ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category–I University by MHRD-UGC]
(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003

DIRECTORATE OF DISTANCE EDUCATION

M.Sc. [Computer Science]

III - Semester

341 31

CRYPTOGRAPHY AND NETWORK SECURITY

Author

Dr. A. PADMAPRIYA
Associate Professor
Department of Computer Science
Alagappa University
Karaikudi

SYLLABI-BOOK MAPPING TABLE

Cryptography and Network Security

CONTENTS

BLOCK I : COMPUTER SECURITY INTRODUCTION

BLOCK II : BLOCK CIPHERS AND DES

BLOCK III : PUBLIC KEY CRYPTOGRAPHY AND RSA

BLOCK IV : MESSAGE AUTHENTICATION CODES

BLOCK V : TRANSPORT LEVEL SECURITY

BLOCK – 1 COMPUTER SECURITY INTRODUCTION

NOTES

UNIT -1 INTRODUCTION TO NETWORK SECURITY

Structure

1.0 INTRODUCTION

We all know that Communication is the greatest gift of mankind. Communication Networks enables the exchange of information among the networked computing devices along network links. RFC 2828 defines **information** as “*facts and ideas, which can be represented (encoded) as various forms of data*”.

A RFC (Request For Comments) is the formal document from the Internet Engineering Task Force (IETF). It is a type of text document from the technology community.

Now we are living in the information age. More information is being created, stored, processed, and communicated using computers and networks. The ***Internet***, sometimes called simply "the Net," is a worldwide system of computer networks - a network of networks in which users at any one computer can, if they have permission, get information from any other computer. Thus Computers are increasingly interconnected creating new pathways to the information assets. The threats of Information are becoming more widespread and more sophisticated. Anyone on the Net is vulnerable to attacks because, it has

- no central control
- no central authority
- no common legal oversight or regulations

NOTES

- no standard acceptable use policy

Network Security refers to any distinct activity designed to protect usability, reliability, integrity, and safety of your network and data.
In order to secure the information, it should be preserved from unauthorized access and change as well as available to the authorized entity when it is needed. Information delayed is actually information denied.

The field of **network and Internet security** consists of measures to detect, prevent, and correct security violations that involve the transmission of information.

1.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the concepts of Computer Security
- Describe the OSI Security Architecture
- Explain about Security Attacks

1.2COMPUTER SECURITY CONCEPTS

Let us start our learning process with some common examples of violations on security particularly on informationtransmitted through a network. The normal flow of information communicated is shown in the following figure.

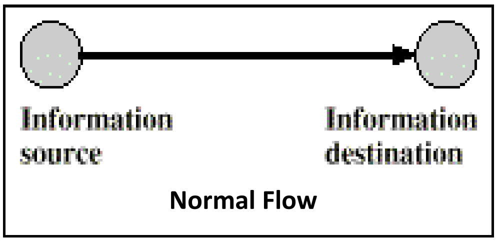


Fig 1.1. Normal Information Flow

** User A transmits a file containing sensitive information to user B. An unauthorized User C captures a copy of the file during its transmission.

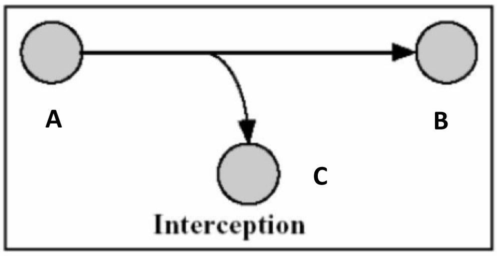


Fig 1.2. Interception

** A network manager, D, transmits a message to a computer, E, under its management. Let us say themessage instructs E to update an authorization file to grant access right to new users. User F intercepts themessage, alters its contents to add or delete entries, and then forwards the message to E. On accepting the message as coming from manager D and E updates its authorization fileaccordingly.

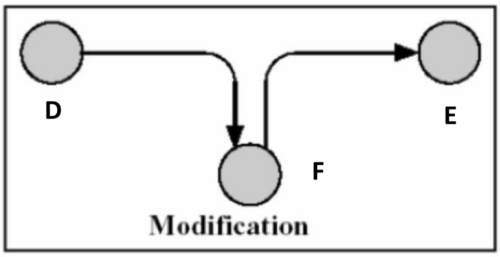


Fig 1.3. Modification

** Rather than intercept a message, user F constructs its own message with the desired entriesand transmits that message to E as if it had come from manager D. Computer E accepts themessage as coming from manager D and updates its authorization file accordingly.

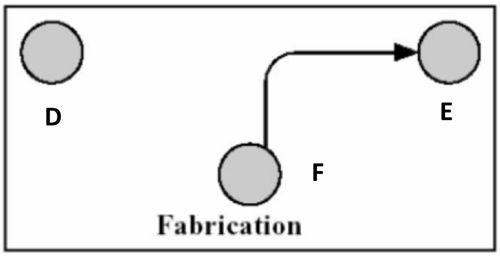


Fig 1.4. Fabrication

** An employee is fired without warning. The personnel manager sends a message to a serversystem to invalidate the employee's account. During the invalidation process, theserver is to post a notice to the employee's file as confirmation of the action. The employee isable to intercept the message and delay it long enough to make a final access to the server toretrieve sensitive information. The message is then forwarded, the action taken, and theconfirmation posted. The employee's action may go unnoticed for some considerable time.

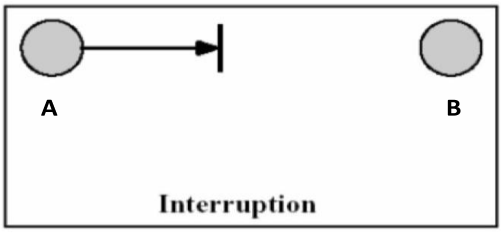


Fig 1.5. Interruption

** A message is sent from a customer to a stockbroker with instructions for various transactions.Subsequently, the investments lose value and the customer denies sending the message.

NOTES

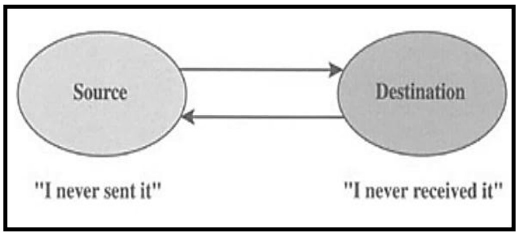


Fig 1.6. Repudiation

Although this list by no means exhausts the possible types of network securityviolations, it illustrates the range of concerns of network security. The *National Institute of Standards and Technology (NIST)Computer Security Handbook* defines the term ***Computer Security***as follows:

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information / data, and telecommunications).

This definition introduces three key objectives that are at the heart ofcomputer security:

Confidentiality: This term covers two related concepts:

- ***Dataconfidentiality:*** *Assures that private or confidential information is not made available or disclosed to unauthorized individuals.*
- ***Privacy:*** *Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.*

Integrity: This term covers two related concepts:

- ***Data integrity:*** *Assures that information and programs are changed only in a specified and authorized manner.*
- ***System integrity:*** *Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.*

Availability: Assures that systems work promptly and service is not denied toauthorize users.

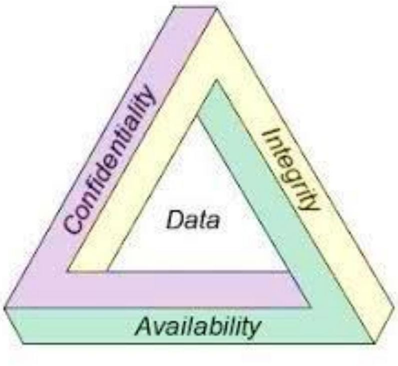


Fig 1.7. CIA Triad

These three concepts form what is often referred to as the CIA triad(Figure 1.7).The *requirements* and *definition of a loss* of security in each category is given below.

NOTES

Confidentiality:

- ✓ Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
- *A loss of confidentiality is the unauthorized disclosure of information.*

Integrity:

- ✓ Guarding against improper information modification or destruction,including ensuring information nonrepudiation and authenticity.
- *A loss of integrity is the unauthorized modification or destruction of information.*

Availability:

- ✓ Ensuring timely and reliable access to and use of information.
- *A loss of availability is the disruption of access to or use of information or an information system.*

In addition to the above mentioned three objectives two additional concepts need to present a complete picture. They are

Authenticity: The property of being genuine and being able to be verified andtrusted; confidence in the validity of a transmission, a message, or messageoriginator. This means verifying that users are who they say they are and thateach input arriving at the system came from a trusted source.

Accountability: The security goal that generates the requirement for actionsof an entity to be traced uniquely to that entity. This supports nonrepudiation,deterrence, fault isolation, intrusion detection and prevention, andafter-action recovery and legal action. Because truly secure systems are notyet an achievable goal, we must be able to trace a security breach toa responsible party. Systems must keep records of their activities to permitlater forensic analysis to trace security breaches or to aid in transactiondisputes.

Check Your Progress 1

1. What do you mean by Computer Security?
2. State the three prime objectives of Computer Security.
3. What is the need for Authenticity and Accountability?

1.3CHALLENGES OF COMPUTER SECURITY

Computer and network security is both fascinating and complex. Some of the reasonsfollow:

1. ***Security is not as simple as it might first appear*** to the novice. The requirements seem to be straightforward; indeed, most of the major requirements for security services can be given self-explanatory, one-word labels: confidentiality, authentication, nonrepudiation, or integrity. But the mechanisms used to meet those requirements can be quite complex, and understanding them may involve rather subtle reasoning.

2. In developing a particular security mechanism or algorithm, ***one must always consider potential attacks on those security features***. In many cases, successful attacks are designed by looking at the problem in a completely different way, therefore exploiting an unexpected weakness in the mechanism.

3. Because of point 2, the procedures used to provide particular services are often counterintuitive. Typically, a ***security mechanism is complex***, and it is not obvious from the statement of a particular requirement that such elaborate measures are needed. It is only when the various aspects of the threat are considered that elaborate security mechanisms make sense.

4. Having designed various security mechanisms***, it is necessary to decide where to use them***. This is true both in terms of physical placement (e.g., at what points in a network are certain security mechanisms needed) and in a logical sense [e.g., at what layer or layers of an architecture such as TCP/IP (Transmission Control Protocol/Internet Protocol) should mechanisms be placed].

5. Security mechanisms typically involve more than a particular algorithm or protocol. They also require that participants be in possession of some secret information (e.g., an encryption key), which raises questions about the ***creation, distribution, and protection of that secret information***. There also may be a reliance on communications protocols whose behaviour may complicate the task of developing the security mechanism. For example, if the proper functioning of the security mechanism requires setting time limits on the transit time of a message from sender to receiver, then any protocol or network that introduces variable, unpredictable delays may render such time limits meaningless.

6. Computer and network security is essentially a battle of wits between a perpetrator who tries to find holes and the designer/administrator who tries to close them. The great advantage that the attacker has is that he or she need only find a single weakness, while the ***designer mustfind and eliminate all weaknesses*** to achieve perfect security.

7. There is a natural tendency on the part of users and system managers to perceive little benefit from ***security investment until a security failure occurs***.

8. Security ***requires regular, even constant, monitoring***, and this is difficult in today's short-term, overloaded environment.

9. Security is still too often an afterthought to be incorporated into a system after the design is complete rather than ***being an integral part of the design process***.

10. Many users and even security ***administrators view strong security as an obstacle to efficient and user-friendly operation*** of an information system or use of information.

1.4 THEOSI SECURITY ARCHITECTURE

Do you know?

Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Attack

An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

To assess effectively the security needs of an organization and to evaluate and choosevarious security products and policies, the manager responsible for security needssome systematic way of defining the requirements for security and characterizing theapproaches to satisfying those requirements. This is difficult enough in a centralizeddata processing environment; with the use of local and wide area networks, theproblems are compounded.

International Telecommunication Union (ITU) *Telecommunication Standardization Sector* (ITU-T) Recommendation X.800, Security Architecture for *Open Systems Interconnection* (OSI), defines such asystematic approach.

The OSI security architecture is useful to managers as a wayof organizing the task of providing security. Furthermore, because this architecturewas developed as an international standard, computer and communications vendorshave developed security features for their products and services that relate to thisstructured definition of services and mechanisms.The OSI security architecturefocuses on security attacks, mechanisms, and services.

NOTES

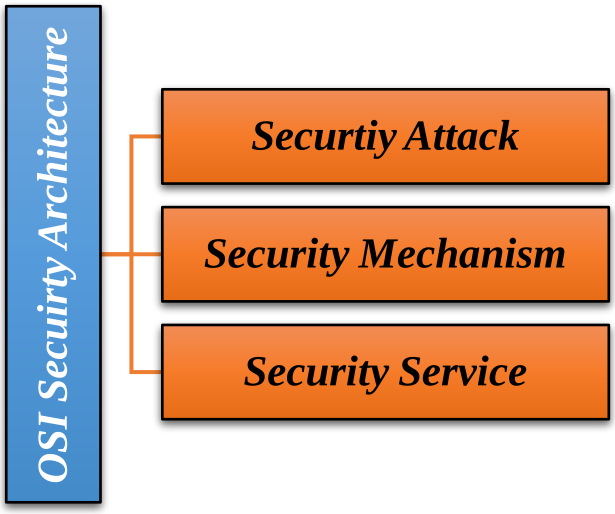


Fig 1.8. OSI Security Architecture

These can be definedbriefly as

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.

1.5SECURITY ATTACKS

Security attacks can be categorized as ***passive attacks***and ***active attacks***.

Passive Attacks

A *passive attack* attempts to learn ormake use of information from the system but does not affect system resources.

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

- ***Release of message contents*** means the opponent gains access to the message being transmitted. It is also called as *Snooping*
- In ***traffic analysis***, even though the message is encrypted, the opponent could determine the location and identity of communicating hosts and

could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Example :Interception

Passive attacks are very difficult to detect, because they do not involve any alteration of the data.Typically, the message traffic is sent and received in an apparentlynormal fashion, and neither the sender nor receiver is aware that a third partyhas read the messages or observed the traffic pattern. However, it is feasible to preventthe success of these attacks, usually by means of encryption.Thus, the emphasisin dealing with passive attacks is on prevention rather than detection.

Active attacks

An*active attack* attempts to alter system resources or affect their operation.

They involve some modification of the data stream or the creation of a falsestream and can be subdivided into four categories:

A ***masquerade***takes place when one entity pretends to be a different entity i.e. attacker impersonates somebody. It is also called as *spoofing*.

Example : Fake bank websites

Replayinvolves the passive capture of a data unit and its subsequentretransmissionto produce an unauthorized effect. Here, attacker obtains the copy of message sent by user and later tries to replay it.

Example : Attacker tries to receive another payment from copy of user's claim

Modification of messagessimply means that some portion of a legitimatemessage is altered, or that messages are delayed or reordered, to produce an unauthorizedeffect.

Example : Modifying name of the beneficiary account during online fund transfers.

The ***denial of service***prevents or inhibits the normal use or management ofcommunications facilities. Another form of service denial is the disruption of anentire network, either by disabling the network or by overloading it with messages soas to degrade performance.

Example :Repudiation, Blocking services of a commercial website by overloading requests during festival offer times

Active attacks present the opposite characteristics of passive attacks.Whereaspassive attacks are difficult to detect, measures are available to prevent their success On the other hand, it is quite difficult to prevent active attacks absolutely because ofthe wide variety of potential physical, software, and network vulnerabilities. Instead,the goal is to detect

NOTES

active attacks and to recover from any disruption ordelays caused by them. If the detection has a deterrent effect, it may also contributeto prevention.

Check Your Progress 2

4. What are the aspects involved in OSI Security architecture?
5. What are the types of security attacks?
6. Differentiate passive and active attacks

1.6ANSWERS TO CHECK YOUR PROGRESS

1. Computer Security is defined as the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources such as

    i. Hardware
    ii. Software
    iii. Firmware
    iv. Information / data
    v. Telecommunications

2. The three prime objectives of Computer Security are Confidentiality, Integrity and Availability (CIA).

3. Authenticity and Accountability are important because
    i. Authenticity is the property of being genuine. It is needed to ensure that each input arriving at the system came from a trusted source.
    ii. Accountability supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, andafter-action recovery and legal action.

4. The aspects involved in the OSI security architecture are
    i. Security attacks
    ii. Security mechanisms
    iii. Security services

5. Security attacks are classified as Passive attacks and Active attacks

6. The differences between the active and passive attacks are listed below

Passive Attack	Active Attack
Attackers goal is to obtain information	Attackers goal is to alter information
Does not modify / harm information	Change / harm information
System is not affected	System is affected
Confidentiality threat	Integrity & Availability threat
Difficult to detect until the sender / receiver finds out leakages	Easy to detect
Can be prevented by encipherment	Hard to prevent (Attacker can launch them in variety of ways)

1.7SUMMARY

This unit describes the basic concepts about Computer Security. Internet is a network ofnetworks where lot of information is available and is meant to be utilized by you. Noone owns the Internet. It consists of a large number of Interconnected autonomousnetworks that connect millions of computers across the world. This lack of central control leads to several security issues. The unit describes thebasic concepts, challenges of Computer Security, the CIA triads, OSI Security Architecture and Security attacks. Check your progress helps you to understand the concepts better.

1.8KEY WORDS

- The **Open Systems Interconnection (OSI) security architecture** provides a systematic framework for defining security attacks, mechanisms, and services.
- **Security attacks** are classified as either passive attacks, which include unauthorized reading of a message of file and traffic analysis or active attacks, such as modification of messages or files, and denial of service.

1.9 SELF-ASSESSMENT EXERCISES

Short Questions

1. “Information Security of Internet is promising” – Justify.
2. Define Network Security.
3. Give some examples for security violations
4. How will you classify network security attacks?

NOTES

Detail Questions

1. Discuss in detail about CIA triads.
2. Describe the OSI Security architecture.
3. Explain about the security attacks with suitable examples.

1.10SUGGESTED READINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Stallings, W., and Brown, L. *Computer Security*. Upper Saddle River, NJ: Prentice Hall, 2008
3. Browne, P. “Computer Security - A Survey.” *ACM SIGMIS Database*, Fall 1972.
4. Web Reference :IEEE Technical Committee on Security and Privacy: Copies of their newsletter and information on IEEE-related activities.

UNIT -2 SECURITY SERVICES AND MECHANISMS

NOTES

Structure

2.0 INTRODUCTION

The security can be achieved through security services and mechanisms. This unit will cover security services, mechanisms and model for network security. The commonly accepted aspects of security are the security services. There are five security services. In this unit the security services are discussed. The services can be provided through security mechanisms. The common model for network security is described in this unit.

2.1 OBJECTIVES

After going through this unit, you will be able to:

- Explain aboutsecurity services
- Describe the need for security mechanisms
- Understand the model for network security

2.2 SECURITY SERVICES

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

As per RFC 2828, security service is processing or communication service that is provided by a system to give a specific kind of protection to system resources; security servicesimplement security policies and are implemented by security mechanisms.

According to X.800, there are five categories of security services and fourteen specific services. The five categories of security services are shown in the following figure 2.1.

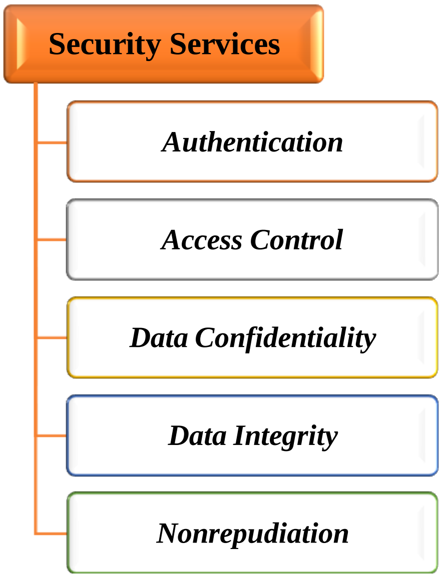


Fig 2.1.Categories of Security Services

Authentication

The authentication service is concerned with assuring that a communication isauthentic.

Case	Function of authentication service
Single Message such as a warning or alarm signal	To assure the recipient that the message is from the source that it claims to be from
Ongoing interaction such as theconnection of a terminal to a host	1. At the time of connection initiation, the service assures that the two entities are authentic, that is, each is the entity that it claims to be. 2. The service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception

Two specific authentication services are defined in X.800.

Peer Entity Authentication

Two entities are considered peers ifthey implement to same protocol in different systems; e.g., two TCP modulesin two communicating systems. Peer entity authentication is providedfor use at the establishment of, or at times during the data transfer phaseof, a connection. It attempts to provide confidence that an entity is notperforming either a masquerade or an unauthorized replay of a previousconnection.

- Provides for the evidence which confirms the identity of a peer entity in an association.
- Used in association with a logical connection to provide confidence in the identity of the entities connected.

Data-Origin Authentication

It does not provide protection against the duplication ormodification of data units. This type of service supports applications likeelectronic mail, where there are no prior interactions between the communicatingentities.

- Provides for the evidence which confirms the source of a data unit.
- In a connectionless transfer, provides assurance thatthe source of received data is as claimed.

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 trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

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.

Service	Function
Connection Confidentiality	The protection of all user data on a connection
Connectionless Confidentiality	The protection of all user data in a single data block
Selective-Field Confidentiality	The confidentiality of selected fields within the user data on a connection or in a single data block
Traffic-Flow Confidentiality	The protection of the information that might be derived from observation of traffic flows

Data Integrity

The assurance that data received are exactly assent by an authorized entity (i.e., contain nomodification, insertion, deletion, or replay).

NOTES

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.

Service	Function
Connection Integrity with Recovery	Provides for the integrity of all user data on a connection and detects anymodification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted
Connection Integrity without Recovery	As above, but provides only detection without recovery
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.
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.
Selective-Field Connectionless Integrity	Provides for the integrity of selected fields within a singleconnectionless data block; takes the form of determination of whether the selected fields have been modified

Nonrepudiation

Provides protection against denial by one of theentities involved in a communication of havingparticipated in all or part of the communication

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

Service	Function
Nonrepudiation, Origin	Proof that the message was sent by the specified party
Nonrepudiation,	Proof that the message was received

Destination	by the specified party.

Availability Service

An availability service is one that protects a system to ensure its availability.

Both X.800 and RFC 2828 define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system. A variety of attacks can result in the loss of or reduction in availability. X.800 treats availability as a property to be associated with various security services. This service addresses the security concerns raised by denial-of-service attacks. It depends on proper management and control of system resources and thus depends on access control service and other security services.

2.3 SECURITY MECHANISMS

A **security mechanism** is any process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.The mechanisms aredivided into those that are implemented in a specific protocol layer, such as TCP oran application-layer protocol, and those that are not specific to any particularprotocol layer or security service.

The security mechanisms defined in X.800 are listed below. They are grouped as

- Specific Security Mechanisms
- Pervasive Security Mechanisms

The ***specific security mechanisms*** may be incorporated into the appropriate protocollayer in order to provide some of the OSI securityservices.

Mechanism	Explanation
Encipherment	The use of mathematical algorithms to transform data into a form that is not readily intelligible. Thetransformation and subsequent recovery of thedata depend on an algorithm and zero or more encryption keys.
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).
Access Control	A variety of mechanisms that enforce access rights to resources.
Data Integrity	A variety of mechanisms used to assure the integrity of a data unit or stream of data units
Authentication Exchange	A mechanism intended to ensure the identity of anentity by means of information exchange
Traffic Padding	The insertion of bits into gaps in a data stream tofrustrate traffic analysis attempts
Routing Control	Enables selection of particular physically secureroutes for certain data and allows routing changes,especially when a breach of security is suspected
Notarization	The use of a trusted third party to assure certainproperties of a data exchange.

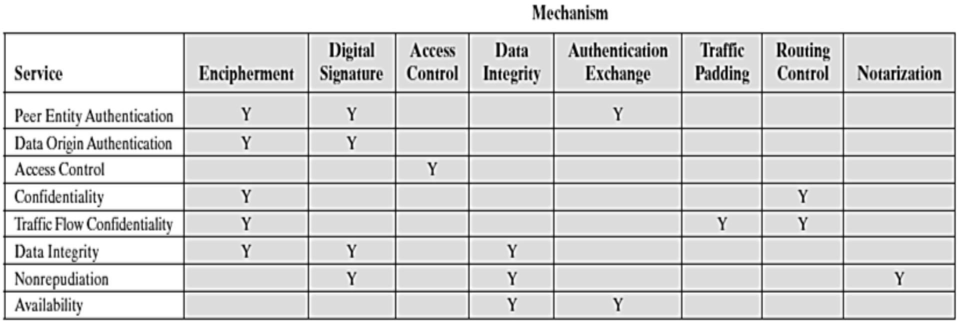
X.800 distinguishes between reversible enciphermentmechanisms and irreversible encipherment mechanisms.

- A ***reversible encipherment*** mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted.
- ***Irreversible encipherment*** mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications.

The ***pervasive security mechanisms*** are mechanisms that are not specific to any particularOSI security service or protocol layer.

Mechanism	Explanation
Trusted Functionality	That which is perceived to be correct with respect tosome criteria (e.g., as established by a security policy).
Security Label	The marking bound to a resource (which may be a data unit) that names or designates the securityattributes of that resource.
Event Detection	Detection of security-relevant events.
Security Audit Trail	Data collected and potentially used to facilitate a security audit, which is an independent review andexamination of system records and activities.
Security Recovery	Deals with requests from mechanisms, such as eventhandling and management functions, and takesrecovery actions.

The relationship between the security service and mechanisms is given below.



	Mechanism							
Service	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			

Check Your Progress 1

1. What do you mean by Security services?
2. What is an availability service?
3. State the uses of encipherment.

2.4 MODEL FOR NETWORK SECURITY

A model for network security is shown in the following figure.

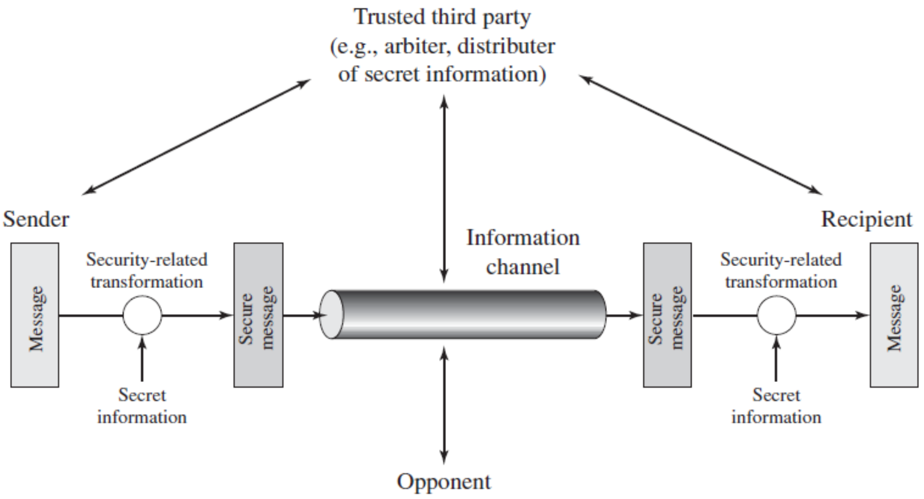


Fig 2.2.Model for Network Security

A ***message*** is to be transferred from one party (***sender***) to another (***recipient***) across some sortof Internet service. The two parties, who are the *principals* in this transaction, mustcooperate for the exchange to take place.

A logical ***information channel*** is establishedby defining a route through the Internet from source to destination and by thecooperative use of communication protocols (e.g., TCP/IP) by the two principals.

NOTES

Security aspects come into play when it is necessary or desirable to protectthe information transmission from an ***opponent*** who may present a threat to confidentiality,authenticity, and so on.

All the techniques for providing security havetwo components:

- 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.

- 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

A ***trusted third party*** may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information tothe two principals while keeping it from any opponent. Or a third party may beneeded to arbitrate disputes between the two principals concerning the authenticityof a message transmission.

This general model shows that there are four basic tasks in designing a particularsecurity 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.

Check Your Progress 2

4. What do you mean by Information channel?
5. What are the two components of the techniques providing security?
6. What is the need for Trusted third party?

2.5 ANSWERS TO CHECK YOUR PROGRESS

1. ***Security service*** as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

2. An availability service is one that protects a system to ensure its availability
3. Encipherment will transform data into a form that is not readily intelligible using mathematical algorithms. This transformation will preserve the data from the opponent.

4. A logical ***information channel*** is establishedby defining a route through the Internet from source to destination and by thecooperative use of communication protocols.

5. The two components are security related transformations and secret information shared between the sender and receiver.

6. A ***trusted third party*** may be needed to achieve secure transmission.
   - May be responsible for distributing the secret information tothe two principals (sender and receiver)
   - May beneeded to arbitrate disputes between the two principals concerning the authenticityof a message transmission

2.6 SUMMARY

Detailed description about the five major categories of security services are given in this unit. In addition to that, thedifferent mechanisms to provide the security services areelaborately discussed. The mechanisms are tabulated for better understanding. The model for network security is explained in the final section on this unit. Also the tasks involved in designing a security service is discussed.

2.7 KEYWORDS

- A **security mechanism** is any process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack. Examples of mechanisms are encryption algorithms, digital signatures, and authentication protocols.

- **Authentication** is the assurance that the communicating entity is theone that it claims to be.

- **Access control** is the prevention of unauthorized use of a resource

- **Data confidentiality** is the protection of data from unauthorized disclosure.

NOTES

- **Data integrity** is the assurance that data received are exactly as sent by an authorized entity

- **Nonrepudiation** provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

- **Security services** include authentication, access control, data confidentiality, data integrity, nonrepudiation, and availability.

2.8 SELF-ASSESSMENT EXERCISES

Short Questions

1. What is the difference between peer entity authentication and data origin authentication?
2. What do you mean by security mechanisms?
3. What are components of the network security model?
4. What are the types of encipherment mechanisms?

Detail Questions

1. Explain about security services.
2. Discuss in detail about the security mechanisms
3. How will you relate security services with mechanisms? Discuss.
4. Describe about the model for network security.

2.9SUGGESTED READINGS

1. William Stallings, "Cryptography and Network Security Principles and Practice", Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Fraser, B. *Site Security Handbook.* RFC 2196, September 1997.
4. Web Reference :IBM Knowledge centre.

UNIT - 3 CLASSIC ENCRYPTION TECHNIQUES

NOTES

Structure

3.0 INTRODUCTION

Symmetric encryption also referred to as conventional encryption or single-keyencryption was the only type of encryption in use prior to the development of publickeyencryption in the 1970s. It remains by far the most widely used of the two types ofencryption. This unit will discuss about the model for general symmetric encryption. The traditional ciphers namely the substitution and transpositions ciphers are discussed to understand the context within which the symmetric encryption algorithms are used.

3.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the symmetric encryption model
- Describe the working of traditional ciphers
- Encrypt and decrypt messages using simple substitution methods

NOTES

- Understand the context within which the algorithms are used
- Understand the weakness of encryption methods

3.2 BASIC TERMS

Before entering into this unit, the readers need to know the definition of the basic terms. They are given below.

- An original message is known as the ***plaintext***, while the coded message is called the ***ciphertext.***
- The process of converting from plaintext to ciphertext is known as ***enciphering or encryption***; restoring the plaintext from the ciphertext is ***deciphering or decryption***.
- The many schemes used for encryption constitute the area of study known as ***cryptography***. Such a scheme is known as a cryptographic system or a cipher.
- Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of ***cryptanalysis***. Cryptanalysis is what the layperson calls "breaking the code."
- The areas of cryptography and cryptanalysis together are called ***cryptology***.

3.3 SYMMETRIC CIPHER MODEL

A symmetric encryption scheme has five components as shown in figure 3.1.

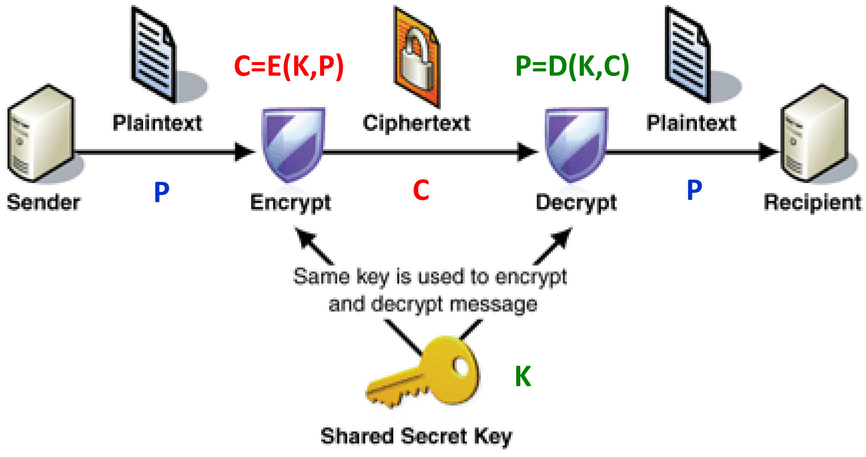


Fig 3.1.Simplified Model of Symmetric Encryption

1. ***Plaintext (P):*** This is the original intelligible message or data that is fed into the algorithm as input.

2. ***Encryption algorithm (E):*** The encryption algorithm performs various substitutions and transformations on the plaintext.

3. ***Secret key (K):*** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of

the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

4. ***Ciphertext (C):*** 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. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.

5. ***Decryption algorithm (D):*** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Let us have an example for symmetric encryption

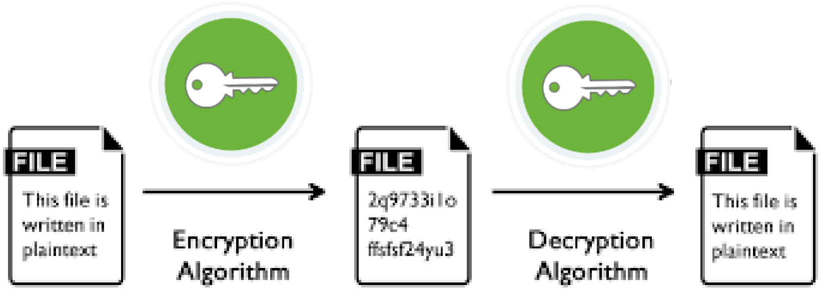

FILE
This file is written in plaintext
Encryption Algorithm
FILE
2q9733i lo 79c4 ffsfsf24yu3
Decryption Algorithm
FILE
This file is written in plaintext

Fig 3.2. Example for Symmetric Encryption

The plain text is converted to cipher text and vice versa using single key. This key will be shared between the sender and the receiver.

There are two *requirements for secure use of conventional encryption*:

1) ***A strong encryption algorithm***

   The algorithm should be strong enough such that even 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) ***Secure transmission of Key***

   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.

Let us take a closer look into this.

A source produces a message in plaintext,$X = [X_1, X_2, ..., X_M]$.
For encryption, a key of the form$K = [K_1, K_2, ..., K_M]$is generated.
With the message X and the encryption key K as input, the encryption algorithm forms the cipher text $Y= [Y_1, Y_2, ..., Y_N]$.

Then, $Y= E(K, X)$

The intended receiver, in possession of the key, is able to invert the transformation using decryption algorithm and the secret key.

$X= D (K, Y)$

An opponent, observing Y but not having access to K or X, may attempt to recover X or K or both X and K. It is assumed that the opponent knows the encryption (E) and decryption (D) algorithms. If the opponent is interested in only this particular message, then the focus of the effort is to recover X by generating a plaintext estimate$\widehat{X}$. Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover K by generating an estimate $\widehat{K}$.

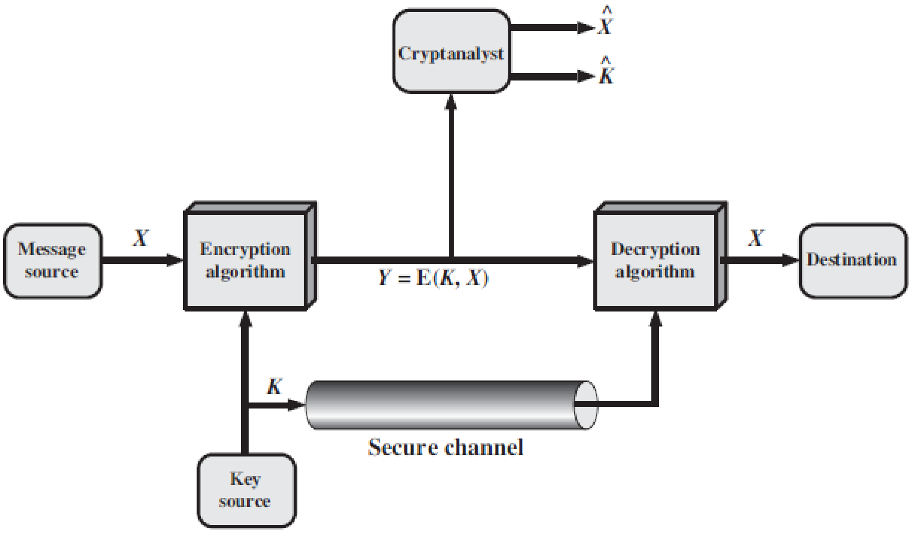


Fig 3.3. Model of Conventional Cryptosystem

3.3.1 Cryptography

Cryptographic systems are characterized along three independent dimensions:

1. *The type of operations used for transforming plaintext to ciphertext*.
    - All encryption algorithms are based on either substitution or transposition
    - **Substitution -** each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element
    - **Transposition -** elements in the plaintext are rearranged.
    - The fundamental requirement is that no information be lost (that is, that all operations are reversible).
    - Most 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, private-key, secret-key, or conventional encryption.
- If the sender and receiver use different keys, the system is referred to as **asymmetric**, two-key, or public-key encryption.

3. *The way in which the plaintext is processed*.
   - The plaintext may be processed either as block or as stream.
   - 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.

3.3.2 Cryptanalysis and Brute-Force Attack

Generally, the attack is made to recover the key-in-use instead of recovering the plain text of a single cipher text. There are two general approaches in attack, the first one is Cryptanalysis and the second one is Brute Force attack.

Cryptanalysis is a branch of the cryptography subject dealing with the solution and acquiring of cryptic messages. Cryptanalytic attacks rely on the nature of the algorithm plusperhaps some knowledge of the general characteristics of the plaintext oreven some sample plaintext–ciphertext pairs. This type of attack exploits thecharacteristics of the algorithm to attempt to deduce a specific plaintext or todeduce the key being used.

.

The following table summarizes the various types of cryptanalytic attacks based on theamount of information known to the cryptanalyst.

Table 3.1. Types of attacks on Encrypted messages

Type of Attack	Known to Cryptanalyst	Remarks
Ciphertext Only	• Encryption algorithm • Ciphertext	The ciphertext-only attack is the easiest to defend against because the opponenthas the least amount of information to work with. *Only relatively weak algorithms fails to withstand the Ciphertext attack*
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext–ciphertext pairs formed with the secret key	The analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed. *Generally, an encryption algorithm is designed to*

NOTES

		withstand a known-plaintextattack. *Example : there may be a standardized header or banner to an electronic funds transfer message*
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertextgenerated with the secret key	If the analyst is able to choose the messages to encrypt, the analyst maydeliberately pick patterns that can be expected to reveal the structure of the key *Example : source code for a program developed by Corporation X might include a copyright statement in some standardized position*
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key	It is less commonly employed as cryptanalytic techniques but is nevertheless possible avenues of attack
Chosen Text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its correspondingciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key	It is less commonly employed as cryptanalytic techniques but is nevertheless possible avenues of attack

An encryption scheme is***unconditionally secure*** if the ciphertext generated by the scheme does not containenough information to determine uniquely the corresponding plaintext, nomatter how much ciphertext is available.

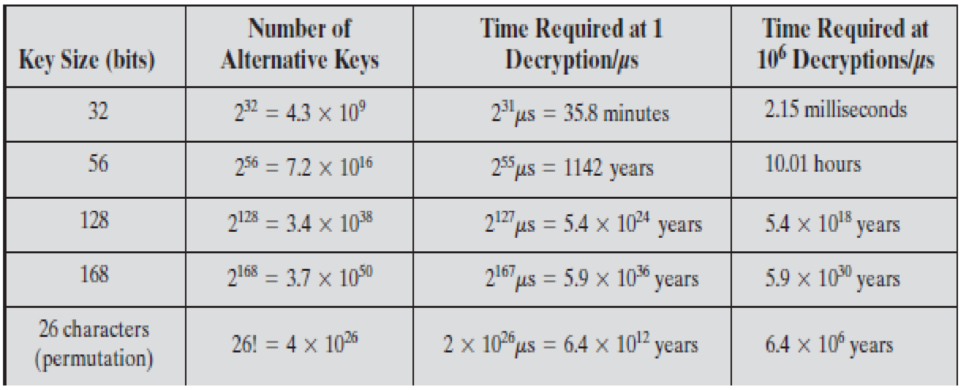
An encryption scheme is said to be ***computationally secure***if either of the two criteria are met.

- ✓ 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 tries every possible key on cipher text till a relevant message is obtained. A brute force attack is a trial-and-error method used to acquire details such as a user's password or Personal Identification Number (PIN). Usually, a brute force attack takes place by automated software which generates a large number of consecutive guesses likely to be desired data. Dictionary attack is one example of a type of brute force attack, which tries different words from the dictionary. This attack also tries different combinations of letters, numbers and special characters. The following table shows the average time required for exhaustive key search for varying key size.

Table 3.2. Average time required for Exhaustive Key Search



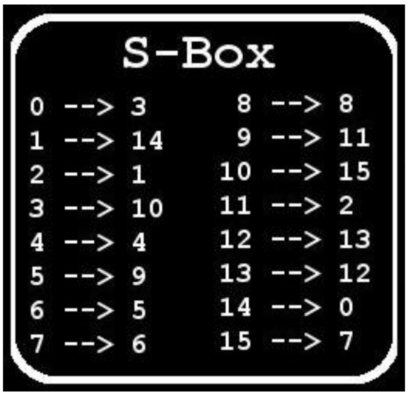
Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}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

Check Your Progress 1

1. State the two requirements for the secure user of conventional encryption.
2. What do you mean by symmetric and asymmetric cryptography?
3. What are block and stream ciphers?
4. List the different types of attacks on encrypted messages.
5. When will you say that an encryption scheme is computationally secure?

3.4SUBSTITUTION CIPHERS

A ***substitution*** technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequenceof bits, then substitution involves replacing plaintext bit patterns with ciphertext bitpatterns.



```
S-Box
0 --> 3      8 --> 8
1 --> 14     9 --> 11
2 --> 1     10 --> 15
3 --> 10    11 --> 2
4 --> 4     12 --> 13
5 --> 9     13 --> 12
6 --> 5     14 --> 0
7 --> 6     15 --> 7
```

Substitution (S-Box)

3.4.1 Caesar Cipher

The earliest known, and the simplest, use of a substitution cipher was by JuliusCaesar.The Caesar cipher involves replacing each letter of the alphabet with the letterstanding "*k*" places further down the alphabet.

Let us define the transformations for all the alphabets. Here the alphabet set is wrapped around, which means *a* follows *z*. If the numbers 0 to 25 are assigned to alphabets then,

Position	0	1	2	3	4	5	6	7	8	9	10	11	12
Alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m

Position	13	14	15	16	17	18	19	20	21	22	23	24	25
Alphabet	n	o	p	q	r	s	t	u	v	w	x	y	z

In the transformation given below, *k* = 16. Here *k* is the Key.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	Q	W	E	R	T	Y	U	I	O	P	A	S	D

Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	F	G	H	J	K	L	Z	X	C	V	B	N	M

In the above transformation the plaintext is given in lower case letters and ciphertext is given in upper case letters for better understanding.

For example,

Plaintext : kill king night

During encryption every alphabet is substituted by another alphabet standing 16 places further.

a at position 0 is substituted by (0+16) mod 26
= 16th position

i.e. alphabet *Q*
k at position 10 is substituted by (10+16) mod 26 = 0th position
i.e. alphabet *A*

*Note :*If the result is 26 or larger, we subtract 26 so that it falls back in the desired range.

Plaintext	k	i	l	l	k	i	n	g	n	i	g	h	t
Ciphertext	A	O	S	S	A	O	F	U	F	O	U	I	Z

During decryption the process is reversed. Every alphabet is substituted by another alphabet as given below.

Q at position 16 is substituted by (16 – 16) mod 26 = 0th position
i.e. alphabet *a*

A at position 0 is substituted by (0 – 16) mod 26 = 10th position
i.e. alphabet *k*

*Note :*Add 26 to the answer if it turns out negative. Here, the result of 0 – 16 is negative, so26 is added to the answer -16+26 = 10. This can be easily understood with the help of the number circle in the below figure.

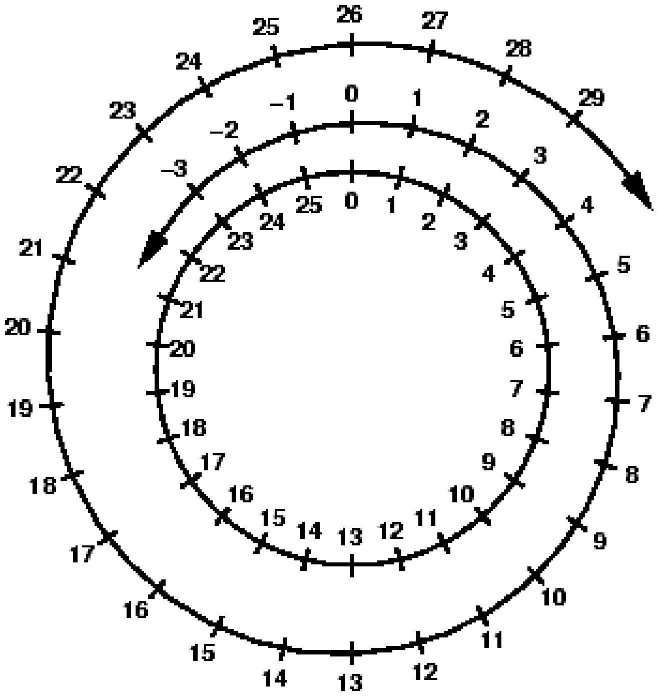


Fig 3.4. Number Circle for base 26

In general, c = E(16, p) = (p+16) mod 26

NOTES

A shift of *k* characters is the general Caesar algorithm. Encryption and decryption formulas are given as:

Encryption c = E (*k*, p) = (p + *k*) mod 26, where *k*= 1 to 25

Decryption p = D (*k*, c) = (c-*k*) mod 26

It is known that if Caesar cipher technique is used, cryptanalytic attack is easy. Attacker can try values 1 to 25 for *k* systematically and whichever *k* gives intelligible text is the key used.

Table 3.3. Brute-Force Cryptanalysis of Caesar cipher

	A	O	S	S	A	O	F	U	F	O	U	I	Z
KEY													
1	z	x	a	a	z	x	c	v	c	x	v	w	i
2	y	w	z	z	y	w	b	u	b	w	u	v	h
3	x	v	y	y	x	v	a	t	a	v	t	u	g
4	w	u	x	x	w	u	z	s	z	u	s	t	f
5	v	t	w	w	v	t	y	r	y	t	r	s	e
6	u	s	v	v	u	s	x	q	x	s	q	r	d
7	t	r	u	u	t	r	w	p	w	r	p	q	c
8	s	q	t	t	s	q	v	o	v	q	o	p	b
9	r	p	s	s	r	p	u	n	u	p	n	o	a
10	q	o	r	r	q	o	t	m	t	o	m	n	z
11	p	n	q	q	p	n	s	l	s	n	l	m	y
12	o	m	p	p	o	m	r	k	r	m	k	l	x
13	n	l	o	o	n	l	q	j	q	l	j	k	w
14	m	k	n	n	m	k	p	i	p	k	i	j	v
15	l	j	m	m	l	j	o	h	o	j	h	i	u
16	**k**	**i**	**l**	**l**	**k**	**i**	**n**	**g**	**n**	**i**	**g**	**h**	**t**
17	j	h	k	k	j	h	m	f	m	h	f	g	s
18	i	g	j	j	i	g	l	e	l	g	e	f	r
19	h	f	i	i	h	f	k	d	k	f	d	e	q
20	g	e	h	h	g	e	j	c	j	e	c	d	p
21	f	d	g	g	f	d	i	b	i	d	b	c	o
22	e	c	f	f	e	c	h	a	h	c	a	b	n
23	d	b	e	e	d	b	g	z	g	b	z	a	m
24	c	a	d	d	c	a	f	y	f	a	y	z	l
25	b	z	c	c	b	z	e	x	e	x	x	y	k

In the above problem, the plaintext leaps out in the 16th attempt.

Three important characteristics of this problem enabled us to use a brute-forcecryptanalysis:

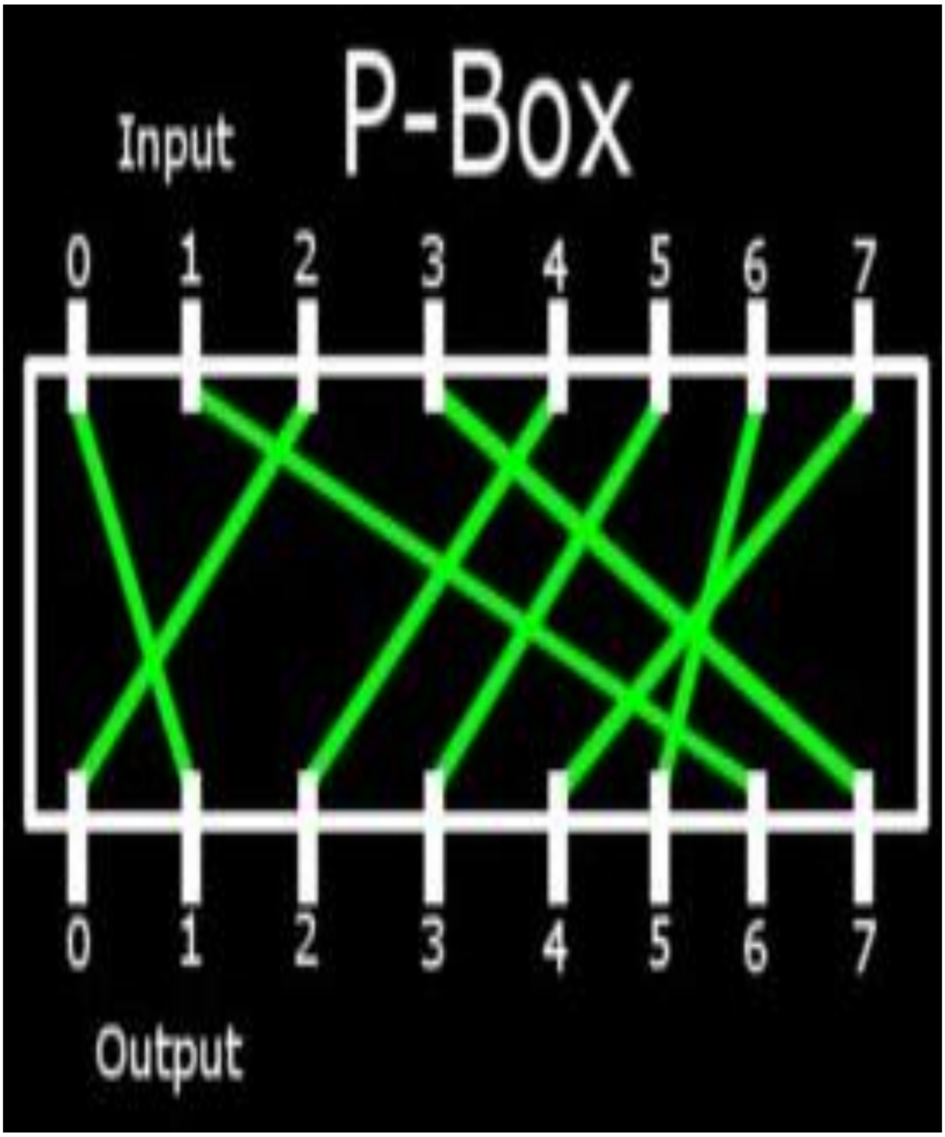
1. The encryption and decryption algorithms are known.
2. There are only 25 keys to try.

3. The language of the plaintext is known and easily recognizable.

3.4.2 MonoalphabeticCiphers

Caesar Cipher is not a secure cryptosystem because there are only 25 possible keys to try out. An attacker can carry out an exhaustive key search with available limited computing resources.Monoalphabetic Cipher is an improvement to the Caesar Cipher. Instead of shifting the alphabets by some number, this scheme uses some permutation of the letters in alphabet.

A **permutation** of a finite set of elements*S*is an ordered sequence of all the elements of *S*, with each element appearing exactlyonce.



Permutation (P-Box)

For example, if S = { a, b, c }, there are six permutations of S as given below

abc, acb, bac, bca, cab, cba

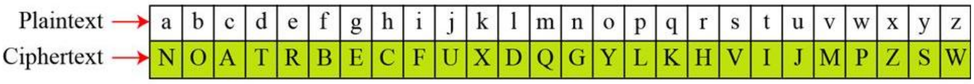
In general, there are *n*! permutations of a set of *n*elements, because the firstelement can be chosen in one of *n* ways, the second in *n-1*ways, the third in*n-2* ways, and so on.

If the “cipher” line can be any permutation of the 26 alphabetic characters then, there are

26! = 403291461126605635584000000 = 4 x 10^{26} keys

This large key space would eliminate the brute-force attack.

Instead of relying on a specific pattern for converting plaintext to ciphertext (Caesar ciphers), it is a better solution to create a mapping between each plaintext character and the corresponding ciphertext character.



Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

Fig 3.5.Monoalphabetic Substitution Table

Note : There is no order in the mapping. It is an example for permutation.

Let us take the same example,

Plaintext : kill king night

The ciphertext is found based on the mapping table in figure 3.5.

Plaintext	**k**	**i**	**l**	**l**	**k**	**i**	**n**	**g**	**n**	**i**	**g**	**h**	**t**
Ciphertext	X	F	D	D	X	F	G	E	G	F	E	C	I

Here for every *k*, the alphabet *X* is substituted. This may give clue to the cryptanalyst. Moreover, the ciphertext is also in English in the above example.

There is, however, another line of attack. If the cryptanalyst knows the nature of the plaintext, then the analyst can exploit the regularities of the language.

Some of them are

- *Relative frequency of letters* – may be determined and compared with the standard frequency distribution in English.
- *Digrams* – the most common two letter combinations such as *th*, *in*, *er*, *re*, and *an*
- *Trigrams* – the most common three letter combinations such as *the*, *ing*, *and*, and *ion*

3.4.3 Play fair ciphers

The best-known multiple-letter encryption cipher is the Playfair, which treats digramsin the plaintext as single units and translates these units into ciphertextdigrams.The Playfair algorithm is based on the use of a 5 × 5 matrix of letters constructedusing a keyword.

There will be no single translation for each letter of the alphabet. This means, we couldn't decide that every *k* will be turned into *X*.

- ✓ Here a 5 ×5 matrix is created with characters of English alphabet.
- ✓ First a keyword is chosen.
- ✓ Imagine if the keyword is "*Keyword*" (which has no repetition in letters).
- ✓ Now write the letters of that word in the first squares of a five by five matrix

K	E	Y	W	O
R	D			

- ✓ Remaining cells are filled by alphabets (not already entered) in order.
- ✓ As the number of characters is 26 which is one greater than the number of cells, one cell will have two letters. A single cell will have I, J.

K	E	Y	W	O
R	D	A	B	C
F	G	H	IJ	L
M	N	P	Q	S
T	U	V	X	Z

Plaintext is encrypted twoletters at a time, according to the following rules:

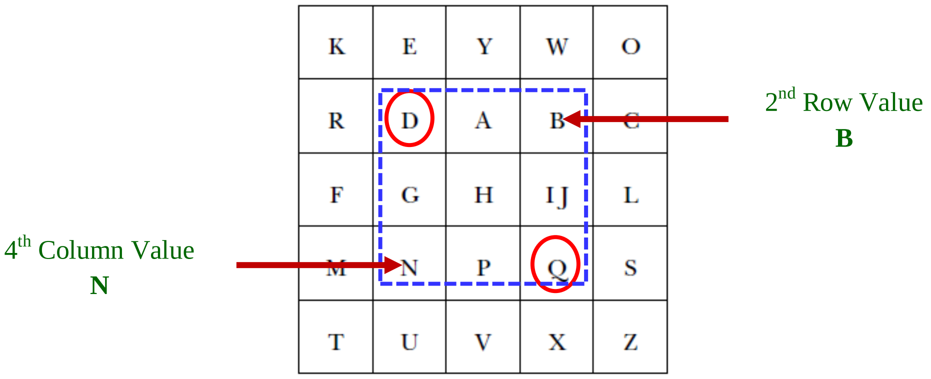
i. ***Repeating plaintext letters*** that are in the same pair are separated with a filler letter, such as *x*, so that ***balloon*** would be treated as ***ba lxlo on***.

ii. Two ***plaintext letters that fall in the same row*** of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ***wo*** is encrypted as ***OK***.

iii. Two ***plaintext letters that fall in the same column*** are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, ***kt*** is encrypted as ***RK***.

iv. ***Otherwise***, each plaintext letter in a pair is *replaced by the letter that lies in its own row and the column* occupied by the other plaintext letter.

Thus, *dq* becomes *BN*



K	E	Y	W	O
R	D	A	B	C
F	G	H	IJ	L
M	N	P	Q	S
T	U	V	X	Z

and*fw* becomes *IK* (or *JK*, as the encipherer wishes).

For example, the sentence “Why, don’t you?” becomes

Plaintext : WH YD ON TY OU

Ciphertext : YI EA ES VK EZ

3.4.4 Polyalphabetic Ciphers

Another way to improve on the simple monoalphabetic technique is to use differentmonoalphabetic substitutions as one proceeds through the plaintext message. Thegeneral name for this approach is **polyalphabetic substitution** cipher. All these techniqueshave the following features in common:

i. A set of related monoalphabetic substitution rules is used.

ii. A key determines which particular rule is chosen for a given transformation.

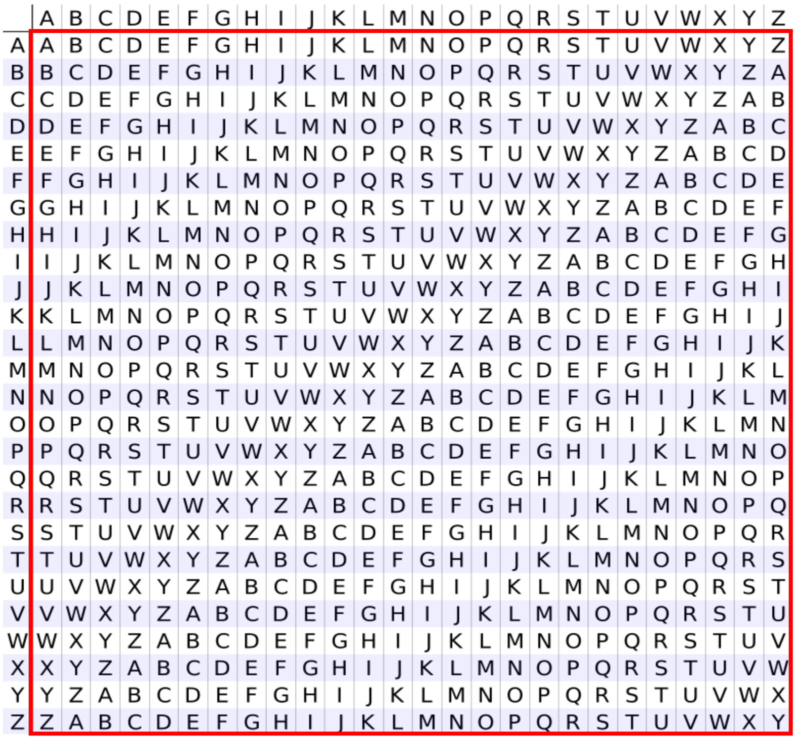
In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in plaintext and a character in ciphertext is one-to-many. Polyalphabetic ciphers have the advantage of hiding the letter frequency of the underlying language. So our key should have stream of subkeys.

Plaintext : $\mathbf{P = P_1P_2P_3\ldots.}$
Ciphertext : $\mathbf{C= C_1C_2C_3\ldots.}$
Key : $\mathbf{K= K_1K_2K_3\ldots}$

Encryption : $\mathbf{C_i =E(P_i, K_i)}$
Decryption : $\mathbf{P_i = D(C_{i,} K_i)}$

3.4.5 Vigenere Ciphers

One interesting kind of polyalphabetic cipher was designed by Blaise de Vigenere, a sixteenth century French mathematician. For encrypting the messages, a table of alphabets can be used. This table of alphabets is termed as ***tabula recta***, ***Vigenère square*** or ***Vigenère table***.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fig 3.6.Vigenere Table

It has the alphabet written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar ciphers. At different points in the encryption process, the cipher uses a different alphabet from one of the rows. The alphabet used at each point depends on a repeating keyword.

Example

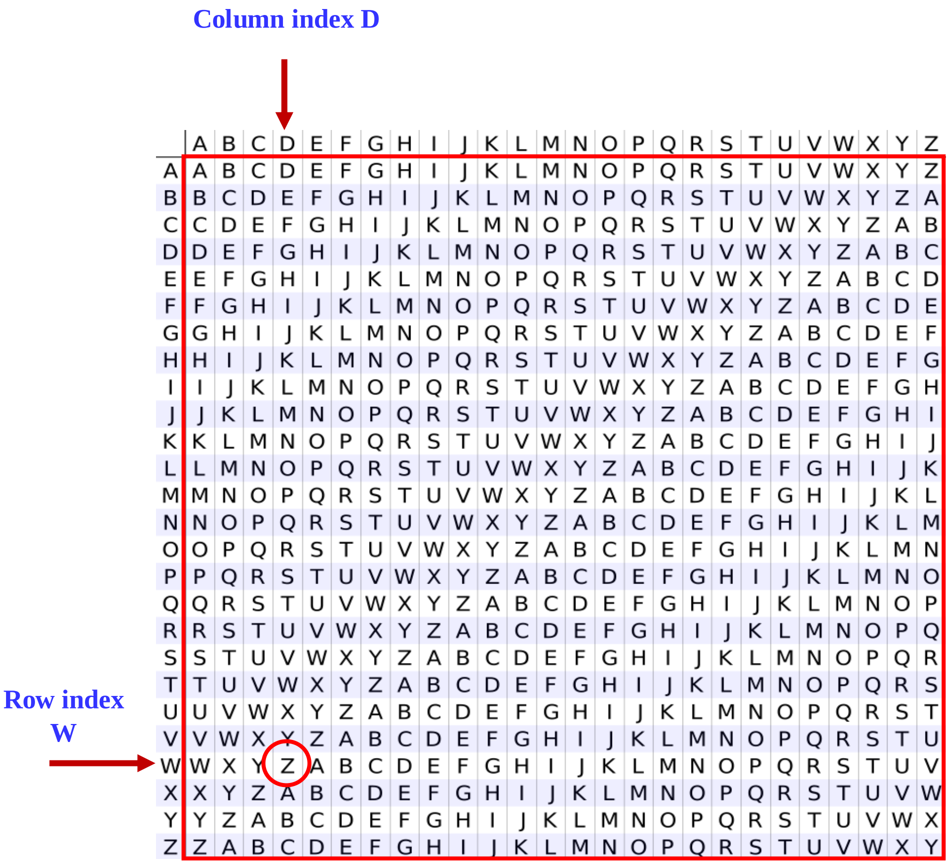
Let us choose a key word say “deceptive”. To get cipher letter copy keywords as many times and write it on top of the plain text.

Keyword :deceptivedeceptivedeceptive

Plaintext: wearediscoveredsaveyourself

For, key K_1 = d, plaintext P_1 = w, ciphertext C_1 will be computed as given below.

Here K_1 will act as the column index and P_1 will act as the row index, the ciphertext$C_1$is at the intersection of row and column index.


Column index D
Row index W

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Key: deceptivewearediscoveredsav

Plaintext: wearediscoveredsaveyourself

Ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

The strength of this cipher is that there are multiple ciphertext letters for eachplaintext letter, one for each unique letter of the keyword.Thus, the letter frequencyinformation is hidden.However not all knowledge of plain text is lost. For example the same letters “red” is coded as “VTW” twice. Attack will proceed to find same pattern in cipher text.

The periodic nature of the keyword can be eliminated by using a nonrepeatingkeyword that is as long as the message itself.Vigenère proposed what is referred toas an ***autokey system***, in which a keyword is concatenated with the plaintext itself toprovide a running key.

For the above example,

Key: deceptivewearediscoveredsav

Plaintext: wearediscoveredsaveyourself

Ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA

Even this scheme is vulnerable to cryptanalysis. Because the key and the plaintextshare the same frequency distribution of letters, a statistical technique can beapplied

3.4.6 Vernam Cipher

The ultimate defence against such a cryptanalysis is to choose akeyword that is as long as the plaintext and has no statistical relationship to it. Sucha system was introduced by an AT&T engineer named Gilbert Vernam in 1918. Hissystem works on binary data (bits) rather than letters.The ciphertext can be obtained using

$$c_i = p_i \oplus k_i$$

where,

$p_i = i^{th}$ binary digit of plaintext
$k_i = i^{th}$ binary digit of key
$c_i = i^{th}$ binary digit of ciphertext
$\oplus$ = exclusive-or (XOR) operation

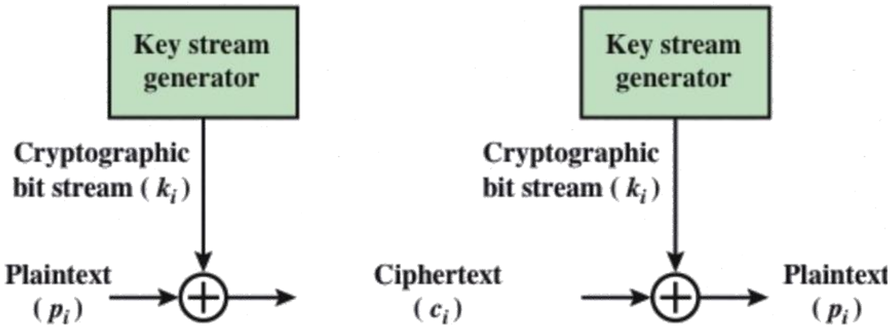


Fig 3.7.Vernam Cipher

Thus, the ciphertext is generated by performing the bitwise XOR of the plaintext and the key. Because of the properties of the XOR, decryption simply involves the same bitwise operation.

NOTES

$$p_i = c_i \oplus k_i$$

Encyption:
P = 01101110
K = 11011001
C = 10110111

Decryption:
C = 10110111
K = 11011001
P = 01101110

3.4.7 One-Time Pad

An Army Signal Corp officer, Joseph Mauborgne, proposed an improvement to theVernam cipher that yields the ultimate in security. Mauborgne suggested ***using a random key that is as long as the message***, so that the key need not be repeated. Inaddition, ***the key is to be used to encrypt and decrypt a single message, and then is discarded***.

Each new message requires a new key of the same length as the new message.Such a scheme, known as a one-time pad, is unbreakable. It produces randomoutput that bears no statistical relationship to the plaintext. Because the ciphertextcontains no information whatsoever about the plaintext, there is simply no way tobreak the code.

The security of the one-time pad is entirely due to the randomness of the key.If the stream of characters that constitute the key is truly random, then the stream ofcharacters that constitute the ciphertext will be truly random. Thus, there are nopatterns or regularities that a cryptanalyst can use to attack the ciphertext.

In theory, we need look no further for a cipher.The one-time pad offers completesecurity but, in practice, has two fundamental difficulties:

i. There is the practical ***problem ofmaking large quantities of random keys***.Anyheavily used system might require millions of random characters on a regularbasis. Supplying truly random characters in this volume is a significant task.
ii. Even more frightening is the ***problem of key distribution and protection***. Forevery message to be sent, a key of equal length is needed by both sender andreceiver.Thus, a huge key distribution problem exists.

Because of these difficulties, the one-time pad is of limited utility and is usefulprimarily for low-bandwidth channels requiring very high security.

The one-time pad is the only cryptosystem that exhibits what is referred to as***PERFECT SECRECY***.

3.5 TRANSPOSITION CIPHERS

All the techniques examined so far involve the substitution of a ciphertext symbolfor a plaintext symbol. A very different kind of mapping is achieved by performingsome sort of permutation on the plaintext letters. This technique is referred to as atransposition cipher. Transposition ciphers simply reorder the plaintext.

3.5.1 Rail Fence Ciphers

The Rail Fence Cipher is a transposition cipher. It rearranges the plaintext letters by drawing them in a way that they form a shape of the rails of an imaginary fence. The Rail Fence Cipher was invented in ancient times. It was used by the Greeks, who created a special tool, called scytale, to make message encryption and decryption easier.

To encrypt the message, the letters should be written in a zigzag pattern, going downwards and upwards between the levels of the top and bottom imaginary rails. The shape that is formed by the letters is similar to the shape of the top edge of the rail fence.

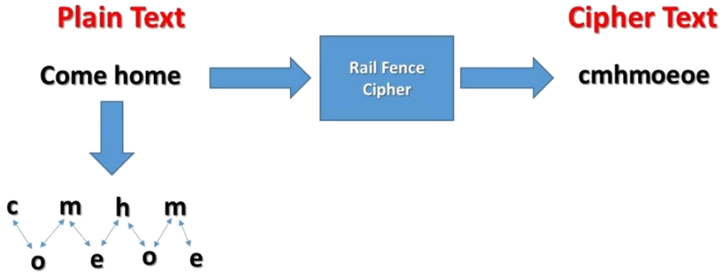


Fig 3.8. Rail Fence Cipher

Next, all the letters should be read off and concatenated, to produce one line of ciphertext. The letters should be read in rows, usually from the top row down to the bottom one.

Plaintext T H I S I S A S E C R E T M E S S A G E

Rail Fence Encoding ***key = 3***

T				I				E				T				S			
	H		S		S		S		C		E		M		S		A		E
		I				A				R				E				G	

Ciphertext T I E T S H S S S C E M S A E I A R E G

Plaintext T H I S I S A S E C R E T M E S S A G E

Rail Fence Encoding ***key = 4***

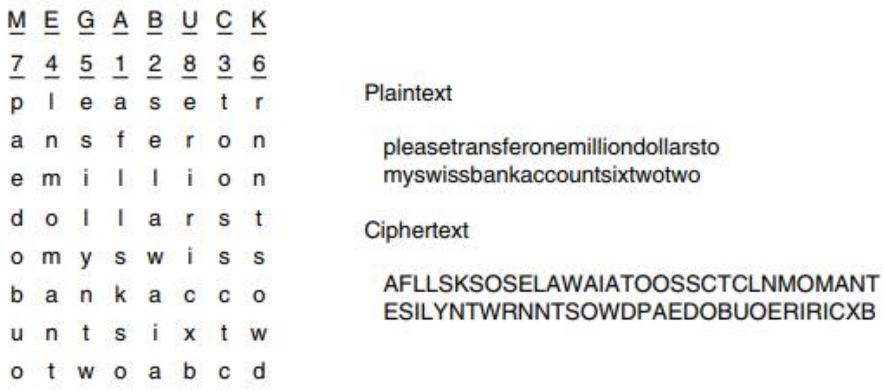
T						A						T						G	
	H				S		S				E		M				A		E
		I		I				E		R				E		S			
			S						C						S				

Ciphertext T A T G H S S E M A E I I E R E S S C S

NOTES

3.5.2 Columnar Transposition Ciphers

The columnar transposition cipher is a fairly simple, easy to implement cipher.The cipher is keyed by a word or phrase not containing any repeated letters.



M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

In this example, MEGABUCK is the key. The purpose of the key is to order the columns, with column 1 being under the key letter closest to the start of the alphabet, and so on. The plaintext is written horizontally, in rows, padded to fill the matrix if need be. The ciphertext is read out by columns, starting with the column whose key letter is the lowest.

To break a transposition cipher, the cryptanalyst must first be aware that he is dealing with a transposition cipher. By looking at the frequency of E, T, A, O, I, N, etc., it is easy to see if they fit the normal pattern for plaintext. If so, the cipher is clearly a transposition cipher, because in such a cipher every letter represents itself, keeping the frequency distribution intact. The next step is to make a guess at the number of columns. After that the cryptanalyst will look for digrams and trigrams.

Check Your Progress 2

6. What is permutation?
7. What are digrams and trigrams?
8. What do you mean by autokey system?
9. What are the practical difficulties arise while using one-time pads?
10. State the difference between substitution and transposition.

3.6 ANSWERS TO CHECK YOUR PROGRESS

1. The two requirements for the secure user of conventional encryption are
   - Strong encryption algorithm
   - Secure transmission of key

NOTES

2. 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 use different keys, the system is referred to asasymmetric, two-key, or public-key encryption

3. 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.

4. Different types of attack on encrypted messages are
   - Ciphertext Only
   - Known Plaintext
   - Chosen Plaintext
   - Chosen Ciphertext
   - Chosen Text

5. An encryption algorithm is said to be computationally secure if it meets 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.

6. A permutation of a finite set of elements*S*is an ordered sequence of all the elements of *S*, with each element appearing exactly once

7. Digrams are two letter combinations and Trigrams are three letter combinations

8. An autokey system is a system in which a keyword is concatenated with the plaintext itself toprovide a running key.

9. The practical difficulties of using one-time pads are generation of large quantities of random keys and problem of key distribution & protection.

10. The difference between substitution and transposition is given below
    - Substitution techniques map plaintext elements (characters, bits) into ciphertext elements.
    - Transposition techniques systematically transpose the positions of plaintext elements

3.7 SUMMARY

The classical encryption techniques are discussed in this unit. The symmetric encryption model is explained here. The types of encryption techniques are described. The methods of assessing the strength of the cryptographic algorithms are also discussed here. The working of different substitution as well as transposition ciphers are explained with suitable examples in this unit.

3.8 KEYWORDS

NOTES

- **Symmetric encryption** is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventionalencryption.
- Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryptionalgorithm, the plaintext is recovered from the ciphertext.
- The **two types of attack** on an encryption algorithm are cryptanalysis, based on properties of the encryption algorithm, and brute-force, which involves trying all possible keys.
- Traditional (precomputer) symmetric ciphers use **substitution** and/or **transposition** techniques.
- **Substitut**ion techniques map plaintext elements (characters, bits) into ciphertext elements.
- **Transposition** techniques systematicallytranspose the positions of plaintext elements.

3.9 SELF-ASSESSMENT EXERCISES

Short Questions

1. What is the difference between symmetric and asymmetric encryption?
2. What do you mean by cryptanalysis?
3. What is Brute-force attacks?
4. Discuss about play fair ciphers.
5. Write a note on Vernam Ciphers.

Detail Questions

1. Explain the symmetric cipher model.
2. How will you categorize the cryptographic systems? Discuss.
3. Discuss about various attacks on encrypted messages.
4. Describe the working of Caesar ciphers.
5. Discuss about monoalphabetic and polyalphabetic ciphers.
6. Write a note on one-time pads.
7. Discuss about the transposition ciphers.

3.10SUGGESTED READINGS

1. William Stallings, "Cryptography and Network Security Principles and Practice", Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, "Computer Networks", 5th Edition, Prentice Hall
4. http://www.math.stonybrook.edu/~scott/papers/MSTP/crypto/3Caesar_Cipher.html.

BLOCK – II
BLOCK CIPHERS AND DES

UNIT- 4 DATA ENCRYPTION STANDARD

Structure

4.0 INTRODUCTION

The objective of this section is to introduce the fundamental principles of modern symmetric ciphers. Many symmetric block encryption algorithms in current use are based on a structurereferred to as a Feistel block cipher. For that reason, it is important toexamine the design principles of the Feistel cipher.The Data Encryption Standard (DES) has been the most widely usedencryption algorithm until recently. Although numerous symmetric ciphers have beendeveloped, since the introduction of DES, and although it is destined to be replaced bythe Advanced Encryption Standard (AES), DES remains the most important suchalgorithm. Furthermore, a detailed study of DES provides an understanding of theprinciples used in other symmetric ciphers.

4.1 OBJECTIVES

After going through this unit, you will be able to:

- To illustrate the principles of modern symmetric ciphers
- Understand the working of DES
- Access the strength of DES

NOTES

4.2 BLOCK CIPHER PRINCIPLES

We begin with a comparison ofstream ciphers and block ciphers. Then we discuss the motivation for the Feistelblock cipher structure.
A ***stream cipher*** is one that encrypts a digital data stream one bit or one byte at a time.Examples of classical stream ciphers are the auto keyed Vigenère cipher and the Vernam cipher.

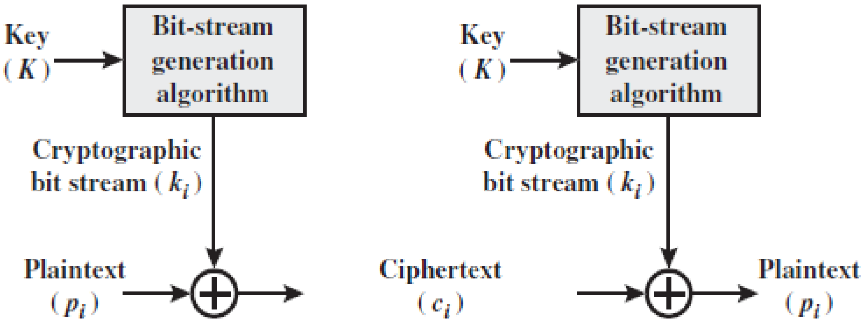

Key
(K)
Bit-stream generation algorithm
Cryptographic bit stream (k_i)
Plaintext
(p_i)
Key
(K)
Bit-stream generation algorithm
Cryptographic bit stream (k_i)
Ciphertext
(c_i)
Plaintext
(p_i)

Fig 4.1. Stream Cipher

The bit-stream generator is akey-controlled algorithm and must produce a bit stream that is cryptographicallystrong.Now, the two users need only share the generating key, and each can producethe keystream.

A ***block cipher*** is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically, a block size of 64 or 128 bits is used. As with a stream cipher, the two users share a symmetric encryptionkey.

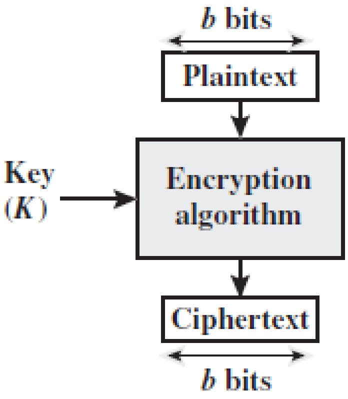


Fig 4.2. Block Cipher

4.2.1 Motivation for the Feistel Cipher Structure

A block cipher operates on a plaintext block of *n* bits to produce a ciphertextblock of *n* bits. There are 2^npossible different plaintext blocks and, forthe encryption to be reversible (i.e., for decryption to be possible), each mustproduce a unique ciphertext block. Such a transformation is called reversible, ornonsingular.The following examples illustrate nonsingular and singular transformationsfor *n = 2*.

Reversible Mapping		Irreversible Mapping	
Plaintext	**Ciphertext**	**Plaintext**	**Ciphertext**
00	11	00	11
01	10	01	10
10	00	10	01
11	01	11	01

In the latter case, a ciphertext of 01 could have been produced by one of two plaintextblocks. Obliviously it comes under the irreversible mapping.

The following figure 4.3 illustrates the logic of a general substitution cipher for *n = 4*.A 4-bit input produces one of 16 possible input states, which is mapped by thesubstitution cipher into a unique one of 16 possible output states, each of which isrepresented by 4 ciphertext bits.

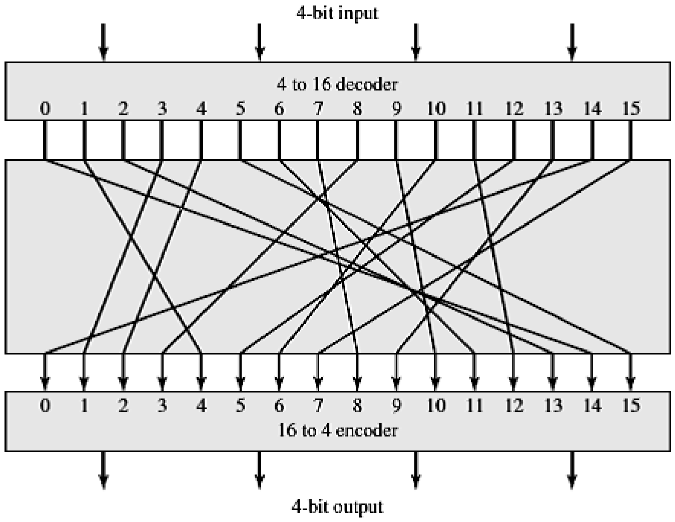
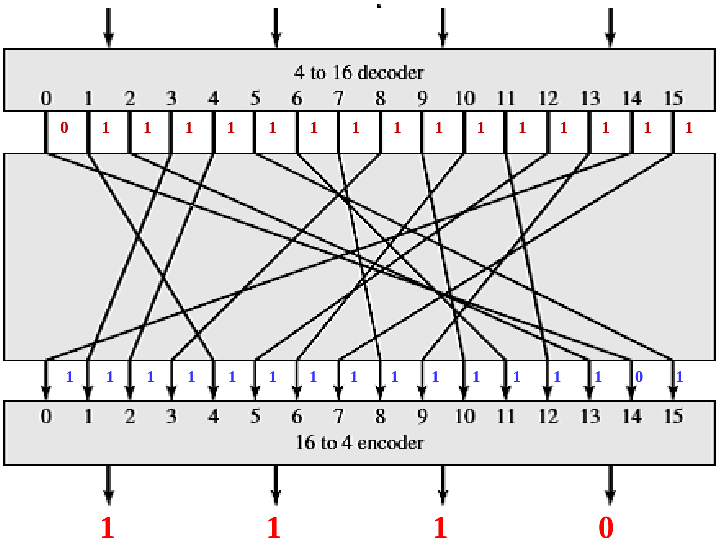


Fig 4.3.General n-bit-n-bit Block Substitution (shown with n = 4)

Let us consider the input as 0000, the process of block substitution as per figure 4.3 is performed as given below.

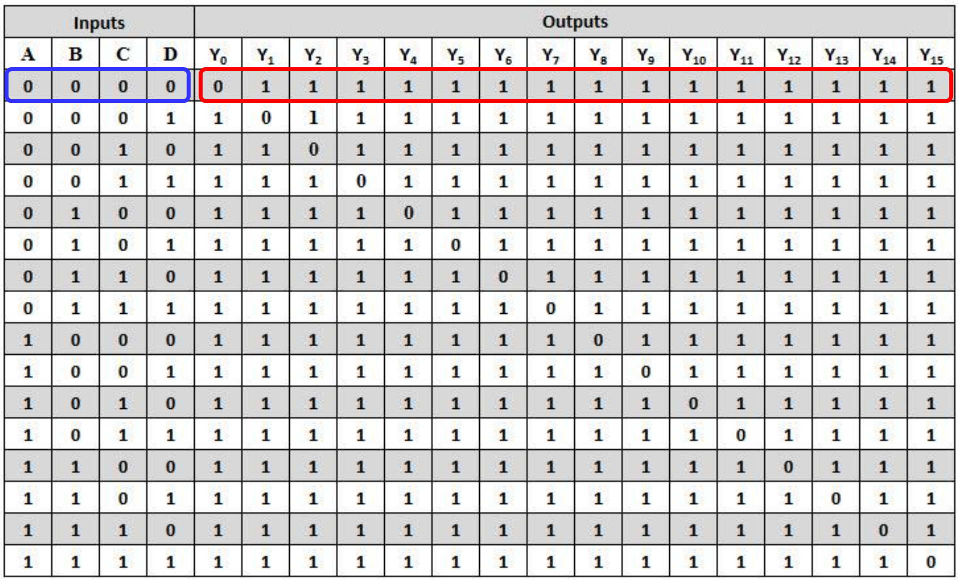
0 0 0 0

NOTES



The truth tables of the 4 to 16 decoder and 16 to 4 encoder are given below for reference.

Table4.1. Truth Table of 4 to 16 Decoder

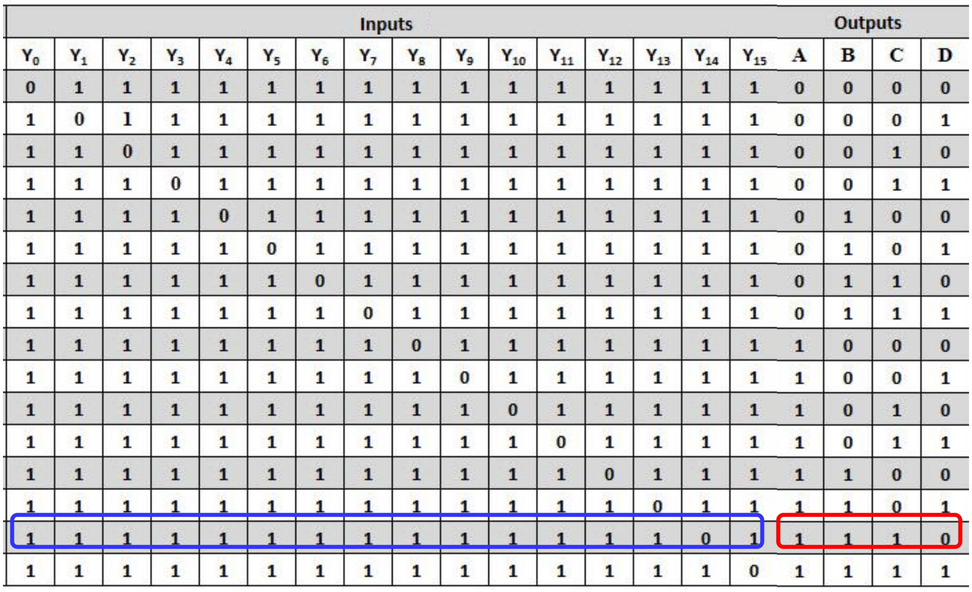


Inputs				Outputs															
A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

The input 0000 of 4 to 16 decoder is converted to 0111 1111 1111 1111 using truth table 4.1.

Table4.2. Truth Table of 16 to 4 Encoder

NOTES

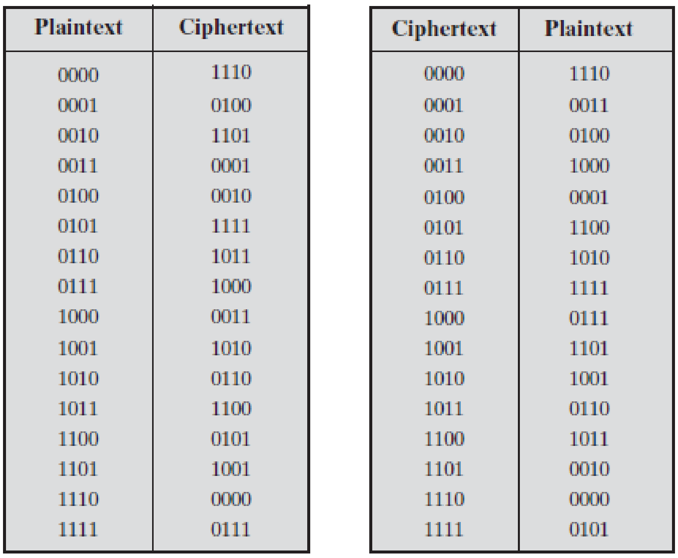


Inputs																Outputs			
Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	A	B	C	D
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	1
1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1

The input 1111 1111 1111 1101of 16 to 4 encoder is converted to 1110 using truth table 4.2.

The encryption and decryption mappings can bedefined by tabulation, as shown in Table 4.3.This is the most general form of blockcipher and can be used to define any reversible mapping between plaintext andciphertext.Feistel refers to this as the ***ideal block cipher***, because it allows for the maximumnumber of possible encryption mappings from the plaintext block.

Table4.3. Encryption and Decryption tables for Substitution Cipher of Fig 4.3



Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

But there is a practical problem with the ideal block cipher. Such systems are vulnerable to a statistical analysis of the plaintext. If *n* (block size) is sufficiently large and an arbitrary reversible substitution between plaintext and ciphertext is allowed, then the statistical characteristics of the source plaintext are masked such that this type of cryptanalysis is infeasible. An arbitrary reversible substitution cipher (the ideal block cipher) for a large block size is not practical, however, from an implementation and performance point of view.

In considering these difficulties, Feistel points out that what is needed is an approximation to the ideal block cipher system for large *n*, built up out of components that are easily realizable.

4.2.2 The Feistel Cipher

The Feistel cipher or Feistel Network is named after Horst Feistel, who developed it while working at IBM. He and a colleague, Don Coppersmith, published a cipher called Lucifer in 1973 that was the first public example of a cipher using a Feistel structure. Due to the benefits of the Feistel structure, other encryption algorithms based upon the structure and upon Lucifer have been created and adopted for common use.

Note:

> *Please don't be confused by the name Feistel cipher. Feistel cipher is not one particular cipher.*
> *It is a structure on which many ciphers such as the Lucifer cipher are based.*

Feistel proposed that we can approximate the ideal block cipher by utilizing the concept of a product cipher, which is the execution of two or more simple ciphers in sequence in such a way that the final result or product is cryptographically stronger than any of the component ciphers. The essence of the approach is to *develop a block cipher with a key length of k bits and a block length of n bits, allowing a total of* 2^k *possible transformations*, rather than the $2^n!$transformations available with the ideal block cipher. In particular, Feistel proposed the use of a cipher that alternates substitutions and permutations.

A Feistel cipher is a multi-round cipher that divides the current internal state of the cipher into two parts and operates only on a single part in each round of encryption or decryption. Between rounds, the left and right sides of the internal states switch sides.

Diffusion and Confusion

The terms *diffusion* and *confusion* were introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system. In Shannon's original definitions:

1. ***Confusion*** refers to making the relationship between the key and the ciphertext as complex and as involved as possible
2. ***Diffusion*** refers to the property that redundancy in the statistics of the plaintext is "dissipated" in the statistics of the ciphertext.

Feistel Cipher Structure

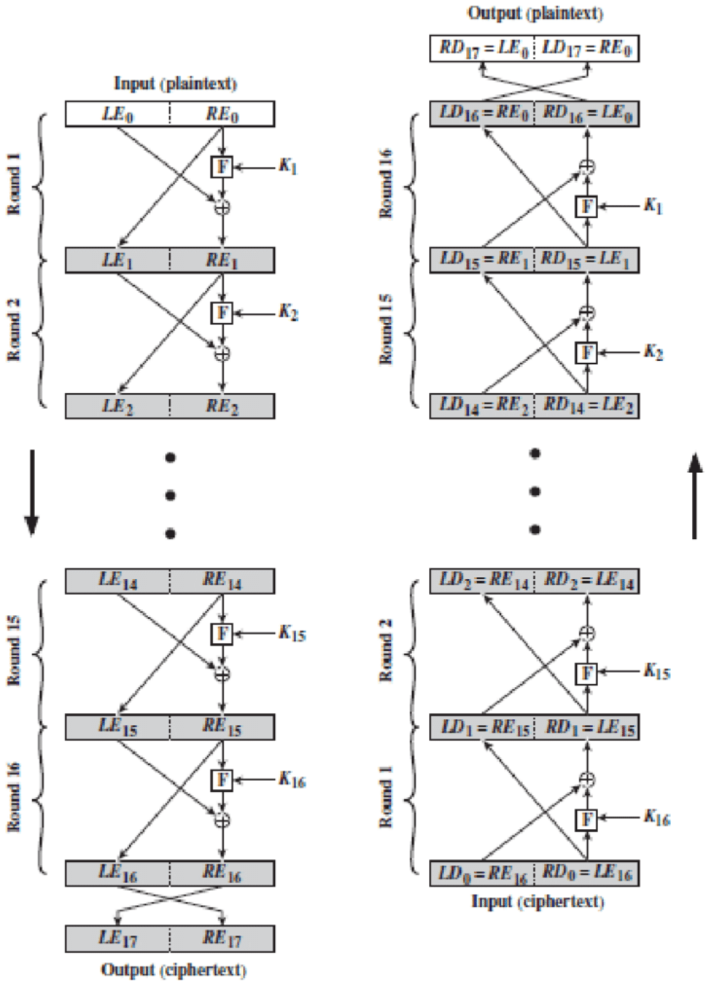


Fig 4.4.Fiestal Encryption and Decryption (16 rounds)

The left-hand side of Figure 4.4 depicts the structureproposed by Feistel.The inputs to the encryption algorithm are a plaintext block of length$2w$ bits and a key K. The plaintext block is divided into two halves, L_0 and R_0. The two halves of the data pass through rounds of processing and then combine to produce the ciphertext block. Each round has as inputs derived from L_{i-1} and R_{i-1} the previous round, as well as a subkey$K_i$derived from the overall K. In general, the subkeys$K_i$are different K from and from each other.

In the figure, 16 rounds are used, although any number of rounds could be implemented. 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 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 subkeyK_i. Following this substitution, a **permutation** is performed that consists of the interchange of the two halves of the data. This structure is a particular form of the substitution-permutation network (*SPN*) proposed by Shannon.

The exact realization of a Feistel network depends on the choice of the following parameters and design features:

NOTES

- ***Block size:***Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion. Traditionally, a block size of 64 bits has been considered a reasonable trade off and was nearly universal in block cipher design. However, the new AES uses a 128-bit block size.
- ***Key size:***Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion. Key sizes of 64 bits or less are now widely considered being inadequate and 128 bits has become a common size.
- ***Number of rounds:***The essence of the Feistel 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 F:***Again, greater complexity generally means greater resistance to cryptanalysis.

There are two other considerations in the design of a Feistel 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.

Check Your Progress 1

1. What is stream cipher?
2. What do you mean by block cipher?
3. When a block cipher is considered as ideal block cipher?
4. What is the difference between confusion and diffusion process?
5. State the parameters considered during Fiestel design.

4.3 THE DATA ENCRYPTION STANDARD

The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now called the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard. The algorithm itself is referred to as the Data Encryption Algorithm. For DES, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.

The DES enjoys widespread use. It has also been the subject of much controversy concerning how secure the DES is. To appreciate the nature of the controversy, let us quickly review the history of the DES.

Table4.4. Evolution of DES

Year	Event
1960	IBM‟ s project in cryptography led by Horst Feistel
1971	• Development of an algorithm LUCIFER, which was sold to Lloyd's of London for use in a cash-dispensing system, also developed by IBM • LUCIFER is a Feistel block cipher that operates on blocks of 64 bits, using a key size of 128 bits • Outcome of this effort was a refined version of LUCIFER that was more resistant to cryptanalysis but that had a reduced key size of 56 bits, in order to fit on a single chip
1973	NBS invited proposals for a national cipher standard
1975	DES is published in the Federal Register for comment
1977	DES is published as a standard
1994	NIST extended use of DES federal system for 5 years
1999	NIST recommended Triple DES for federal use

DES Encryption

The overall scheme for DES encryption is illustrated in Figure 4.5. As with anyencryption scheme, there are two inputs to the encryption function: the plaintext tobe encrypted and the key. In this case, the plaintext must be 64 bits in length and thekey is 56 bits in length.

NOTES

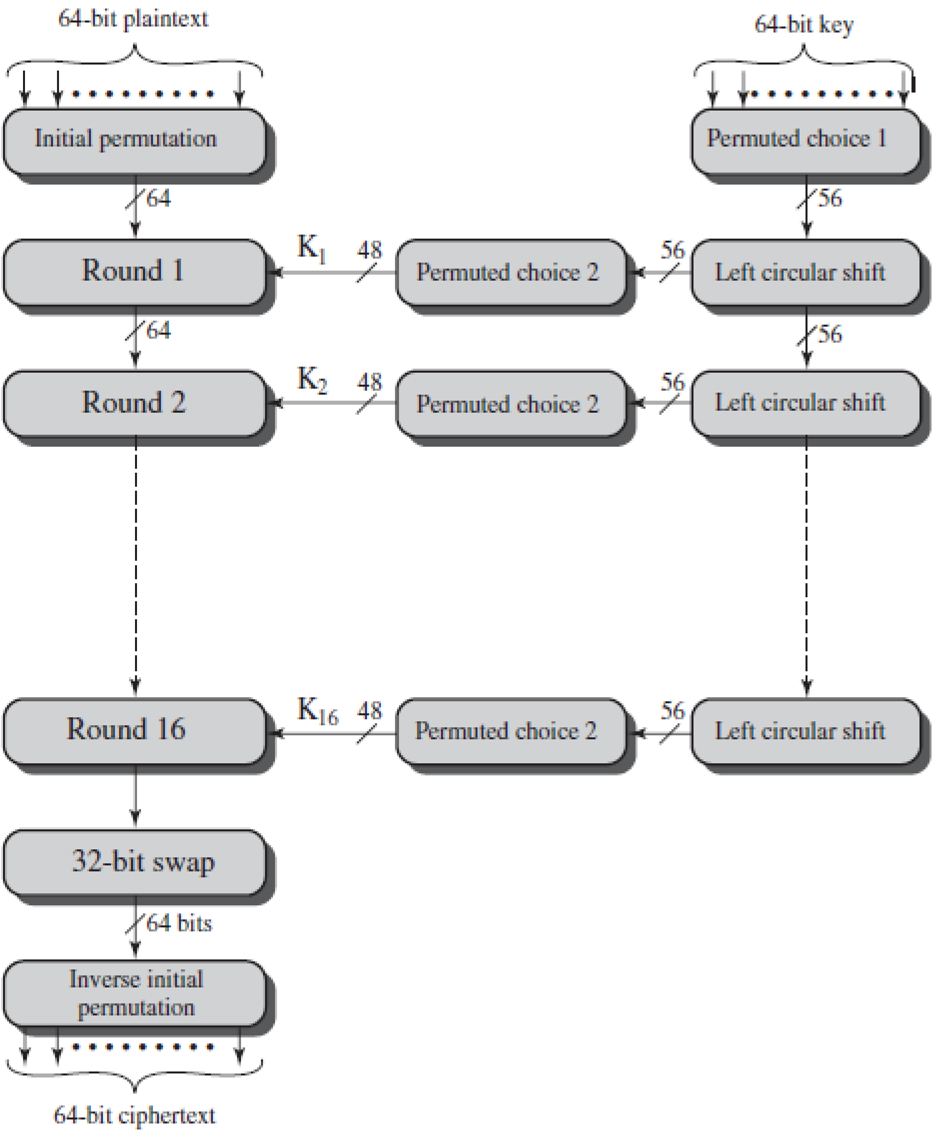


Fig 4.5.General Depiction of DES Encryption Algorithm

Looking at the **left-hand side** of the figure, we can see that the *processing of the plaintext* proceeds in three phases.

1. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*.
2. This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the **preoutput**.
3. Finally, the preoutput is passed through a permutation that is the inverse of the initial permutation function, to produce the 64-bit cipher text. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher.

The **right-hand** portion of figure shows the *way in which the 56-bit key is used*.

1. Initially, the key is passed through a permutation function.

2. Then, for each of the sixteen rounds, a subkey (K_i) is produced by the combination of a left circular shift and a permutation.
3. The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

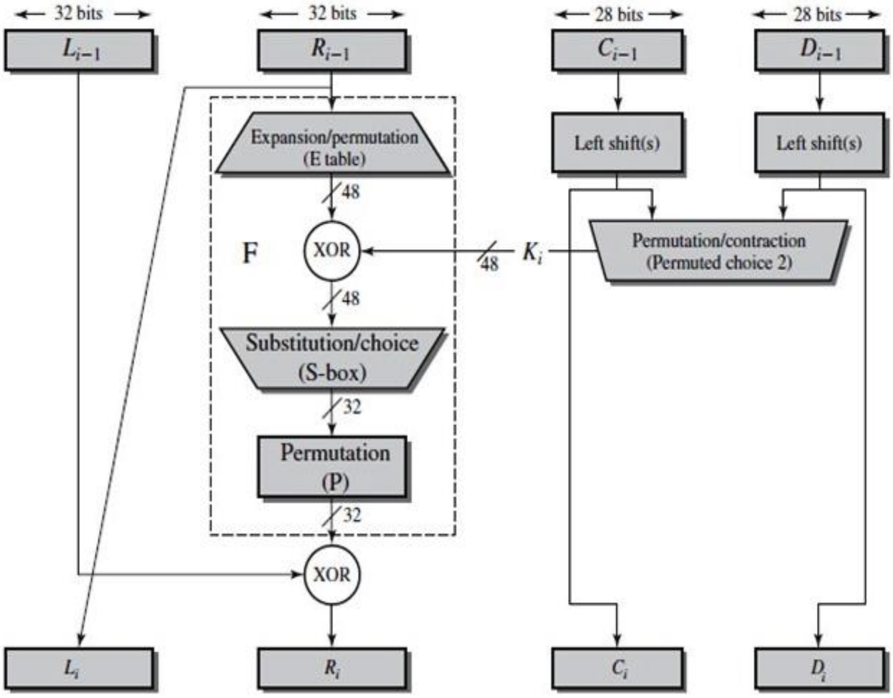


Fig 4.6.Single Round of DES Algorithm

Figure 4.6 shows the internal structure of a single round.Again, begin by focusing on the left-hand side of the diagram. The left and righthalves of each 64-bit intermediate value are treated as separate32-bit quantities, labeled L (left) and R (right). As in any classic Feistel cipher, theoverall processing at each round can be summarized in the following formulas:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

The round key K_i is 48 bits. The$R$ input is 32 bits. This Rinput is firstexpanded to 48 bits by using a table that defines a permutation plus an expansionthat involves duplication of 16 of the Rbits.The resulting 48 bits areXORed with K_i. This 48-bit result passes through a substitution function thatproduces a 32-bit output. The role of the S-boxes in the function F is illustrated in Figure 4.7.

NOTES

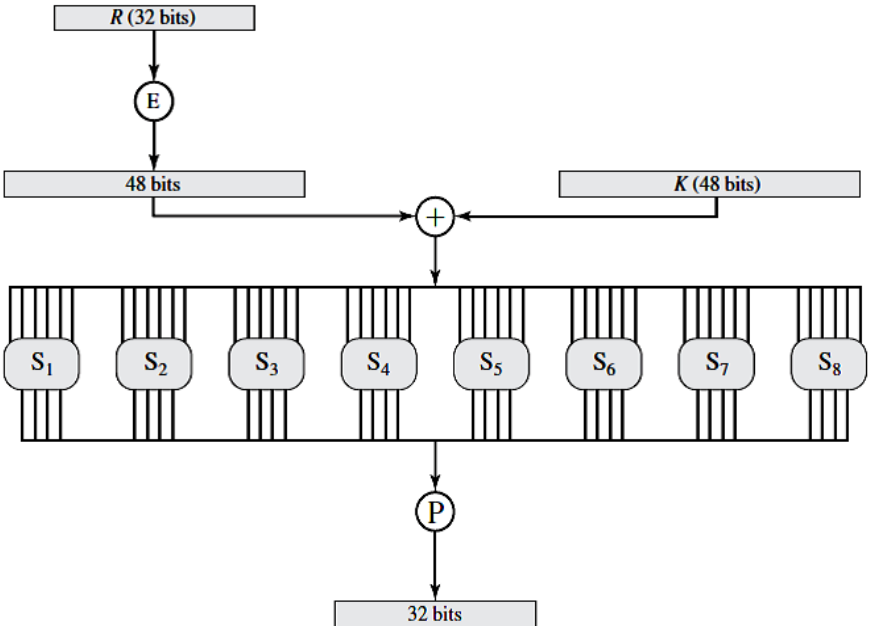


Fig 4.7.Calculation of F(R, K)

The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.The first and last bits of the input to box $S_i$form a 2-bit binarynumber to select one of four substitutions defined by the four rows in the table for S_i.The middle four bits select one of the sixteen columns.The decimal value in the cellselected by the row and column is then converted to its 4-bit representation to producethe output.The 32-bit output from the eight S-boxes is thenpermuted, so that on the next round, the output from each S-box immediatelyaffects as many others as possible.

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipherkey. However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual keygeneration process, as shown in Figure4.8. The steps involved in key generation are depicted in Figure 4.9.

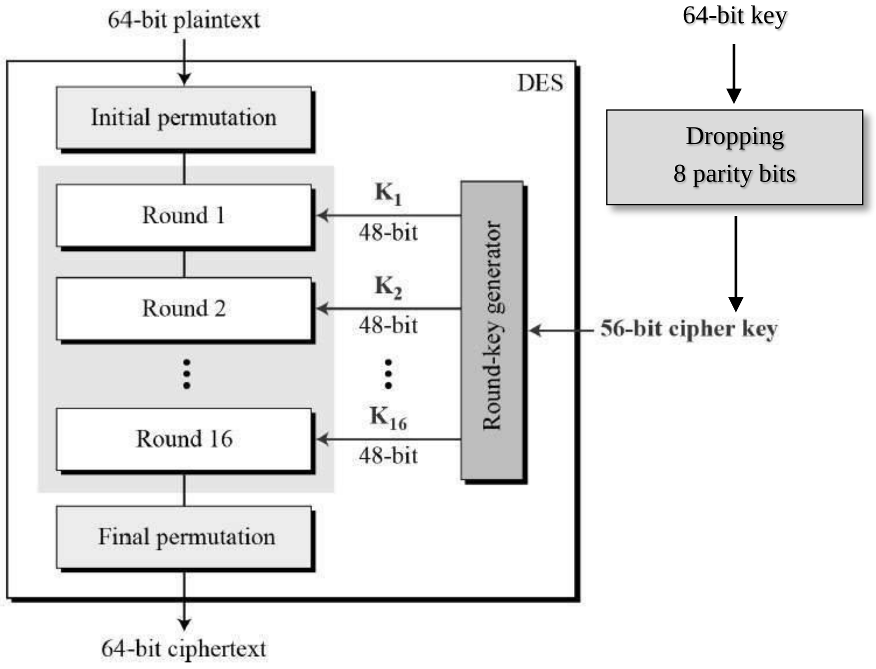
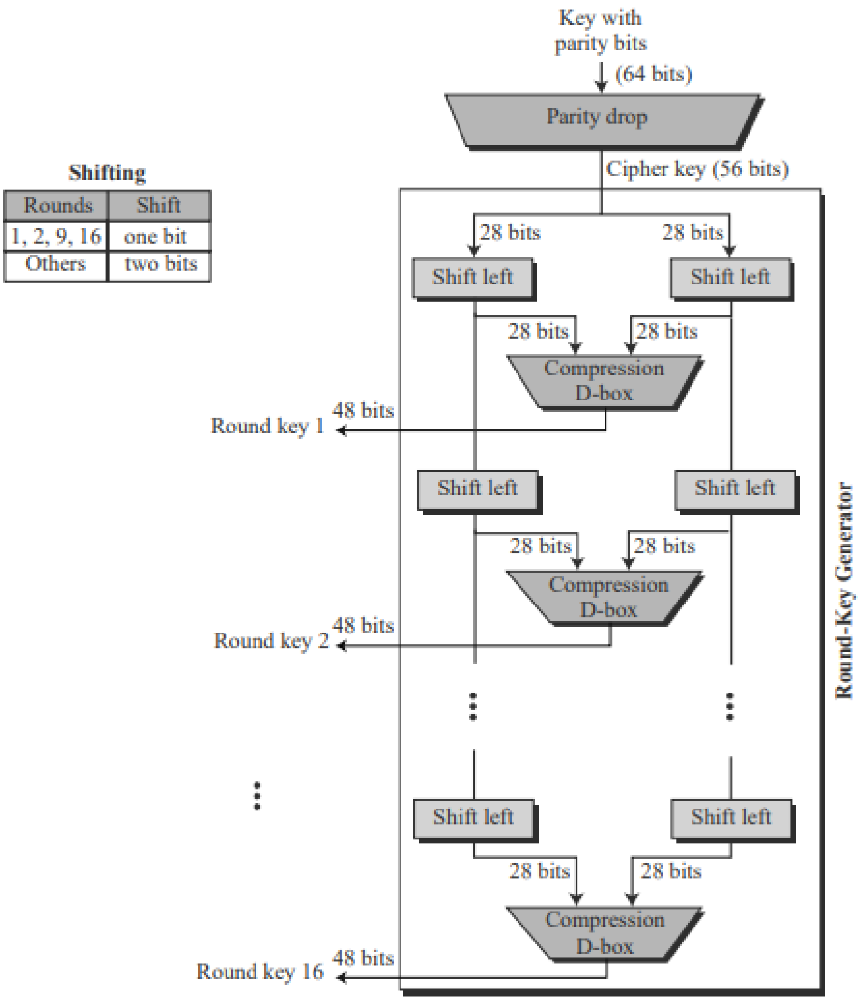


Fig 4.8.Simplified Block diagram of DES



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

Fig 4.9.Key Generation

NOTES

The operations involved in the key generation are explained below

Parity Drop

It drops the parity bits (bits 8, 16, 24, 32, …, 64) from the 64-bit key and permutes the rest of the bits.

Shift Left

After the straight permutation, the key is divided into two 28-bit parts. Each part is shifted left (circular shift) one or two bits. In rounds 1, 2, 9, and 16, shifting is one bit; in the other rounds, it is two bits. The two parts are then combined to form a 56-bit part.

Compression D-box

The compression D-box changes the 58 bits to 48 bits, which are used as a key for a round.

DES Decryption

As with any Feistel cipher, decryption uses the same algorithm as encryption, exceptthat the application of the sub keys is reversed.

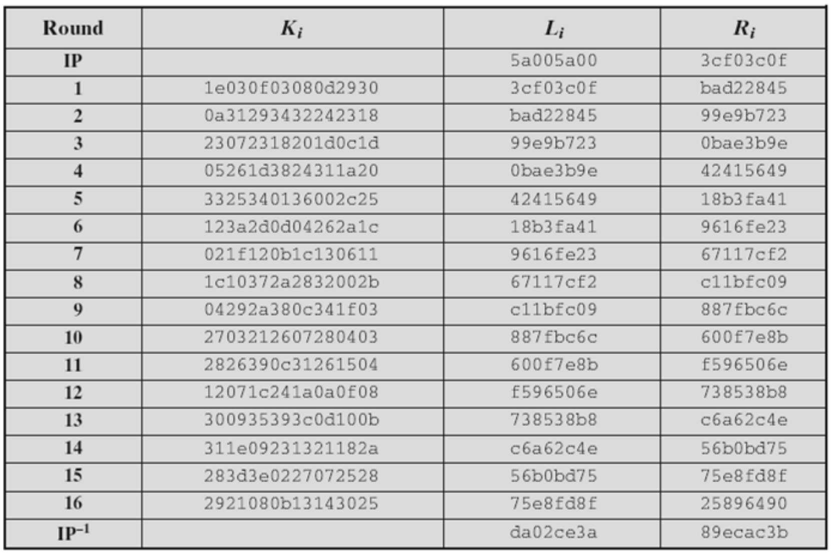
4.4 DES EXAMPLE

Let us consider the following example. Here, the plaintext is a hexadecimal palindrome.The plaintext and the key are as follows

Plaintext	:	02468aceeca86420
Key	:	0f1571c947d9e859

Table 4.5 shows the progression of the algorithm. The first row shows the 32-bitvalues of the left and right halves of data after the initial permutation. The next16 rows show the results after each round. Also shown is the value of the 48-bitsubkey generated for each round. Note that *Li* = *Ri-1*.The final row shows the leftandright-hand values after the inverse initial permutation.

Table 4.5 DES Example



Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

Note: DES subkeys are shown as eight 6-bit values in hex format

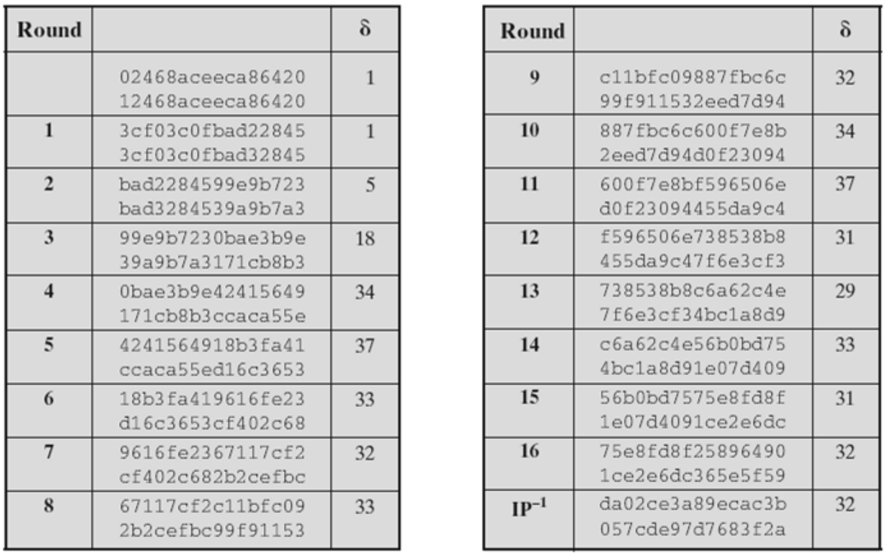
The resulting ciphertext of the above example is given below.

Plaintext	:	02468aceeca86420
Key	:	0f1571c947d9e859
Ciphertext	:	da02ce3a89ecac3b

The Avalanche Effect

A desirable property of any encryption algorithm is that a small change in either theplaintext or the key should produce a significant change in the ciphertext. In particular,a change in one bit of the plaintext or one bit of the key should produce achange in many bits of the ciphertext. This is referred to as the avalanche effect. Ifthe change were small, this might provide a way to reduce the size of the plaintext orkey space to be searched.

Table 4.6 Avalanche effect in DES : Change in Plaintext



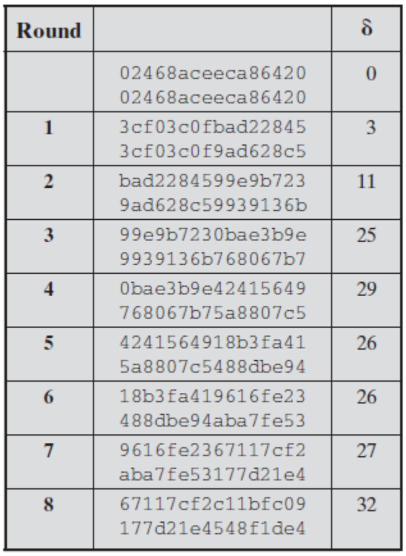
Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

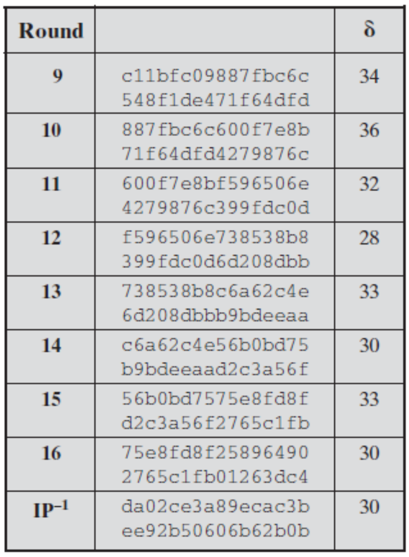
Note : δ represents the number of bits positions changed

Table 4.7 Avalanche effect in DES : Change in Key

NOTES



Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32



Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

Using the example from Table 4.5, Table 4.6 shows the result when a bit of the plaintext is changed, so that the plaintext is 12468aceeca86420. Thesecond column of the table shows the intermediate 64-bit values at the end of eachround for the two plaintexts.The third column shows the number of bits that differbetween the two intermediate values. The table shows that, after just three rounds,18 bits differ between the two blocks. On completion, the two ciphertexts differ in32 bit positions.

Table 4.7 shows a similar test using the original plaintext of with two keys thatdiffer in only one bit position: the original key, 0f1571c947d9e859, and thealtered key, 1f1571c947d9e859.
Again, the results show that about half of the bits inthe ciphertext differ and that the avalanche effect is pronounced after just a few rounds.

4.5 THE STRENGTH OF DES

Since its adoption as a federal standard, there have been stable concerns aboutthe level of security provided by DES. These concernsfall into twoareas: key size and the nature of the algorithm.

The Use of 56-Bit Keys

With a key length of 56 bits, there are 2^{56} possible keys, which is approximately7.2 x 10^{16}keys. Thus, on the face of it, a brute-force attack appears impractical.Assuming that, on average, half the key space has to be searched, a single machineperforming one DES encryption per microsecond would take more than a thousandyears to break the cipher.

The Nature of the DES Algorithm

Another concern is the possibility that cryptanalysis is possible by exploiting thecharacteristics of the DES algorithm. The focus of concern has been on the eightsubstitution tables, or S-boxes, that are used in all iteration. Because the designcriteria for these boxes, and indeed for the entire algorithm, were not made public,there is a suspicion that the boxes were constructed in such a way that cryptanalysisis possible for an opponent who knows the weaknesses in the S-boxes.This assertionis tempting, and over the years a number of regularities and unexpected behavioursof the S-boxes have been discovered. Despite this, no one has so far succeeded indiscovering the supposed fatal weaknesses in the S-boxes.

Timing Attacks

A timing attack is one in which information about the key or the plaintextis obtained by observing how long it takes a given implementation to performdecryptions on various ciphertexts.A timing attack exploits the fact that an encryptionor decryption algorithm often takes slightly different amounts of time on different inputs.

Check Your Progress 2

6. How many rounds are used in DES?
7. Write down the equation used to calculate the left and right hand side of DES steps.
8. What are the operations involved in key generation of DES?
9. What do you mean by Avalance Effect?
10. List the parameters considered to assess the strength of any encryption algorithm.

4.6 ANSWERS TO CHECK YOUR PROGRESS

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

2. 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. Typically, a block size of 64 or128 bits is used.

3. A block cipher is said to be ideal block cipher, if it allows for the maximumnumber of possible encryption mappings from the plaintext block

4. Confusion refers to making the relationship between the key and the ciphertext as complex and as involved as possible

   Diffusion refers to the property that redundancy in the statistics of the plaintext is "dissipated" in the statistics of the ciphertext

5. Parameters considered during Fiestel design are:
   - Block size
   - Key size
   - Number of rounds

- Subkey generation algorithm
- Round function F

6. In DES algorithm totally 16 rounds are used. Each round consists of different operations.

7. The left and right side are calculated using the following equations

   $L_i = R_{i-1}$

   $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

8. The operations used involved in key generation of DES are
   - Parity drop
   - Shift left
   - Compression D-box

9. Avalance effect is a desirable property of any encryption algorithm is that a small change in either theplaintext or the key should produce a significant change in the ciphertext

10. The parameters considered to assess the strength of any encryption algorithm are
    - Size of the key
    - Nature of the algorithm

4.7 SUMMARY

The principles of modern symmetric ciphers are described in this unit. One of the most widely used symmetric cipher – Data Encryption Standard (DES) is explained here. The steps involved in this encryption technique are described. A detailed study of DES is done to understand the principles of symmetric ciphers. The parameters used to assess the strength of the encryption algorithm are also discussed here. Although numerous symmetric ciphers have beendeveloped since the introduction of DES, it is destined to be replaced bythe Advanced Encryption Standard (AES), DES remains the most important suchalgorithm.

4.8 KEYWORDS

- A **block cipher** is an encryption/decryption scheme in which a block ofplaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Many block ciphers have a **Feistel structure**. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded sothat a different key is used for each round.
- **Substitution**: Each plaintext element or group of elements is uniquelyreplaced by a corresponding ciphertext element or group of elements.

- **Permutation**: A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence ischanged.
- The **Data Encryption Standard** (DES) has been the most widely used encryption algorithm until recently. It exhibits the classic Feistel structure.DES uses a 64-bit block and a 56-bit key.

4.9 SELF-ASSESSMENT EXERCISES

Short Questions

1. What is the difference between stream and block cipher?
2. What do you mean by Fiestel cipher?
3. State the importance of Substitution and Permutation.
4. Discuss about confusion and diffusion process.

Detail Questions

5. Explain in detail about the block cipher principles.
6. Describe the Fiestel cipher structure.
7. Discuss the design features of Fiestel cipher.
8. Describe the working of DES.
9. Write a note on Avalanche effect of DES.
10. Assess the strength of DES.

4.10SUGGESTED READINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall
4. Data Encryption Standard, Federal Information Processing Standard (FIPS) Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C. (January 1977)
5. https://academic.csuohio.edu/yuc/security/Chapter_06_Data_Encription_Standard.pdf

BLOCK – II
BLOCK CIPHERS AND DES

UNIT - 5 CRYPTANALYSIS AND BLOCK CIPHER DESIGNPRINCIPLES

Structure

5.0 INTRODUCTION

Although numerous symmetric ciphers have beendeveloped since the introduction of DES, and although it is destined to be replaced bythe Advanced Encryption Standard (AES), DES remains the most important suchalgorithm. Furthermore, a detailed study of DES provides an understanding of theprinciples used in other symmetric ciphers.In this section, we provide a brief overview of the two most powerful and promising approaches: differential cryptanalysis and linear cryptanalysis. In addition to that more general discussion of block cipher design is also provided here.

5.1 OBJECTIVES

After going through this unit, you will be able to:

- To understand differential and linear cryptanalysis
- Describe Cryptanalysis of the DES
- Know the block cipher design principles

NOTES

5.2 DIFFERENTIALCRYPTANALYSIS

For most of its life, the prime concern with DES has been its vulnerability to brute-force attack because of its relatively short (56 bits) key length. However, there has also been interest in finding cryptanalytic attacks on DES. With the increasing popularity of block ciphers with longer key lengths, including triple DES, brute-force attacks have become increasingly impractical. Thus, there has been increased emphasis on cryptanalytic attacks on DES and other symmetric block ciphers.

One of the most significant advances in cryptanalysis in recent years is differential cryptanalysis. In this section, we discuss the technique and its applicability to DES.

5.2.1 HISTORY

Differential cryptanalysis was not reported in the open literature until 1990. The first published effort appears to have been the cryptanalysis of a block cipher called FEAL by Murphy. This was followed by a number of papers by Biham and Shamir, who demonstrated this form of attack on a variety of encryption algorithms and hash functions.

5.2.2 DIFFERENTIALCRYPTANALYSISATTACK

The differential cryptanalysis attack is complex. The rationale behind differential cryptanalysis is to observe the behavior of pairs of text blocks evolving along each round of the cipher, instead of observing the evolution of a single text block. Here, we provide a brief overview so that you can get the flavor of the attack.

We begin with a change in notation for DES. Consider the original plaintext block *m* to consist of two halves m_0, m_1. Each round of DES maps the right-hand input into the left-hand output and sets the right-hand output to be a function of the left-hand input and the subkey for this round. So, at each round, only one new 32-bit block is created. If we label each new block m_i(2<= *i*<= 17), then the intermediate message halves are related as follows

$$m_{i+1} = m_{i-1} \oplus f(m_i K_i), \qquad i = 1,2,3 \dots\dots,16$$

In differential cryptanalysis, we start with two messages, m and $m`$ , with a known XOR difference $\Delta m = m \oplus m`$, and consider the difference between the intermediate message halves: $\Delta m_i = m_i \oplus m`_i$. Then we have

$\Delta m_{i+1} = m_{i+1} \oplus m`_{i+1}$

NOTES

The overall strategy of differential cryptanalysis is based on these considerations for a single round. The procedure is to begin with two plaintext messages *m* and *m*` with a given difference and trace through a probable pattern of differences after each round to yield a probable difference for the ciphertext.

5.3 LINEARCRYPTANALYSIS

A more recent development is linear cryptanalysis. This attack is based on finding linear approximations to describe the transformations performed in DES. This method can find a DES key given 2^{43} known plaintexts, as compared to 2^{47} chosen plaintexts for differential cryptanalysis. Although this is a minor improvement, because it may be easier to acquire known plaintext rather than chosen plaintext, it still leaves linear cryptanalysis infeasible as an attack on DES. So far, little work has been done by other groups to validate the linear cryptanalytic approach.

We now give a brief summary of the principle on which linear cryptanalysis is based. For a cipher with *n*-bit plaintext and ciphertext blocks and an *m*-bit key, let the plaintext block be labeled P[1],.....P[*n*] , the cipher text block C[1],......C[*n*] ,and the key K[1], ...K[*m*]. Then define

$$A[i,j,....,k] = A[i] \oplus A[j] \oplus ... \oplus A[k]$$

The objective of linear cryptanalysis is to find an effective linear equation of the form:

$$P[\alpha_1, \alpha_2, ... , \alpha_a] \oplus C[\beta_1, \beta_2, ... , \beta_b] = K[\gamma_1, \gamma_2, ... , \gamma_c]$$

(where*x=0 or 1; 1<=a; b<=n; c<=m*; and where the α, β, and γ terms represent fixed, unique bit locations) that holds with probability $p \neq 0.5$. The further *p* is from 0.5, the more effective the equation. Once a proposed relation is determined, the procedure is to compute the results of the left-hand side of the preceding equation for a large number of plaintext–ciphertext pairs. If the result is 0 more than half the time, assume $K[\gamma_1, \gamma_2,..... \gamma_c] = 0$. If it is 1 most of the time, assume $K[\gamma_1, \gamma_2,..... \gamma_c] = 1$. This gives us a linear equation on the key bits. Try to get more such relations so that we can solve for the key bits. Because we are dealing with linear equations, the problem can be approached one round of the cipher at a time, with the results combined

5.4 BLOCKCIPHERDESIGNPRINCIPLES

Although much progress has been made in designing block ciphers that are cryptographically strong, the basic principles have not changed all that much since the work of Feistel and the DES design team in the early 1970s. It is useful to begin this discussion by looking at the published design criteria used in the DES effort. Then we look at three critical aspects of block cipher design: the number of rounds, design of the function F, and key scheduling.

DES Design Criteria

The criteria used in the design of DES, as reported in [COPP94], focused on the design of the S-boxes and on the P function that takes the output of the S-boxes.

The criteria for the S-boxes are as follows,

1.No output bit of any S-box should be too close a linear function of the input bits. Specifically, if we select any output bit and any subset of the six input bits, the fraction of inputs for which this output bit equals the XOR of these input bits should not be close to 0 or 1, but rather should be near 1/2.

2. Each row of an S-box (determined by a fixed value of the leftmost and rightmost input bits) should include all 16 possible output bit combinations.

3. If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.

4. If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.

5. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.

6. For any nonzero 6-bit difference between inputs, no more than eight of the 32 pairs of inputs exhibiting that difference may result in the same output difference.

7. This is a criterion similar to the previous one, but for the case of three S-boxes.

Coppersmith pointed out that the first criterion in the preceding list was needed because the S-boxes are the only nonlinear part of DES. If the S-boxes were linear (i.e., each output bit is a linear combination of the input bits), the entire algorithm would be linear and easily broken. We have seen this phenomenon with the Hill cipher, which is linear. The remaining criteria were primarily aimed at thwarting differential cryptanalysis and at providing good confusion properties.

The criteria for the permutation P are as follows.

1. The four output bits from each S-box at round *i* are distributed so that two of them affect (provide input for) "middle bits" of round (*i* + 1) and the other two affect end bits. The two middle bits of input to an S-box are not shared with adjacent S-boxes. The end bits are the two left-hand bits and the two right-hand bits, which are shared with adjacent S-boxes.

2. The four output bits from each S-box affect six different S-boxes on the next round, and no two affect the same S-box.

3. For two S-boxes *j*, *k*, if an output bit from S_j affects a middle bit of S_k on the next round, then an output bit from S_k cannot affect a middle bit of S_j. This implies that, for *j*=*k*, an output bit from S_j must not affect a middle bit of S_j.

These criteria are intended to increase the diffusion of the algorithm.

The cryptographic strength of a Feistel cipher derives from three aspects of the design: the number of rounds, the function F, and the key schedule algorithm.

- The greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak F. In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack.
- The function F provides the element of confusion in a Feistel cipher. Thus, it must be difficult to "unscramble" the substitution performed by F.
- A final area of block cipher design, and one that has received less attention than S-box design, is the key schedule algorithm.With any Feistel block cipher, the key is used to generate one subkey for each round. In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key

Check Your Progress 1

1. What are the most popular types of cryptanalysis attack?
2. List the three critical aspects of block cipher design.
3. Which forms the basis of the DES design criteria?

5.5 ANSWERSTOCHECKYOURPROGRESS

1. The most popular types of cryptanalysis attack are,
    i. ***Differential cryptanalysis***
    ii. ***Linear cryptanalysis***
2. The three critical aspect of block cipher design are
    i. Number of rounds
    ii. Design of the function F
    iii. Key scheduling

3. The prime focus of the DES design criteria is the S box and the P function.

5.6 SUMMARY

Differential cryptanalysis aims to observe the behavior of pairs of text blocks evolving along each round of the cipher, instead of observing the evolution of a single text block. Linear cryptanalysis attack is based on finding linear approximations to describe the transformations performed in any encryption algorithm. Three critical aspects of block cipher design are the number of rounds, design of the function F, and key scheduling. The criteria used in the design of DES, focused on the design of the S-boxes and on the P function that takes the output of the S-boxes.

5.7 KEYWORDS

- A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block ofequal length.
- Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded sothat a different key is used for each round.
- The Data Encryption Standard (DES) has been the most widely used encryption algorithm until recently. It exhibits the classic Feistel structure.DES uses a 64-bit block and a 56-bit key.
- Two important methods of cryptanalysis are differential cryptanalysis and linear cryptanalysis. DES has been shown to be highly resistant to these twotypes of attack.

5.8SELF-ASSESSMENTEXERCISES

Short Questions

1. Discuss the need for cryptanalysis.
2. Write a note on Differential cryptanalysis
3. What is linear cryptanalysis?

Detail Questions

1. Discuss about the two types of cryptanalysis.
2. Describe the design criteria for S box of DES.
3. Explain the design criteria for Permutation P.

5.9SUGGESTEDREADINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

UNIT - 6 ADVANCED ENCRYPTION STANDARD (AES)

NOTES

Structure

6.0 INTRODUCTION

The Advanced Encryption Standard (AES) was published by the National Institute of Standards and Technology (NIST) in 2001. AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications. Compared to public-key ciphers such as RSA, the structure of AES and most symmetric ciphers is quite complex and cannot be explained as easily as many other cryptographic algorithms.

6.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the finite field arithmetic
- Describe the AES structure
- Understand the working of AES

6.2 FINITE FIELD ARITHMETIC

In AES, all operations are performed on 8-bit bytes. In particular, the arithmetic operations of addition, multiplication, and division are performed over the finite field GF(2^8). In essence, a field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set.

Division is defined with the following rule:*$a/b=a(b^{-1})$*. An example of a finite field (one with a finite number of elements) is the set Z_p consisting of all the integers [0,1,.... *p-1*] , where *p* is a prime number and in which arithmetic is carried out modulo *p*.

With the appropriate definition of arithmetic operations, each such set *S* is a finite field. The definition consists of the following elements.

1. Arithmetic follows the ordinary rules of polynomial arithmetic using the basic rules of algebra with the following two refinements.
2. Arithmetic on the coefficients is performed modulo 2. This is the same as the XOR operation.
3. If multiplication results in a polynomial of degree greater than n-1, then the polynomial is reduced modulo some irreducible polynomial *m(x)* of degree *n*. That is, we divide by *m(x)* and keep the remainder. For a polynomial *f(x)*, the remainder is expressed as *r(x) = f(x) mod m(x)*. A polynomial *m(x)* is called **irreducible** if and only if *m(x)* cannot be expressed as a product of two polynomials, both of degree lower than that of *m(x)*.

To summarize, AES operates on 8-bit bytes. Addition of two bytes is defined as the bitwise XOR operation. Multiplication of two bytes is defined as multiplication in the finite field GF(2^8), with the irreducible polynomial $m(x)=x^8+x^4+x^3+x+1$.

.

6.3 AES STRUCTURE

Figure 6.1 shows the overall structure of the AES encryption process. The cipher takes a plaintext block size of 128 bits, or 16 bytes. The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits). The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.

The cipher consists of *N* rounds, where the number of rounds depends on the key length: 10 rounds for a 16-byte key, 12 rounds for a 24-byte key, and 14 rounds for a 32-byte key as given in the below table 6.1.

The first *N-1* rounds consist of four distinct transformation functions:

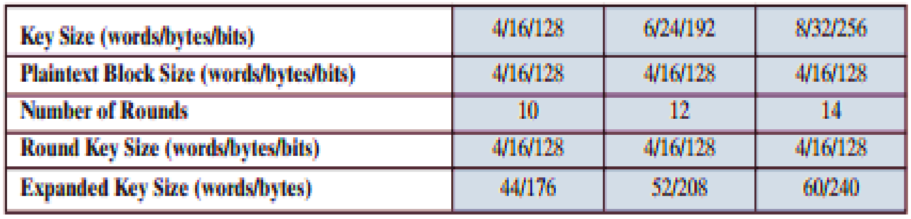
i. SubBytes
ii. ShiftRows
iii. MixColumns and
iv. AddRoundKey

which are described subsequently. The final round contains only three transformations, and there is an initial single transformation (AddRoundKey) before the first round, which can be considered Round 0. Each transformation takes one or more 4x4 matrices as input and produces a 4x4 matrix as output. The output of each round is a 4x4 matrix, with the

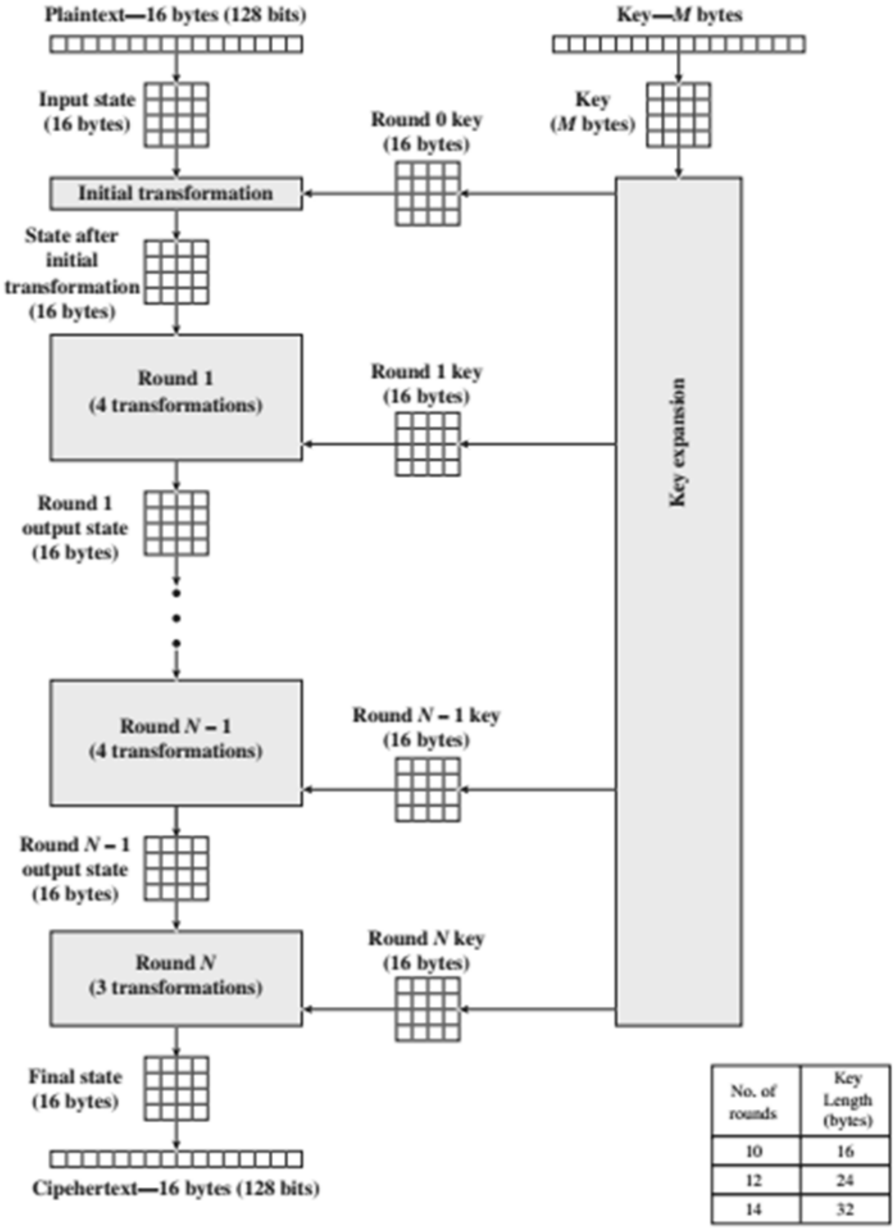
NOTES

output of the final round being the ciphertext. Also, the key expansion function generates *N+1* round keys, each of which is a distinct 4x4 matrix. Each round key serve as one of the inputs to the AddRoundKey transformation in each round.

Table 6.1 AES Parameters



Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240


Plaintext—16 bytes (128 bits)
Key—*M* bytes
Input state (16 bytes)
Key (*M* bytes)
Round 0 key (16 bytes)
Initial transformation
State after initial transformation (16 bytes)
Round 1 (4 transformations)
Round 1 key (16 bytes)
Key expansion
Round 1 output state (16 bytes)
Round *N* – 1 (4 transformations)
Round *N* – 1 key (16 bytes)
Round *N* – 1 output state (16 bytes)
Round *N* (3 transformations)
Round *N* key (16 bytes)
Final state (16 bytes)
Cipehertext—16 bytes (128 bits)

No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 6.1 AES Encryption Process

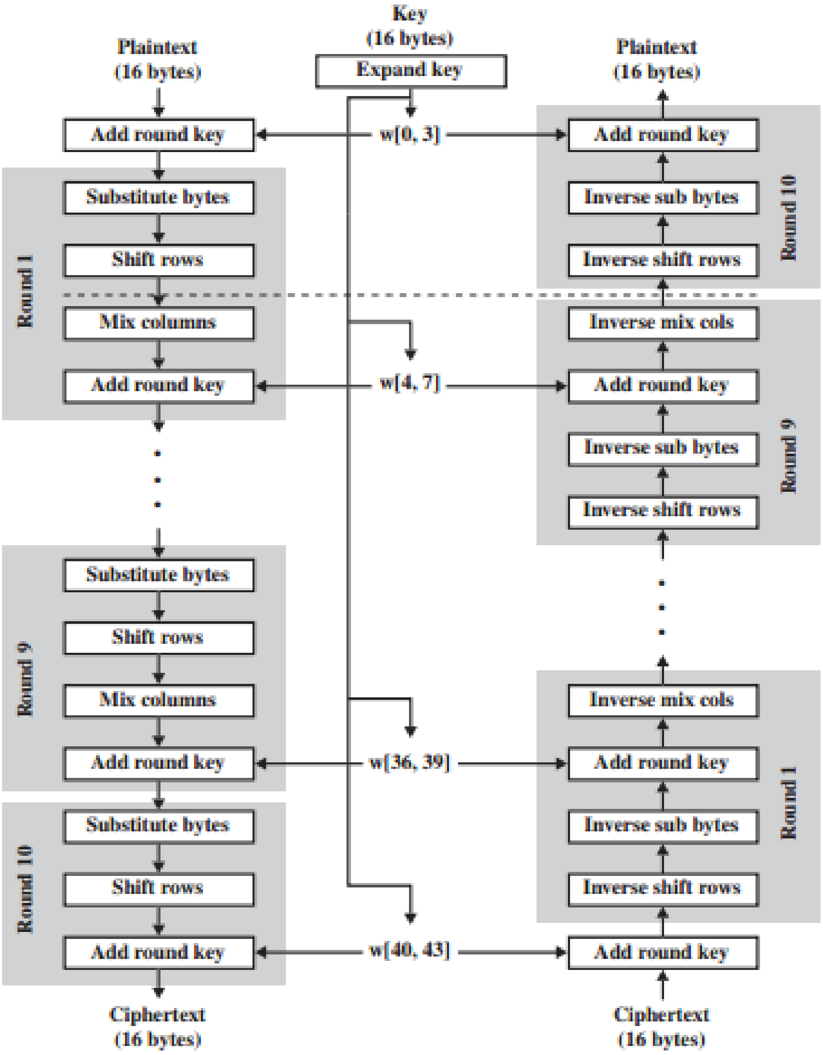
Four different stages are used, one of permutation and three of substitution:

Substitute bytes : Uses an S-box to perform a byte-by-byte substitution of the block

ShiftRows : A simple permutation

MixColumns : A substitution that makes use of arithmetic over

AddRoundKey : A simple bitwise XOR of the current block with a portion of the expanded key



(a) Encryption

(b) Decryption

Figure 6.2. AES Encryption and Decryption

6.4 AES TRANSFORMATION FUNCTION

This section will discuss about each of the four transformations used in AES. Foreach stage, we describe the forward (encryption) algorithm and the inverse (decryption) algorithm for the stage.

Figure 6.3 presents the over view of a single round of AES, emphasizing the mechanismsand inputs of each transformation.

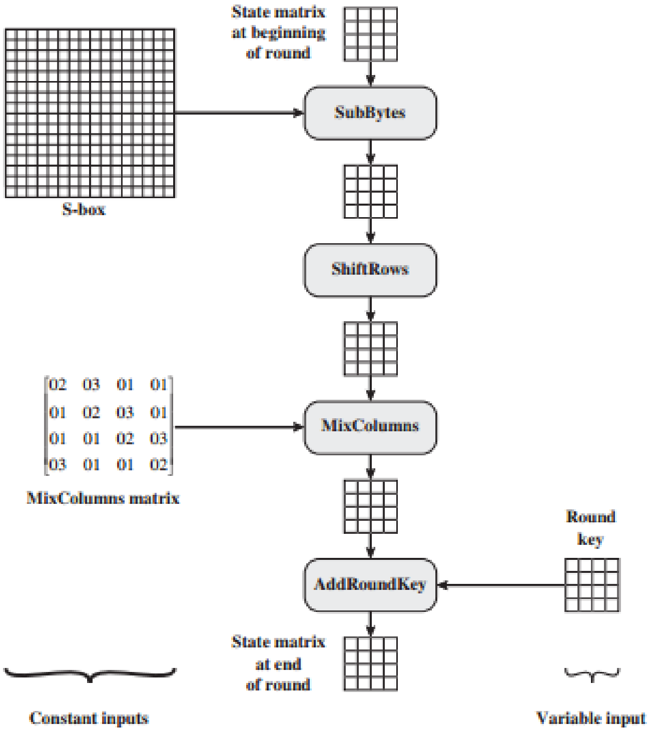


Fig 6.8 Inputs for Single AES Round

6.4.1 SUBSTITUTE BYTES TRANSFORMATION

The ***forward substitute byte transformation, called SubBytes***, is a simple table lookup (Figure 6.4). AES defines a 16x16 matrix of byte values, called an S-box, that contains a permutation of all possible 256 8-bit values. Each individual byte of State is mapped into a new byte in the following way: The leftmost 4 bits of the byte are used as a row value and the rightmost 4 bits are used as a column value. These row and column values serve as indexes into the S-box to select a unique 8-bit output value.

The ***inverse substitute byte transformation, called InvSubBytes***, makes use of the inverse S-box.

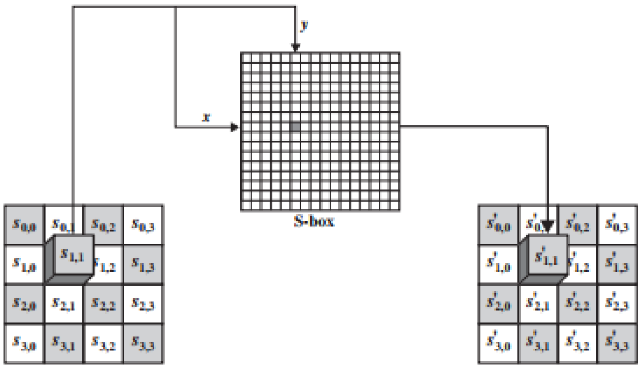

y
x
S-box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig 6.4.Substitute Byte Transformation

6.4.2 SHIFTROWS TRANSFORMATION

The **forward shift row transformation**, called ShiftRows, is depicted in Figure 6.5. The first row of State is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed. For the fourth row, a 3-byte circular left shift is performed.

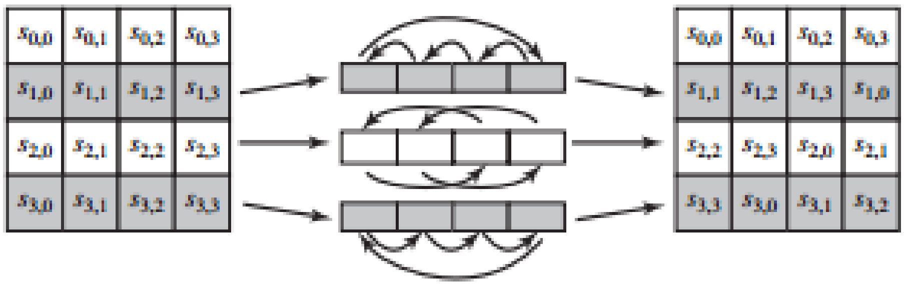

$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Fig 6.5.Shift Row Transformation

The **inverse shift row transformation**, called InvShiftRows, performs the circular shifts in the opposite direction for each of the last three rows, with a 1-byte circular right shift for the second row, and so on.

6.4.3 MIXCOLUMNS TRANSFORMATION

The forward mix column transformation, called MixColumns, operates on each column individually. Each byte of a column is mapped into a new value that is a function of all four bytes in that column. The transformation can be defined by the following matrix multiplication on State.

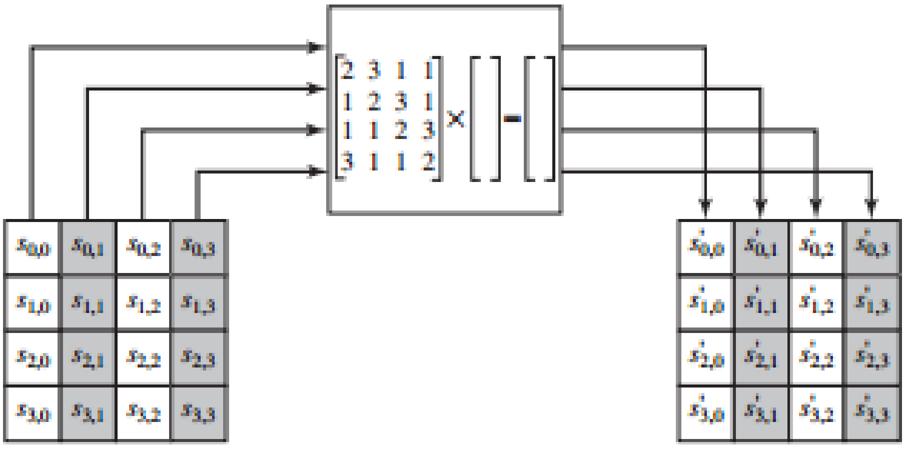

2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2
×
=
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig.6.6.Mix Column Transformation

6.4.4 ADDROUNDKEY TRANSFORMATION

In the forward add round key transformation, called AddRoundKey, the 128 bits of State are bitwise XORed with the 128 bits of the round key. As shown in Figure, the operation is viewed as a columnwise operation between the 4 bytes of a Statecolumn and one word of the round key; it can also be viewed as a byte-level operation.

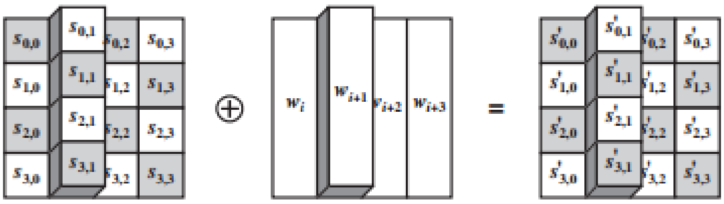
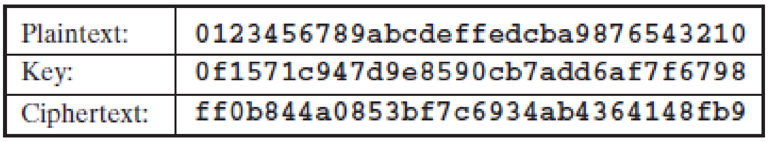

$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
⊕
w_i w_{i+1} w_{i+2} w_{i+3}
=
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig.6.7.Add round key Transformation

The inverse add round key transformation is identical to the forward add round key transformation, because the XOR operation is its own inverse.

Example for AES :



Plaintext:	0123456789abcdeffedcba9876543210
Key:	0f1571c947d9e8590cb7add6af7f6798
Ciphertext:	ff0b844a0853bf7c6934ab4364148fb9

NOTES

6.5AES IMPLEMENTATION

Two separate changes are needed to bring the decryption structure in line with the encryption structure. An encryption round has the structure SubBytes, ShiftRows, MixColumns, AddRoundKey. The standard decryption round has the structure InvShiftRows, InvSubBytes, AddRoundKey, InvMixColumns. Thus, the first two stages of the decryption round need to be interchanged, and the second two stages of the decryption round need to be interchanged.

Interchanging InvShiftRows and InvSubBytes:

InvShiftRows affects the sequence of bytes in State but does not alter byte contents and does not depend on byte contents to perform its transformation. InvSubBytes affects the contents of bytes in State but does not alter byte sequence and does not depend on byte sequence to perform its transformation. Thus, these two operations commute and can be interchanged. For a given State S_i,

InvShiftRows [InvSubBytes (S_i)] =InvSubBytes [InvShiftRows (S_i)]

Interchanging AddRoundKey and InvMixColumns:

The transformations AddRoundKey and InvMixColumns do not alter the sequence of bytes in State. If we view the key as a sequence of words, then both AddRoundKey and InvMixColumns operate on Stateone column at a time. These two operations are linear with respect to the column input.

That is, for a given State S_i and a given round key W_j,

$$\begin{aligned}&\text{InvMixColumns}\,(Si\ \oplus \text{W}j)\\&\qquad= [\text{InvMixColumns}\,(Si\,)]\ \oplus\ \text{InvMixColumns}\,(\text{W}j\,)]\end{aligned}$$

<u>Check Your Progress 1</u>

1. What is an irreducible polynomial?
2. State the importance of key length in AES.
3. What are the four distinct transformation functions?

6.6ANSWERSTOCHECKYOURPROGRESS

1. A polynomial m(x) is called irreducible if and only if m(x) cannot be expressed as a product of two polynomials, both of degree lower than that of m(x).

2. The cipher takes a plaintext block size of 128 bits, or 16 bytes. The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits). The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.

3. The four distinct transformation functions are
   i. SubBytes
   ii. ShiftRows
   iii. MixColumns and
   iv. AddRoundKey.

6.7 SUMMARY

The unit begins with the description of finite field arithmetic. Next to that the unit explains the working of the AES algorithm. The four transformation functions of AES are described with neat diagram. The changes to be carried out during the implementation phase of AES are also described in this unit.

6.8 KEYWORDS

- AES is a block cipher intended to replace DES for commercial applications. It uses a 128-bit block size and a key size of 128, 192, or 256 bits.
- AES does not use a Feistel structure. Instead, each full round consists of four separate functions: byte substitution, permutation, arithmetic operations over a finite field, and XOR with a key

6.9SELF-ASSESSMENTEXERCISES

Short Questions

1. Write a note on finite field arithmetic.
2. What do you mean by substitute byte transformations?
3. What is shift rows transformation?
4. Describe the mix column transformation?
5. Explain the add round key.

Detail Questions

1. Discuss in detail about AES transformation function.
2. What are the changes to be incorporated during AES implementation? Describe.

NOTES

6.10SUGGESTEDREADINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

BLOCK – III PUBLIC KEY CRYPTOGRAPHY AND RSA

UNIT - 7 PRINCIPLES OF PUBLIC-KEY CRYPTOSYSTEM

Structure

7.0 INTRODUCTION

The development of public-key cryptography is the greatest and perhaps the onlytrue revolution in the entire history of cryptography. From its earliest beginningsto modern times, virtually all cryptographic systems have been based on the elementarytools of substitution and permutation.

Public-key algorithms are based on mathematical functionsrather than on substitution and permutation. More important, public-key cryptographyis asymmetric, involving the use of two separate keys, in contrast to symmetric encryption,which uses only one key. The use of two keys has profound consequences in theareas of confidentiality, key distribution, and authentication.

This unit will bring the basic terminology as well as the steps involved in the working of RSA algorithm.

7.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the terminologies of public key cryptography

- Differentiate the private and public key cryptography
- Understand the working of RSA algorithms

7.2 PUBLIC KEY CRYPTOGRAPHY

Before proceeding, we should know the basic terminology related to public-key crypto systems.

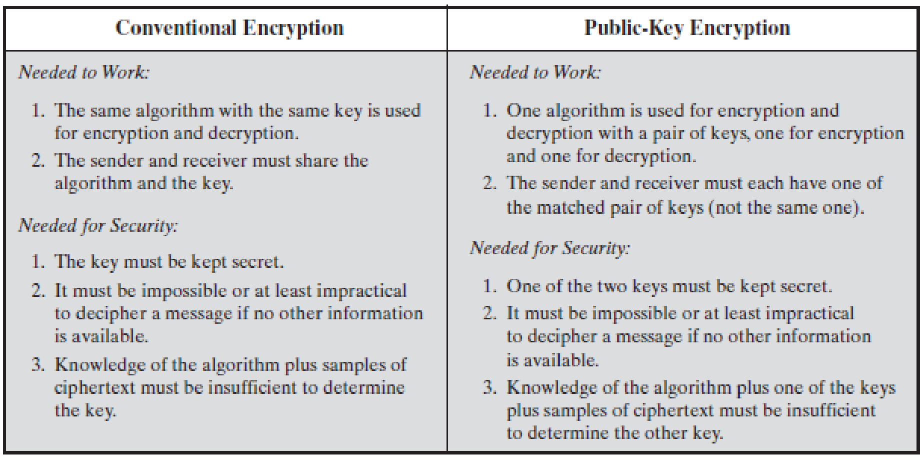
Asymmetric Keys
Two related keys, a public key and a private key, that are used to perform complementary operations, such asencryption and decryption or signature generation and signature verification.

Public Key Certificate
A digital document issued and digitally signed by the private key of a Certification Authority that binds thename of a subscriber to a public key.The certificate indicates that the subscriber identified in the certificate hassole control and access to the corresponding private key.

Public Key (Asymmetric) Cryptographic Algorithm
A cryptographic algorithm that uses two related keys, a public key and a private key.The two keys have the property that deriving the private key from the public key is computationally infeasible.

Public Key Infrastructure (PKI)
A set of policies, processes, server platforms, software and workstations used for the purpose of administeringcertificates and public-private key pairs, including the ability to issue, maintain, and revoke public keycertificates.

The comparison of conventional and public key encryption algorithms are shown below.



Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key.	*Needed to Work:* 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security: 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristic.

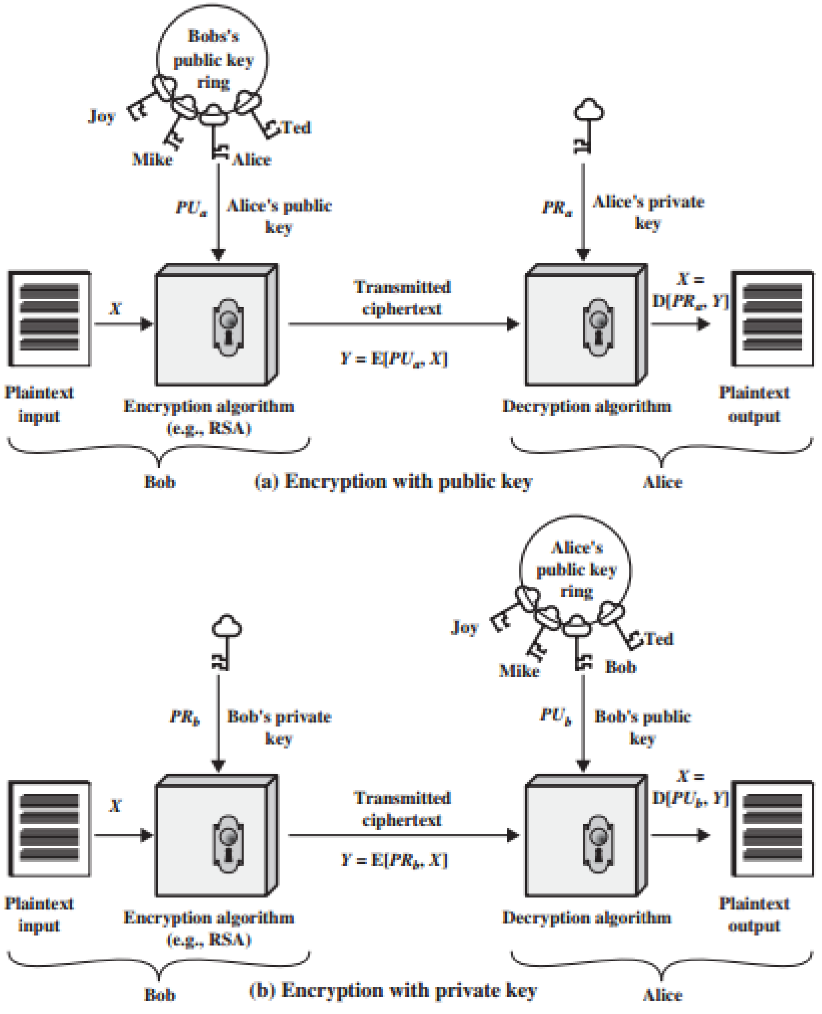
- *It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.*

In addition, some algorithms, such as RSA, also exhibit the following characteristic.

- *Either of the two related keys can be used for encryption, with the other used for decryption.*

A public-key encryption scheme has six ingredients, namely

- Plaintext
- Encryption algorithm
- Public key and private key
- Ciphertext
- Decryption algorithm



(a) Encryption with public key

(b) Encryption with private key

NOTES

Fig 7.1 Public key cryptography

Plaintext: This is the readable message or data that is fed into the algorithm as input.

Encryption algorithm: The encryption algorithm performs various transformations on the plaintext.

Public and private keys: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.

Ciphertext : This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.

Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The essential steps are the following.

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. As Figure 7.1a suggests, each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.
4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

.

7.3 THE RSA ALGORITHM

The pioneering paper by Diffie and Hellman introduced a new approach to cryptography and, in effect, challenged cryptologists to come up with a cryptographic algorithm that met the requirements for public-key systems. A number of algorithms have been proposed for public-key cryptography. Some of these, though initially promising, turned out to be breakable.

One of the first successful responses to the challenge was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978. The Rivest-Shamir-Adleman (RSA) scheme has since that time reigned supreme as the most widely accepted and implemented general-purpose approach to public-key encryption.

The RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and *n*-1 for some n. A typical size for n is 1024 bits, or 309 decimal digits. That is, *n*is less than 2^{1024}.

7.4 DESCRIPTION OF THE ALGORITHM

RSA makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n. That is, the block size must be less than or equal to log2(n)+1; in practice, the block size is *i* bits, where $2^i < n \leq 2^{i+1}$.

Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C.

$$C = M^e \bmod n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Both sender and receiver must know the value of *n*. The sender knows the value of *e*, and only the receiver knows the value of *d*. Thus, this is a public-key encryption algorithm with a public key of PU={e,n} and a private key of PR={d,n}. For this algorithm to be satisfactory for public-key encryption, the following requirements must be met.

1. It is possible to find values of *e,d,n* such that $M^{ed} \bmod n = M$ for all $M<n$.
2. It is relatively easy to calculate M^e mod *n* and C^d mod *n* for all values of $M<n$.
3. It is infeasible to determine *d* given *e* and *n*.

Key Generation Alice	
Select *p, q*	*p* and *q* both prime, *p q*
Calculate $n = p \times q$	
Calculate $Ø(n) = (p-1)(q-1)$	
Select integer *e*	$gcd(Ø (n),e)=1;1<e< Ø(n)$
Calculate *d*	$d=e-1(mod (n))$
Public key	$PU = (e,n)$
Private key	$PR = (d,n)$
Encryption by Bob with Alice's Public Key	
Plaintext	$M < n$
Ciphertext	$C = M^e \bmod n$
Decryption by Alice with Alice's Public Key	
Ciphertext	C
Plaintext	$M = C^d \bmod n$

Fig 7.2RSA Algorithm

Check Your Progress 1

1. What are asymmetric keys?
2. What do you mean by public key certificate?
3. What is public key algorithm?
4. Define Public key Infrastructure.

7.5ANSWERSTOCHECKYOURPROGRESS

1. ***Asymmetric Keys***
   Two related keys, a public key and a private key, that are used to perform complementary operations, such asencryption and decryption or signature generation and signature verification.

2. ***Public Key Certificate***
   A digital document issued and digitally signed by the private key of a Certification Authority that binds thename of a subscriber to a public key.The certificate indicates that the subscriber identified in the certificate hassole control and access to the corresponding private key.

3. ***Public Key (Asymmetric) Cryptographic Algorithm***
   A cryptographic algorithm that uses two related keys, a public key and a private key.The two keys have the property that deriving the private key from the public key is computationally infeasible.

4. ***Public Key Infrastructure (PKI)***
   A set of policies, processes, server platforms, software and workstations used for the purpose of administeringcertificates and public-private key pairs, including the ability to issue, maintain, and revoke public keycertificates.

7.6 SUMMARY

In this unit a brief introduction to the terminologies of public key cryptosystem is given. The working of asymmetric key cryptography is described with an example. The most popular RSA algorithm is described in this unit.

7.7 KEYWORDS

- Asymmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the different keys—one a public key and one a private key. It is also known as public-key encryption.
- Asymmetric encryption transforms plaintext into ciphertext using a one of two keys and an encryption algorithm. Using

the paired key and a decryption algorithm, the plaintext is recovered from the ciphertext.

- Asymmetric encryption can be used for confidentiality, authentication, or both.
- The most widely used public-key cryptosystem is RSA. The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.

7.8SELF-ASSESSMENTEXERCISES

Short Questions

1. List the key terms of public key crypto systems.
2. What is difference between conventions crypto systems and public key crypto systems?

Detail Questions

1. Explain the characteristics of public key crypto systems
2. Describe the RSA algorithm.

7.9SUGGESTEDREADINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

UNIT- 8 OTHER PUBLIC KEY CRYPTOSYSTEMS– PART I

Structure

8.0 INTRODUCTION

The following are some of many Public key cryptosystems (PKCS) available publicly. In this section the following cryptosystem are briefly explained.

- Diffie-Hellman Key Exchange
- Elgamal Cryptosystem

This chapter begins with a description of one of the earliest and simplest PKCS: Diffie-Hellman key exchange. The Elagamal cryptosystem is also explained in this unit.

8.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the working of Diffie-Hellman Key exchange
- Describe the working of Elagamal cryptosystem

8.2 DIFFIE-HELLMAN KEY EXCHANGE

The first published public-key algorithm appeared in the seminal paper by Diffie and Hellman that defined public-key cryptography and is generally referred to as Diffie-Hellman key exchange. A number of commercial products employ this key exchange technique.

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values.

The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Briefly, we can define the discrete logarithm in the following way. A ***primitive root*** of a prime number *p* as one whose powers modulo *p* generate all the integers from *1* to *p-1*. That is, if *a* is a primitive root of the prime number *p*, then the numbers

$$a \bmod p,\ a^2 \bmod p,\ \dots,\ a^{p-1} \bmod p$$

are distinct and consist of the integers from *1* through *p-1* in some permutation.

For any integer *b* and a primitive root *a* of prime number *p*, we can find a unique exponent *i* such that

$$b = a^j\ (\mathrm{mod}\ p) \qquad \text{where } 0<=i<= (p-1)$$

The exponent *i* is referred to as the discrete logarithm of b for the base *a*, mod *p*. We express this value as $\mathrm{dlog}_{a,p}(b)$.

Global Public Elements	
Q *A*	Prime number $\alpha < q$ and α a primitive root of *q*
User A Key Generation	
Select private X_A Calculate public Y_A	$X_A < q$ $Y_A = \alpha^{XA} \bmod q$
User B Key Generation	
Select private X_B Calculate public Y_B	$X_B<q$ $Y_B = \alpha^{XB} \bmod q$
Calculation of Secret Key by User A	
$K = (Y_B)^{XA} \bmod q$	
Calculation of Secret Key by user B	
$K = (Y_A)^{XB} \bmod q$	

Fig.8.1.The Diffie-Hellman Key Exchange Algorithm

The security of the Diffie-Hellman key exchange lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms. For large primes, the latter task is considered infeasible.

Suppose that user A wishes to set up a connection with user B and use a secretkey to encrypt messages on that connection. User A can generate a one-time private key X_A, calculate Y_A, and send that to user B. User B responds by generating a private value X_B, calculating Y_B, and sending Y_B to user A. Both users can now calculate the key. The necessary public values and would need to be known ahead of time. Alternatively, user A could pick values for q and α and include those in the first message.

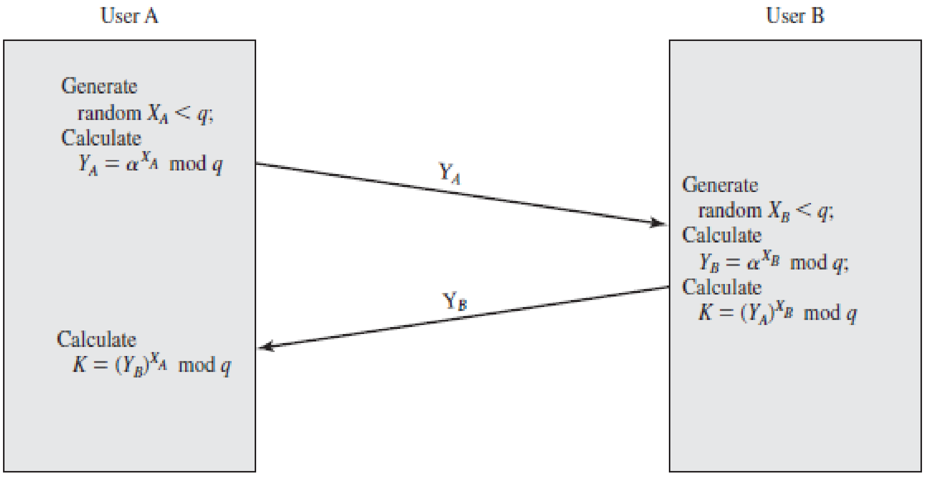

User A
User B
Generate random $X_A < q$;
Calculate $Y_A = \alpha^{X_A} \bmod q$
Y_A
Generate random $X_B < q$;
Calculate $Y_B = \alpha^{X_B} \bmod q$;
Calculate $K = (Y_A)^{X_B} \bmod q$
Y_B
Calculate $K = (Y_B)^{X_A} \bmod q$

Fig. 8.2.Diffie-Hellman Key Exchange

Man-in-the-Middle Attack

The protocol depicted in Figure 8.2 is insecure against a man-in-the-middle attack.Suppose Alice and Bob wish to exchange keys, and Darth is the adversary.

1. Darth prepares for the attack by generating two random private keys X_{D1} and X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.
2. Alice transmits Y_A to Bob.
3. Darth intercepts Y_A and transmits Y_{D1} to Bob. Darth also calculates K2.
4. Bob receives Y_{D1} and calculates K1.
5. Bob transmits Y_B to Alice.
4. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates K1.
5. Alice receives Y_{D2} and calculates K2.

At this point, Bob and Alice think that they share a secret key, but instead Boband Darth share secret key K1 and Alice and Darth share secret key K2.

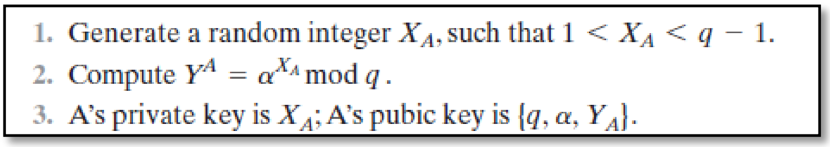
The key exchange protocol is vulnerable to such an attack because it doesnot authenticate the participants. This vulnerability can be overcome with the useof digital signatures and public-key certificates.

8.3 ELGAMMAL CRYPTOGRAPHIC SYSTEM

In 1984, T. Elgamal announced a public-key scheme based on discrete logarithms, closely related to the Diffie-Hellman technique. The ElGamal cryptosystem is used in some form in a number of standards including the digital signature standard (DSS), and the S/MIME e-mail standard.

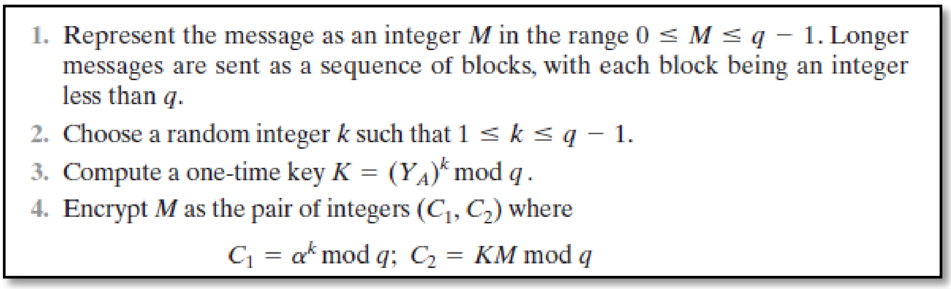
As with Diffie-Hellman, the global elements of ElGamal are a prime number q and α, which is a primitive root of q.

User A generates a private/public key pair as follows:



1. Generate a random integer X_A, such that $1 < X_A < q - 1$.
2. Compute $Y^A = \alpha^{X_A} \bmod q$.
3. A's private key is X_A; A's pubic key is $\{q, \alpha, Y_A\}$.

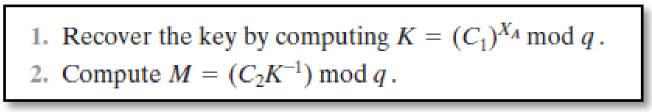
Any user B that has access to A's public key can encrypt a message as follows:



1. Represent the message as an integer M in the range $0 \leq M \leq q - 1$. Longer messages are sent as a sequence of blocks, with each block being an integer less than q.
2. Choose a random integer k such that $1 \leq k \leq q - 1$.
3. Compute a one-time key $K = (Y_A)^k \bmod q$.
4. Encrypt M as the pair of integers (C_1, C_2) where

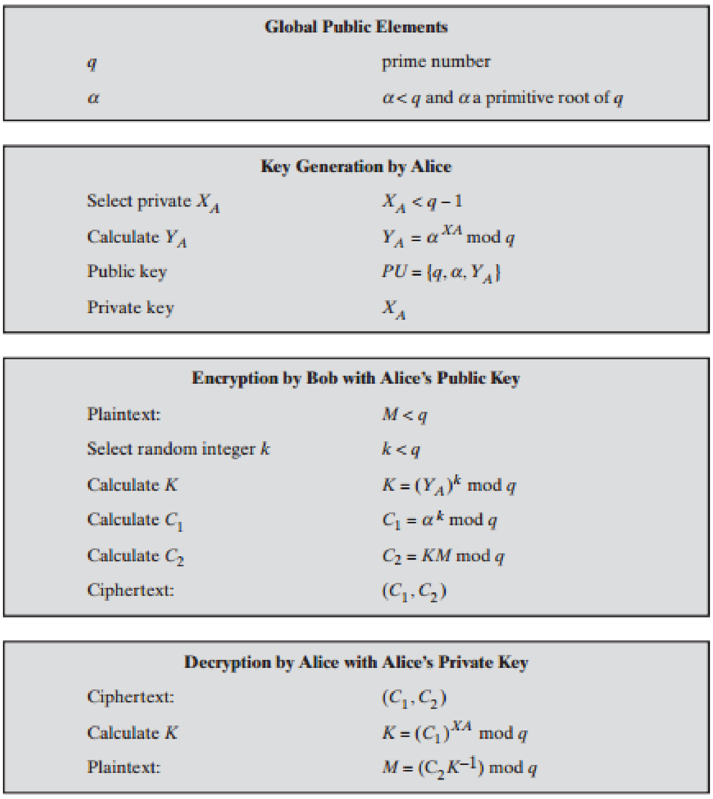
$$C_1 = \alpha^k \bmod q; \; C_2 = KM \bmod q$$

User A recovers the plaintext as follows:



1. Recover the key by computing $K = (C_1)^{X_A} \bmod q$.
2. Compute $M = (C_2K^{-1}) \bmod q$.

We can restate the ElGamal process as follows, using below Figure.



Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice

Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{XA} \bmod q$
Public key	$PU = \{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key

Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key

Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{XA} \bmod q$
Plaintext:	$M = (C_2K^{-1}) \bmod q$

Fig. 8.3.Elgammal Cryptographic System

1. Bob generates a random integer *k*.
2. Bob generates a one-time key *K* using Alice's public-key components Y_A, *q*, and *k* .
3. Bob encrypts *k* using the public-key component α, yielding C_1. C_1 provides sufficient information for Alice to recover *K*.
4. Bob encrypts the plaintext message *M* using *K* .
5. Alice recovers *K* from C_1 using her private key.
6. Alice uses K^{-1} to recover the plaintext message from C_2.

The security of ElGamal is based on the difficulty of computing discrete logarithms.

<u>Check Your Progress 1</u>

1. What is a primitive root?
2. How to overcome men in middle attack in key exchange protocols?
3. State the basis of ElGamal security?

8.4ANSWERSTOCHECKYOURPROGRESS

1. A ***primitive root*** of a prime number *p* as one whose powers modulo *p* generate all the integers from *1* to *p-1.*

2. The key exchange protocol is vulnerable to men in middle attack because it doesnot authenticate the participants. This vulnerability can be overcome with the useof digital signatures and public-key certificates.

3. The security of ElGamal is based on the difficulty of computing discrete logarithms.

8.5 SUMMARY

This section describes the Diffie-Hellman and ElGamal algorithms. The workings of these algorithms are explained using simple steps. Also the issues related to attacks are addressed. The suitable measures for the issues are also suggested.

8.6 KEYWORDS

- A simple public-key algorithm is Diffie-Hellman key exchange.This protocolenables two users to establish a secret key using a public-keyscheme based on discrete logarithms. The protocol is secure only if theauthenticity of the two participants can be established.

8.7SELF-ASSESSMENTEXERCISES

Short Questions

1. What do you mean by men in the middle attack?
2. How B can encrypt a message using A’s public key in ElGamal algorithm?

Detail Questions

1. Describe the Diffie-Hellman algorithm.
2. Explain the ElGamal algorithm.

8.8SUGGESTEDREADINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

UNIT- 9 OTHER PUBLIC KEY CRYPTOSYSTEMS – PART II

Structure

9.0 INTRODUCTION

The principal attraction of ECC, compared to RSA, is that it appears to offerequal security for a far smaller key size, thereby reducing processing overhead. Onthe other hand, although the theory of ECC has been around for some time, it isonly recently that products have begun to appear and that there has been sustainedcryptanalytic interest in probing for weaknesses. Accordingly, the confidence levelin ECC is not yet as high as that in RSA.

ECC is fundamentally more difficult to explain than either RSA or Diffie-Hellman, and a full mathematical description is beyond the scope of this section.This section and the next give some background on elliptic curves and ECC.

9.1 OBJECTIVES

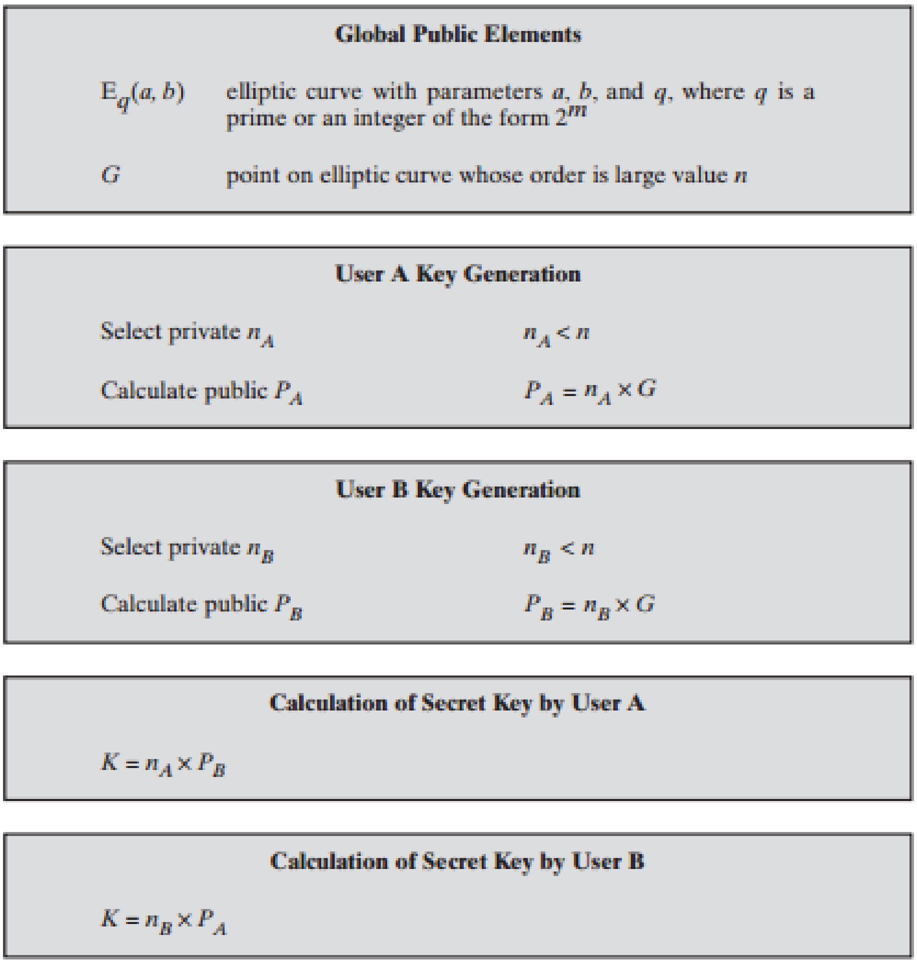
After going through this unit, you will be able to:

- Describe the working of Elliptic Curve Cryptography
- Understand pseudo random number generation based on asymmetric cipher

9.2 ELLIPTIC CURVECRYPTOGRAPHY

Most of the products and standards that use public-key cryptography for encryption and digital signatures use RSA.As we have seen, the key length for secure RSA use has increased over recent years, and this has put a heavier processing load on applications using RSA. This burden has ramifications, especially for electronic commerce sites that conduct large numbers of secure transactions. A competing system challenges RSA: elliptic curve cryptography (ECC). ECC is showing up in standardization efforts, including the IEEE P1363 Standard for Public-Key Cryptography.

An ***elliptic curve***is defined by an equation in two variables with coefficients. For cryptography, the variables and coefficients are restricted to elements in a finite field, which results in the definition of a finite abelian group.



Global Public Elements

$E_q(a, b)$ elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m

G point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A $n_A < n$

Calculate public P_A $P_A = n_A \times G$

User B Key Generation

Select private n_B $n_B < n$

Calculate public P_B $P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

Figure 9.1.ECC Diffie-Hellman Key Exchange

Elliptic curve cryptography makes use of elliptic curves in which the variables and coefficients are all restricted to elements of a finite field. Two families of elliptic curves are used in cryptographic applications: prime curves over Z_p and binary curves over $GF(2^m)$.

The firsttask in this system is to encode the plaintext message *m*to be sent as an *x*–*y* pointP_m. It is the point P_m that will be encrypted as a ciphertext and subsequentlydecrypted.

To encrypt and send a message P_m to B, A chooses a random positive integer *k*and produces the ciphertextC_m consisting of the pair of points

$$C_m = \{kG, P_m + kP_B\}$$

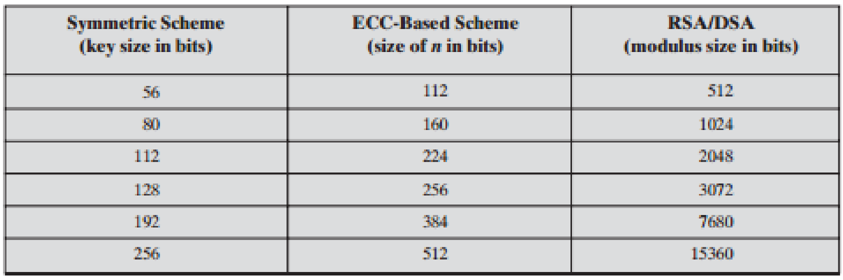
Note that A has used B's public keyP_B. To decrypt the ciphertext, B multiplies thefirst point in the pair by B's secret key and subtracts the result from the secondpoint:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$$

9.3 SECURITY OF ECC

The security of ECC depends on how difficult it is to determine *k* given *kP* and *P*. This is referred to as the elliptic curve logarithm problem. The fastest known technique for taking the elliptic curve logarithm is known as the Pollard rho method. Below table compares various algorithms, by showing comparable key sizes in terms of computational effort for cryptanalysis. As can be seen, a considerably smaller key size can be used for ECC compared to RSA. Furthermore, for equal key lengths, the computational effort required for ECC and RSA is comparable. Thus, there is a computational advantage to using ECC with a shorter key length than a comparably secure RSA.

Table 9.1. Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis



Symmetric Scheme (key size in bits)	ECC-Based Scheme (size of *n* in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

9.4 PSEUDORANDOM NUMBER GENERATION BASED ON AN ASYMMETRIC CIPHER

An asymmetric encryption algorithm produces apparently random output and can be used to build a PRNG. Because asymmetric algorithms are typically much slower than symmetric algorithms, asymmetric algorithms are not used to generate open-ended PRNG bit streams. Rather, the asymmetric approach is useful for creating a pseudorandom function (PRF) for generating a short pseudorandom bit sequence.In this section, we examine two PRNG designs based on pseudorandom functions.

PRNG BASED ON RSA

For a sufficient key length, the RSA algorithm is considered secure and is a good candidate to form the basis of a PRNG. Such a PRNG, known as the Micali-Schnorr PRNG, is recommended in the ANSI standard X9.82 (Random Number Generation) and in the ISO standard 18031 (Random Bit Generation).The PRNG is illustrated in Figure 9.2. As can be seen, this PRNG has much the same structure as the output feedback (OFB) mode used as a PRNG. In this case, the encryption algorithm is RSA rather than a symmetric block cipher. Also, a portion of the output is fed back to the next iteration of the encryption algorithm and the remainder of the output is used as pseudorandom bits. The motivation for this separation of the output into two distinct parts is so that the pseudorandom bits from one stage do not provide input to the next stage. This separation should contribute to forward unpredictability.

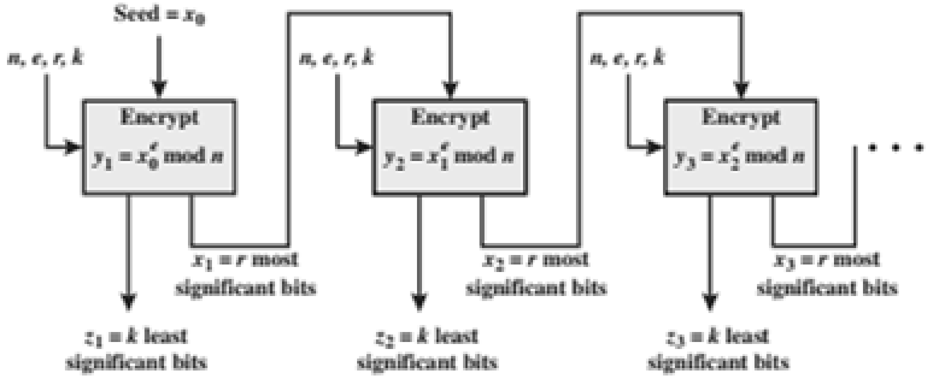

Seed = x_0
n, e, r, k
Encrypt
$y_1 = x_0^e$ mod n
n, e, r, k
Encrypt
$y_2 = x_1^e$ mod n
n, e, r, k
Encrypt
$y_3 = x_2^e$ mod n
• • •
x_1 = r most significant bits
x_2 = r most significant bits
x_3 = r most significant bits
z_1 = k least significant bits
z_2 = k least significant bits
z_3 = k least significant bits

Figure 9.2 Micali-Schnorr Pseudorandom Bit Generator

PRNG BASED ON ECC

In this subsection, we briefly summarize a technique developed by the U.S. National Security Agency (NSA) known as dual elliptic curve PRNG (DEC PRNG). This technique is recommended in NIST SP 800-90, the ANSI standard X9.82, and the ISO standard 18031. There has been some controversy regarding both the security and efficiency of this algorithm compared to other alternatives.

Algorithm

Let P and Q be two known points on a given elliptic curve. The seed of the DEC PRNG is a random integer, where $s_0 \varepsilon$ {0, 1, ... , ≠ E(GF(p))-1}, where ≠E(GF(p)) denotes the number of points on the curve. Let *x* denote a function that gives the *x*-coordinate of a point of the curve. Let $\text{lsb}_i(s)$ denote the *i* least significant bits of an integer *s*. The DEC PRNG transforms the seed into the pseudorandom sequence of length 240*k*, *k*> 0, as follows.

```
fori = 1 to k do
    Set $s_i \leftarrow x(s_{i-1} P)$
```

```
            Set $r_i$  ← $lsb_{240}$ $(x(s_i\ Q))$
      end for
      Return $r_1$, ... ,$r_k$
```

Given the security concerns expressed for this PRNG, the only motivation for its use would be that it is used in a system that already implements ECC but does not implement any other symmetric, asymmetric, or hash cryptographic algorithm that could be used to build a PRNG.

Check Your Progress 1

1. What is an elliptic curve?
2. What do you mean by elliptical curve cryptography?
3. Expand PRNG.

9.5ANSWERSTOCHECKYOURPROGRESS

1. An ***elliptic curve*** is defined by an equation in two variables with coefficients. For cryptography, the variables and coefficients are restricted to elements in a finite field, which results in the definition of a finite abelian group

2. ***Elliptic curve cryptography*** makes use of elliptic curves in which the variables and coefficients are all restricted to elements of a finite field.

3. ***PRNG – Pseudo Random Number Generation***

9.6 SUMMARY

This section describes the other two public key cryptography algorithms namely Elliptical curve cryptography and Pseudo Random Number Generation based on asymmetric cipher. The workings of these algorithms as well as their security aspects are described in this section.

9.7 KEYWORDS

- Elliptic curve arithmetic can be used to develop a variety of elliptic curve cryptography (ECC) schemes, including key exchange, encryption, and digital signature.

- For purposes of ECC, elliptic curve arithmetic involves the use of an elliptic curve equation defined over a finite field. The coefficients and variables in the equation are elements of a finite field. Schemes using Z_P and GF(2^m) have been developed

9.8 SELF-ASSESSMENT EXERCISES

Short Questions

1. Write a note on security of ECC?
2. What is specialty of PRNG based on RSA and PRNG based on ECC?

Detail Questions

1. Describe the Elliptic Curve cryptography.
2. Explain the PRNG.

9.9 SUGGESTED READINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

BLOCK – IV MESSAGE AUTHENTICATION CODES

UNIT- 10 MESSAGE UTHENTICATION REQUIREMENTS

Structure

10.0 INTRODUCTION

One of the most fascinating and complex areas of cryptography are that of messageauthentication and the related area of digital signatures. It would be impossible, inanything less than book length, to exhaust all the cryptographic functions and protocolsthat have been proposed or implemented for message authentication and digitalsignatures. Instead, the purpose of this unit and the next is to provide a broadoverview of the subject and to develop a systematic means of describing the variousapproaches.

10.1 OBJECTIVES

After going through this unit, you will be able to:

- To introduce the requirements for authentication
- Learns the fundamental approach to message authentication
- Understands about message authentication code (MAC)

10.2 MESSAGE AUTHENTICATION

In the context of communications across a network, the following attacks can be identified.

1. ***Disclosure:*** Release of message contents to any person or process not possessing the appropriate cryptographic key.
2. ***Traffic analysis:*** Discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.
3. ***Masquerade:*** Insertion of messages into the network from a fraudulent source.
4. ***Content modification:*** Changes to the contents of a message, including insertion, deletion, transposition, and modification.
5. ***Sequence modification:*** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
6. ***Timing modification:*** Delay or replay of messages.
7. ***Source repudiation:*** Denial of transmission of message by source.
8. ***Destination repudiation:*** Denial of receipt of message by destination.

Measures to deal with the first two attacks are in the realm of message confidentiality and are dealt with in Part One. Measures to deal with items (3) through (6) in the foregoing list are generally regarded as message authentication. Mechanisms for dealing specifically with item (7) come under the heading of digital signatures. Generally, a digital signature technique will also counter some or all of the attacks listed under items (3) through (6). Dealing with item (8) may require a combination of the use of digital signatures and a protocol designed to counter this attack.

In summary, message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. Message authentication may also verify sequencing and timeliness.

A ***digital signature*** is an authentication technique that also includes measures to counter repudiation by the source.

10.3 MESSAGE AUTHENTICATION FUNCTIONS

Any message authentication or digital signature mechanism has two levels of functionality. At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message. This lower-level function is then used as a primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

There are types of functions that may be used to produce an authenticator.

These may be grouped into three classes.

- ***Hash function:*** A function that maps a message of any length into a fixedlength hash value, which serves as the authenticator
- ***Message encryption:*** The ciphertext of the entire message serves as its authenticator

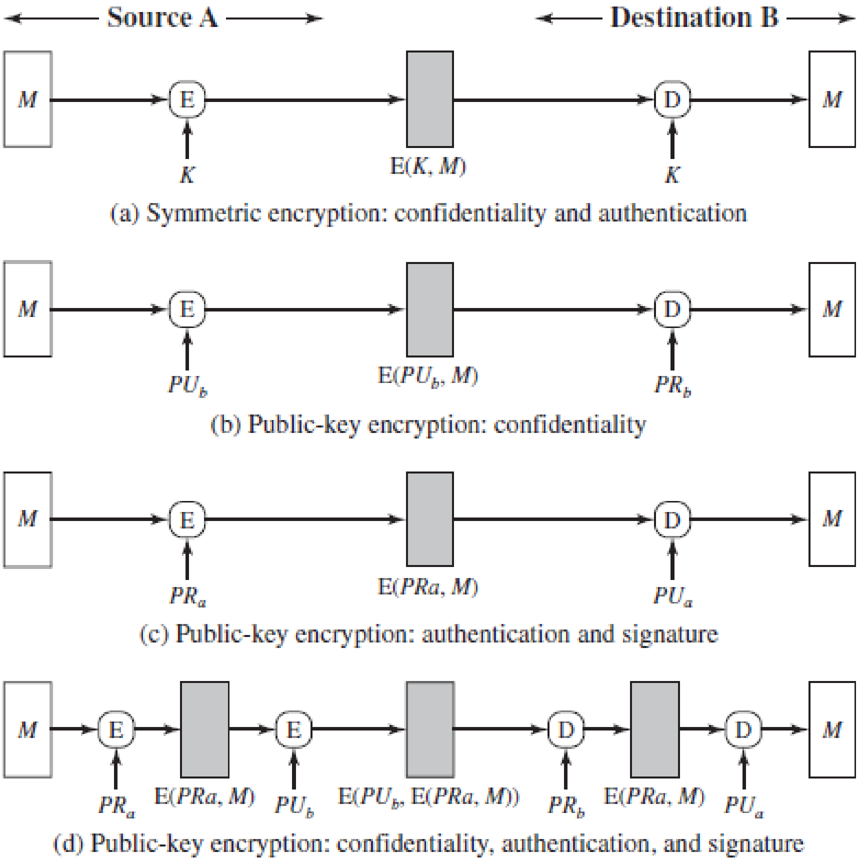

Source A
Destination B
M
E
K
E(K, M)
D
K
M
(a) Symmetric encryption: confidentiality and authentication
M
E
PU_b
E(PU_b, M)
D
PR_b
M
(b) Public-key encryption: confidentiality
M
E
PR_a
E(PRa, M)
D
PU_a
M
(c) Public-key encryption: authentication and signature
M
E
PR_a
E(PRa, M)
E
PU_b
E(PU_b, E(PRa, M))
D
PR_b
E(PRa, M)
D
PU_a
M
(d) Public-key encryption: confidentiality, authentication, and signature

Figure 10.1 Basic Uses of Message Encryption

- ***Message authentication code (MAC):*** A function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

10.4 MESSAGE AUTHENTICATION CODES

An alternative authentication technique involves the use of a secret key to generate a small fixed-size block of data, known as a ***cryptographic checksum*** or MAC,that is appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key. When A has a message to send to B, it calculates the MAC as a function of the message and the key:

$$MAC=MAC(K,M)$$

where

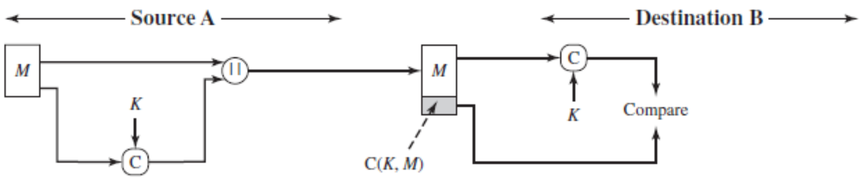
M = input message

C = MAC function

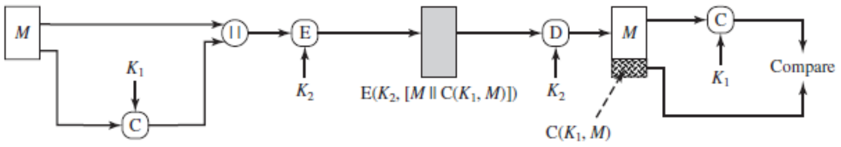
K = shared secret key

MAC = message authentication code

The message plus MAC are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the same secret key, to generate a new MAC. The received MAC is compared to the calculated MAC (as shown in Figure below).

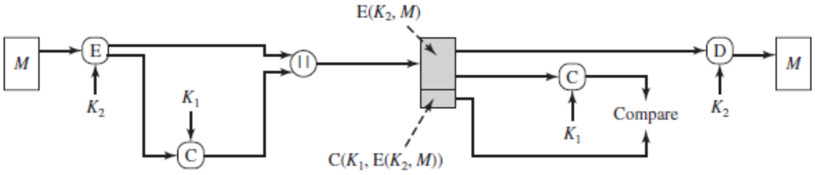


(a) Message Authentication



(b) Message Authentication and Confidentiality; authentication tied to plaintext

NOTES



(c) Message authentication and confidentiality, authentication tied to plaintext

Figure 10.1 Basic Uses of Message Authentication (MAC)

If we assume that only the receiver and the sender know the identityof the secret key, and if the received MAC matches the calculated MAC, then

1. The receiver is assured that the message has not been altered. If an attacker alters the message but does not alter the MAC, then the receiver’s calculation of the MAC will differ from the received MAC. Because the attacker is assumed not to know the secret key, the attacker cannot alter the MAC to correspond to the alterations in the message.
2. The receiver is assured that the message is from the alleged sender. Because no one else knows the secret key, no one else could prepare a message with a proper MAC.
3. If the message includes a sequence number (such as is used with HDLC,X.25, and TCP), then the receiver can be assured of the proper sequence because an attacker cannot successfully alter the sequence number.

A MAC function is similar to encryption. One difference is that the MACalgorithm need not be reversible, as it must be for decryption. In general, the MACfunction is a many-to-one function.The domain of the function consists of messagesof some arbitrary length, whereas the range consists of all possible MACs and allpossible keys.

10.5 REQUIREMENTS FOR MAC

A MAC, also known as a cryptographic checksum, is generated by a function C ofthe form

$$T = MAC(K,M)$$

Where*M* is a variable-length message, *K* is a secret key shared only by sender andreceiver, and MAC(*K*, *M*) is the fixed-length authenticator, sometimes called a ***tag***.The tag is appended to the message at the source at a time when the message isassumed or known to be correct.The receiver authenticates that message by re-computingthe tag.

Message Authentication Requirements

NOTES

Let us state therequirements for the function. Assume that an opponent knows the MAC functionbut does not know *K*. Then the MAC function should satisfy the followingrequirements.

- ✓ An opponent is able to construct a new message to match a given tag, even though the opponent does not know and does not learn the key.
- ✓ The need to thwart a brute-force attack based on chosen plaintext. That is, if we assume that the opponent does not know but does have access to the MAC function and can present messages for MAC generation, then the opponent could try various messages until finding one that matches a given tag. If the MAC function exhibits uniform distribution, then a brute-force method would require, on average, attempts before finding a message that fits a given tag.
- ✓ The final requirement dictates that the authentication algorithm should not be weaker with respect to certain parts or bits of the message than others.

Check Your Progress 1

1. What do you mean by Traffic Analysis?
2. What is digital signature?
3. List any three types of attacks
4. What is MAC?

10.6 ANSWER TO CHECK YOUR PROGRESS

1. Traffic analysis is the discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined
2. A ***digital signature*** is an authentication technique that also includes measures to counter repudiation by the source
3. Some of the types of attacks are
    a. Disclosure
    b. Traffic analysis
    c. Masquerade
    d. Content modification
    e. Sequence modification
    f. Timing modification
    g. Source repudiation
    h. Destination repudiation

4. An alternative authentication technique involves the use of a secret key to generate a small fixed-size block of data, known as a ***cryptographic checksum*** or MAC

10.7 SUMMARY

This unit describes the possible types of attacks of network security. The message authentication functions are elaborated next to that. The need for message authentication codes (MAC) is highlighted then. The MAC is also called as checksum. The MAC function, its components are clearly explained to get better understanding and to expose its importance. The requirements of MAC are analysed and listed in this unit.

10.8 KEYWORDS

- Message authentication is a mechanism or service used to verify theintegrity of a message. Message authentication assures that data receivedare exactly as sent by (i.e., contain no modification, insertion, deletion, orreplay) and that the purported identity of the sender is valid.
- Symmetric encryption provides authentication among those who share thesecret key.
- A message authentication code (MAC) is an algorithm that requires theuse of a secret key

10.9 SELF-ASSESSMENT EXERCISES

Short Questions

1. List the possible types of network security attacks.
2. Explain the MAC function

Detail Questions

1. Describe about MAC codes.
2. Write short notes on MAC security requirements.

10.10 SUGGESTED READINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

UNIT- 11 SECURITY OF MACs

Structure

NOTES

11.0 INTRODUCTION

Thisunit dealswith the security considerationsfor MACs.Here the two categories of attacks in MAC are described. This is followed by a discussion of specific MACs in two categories:those built from cryptographic hash functions and those built using a block cipher mode of operation.

11.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the security of MACs
- Know about MAC based on hash functions
- Learn about MAC based on block ciphers

11.2 SECURITY OF MACs

Just as with encryption algorithms and hash functions, we can group attacks on MACs into two categories:

- Brute-force attacks
- Cryptanalysis

A ***Brute-force attack*** on a MAC is a more difficult undertaking than a brute-force attack on a hash function because it requires known message-tag pairs. To attack a hash code, we can proceed in the following way. Given a fixed message *x* with *n*-bit hash code *h=H(x)*, a brute-force method of finding a collision is to pick a random bit string *y* and check if *H(y) = H(x)*. The attacker can do this repeatedly off line. Whether an off-line attack can be used on a MAC algorithm depends on the relative size of the key and the tag.

As with encryption algorithms and hash functions, ***Cryptanalytic*** attacks on MAC algorithms seek to exploit some property of the algorithm to perform some attack other than an exhaustive search. The way to measure the resistance of a MAC algorithm to cryptanalysis is to compare its strength to the effort required for a brute-force attack. That is, an ideal MAC algorithm will require a cryptanalytic effort greater than or equal to the brute-force effort.

There is much more variety in the structure of MACs than in hash functions, so it is difficult to generalize about the cryptanalysis of MACs. Furthermore, far less work has been done on developing such attacks.

11.3 MACS BASED ON HASH FUNCTIONS: HMAC

The **motivations** for this interest are

1. Cryptographic hash functions such as MD5 and SHA generally execute faster in software than symmetric block ciphers such as DES.
2. Library code for cryptographic hash functions is widely available.

With the development of AES and the more widespread availability of code for encryption algorithms, these considerations are less significant, but hash-based MACs continue to be widely used.

A hash function such as SHA was not designed for use as a MAC and cannot be used directly for that purpose, because it does not rely on a secret key.

HMAC Design Objectives

RFC 2104 lists the following design objectives for HMAC.

- To use, without modifications, available hash functions. In particular, to use hash functions that perform well in software and for which code is freely and widely available.
- To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required.

- To preserve the original performance of the hash function without incurring a significant degradation.
- To use and handle keys in a simple way.
- To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions about the embedded hash function.

HMAC Algorithm

The following diagram illustrates the overall operation of HMAC, Define the following terms.

H = embedded hash function (e.g., MD5, SHA-1, RIPEMD-160)
IV = initial value input to hash function
M = message input to HMAC (including the padding specified in the embedded hash function)
Y_i = i th block of M, $0 \le i \le (L-1)$
L = number of blocks in M
b = number of bits in a block
n = length of hash code produced by embedded hash function
K = secret key; recommended length is $\ge n$; if key length is greater than b, the key is input to the hash function to produce an n-bit key

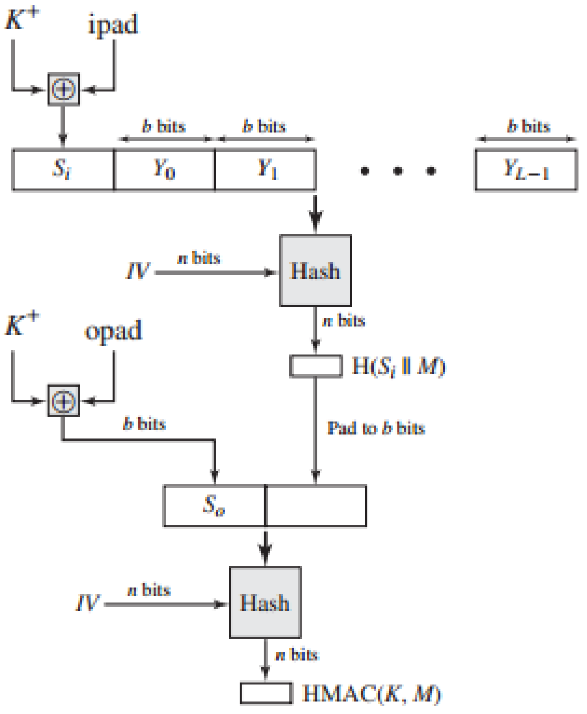


Figure 11.1 HMAC structure

K^+ = K padded with zeros on the left so that the result is b bits in length
ipad = 00110110 (36 in hexadecimal) repeated $b/8$ times
opad = 01011100 (5C in hexadecimal) repeated $b/8$ times

Then HMAC can be expressed as

$$\text{HMAC}(K, M) = \text{H}[(K^{+} \oplus \text{opad}) \| \text{H}[(K^{+} \oplus \text{ipad}) \| M]]$$

We can describe the algorithm as follows.

1. Append zeros to the left end of *K* to create a *b* -bit string K^{-1} (e.g., if *K* is of length 160 bits and *b* = 512, then *K* will be appended with 44 zeroes).
2. XOR (bitwise exclusive-OR) $K^{+}$with ipad to produce the *b*-bit block S_i.
3. Append *M* to S_i.
4. Apply *H* to the stream generated in step 3.
5. XOR K^{+} with opad to produce the *b*-bit block S_0.
6. Append the hash result from step 4 to S_0.
7. Apply H to the stream generated in step 6 and output the result.

The more effective implementation is shown in the following figure.

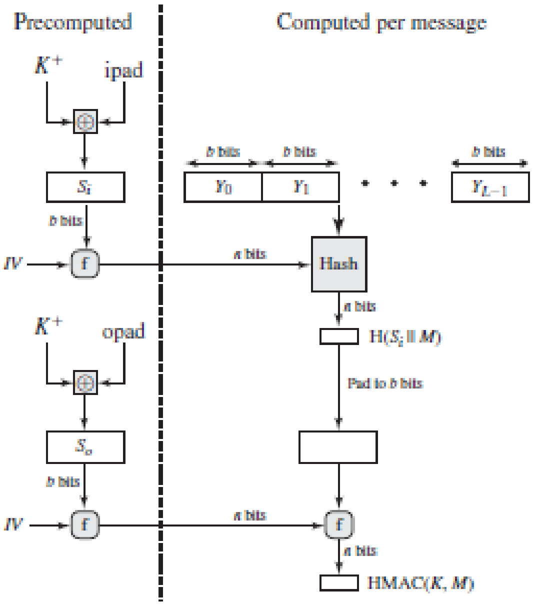


Figure 11.2 Effective Implementation of HMAC

The security of any MAC function based on an embedded hash function dependsin some way on the cryptographic strength of the underlying hash function. Theappeal of HMAC is that its designers have been able to prove an exact relationshipbetween the strength of the embedded hash function and the strength of HMAC.

11.4 MACS BASED ON BLOCK CIPHERS: DAA AND CMAC

In this section, we look at two MACs that are based on the use of a block ciphermode of operation

Data Authentication Algorithm

The Data Authentication Algorithm(DAA), based on DES, has been one of the most widely used MACs for a number of years. The algorithm is both a FIPS publication (FIPS PUB 113) and an ANSI standard (X9.17). However, as we discuss subsequently, security weaknesses in this algorithm have been discovered, and it is being replaced by newer and stronger algorithms.

The algorithm can be defined as using the cipher block chaining (CBC) mode of operation of DES with an initialization vector of zero. The data (e.g., message, record, file, or program) to be authenticated are grouped into contiguous 64-bit blocks: D_1, D_2,.....,D_N .

If necessary, the final block is padded on the right with zeroes to form a full 64-bit block. Using the DES encryption algorithm E and a secret key *K*, a data authentication code (DAC) is calculated as follows (Figure 11.3).

$$
\begin{aligned}
O_1 &= E(K,D)\\
O_2 &= E(K, [D_2 \oplus O_1])\\
O_3 &= E(K, [D_3 \oplus O_2])\\
&\ldots\ldots\\
&\ldots\ldots\\
O_N &= E(K, [D_N \oplus O_{N-1}])
\end{aligned}
$$

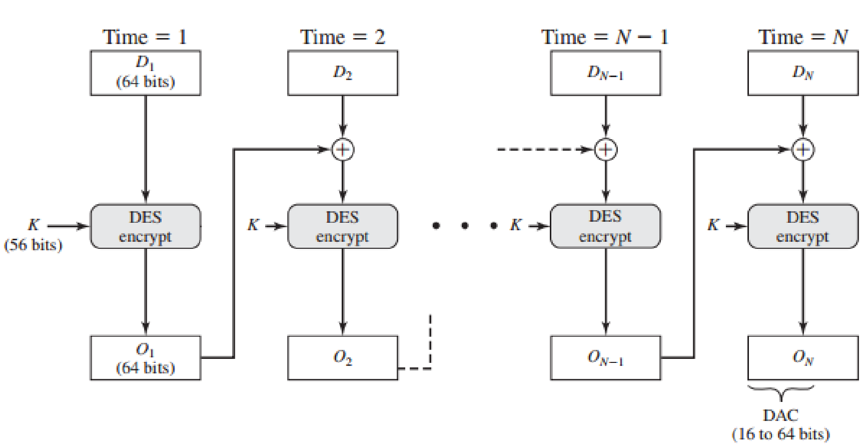


Figure 11.3 Data Authentication Algorithm (FIPS PUB 113)

The DAC consists of either the entire block O_N or the leftmost *M* block, with 16 <= *M* <= 64.

Cipher-Based Message Authentication Code (CMAC)

Cipher-based Message Authentication Code (CMAC) is mode of operation for use with AES and triple DES. It is specified in NIST Special Publication 800-38B.

First, let us define the operation of CMAC when the message is an integer multiple n of the cipher block length b. For AES, b = 128, and for triple DES, b=64. The message is divided into n blocks (M_1, M_2, ... ,M_n). The algorithm makes use of a k-bit encryption key and an n-bit constant, $K1$. For AES, the key size k is 128, 192, or 256 bits; for triple DES, the key size is 112 or 168 bits. CMAC is calculated as follows (Figure 11.4).

$$C_1 = E(K,M_1)$$

$$C_2 = E(K, [M_2 \oplus C_1])$$

$$C_3 = E(K, [M_3 \oplus C_2])$$

$$...$$

$$...$$

$$...$$

$$C_n = E(K, [M_n \oplus C_{n-1} \oplus K_1])$$

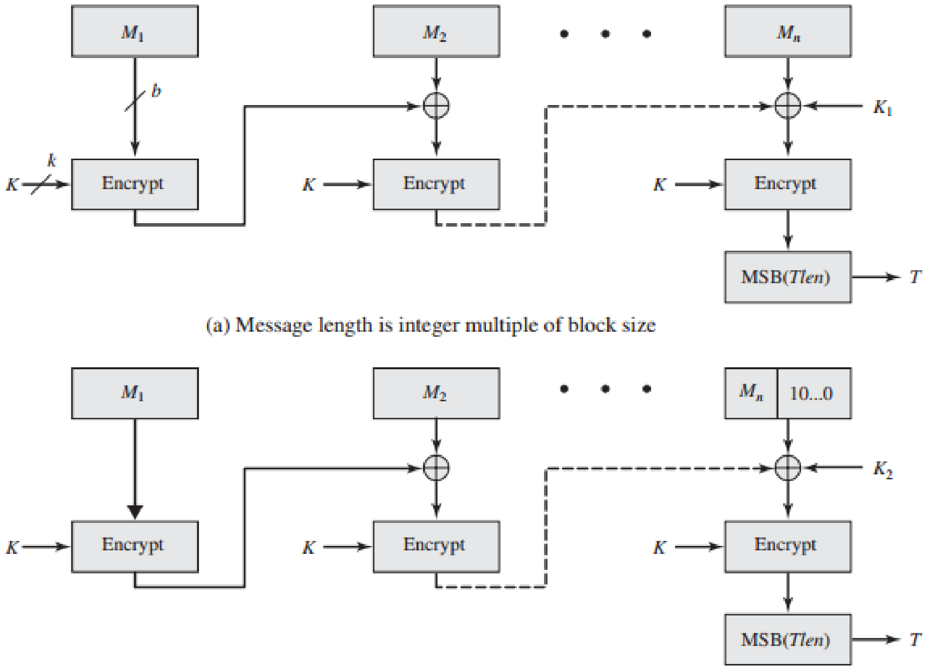
$$T = MSB_{Tlen}(C_n)$$

where

T =message authentication code, also referred to as the tag

Tlen =bit length of T

MSBs(X) =the s leftmost bits of the bit string X



(a) Message length is integer multiple of block size

(b) Message length is not integer multiple of block size

Figure 11.4 Cipher-Based Message Authentication Code (CMAC)

If the message is not an integer multiple of the cipher block length, then the final block is padded to the right (least significant bits) with a 1 and as many 0s as necessary so that the final block is also of length *b*. The CMAC operation then proceeds as before, except that a different *n*-bit key *K2* is used instead of *K1*.

Check Your Progress 1

1. What are the two categories of MAC attacks?
2. Which algorithm is most widely used in MAC
3. What is CMAC?

11.5 ANSWERS TO CHECK YOUR PROGRESS

1. We can group attacks on MACs into two categories:
   a. Brute-force attacks
   b. Cryptanalysis

2. The Data Authentication Algorithm (DAA), based on DES, has been one of the most widely used MACs for a number of years.

3. Cipher-based Message Authentication Code (CMAC) is mode of operation for use with AES and triple DES.

11.6 SUMMARY

Thisunit describes the security considerationsfor MACs and the two categories of attacks in MAC are described. Two specific MACs namely, those built from cryptographic hash functions and those built using a block cipher mode of operation are discussed in this unit

11.7 KEYWORDS

- A MAC takes a variable-length message and a secretkey as input and produces an authentication code.A recipient in possessionof the secret key can generate an authentication code to verify theintegrity of the message.
- One means of forming a MAC is to combine a cryptographic hash functionin some fashion with a secret key.
- Another approach to constructing a MAC is to use a symmetric blockcipher in such a way that it produces a fixed-length output for a variablelengthinput.

11.8 SELF-ASSESSMENT EXERCISES

NOTES

Short Questions

1. What is Brute-force attack?
2. Define the term Cryptanalysis
3. What is DAA?

Detail Questions

1. How will you secure MACs? Describe.
2. Explain in detail about the ways in which MACs can be built.

11.9 SUGGESTED READINGS

1. William Stallings, "Cryptography and Network Security Principles and Practice", Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, "Computer Networks", 5th Edition, Prentice Hall

UNIT-12 DIGITAL SIGNATURES

NOTES

Structure

12.0 INTRODUCTION

The most important development from the work on public-key cryptography is thedigital signature.The digital signature provides a set of security capabilities that wouldbe difficult to implement in any other way.We begin this unit with an overview of digital signatures.Then, we introducethe Digital Signature Standard (DSS).

12.1 OBJECTIVES

After going through this unit, you will be able to:

- Know the properties of Digital signatureand its requirements
- Understand about Digital signature
- Learn about Digital Signature Standard

12.2 PROPERTIESOF DIGITAL SIGNATURE

Figure 12.1 is a generic model of the process of making and using digital signatures.Bob can sign a message using a digital signature generation algorithm.The inputsto the algorithm are the message and Bob’s private key. Any other user, say Alice, canverify the signature using a verification algorithm, whose inputs are the message, thesignature, and Bob’s public key.

NOTES

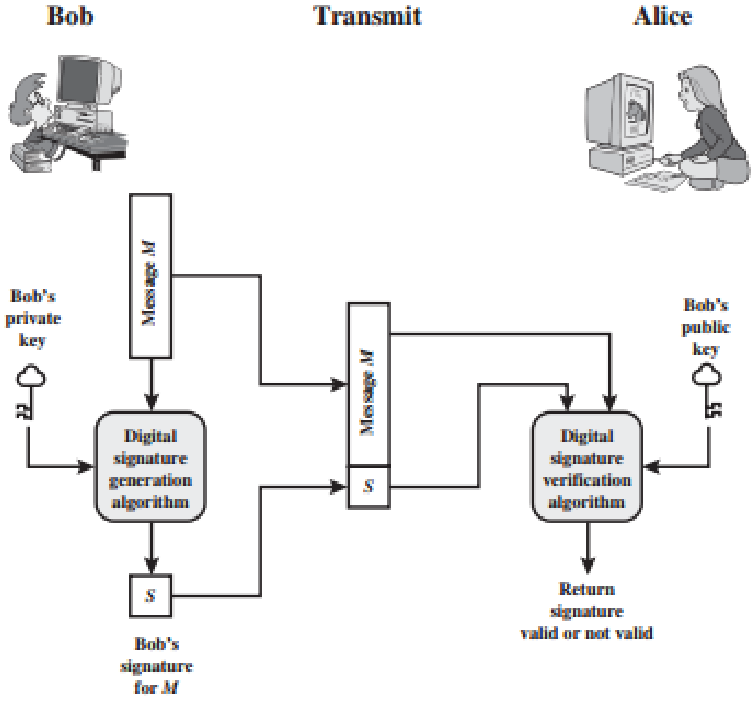

Bob
Transmit
Alice
Message M
Bob's private key
Digital signature generation algorithm
S
Bob's signature for M
Message M
S
Bob's public key
Digital signature verification algorithm
Return signature valid or not valid

Figure 12.1 Generic Model of Digital Signature Process

Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two are possible.

Consider the following disputes thatcould arise.

1. Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.

2. John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

Both scenarios are of legitimate concern.

- Here is an example of the first scenario: An electronic funds transfer takes place, and the receiver increases the amount of funds transferred and claims that the larger amount had arrived from the sender.
- An example of the second scenario is that an electronic mail message contains instructions to a stockbroker for a transaction that subsequently turns out badly. The sender pretends that the message was never sent.

In situations where there is not complete trust between sender and receiver,something more than authentication is needed. The most attractive solution tothis problem is the digital signature.

The digital signature must have the followingproperties:

- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes.

Thus, the digital signature function includes the authentication function.

In simplified terms, the essence of the digital signaturemechanism is shown in Figure 12.2.

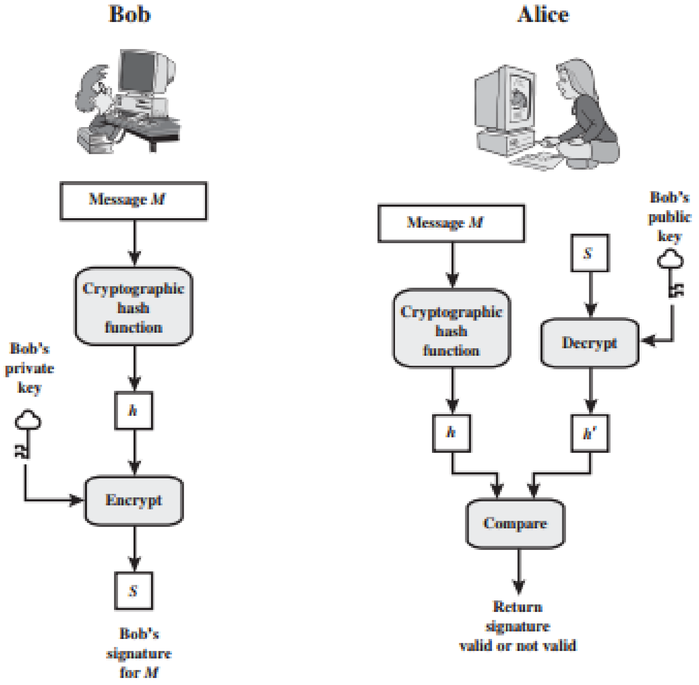


Figure 12.2 Simplified Depictions of Essential Elements of Digital Signature Process

12.3 DIGITAL SIGNATURE REQUIREMENTS

The following are the requirements for a digital signature

- The signature must be a bit pattern that depends on the message being signed.
- The signature must use some information unique to the sender to prevent both forgery and denial.
- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature.

- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be practical to retain a copy of the digital signature in storage.

A secure hash function, embedded in a scheme such as that of Figure 12.2, provides a basis for satisfying these requirements. However, care must be taken in the design of the details of the scheme.

12.4 ELGAMMAL DIGITAL SIGNATURE SCHEME

ElGamal encryption scheme is designed to enable encryption by a user's public key with decryption by the user's private key. The ElGamal signature scheme involves the use of the private key for encryption and the public key for decryption.

Before proceeding, we need a result from number theory. For a prime number *q*, if α is a primitive root of *q*, then

$$\alpha, \alpha^2, \ldots\ldots, \alpha^{q\text{-}1}$$

are distinct (mod *q*). It can be shown that, if α is a primitive root of *q*, then

1. For any integer m, αm= 1 (mod q) if and only if

   m = 0 (mod q -1).

2. For any integers i,j αi= αj (mod q), if and only if

   i=j (mod q -1).

As with ElGamal encryption, the global elements of **ElGamal digital signature** are a prime number *q* and α, which is a primitive root of *q*. User A generates a private/public key pair as follows.

1. Generate a random integer X_A, such that $1<X_A<q\text{-}1$.
2. Compute $Y_A= \alpha^{XA}$ mod *q*.
3. A's private key is X_A; A's pubic key is {q, α, YA}.

To sign a message *M*, user A first computes the hash *m* = H(*M*), such that *m* is an integer in the range 0 <= *m* <= *q*-1. A then forms a digital signature as follows.

1. Choose a random integer K such that 1<=K<=q-1 and gcd (K,q-1) = 1. That is, K is relatively prime to q-1.
2. Compute S1= αK mod q. Note that this is the same as the computation of C1 for ElGamal encryption.
3. Compute K-1 mod (q-1). That is, compute the inverse of K modulo q-1.
4. Compute S2= K-1(m - XAS1) mod (q-1).
5. The signature consists of the pair (S1, S2).

Any user B can verify the signature as follows.

1. Compute $V_1 = \alpha^m \bmod q$.
2. Compute $V_2 = (Y_A)^{S1}(S_1)^{S2} \bmod q$.

The signature is valid if V_1=V_2.

12.5 SCHNORR DIGITAL SIGNATURE SCHEME

The Schnorr scheme minimizes the message-dependent amount of computation required to generate a signature. The main work for signature generation does not depend on the message and can be done during the idle time of the processor. The message-dependent part of the signature generation requires multiplying a *2n*-bit integer with an *n*-bit integer.

The scheme is based on using a prime modulus *p*, with *p-1* having a prime factor *q* of appropriate size; that is, *p-1*$\underline{=}$ *(mod q)*. Typically, we use and *p* $\underline{\sim}$ 2^{1024} and *q* $\underline{\sim}$ 2^{160}. Thus *p*, is a 1024-bit number, and *q* is a 160-bit number, which is also the length of the SHA-1 hash value.

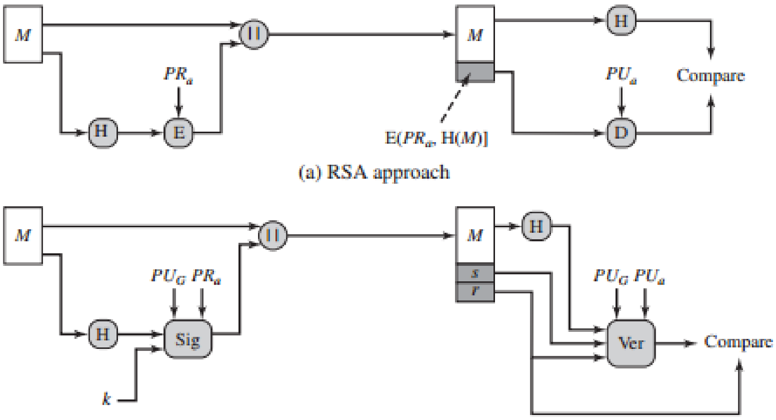
The first part of this scheme is the generation of a private/public key pair, which consists of the following steps.

1. Choose primes *p* and *q*, such that *q* is a prime factor of *p*-1.
2. Choose an integer *a*, such that α^q = 1 mod *p*. The values *a*, *p*, and *q* comprise a global public key that can be common to a group of users.
3. Choose a random integer *s* with 0 < s < q. This is the user's private key.
4. Calculate $v = a^{-s}$ mod *p*. This is the user's public key.

12.6 DIGITAL SIGNATURE STANDARD

NOTES

The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS 186, known as the Digital Signature Standard (DSS). The DSS makes use of the Secure Hash Algorithm (SHA) and presents a new digital signature technique, the Digital Signature Algorithm (DSA).



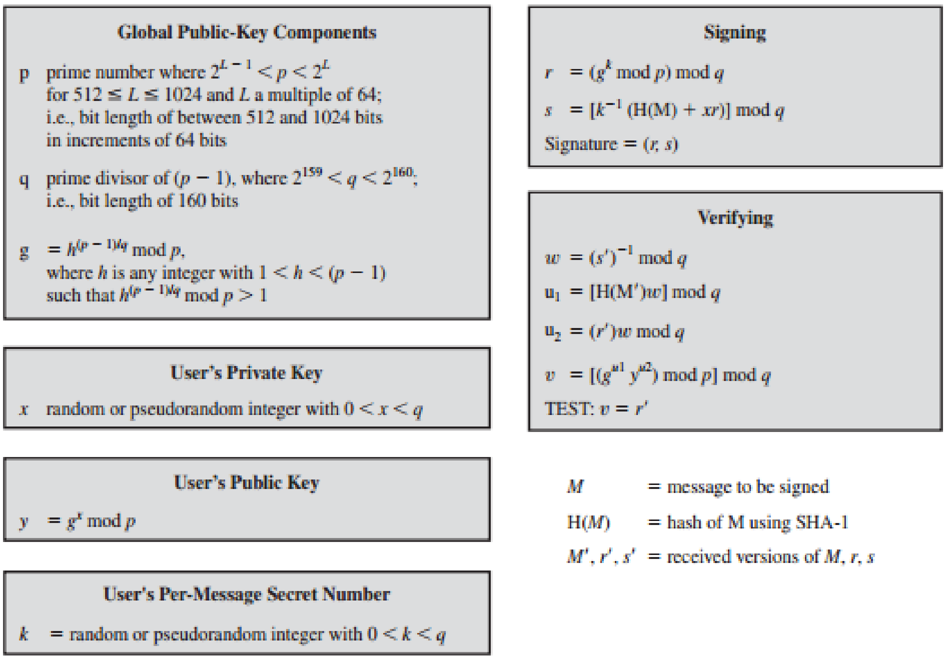
(a) RSA approach

(b) DSS approach

Figure 12.3 Two Approaches to Digital Signatures

The DSS uses an algorithm that is designed to provide only the digital signature function. Unlike RSA, it cannot be used for encryption or key exchange. Nevertheless, it is a public-key technique.

The DSA is based on the difficulty of computing discrete logarithms and is based on schemes originally presented by ElGamal and Schnorr. Figure 12.4 summarizes the algorithm.



Global Public-Key Components

p prime number where $2^{L-1} < p < 2^L$ for $512 \le L \le 1024$ and L a multiple of 64; i.e., bit length of between 512 and 1024 bits in increments of 64 bits

q prime divisor of $(p - 1)$, where $2^{159} < q < 2^{160}$; i.e., bit length of 160 bits

g $= h^{(p-1)/q} \bmod p$, where h is any integer with $1 < h < (p - 1)$ such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

x random or pseudorandom integer with $0 < x < q$

User's Public Key

$y = g^x \bmod p$

User's Per-Message Secret Number

k = random or pseudorandom integer with $0 < k < q$

Signing

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1} (H(M) + xr)] \bmod q$

Signature = (r, s)

Verifying

$w = (s')^{-1} \bmod q$

$u_1 = [H(M')w] \bmod q$

$u_2 = (r')w \bmod q$

$v = [(g^{u1} y^{u2}) \bmod p] \bmod q$

TEST: $v = r'$

M	=	message to be signed
$H(M)$	=	hash of M using SHA-1
M', r', s'	=	received versions of M, r, s

Figure 12.4 The Digital Signature Algorithm (DSA)

NOTES

Check Your Progress 1

1. State the properties of Digital Signature.
2. Expand DSS, SHA, DSA.

12.7 ANSWERS TO CHECK YOUR PROGRESS

1. The digital signature must have the following properties:
   i. It must verify the author and the date and time of the signature.
   ii. It must authenticate the contents at the time of the signature.
   iii. It must be verifiable by third parties, to resolve disputes.
2. Digital Signature Standard (DSS)
   Secure Hash Algorithm (SHA)
   Digital Signature Algorithm (DSA)

12.8 SUMMARY

In this unit the properties of Digital signatureand its requirements are clearly described. The basic concepts related to Digital signature are explained with suitable examples. In addition to this Digital Signature Standard (DSS) is discussed in this unit.

12.9 KEYWORDS

- A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.
- Typically the signature is formed by taking the hash of the message and encrypting the message with the creator’s private key. The signature guarantees the source and integrity of the message.
- The digital signature standard (DSS) is an NIST standard that uses the secure hash algorithm (SHA).

NOTES

12.10 SELF-ASSESSMENT EXERCISES

Short Questions

1. State the properties of digital signature.
2. What are the requirements of Digital Signature?
3. What is DSS?

Detail Questions

1. Explain in detail about Elgamal Digital Signature Scheme
2. Describe the Schnorr digital signature scheme
3. Discuss about DSS.

12.11 SUGGESTED READINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

NOTES

BLOCK – V
TRANSPORT LEVEL SECURITY

UNIT- 13 WEB SECURITY

Structure

13.0 INTRODUCTION

Virtually all businesses, most government agencies, and many individuals now haveWeb sites.The number of individuals and companies with Internet access is expandingrapidly and all of these have graphical Web browsers. As a result, businesses are enthusiasticabout setting up facilities on the Web for electronic commerce. But the reality isthat the Internet and the Web are extremely vulnerable to compromises of varioussorts. As businesses wake up to this reality, the demand for secure Web services grows.

13.1 OBJECTIVES

After going through this unit, you will be able to:

- Address the web security considerations
- Understand the SSL
- Describe the TLS

13.2 WEB SECURITY CONSIDERATIONS

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets. The security tools and approaches discussed earlier are relevant to the issue of Web security. The Web

NOTES

presents new challenges not generally appreciated in the context of computer and network security.

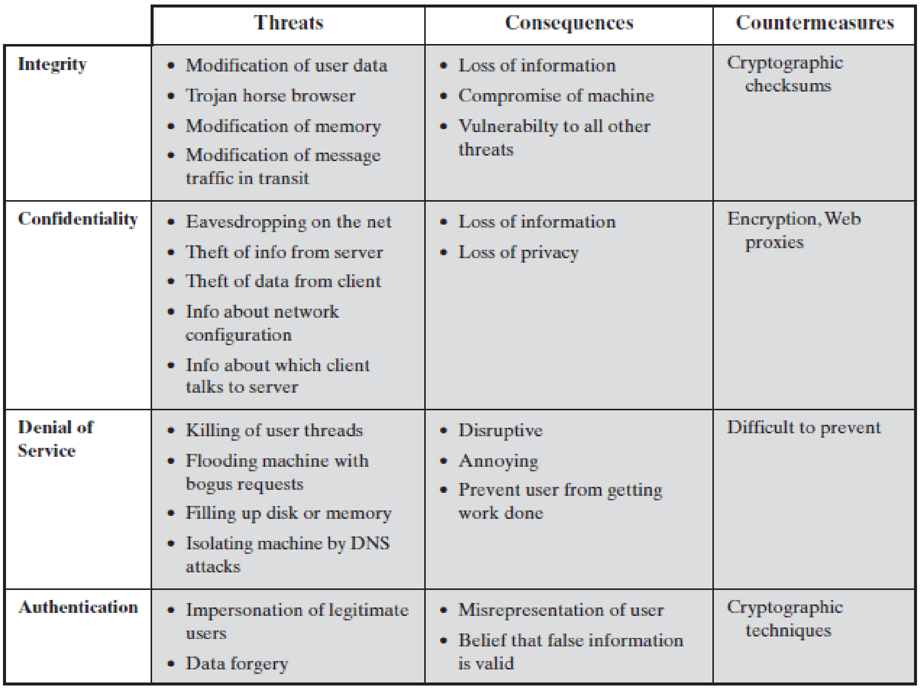
- The Internet is two-way. Unlike traditional publishing environments—even electronic publishing systems involving teletext, voice response, or fax-back— the Web is vulnerable to attacks on the Web servers over the Internet
- The Web is increasingly serving as a highly visible outlet for corporate and product information and as the platform for business transactions. Reputations can be damaged and money can be lost if the Web servers are subverted.
- Although Web browsers are very easy to use, Web servers are relatively easy to configure and manage, and Web content is increasingly easy to develop, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws. The short history of the Web is filled with examples of new and upgraded systems, properly installed, that are vulnerable to a variety of security attacks.
- A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex. Once the Web server is subverted, an attacker may be able to gain access to data and systems not part of the Web itself but connected to the server at the local site.
- Casual and untrained (in security matters) users are common clients for Web-based services. Such users are not necessarily aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures.

Web security Threats

Table 13.1 provides a summary of the types of security threats faced when using the Web.

Table13.1 Comparison of Threats on the Web

NOTES



	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping on the net • Theft of info from server • Theft of data from client • Info about network configuration • Info about which client talks to server	• Loss of information • Loss of privacy	Encryption, Web proxies
Denial of Service	• Killing of user threads • Flooding machine with bogus requests • Filling up disk or memory • Isolating machine by DNS attacks	• Disruptive • Annoying • Prevent user from getting work done	Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	Cryptographic techniques

One way to group these threats is in terms of ***passive*** and ***active*** attacks.

- ✓ Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.
- ✓ Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.

Another way to classify Web security threats is in terms of the location of the threat:

1. Web server
2. Web browser
3. Network traffic between browser and server.

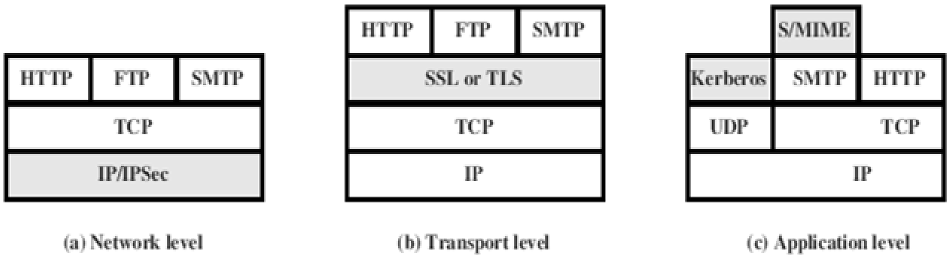
Web Traffic Security Approaches

A number of approaches to providing Web security are possible. The variousapproaches that have been considered are similar in the services they provide and,to some extent, in the mechanisms that they use, but they differ with respect to theirscope of applicability and their relative location within the TCP/IP protocol stack.

Figure 13.1 illustrates this difference. The possible ways to provide Web Security are

- To use IP security (IPsec) (Figure 13.1 (a))
  - ✓ The advantage of using IPsec is that it is transparentto end users and applications and provides a general-purpose solution.

  - ✓ Furthermore, IPsec includes a filtering capability so that only selected traffic needincur the overhead of IPsec processing
- To implement security just above TCP (Figure 13.1 (b))
  - ✓ The foremost example of this approach is the Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport LayerSecurity (TLS).
  - ✓ At this level, there are two implementation choices.
  - ✓ For full generality,SSL (or TLS) could be provided as part of the underlying protocol suite andtherefore be transparent to applications.
  - ✓ Alternatively, SSL can be embedded inspecific packages.
  - ✓ For example, Netscape and Microsoft Explorer browsers comeequipped with SSL, and most Web servers have implemented the protocol
- To embed within the particular application (Figure 13.1 (c))
  - ✓ Figure 13.1c shows examples of this architecture.
  - ✓ The advantage of thisapproach is that the service can be tailored to the specific needs of a given application.



(a) Network level

(b) Transport level

(c) Application level

Figure 13.1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

13.3 SECURE SOCKET LAYER

SSL is designed to make use of TCP to provide a reliable end-to-end secure service.SSL is not a single protocol but rather two layers of protocols, as illustrated in Figure 13.2.

The SSL Record Protocol provides basic security services to various higher layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol. These SSL-specific protocols are used in the management of SSL exchanges.

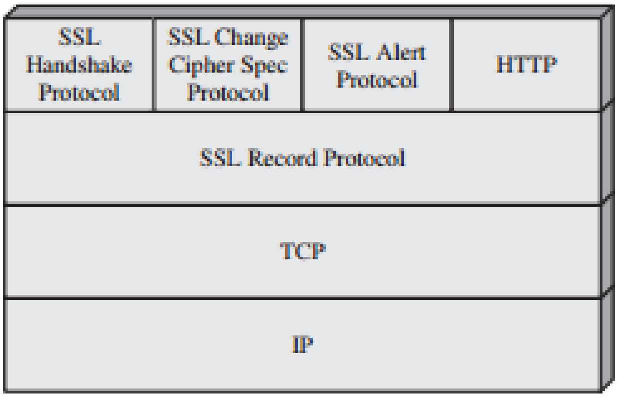


Figure 13.2 SSL Protocol Stack

Two important SSL concepts are the SSL session and the SSL connection, which are defined in the specification as follows.

Connection:

A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

Session:

An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Between any pair of parties (applications such as HTTP on client and server),there may be multiple secure connections. In theory, there may also be multiplesimultaneous sessions between parties, but this feature is not used in practice.

There are a number of states associated with each session. Once a session isestablished, there is a current operating state for both read and write (i.e., receiveand send). In addition, during the Handshake Protocol, pending read and writestates are created. Upon successful conclusion of the Handshake Protocol, thepending states become the current states.

A session state is defined by the following parameters.

- **Session identifier:** An arbitrary byte sequence chosen by the server to identify an active or resumable session state.

- **Peer certificate:** An X509.v3 certificate of the peer. This element of the state may be null.

- **Compression method:** The algorithm used to compress data prior to encryption.

- **Cipher spec:** Specifies the bulk data encryption algorithm (such as null, AES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also defines cryptographic attributes such as the hash_size.

- **Master secret:** 48-byte secret shared between the client and server.

- **Is resumable:** A flag indicating whether the session can be used to initiate new connections. A connection state is defined by the following parameters.

- **Server and client random:** Byte sequences that are chosen by the server and client for each connection.

- **Server write MAC secret:** The secret key used in MAC operations on data sent by the server.

- **Client write MAC secret:** The secret key used in MAC operations on data sent by the client.

- **Server write key:** The secret encryption key for data encrypted by the server and decrypted by the client.

- **Client write key:** The symmetric encryption key for data encrypted by the client and decrypted by the server.

- **Initialization vectors:** When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter, the final ciphertext block from each record is preserved for use as the IV with the following record.

- **Sequence numbers:** Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed $2^{64} - 1$.

The SSL Record Protocol provides two services for SSL connections:

- **Confidentiality:** The Handshake Protocol defines a shared secret key that isused for conventional encryption of SSL payloads.
- **Message Integrity:** The Handshake Protocol also defines a shared secret keythat is used to form a message authentication code (MAC).

Figure 13.2 indicates the overall operation of the SSL Record Protocol. TheRecord Protocol takes an application message to be transmitted, fragments the datainto manageable blocks, optionally compresses the data, applies a MAC, encrypts,adds a header, and transmits the resulting unit in a TCP segment. Received data aredecrypted, verified, decompressed, and reassembled before being delivered tohigher-level users.

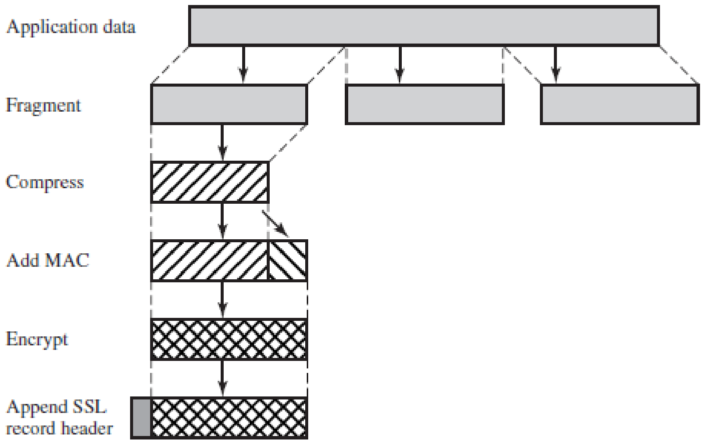


Figure 13.2.SSL Record Protocol Operation

The final step of SSL Record Protocol processing is to prepare a headerconsisting of the following fields:

- **Content Type (8 bits):** The higher-layer protocol used to process the enclosed fragment.
- **Major Version (8 bits):** Indicates major version of SSL in use. For SSLv3, the value is 3.
- **Minor Version (8 bits):** Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits):** The length in bytes of the plaintext fragment (or compressed fragment if compression is used).The maximum value is 2^{14} + 2048.

Figure 13.3 illustrates the SSL record format.

NOTES

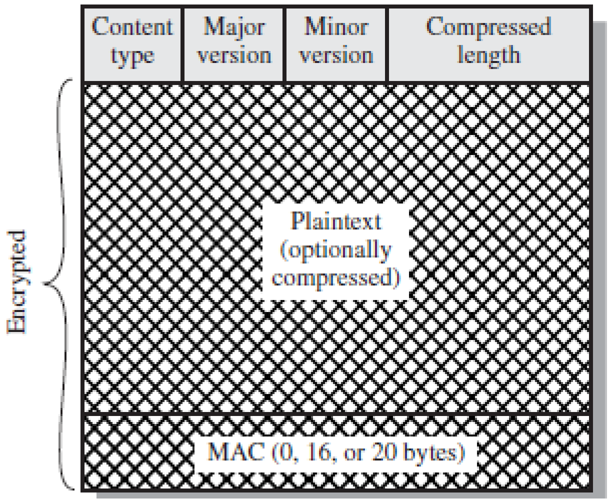


Figure 13.3.SSL Record Format

13.4 TRANSPORT LAYER SECURITY

Transport layer security (TLS) is a protocol that provides communication security between client/server applications that communicate with each other over the Internet. It enables privacy, integrity and protection for the data that's transmitted between different nodes on the Internet. TLS is a successor to the secure socket layer (SSL) protocol.

The TLS Record Format is the same as that of the SSL Record Format(Figure 13.3), and the fields in the header have the same meanings.The one differenceis in version values. For the current version of TLS, the major version is 3 andthe minor version is 3.

TLS primarily enables secure Web browsing, applications access, data transfer and most Internet-based communication. It prevents the transmitted/transported data from being eavesdropped or tampered. TLS is used to secure Web browsers, Web servers, VPNs, database servers and more. TLS protocol consists of two different layers of sub-protocols:

- TLS Handshake Protocol: Enables the client and server to authenticate each other and select an encryption algorithm prior to sending the data
- TLS Record Protocol: It works on top of the standard TCP protocol to ensure that the created connection is secure and reliable. It also provides data encapsulation and data encryption services.

13.5 COMPARISON OF TLS AND SSL PROTOCOLS

NOTES

There are main eight differences between TLS and SSLv3 protocols. These are as follows.

Protocol Version − The header of TLS protocol segment carries the version number 3.1 to differentiate between number 3 carried by SSL protocol segment header.

Message Authentication − TLS employs a keyed-hash message authentication code (H-MAC). Benefit is that H-MAC operates with any hash function, not just MD5 or SHA, as explicitly stated by the SSL protocol.

Session Key Generation − There are two differences between TLS and SSL protocol for generation of key material.

- Method of computing pre-master and master secrets is similar. But in TLS protocol, computation of master secret uses the HMAC standard and pseudorandom function (PRF) output instead of ad-hoc MAC.
- The algorithm for computing session keys and initiation values (IV) is different in TLS than SSL protocol.

Alert Protocol Message

- TLS protocol supports all the messages used by the Alert protocol of SSL, except No certificate alert message being made redundant. The client sends empty certificate in case client authentication is not required.
- Many additional Alert messages are included in TLS protocol for other error conditions such as record_overflow, decode_error etc.

Supported Cipher Suites − SSL supports RSA, Diffie-Hellman and Fortezza cipher suites. TLS protocol supports all suits except Fortezza.

Client Certificate Types − TLS defines certificate types to be requested in a certificate_request message. SSLv3 support all of these. Additionally, SSL support certain other types of certificate such as Fortezza.

CertificateVerify and Finished Messages −

- ✓ In SSL, complex message procedure is used for the certificate_verify message. With TLS, the verified information is contained in the handshake messages itself thus avoiding this complex procedure.

- ✓ Finished message is computed in different manners in TLS and SSLv3.

Padding of Data − In SSL protocol, the padding added to user data before encryption is the minimum amount required to make the total data-size equal to a multiple of the cipher's block length. In TLS, the padding can be any amount that results in data-size that is a multiple of the cipher's block length, up to a maximum of 255 bytes.

Check Your Progress 1

1. What are the possible ways to provide Web Security?
2. Name the services offered by SSL
3. What is TLS?

13.6 ANSWERS TO CHECK YOUR PROGRESS

1. The possible ways to provide Web Security are
   - To use IP security (IPsec)
   - To implement security just above TCP
   - To embed within the particular application

2. The SSL Record Protocol provides two services for SSL connections:
   - Confidentiality
   - Message Integrity

3. Transport layer security (TLS) is a protocol that provides communication security between client/server applications that communicate with each other over the Internet. It enables privacy, integrity and protection for the data that's transmitted between different nodes on the Internet

13.7 SUMMARY

In this unit, we begin with a discussion of the general requirements for Web security and then focus on three standardized schemes that are becoming increasingly important as part of Web commerce and that focus on security at the transport layer.

13.8 KEYWORDS

- Secure Socket Layer (SSL) provides security services between TCP and applications that use TCP. The Internet standard version is called Transport Layer Service (TLS).

NOTES

- SSL/TLS provides confidentiality using symmetric encryption and message integrity using a message authentication code.
- SSL/TLS includes protocol mechanisms to enable two TCP users to determine the security mechanisms and services they will use.

13.9 SELF-ASSESSMENT EXERCISES

Short Questions

1. What are Web Security threats?
2. State the approaches to provide web security.
3. What is connection?
4. What do you mean by session?
5. What are the services offered by SSL?

Detail Questions

1. Describe about Security Socket Layer.
2. Write short notes on Transport Layer Security.
3. Compare TLS and SSL protocols

13.10 SUGGESTED READINGS

1. William Stallings, "Cryptography and Network Security Principles and Practice", Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, "Computer Networks", 5th Edition, Prentice Hall

NOTES

UNIT -14 E-MAIL AND IP SECURITY

Structure

14.0 INTRODUCTION

There are application-specific security mechanisms for a number of applicationareas, including electronic mail (S/MIME, PGP), client/server (Kerberos),Webaccess (Secure Sockets Layer), and others. However, users have security concernsthat cut across protocol layers. For example, an enterprise can run a secure, privateIP network by disallowing links to untrusted sites, encrypting packets that leave thepremises and authenticating packets that enter the premises. By implementingsecurity at the IP level, an organization can ensure secure networking not only forapplications that have security mechanisms but also for the many security-ignorantapplications.

14.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand about E-mail Security
- Understand the IP Security

14.2 E-MAIL

In virtually all distributed environments, electronic mail is the most heavily used network-based application. Users expect to be able to, and do, send e-mail to others who are connected directly or indirectly to the Internet, regardless of host operating system or communications suite. With the explosively growing reliance on e-mail, there grows a demand for authentication and confidentiality services. Two schemes stand out as approaches that enjoy widespread use: Pretty Good Privacy (PGP) and S/MIME. PGP is explained in this unit.

NOTES

14.3 PRETTY GOOD PRIVACY

Pretty Good Privacy (PGP) is a remarkable phenomenon. Largely the effort of a single person, Phil Zimmermann, PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications. In essence, Zimmermannhas done the following:

1. Selected the best available cryptographic algorithms as building blocks.
2. Integrated these algorithms into a general-purpose application that is independentof operating system and processor and that is based on a small set ofeasy-to-use commands.
3. Made the package and its documentation, including the source code, freelyavailable via the Internet, bulletin boards, and commercial networks such asAOL (America On Line).
4. Entered into an agreement with a company (Viacrypt, now NetworkAssociates) to provide a fully compatible, low-cost commercial version of PGP.

PGP has grown explosively and is now widely used.A number of reasons canbe cited for this growth.

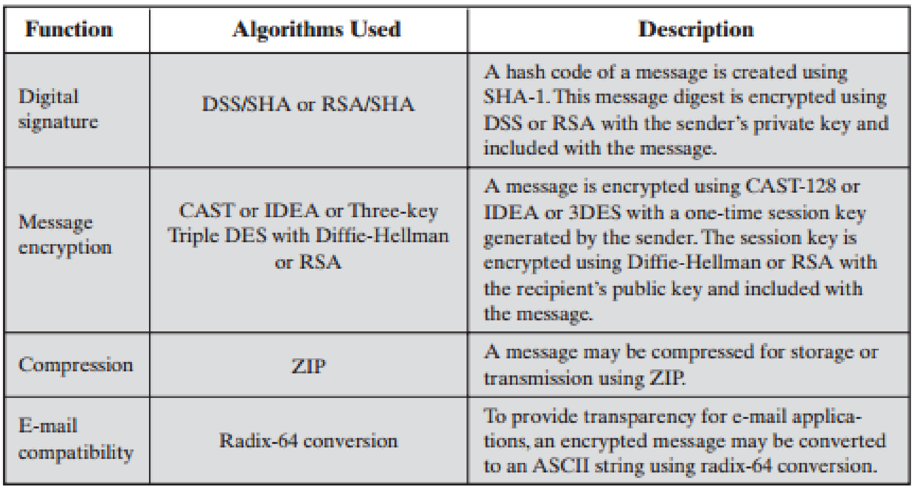
- It is available free worldwide in versions that run on a variety of platforms, including Windows, UNIX, Macintosh, and many more. In addition, the commercial version satisfies users who want a product that comes with vendor support.
- It is based on algorithms that have survived extensive public review and are considered extremely secure. Specifically, the package includes RSA, DSS, and Diffie-Hellman for public-key encryption; CAST-128, IDEA, and 3DES for symmetric encryption; and SHA-1 for hash coding.
- It has a wide range of applicability, from corporations that wish to select and enforce a standardized scheme for encrypting files and messages to individuals who wish to communicate securely with others worldwide over the Internet and other networks.
- It was not developed by, nor is it controlled by, any governmental or standards organization. For those with an instinctive distrust of “the establishment,” this makes PGP attractive.

- PGP is now on an Internet standards track (RFC 3156; MIME Security with OpenPGP).

NOTES

The actual operation of PGP, as opposed to the management of keys, consists of fourservices: authentication, confidentiality, compression, and e-mail compatibility.

Table 14. 1 Summary of PGP Services



Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

Authentication

1. The sender creates a message.
2. SHA-1 is used to generate a 160-bit hash code of the message.
3. The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.
4. The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
5. The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

Confidentiality

1. The sender generates a message and a random 128-bit number to be used asa session key for this message only.
2. The message is encrypted using CAST-128 (or IDEA or 3DES) with the sessionkey.
3. The session key is encrypted with RSA using the recipient's public key and isprepended to the message.
4. The receiver uses RSA with its private key to decrypt and recover the sessionkey.
5. The session key is used to decrypt the message

NOTES

Compression

1. The signature is generated before compression for two reasons:
   a. It is preferable to sign an uncompressed message so that one can store onlythe uncompressed message together with the signature for future verification.If one signed a compressed document, then it would be necessary eitherto store a compressed version of the message for later verification or torecompress the message when verification is required.
   b. Even if one were willing to generate dynamically a recompressed messagefor verification, PGP's compression algorithm presents a difficulty.The algorithmis not deterministic; various implementations of the algorithm achievedifferent tradeoffs in running speed versus compression ratio and, as a result,produce different compressed forms. However, these different compressionalgorithms are interoperable because any version of the algorithm cancorrectly decompress the output of any other version. Applying the hashfunction and signature after compression would constrain all PGP implementationsto the same version of the compression algorithm.
2. Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than theoriginal plaintext, cryptanalysis is more difficult.

E-mail Compatibility

When PGP is used, at least part of the block to be transmittedis encrypted. If only the signature service is used, then the message digest isencrypted (with the sender's private key). If the confidentiality service is used, themessage plus signature (if present) are encrypted (with a one-time symmetric key).Thus, part or the entire resulting block consists of a stream of arbitrary 8-bit octets.However, many electronic mail systems only permit the use of blocks consisting ofASCII text.To accommodate this restriction, PGP provides the service of convertingthe raw 8-bit binary stream to a stream of printable ASCII characters

A ***message*** consists of three components: the message component, a signature(optional), and a session key component (optional).

The ***message component*** includes the actual data to be stored or transmitted,as well as a filename and a timestamp that specifies the time of creation.

The ***signature component*** includes the following.

- ✓ <u>*Timestamp:*</u> The time at which the signature was made.
- ✓ <u>*Message digest:*</u> The 160-bit SHA-1 digest encrypted with the sender's privatesignature key. The digest is calculated over the signature timestamp concatenatedwith the data portion of the message component. The inclusion of thesignature timestamp in the digest insures against replay types of attacks. Theexclusion of the filename and timestamp portions of the message componentensures that detached signatures are exactly the same as attached signatures prefixed to the message. Detached signatures are calculated on a separate filethat has none of the message component header fields.
- ✓ <u>*Leading two octets of message digest:*</u> Enables the recipient to determine ifthe correct public key was used to decrypt the message digest for authenticationby comparing this plaintext copy of the first two octets with the first twooctets of the decrypted digest. These octets also serve as a 16-bit frame checksequence for the message.
- ✓ <u>*Key ID of sender's public key:*</u> Identifies the public key that should be used todecrypt the message digest and, hence, identifies the private key that was usedto encrypt the message digest.

The message component and optional signature component may be compressedusing ZIP and may be encrypted using a session key

The ***session key component*** includes the session key and the identifier of therecipient's public key that was used by the sender to encrypt the session key.

The entire block is usually encoded with radix-64 encoding.

14.4 IP SECURITY

IP-level security encompasses three functional areas: authentication, confidentiality, and key management. The authentication mechanism assures that a received packet was, in fact, transmitted by the party identified as the source in the packet header. In addition, this mechanism assures that the packet has not been altered in transit. The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties. The key management facility is concerned with the secure exchange of keys.

14.5 IP SECURITY OVERVIEW

In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636). The report identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IPv4 and the future IPv6. This means that vendors can begin offering these features now, and many vendors now do have some IPsec capability in their products. The IPsec specification now exists as a set of Internet standards.IPsec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet.

Figure 14.1 is a typical scenario of IPsec usage. An organization maintains LANs at dispersed locations. Non secure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPsec protocols are used. These protocols operate in networking devices, such as a router or firewall that connect each LAN to the outside world. The IPsec networking device will typically encrypt and compress all traffic going into the WAN and decrypt and decompress traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN. Secure transmission is also possible with individual users who dial into the WAN. Such user workstations must implement the IPsec protocols to provide security.

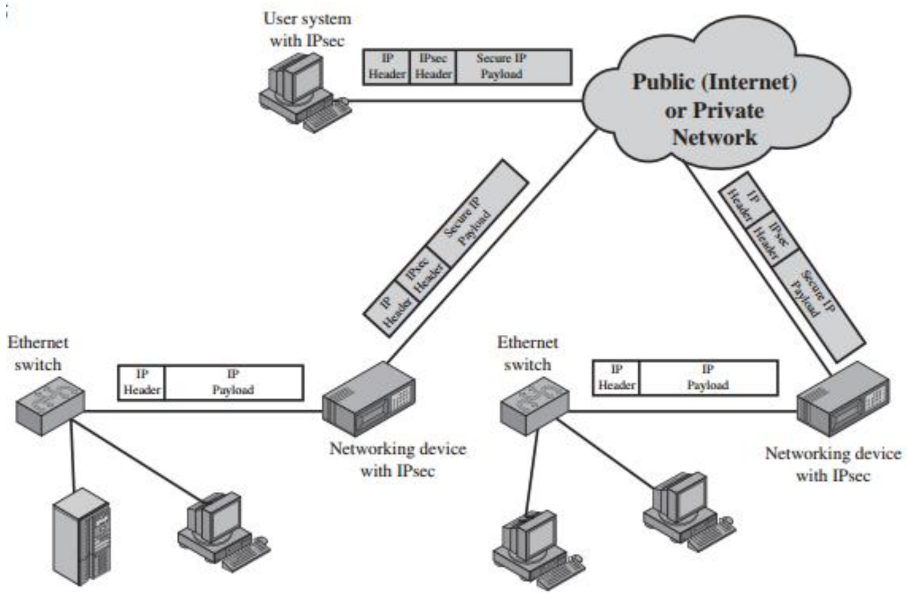


Figure 14.1 An IP Security Scenario

14.6 IP SECURITY POLICY

Fundamental to the operation of IPsec is the concept of a security policy applied to each IP packet that transits from a source to a destination. IPsec policy is determined primarily by the interaction of two databases, the security association database (SAD) and the security policy database (SPD). This section provides an overview of these two databases and then summarizes their use during IPsec operation. Figure 19.2 illustrates the relevant relationships.

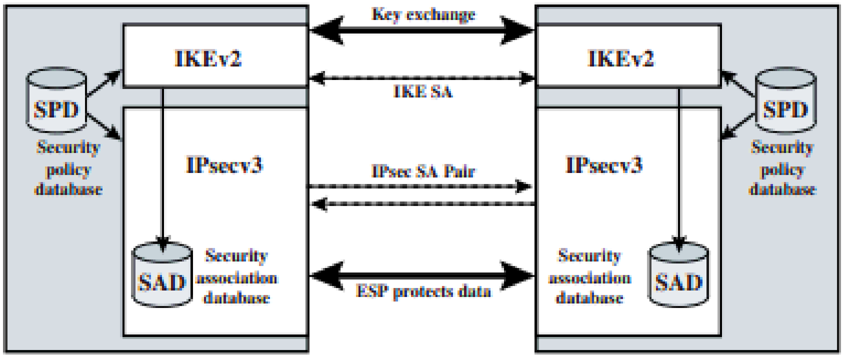


Figure 14.2 IPsec Architecture

- ✓ ***Security Parameters Index (SPI):*** A bit string assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
- ✓ ***IP Destination Address:*** This is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- ✓ ***Security Protocol Identifier:*** This field from the outer IP header indicates whether the association is an AH or ESP security association.

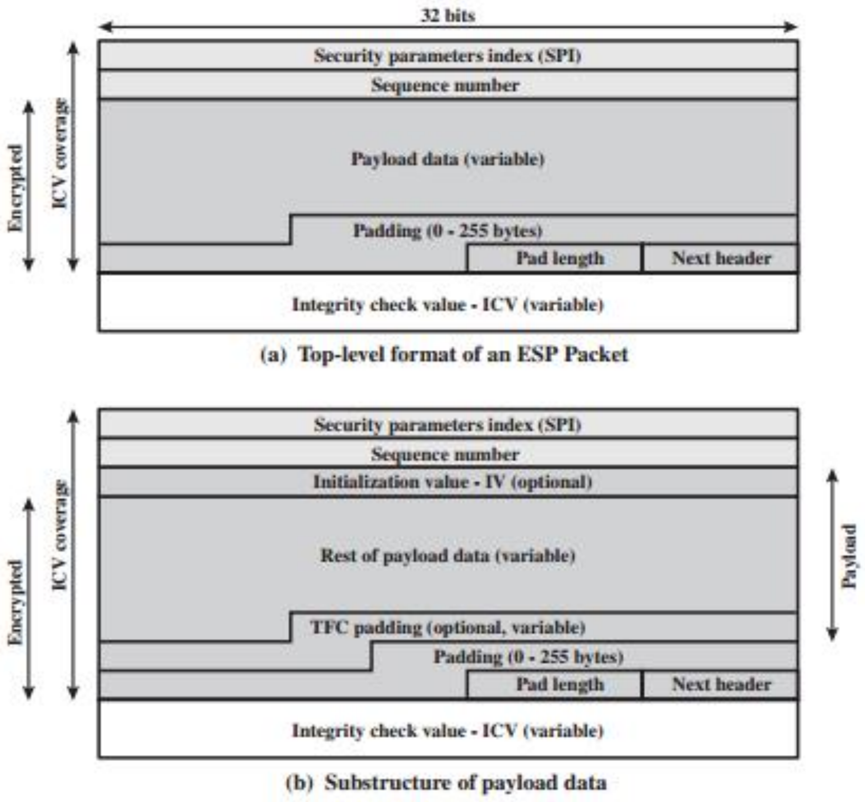
14.7 ENCAPSULATING SECURITY PAYLOAD

ESP can be used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and (limited) traffic flow confidentiality. The set of services provided depends on options selected at the time of Security Association (SA) establishment and on the location of the implementation in a network topology.

ESP can work with a variety of encryption and authentication algorithms, including authenticated encryption algorithms such as GCM. Figure 14.3 a shows the top-level format of an ESP packet.

It contains the following fields.

- *Security Parameters Index (32 bits):* Identifies a security association.
- *Sequence Number (32 bits):* A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.
- *Payload Data (variable):* This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
- *Padding (0 – 255 bytes):* The purpose of this field is discussed later.
- *Pad Length (8 bits):* Indicates the number of pad bytes immediately preceding this field.
- *Next Header (8 bits):* Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP)
- *Integrity Check Value (variable):* A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.


32 bits
Encrypted
ICV coverage
Security parameters index (SPI)
Sequence number
Payload data (variable)
Padding (0 - 255 bytes)
Pad length
Next header
Integrity check value - ICV (variable)
Security parameters index (SPI)
Sequence number
Initialization value - IV (optional)
Rest of payload data (variable)
TFC padding (optional, variable)
Padding (0 - 255 bytes)
Pad length
Next header
Integrity check value - ICV (variable)
Encrypted
ICV coverage
Payload

(a) Top-level format of an ESP Packet

(b) Substructure of payload data

Figure 14.3 ESP Packet Format

Two additional fields may be present in the payload (Figure 14.3 b). An initialization value (IV), or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for ESP. If tunnel mode is being used, then the IPsec implementation may add traffic flow confidentiality (TFC)padding after the Payload Data and before the Padding field, as explained subsequently.

NOTES

Check Your Progress 1

1. What are the operations involved in PGP?
2. List the components of message.
3. What is SPI?
4. Define IP Destination Address.
5. What do you mean by Security Protocol Identifier?

14.8 ANSWERS TO CHECK YOUR PROGRESS

1. The actual operation of PGP, as opposed to the management of keys, consists of four services: authentication, confidentiality, compression, and e-mail compatibility

2. A ***message*** consists of three components: the message component, a signature (optional), and a session key component (optional).

3. ***Security Parameters Index (SPI):*** A bit string assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.

4. ***IP Destination Address:*** This is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.

5. ***Security Protocol Identifier:*** This field from the outer IP header indicates whether the association is an AH or ESP security association

14.9 SUMMARY

This unit describes about the need for Electronic Mail Security. It addresses the Pretty Good Privacy aspects for E-Mail security. In addition to that the IP security components, its policy and architecture are explained in this unit. Finally the Encapsulating Security Payload (ESP) is described.

14.10 KEYWORDS

- PGP is an open-source, freely available software package for e-mail security. It provides authentication through the use of digital signature, confidentiality through the use of symmetric

NOTES

block encryption, compression using the ZIP algorithm, and e-mail compatibility using the radix-64 encoding scheme.

- PGP incorporates tools for developing a public-key trust model and public-key certificate management.
- IP security (IPsec) is a capability that can be added to either current versionof the Internet Protocol (IPv4 or IPv6) by means of additional headers.
- IPsec encompasses three functional areas: authentication, confidentiality,and key management.

14.11 SELF-ASSESSMENT EXERCISES

Short Questions

1. List the operations of PGP.
2. What are the components of a message?
3. What do you mean by IP security?
4. What are the fields of ESP?

Detail Questions

1. Discuss about E-mail Security
2. Explain in detail about PGP.
3. Describe the IP Security Overview
4. Write short notes on IP Security Policy.

14.12 SUGGESTED READINGS

1. William Stallings, “Cryptography and Network Security Principles and Practice”, Pearson, 5th Edition
2. Behrouz A Forouzan, Cryptography and Network security, McGraw Hill
3. Andrew S Tanenbaum, “Computer Networks”, 5th Edition, Prentice Hall

MODEL QUESTION PAPER

DISTANCE EDUCATION

M. Sc DEGREE EXAMINATION

341 31 – CRYPTOGRAPHY AND NETWORK SECURITY

Third Semester

(CBCS – 2018-19 Academic Year Onwards)

Time : 3 hours **Max Marks :75**

PART - A (10 x 2=20 Marks)

Answer all questions.

1. Define the term Network Security

2. What do you mean by Security Attacks?

3. What is the strength of DES?

4. List the most commonly used cryptanalysis.

5. What do you mean by public key cryptography?

6. Differentiate private and public key cryptography.

7. What is MAC?

8. Define the term Digital Signature.

9. List the operations of PGP.

10. Expand SSL and ESP.

PART - B (5 x 5 Marks = 25 Marks)

Answer all questions choosing either (a) or (b)

11.a). Write a note on Security Mechanisms.

OR

11. b). Describe the model for Network Security.

12.a). What do you mean by cryptanalysis? Explain.

OR

12. b). Write a note on block cipher design principles.

13.a). Describe the principles of public key cryptography.

OR

13. b). Discuss about Elgamal Cryptography.

14.a). Explain about Message authentication codes.

OR

14. b). Describe Digital Signature Standard.

15.a). Explain the Web Security Considerations.

NOTES

OR

15. b). Write a note on IP Security Policy.

Part – C (3 x 10 = 30 Marks)

Answer any three questions.

16. Explain in detail about Classical Encryption Techniques.

17. Explain about AES.

18. Elaborate on Elliptic Curve Cryptography.

19. Describe the Security of MACs.

20. Describe about PGP.

****