

UNIT - I: INTRODUCTION: Security Attacks(Interruption, Interception, Modification and Fabrication), Security Services (Confidentiality, Authentication, Integrity, Non-repudiation, Access Control and Availability) and Mechanisms. A Model for Internetwork security, Conventional Encryption Principles, Conventional Encryption Algorithms (DES, Triple DES and AES), Cipher Block Modes of Operations (CBC, CFB only), Stream Ciphers and RC4, Location of Encryption Devices, Key Distribution.

TEXTBOOK William Stallings, Network Security Essentials (Applications and Standards), Pearson Education.

UNIT-I

- **Security attack**: Any action that compromises the security of information owned by an organization.
- **Security mechanism**: A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.
- **Security service**: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

SECURITY ATTACK

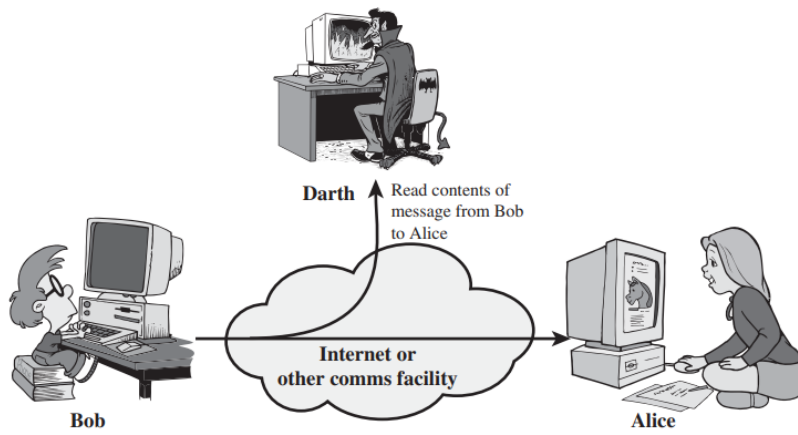
The following four are general categories of attack

- Interruption**: An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability. Examples include destruction of a piece of hardware, such as a hard disk, the cutting of a communication line, or the disabling of the file management system.
- Interception**: An unauthorized party gains access to an asset. This is an attack on confidentiality. The unauthorized party could be a person, a program, or a computer. Examples include wiretapping to capture data in a network, and the unauthorized copying of files or programs.
- Modification**: An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. Examples include changing values in a data files, altering a program so that it performs differently, and modifying the content of messages being transmitted in a network
- Fabrication**: An authorized party inserts counterfeit objects into the system. This is an attack on authenticity. Examples include the insertion of spurious messages in a network or the addition of records to a file.

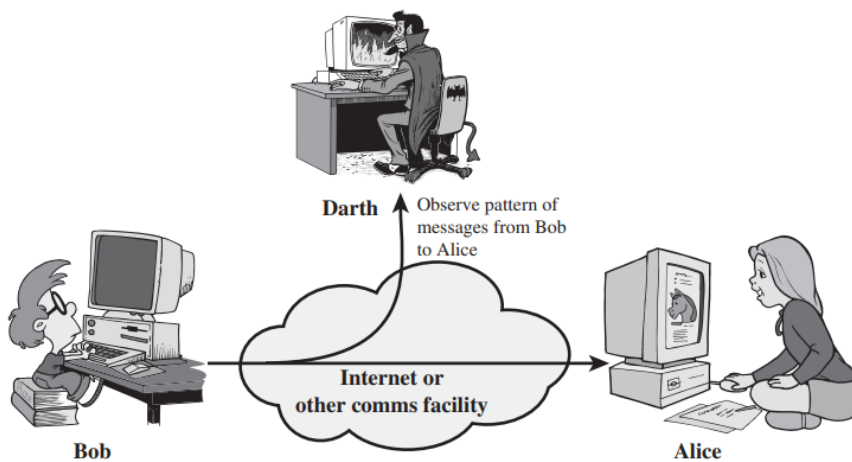
A useful means of classifying security attacks, is in terms of **passive attacks and active attacks**. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

Passive Attacks:

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.



(a) Release of message contents



(b) Traffic analysis

The (i) **release of message contents** is easily understood (Figure a). A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

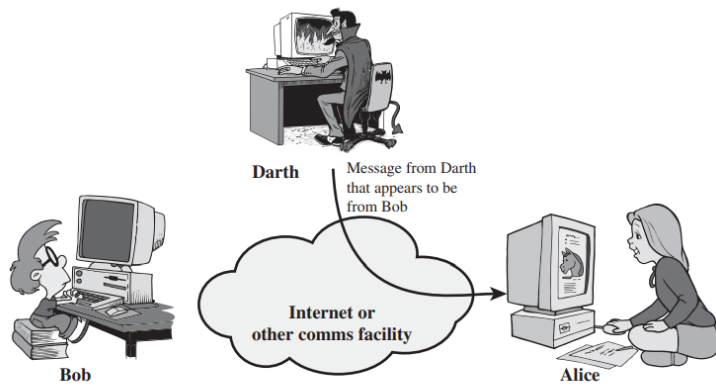
A second type of passive attack, (ii) **traffic analysis**, is subtler (Figure b). Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption.

Passive attacks are very difficult to detect, because they do not involve any alteration of the data.

Active attacks:

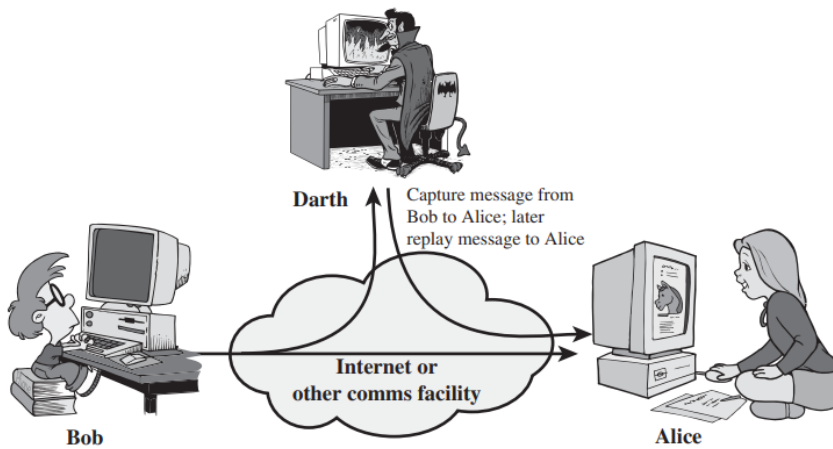
Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

A (i) **masquerade attack** is any attack that uses a forged identity (such as a network identity) to gain unofficial access to a personal or organisational computer. Masquerade attacks are generally performed by using either stolen passwords.



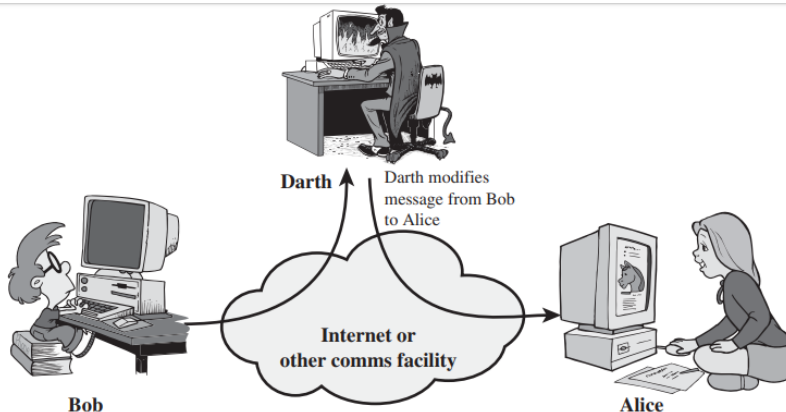
(a) Masquerade

(ii) **Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect



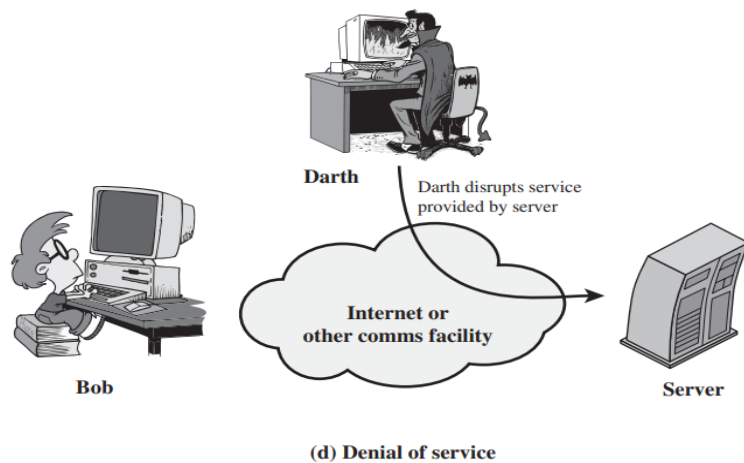
(b) Replay

(iii) **Modification of messages** simply means that some portion of a message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure 1.3c). For example, a message meaning “Allow John Smith to read confidential file accounts” is modified to mean “Allow Fred Brown to read confidential file accounts.”



(c) Modification of messages

(iv) The **denial of service** prevents or inhibits the normal use or management of communications facilities. In this entity may suppress all messages directed to a particular destination (e.g., the security audit service), or either by disabling the network or by overloading it with messages so as to degrade performance.



Comparison between Passive and Active attack

Active Attack	Passive Attack
(i) Active attacks are not difficult to detect	(i) Passive attacks are difficult to detect
(ii) It is quite difficult to prevent active attacks absolutely because of the wide variety of potential physical, software, and network vulnerabilities	(ii) Measures are available to prevent their success

SECURITY SERVICES:

One useful classification of security services is the following

- i. Data Confidentiality
- ii. Authentication
- iii. Integrity
- iv. Nonrepudiation
- v. Access Control
- vi. Availability

i) **Data Confidentiality:** Confidentiality is the protection of transmitted data from passive attacks. With respect to the content of a data transmission, several levels of protection can be identified. The broadest service protects all user data transmitted between two users over a period of time.

The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility.

(ii) **Authentication:** The authentication service is concerned with assuring that a communication is authentic.

In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from.

In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. First, at the time of connection initiation, the service assures that the two entities are authentic (that is, that each is the entity that it claims to be). Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade.

(iii) **Data Integrity:** As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. Again, the most useful and straightforward approach is total stream protection.

A connection-oriented integrity service deals with a stream of messages and assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays.

On the other hand, a connectionless integrity service deals with individual messages without regard to any larger context and generally provides protection against message modification only.

(iv) **Nonrepudiation:** This prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the authorized sender had sent the message. Similarly, when a message is received, the sender can prove that the authorized receiver in fact received the message.

(v) **Access Control:** In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity must be provided with access rights.

(vi) **Availability Service :** This define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorized system entity. This service addresses the security concerns raised by denial-of-service attacks.

Table 1.2 Security Services (X.800)

<p style="text-align: center;">AUTHENTICATION</p> <p>The assurance that the communicating entity is the one that it claims to be.</p> <p>Peer Entity Authentication Used in association with a logical connection to provide confidence in the identity of the entities connected.</p> <p>Data-Origin Authentication In a connectionless transfer, provides assurance that the source of received data is as claimed.</p> <p style="text-align: center;">ACCESS CONTROL</p> <p>The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).</p> <p style="text-align: center;">DATA CONFIDENTIALITY</p> <p>The protection of data from unauthorized disclosure.</p> <p>Connection Confidentiality The protection of all user data on a connection.</p> <p>Connectionless Confidentiality The protection of all user data in a single data block.</p> <p>Selective-Field Confidentiality The confidentiality of selected fields within the user data on a connection or in a single data block.</p> <p>Traffic-Flow Confidentiality The protection of the information that might be derived from observation of traffic flows.</p>	<p style="text-align: center;">DATA INTEGRITY</p> <p>The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).</p> <p>Connection Integrity with Recovery Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.</p> <p>Connection Integrity without Recovery As above, but provides only detection without recovery.</p> <p>Selective-Field Connection Integrity Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.</p> <p>Connectionless Integrity Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.</p> <p>Selective-Field Connectionless Integrity Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.</p> <p style="text-align: center;">NONREPUDIATION</p> <p>Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.</p> <p>Nonrepudiation, Origin Proof that the message was sent by the specified party.</p> <p>Nonrepudiation, Destination Proof that the message was received by the specified party.</p>
--	--

SECURITY MECHANISMS

The below Table lists the security mechanisms defined in X.800.

SPECIFIC SECURITY MECHANISMS	PERVASIVE SECURITY MECHANISMS
<p>May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.</p> <p>Encipherment The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.</p> <p>Digital Signature Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).</p> <p>Access Control A variety of mechanisms that enforce access rights to resources.</p>	<p>Mechanisms that are not specific to any particular OSI security service or protocol layer.</p> <p>Trusted Functionality That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).</p> <p>Security Label The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.</p> <p>Event Detection Detection of security-relevant events.</p> <p>Security Audit Trail Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.</p> <p>Security Recovery</p>
<p>Data Integrity A variety of mechanisms used to assure the integrity of a data unit or stream of data units.</p> <p>Authentication Exchange A mechanism intended to ensure the identity of an entity by means of information exchange.</p> <p>Traffic Padding The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.</p> <p>Routing Control Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.</p> <p>Notarization The use of a trusted third party to assure certain properties of a data exchange.</p>	<p>Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.</p>

Table 1.4 Relationship Between Security Services and Mechanisms

Service	Mechanism							
	Encipherment	Digital Signature	Access Control	Data Integrity	Authentication Exchange	Traffic Padding	Routing Control	Notarization
Peer Entity Authentication	Y	Y			Y			
Data-Origin Authentication	Y	Y						
Access Control			Y					
Confidentiality	Y						Y	
Traffic-Flow Confidentiality	Y					Y	Y	
Data Integrity	Y	Y		Y				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

A MODEL FOR NETWORK SECURITY

A model for much of what we will be discussing is captured, in very general terms. A message is to be transferred from one party to another across some sort of Internet service. The two parties, who are the principals in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP).

All of the techniques for providing security have two components:

1. A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.
2. Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

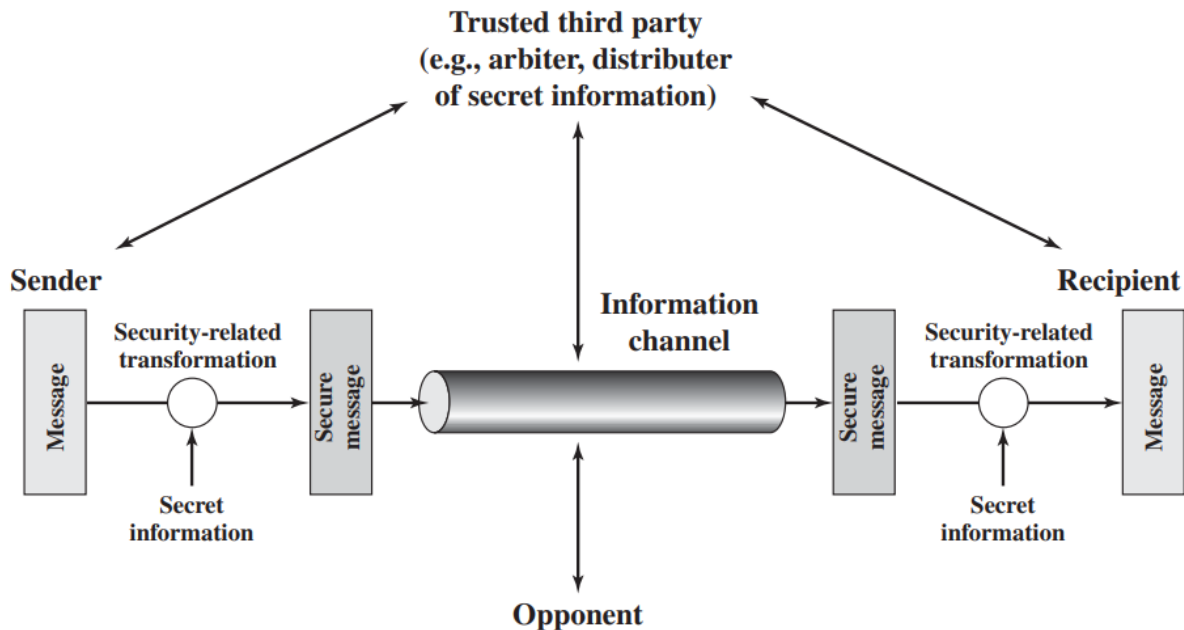


Figure 1.4 Model for Network Security

A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information between the sender and receiver while keeping it away from any opponent.

This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

A general model of these other situations is illustrated by below Figure, which reflects a concern for protecting an information system from unwanted access. Most readers are familiar with the concerns caused by the existence of hackers who attempt to penetrate systems that can be accessed over a network. The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system. The intruder can be a disgruntled employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).

Programs can present two kinds of threats:

1. **Information access threats:** Intercept or modify data on behalf of users who should not have access to that data.
2. **Service threats:** Exploit service flaws in computers to inhibit use by legitimate users.

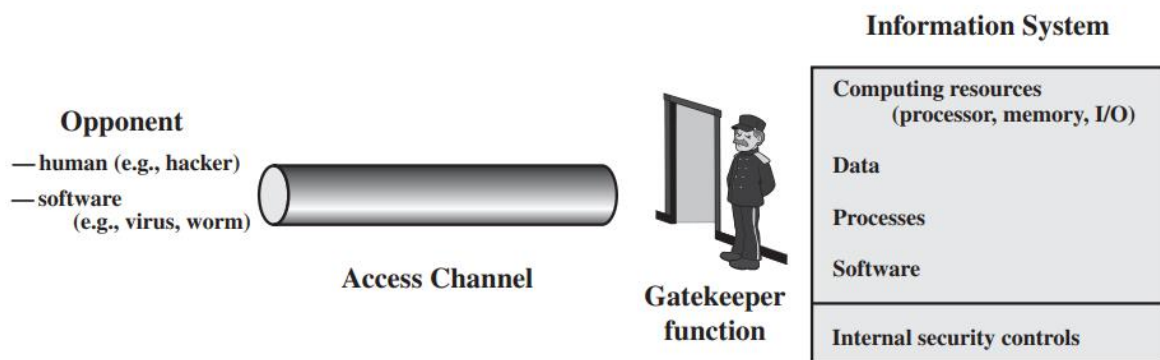


Figure 1.5 Network Access Security Model

Viruses and worms are two examples of software attacks. Such attacks can be introduced into a system by means of a disk that contains the unwanted logic concealed in otherwise useful software. They also can be inserted into a system across a network.

The security mechanisms needed to cope with unwanted access fall into two broad categories

- (i) The first category might be termed a gatekeeper function. It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses, and other similar attacks.
- (ii) The second line of defence consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

Possible questions:

1. What is the OSI security architecture?
2. What is the difference between passive and active security threats?
3. List and briefly define categories of passive and active security attacks.
4. List and briefly define categories of security services.
5. List and briefly define categories of security mechanisms.

SYMMETRIC ENCRYPTION PRINCIPLES/CONVENTIONAL ENCRYPTION PRINCIPLES

A symmetric encryption scheme has five ingredients

- **Plaintext:** This is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the same secret key and produces the original plaintext.

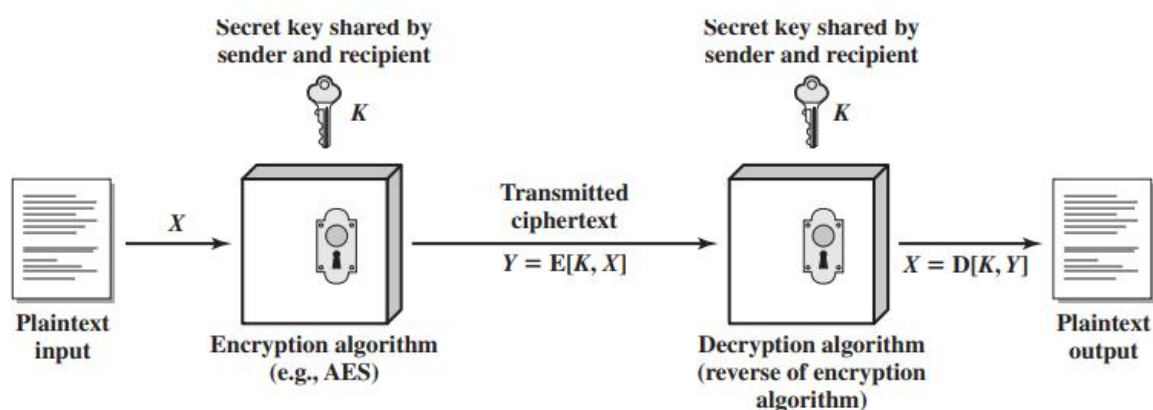


Figure 2.1 Simplified Model of Symmetric Encryption

There are two requirements for secure use of symmetric encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key.
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

It is important to note that the security of symmetric encryption depends on the secrecy of the key, not the secrecy of the algorithm. That is, it is assumed that it is impractical to decrypt a message on the basis of the ciphertext plus knowledge of the encryption/decryption algorithm. In other words, we do not need to keep the algorithm secret; we need to keep only the key secret.

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.

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.

Cryptanalysis:

The process of attempting to discover the plaintext or key is known as cryptanalysis. The strategy used by the cryptanalyst depends on the nature of the encryption scheme and the information available to the cryptanalyst.

One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plaintext that is concealed, such as English or French text, an EXE file, a Java source listing, an accounting file, and so on.

- The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of information to work with.
- Closely related to the known-plaintext attack is what might be referred to as a probable-word attack. If the opponent is working with the encryption of some general prose message, he or she may have little knowledge of what is in the message.
- if the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key, then it is called chosen plaintext attack

The below Table2.1 summarizes the various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst.

Table 2.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext only	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext to be decoded
Known plaintext	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext to be decoded • One or more plaintext–ciphertext pairs formed with the secret key
Chosen plaintext	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext to be decoded • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	<ul style="list-style-type: none"> • 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

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.

The below 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.

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

Table 2.2 Average Time Required for Exhaustive Key Search

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

FEISTEL CIPHER STRUCTURE

The inputs to the encryption algorithm are a plaintext block of length $2w$ bits and a key K . The plaintext block is divided into two halves, LE_0 and RE_0 . The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block. Each

round i has as inputs LE_{i-1} and RE_{i-1} derived from the previous round, as well as a subkey K_i derived from the overall K . In general, the subkeys K_i are different from K and from each other and are generated from the key by a subkey generation algorithm.

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 K_i . Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data

Decryption with a symmetric block cipher is essentially the same as the encryption process. The rule is as follows: Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order. That is, use K_n in the first round, K_{n-1} in the second round, and so on until K_1 is used in the last round. This is a nice feature, because it means we need not implement two different algorithms—one for encryption and one for decryption.

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 tradeoff 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. 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.

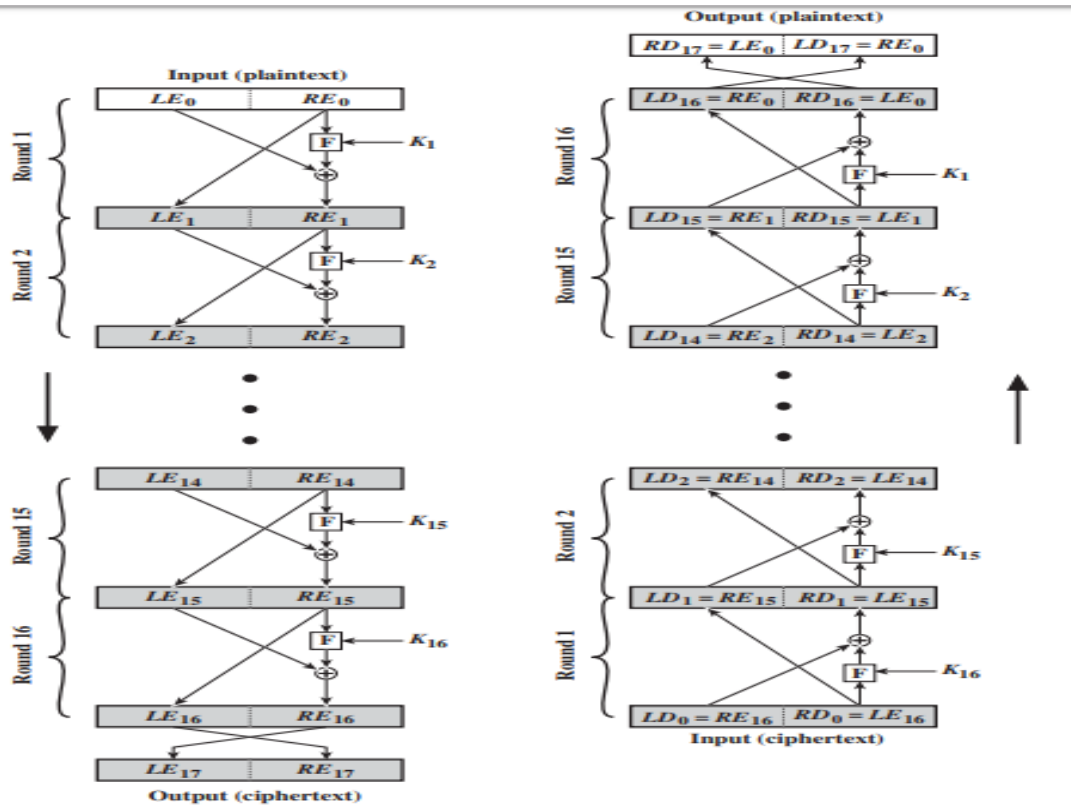


Figure 2.2 Feistel Encryption and Decryption (16 rounds)

SYMMETRIC BLOCK ENCRYPTION ALGORITHMS/ CONVENTIONAL ENCRYPTION ALGORITHM

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.

The three types of block cipher algorithms are:

- Data Encryption Standard (DES)
- Triple DES (3DES)
- Advanced Encryption Standard (AES)

DATA ENCRYPTION STANDARD

The most widely used encryption scheme is based on the **Data Encryption Standard (DES)** issued in 1977. The algorithm itself is referred to as the Data Encryption Algorithm (DEA)

Description of algorithm: The plaintext is 64 bits in length and the key is 56 bits in length longer plaintext amounts are processed in 64-bit blocks. 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 K_i in reverse order. That is, use K_{16} on the first iteration, K_{15} on the second iteration, and so on until K_1 is used on the 16th and last iteration.

THE STRENGTH OF DES:

1. 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 and exploit 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.

2. A more serious concern is key length. With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2×10^{16} keys. Thus, on the face of it, a bruteforce attack appears impractical.

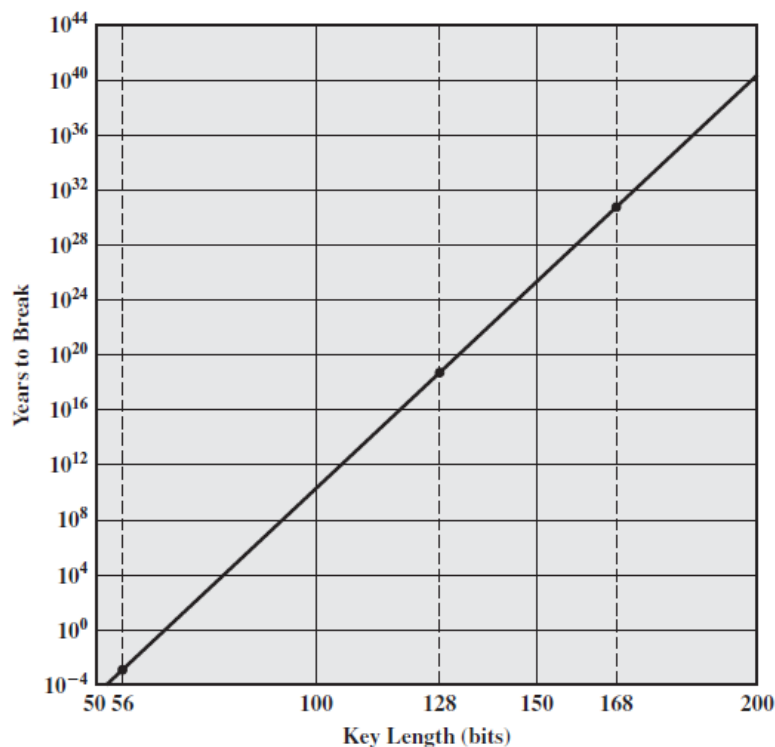
Weakness/Disadvantage of DES:

DES finally and definitively proved insecure in July 1998, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose “DES cracker” machine that was built for less than \$250,000. The attack took less than three days. The EFF has published a detailed description of the machine, enabling others to build their own cracker [EFF98]. And, of course, hardware prices will continue to drop as speeds increase, making DES virtually worthless.

If the only form of attack that could be made on an encryption algorithm is brute force, then the way to counter such attacks is obvious use longer keys

For a 128-bit key, which is common among contemporary algorithms, it would take over 10^{18} years to break the code using the EFF cracker. Even if we managed to speed up the cracker by a factor of 1 trillion (10^{12}), it would still take over 1 million years to break the code. So a 128-bit key is guaranteed to result in an algorithm that is unbreakable by brute force.

The below Figure shows how long it would take to crack a DES-style algorithm as a function of key size



Triple DES

Triple DES (3DES) was first standardized for use in financial applications in ANSI standard X9.17 in 1985. 3DES was incorporated as part of the Data Encryption Standard in 1999 with the publication of FIPS 46-3.

The below figure shows the working of triple DES

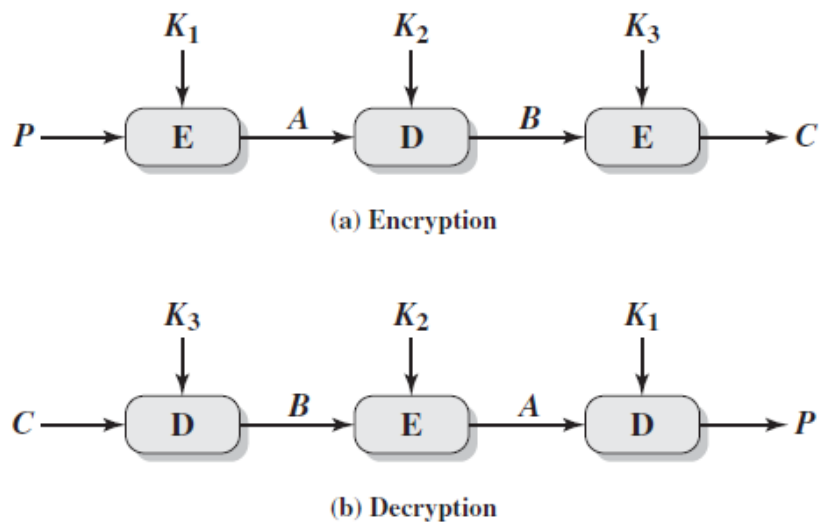


Figure 2.4 Triple DES

3DES uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence,

$$C = E(K_3, D(K_2, E(K_1, P)))$$

where

C ciphertext

P plaintext

$E[K, X]$ encryption of X using key K

$D[K, Y]$ decryption of Y using key K

Decryption is simply the same operation with the keys reversed

$$P = D(K_1, E(K_2, D(K_3, C)))$$

With three distinct keys, 3DES has an effective key length of 168 bits. Furthermore, with a 168-bit key length, brute-force attacks are effectively impossible.

Advantages/Strength of Triple DES:

1. 3DES has two attractions that assure its widespread use over the next few years. First, with its 168-bit key length, it overcomes the vulnerability to brute-force attack.
2. Second, the underlying encryption algorithm in 3DES is the same as in DEA. This algorithm has been subjected to more scrutiny than any other encryption algorithm over a longer period of time, and no effective cryptanalytic attack based on the algorithm rather than brute force has been found.

Disadvantages/Weakness of triple DES:

1. The principal drawback of 3DES is that the algorithm is relatively slow to respond in software. The original DEA was designed for mid-1970s hardware implementation and does not produce efficient software code. 3DES, which has three times as many rounds as DEA, is correspondingly slower.
2. A secondary drawback is that both DEA and 3DES use a 64-bit block size. For reasons of both efficiency and security, a larger block size is desirable

Advanced Encryption Standard

Advanced Encryption Standard (AES), which should have a security strength equal to or better than 3DES and significantly improved efficiency.

In addition to these general requirements, NIST specified that AES 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.

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 2001. NIST selected Rijndael as the proposed AES algorithm. The two researchers who developed and submitted Rijndael for the AES are both cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen.

OVERVIEW OF THE ALGORITHM: AES uses a block length of 128 bits and a key length that can be 128, 192, or 256 bits. In the description of this section, we assume a key length of 128 bits.

The following comments give some insight into AES.

1. One noteworthy feature of this structure is that it is not a Feistel structure. Recall that in the classic Feistel structure, half of the data block is used to modify the other half of the data block, and then the halves are swapped. AES does not use a Feistel structure but processes the entire data block in parallel during each round using substitutions and permutation
2. The key that is provided as input is expanded into an array of forty-four 32-bit words $w[i]$. Four distinct words (128 bits) serve as a round key for each round.
3. Four different stages are used, one of permutation and three of substitution:
 - **Substitute bytes:** The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns
 - **Shift rows:** Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows –

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.
 - **Mix columns:** Each column of four bytes is now transformed using a special mathematical function.
 - **Add round key:** A simple bitwise XOR of the current block with a portion of the expanded key.
4. The structure is quite simple. For both encryption and decryption, the cipher begins with an Add Round Key stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages.
5. Only the Add Round Key stage makes use of the key. For this reason, the cipher begins and ends with an Add Round Key stage.
6. The final round of both encryption and decryption consists of only three stages

Disadvantage/Weakness of the AES

1. The Add Round Key stage by itself would not be formidable. The other three stages together scramble the bits, but by themselves, they would provide no security because they do not use the key.
2. Each stage is easily reversible. For the Substitute Byte, Shift Row, and Mix Columns stages, an inverse function is used in the decryption algorithm. For the Add Round Key stage, the inverse is achieved by XORing the same round key to the block

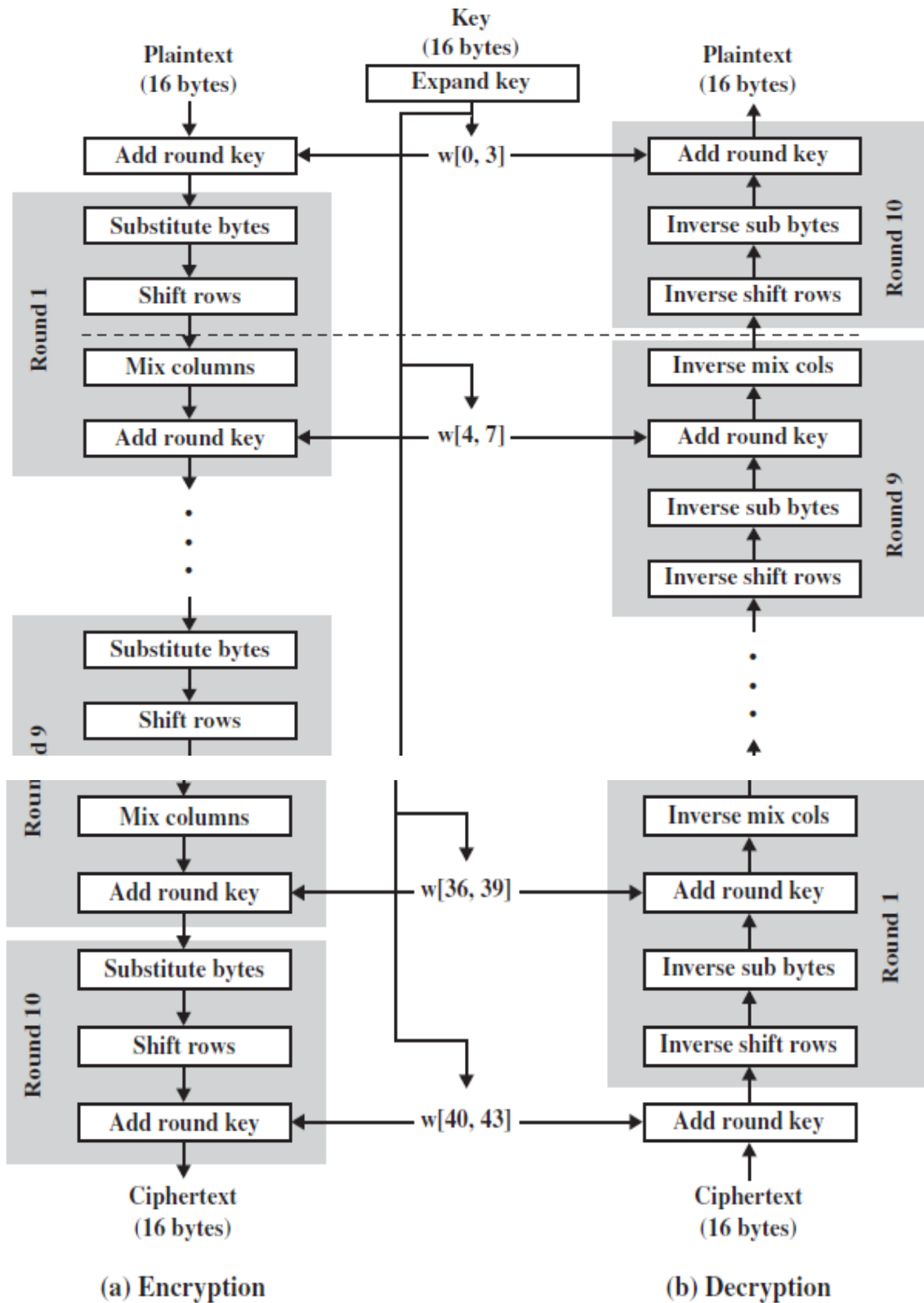


Figure 2.5 AES Encryption and Decryption

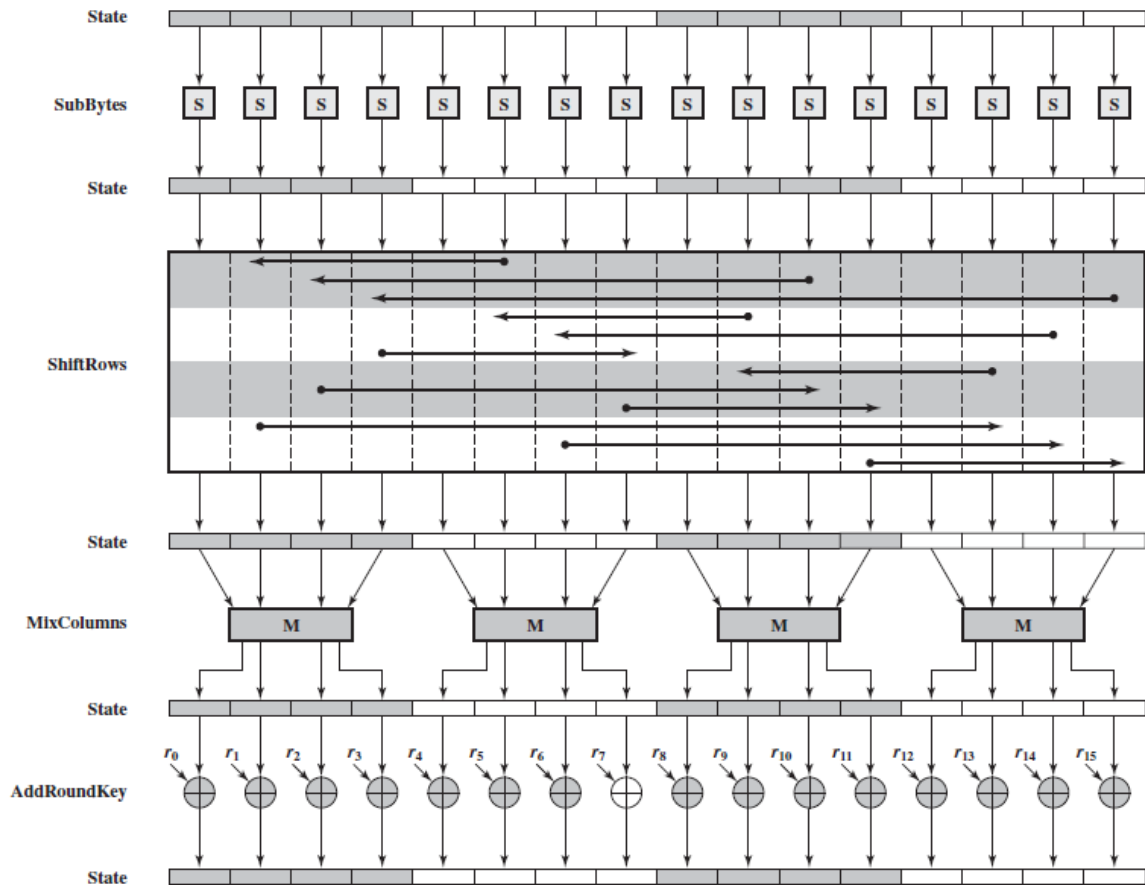


Figure 2.6 AES Encryption Round

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.

Stream Cipher Structure:

In this structure, a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random. A pseudorandom stream is one that is unpredictable without knowledge of the input key and which has an apparently random character. The output of the generator, called a **keystream**, is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation

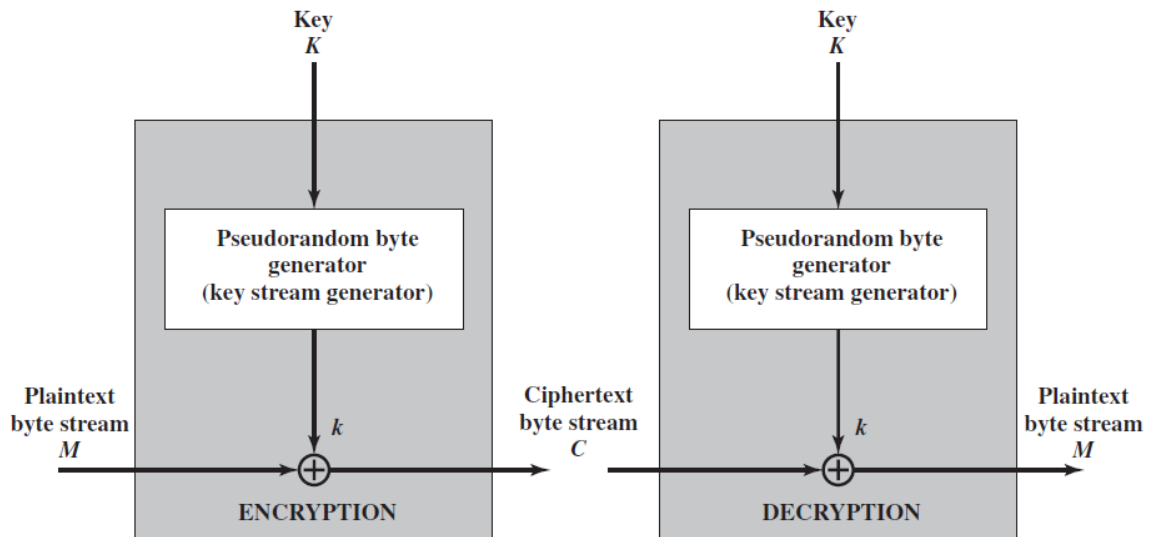


Figure 2.8 Stream Cipher Diagram

$$\begin{array}{r}
 11001100 \text{ plaintext} \\
 \oplus 01101100 \text{ key stream} \\
 \hline
 10100000 \text{ ciphertext}
 \end{array}$$

Decryption requires the use of the same pseudorandom sequence:

$$\begin{array}{r}
 10100000 \text{ ciphertext} \\
 \oplus 01101100 \text{ key stream} \\
 \hline
 11001100 \text{ plaintext}
 \end{array}$$

[KUMA97] lists the following important design considerations for a stream cipher.

1. A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats. The longer the period of repeat, the more difficult it will be to do cryptanalysis.
2. The more random-appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult.
3. Note from Figure 2.8 that the output of 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. The same considerations as apply for block ciphers are valid here. Thus, with current technology, a key length of at least 128 bits is desirable.

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 bits.

Cipher Block Chaining Mode

In the **cipher block chaining (CBC) mode** (Figure 2.10), the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; the same key is used for each block. In effect, we have chained together the processing of the sequence of plaintext blocks. The input to the encryption function For decryption, each cipher block is passed through the decryption algorithm. The result is XORed with the preceding ciphertext block to produce the plaintext block. To see that this works, we can write

$$C_j = E(K, [C_{j-1} \oplus P_j])$$

where $E(K, X)$ is the encryption of plaintext X using key K , and \oplus is the exclusive-OR operation. Then

$$D(K, C_j) = D(K, E(K, [C_{j-1} \oplus P_j]))$$

To produce the first block of ciphertext, an initialization vector (IV) is XORed with the first block of plaintext. On decryption, the IV is XORed with the output of the decryption algorithm to recover the first block of plaintext.

The IV must be known to both the sender and receiver. For maximum security, the IV should be protected as well as the key

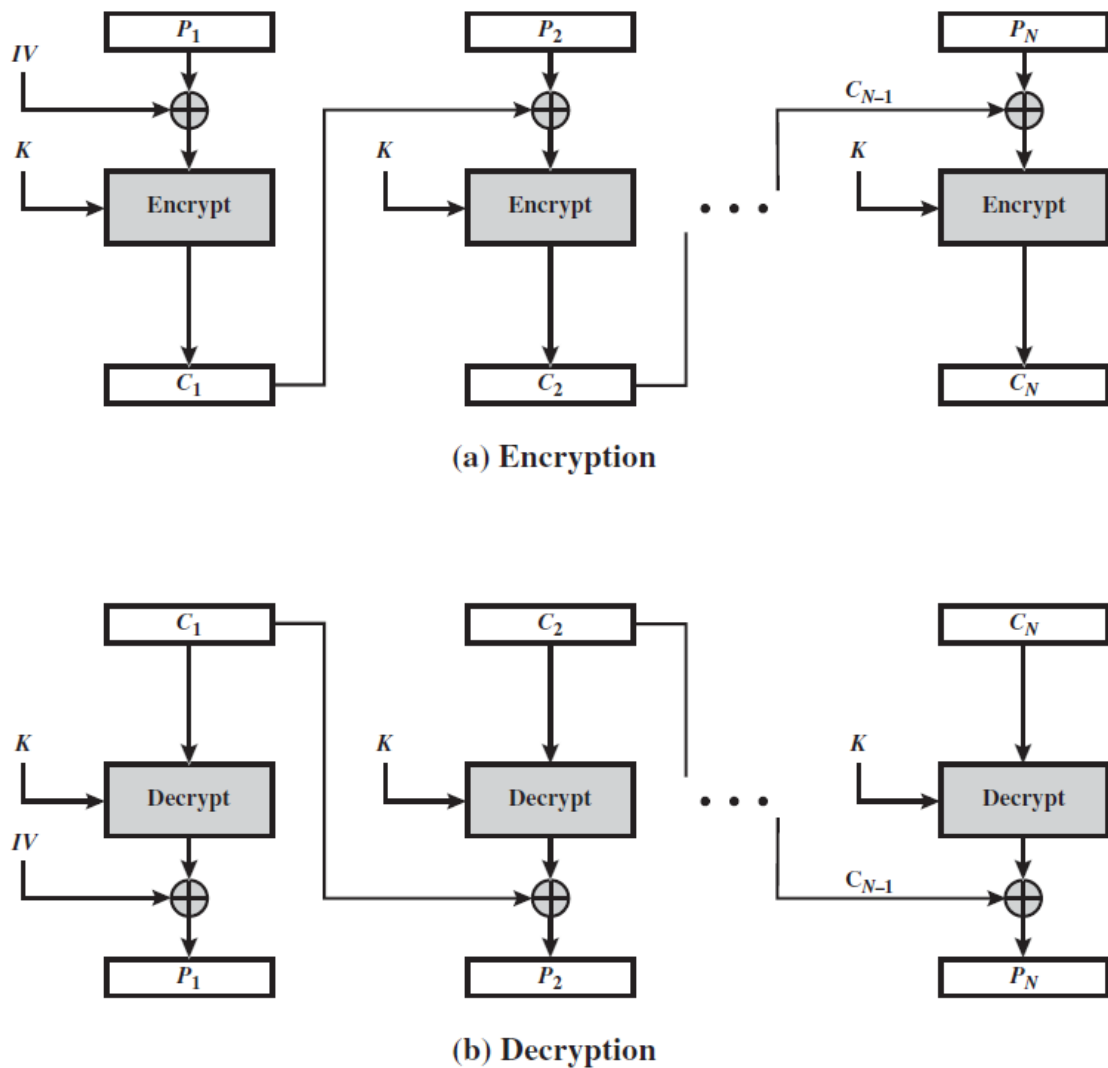


Figure 2.10 Cipher Block Chaining (CBC) Mode

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 P_1 to produce the first unit of ciphertext C_1 , which is then transmitted. In addition, the contents of the shift register are shifted left by s bits, and C_1 is placed in the rightmost (least significant) s bits of the shift register. This process continues until all plaintext units have been encrypted.

For decryption, the same scheme is used, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit. Note that it is the *encryption* function that is used, not the decryption function. This is easily explained. Let $S_s(X)$ be defined as the most significant s bits of X . Then

$$C_1 = P_1 \oplus S_s[E(K, IV)]$$

Therefore,

$$P_1 = C_1 \oplus S_s[E(K, IV)]$$

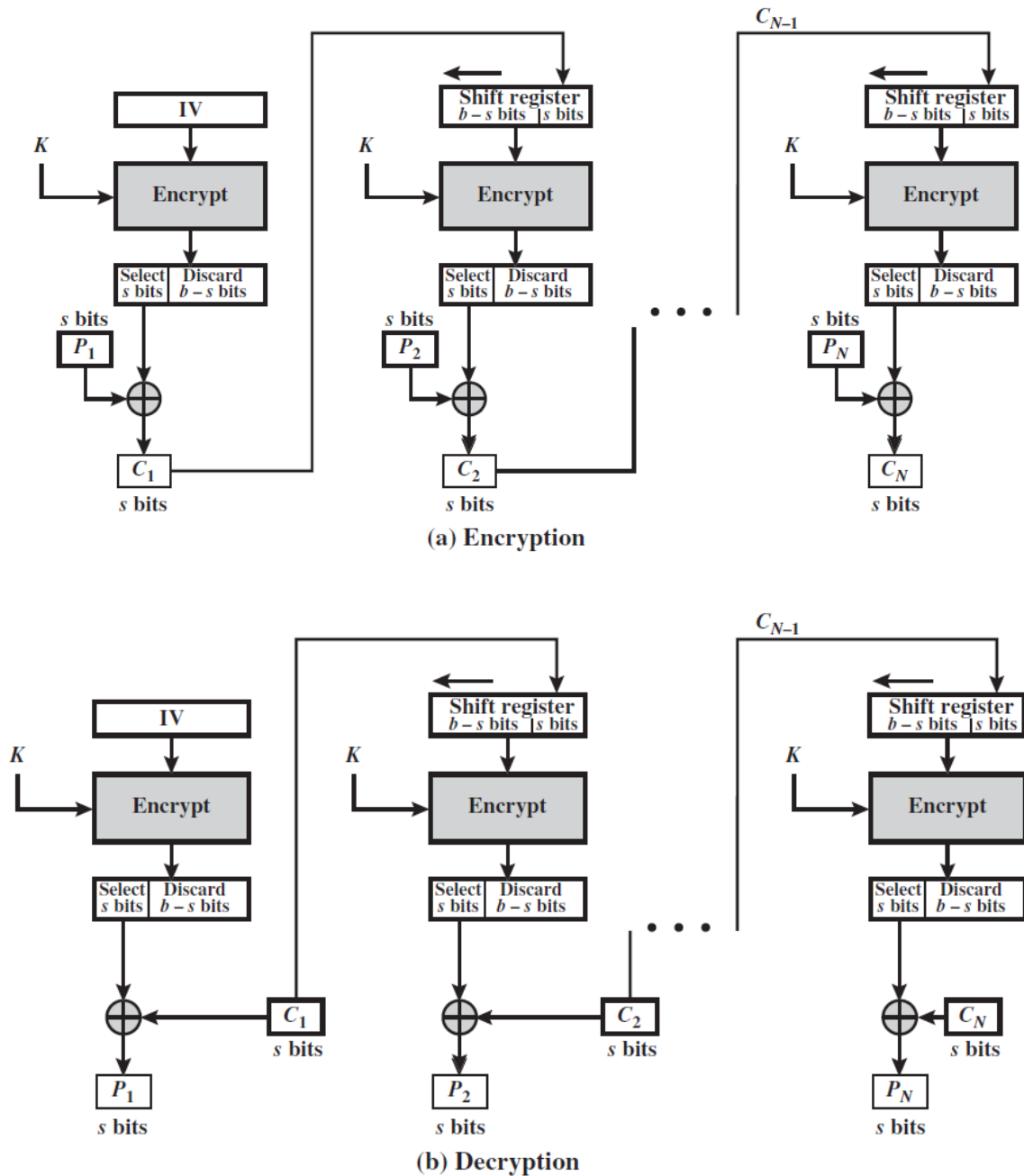


Figure 2.11 s -bit Cipher Feedback (CFB) Mode

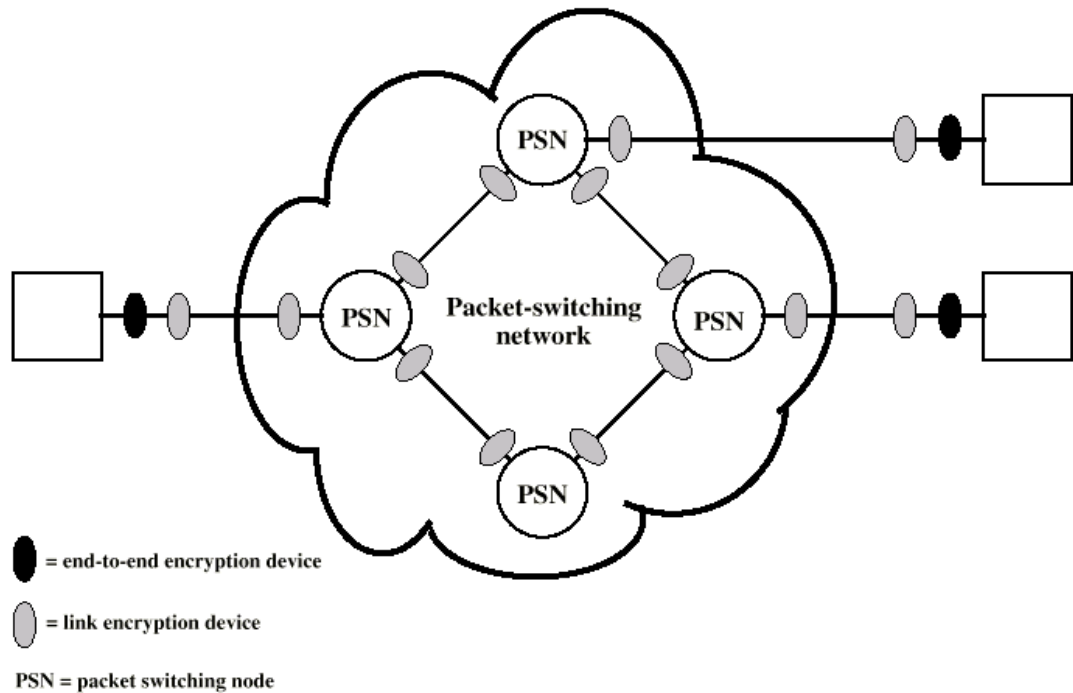
LOCATION OF ENCRYPTION DEVICES

- The most powerful, and most common, approach to countering the threats to network security is encryption.

- In using Encryption, we need to decide what to encrypt and where the encryption gear should be located. There are two fundamental alternatives:

Link

– End to End



Link Encryption:

- Each communication link equipped at both ends
- All traffic secure
- High level of security
- Requires lots of encryption devices
- Message must be decrypted at each switch to read address (virtual circuit number)

End to End Encryption:

- Encryption done at ends of system
- Data in encrypted form crosses network unaltered
- Destination shares key with source to decrypt
- Host can only encrypt user data

Otherwise switching nodes could not read header or route packet

- Traffic pattern not secure

KEY DISTRIBUTION

- Key selected by A and delivered to B
- Third party selects key and delivers to A and B
- Use old key to encrypt and transmit new key from A to B
- Use old key to transmit new key from third party to A and B