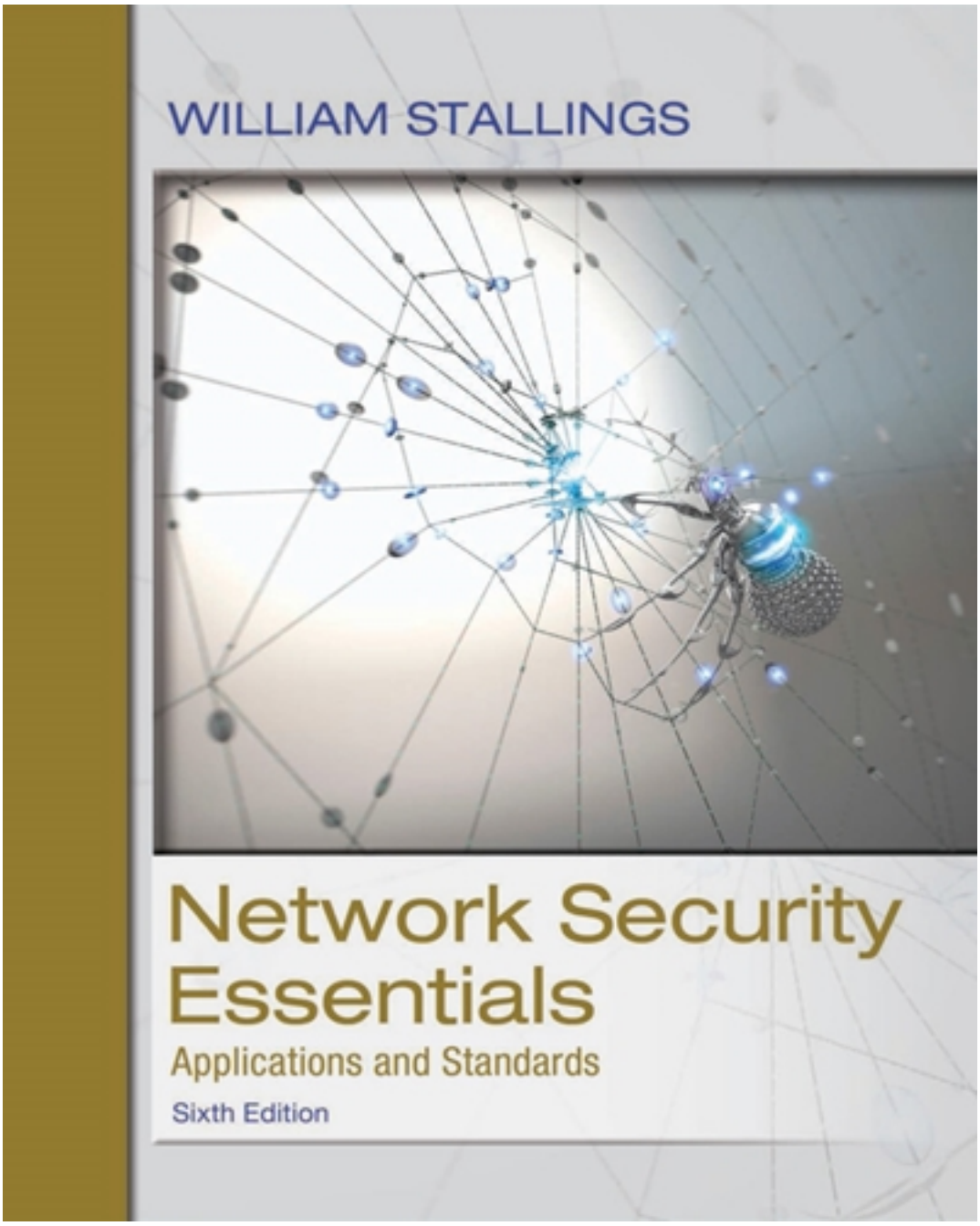
Solutions for Network Security Essentials Applications and Standards 6th Edition by Stallings

CLICK HERE TO ACCESS COMPLETE Solutions



Solutions

CHAPTER 2 SYMMETRIC ENCRYPTION AND MESSAGE CONFIDENTIALITY

ANSWERS TO QUESTIONS

2.1 Plaintext, encryption algorithm, secret key, ciphertext, decryption algorithm.

2.2 Permutation and substitution.

2.3 One secret key.

2.4 A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. 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.

2.5 Cryptanalysis and brute force.

2.6 In some modes, the plaintext does not pass through the encryption function, but is XORed with the output of the encryption function. The math works out that for decryption in these cases, the encryption function must also be used.

2.7 With triple encryption, a plaintext block is encrypted by passing it through an encryption algorithm; the result is then passed through the same encryption algorithm again; the result of the second encryption is passed through the same encryption algorithm a third time. Typically, the second stage uses the decryption algorithm rather than the encryption algorithm.

2.8 There is no cryptographic significance to the use of decryption for the second stage. Its only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES by repeating the key.

ANSWERS TO PROBLEMS

2.1 a.

2	8	10	7	9	6	3	1	4	5
C	R	Y	P	T	O	G	A	H	I
B	E	A	T	T	H	E	T	H	I
R	D	P	I	L	L	A	R	F	R
O	M	T	H	E	L	E	F	T	O
U	T	S	I	D	E	T	H	E	L
Y	C	E	U	M	T	H	E	A	T
R	E	T	O	N	I	G	H	T	A
T	S	E	V	E	N	I	F	Y	O
U	A	R	E	D	I	S	T	R	U
S	T	F	U	L	B	R	I	N	G
T	W	O	F	R	I	E	N	D	S

4	2	8	10	5	6	3	7	1	9
N	E	T	W	O	R	K	S	C	U
T	R	F	H	E	H	F	T	I	N
B	R	O	U	Y	R	T	U	S	T
E	A	E	T	H	G	I	S	R	E
H	F	T	E	A	T	Y	R	N	D
I	R	O	L	T	A	O	U	G	S
H	L	L	E	T	I	N	I	B	I
T	I	H	I	U	O	V	E	U	F
E	D	M	T	C	E	S	A	T	W
T	L	E	D	M	N	E	D	L	R
A	P	T	S	E	T	E	R	F	O

```
ISRNG  BUTLF  RRAFR  LIDLP  FTIYO  NVSEE  TBEHI  HTETA
EYHAT  TUCME  HRGTA  IOENT  TUSRU  IEADR  FOETO  LHMET
NTEDS  IFWRO  HUTEL  EITDS
```

b. The two matrices are used in reverse order. First, the ciphertext is laid out in columns in the second matrix, taking into account the order dictated by the second memory word. Then, the contents of the second matrix are read left to right, top to bottom and laid out in columns in the first matrix, taking into account the order dictated by the first memory word. The plaintext is then read left to right, top to bottom.

c. Although this is a weak method, it may have use with time-sensitive information and an adversary without immediate access to good cryptanalysis (e.g., tactical use). Plus it doesn't require anything more than paper and pencil, and can be easily remembered.

2.2 a. Let –X be the additive inverse of X. That is –X $\boxed{+}$ X = 0. Then:

$$P = (C \boxed{+} -K_1) \oplus K_0$$

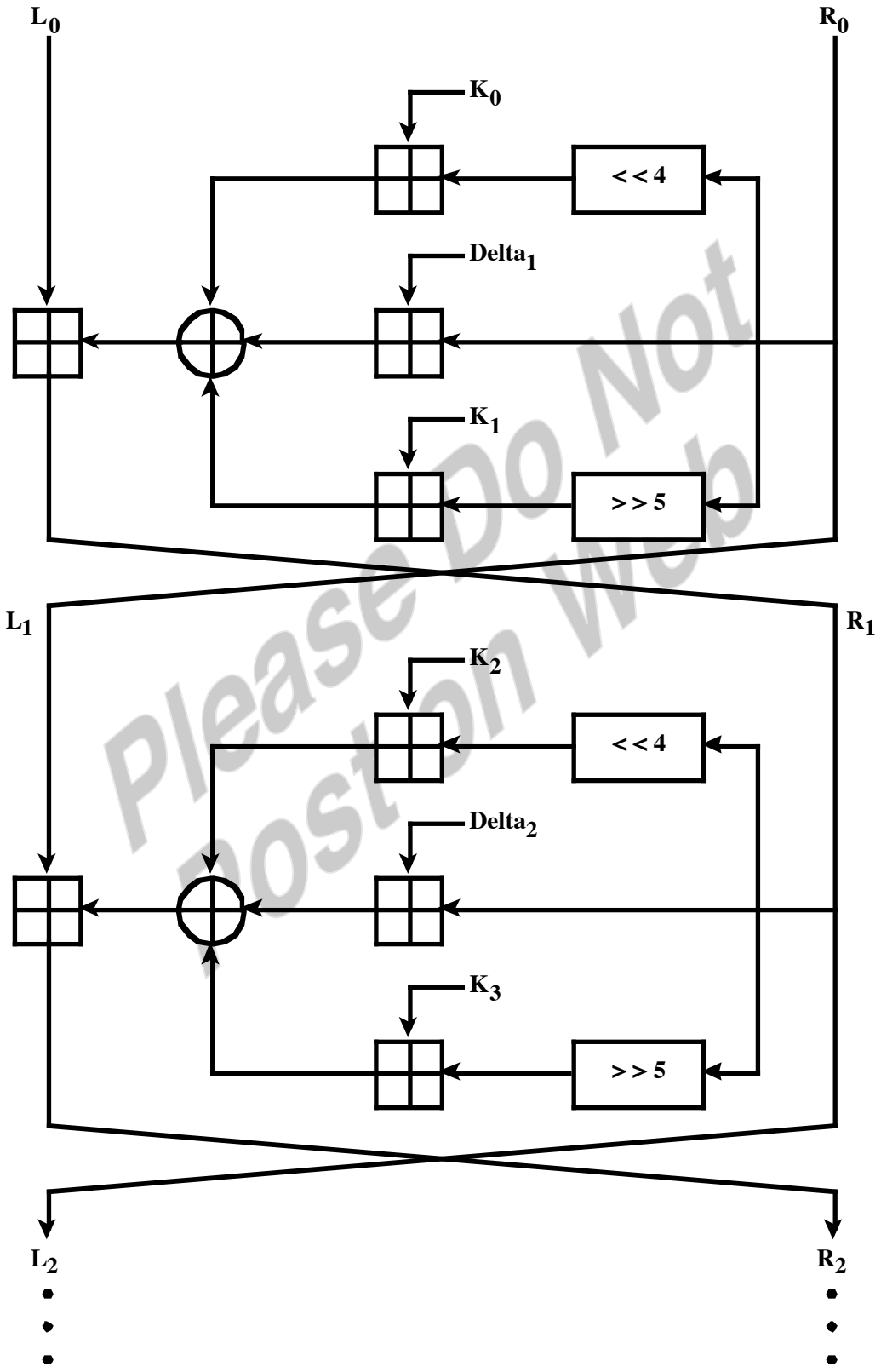
b. First, calculate –C'. Then –C' = (P' $\oplus$ K_0) $\boxed{+}$ (–K_1). We then have:

C $\boxed{+}$ –C' = (P $\oplus$ K_0) $\boxed{+}$ (P' $\oplus$ K_0)

However, the operations $\boxed{+}$ and $\oplus$ are not associative or distributive with one another, so it is not possible to solve this equation for K_0.

2.3 a. The constants ensure that encryption/decryption in each round is different.

b. First two rounds:



c. First, let's define the encryption process:

$L_2 = L_0 \boxplus [(R_0 << 4) \boxplus K_0] \oplus [R_0 \boxplus \delta_1] \oplus [(R_0 >> 5) \boxplus K_1]$

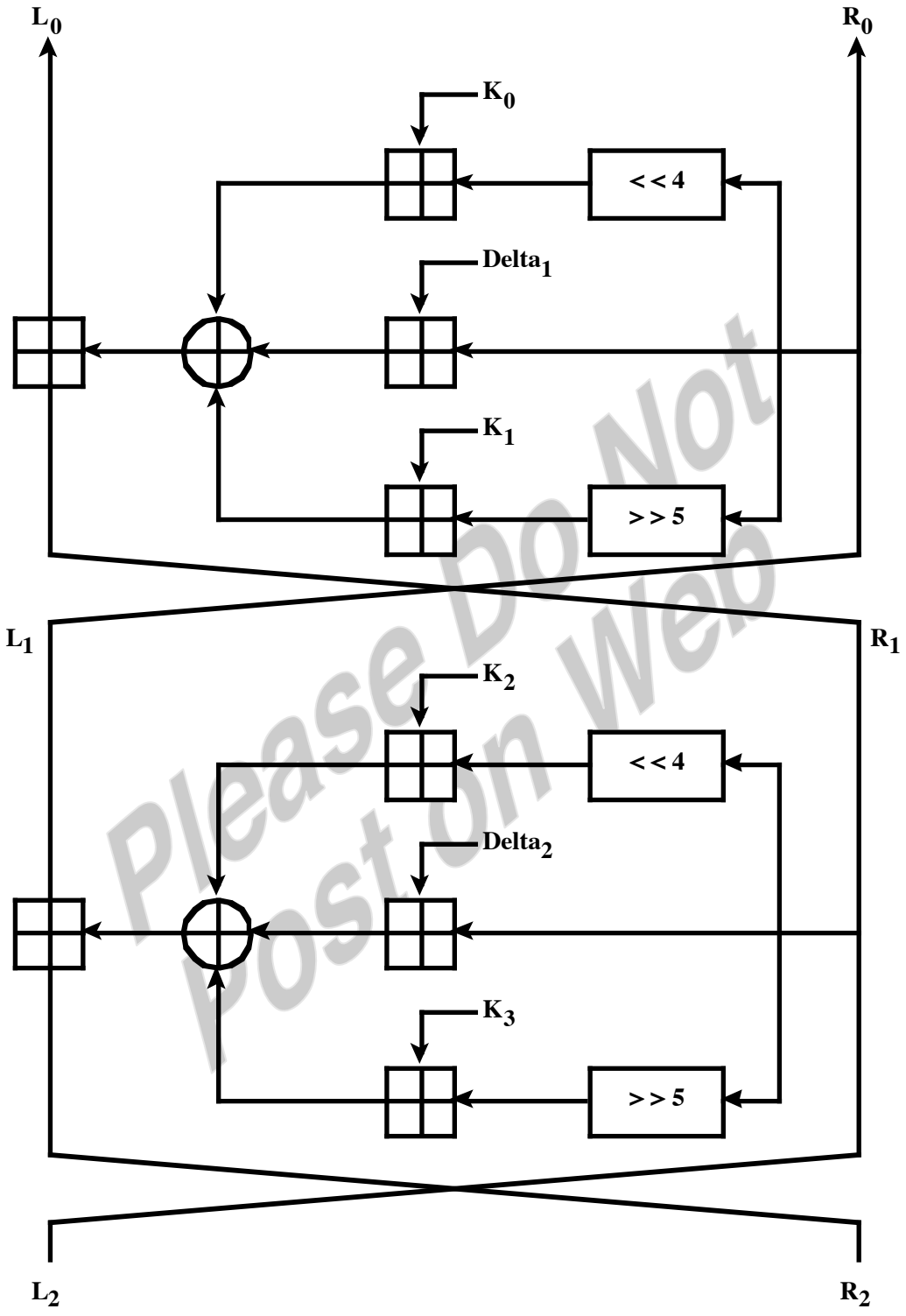
$R_2 = R_0 \boxplus [(L_2 << 4) \boxplus K_2] \oplus [L_2 \boxplus \delta_2] \oplus [(L_2 >> 5) \boxplus K_3]$

Now the decryption process. The input is the ciphertext (L_2, R_2), and the output is the plaintext (L_0, R_0). Decryption is essentially the same as encryption, with the subkeys and delta values applied in reverse order. Also note that it is not necessary to use subtraction because there is an even number of additions in each equation.

$R_0 = R_2 \boxplus [(L_2 << 4) \boxplus K_2] \oplus [L_2 \boxplus \delta_2] \oplus [(L_2 >> 5) \boxplus K_3]$

$L_0 = L_2 \boxplus [(R_0 << 4) \boxplus K_0] \oplus [R_0 \boxplus \delta_1] \oplus [(R_0 >> 5) \boxplus K_1]$

d.



2.4 To see that the same algorithm with a reversed key order produces the correct result, consider Figure 2.2, which shows the encryption process going down the left-hand side and the decryption process going up the right-hand side for a 16-round algorithm (the result would be the same for any number of rounds). For clarity, we use the notation LE_i and RE_i

for data traveling through the encryption algorithm and LD_i and RD_i for data traveling through the decryption algorithm. The diagram indicates that, at every round, the intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped. To put this another way, let the output of the ith encryption round be $LE_i||RE_i$ (L_i concatenated with R_i). Then the corresponding input to the (16 – i)th decryption round is $RD_i||LD_i$.

Let us walk through the figure to demonstrate the validity of the preceding assertions. To simplify the diagram, it is unwrapped, not showing the swap that occurs at the end of each iteration. But note that the intermediate result at the end of the ith stage of the encryption process is the 2w-bit quantity formed by concatenating LE_i and RE_i, and that the intermediate result at the end of the ith stage of the decryption process is the 2w-bit quantity formed by concatenating LD_i and RD_i. After the last iteration of the encryption process, the two halves of the output are swapped, so that the ciphertext is $RE_{16}||LE_{16}$. The output of that round is the ciphertext. Now take that ciphertext and use it as input to the same algorithm. The input to the first round is $RE_{16}||LE_{16}$, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process.

Now we would like to show that the output of the first round of the decryption process is equal to a 32-bit swap of the input to the sixteenth round of the encryption process. First, consider the encryption process. We see that:

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

On the decryption side:

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

The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$. Therefore, the output of the first round of the decryption process is $LE_{15}||RE_{15}$, which is the 32-bit swap of the input to the sixteenth round of the encryption. This

correspondence holds all the way through the 16 iterations, as is easily shown. We can cast this process in general terms. For the ith iteration of the encryption algorithm:

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

Rearranging terms:

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

Thus, we have described the inputs to the ith iteration as a function of the outputs, and these equations confirm the assignments shown in the right-hand side of the following figure.

Finally, we see that the output of the last round of the decryption process is $RE_0||LE_0$. A 32-bit swap recovers the original plaintext, demonstrating the validity of the Feistel decryption process.

2.5 Because of the key schedule, the round functions used in rounds 9 through 16 are mirror images of the round functions used in rounds 1 through 8. From this fact we see that encryption and decryption are identical. We are given a ciphertext c. Let $m' = c$. Ask the encryption oracle to encrypt m'. The ciphertext returned by the oracle will be the decryption of c.

2.6 For $1 \le i \le 128$, take $c_i \in \{0, 1\}^{128}$ to be the string containing a 1 in position i and then zeros elsewhere. Obtain the decryption of these 128 ciphertexts. Let $m_1, m_2, \ldots, m_{128}$ be the corresponding plaintexts. Now, given any ciphertext c which does not consist of all zeros, there is a unique nonempty subset of the c_i's which we can XOR together to obtain c. Let $I(c) \subseteq \{1, 2, \ldots, 128\}$ denote this subset. Observe

$$c = \bigoplus_{i \in I(c)} c_i = \bigoplus_{i \in I(c)} E(m_i) = E\left(\bigoplus_{i \in I(c)} m_i\right)$$

Thus, we obtain the plaintext of c by computing $\bigoplus_{i \in I(c)} m_i$. Let **0** be the all-zero string. Note that **0** = **0** ⊕ **0**. From this we obtain E(**0**) = E(**0** ⊕ **0**) = E(**0**) ⊕ E(**0**) = **0**. Thus, the plaintext of c = **0** is m = **0**. Hence we can decrypt every $c \in \{0, 1\}^{128}$.

2.7 a.

Pair	Probability
00	$(0.5 - \partial)^2 = 0.25 - \partial + \partial^2$
01	$(0.5 - \partial) \times (0.5 + \partial) = 0.25 - \partial^2$
10	$(0.5 + \partial) \times (0.5 - \partial) = 0.25 - \partial^2$
11	$(0.5 + \partial)^2 = 0.25 + \partial + \partial^2$

b. Because 01 and 10 have equal probability in the initial sequence, in the modified sequence, the probability of a 0 is 0.5 and the probability of a 1 is 0.5.

c. The probability of any particular pair being discarded is equal to the probability that the pair is either 00 or 11, which is $0.5 + 2\partial^2$, so the expected number of input bits to produce x output bits is $x/(0.25 - \partial^2)$.

d. The algorithm produces a totally predictable sequence of exactly alternating 1's and 0's.

2.8 a. For the sequence of input bits $a_1, a_2, \ldots, a_n$, the output bit b is defined as:

$$b = a_1 \oplus a_2 \oplus \ldots \oplus a_n$$

b. $0.5 - 2\partial^2$

c. $0.5 - 8\partial^4$

d. The limit as n goes to infinity is 0.5.

2.9 Use a key of length 255 bytes. The first two bytes are zero; that is K[0] = K[1] = 0. Thereafter, we have: K[2] = 255; K[3] = 254; … K[255]= 2.

2.10 a. Simply store i, j, and S, which requires $8 + 8 + (256 \times 8) = 2064$ bits

b. The number of states is $[256! \times 256^2] \approx 2^{1700}$. Therefore, 1700 bits are required.

2.11 a. By taking the first 80 bits of $v \| c$, we obtain the initialization vector, v. Since v, c, k are known, the message can be recovered (i.e., decrypted) by computing $\text{RC4}(v \| k) \oplus c$.

b. If the adversary observes that $v_i = v_j$ for distinct i, j then he/she knows that the same key stream was used to encrypt both m_i and m_j. In this case, the messages m_i and m_j may be vulnerable to the type of cryptanalysis carried out in part (a).

c. Since the key is fixed, the key stream varies with the choice of the 80-bit v, which is selected randomly. Thus, after approximately

$\sqrt{\frac{\pi}{2}2^{80}} \approx 2^{40}$ messages are sent, we expect the same *v*, and hence the same key stream, to be used more than once.

d. The key *k* should be changed sometime before 2^{40} messages are sent.

2.12 a. No. For example, suppose C_1 is corrupted. The output block P_3 depends only on the input blocks C_2 and C_3.

b. An error in P_1 affects C_1. But since C_1 is input to the calculation of C_2, C_2 is affected. This effect carries through indefinitely, so that all ciphertext blocks are affected. However, at the receiving end, the decryption algorithm restores the correct plaintext for blocks except the one in error. You can show this by writing out the equations for the decryption. Therefore, the error only effects the corresponding decrypted plaintext block.

2.13 In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In CBC decryption, however, the input blocks for the inverse cipher function (i.e., the ciphertext blocks) are immediately available, so that multiple inverse cipher operations can be performed in parallel.

2.14 If an error occurs in transmission of ciphertext block C_i, then this error propagates to the recovered plaintext blocks P_i and P_{i+1}.

2.15 After decryption, the last byte of the last block is used to determine the amount of padding that must be stripped off. Therefore there must be at least one byte of padding.

2.16 a. Assume that the last block of plaintext is only *L* bytes long, where $L < 2w/8$. The encryption sequence is as follows (The description in RFC 2040 has an error; the description here is correct.):

1. Encrypt the first $(N - 2)$ blocks using the traditional CBC technique.
2. XOR P_{N-1} with the previous ciphertext block C_{N-2} to create Y_{N-1}.
3. Encrypt Y_{N-1} to create E_{N-1}.
4. Select the first *L* bytes of E_{N-1} to create C_N.
5. Pad P_N with zeros at the end and exclusive-OR with E_{N-1} to create Y_N.
6. Encrypt Y_N to create C_{N-1}.

The last two blocks of the ciphertext are C_{N-1} and C_N.

b. $P_{N-1} = C_{N-2} \oplus D(K, [C_N \| X])$

$P_N \| X = (C_N \| 00\ldots0) \oplus D(K, [C_{N-1}])$

P_N = left-hand portion of $(P_N \| X)$

where $\|$ is the concatenation function

2.17 **a.** Assume that the last block (P_N) has j bits. After encrypting the last full block (P_{N-1}), encrypt the ciphertext (C_{N-1}) again, select the leftmost j bits of the encrypted ciphertext, and XOR that with the short block to generate the output ciphertext.

b. While an attacker cannot recover the last plaintext block, he can change it systematically by changing individual bits in the ciphertext. If the last few bits of the plaintext contain essential information, this is a weakness.

2.18 Nine plaintext characters are affected. The plaintext character corresponding to the ciphertext character is obviously altered. In addition, the altered ciphertext character enters the shift register and is not removed until the next eight characters are processed.

PROJECT GUIDE:

PRACTICAL SECURITY ASSESSMENTS

Lawrie Brown

Supplement to
Cryptography and Network Security, Fifth Edition
William Stallings
Prentice Hall 2010
ISBN-10: 0136097049
http://williamstallings.com/Crypto/Crypto5e.html

Overview

Examining the current infrastructure and practices of an existing organization is one of the best ways of developing skills in assessing its security posture. Students, working either individually or in small groups, select a suitable small- to medium-sized organization. They then interview some key personnel in that organization in order to conduct a suitable selection of security risk assessment and review tasks as it relates to the organization's IT infrastructure and practices. As a result, they can then recommend suitable changes to improve the organization's IT security. These activities help students develop an appreciation of current security practices, and the skills needed to review these and recommend changes.

Initial Tasks

Organization Selection

Students need to first select the organization they will use in conjunction with the course lecturer. It should be of small to medium size, with sufficient people to require a modest internal IT network with internal servers, as well as a connection to some larger external network (e.g. Internet) and some external servers. While it does not have to be an actual organization, it will greatly assist if it is, and you are familiar with it and its IT operations and requirements. Suitable choices of organization might be: a small to medium sized business, a division/unit of a larger business/government department, a school, a university department, or other moderate sized organization. It should have some 10's of people using its systems. It is important that you obtain permission from your selected organization undertake this type of analysis. While the results can be kept confidential, they can hopefully benefit both parties (you from doing the exercise, and the organization from the results!)

It is useful to view your relationship with the organization as that of an external security consultant, brought in to advise on aspects of its IT security. So the organization should be one you either know, or perhaps better still, has someone you

can interview about its IT systems and use. Hence good choices would be an organization where a relative or friend of the student or a team member works, or for which there is some other existing relationship.

Security Risk Assessment

The initial task is to conduct a preliminary security risk assessment for your chosen organization, using the process described in chapters 16 and 17 of the text. Specifically, the goal is to perform an initial analysis for a combined risk assessment approach. This will involve an attempt at each of these steps:

1. Establish Context and Assets (organizational risk profile and key IT assets)
2. Identify Risks to Key Assets (threats, threats sources, vulnerabilities)
3. Analyze those Risks (existing controls, likelihood, consequence, resultant risk)
4. Assess & Prioritize Risks (document, evaluate, suggest treatment)
5. Develop a Risk Treatment Plan (how treat, possible additional controls)

Given the limited time available for doing this work as an exercise, it will be necessary to limit the time taken and outcomes from this process. The goal is not to conduct a comprehensive assessment, but simply to attempt each step, provide an initial assessment, and to develop an understanding of the issues involved in undertaking this process.

The outcome from this task should be a summary of the context, and an annotated risk register identifying some key assets and significant threats to them. The annotations should identify the reasons and sources for all items in the risk register.

Alternative Subsequent Tasks

Once the organization has been selected, and a preliminary security risk assessment conducted, a selection of the following tasks may be undertaken, as appropriate to the

organization of course teaching outcomes. The outcome from each of these tasks would be a brief report detailing and justifying their findings.

User Authentication and Access Controls

The student / teams should review the current mechanisms used for IT system user authentication by their organization. Given the outcomes of the risk assessment, you should identify whether changing these mechanisms would be an appropriate control to improve its security posture. If so, suggest what improved mechanisms could be used, and obtain an indication of the costs involved in their implementation.
Also you should review the categorization of users into groups that may then be used for access control decisions to IT resources. Indicate whether you believe the existing groupings are appropriate, or whether there are better alternatives.

Database Security

If the organization uses databases on one or more DBMS systems, the student / teams should review the current security mechanisms used by these systems. Identify the groups of users defined, and the access controls the DBMS assigns to them. This is likely best done as an extension of the previous task. Indicate whether you believe the existing groupings and access controls are appropriate, or whether there are better alternatives.

PC / Workstation Security

Given the known problems with import of malware onto client PC's or workstations, the student / teams should review the current mechanisms used to configure and update such systems in their organization, and identify any anti-virus, anti-spyware, and personal firewall products currently used. Suggest whether you believe these mechanisms should be improved, stating your reasons.

Server Security

The student / teams should review the management and security configuration of a key server for their organization. Ideally it should be one identified in the preliminary risk assessment as being subject to a significant risk, and hence needing improved security. You need to decide on which server you will analyze, and detail why it is selected it, and what its importance is to the organization. Then detail the server's security requirements, identifying:

- what information it contains, and how sensitive that information is
- what applications it runs, how they manipulate the information stored, and how critical their availability is
- who has access to the system, and what type of access they have
- who has administrative access to the system, and how this is controlled
- what change management procedures are used to manage its configuration

The student / teams should then detail how they would alter a basic operating system and applications installation process to provide a suitable level of security on this server. They should research ways of installing and armoring the chosen O/S, and the key applications used, to suit their server's security requirements. The information in chapters 23 and 24 in the text will likely be of use, as will numerous guides on armoring and latest vulnerability reports provided by organizations such as CERT, CIO, NIST/NSA, SANS etc. Your report should identify where and why changes should be made to a standard installation process, rather than simply reproducing pages of generic install information. Appropriate citations of key references should be given where more detailed information is appropriate.

Network Perimeter Security

The student / teams should review their organization's network perimeter security arrangements, that is their use of firewalls, intrusion detection/prevention systems etc. Given the preliminary risk assessment, you should review what access policy is being used for network traffic across the perimeter, and whether you would recommend changes to it. You should construct a table detailing the network services allowed to or across the network perimeter, of a form similar to that in Table 9.1. You should also state what the default access policy should be (discard or forward), with justification.

You should then suggest an appropriate firewall topology, being one of the options listed at the end of section 9.5 in the text, with details justifying its selection. If possibly, obtain a rough indication of the costs involved in implementing this topology.

Software Security

The student / teams should identify whether their organization uses critical software which is exposed to possible external attack. This would most likely be software running on an externally visible web server to handle responses to forms or other dynamic data handling. You should check what version of software is being used, what the most current version available is, and match this with any reports of known vulnerabilities in this software, as provided by organizations such as CERT, CIO, NIST/NSA, SANS etc. You should detail the threat posed to the organization by any known vulnerabilities, and whether you would recommend this software be upgraded, or hardened in some manner.

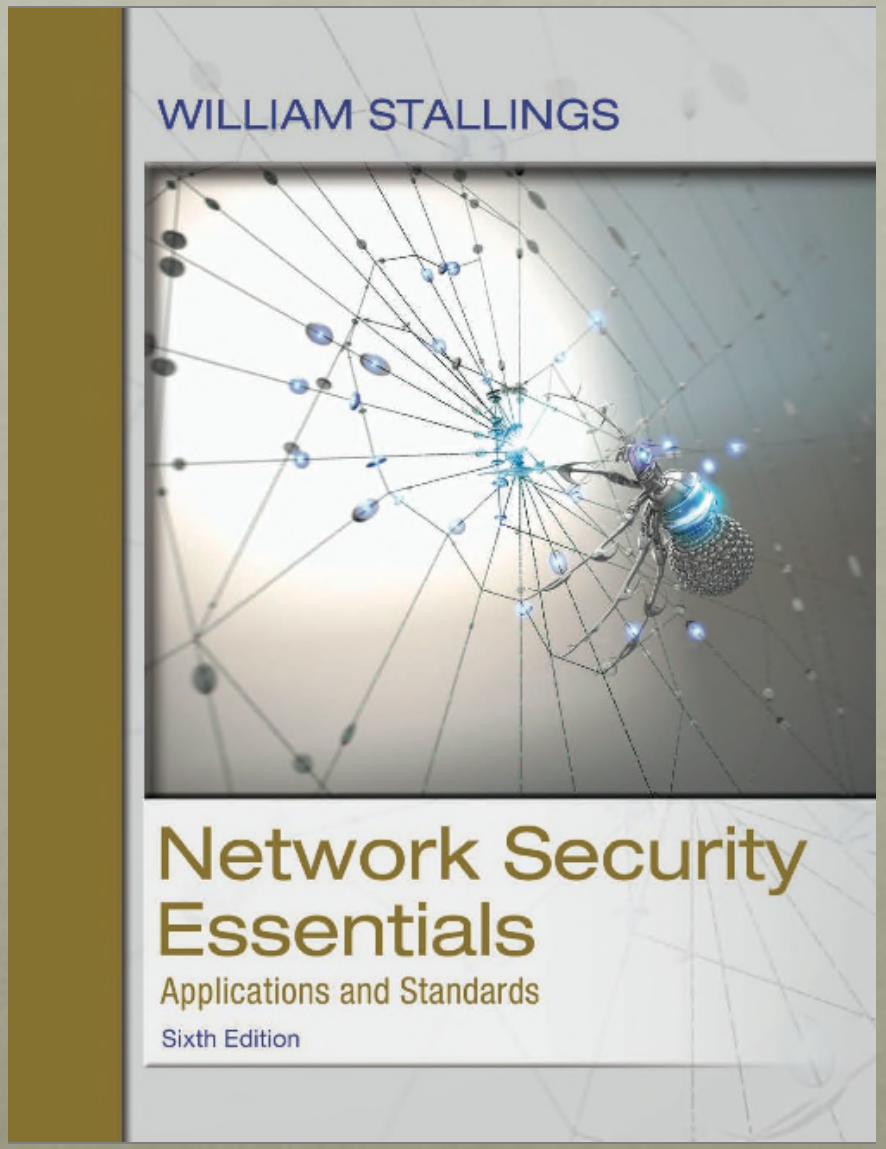
Security Policy

The student / teams should review the current organizational security policy. You should correlate the broad structure of the policy with the recommendations for security policy content given in resources such as ISO17799, COBIT, or Information Security Forum (as detailed in section 14.2 of the text). Indicate whether there are any areas not covered in the existing policy that you believe ought to be.

NETWORK SECURITY ESSENTIALS

Sixth Edition

by William Stallings



CHAPTER 2

Symmetric Encryption and Message Confidentiality

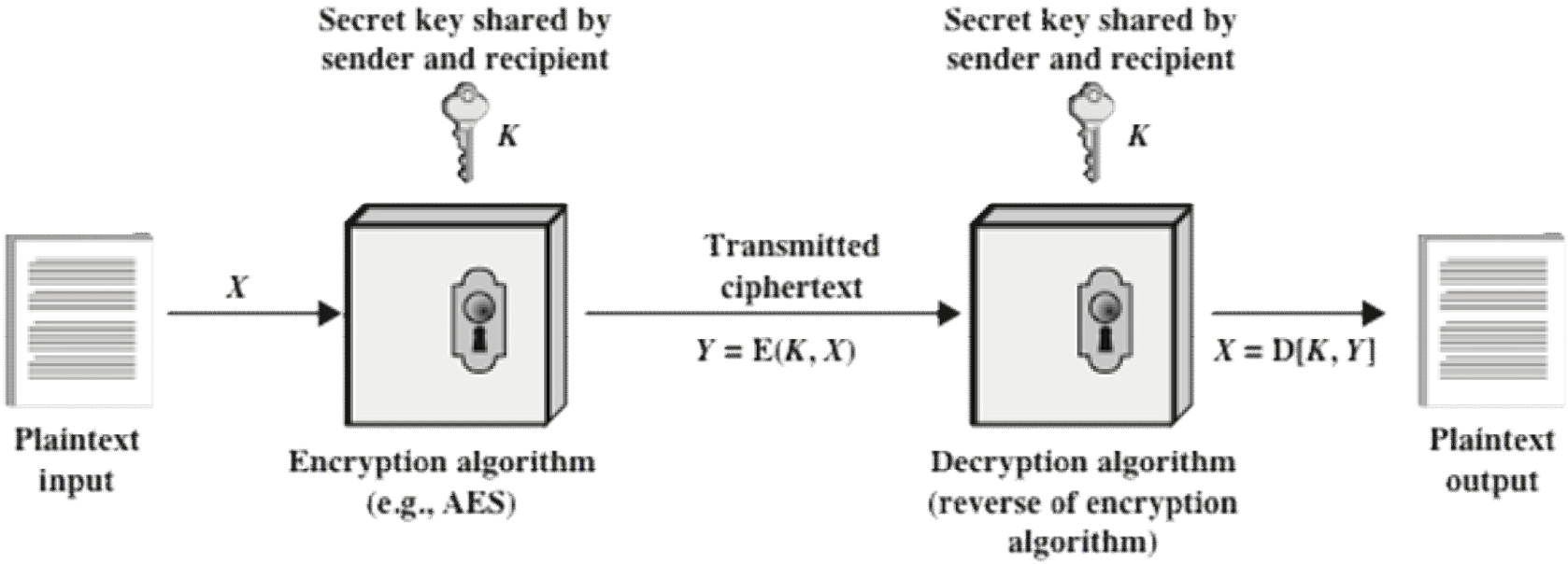


Figure 2.1 Simplified Model of Symmetric Encryption

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:

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

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

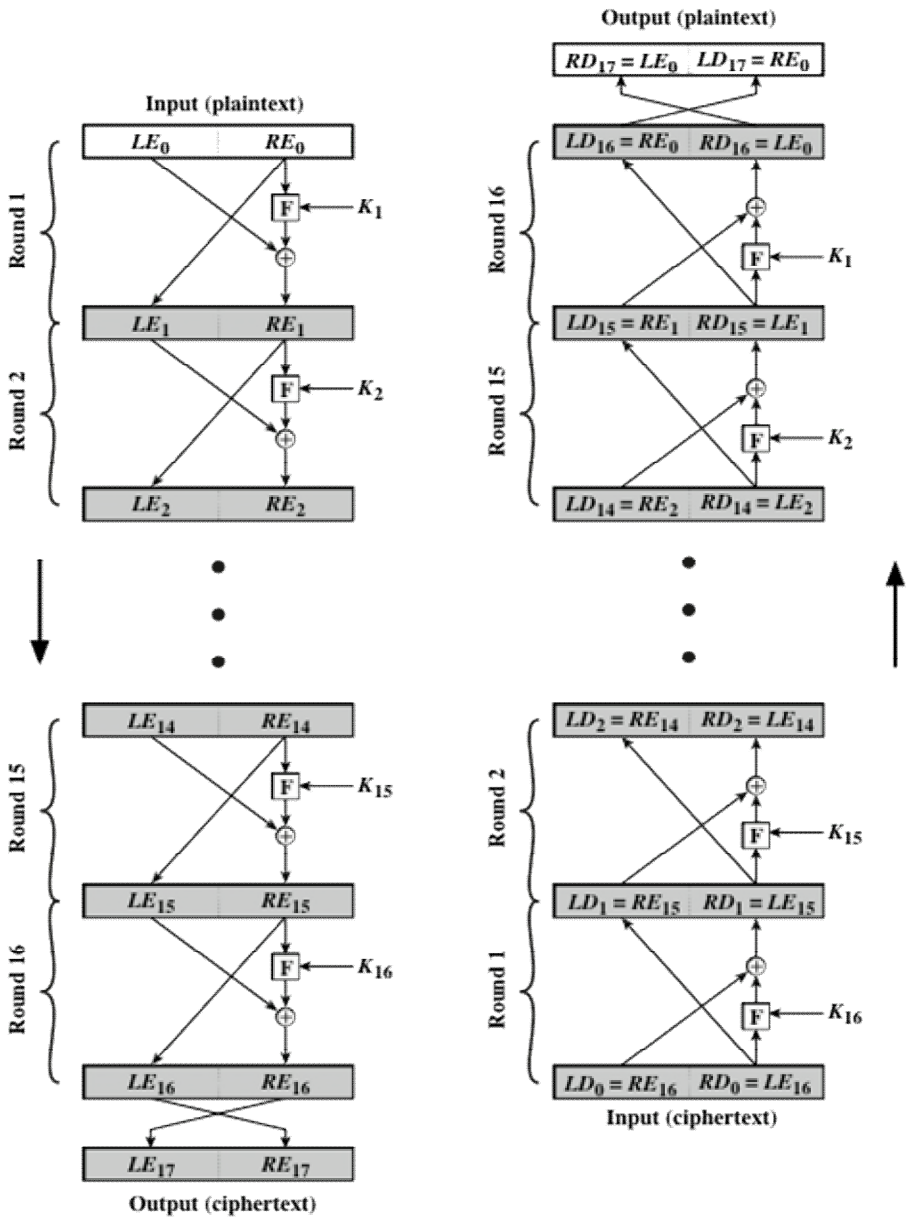


Figure 2.2 Feistel Encryption and Decryption (16 rounds)

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

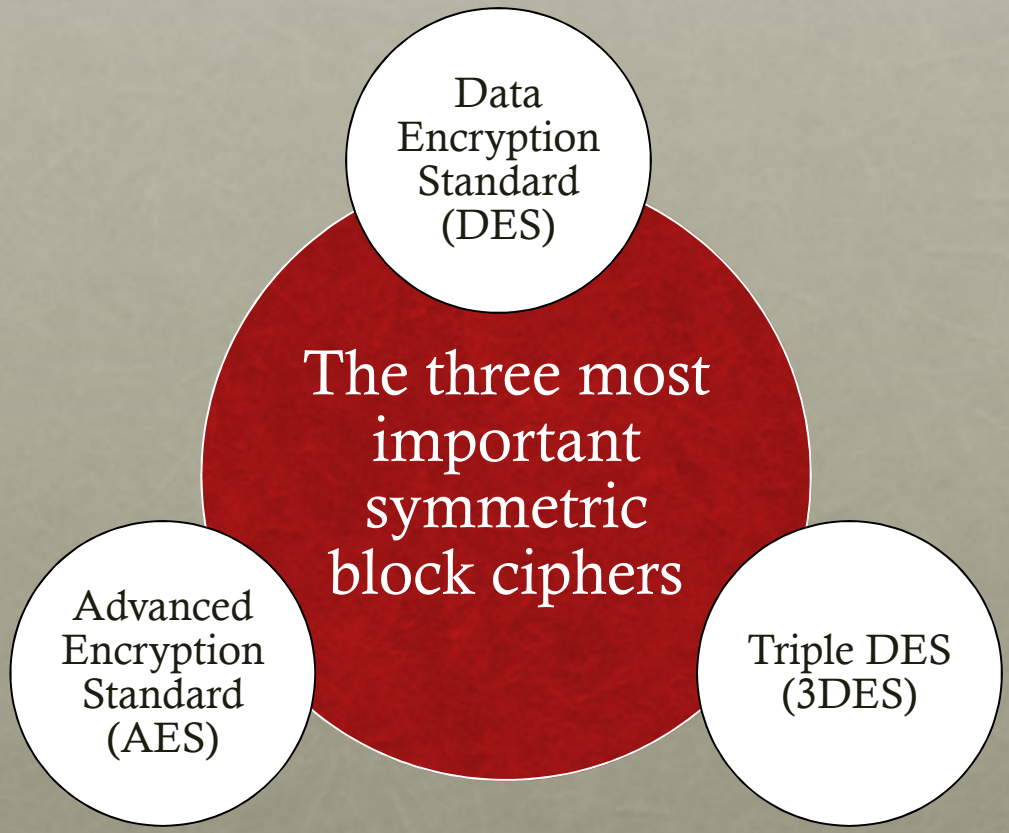
- In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the seed of execution of the algorithm becomes a concern

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

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)

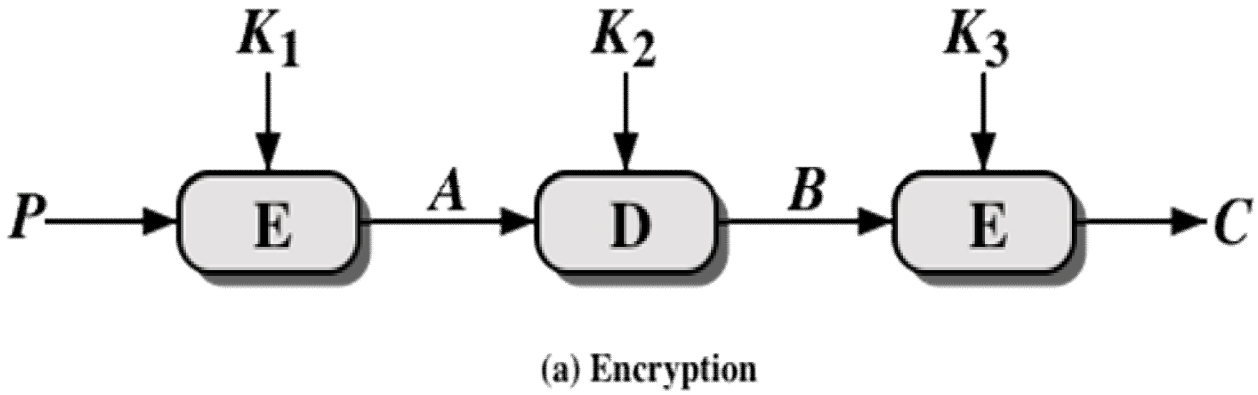
DES ALGORITHM

- 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

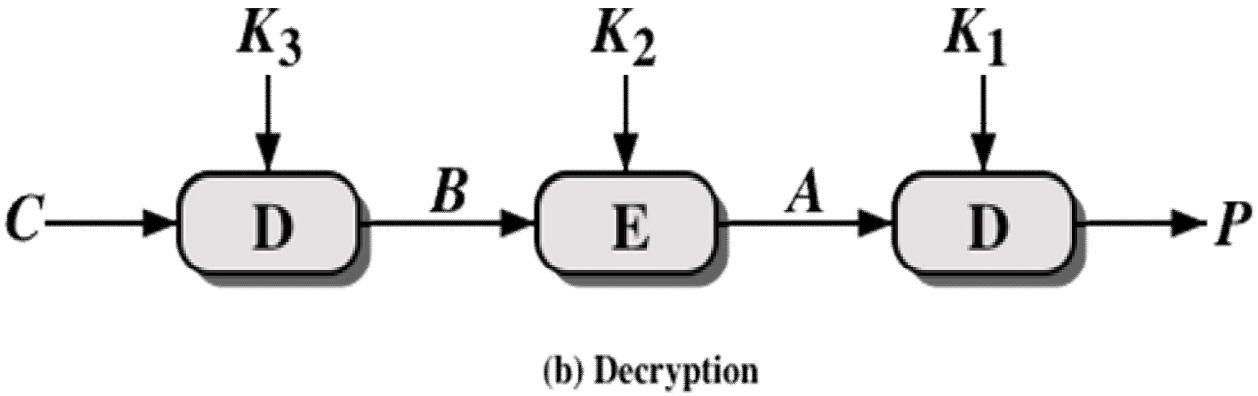
TABLE 2.2

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



(a) Encryption



(b) Decryption

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

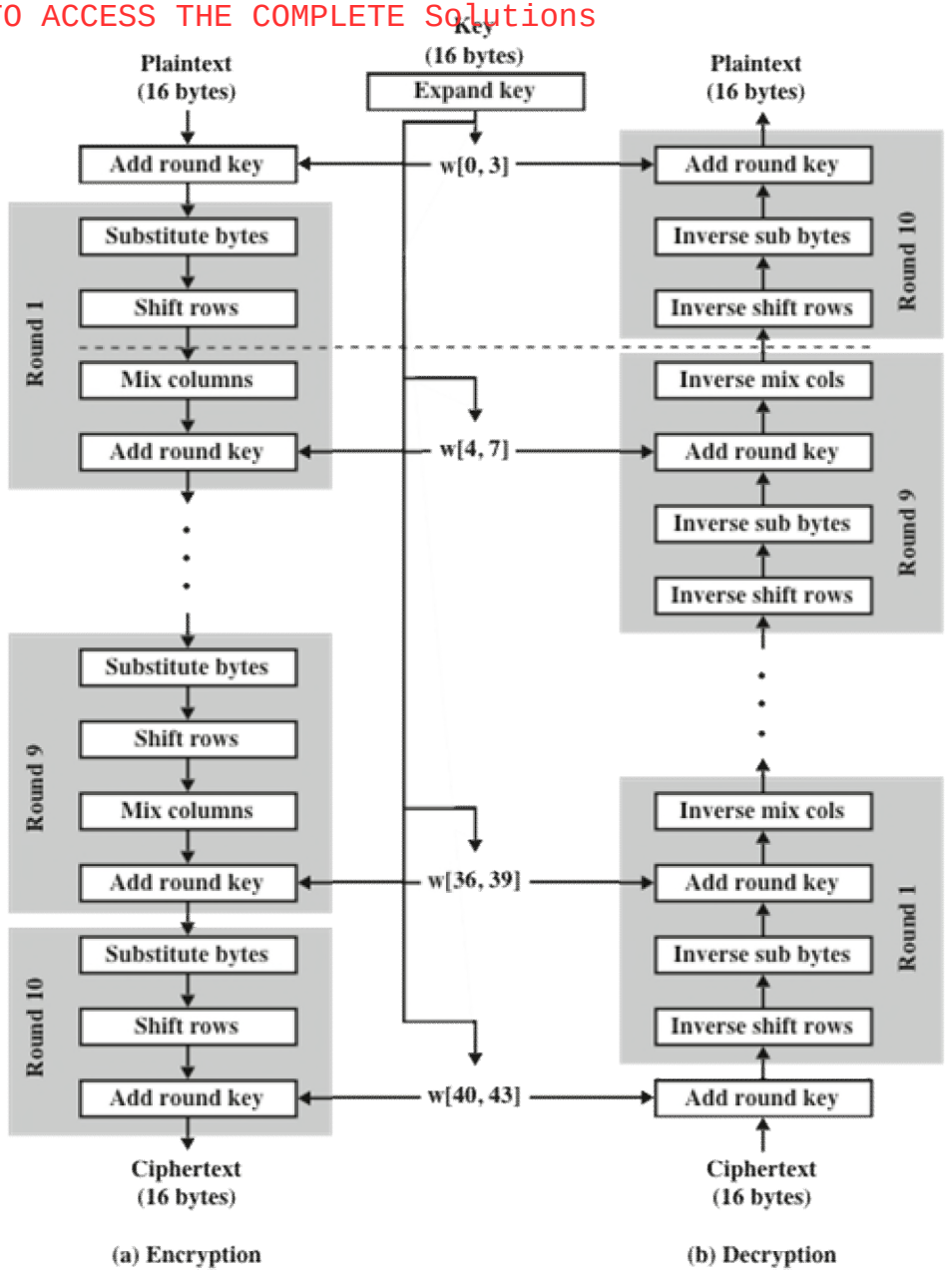


Figure 2.4 AES Encryption and Decryption

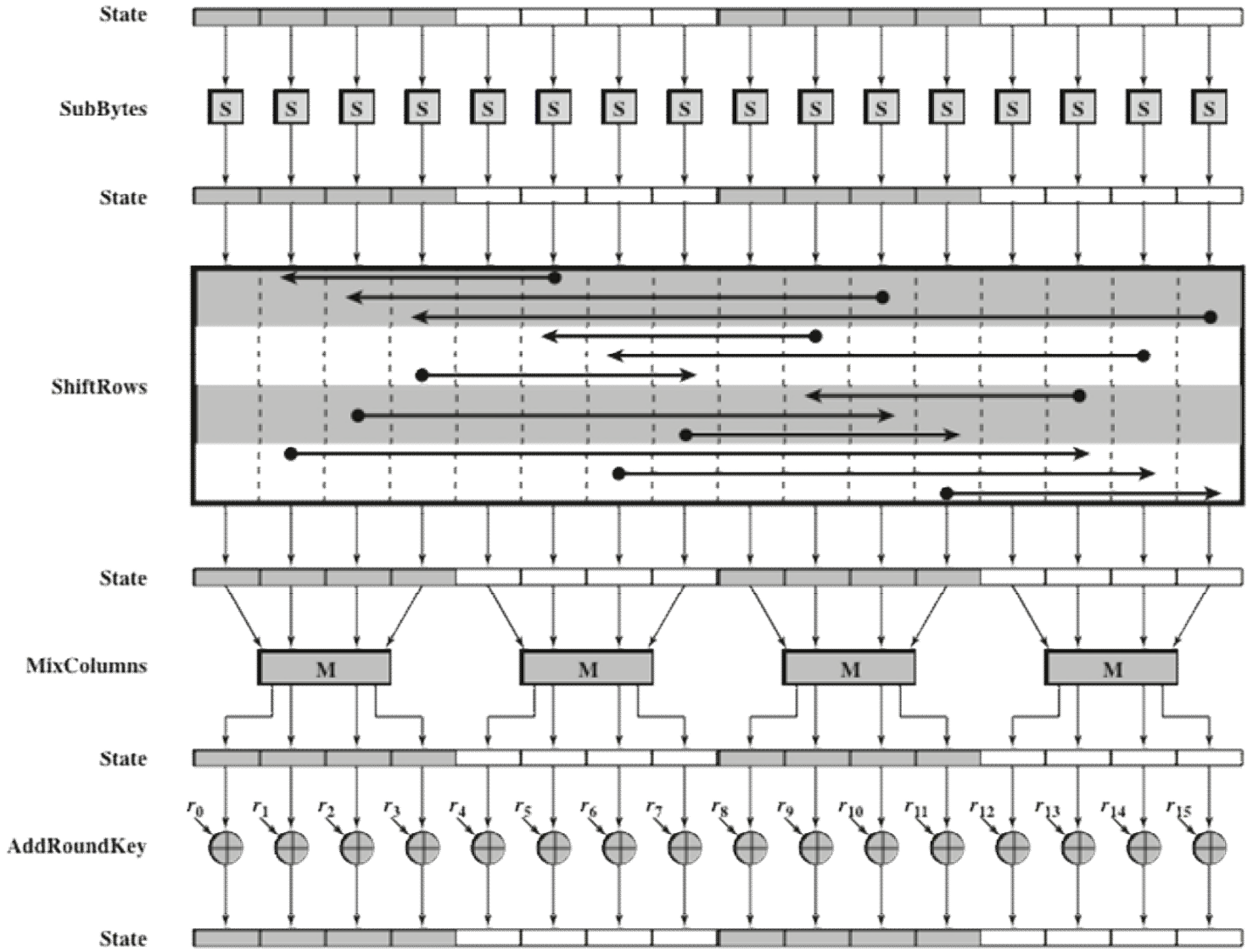


Figure 2.5 AES Encryption Round

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

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

Uniform distribution

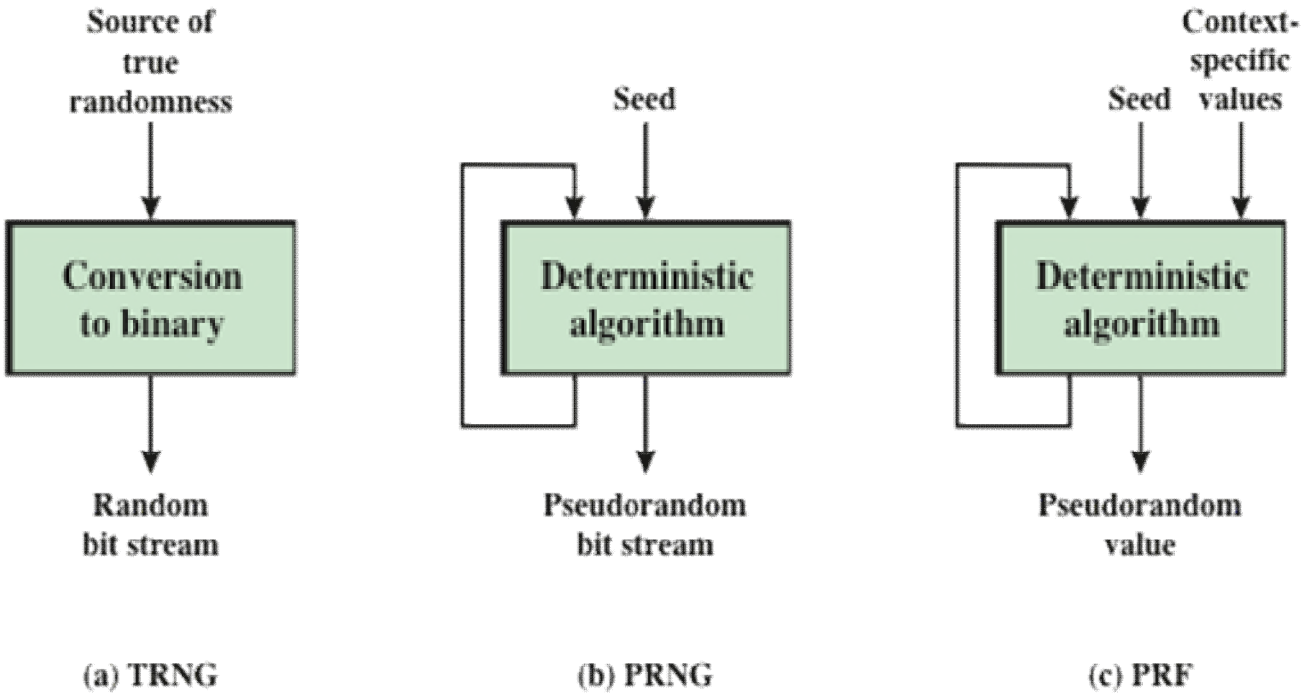
- The distribution of bits in the sequence should be uniform
- Frequency of occurrence of ones and zeros should be approximately the same

Independence

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



TRNG = true random number generator
PRNG = pseudorandom number generator
PRF = pseudorandom function

Figure 2.6 Random and Pseudorandom Number Generators

ALGORITHM DESIGN

Purpose-built algorithms	• Designed specifically and solely for the purpose of generating pseudorandom bit streams
Algorithms based on existing cryptographic algorithms	• Cryptographic 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

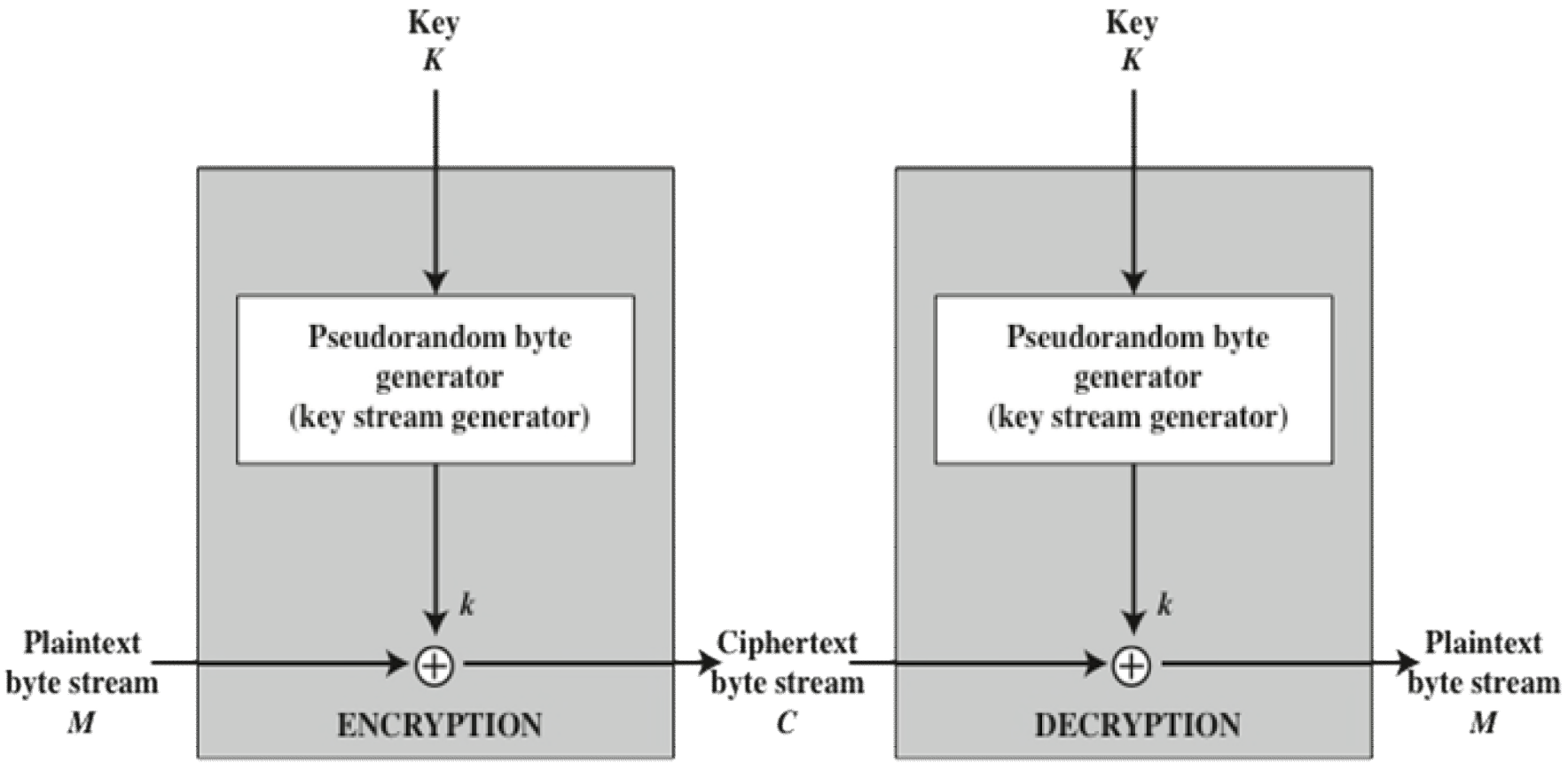


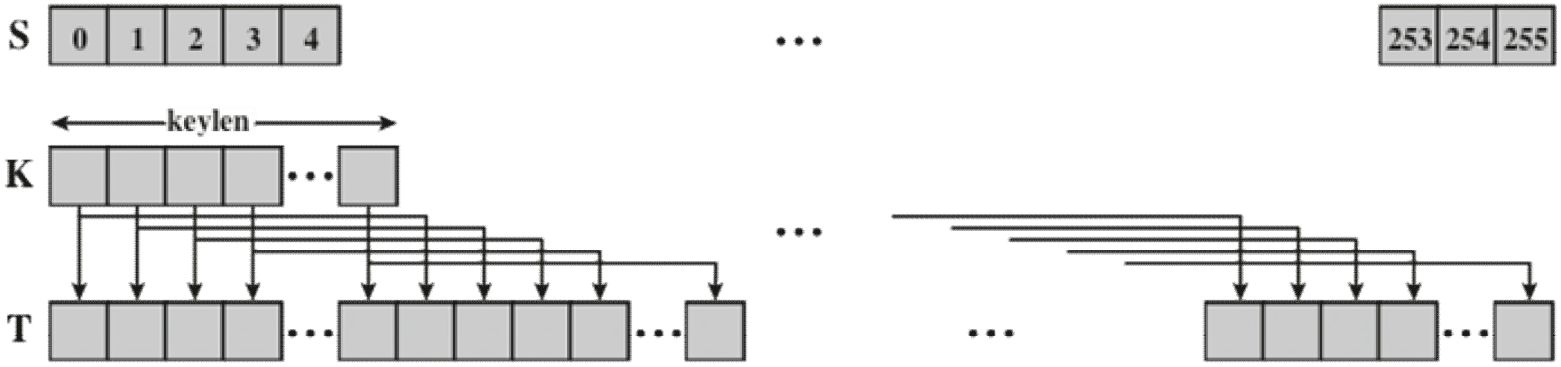
Figure 2.7 Stream Cipher Diagram

STREAM CIPHER DESIGN CONSIDERATIONS

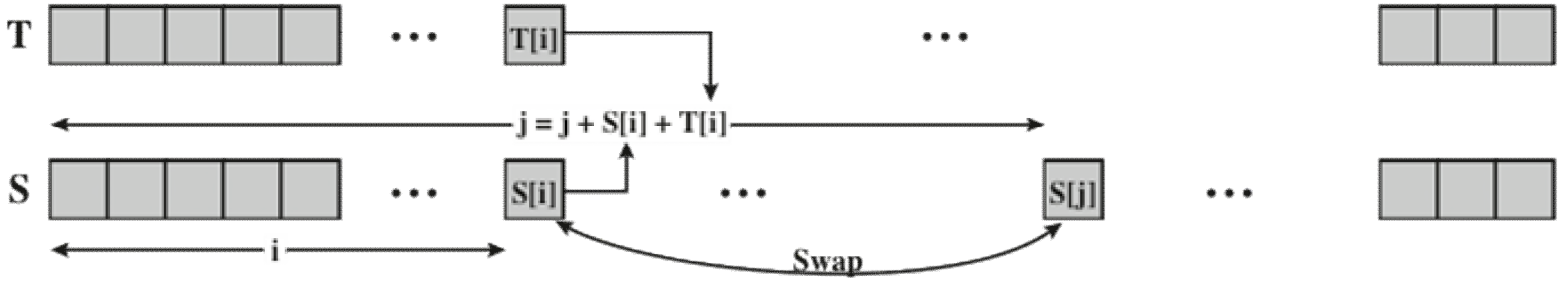
- The encryption sequence should have a large period
  - The longer the period of repeat, the more difficult it will be to do cryptanalysis
- The keystream should approximate the properties of a true random number stream as close as possible
  - The more random-appearing the keystream is, the more randomized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

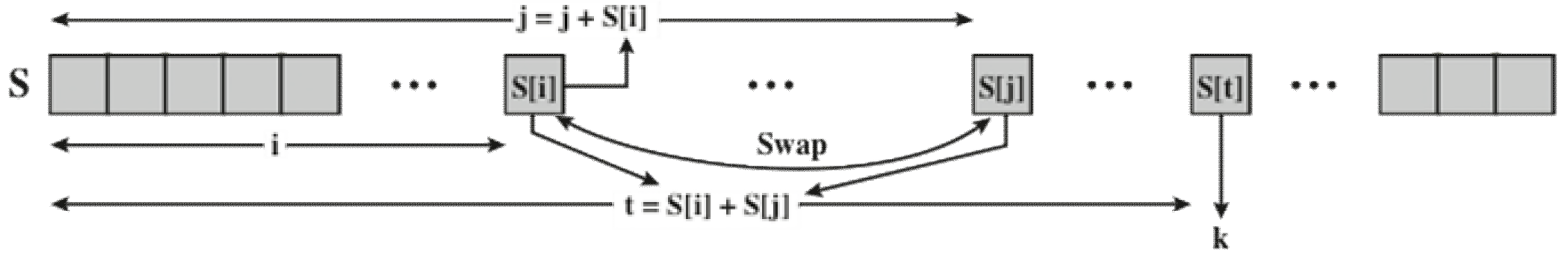
- A stream cipher designed in 1987 by Ron Rivest for RSA Security
- It is a variable key-size stream cipher with byte-oriented operations
- The algorithm is based on the use of a random permutation
- Is used in the Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards that have been defined 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



(a) Initial state of S and T



(b) Initial permutation of S



(c) Stream Generation

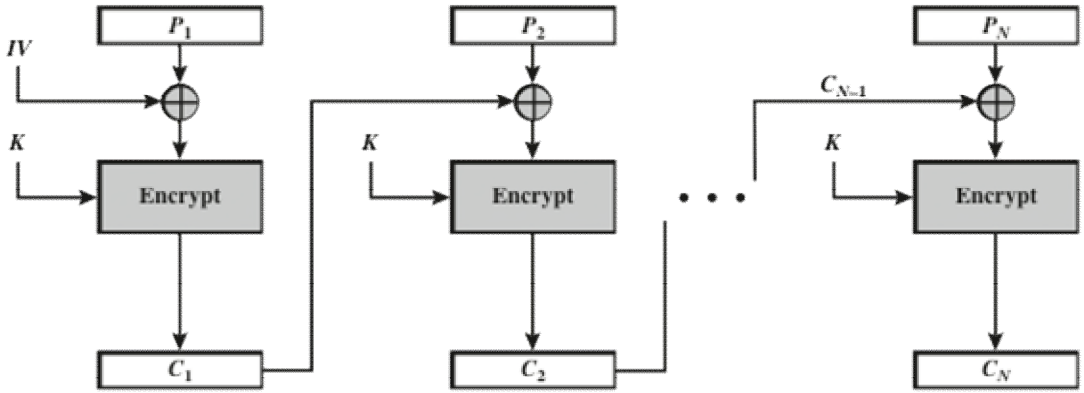
Figure 2.8 RC4

CIPHER BLOCK MODES OF OPERATION

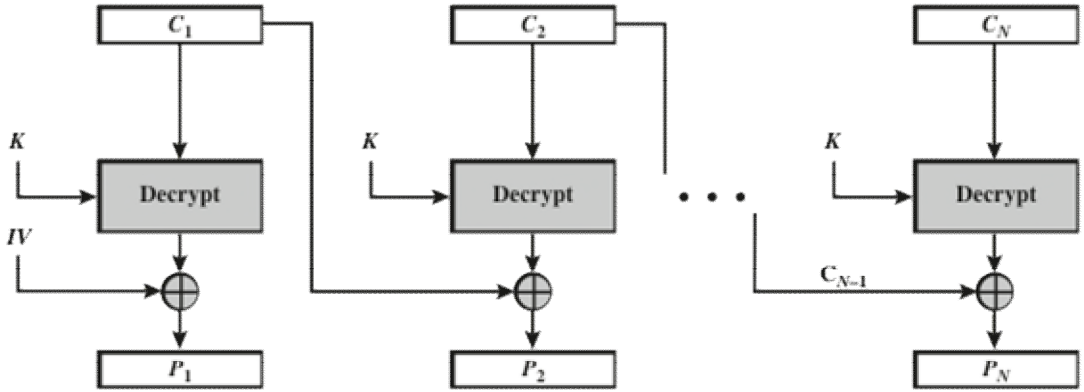
- A symmetric block cipher processes one block of data at a time
  - In the case of DES and 3DES, the block length is b=64 bits
  - For AES, the block length is b=128
  - For longer amounts of plaintext, it is necessary to break the plaintext into b-bit blocks, padding the last block if necessary
- Five 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

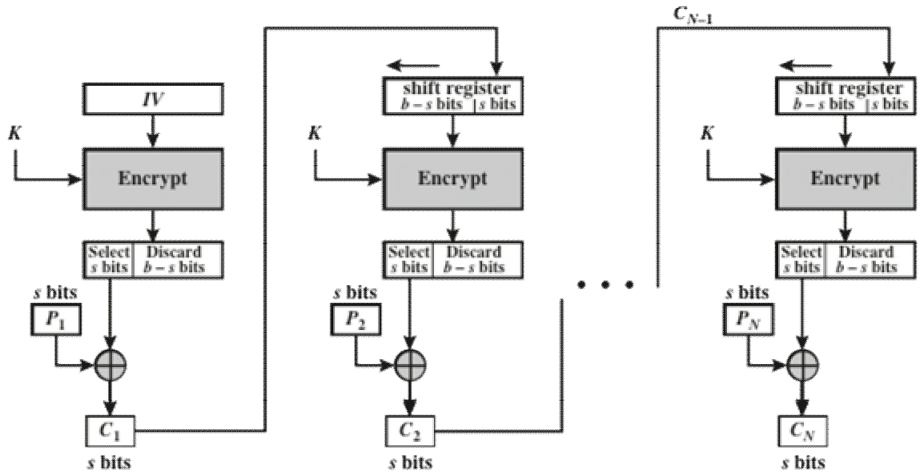


(a) Encryption

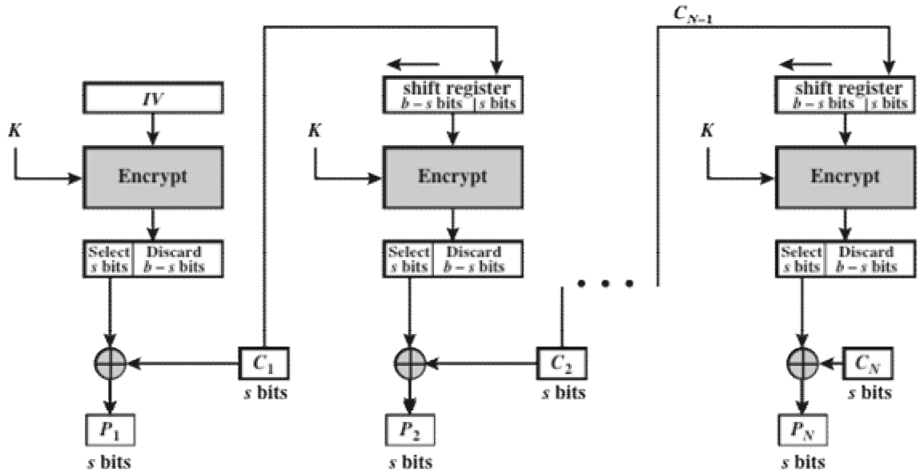


(b) Decryption

Figure 2.9 Cipher Block Chaining (CBC) Mode

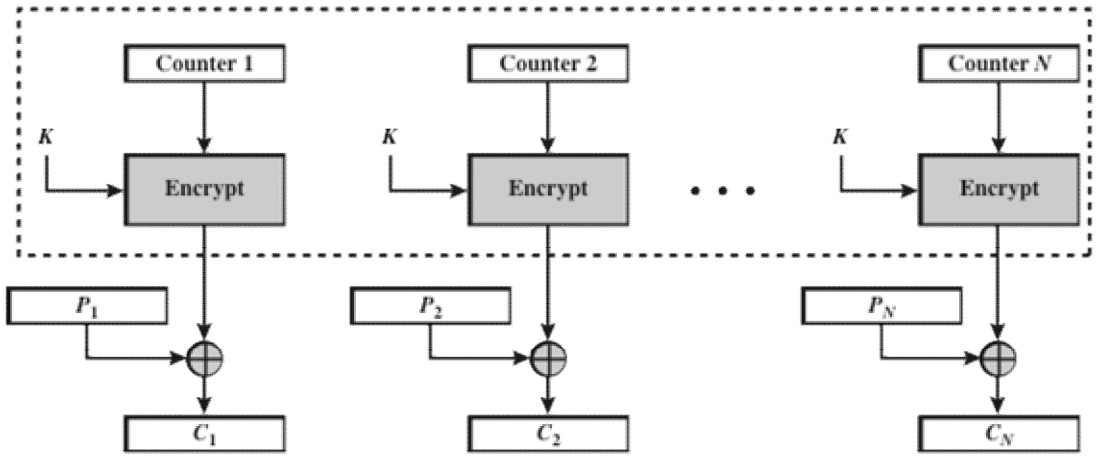


(a) Encryption

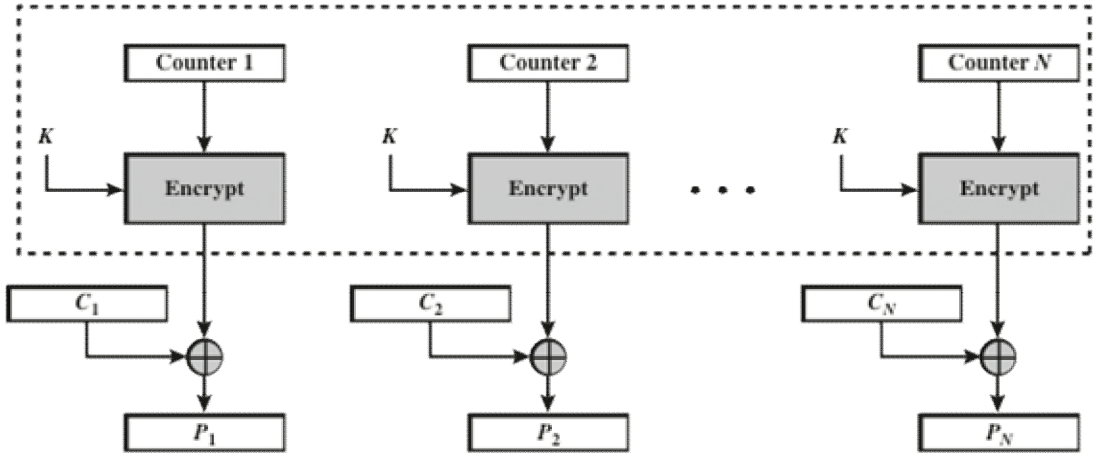


(b) Decryption

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



(a) Encryption



(b) Decryption

Figure 2.11 Counter (CTR) Mode

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 ith 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

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