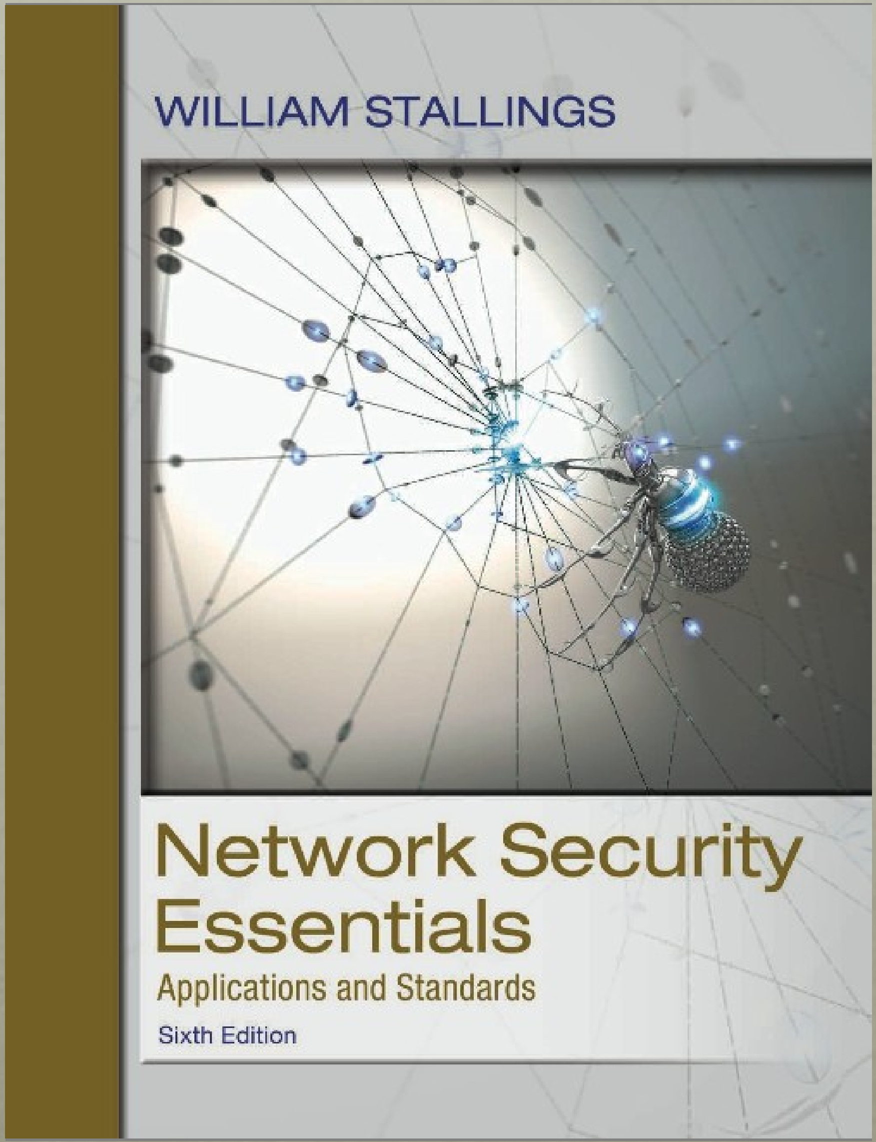
Network Security Essentials

Sixth Edition

by William Stallings



Chapter 2

Symmetric Encryption and Message Confidentiality

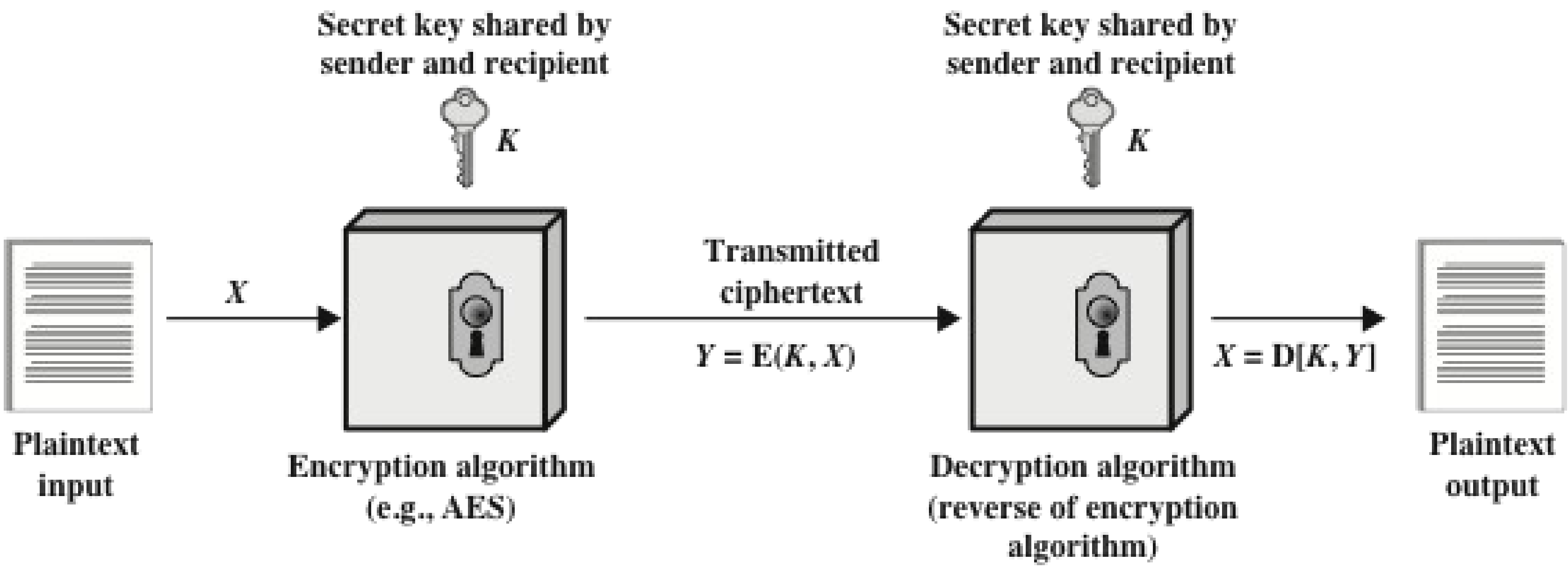


Figure 2.1 Simplified Model of Symmetric Encryption

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
- 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
  - Manufacturers can and have developed low-cost chip implementations of data encryption algorithms
  - These chips are widely available and incorporated into a number of products

Cryptography

Cryptographic systems are generically classified along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext. All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (i.e., that all operations be reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.

2. The number of keys used. If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver each use a different key, the system is referred to as asymmetric, two-key, or public-key encryption.

3. The way in which the plaintext is processed. 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.

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

Table 2.1 Types of Attacks on Encrypted Messages

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

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

Feistel Cipher Design Elements

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.

• 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 from 10 to 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.

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 a hardware implementation. Accordingly, 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 cryptanalyze, there is great benefit in making the algorithm easy to analyze. That is, 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. DES, for example, does not have an easily analyzed functionality.

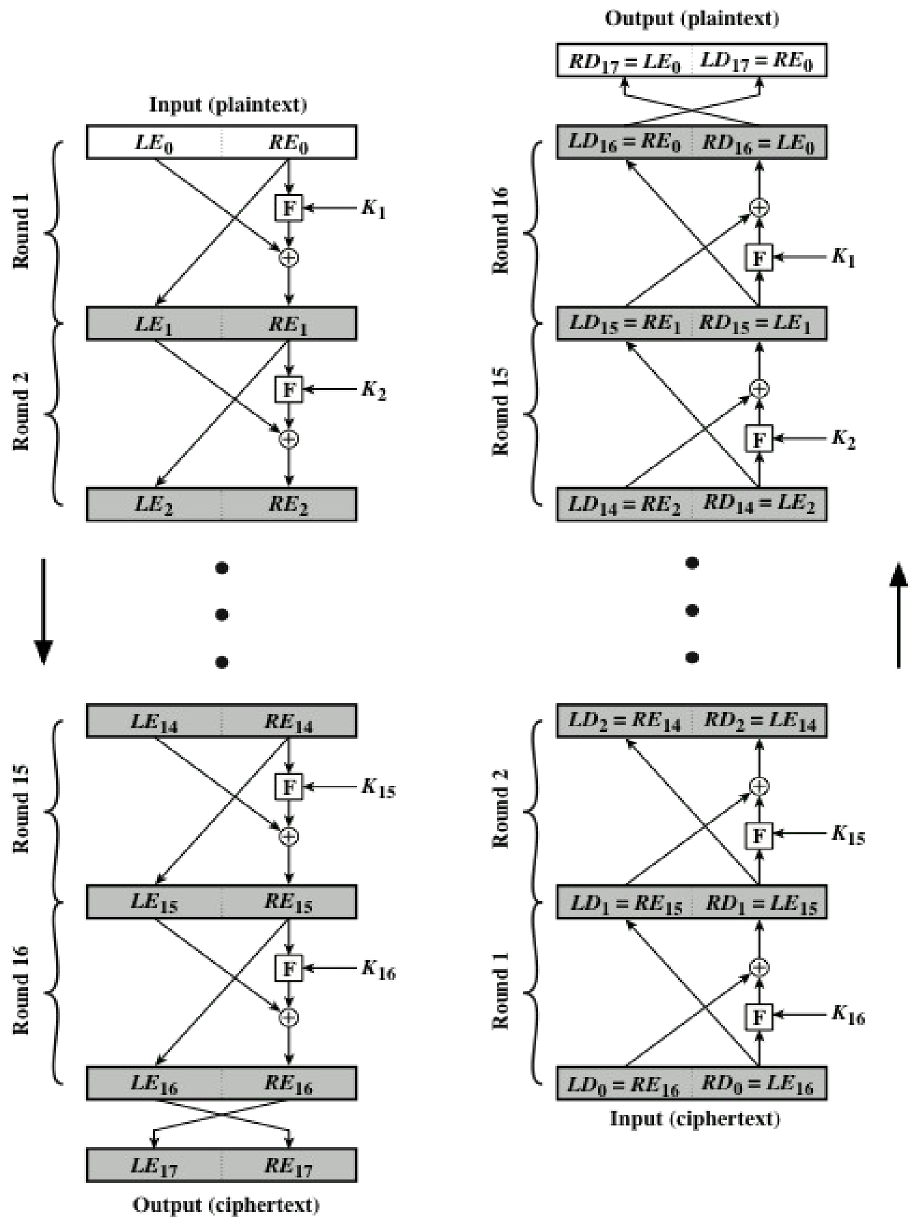


Figure 2.2 Feistel Encryption and Decryption (16 rounds)

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)

DES algorithm

- Description of the algorithm:
  - Plaintext is 64 bits in length
  - Key is 56 bits in length
  - Structure is a minor variation of the Feistel network
  - 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
      - Speed of commercial, off-the-shelf processors threatens the security

Table 2.2

Average Time Required for Exhaustive Key Search

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years

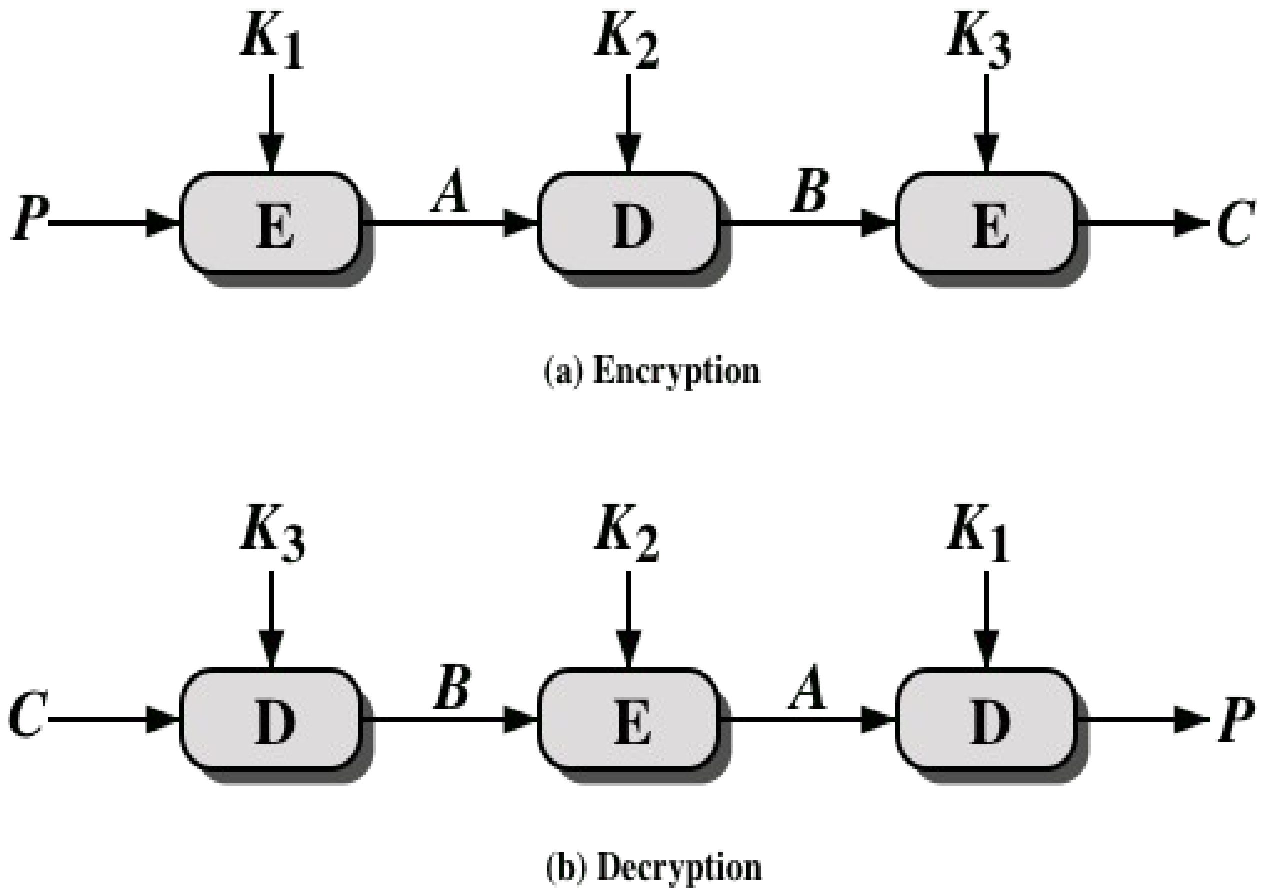


Figure 2.3 Triple DES

3DES guidelines

- FIPS 46-3 includes the following guidelines for 3DES:
  - 3DES is the FIPS-approved symmetric encryption algorithm of choice
  - 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)

- In 1997 NIST issued a call for proposals for a new AES:
  - Should have a security strength equal to or 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
- NIST selected Rijndael as the proposed AES algorithm
  - FIPS PUB 197
  - Developers were two cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen

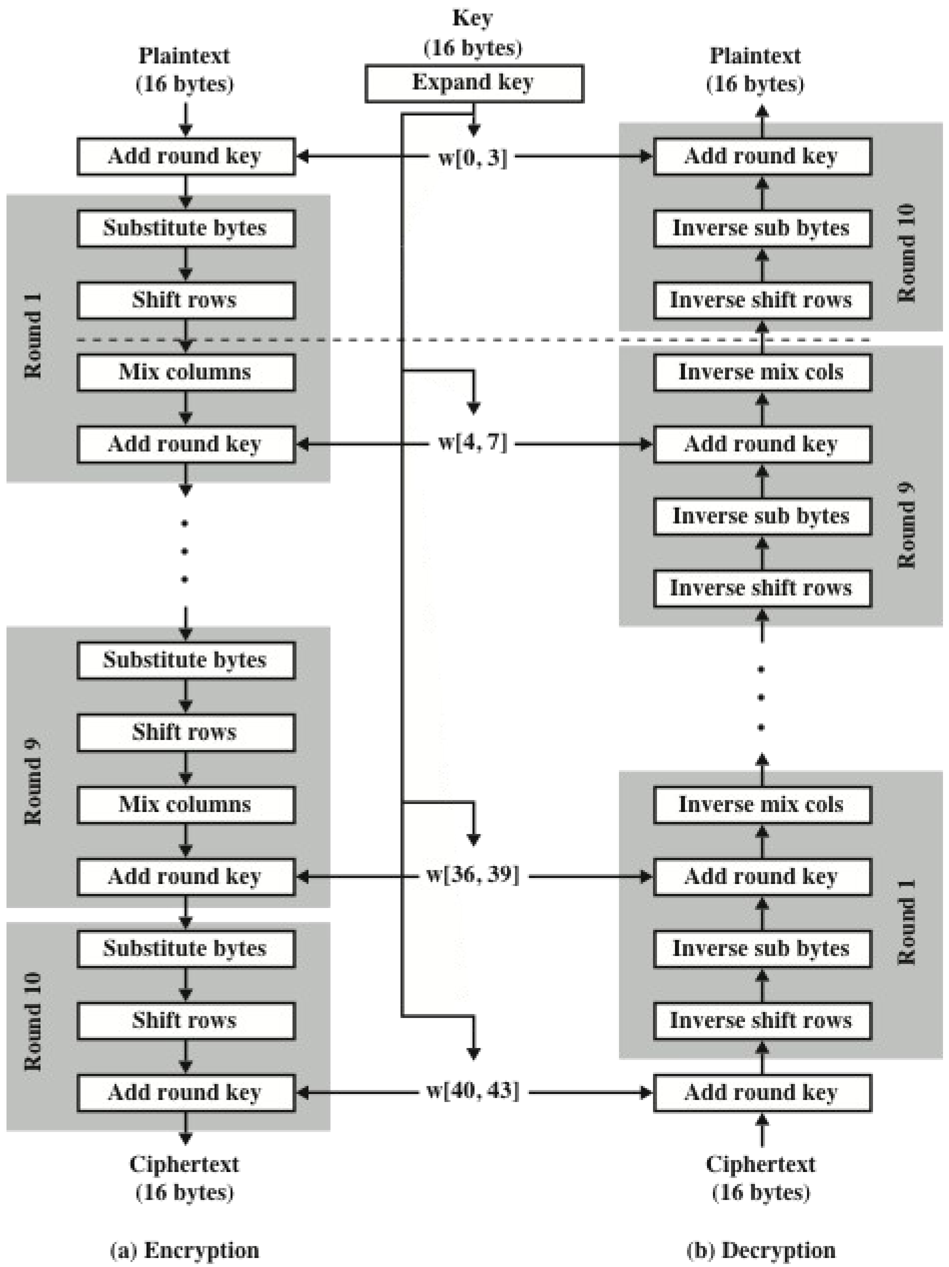


Figure 2.4 AES Encryption and Decryption

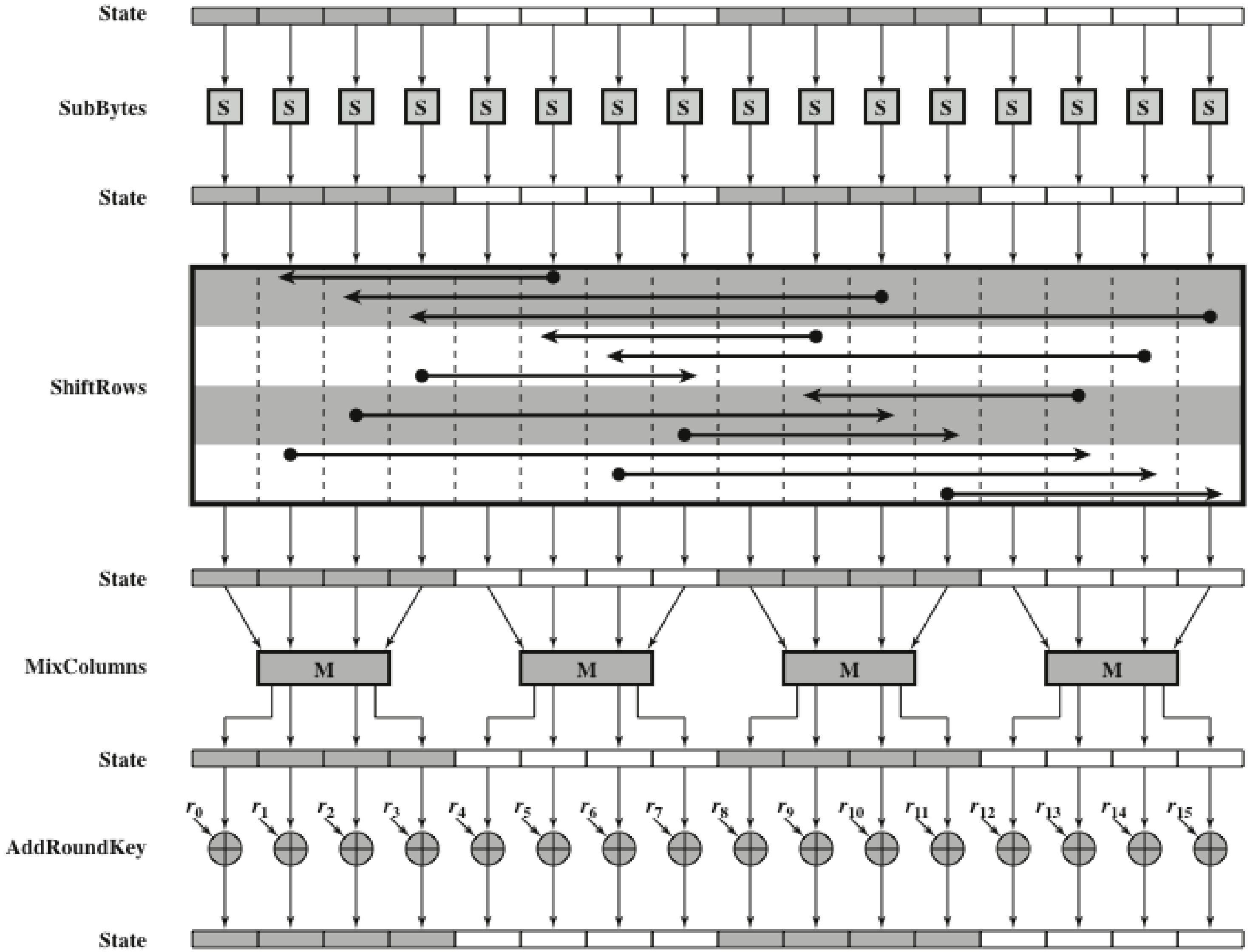


Figure 2.5 AES Encryption Round

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

Traditionally, the concern in the generation of a sequence of allegedly random numbers has been that the sequence of numbers be random in some well defined statistical sense.

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; that is, the frequency of occurrence of ones and zeros should be approximately the same.

- Independence: No one subsequence in the sequence can be inferred from the others.

Although there are well-defined tests for determining that a sequence of numbers matches a particular distribution, such as the uniform distribution, there is no such test to “prove” independence. Rather, a number of tests can be applied to demonstrate if a sequence does not exhibit independence. The general strategy is to apply a number of such 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

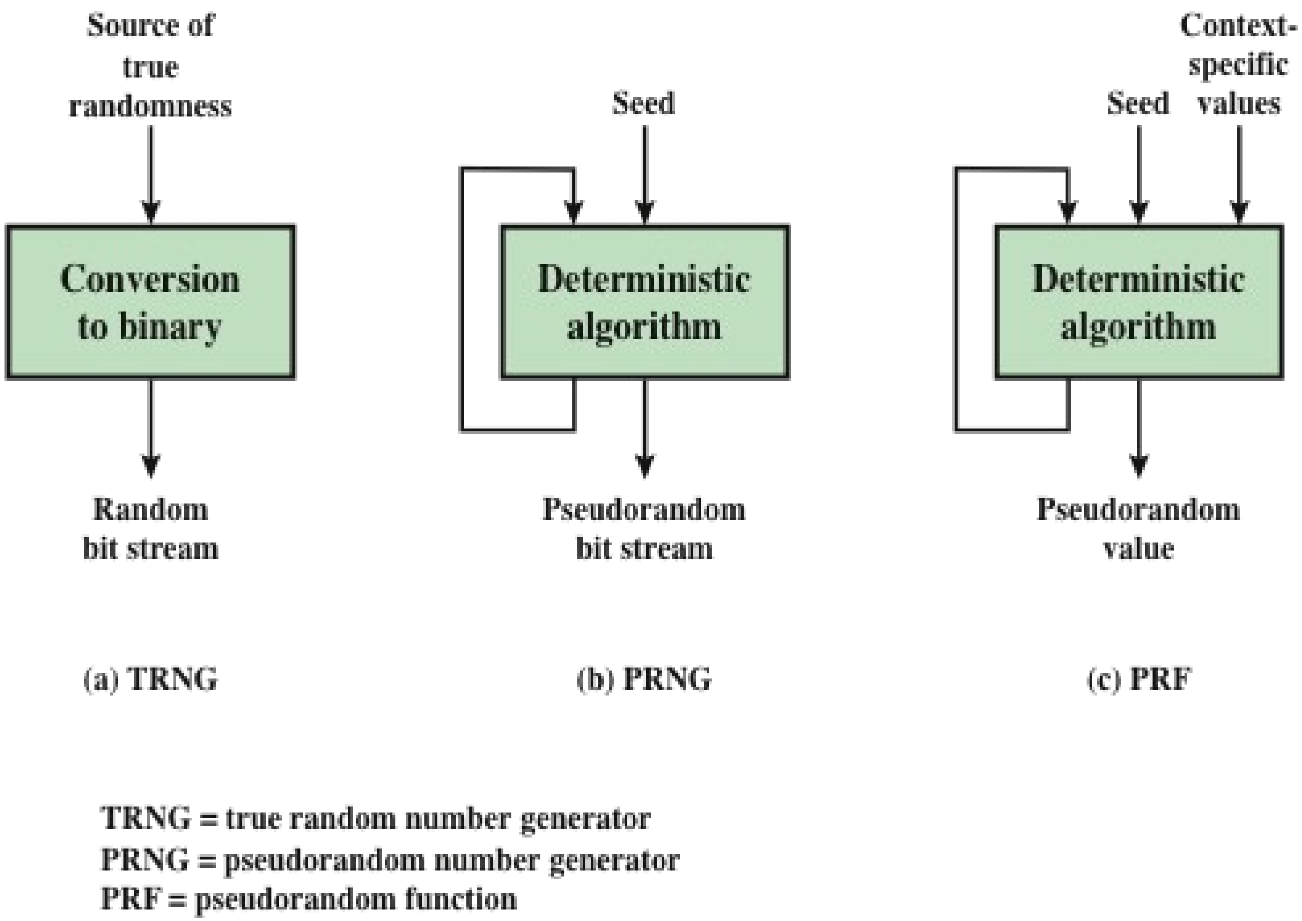


Figure 2.6 Random and Pseudorandom Number Generators

Algorithm design

Cryptographic PRNGs have been the subject of much research over the years, and a wide variety of algorithms have been developed. These fall roughly into two categories:

• Purpose-built algorithms: These are algorithms designed specifically and solely for the purpose of generating pseudorandom bit streams. Some of these algorithms are used for a variety of PRNG applications; several of these are described in the next section. Others are designed specifically for use in a stream cipher. The most important example of the latter is RC4, described in the next section.

• Algorithms based on existing cryptographic algorithms: Cryptographic algorithms have the effect of randomizing input. Indeed, this is a requirement of such algorithms. For example, if a symmetric block cipher produced ciphertext that had certain regular patterns in it, it would aid in the process of cryptanalysis. Thus, cryptographic algorithms 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

Any of these approaches can yield a cryptographically strong PRNG. A purpose-built algorithm may be provided by an operating system for general use. For applications that already use certain cryptographic algorithms for encryption or authentication, it makes sense to re-use the same code for the PRNG. Thus, all of these approaches are in common use.

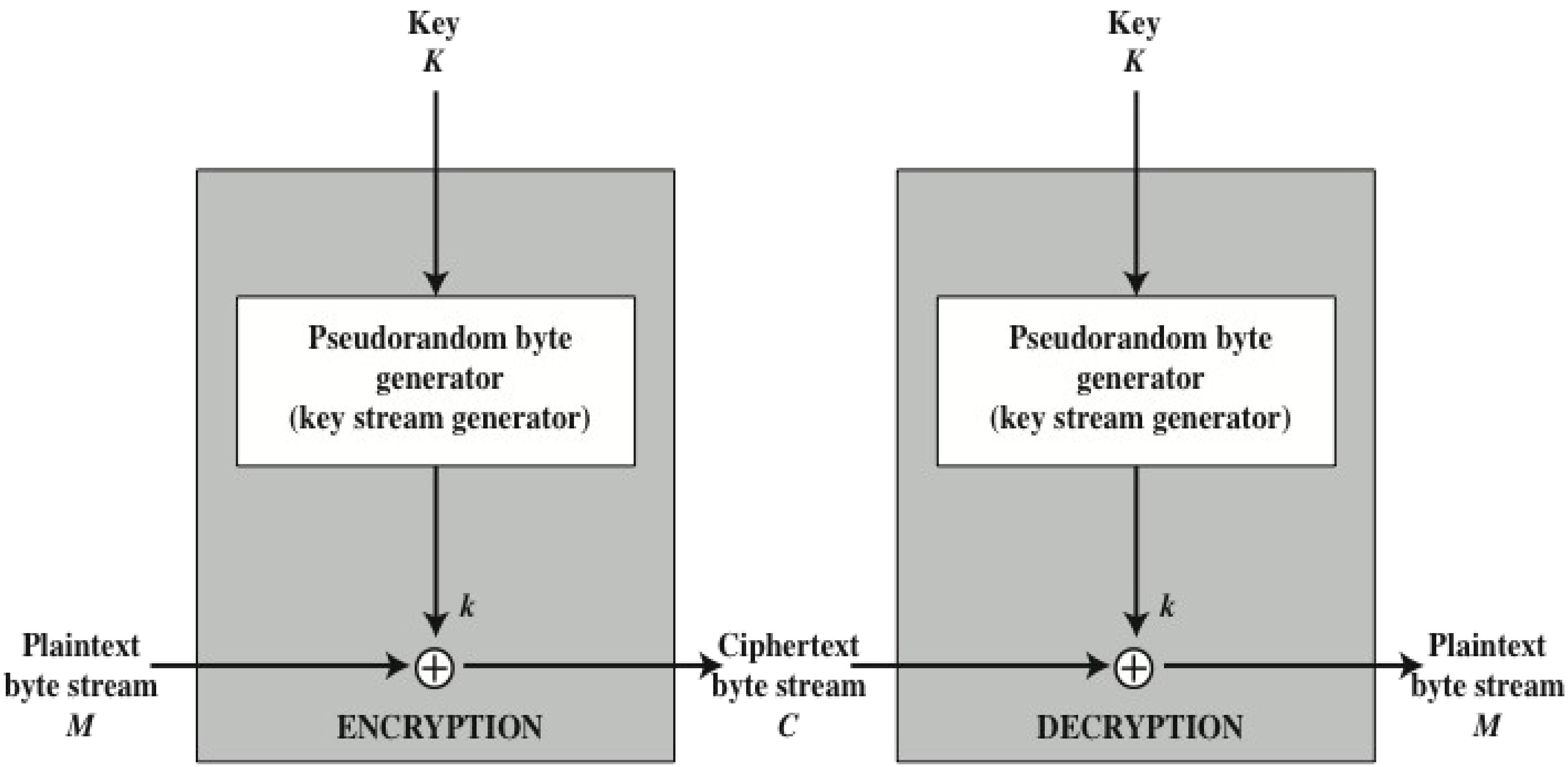


Figure 2.7 Stream Cipher Diagram

Stream Cipher design considerations

- The encryption sequence should have a large period
  - The longer the period of repeat, the more difficult it will be to do cryptanalysis
- The keystream should approximate the properties of a true random number stream as close as possible
  - The more random-appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult
- 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

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
- 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

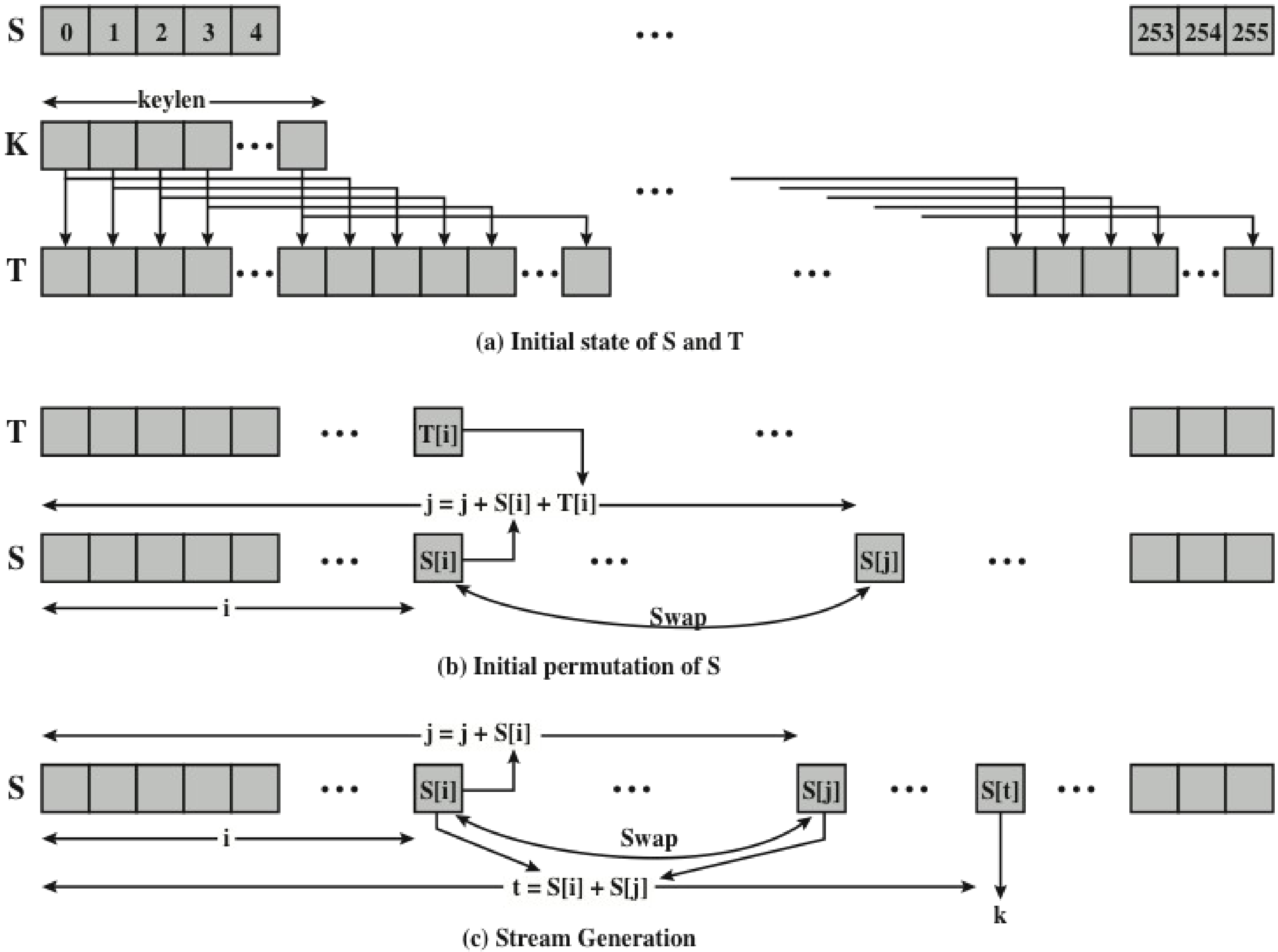


Figure 2.8 RC4

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
- Five modes of operation have been defined by NIST
  - Intended to cover virtually all of 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

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

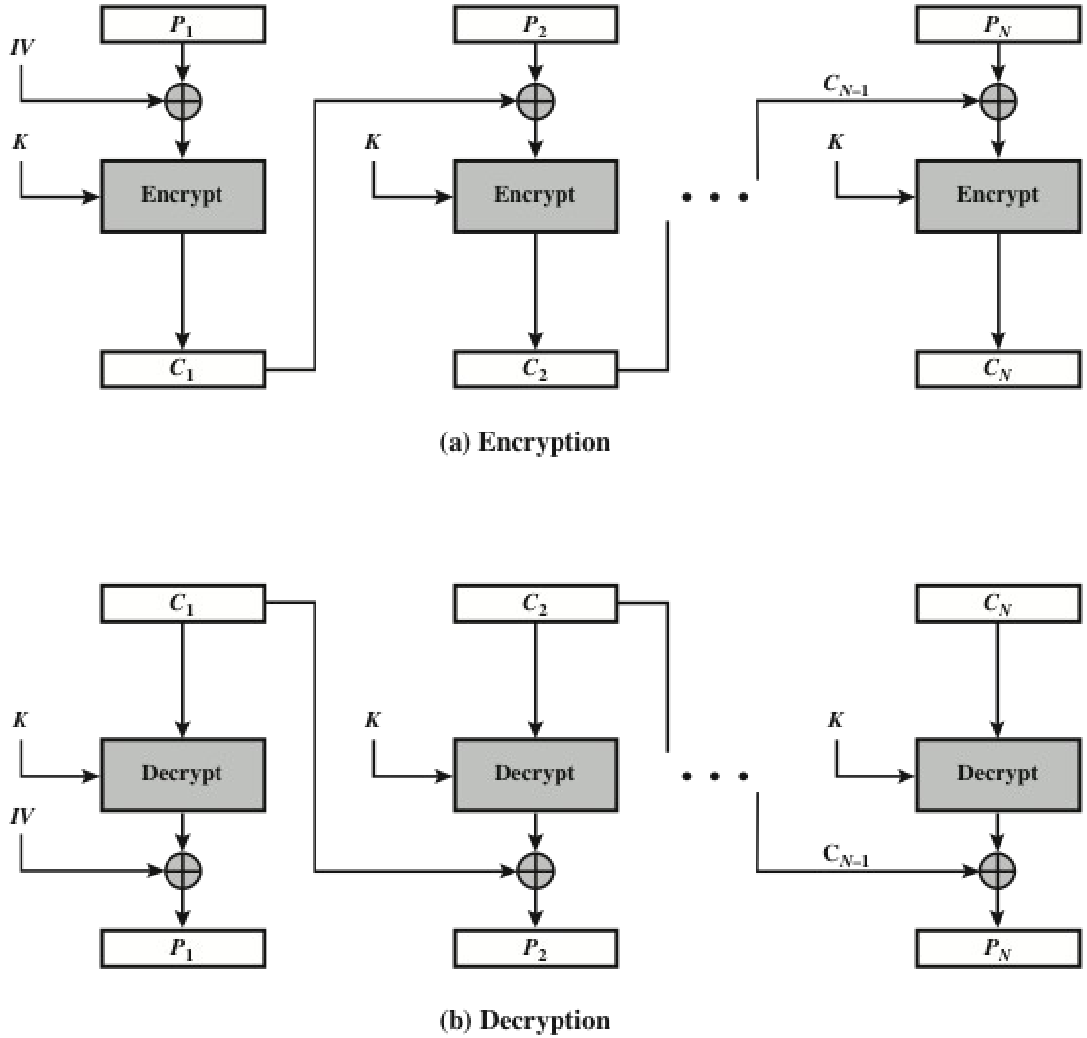


Figure 2.9 Cipher Block Chaining (CBC) Mode

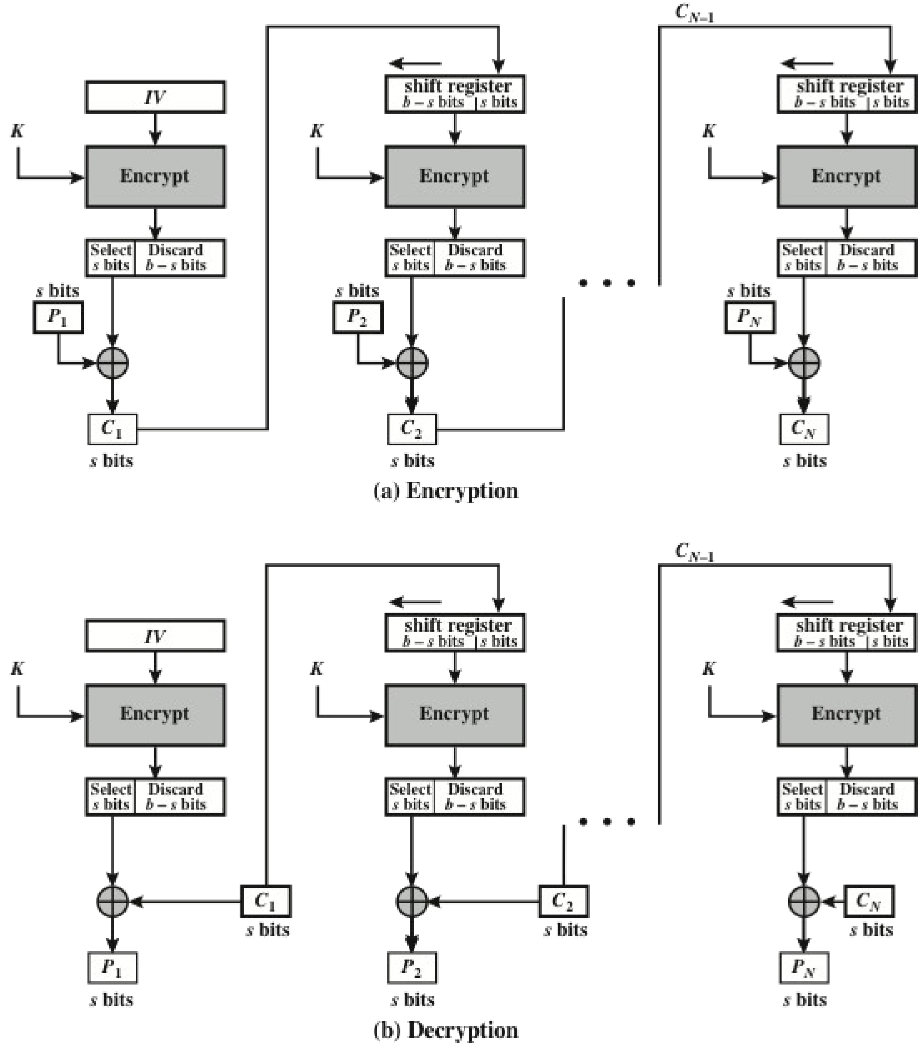


Figure 2.10 s-bit Cipher Feedback (CFB) Mode

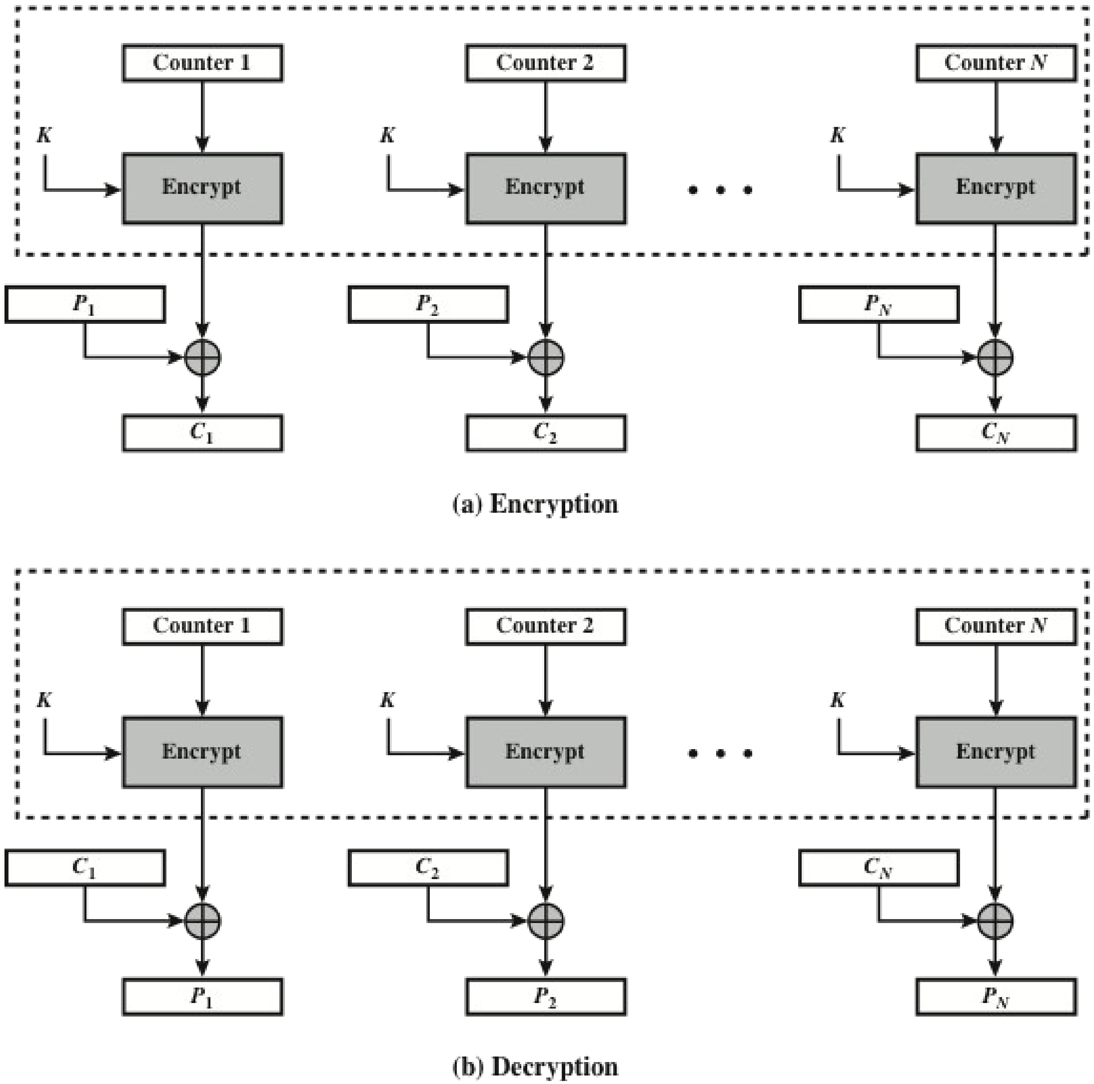


Figure 2.11 Counter (CTR) Mode

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

Advantages of CTR mode

- Hardware efficiency
  - Encryption/decryption can be done in parallel on multiple blocks of plaintext or ciphertext
  - Throughput is only limited by the amount of parallelism that is achieved
- 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
- Random access
  - The *i*th block of plaintext or ciphertext can be processed in random-access fashion
- Provable security
  - It can be shown that CTR is at least as secure as the other modes discussed in this section
- Simplicity
  - Requires only the implementation of the encryption algorithm and not the decryption algorithm