊕ 01101100  key stream
  10100000  ciphertext
```

Decryption requires the use of the same pseudorandom sequence:

```
  10100000  ciphertext
⊕ 01101100  key stream
  11001100  plaintext
```

Stream Cipher design considerations

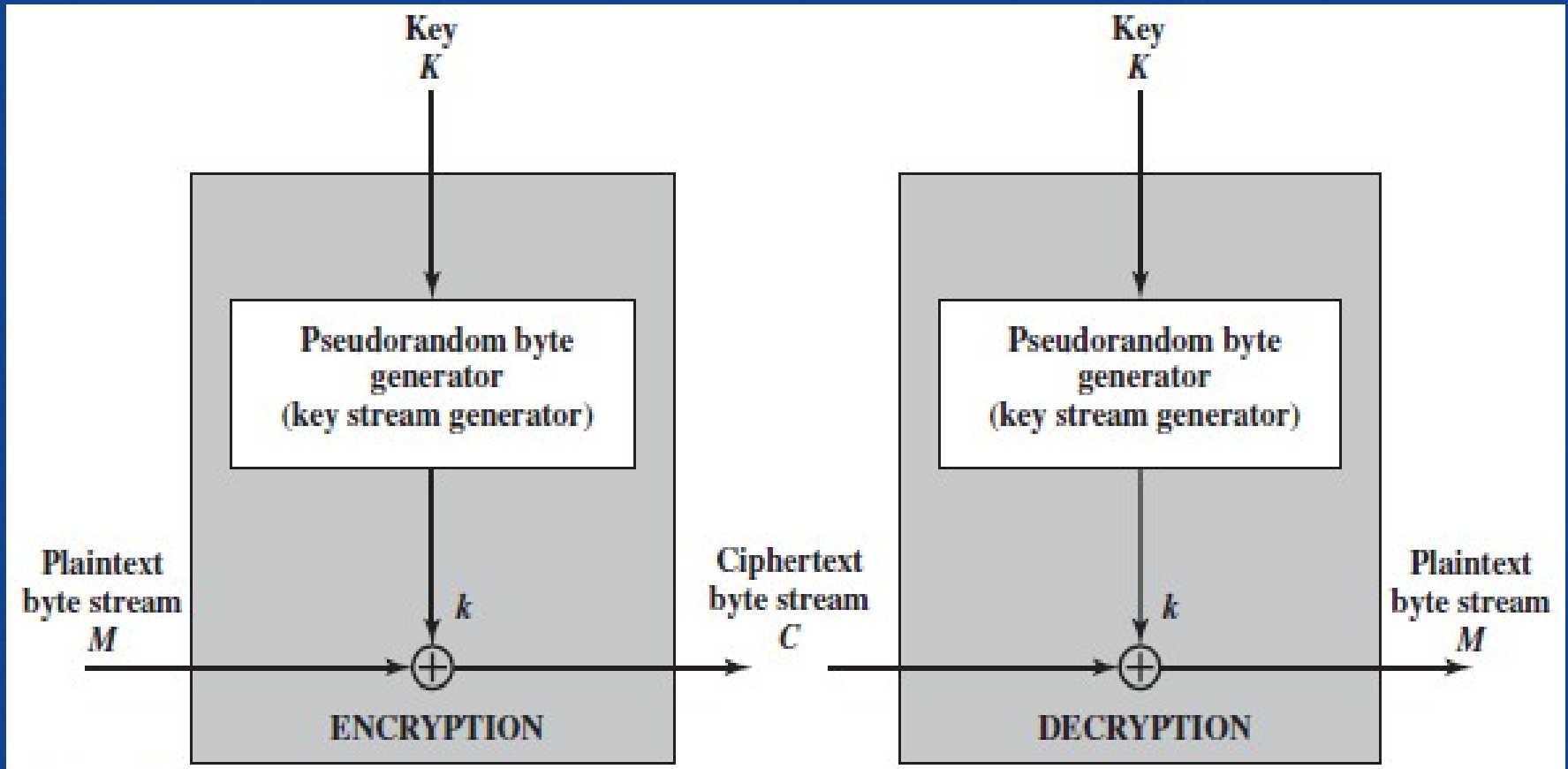


Figure 2.8 Stream Cipher Diagram

Stream Cipher design considerations

Stream Cipher design considerations

1) Pseudorandom number generator uses a function that produce a stream of bits which is eventually repeated. The encryption sequence should have a large period The longer the period of repeat, the more difficult it will be to do cryptanalysis
2) The keystream should approximate the properties of a true random number stream as close as possible E.g there should be equal number of 1s and 0s . The more random-appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult
3) The pseudorandom number generator is conditioned on the value of the input key To guard against brute-force attacks, the key needs to be sufficiently long With current technology, a key length of at least 128 bits is desirable

- With properly designed pseudorandom number generator , stream cipher is secure as a block cipher.
- Advantage of stream cipher is that it is more faster than the block cipher.
- RC4 is the algorithm of stream cipher.

RC4 algorithm

- A stream cipher designed in 1987 by Ron Rivest for RSA Security
- It is a variable key-size stream cipher with byte-oriented operations
- The algorithm is based on the use of a random permutation of bytes.
- Is used in the Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards that have been defined for communication between Web browsers and servers
- Also used in the Wired Equivalent Privacy (WEP) protocol and the newer WiFi Protected Access (WPA) protocol that are part of the IEEE 802.11 wireless LAN standard

Consists of 2 parts:

- Key Scheduling Algorithm (KSA)
- Pseudo-Random Generation Algorithm (PRGA)

After that it will perform encryption -decryption.

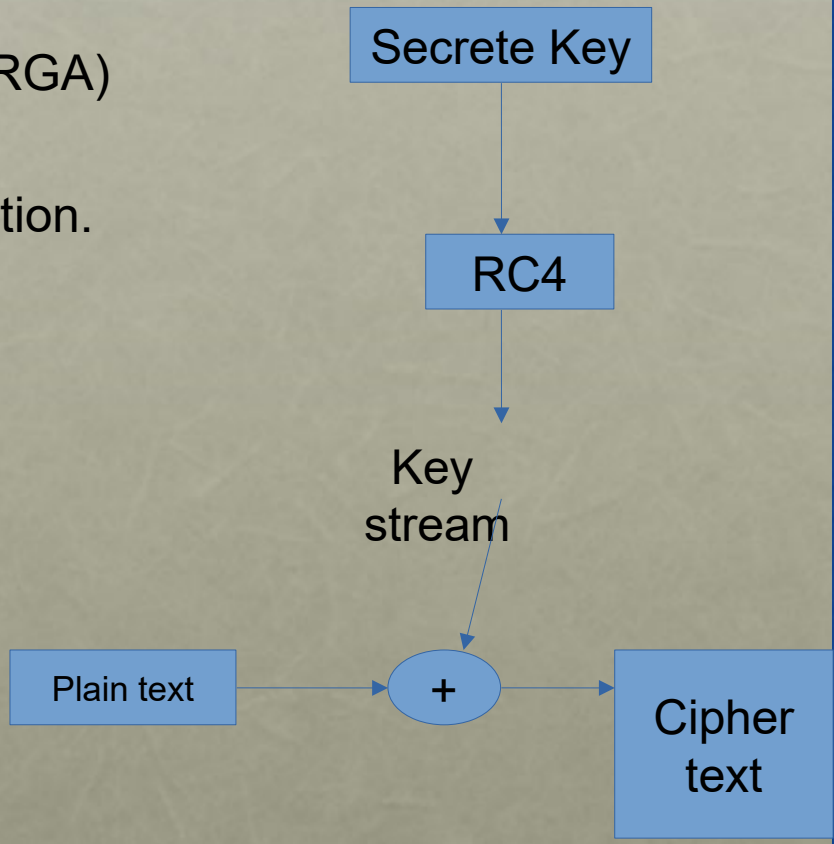
KSA

Generate State array

PRGA on the KSA

Generate keystream

XOR keystream with the data to generated encrypted stream



RC4 algorithm

- **Key-Generation Algorithm**
- A variable-length key from 1 to 256 bytes(8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S[0] to S[255].
- For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion, then the entries in S are permuted again.

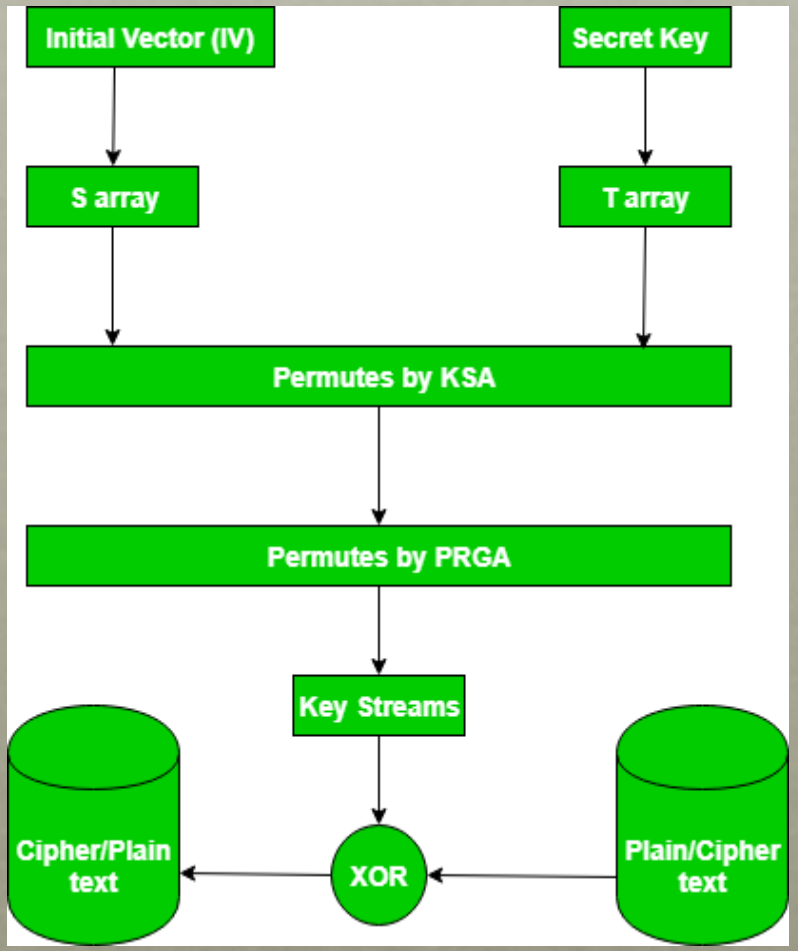
RC4 algorithm

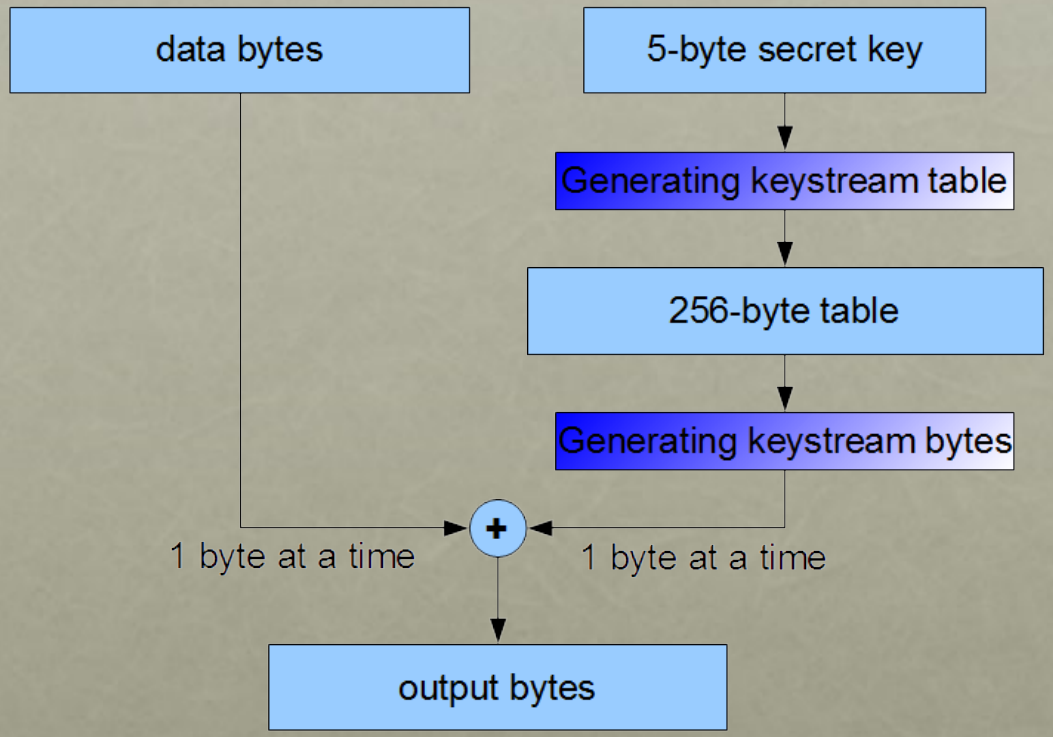
- **Key-Scheduling Algorithm:**
- 1. Initialization of S: The entries of S are set equal to the values from 0 to 255 in ascending order s[0] = 0 s[1] =1 s[255] = 255,
- A temporary vector T, is created.
- If the length of the key k is 256 bytes, then k is assigned to T. Otherwise, for a key with length(k-len) bytes, the first k-len elements of T as copied from K, and then K is repeated as many times as necessary to fill T.
- Use T to produce the initial permutation of S. Starting with S[0] to S[255], and for each S[i] algorithm swap it with another byte in S according to a scheme dictated by T[i], but S will still contain values from 0 to 255

RC4 algorithm

- **2. Pseudo random generation algorithm (Stream Generation):**
- Once the vector S is initialized, the input key will not be used. In this step, for each S[i] algorithm swap it with another byte in S according to a scheme dictated by the current configuration of S.
- After reaching S[255] the process continues, starting from S[0] again.
- **3. Encrypt using X-OR()**

BLOCK DIAGRAM





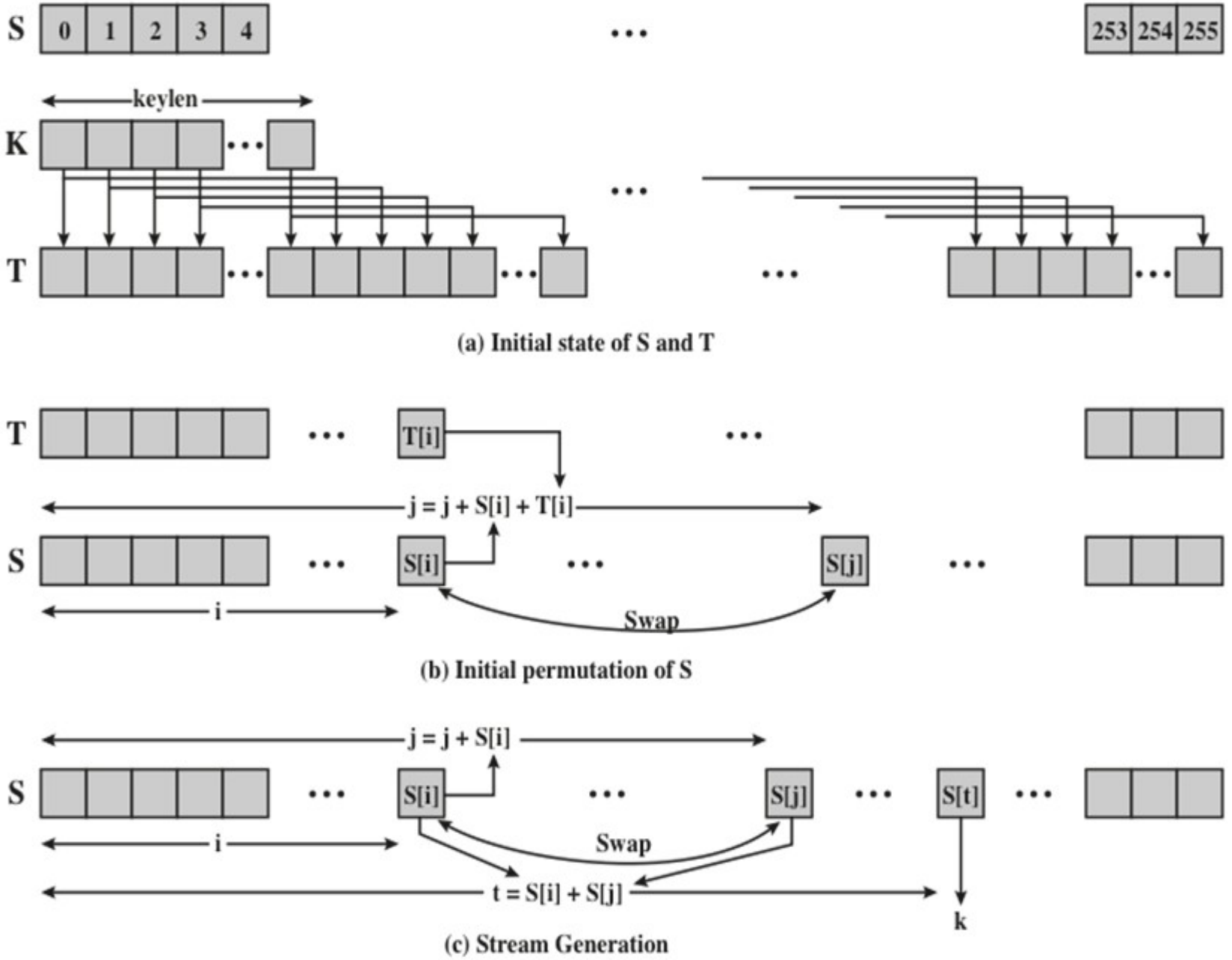


Figure 2.8 RC4

Cipher block Modes of Operation

- A symmetric block cipher processes one block of data at a time
  - In the case of DES and 3DES, the block length is b=64 bits
  - For AES, the block length is b=128
  - For longer amounts of plaintext, it is necessary to break the plaintext into b-bit blocks, padding the last block if necessary
- Four modes of operation have been defined by NIST to apply the block cipher
  - Intended to cover all the possible applications of encryption for which a block cipher could be used
  - Intended for use with any symmetric block cipher, including triple DES and AES

- Mode of Operations
  - ECB
  - Cipher block chaining (CBC)
  - Cipher Feedback Mode
  - Counter Mode

Electronic Codebook Mode (ECB)

- Plaintext is handled *b* bits at a time and each block of plaintext is encrypted using the same key
- The term "codebook" is used because, for a given key, there is a unique ciphertext for every *b*-bit block of plaintext
  - One can imagine a gigantic codebook in which there is an entry for every possible *b*-bit plaintext pattern showing its corresponding ciphertext
- With ECB, if the same *b*-bit block of plaintext appears more than once in the message, it always produces the same ciphertext
  - Because of this, for lengthy messages, the ECB mode may not be secure
  - If the message is highly structured, it may be possible for a cryptanalyst to exploit these regularities

Cipher block chaining (CBC)

- Cipher block chaining (CBC) is a mode of operation for a block cipher.
  - Cipher block chaining uses what is known as an initialization vector (IV) of a certain length. By using this along with a single encryption key,
  - One of CBC's key characteristics is that it uses a chaining process that causes the decryption of a block of ciphertext to depend on all the preceding ciphertext blocks
  - in cipher block chaining, each plaintext block is XORed (numerically combined) with the previous ciphertext block and then encrypted
  - The first step to initiating a cipher block chain is to XOR the first plaintext blocks with an IV -- a unique, fixed-length conversion function This XOR output is then encrypted using a cipher key to produce a ciphertext block,

- Next cipher block use the previous block cipher text and xor with plain text.
- After that it will be encrypted using the encryption key.

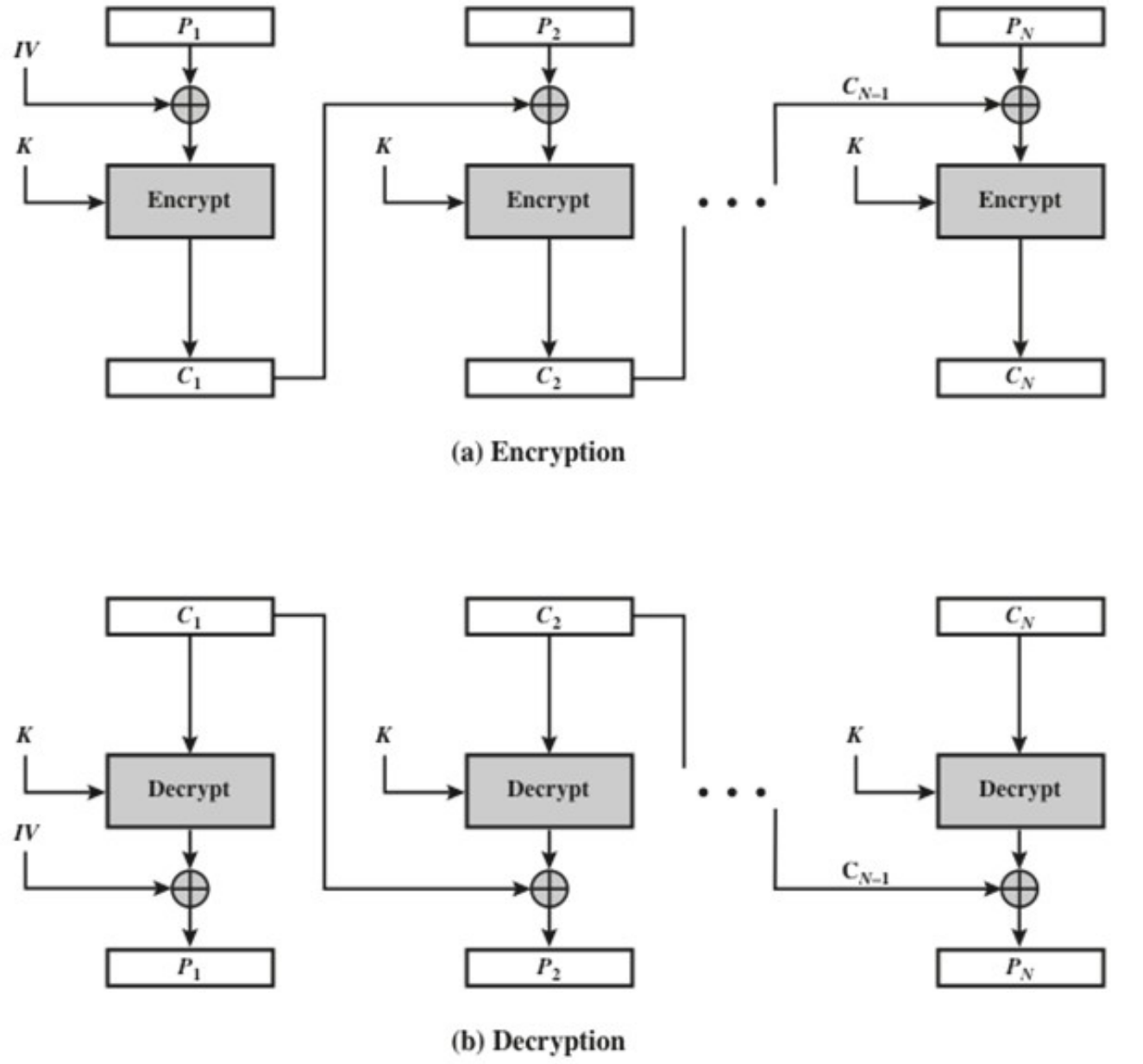


Figure 2.9 Cipher Block Chaining (CBC) Mode

Cipher Feedback Mode

- It is possible to convert any block cipher into a stream cipher by using the cipher feedback (CFB) mode
- One desirable property of a stream cipher is that the ciphertext be of the same length as the plaintext.
- Thus, if 8-bit characters are being transmitted, each character should be encrypted using 8 bits.
- each ciphertext block gets ‘feed back’ into the encryption process in order to encrypt the next plaintext block.
- In the figure, it is assumed that the unit of transmission is s bits; a common value is s 8.

Cipher Feedback Mode

.

- First, consider encryption. The input to the encryption function is a b-bit shift register that is initially set to some initialization vector (IV).
- The leftmost (most significant) s bits of the output of the encryption function are XORed with the first unit of plaintext P1 to produce the first unit of ciphertext $c1_1$, which is then transmitted.
- In addition, the contents of the shift register are shifted left by s bits, and c1 is placed in the rightmost (least significant) c1 bits of the shift register.
- This process continues until all plaintext units have been encrypted.

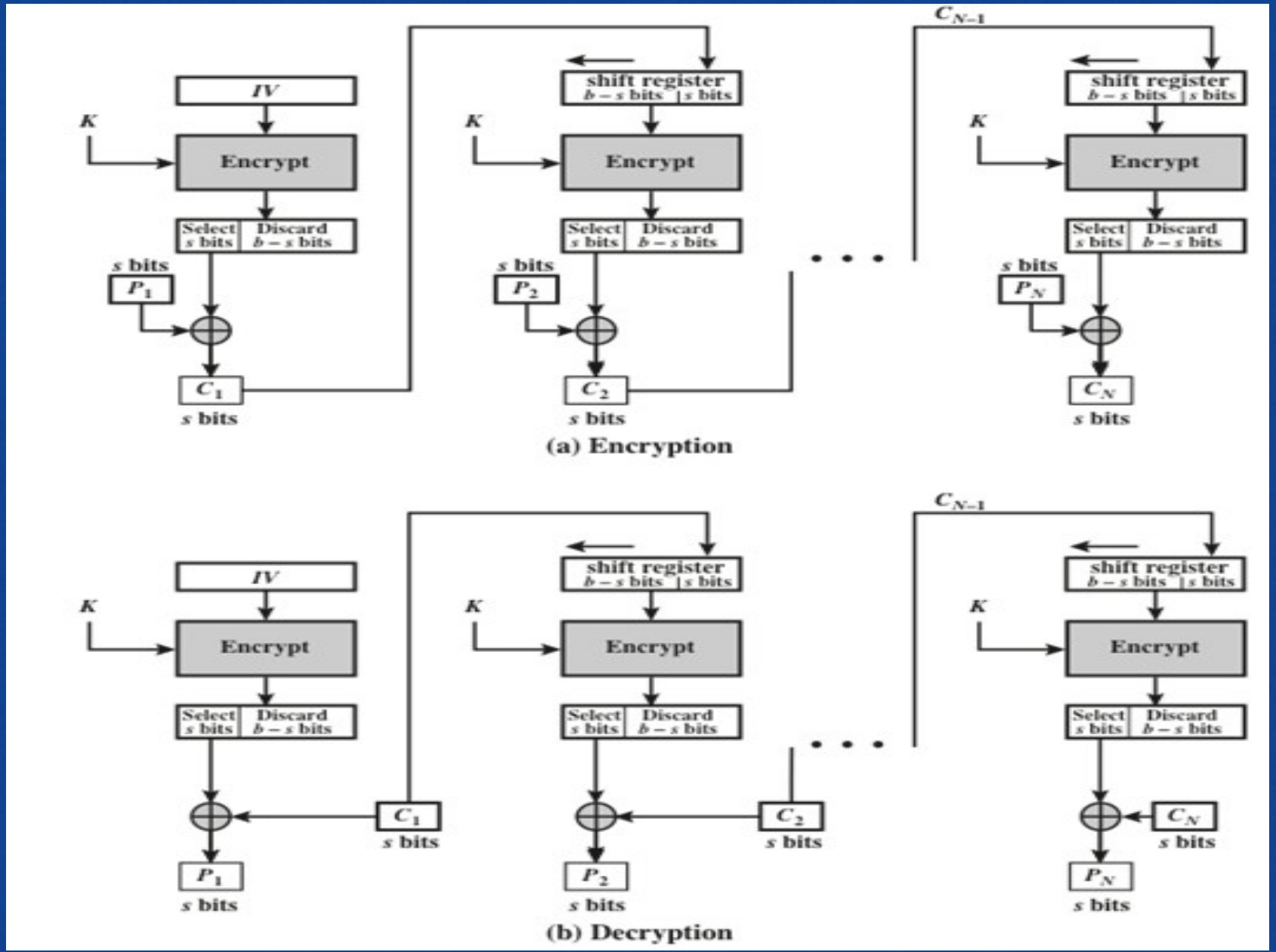


Figure 2.10 *s*-bit Cipher Feedback (CFB) Mode

Advantages of Counter Mode(CTR)

- Hardware efficiency
- Encryption/decryption can be done in parallel on multiple blocks of plaintext or ciphertext
-
- Software efficiency
- Because of the opportunities for parallel execution, processors that support parallel features can be effectively utilized
- Preprocessing
- The execution of the underlying encryption algorithm does not depend on input of the plaintext or ciphertext --- when the plaintext or ciphertext input is presented, the only computation is a series of XORs, greatly enhancing throughput

- Simplicity
- Requires only the implementation of the encryption algorithm and not the decryption algorithm

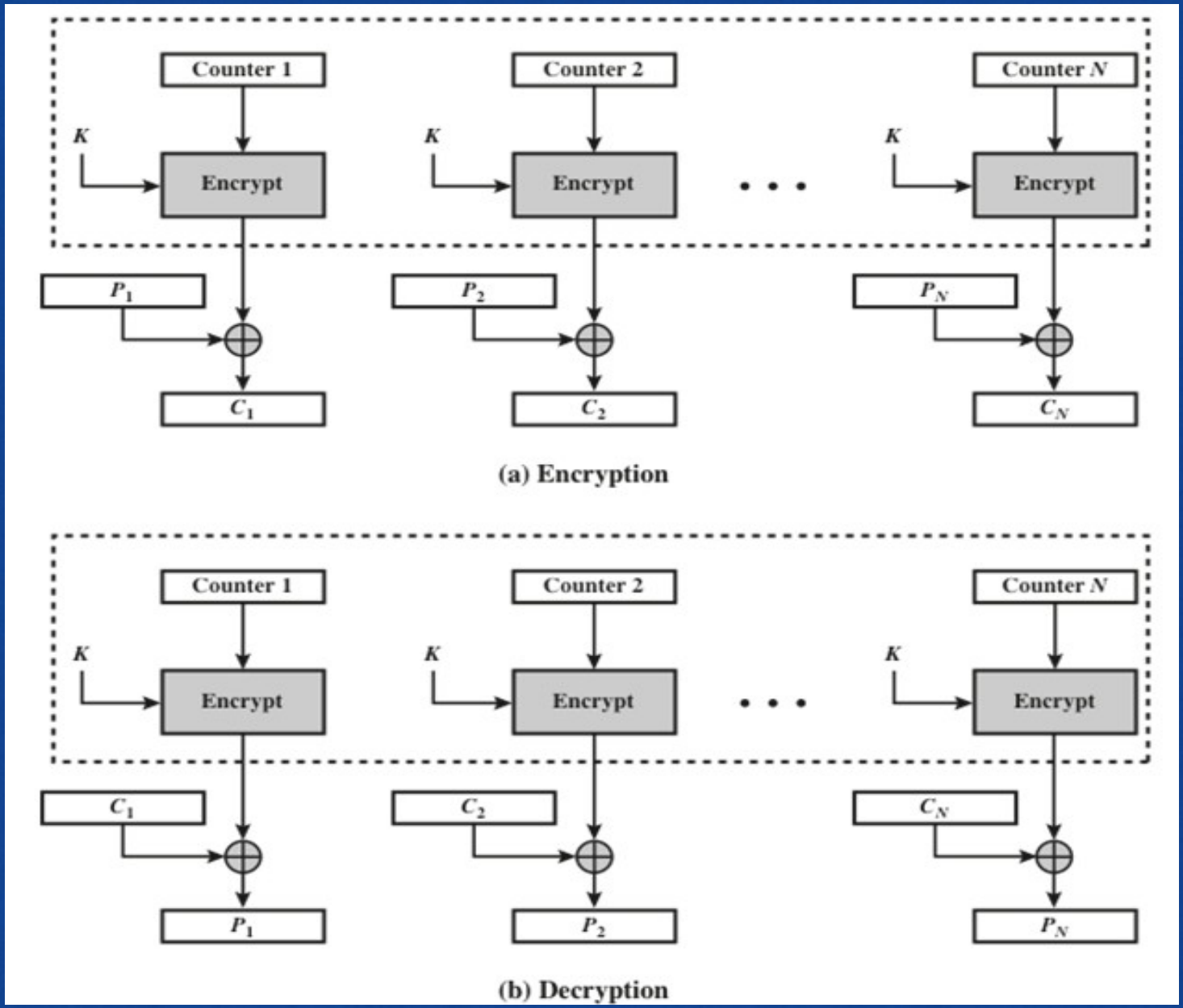


Figure 2.11 Counter (CTR) Mode

Summary

- Symmetric encryption principles
  - Cryptography
  - Cryptanalysis
  - Feistel cipher structure
- Symmetric block encryption algorithms
  - Data encryption standard
  - Triple DES
  - Advanced encryption standard
- Random and pseudorandom numbers
  - The use of random numbers
  - TRNGs, PRNGs, PRFs
  - Algorithm design
- Stream ciphers and RC4
  - Stream cipher structure
  - RC4 algorithm
- Cipher block modes of operation
  - ECB
  - CBC
  - CFB
  - CTR