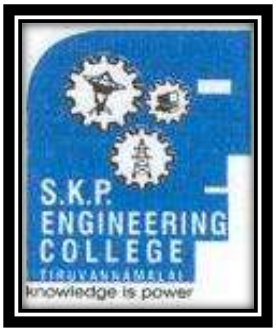
SKP Engineering College

Tiruvannamalai – 606611

A Course Material

on

Cryptography and Network Security



By

G.Rajkumar

Assistant Professor

Electronics and Communication Engineering

Quality Certificate

This is to Certify that the Electronic Study Material

Subject Code:CS 6701

Subject Name: Cryptography and Network Security

Year/Sem: IV / VIII

Being prepared by me and it meets the knowledge requirement of the University curriculum.

Signature of the Author

Name: G.Rajkumar

Designation: Assistant Professor

This is to certify that the course material being prepared by Mr.G.Rajkumar is of the adequate quality. He has referred more than five books and one among them is from abroad author.

Signature of HD

Name:

Seal:

Signature of the Principal

Name: Dr.V.Subramania Bharathi

Seal:

CS6701 CRYPTOGRAPHY AND NETWORK SECURITY L T P C 3 0 0 3

OBJECTIVES:

The student should be made to:

- Understand OSI security architecture and classical encryption techniques.
- Acquire fundamental knowledge on the concepts of finite fields and number theory.
- Understand various block cipher and stream cipher models.
- Describe the principles of public key cryptosystems, hash functions and digital signature.

UNIT I INTRODUCTION & NUMBER THEORY 10

Services, Mechanisms and attacks-the OSI security architecture-Network security model-Classical Encryption techniques (Symmetric cipher model, substitution techniques, transposition techniques, steganography).FINITE FIELDS AND NUMBER THEORY: Groups, Rings, Fields-Modular arithmetic- Euclid's algorithm-Finite fields-Polynomial Arithmetic –Prime numbers-Fermat's and Euler's theorem- Testing for primality -The Chinese remainder theorem- Discrete logarithms.

UNIT II BLOCK CIPHERS & PUBLIC KEY CRYPTOGRAPHY 10

Data Encryption Standard-Block cipher principles-block cipher modes of operation-Advanced Encryption Standard (AES)-Triple DES-Blowfish-RC5 algorithm. Public key cryptography: Principles of public key cryptosystems-The RSA algorithm-Key management – Diffie Hellman Key exchange- Elliptic curve arithmetic-Elliptic curve cryptography.

UNIT III HASH FUNCTIONS AND DIGITAL SIGNATURES 8

Authentication requirement – Authentication function – MAC – Hash function – Security of hash function and MAC –MD5 – SHA – HMAC – CMAC – Digital signature and authentication protocols – DSS – EI Gamal – Schnorr.

UNIT IV SECURITY PRACTICE & SYSTEM SECURITY 8

Authentication applications – Kerberos – X.509 Authentication services – Internet Firewalls for Trusted System: Roles of Firewalls – Firewall related terminology- Types of Firewalls – Firewall designs – SET for E-Commerce Transactions. Intruder – Intrusion detection system – Virus and related threats – Countermeasures – Firewalls design principles – Trusted systems – Practical implementation of cryptography and security.

UNIT V E-MAIL, IP & WEB SECURITY 9

E-mail Security: Security Services for E-mail-attacks possible through E-mail – establishing keys privacy-authentication of the source-Message Integrity-Non-repudiation-Pretty Good Privacy-S/MIME. IPSecurity: Overview of IPSec – IP and IPv6-Authentication Header-Encapsulation Security Payload (ESP)-Internet Key Exchange (Phases of IKE, ISAKMP/IKE Encoding). Web Security: SSL/TLS Basic Protocol-computing the keys- client authentication-PKI as deployed by SSLAttacks fixed in v3-Exportability-Encoding-Secure Electronic Transaction (SET).

TOTAL: 45 PERIODS

OUTCOMES:

Upon Completion of the course, the students should be able to:

Compare various Cryptographic Techniques
Design Secure applications
Inject secure coding in the developed applications

TEXT BOOKS:

1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013. (UNIT I,II,III,IV).
2. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security”, Prentice Hall of India, 2002. (UNIT V).

REFERENCES:

1. Behrouz A. Ferouzan, “Cryptography & Network Security”, Tata Mc Graw Hill, 2007.

2. Man Young Rhee, “Internet Security: Cryptographic Principles”, “Algorithms and Protocols”, Wiley Publications, 2003.

CONTENTS

Unit – I

Introduction & Number Theory

Part – A

1. What is the OSI security architecture? [C01 - L1]
The OSI (open system interconnection) security architecture provides a systematic framework for defining security attacks, mechanisms and services.

2. What is the difference between passive and active attacks? [C01 - L1]
A passive attack attempt to learn or eavesdropping on transmission and it does not affect system resources or affect their operations.
A active attacks involve some modification of the data stream or information.

3. List the categories of passive attacks. [C01 - L1]
Traffic analysis
Release of message

4. List the categories of active attacks. [C01 - L1]
Masquerade, Replay, Modification of massage, Denial of service

5. List the categories of security services. [C01 - L1]
Authentication, Access control, Data confidentiality, Data integrity, Non repudiation

6. List the categories of security mechanisms. [C01 - L1]
Specific security mechanism
Pervasive security mechanism

7. What are the essential ingredients of a symmetric cipher? [C01 - L1]

Plaintext
Encryption algorithm
Secret key
Cipher text
Decryption algorithm

8. What are the two basic functions used in encryption algorithms? [C01 - L1]

The two basic functions used in encryption algorithms are
Substitution
Transposition

9. How many keys are required for two people to communicate via a cipher? [C01 - L1]

If both sender and receiver use the same key, the system is referred to as symmetric, Single key, secret key, or conventional encryption.

If the sender and receiver each use a different key, the system is referred to as asymmetric, two-key, or public-key encryption.

10. What is the difference between a block cipher and a stream cipher? [C01 - L2]

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.

11. What are the two approaches to attacking a cipher? [C01 - L1]

Cryptanalysis
Brute-force attack

12. What is the difference between an unconditionally secure cipher and a computationally secure cipher? [C01 – L2]

An unconditionally secure cipher is a scheme such that if the cipher text generated by the scheme does not contain enough information to determine uniquely the corresponding plain text, no matter how much cipher text is available.

A computationally secure scheme is such that the cost of breaking the cipher exceeds the value of the encrypted information and the time required to break the cipher exceeds the useful lifetime of the information.

13. Briefly define the Caesar cipher. [C01 - H1 - Nov/Dec 2012-Nov/Dec 2013]
The Caesar cipher involves replacing each letter of the alphabet with the letter standing:

Three places further down the alphabet. For example:

Plain: meet me after the toga party
Cipher: PHHW PH DIWHU WKH WRJD SDUWB

14. Briefly define the Playfair cipher? [C01 - L1 - May /June 2011]

The Playfair cipher treats the diagrams in the plaintext as single units and translates these units into ciphertext digrams.

This algorithm is based on the use of a 5 by 5 matrix of letters constructed using keyword. Consider keyword as monarchy.

The matrix is constructed by filling in the letters of the keyword from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters. The letters I, J count as one letter

15. What are the two problems with one-time pad? [C01 - L1]

It makes the problem of making large quantities of random keys.

It also makes the problem of key distribution and protection.

16. What is a transposition cipher? [C01 - L1 -Nov/Dec 2013]

Transposition cipher is a cipher, which is achieved by performing some sort of permutation on the plaintext letters.
Example: Plain text: meet me after the toga party

17. What is Steganography? [C01 - L1 - **May/June 2013]**

In stenography the plain text is hidden. The existence of the message is concealed. For example the sequence of first letters of each word of the overall message in the hidden message.

18. Explain the avalanche effect. [C01 – L2 - Nov/Dec 2012]

It is that a small change in either the plaintext or the key should produce a significant change in the cipher text. A change in one of the bit of the plaintext or one bit of the key should produce a change in many bits of the cipher text.

19. What is the difference between a mono alphabetic cipher and a poly alphabetic cipher? [C01 - L1 - Nov/Dec 2012]
Mono alphabetic cipher: Here a single cipher alphabet is used.

Poly alphabetic cipher: Here a set of related mono alphabetic substitution rules is used.

20. List the types of cryptanalytic attacks. [C01 - L1]
Cipher text only
Known plaintext
Chosen plaintext
Chosen cipher text
Chosen text

21. When an encryption algorithm is said to be computationally secured? [C01 – H1]
The encryption algorithm is said to be computationally secure if

The cost of breaking the cipher exceeds the value of the encrypted information

The time required to break the cipher exceeds the useful time of the information.

22. What are the key principles of security? [C01 - L1]

Key properties of security:
To protect the data during transmission across the networks
Authentication
Confidentiality
Integrity
Access control

23. What types of information might be derived from a traffic analysis attack? [C01 - L1]

The following types of information can be derived from traffic analysis attack:

1. Identities of partners
2. How frequently the partners are communicating
3. Message pattern, message length, or quantity of messages that suggest important information is being exchanged
4. The events that correlate with special conversations between particular Partners

24. What is Rail fence Transposition Technique? [C01 - L1]

In this technique plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

25. Define Threats. [C01 – L1]

Information access threats intercept or modify data on behalf of users who should not have access to that data. Service threats exploit service flaws in computers to inhibit use by legitimate users.

26. What are the aspects of information security? [C01 - L1]

There are three aspects of the information security.

Security attack

Security mechanism

Security Service

27. List some common information integrity functions? [C01 - L1]

Identification
Authorization
Concurrence
Liability
Endorsement
Validation
Time of occurrence

28. What is meant by attack? [C01 - L1]

An attack 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.

29. What are the essential ingredients of a symmetric cipher? [C01 - L1]
A symmetric encryption scheme has five ingredients:

Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.
Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.
Secret Key: The secret key is also input to the encryption algorithm. The key is the value independent of the plaintext. 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.
Cipher text: This is the scrambled message produced as output. It depends on the plaintext and the key.
Decryption algorithm: This is essentially the encryption algorithm in reverse. It takes the cipher text and the secret key and produces the original plaintext.

30. What are the two basic functions used in the encryption algorithm? [C01 - L1]
All the encryption algorithms are based on two general principles:
Substitution: In which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element.
Transposition: In which elements in the plaintext are rearranged.

31. Briefly define the Caesar cipher? [C01 – L3 - Nov/Dec 2012]

The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places down the alphabet .The alphabet is wrapped around, so that the letter following Z is A.

$$C = E(p) = (p + 3) \bmod (26)$$

The general Caesar cipher algorithm is

$$C = E(p) = (p + k) \bmod (26)$$

Where k takes the value in the range 1 to 25

The decryption algorithm is

$$p = D(C) = (C - k) \bmod (26)$$

32. Briefly define the monoalphabetic cipher? [C01 – L3- Nov/Dec 2012]

A monoalphabetic cipher maps from a plain alphabet to cipher alphabet. Here a single cipher alphabet is used per message

A dramatic increase in the key space is achieved by allowing an arbitrary substitution. There are 26! Possible keys. It is referred to as monoalphabetic substitution cipher, because a single cipher alphabet is used per message.

33. What is the difference between a monoalphabetic cipher and a polyalphabetic cipher? [C01 - L1 - AU Nov/Dec 2012]

In monoalphabetic cipher single cipher alphabet is used per message.

But in polyalphabetic cipher there are multiple ciphertext letters for each plaintext letter, one for each unique letter of keyword.

34. Define LFSR. [C01 - L1]

A linear feedback shift register is a shift register whose input is a linear function of its previous state. It is also called linear recursive sequence.

35. What is prime amount? [C01 - L1]

A prime number is an integer that can only be divided without remainder by positive and positive values of itself and by one.

36. What is the meaning of the expression a divides b? [C01 - L1]
Integer a is said to be a divisor of integer b if there is no remainder on division.

37. What is Euler's totient function? [C01 - L1]

Euler's totient function Φ(n) defined as the number of positive integers less than n and

Relatively prime to n. by conversion **Φ(1)=1.**

38. What is Fermat's theorem? [C01 - L1]

Fermat's theorem states the following:

If p is prime and a is a positive integer not divisible by p, then

Ap-1 = 1(mod p)

39. What is Euler's theorem? [C01 - L1 - April /May2011- May/Jun 2014]

Euler's theorem states that for every a and n that are relatively prime.

$a^{\Phi(n)} = 1(\text{mod } n)$

40. Briefly define a group. [C01 – L2]

A group G, sometimes denoted by {G, .} is a set of elements with a binary operation, denoted by . , that associates

41. What is the difference between a Caesar cipher and a polyalphabetic cipher? [C01 – L2 - Nov/Dec 2012]

In caesar cipher single cipher alphabet is used per message. But in polyalphabetic cipher there are multiple ciphertext letters for each plaintext letter, one for each unique letter of keyword.

42. Convert the given text "Anna University" into cipher text using rail fence technique. [C01 – H2 - May/June 2013]

Plain text: Anna University

Cipher text: anuiestnanvry

43. Why modular arithmetic has been used in cryptography? [C01 – L2 - Nov/Dec 2013]

It is used to find multiplicative inverse in cryptography.

44. What are active and passive attacks that compromise information security? [C01 – L2]

1. Passive Attacks: These 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 where an eavesdropper tries to learn the contents of what is being transmitted.

Traffic analysis, where the opponent tries to observe the pattern, frequency and length of messages being exchanged which could be used in guessing the nature of the communication that is taking place.

1. Active Attacks: Active attacks involve some modification of the data stream or the creation of a false stream. These attacks present the opposite characteristics of passive attacks. It is difficult to prevent active attacks absolutely because to do so would require physical protection of all communications facilities and paths at all times.

45. Why random numbers are used in network security? [C01 – L2 -May/Jun 2014]
Nonces in authentication protocols to prevent replay

Session keys

Public key generation

Key stream for a one-time pad

46. What are the two basic functions used in encryption algorithms? [C01 – L1 - Nov/Dec 2014]

The two basic functions used in encryption algorithms are
Substitution
Transposition

47. List the types of cryptanalytic attacks. [C01 – L1 - Nov/Dec 2014]
Cipher text only
Known plaintext
Chosen plaintext
Chosen cipher text
Chosen text

48. What is the difference between a block cipher and a stream cipher? [C01 – L3]

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.

49. Decipher the following cipher text using brute force attack? CMTMROOEOORW using Railfence. [C01 – H3]

Solution:

CTROORMMOEOW

50. What is Security? [C01 – L1]

Security is “the quality or state of being secure-to be free from danger”.

51. What are the basic components of computer Security? [C01 – L1]

Confidentiality - Keeping data and resources hidden b. Integrity - Data integrity (integrity)- Origin integrity (authentication)c. Availability - Enabling access to data and resources

52. What is confidentiality? [C01 – L1]

Confidentiality is the concealment of information or resources. The need for keeping information secret arises from the use of computers in sensitive fields such as government and industry. For example Military and civilian institutions in the government often restrict access to information to those who need that information.

53. What is Integrity? [C01 – L1]

Integrity refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing or unauthorized change.

Integrity includes data integrity (the content of the information) and origin integrity (the Source of the data, often called Authentication).For example A newspaper – The information is printed as received (preserving data integrity) But its source is incorrect (corrupting origin integrity).

54. What is Availability? [C01 – L1]

Availability refers to the ability to use the information or resource desired. Availability is an important aspect of reliability as well as of system design because an unavailable system is at all. For example - Bank’s secondary system server.

55. What is a threat? [C01 – L1]

A threat is a potential violation of security. The violation need not actually occur for there to be a threat.

56. What are the different broad classes of threats? [C01 – H1]

Disclosure – Snooping
Deception - Modification, spoofing, repudiation of origin, denial of receipt
Disruption – Modification
Usurpation - Modification, spoofing, delay, denial of service

57. What do mean by snooping? [C01 – L1]

Snooping, the unauthorized interception of information is a form of disclosure. It is a technique used to gain unauthorized access to computers. It is passive, suggesting simply that some entity is listening to communications.

58. Distinguish between policy and mechanism. [C01 – L3]

Policy: It is a statement of what is, and what is not allowed.
Mechanism: It is a method, tool or procedure for enforcing a security policy.

59. What are the goals of security? [C01 – L1]

1. Prevention: It means that an attack will fail

2. Detection: It is most useful when an attack cannot be prevented, but it can also indicate the effectiveness of preventative measures.

3. Recovery: To stop an attack and to assess and repair any damage caused by that attack, the system continuous to function correctly while an attack is underway.

60. What are the types of security policies? [C01 – L1]
1. Military security policy

2. Commercial security policy

3. Transaction oriented integrity security policy

4. Confidentiality security policy

5. Integrity policy

61.What are the types of access control? [C01 – L1]

Identity based access control

Mandatory access control

Originator controlled access control

62. What is an identity based access control? [C01 – L1]

If an individual user can set an access control mechanism to allow or deny access to an object, that mechanism is a discretionary access control also called an identity based access control.

63. What is a mandatory access control? [C01 – L1]

When a system mechanism controls access to an object and an individual user can not alter that access, the control is a mandatory access control, occasionally called as rule-based access control.

PART - B

1. List and briefly define categories of security services. [C01 – L1]

Introduction

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.

Perhaps a clearer definition is found in RFC 2828, which provides the following definition: a processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms.

X.800 divides these services into five categories and fourteen specific services .We look at each category in turn.

1. Authentication

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

In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. First, at the time of connection initiation, the service assures that the two entities are authentic, that is, that each is the entity that it claims to be.

Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade 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

Used in association with a logical connection to provide confidence in the identity of the entities connected.

Data Origin Authentication

In a connectionless transfer, provides assurance that the source of received data is as claimed.

2. Access Control

The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).

3. Data Confidentiality

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

The protection of data from unauthorized disclosure.

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

4. Data Integrity

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

A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays. The destruction of data is also covered under this service. Thus, the connection-oriented integrity service addresses both message stream modification and denial of service.

On the other hand, a connectionless integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only.
We can make a distinction between service with and without recovery. Because the integrity service relates to active attacks, we are concerned with detection rather than prevention. If a violation of integrity is detected, then the service may simply report this violation, and some other portion of software or human intervention is required to recover from the violation.

Alternatively, there are mechanisms available to recover from the loss of integrity of data, as we will review subsequently. The incorporation of automated recovery mechanisms is, in general, the more attractive alternative.

Connection Integrity with Recovery

Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.

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 single connectionless data block takes the form of determination of whether the selected fields have been modified.

5. Nonrepudiation

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.

6. Availability Service

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 (i.e., a system is available if it provides services according to the system design whenever users request them).

A variety of attacks can result in the loss of or reduction in availability. Some of these attacks are amenable to automated countermeasures, such as authentication and encryption, whereas others require some sort of physical action to prevent or recover from loss of availability of elements of a distributed system.

X.800 treats availability as a property to be associated with various security services. However, it makes sense to call out specifically an availability service.

An availability service is one that protects a system to ensure its availability. 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. List and briefly define categories of security mechanisms. [C01 – L1]

Introduction

The mechanisms are divided into those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service.

Encipherment

The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

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 an entity by means of information exchange.

Traffic Padding

The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

Routing Control

Enables selection of particular physically secure routes 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 certain properties of a data exchange.

Pervasive Security Mechanisms

Mechanisms those are not specific to any particular OSI security service or protocol layer.

Trusted Functionality

That which is perceived to be correct with respect to some 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 security attributes 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 and examination of system records and activities.

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.

It based on one in X.800, indicates the relationship between security services and security mechanisms.

3. List and briefly define categories of passive and active security attacks. or What are the different types of attacks? Or Explain detail about security attacks [C01 – L2-Nov/Dec 2013]

Introduction

Security attacks, uses both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks.* A passive attack attempts to learn or make use of information from the system but does not affect system resources.

An active attack attempts to alter system resources or affect their operation. Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions.

The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.

The release of message contents is easily understood .A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions. A second type of passive attack, traffic analysis, is subtler

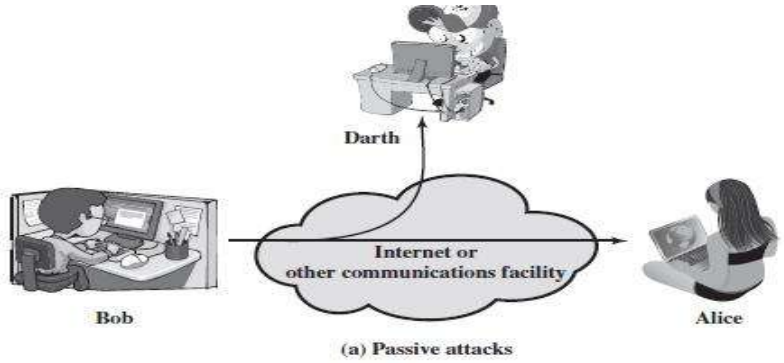
Passive Attacks

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

The release of message contents and
Traffic analysis

The release of message contents is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information.

A second type of passive attack, traffic analysis, is subtler. Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message.



Active Attacks

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories:

Masquerade,
Replay,
Modification of messages, and
Denial of service.

A masquerade - A masquerade attack usually includes one of the other forms of active attack.

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect.

The denial of service prevents or inhibits the normal use or management of communications facilities.

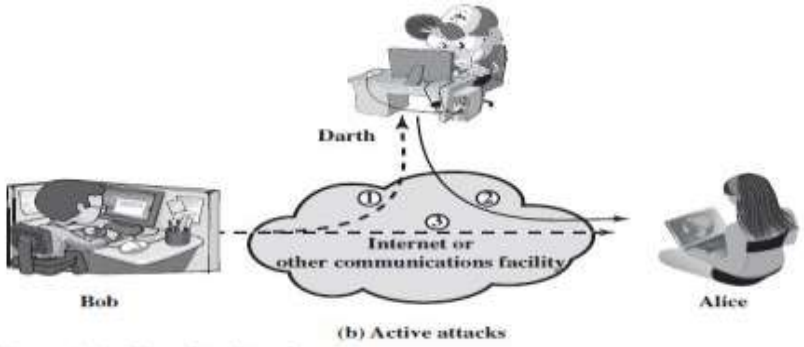


Figure 1.1 Security Attacks

4. Explain in detail about The OSI Security Architecture Contents. [C01 – L2]

Introduction

ITU-T3 Recommendation X.800, *Security Architecture for OSI*, defines such a systematic approach.4 The OSI security architecture is useful to managers as a way of organizing the task of providing security.

For our purposes, the OSI security architecture provides a useful, if abstract, overview of many of the concepts that this book deals with. The OSI security architecture focuses on security attacks, mechanisms, and services.

These can be defined briefly 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.

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 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.

Security Attacks

A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms passive attacks and active attacks.

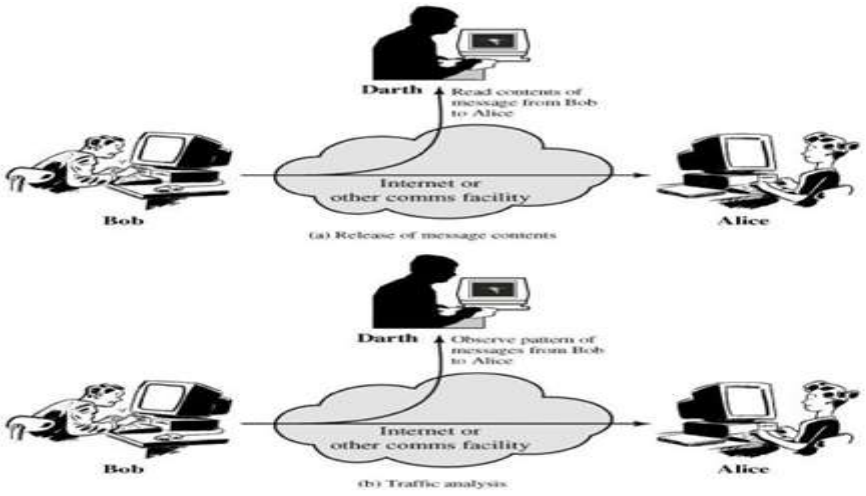
A passive attack attempts to learn or make use of information from the system but does not affect system resources.

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

Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted.

Two types of passive attacks are release of message contents and traffic analysis.
The release of message contents is easily understood .A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information.



5. Write short notes on Network security model. [C01 – L2]

A message is to be transferred from one party to another across some sort of Internet service. The two parties, who are the *principals* in this transaction, must cooperate for the exchange to take place.

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

Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

All the techniques for providing security have two components:

A *security-related transformation* on the information to be sent.

Some secret information shared by the two principals and, it is hoped, unknown to the opponent. A trusted third party may be needed to achieve secure transmission.

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

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.

2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.

4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

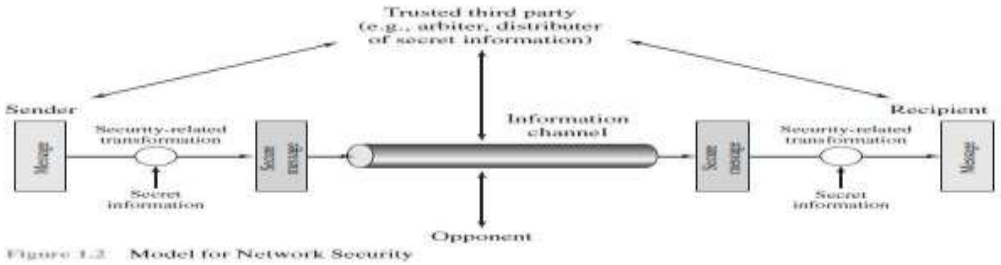


Figure 1.3 Model for Network Security

A general model of these other situations is illustrated in Figure 1.3, which reflects a concern for protecting an information system from unwanted access. The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system.

Programs can present two kinds of threats:

Information access threats: Intercept or modify data on behalf of users who should not have access to that data.

Service threats: Exploit service flaws in computers to inhibit use by legitimate users.

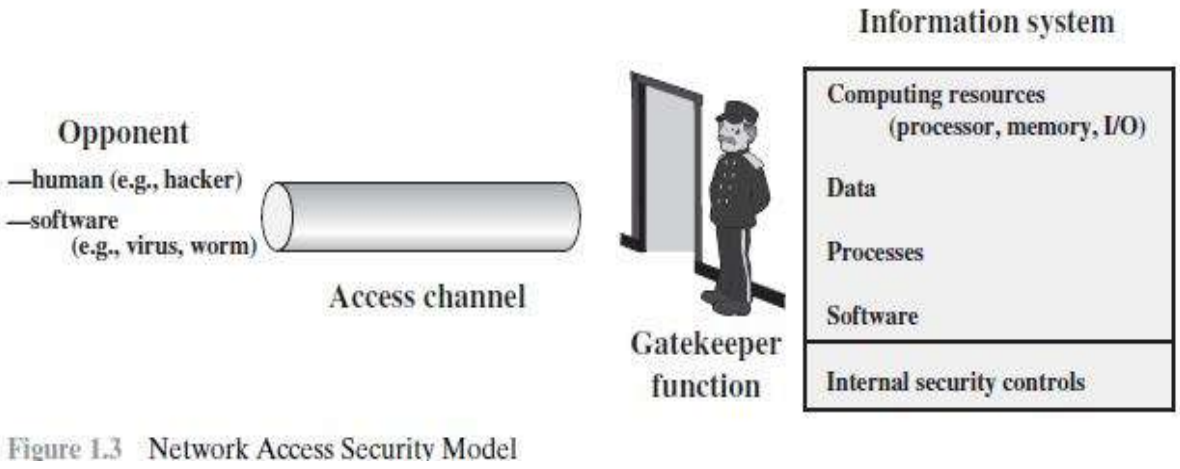


Figure 1.3 Network Access Security Model

Viruses* and *worms are two examples of software attacks. They can also be inserted into a system across a network.

The security mechanisms needed to cope with unwanted access fall into two broad categories (see Figure 1.3).

The first category might be termed a gatekeeper function. It includes password-based login procedures and screening logic that is designed to detect and reject worms, viruses.The second line of defense consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

6. What are the essential ingredients of a symmetric cipher? [C01 – L1]

Contents:

Ingredients
Requirements
Cryptography
Cryptanalysis and Brute-Force Attack
Cryptanalysis
Brute-force attack

Ingredients

A symmetric encryption scheme has five ingredients (Figure 2.1):

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

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

Secret key: The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm.

Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the secret key.

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

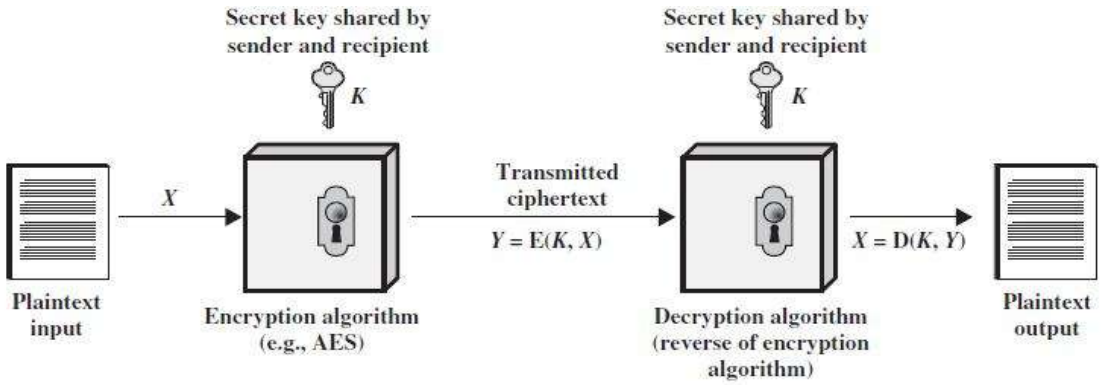


Figure 2.1 Simplified Model of Symmetric Encryption

Requirements

There are two requirements for secure use of conventional encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key.

2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

We assume that it is impractical to decrypt a message on the basis of the ciphertext *plus* knowledge of the encryption/decryption algorithm.

This feature of symmetric encryption is what makes it feasible for widespread use. The algorithm is of low-cost chip implementations of data encryption algorithms. These chips are widely available and incorporated into a number of products.

With the use of symmetric encryption, the principal security problem is maintaining the secrecy of the key. Let us take a closer look at the essential elements of a symmetric encryption scheme, using Figure 2.2. A source produces a message in plaintext, $X = [X_1, X_2, c, X_M]$. The M elements of X are letters in some finite alphabet.

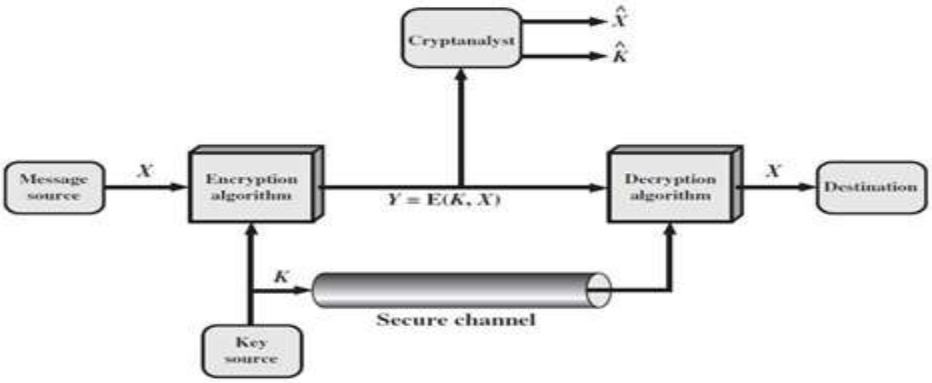


Figure 2.2 Model of Symmetric Cryptosystem

Traditionally, the alphabet usually consisted of the 26 capital letters. Nowadays, the binary alphabet {0, 1} is typically used. For encryption, a key of the form $K = [K_1, K_2, c, K_J]$ is generated. If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel. With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext $Y = [Y_1, Y_2, c, Y_N]$. We can write this as

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

This notation indicates that Y is produced by using encryption algorithm E as a function of the plaintext X, with the specific function determined by the value of the key K.
The intended receiver, in possession of the key, is able to invert the transformation:

$$X=D(K,Y)$$

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 $\hat{X}$.

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 $\hat{K}$

Cryptography

Cryptographic systems are characterized along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext.

2. The number of keys used.
3. The way in which the plaintext is processed.

Cryptanalysis and Brute-Force Attack

There are two general approaches to attacking a conventional encryption scheme:

Cryptanalysis: Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs.
Brute-force attack: The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

7. What are the different substitution techniques involved in classic encryption technique? or
Briefly define the Caesar cipher. Briefly define the monoalphabetic cipher. Briefly define the Playfair cipher. [C01 – L3]

Introduction

The two basic building blocks of all encryption techniques are *substitution* and *transposition*.

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 sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

Caesar Cipher

The *earliest known,* and the *simples*t, use of a substitution cipher was by *Julius Caesar*. The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet. For example,

Note that the alphabet is wrapped around, so that the letter following Z is A. We can define the transformation by listing all possibilities, as follows:

Monoalphabetic Ciphers

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution.

Before proceeding, we define the term *permutation*. 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.

For example, if *S* = {a, b, c}, there are six permutations of *S*:

abc, acb, bac, bca, cab, cba

If, instead, the “cipher” line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4 * 1026 possible keys. This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Such an approach is referred to as a *monoalphabetic substitution*

cipher, because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.

Playfair Cipher

The best-known *multiple-letter encryption* cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext diagrams.

The Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers’s *Have His Carcase*

In this case, the keyword is *monarchy*. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter.

Hill Cipher

The *Hill Algorithm* This encryption algorithm takes *m* successive plaintext letters and substitutes for them *m* ciphertext letters. The substitution is determined by *m* linear

equations in which each character is assigned a numerical value (a = 0, b = 1, c, z = 25).

For example, consider the plaintext “paymoremoney” and use the encryption Key The first three letters of the plaintext are represented by the vector (15 0 24). Then(15 0 24)K = (303 303 531) mod 26 = (17 17 11) = RRL. Continuing in this fashion, the ciphertext for the entire plaintext is RRLMWBKASPDH.

Polyalphabetic Ciphers

Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message.

The general name for this approach is polyalphabetic substitution cipher. All these techniques have the following features in common:

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword. For example, if the keyword is deceptive, the message “we are discovered save yourself” is encrypted.

One-Time Pad

An Army Signal Corp officer, Joseph Mauborgne, proposed an improvement to the Vernam 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.

In addition, 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 random output that bears no statistical relationship to the plaintext. Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code.

8. What are the different techniques involved in transposition techniques of Classical Encryption techniques. [C01 – H2]

A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.

The simplest such cipher is the **rail fence** technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows. For example, to encipher the message “meet me after the toga party” with a rail fence of depth 2.

This sort of thing would be trivial to cryptanalyze. A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. For example,

Thus, in this example, the key is 4312567. To encrypt, start with the column that is labeled 1, in this case column 3. Write down all the letters in that column. Proceed to column 4, which is labeled 2, then column 2, then column 1, then columns 5, 6, and 7.

A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. For the type of columnar transposition just shown, cryptanalysis is fairly straightforward and involves laying out the ciphertext in a matrix and playing around with column positions. Digram and trigram frequency tables can be useful.

9. What is Steganography? Explain. [C01 – L2]

Techniques

The methods of steganography conceal the existence of the message, whereas the methods of cryptography render the message unintelligible to outsiders by various transformations of the text.

A simple form of steganography, but one that is time-consuming to construct, is one in which an arrangement of words or letters within an apparently innocuous text spells out the real message.

For example, the sequence of first letters of each word of the overall message spells out the hidden message. Figure 2.9 shows an example in which a subset of the words of the overall message is used to convey the hidden message. See if you can decipher this; it's not too hard.

Various other techniques have been used historically; some examples are the following:

- **Character marking:** Selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.

- **Invisible ink:** A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper.

- **Pin punctures:** Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.

- **Typewriter correction ribbon:** Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered.

10. Explain the Euclid's algorithm. [C01 – L2]

Introduction

One of the basic techniques of number theory is the Euclidean algorithm, which is a simple procedure for determining the greatest common divisor of two positive integers. First, we need a simple definition: Two integers are relatively prime if their only common positive integer factor is 1.

Greatest Common Divisor

Recall that nonzero b is defined to be a divisor of *a* if *a* = *m*b for some *m*, where *a, b*, and *m* are integers.

We will use the notation gcd(*a, b*) to mean the greatest common divisor of *a* and *b*. The greatest common divisor of *a* & *b* is the largest integer that divides both *a* and *b*.

We also define gcd(0, 0) = 0. More formally, the positive integer c is said to be the greatest common divisor of a and b if

1. *c* is a divisor of *a* and of *b*.
2. Any divisor of *a* and *b* is a divisor of *c*.

An equivalent definition is the following:

GCD(a,b)= max[k,xsuch that [a and k]b]

Because we require that the greatest common divisor be positive,
gcd(a, b) = gcd(a, −b) = gcd(−a, b) = gcd(−a,−b). In general, gcd(a, b) = gcd(|a| , |b|).

GCD(60,24)=GCD (60,-12)=12

Finding the Greatest Common Divisor

Suppose we have integers a, b such that d = gcd(a, b). Because gcd gcd(|a| , |b|) = gcd(a, b), there is no harm in assuming a ≥ b > 0. Now dividing a by b and applying the division algorithm, we can state.

Let us now return to Equation (4.2) and assume that $r_1 \neq 0$. Because $b > r_1$, we can divide b by r_1 and apply the division algorithm to obtain:

In this example, we begin by dividing 1160718174 by 316258250, which gives 3 with a remainder of 211943424. Next we take 316258250 and divide it by 211943424. The process continues until we get a remainder of 0, yielding a result of 1078.

11. The Miller-Rabin test can determine if a number is not prime but cannot determine if a number is prime. How can such an algorithm be used to test primality? [C01 – H3]

Testing for primality

Miller-Rabin Algorithm

The algorithm due to Miller and Rabin [MILL75, RABI80] is typically used to test a large number for primality. Before explaining the algorithm, we need some background.

First, any positive odd integer n ≥ 3 can be expressed as

Two Properties of Prime Numbers

The first property is stated as follows:

If p is prime and a is a positive integer less than p, then a^2 mod p = 1 if and only if either a mod p = 1 or a mod p = −1 mod p = p −1. By the rules of modular arithmetic (a mod p) (a mod p) = a^2 mod p.

The second property is stated as follows:

1. a^q is congruent to *1 modulo p*. That is, a^q mod p = 1, or equivalently, $a^q \equiv 1 (\text{mod } p)$.

2. One of the numbers a^q, $a^{2q\ a2k-l}q$ is congruent to −1 modulo *p.*

Details of the Algorithm

The procedure TEST takes a candidate integer n as input and returns the result composite if n is definitely not a prime, and the result inconclusive if n may or may not be a prime.

A Deterministic Primality Algorithm

All of the algorithms in use, including the most popular (Miller-Rabin), produced a probabilistic result.

AKS developed a relatively simple deterministic algorithm that efficiently determines whether a given large number is a prime. The algorithm, known as the AKS algorithm, does not appear to be as efficient as the Miller- Rabin algorithm.

12. A common formulation of the Chinese remainder theorem (CRT) is as follows: Let $m_1,\ldots, m_k$ be integers that are pairwise relatively prime for $1 \le i, j \le k$, and $i \ne j$. Define M to be the product of all the m_i's. Let $a_1,\ldots, a_k$ be integers. Then the set of congruences: [C01 – H3]

$$x \equiv a1 \pmod{m1}$$

$$x \equiv a2 \pmod{m2}$$

$$\vdots$$

$$x \equiv ak \pmod{mk}$$

has a unique solution modulo M. Show that the theorem stated in this form is true.

The Chinese remainder theorem

One of the most useful results of number theory is the Chinese remainder theorem (CRT). In essence, the CRT says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli.

The CRT can be stated in several ways. We present here a formulation that is most useful from the point of view of this text. An alternative formulation is explored.

where the mi are pairwise relatively prime; that is, gcd(m_i, m_j) = 1 for $1 \le i, j \le k$, and $i \ne j$. We can represent any integer A in Z_M by a k-tuple whose elements are in Z_{mi} using the following correspondence:

where A € Z_M, a_i _ Z_{mi}, and $a_i = A \bmod m_i$ for $1 \le i \le k$.

The mapping of Equation (8.7) is a one-to-one correspondence (called a bijection) between ZM and the Cartesian product $Z_{m1} \times Z_{m2} \times ..... \times Z_{mk}$.

Operations performed on the elements of ZM can be equivalently performed on the corresponding k-tuples by performing the operation independently in each coordinate position in the appropriate system.

Let us demonstrate the first assertion. The transformation from A to ($a_1, a_2, \ldots, a_k$), is obviously unique; that is, each a_i is uniquely calculated as $a_i = A \bmod m_i$.

By the definition of M_i, it is relatively prime to m_i and therefore has a unique multiplicative inverse mod m_i. So Equation (8.8) is well defined and produces a unique value c_i.

We can now compute

$A \bmod m_i$ for $1 \le i \le k$. Note that $c_j \equiv M_j \equiv 0 \pmod{m_i}$ if $j \ne i$, and that $c_i \equiv 1 \pmod{m_i}$. It follows that $a_i = A \bmod m_i$.

The second assertion of the CRT, concerning arithmetic operations, follows from the rules for modular arithmetic. That is, the second assertion can be stated as follows: If

One of the useful features of the Chinese remainder theorem is that it provides a way to manipulate (potentially very large) numbers mod M in terms of tuples of smaller numbers. This can be useful when M is 150 digits or more. However, note that it is necessary to know beforehand the factorization of M.

Unit – II

Block Ciphers & Public Key Cryptography

Part – A

1. What is the purpose of the State array? [C02 - L1]

A single 128-bit block is depicted as a square matrix of bytes. This block is copied into the State array, which is modified at each stage of encryption or decryption. After the final stage, State is copied to an output matrix.

2. How is the S-box constructed? [C02 - L1]

The S-box is constructed in the following fashion:

Initialize the S-box with the byte values in ascending sequence row by row. The first row contains {00}, {01}, {02}... {0F}; the second row contains {10},{11},etc and so on. Thus, the value of the byte at row x, column y is {x y}.

Map each byte in the S-box to its multiplicative inverse in the finite field GF (28); the value {00} is mapped to itself. Consider that each byte in the S-box consists of 8 bits labeled (b7, b6, b5, b4, b3, b2, b1,b0).Apply the following transformation to each bit of each byte in the S-box.

3. Briefly describe Sub Bytes. [C02 – L3]

Sub byte uses an S-box to perform a byte-by-byte substitution of the block. The leftmost 4 bits of the byte are used as 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 value.

4. What is the difference between differential and linear cryptanalysis? [C02 – L2-May/Jun 2011]

In differential cryptanalysis, it breaks the DES in less 255 complexities.

In cryptanalysis, it finds the DES key given 247 plaintexts.

5. Define product cipher. [C02 - L1]

Product cipher performs two or more basic ciphers in sequence in such a way that the final result or product is crypto logically stronger than any of the component ciphers.

6. What was the original set of criteria used by NIST to evaluate candidate AES cipher? [C02 - L1]

The original set of criteria used by NIST to evaluate candidate AES cipher was

- Security
- Actual Security
- Randomness
- Soundness
- Other security factors
- Cost
- Memory Requirements
- Algorithm and Implementation
- Characteristics
- Flexibility
- Hardware and software suitability
- Simplicity Licensing Requirements
- Computational Efficiency

7. What was the final set of criteria used by NIST to evaluate candidate AES Ciphers? [C02 - L1]

The final set of criteria used by NIST to evaluate candidate AES ciphers was:

General Security Software Implementations Restricted-Space Environments Hardware Potential for Instruction Level Parallelism

8. What is power analysis? [C02 - L1]

Power analysis is the power consumed by the smart card at any particular time during the cryptographic operation is related to the instruction being executed and to the data being processed.(Eg) Multiplication consumes more power than addition and writing 1s consumes more power than writing 0s.

9. How many bytes in State are affected by Shift Rows? [C02 - H1]

Totally 6-bytes in state are affected by Shift Rows.

10. Briefly describe Mix Columns. **[C02 – L3]**

Mix Column is substitution that makes use of arithmetic over GF (28).Mix Column 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 the column.

The Mix Column Transformation combined with the shift row transformation ensures that after a few rounds, all output bits depend on all input bits.

11. Briefly describe Add Round Key. [C02 – L3]

In Add Round Key, the 128 bits of State are bit wise XOR with the 128 bits of the round key.

The operation is viewed as a column wise operation between the 4 bytes of a State column and one word of the round key; it can also be viewed as a byte-level operation.

The Add Round Key transformation is as simple as possible and affects every bit of State.

12. Briefly describe the Key Expansion Algorithm. **[C02 – L3]**

The AES key expansion algorithm takes as input a 4-word (16-byte) key and produces linear array of 44 words (156 bytes).

This is sufficient to provide a 4-word round key for the initial Add Round Key stage and each of the 10 rounds of the cipher

13. What is the difference between Sub Bytes and Sub Word?
[C02 - L1- Apr/May 2011]

Sub Bytes:

Sub Bytes uses an S-box to perform a byte-by-byte substitution of the block.

Sub *Word:*

Sub Word performs a byte substitution on each byte of its input word, using the Sbox.

14. What is the difference between Shift Rows and Rot Word? [C02 - L1]

Shift Rows:

Shift Row is simple permutation. It shifts the rows circularly left or right.

Rot Word:

Rot word performs a one-byte circular left shift on a word. This means that an input word [b0, b1, b2, b3] is transformed into [b1,b2,b3,b0].

15. What is triple encryption? [C02 - L1]

Tuchman proposed a triple encryption method that uses only two keys [TUCH79].The function follows an encrypt – decrypt – encrypt (EDE) sequence.

C=Ek1 [Dk2 [Ek1 [P]]]

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

C=Ek1 [Dk2 [Ek1 [P]]] = Ek1 [P]

16. What is a meet-in-the-middle attack? [C02 - L1]

Meet-in-the-middle attack was first described in [DIFF77]. It is based on the Observation that, if we have

C=Ek2 [Ek1 [P]] Then
X=Ek1 [P]= Dk2[C]

Given a known pair, (P,C), the attack proceeds as follows.

First, encrypt P for all 256 possible values of K1. Store these results in a table and then sort the table by the values of X. Next, decrypt C using all 256 possible values of K2.

As each decryption is produced, check the result against the table for a match. If a match occurs, then test the two resulting keys against a new known plaintext-cipher text pair.
If the two keys produce the correct ciphertext, accept them as the correct keys.

17. How many keys are used in triple encryption? [C02 - L1]

Tuchman proposed a triple encryption method that uses only two keys.

18. What is the key size for Blowfish? [C02 - L1]

Blowfish makes use of a key that ranges from 32 bits to 448 bits (one to fourteen 32-bit words).

That key is used to generate 18 32-bit sub keys and four 8*32 S-boxes containing a total of 1024 32-bit entries.

The total is 1042 32-bit values, or 4168 bytes.

19. Why do some block cipher modes of operation only use encryption while others use both Encryption and decryption? [C02 - H1]

Some block cipher modes of operation only use encryption because the input is set to some initialization vector and the leftmost bits of the output of the encryption function are XOR with the first segment of plain text p1 to produce the first unit of cipher text C1 and it is transmitted. While in decryption, the cipher text is XOR with the output of the encryption function to produce the plain text.

20. Mention the functions involved in simplified DES. [C02 - L1]

Initial permutation

A complex function F_k with a key k1

Switching

A complex function F_k with a key k2

21. Define stream cipher and block cipher. [C02 - L1]

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

A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal block.

22. Define symmetric key cryptography and public key cryptography. [C02 - L1]

In symmetric key cryptography, only one key is used for encryption and decryption

In public key cryptography, two keys (public key and private key) are used.

When one key is used for encryption, then the other must be used for decryption. The public key is known to all the participants but the private key is kept secret by the owner.

23. List out the applications of the public key cryptosystems. [C02 - L1]

Encryption / decryption
Digital signature
Key exchange

24. What are the various approaches to attacks the RSA algorithm? [C02 - L1]

Brute force attack
Mathematical attacks
Timing attacks

25. Compare stream cipher with block cipher with example. [C02 – L2]

- Stream Cipher:
Processes the input stream continuously and producing one element at a time.
Example: caeser cipher.
Block cipher:
Processes the input one block of elements at a time producing an output block for each input block. Example: DES.

26. Differentiate symmetric and asymmetric encryption? [C02 – L2]

Symmetric

Symmetric Asymmetric It is a form of cryptosystem in which encryption and decryption performed using the same key. It is a form of cryptosystem in which encryption and decryption performed using two keys. Eg: DES, AES Eg: RSA, ECC

Asymmetric

Asymmetric It is a form of cryptosystem in which encryption and decryption performed using the different key. It is a form of cryptosystem in which encryption and decryption performed using two keys. Eg: RSA, RC4

27. Define cryptanalysis. [C02 - L1]

It is a process of attempting to discover the key or plaintext or both.

28. Define steganography, [C02 - L1]

Hiding the message into some cover media. It conceals the existence of a message.

29. Why network need security? [C02 - L1]

When systems are connected through the network, attacks are possible during transmission time

30. Define Encryption. [C02 - L1]

The process of converting from plaintext to cipher text.

31. Specify the components of encryption algorithm. **[C02 – L2]**

Plaintext
Encryption algorithm
Secret key
ciphertext
Decryption algorithm

32. Define confidentiality and authentication Confidentiality. **[C02 - L1]**

Confidentiality
It means how to maintain the secrecy of message. It ensures that the information in a computer system and transmitted information are accessible only for reading by authorized person.
Authentication
It helps to prove that the source entity only has involved the transaction

33. Define Diffusion & confusion. [C02 - L1]

Diffusion:

It means each plaintext digits affect the value of many cipher text digits which is equivalent to each cipher text digit is affected by many plaintext digits. It can be achieved by performing permutation on the data. It is the relationship between the plaintext and cipher text.

Confusion:

It can be achieved by substitution algorithm. It is the relationship between ciphertext and key.

34. Give the five modes of operation of Block cipher. [C02 – L3- Nov/Dec 2012]

Electronic Codebook (ECB)
Cipher Block Chaining (CBC
Cipher Feedback (CFB)
Output Feedback (OFB)
Counter (CTR)

35. Find gcd (1970, 1066) using Euclid's algorithm? [C02 - H1-May/Jun 2013]

Gcd (1970,1066) = gcd(1066,1970 mod 1066)
= gcd (1066,904)
= 2

36. What is the primitive root of a number? [C02 - L1-Nov/Dec 2012]

We can define a primitive root of a number p as one whose powers generate all the integers from 1 to p-1. That is p, if a is a primitive root of the prime number p then the number

37. What is the difference between differential and linear cryptanalysis? [C02 – L3]

In differential cryptanalysis, it breaks the DES in less 2 5 5 complexities. In cryptanalysis, it finds the DES key given 2^{47} plaintext ts.

38. What is Factoring? [C02 - L1-May/Jun 2012]

Factoring is the decomposition of an object into a product of other objects, or factors, which when multiplied together give the original.

39. Define Differential Cryptanalysis. [C02 – L2-May/Jun 2012]

A technique in which chosen plaintext with particular XOR difference patterns are encrypted. These difference patterns of the resulting ciphertext provide information that can be used to determine the encryption key.

40. List the Block cipher Modes of operation. [C02 - L1-Nov/Dec 2013]

Electronic Codebook (ECB)
Cipher Block Chaining (CBC)
Cipher Feedback (CFB)
Output Feedback (OFB)
Counter (CTR)

41. What is the disadvantage with ECB mode of operation? [C02 - L1-May/Jun 2013]

ECB encrypts highly deterministically
Identical plaintexts result in identical ciphertexts
An attacker recognizes if the same message has been sent twice
Plaintext blocks are encrypted independently of previous blocks
An attacker may reorder ciphertext blocks which results in valid plaintext

42. State whether symmetric and asymmetric cryptographic algorithm need key exchange [C02 - H1-MAY 2014]

A symmetric encryption (ie. symmetric ciphers), k must be secret. The sender and recipient must agree (somehow) on k. No-one else can be allowed to find out k. Anyone else who finds out k, can decrypt all the messages encrypted with k. For that reason, symmetric ciphers are often called "secret key" ciphers

An asymmetric encryption (ie Asymmetric ciphers), the encryption key k is not secret. The recipient (not sender) chooses a so-called "public key" k, and a so-called "private key" p. Then they publish k for all to see - perhaps on their website - but keep p secret.

Senders use k to encrypt their messages to that recipient recipient keeps p secret. Since k is not only public - Asymmetric ciphers are often called "public key" ciphers.

43. What are roles of public and private key? [C02 - L1]

The two keys used for public-key encryption are referred to as the public key and the private key. Invariably, the private key is kept secret and the public key is known publicly. Usually the public key is used for encryption purpose and the private key is used in the decryption side.

44. Specify the applications of the public key cryptosystem? [C02 – L3]

The applications of the public-key cryptosystem can classified as follows

1. **Encryption/Decryption**: The sender encrypts a message with the recipient's public key.
2. **Digital signature**: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to a message or to a small block of data that is a function of the message.
3. **Key Exchange**: Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

45. What is a one way function? [C02 - L1]

One way function is one that map the domain into a range such that every function value has a unique inverse with a condition that the calculation of the function is easy where as the calculations of the inverse is infeasible.

46. Describe in general terms an efficient procedure for picking a prime number? [C02 - L1]

The procedure for picking a prime number is as follows:

1. Pick an odd integer n at random (eg., using a pseudorandom number generator).
2. Pick an integer a<n at random.
3. Perform the probabilistic primality test, such as Miller-Rabin. If n fails the test, reject the value n and go to step 1.
4. If n has passed a sufficient number of tests, accept n; otherwise , go to step 2.

47. Perform encryption and decryption using RSA Alg. for the following. P=7; q=11; e=17; M=8. [C02 – H3]

Soln: n = pq
n = 7*11=77
O(n)=6*10
=60 e =17

d =27

C = Me mod n
C = 817 mod 77=57

M = Cd mod n
= 5727 mod 77= 8

48. What is an elliptic curve? [C02 - L1]

The principle attraction of ECC compared to RSA, is that it appears to offer equal security for a far smaller key size, thereby reducing processing overhead.

49. Define Public-Key Cryptography. [C02 - L1]

Probably most significant advance in the 3000 year history of cryptography Uses two keys a public & a private key Asymmetric since parties are not equal.

Uses clever application of number theoretic concepts to function Complements rather than replaces private key crypto

50. Determine the gcd(24140,16762) using Euclid's algorithm. [C02 – L3]

Soln:

We know,
gcd(a,b)=gcd(b,amod b)gcd(24140,16762)=gcd(16762,7378)
gcd(7378,2006)=gcd(2006,1360) gcd(1360,646)=gcd(646,68) gcd(68,34)=34
gcd(24140,16762) = 34.

51. User A & B exchange the key using Diffie Hellman alg. Assume á=5 q=11 XA=2 XB=3. Find YA, YB, K. [C02 – H2]

Soln:
YA = áXA
mod q =52
mod 11
= 3
YB = áXB mod q
= 53 mod 11
= 4
KA = YBXA mod q
= 42 mod 11
= 5
KB = YAXB mod q
= 33 mod 11
= 5

52. What is public-key certificate? [C02 - L1]

The public-key authority could be a bottleneck in the system, for a user must appeal to the authority for a public key for every other user that it wishes to contact. As before the directory of names and public keys maintained by the authority is vulnerable to tempering.

53. What are the requirements for the use of a public-key certificate scheme? [C02 - L1]

Any participant can read a certificate to determine the name and public key of the certificate's owner.
Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
Only the certificate authority can create and update certificates.
Any participant can verify the currency of the certificate.

PART-B

1. Explain in details Data Encryption Standard. Or Draw the block diagram of single round of DES algorithm and explain the Processing carried out in each block. [C02 - L1-APR/MAY-2011-NOV/DEC 2012-MAY/JUN 2013-MAY/JUN 2014]

Introduction:

The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46) .The algorithms itself is referred to as the Data Encryption Algorithm (DEA).

DES Encryption:

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

Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.

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 pre output

Finally, the preoutput is passed through a permutation [IP -1] that is the inverse of the initial permutation function, to produce the 64-bit ciphertext. 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. Initially, the key is passed through a permutation function. Then, for each of the sixteen rounds, a subkey (Ki) is produced by the combination of a left circular shift and a permutation.

The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

DES Decryption

As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed. Additionally, the initial and final permutations are reversed.

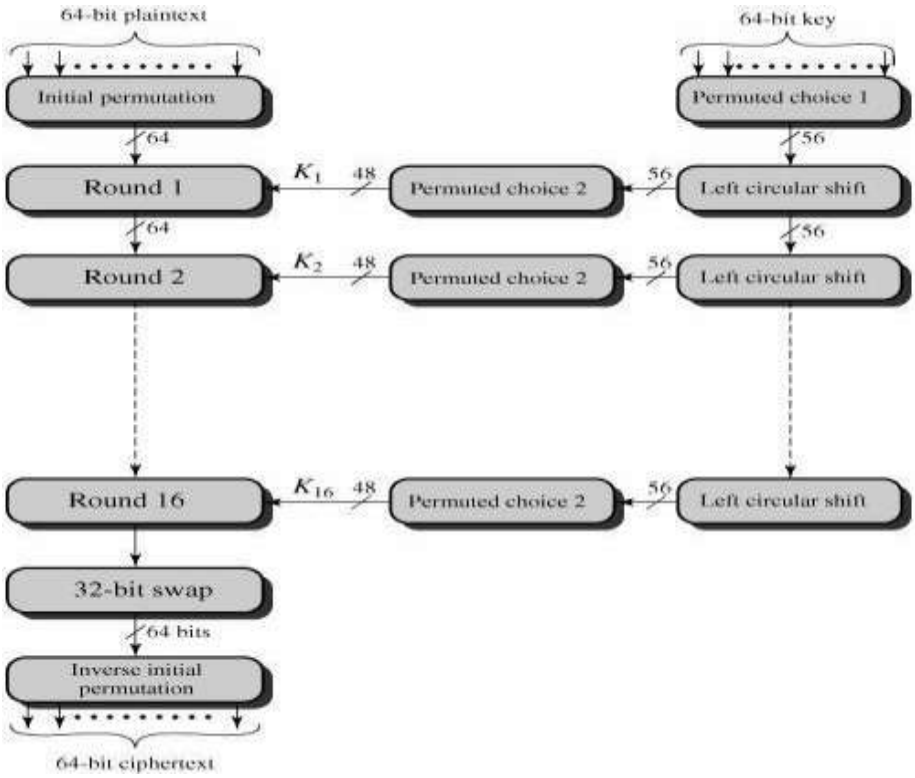


Fig .General Depiction of DES Encryption Algorithm

DES Example

For this example, the plaintext is a hexadecimal palindrome. The plaintext, key, and resulting ciphertext are as follows:

Results

Table 3.2 shows the progression of the algorithm. The first row shows the 32-bit values of the left and right halves of data after the initial permutation. The next 16 rows show the results after each round. Also shown is the value of the 48-bit subkey

The Avalanche Effect

A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the cipher text.

In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the cipher text.

This is referred to as the avalanche effect. If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched.

Using the example from Table 3.2, Table 3.3 shows the result when the fourth bit of the plaintext is changed,

So that the plaintext is **12468aceeca86420**.

The second column of the table shows the intermediate 64-bit values at the end of each round for the two plaintexts. The third column shows the number of bits that differ between 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 in 32 bit positions.

Table 3.4 shows a similar test using the original plaintext of with two keys that differ in only the fourth bit position: the original key, **0f1571c947d9e859**, and the altered key, **1f1571c947d9e859**. Again, the results show that about half of the bits in the ciphertext differ and that the avalanche effect is pronounced after just a few rounds.

The strength of DES

The Use of 56-Bit Keys

The Nature of the DES Algorithm

Timing Attacks

The Use of 56-Bit Keys

With a key length of 56 bits, there are 256 possible keys, which is approximately **7.2 * 1016 keys**. Thus, on the face of it, a brute-force attack appears impractical.

Assuming that, on average, half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.

The Nature of the DES Algorithm

The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration . Because the design criteria 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 cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.

This assertion is tantalizing, and over the years a number of regularities and unexpected behaviors of the S-boxes have been discovered. Despite this, no one has so far succeeded in discovering the supposed fatal weaknesses in the S-boxes.9

Timing Attacks

Timing attacks in more detail in Part Two, as they relate to public-key algorithms. However, the issue may also be relevant for symmetric ciphers.In essence, a timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various cipher texts.

2. Explain the Block cipher Design principles. [C02 – L2]

Introduction

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.

In this section we look at three critical aspects of block cipher design: the number of rounds, design of the function F, and key scheduling

Number of Rounds

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. Let us look first at the choice of the number of rounds.

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.

This criterion was certainly used in the design of DES. Schneier observes that for 16-round DES, a differential cryptanalysis attack is slightly less efficient than brute force:

The differential cryptanalysis attack requires 255.1 operations,10 whereas brute force requires 255.

If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search.

This criterion is attractive, because it makes it easy to judge the strength of an algorithm and to compare different algorithms. In the absence of a cryptanalytic breakthrough, the strength of any algorithm that satisfies the criterion can be judged solely on key length.

Design of Function F

The heart of a Feistel block cipher is the function F, which provides the element of confusion in a Feistel cipher. Thus, it must be difficult to "unscramble" the substitution performed by F.

One obvious criterion is that F be nonlinear, as we discussed previously. The more nonlinear F, the more difficult any type of cryptanalysis will be.

There are several measures of nonlinearity, which are beyond the scope of this book. In rough terms, the more difficult it is to approximate F by a set of linear equations, the more nonlinear F is. Several other criteria should be considered in designing

We would like the algorithm to have good avalanche properties. Recall that, in general, this means that a change in one bit of the input should produce a change in many bits of the output.

A more stringent version of this is the strict avalanche criterion (SAC) [WEBS86], which states that any output bit *j* of an S-box (see Appendix S for a discussion of S-boxes) should change with probability 1/2 when any single input bit *i* is inverted for all *i*, *j*.

Although SAC is expressed in terms of S-boxes, a similar criterion could be applied to F as a whole. This is important when considering designs that do not include S-boxes.

Another criterion proposed in [WEBS86] is the bit independence criterion (BIC), which states that output bits *j* and *k* should change independently when any single input bit *i* is inverted for all *i*, *j*, and *k*. The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function.

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.

No general principles for this have yet been promulgated. Adams suggests [ADAM94] that, at minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion.

3. Explain the Block cipher modes of operation in detail. [C02 – L2]

Electronic Code Book

The simplest mode is the Electronic codebook (ECB) mode, in which plaintext is handled one block at a time and each block of plaintext is encrypted using the same key (Figure 6.3). The term codebook is used because, for a given key, there is a unique ciphertext for every b-bit block of plaintext.

Therefore, we can imagine a gigantic codebook in which there is an entry for every possible b-bit plaintext pattern showing its corresponding ciphertext.

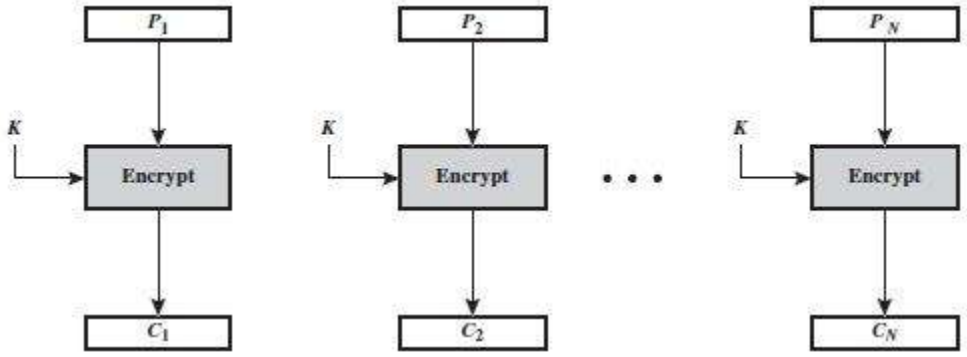
For a message longer than b bits, the procedure is simply to break the message into b-bit blocks, padding the last block if necessary. Decryption is performed one block at a time, always using the same key.

In Figure 6.3, the plaintext (padded as necessary) consists of a sequence of b-bit blocks, P1, P2,c, PN; the corresponding sequence of ciphertext blocks is C1, C2, c, CN. We can define ECB mode as follows.

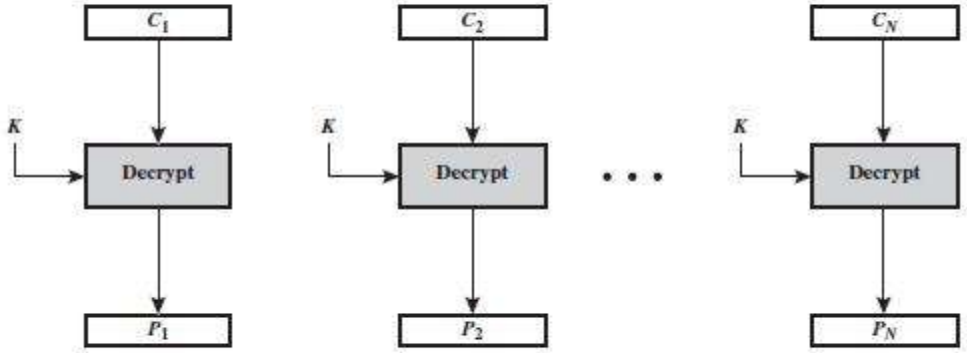
The ECB method is ideal for a short amount of data, such as an encryption key. Thus, if you want to transmit a DES or AES key securely, ECB is the appropriate mode to use. The most significant characteristic of ECB is that if the same b-bit block of plaintext appears more than once in the message, it always produces the same ciphertext.

For lengthy messages, the ECB mode may not be secure. If the message is highly structured, it may be possible for a cryptanalyst to exploit these regularities.

For example, if it is known that the message always starts out with certain predefined fields, then the cryptanalyst may have a number of known plaintext– ciphertext pairs to work with. If the message has repetitive elements with a period of repetition a multiple of b bits, then these elements can be identified by the analyst. This may help in the analysis or may provide an opportunity for substituting or rearranging blocks.



(a) Encryption



(b) Decryption

Figure 6.3 Electronic Codebook (ECB) Mode

We now turn to more complex modes of operation. lists the following criteria and properties

For evaluating and constructing block cipher modes of operation that are superior to ECB:

Overhead: The additional operations for the encryption and decryption operation when compared to encrypting and decrypting in the ECB mode.

Error recovery: The property that an error in the ith cipher text block is inherited by only a few plaintext blocks after which the mode resynchronizes.

Error propagation: The property that an error in the ith ciphertext block is inherited by the i th and all subsequent plaintext blocks. What is meant here is a bit error that occurs

in the transmission of a cipher text block, not a computational error in the encryption of a plaintext block.

Diffusion: How the plaintext statistics are reflected in the ciphertext. Low entropy plaintext blocks should not be reflected in the ciphertext blocks. Roughly, low entropy equates to predictability or lack of randomness

Security: Whether or not the ciphertext blocks leak information about the plaintext blocks.

Cipher Block Chaining Mode

To overcome the security deficiencies of ECB, we would like a technique in which the same plaintext block, if repeated, produces different ciphertext blocks. A simple way to satisfy this requirement is the cipher block chaining (CBC) mode

In this scheme, the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; the same key is used for each block. In effect, we have chained together the processing of the sequence of plaintext blocks.

The input to the encryption function for each plaintext block bears no fixed relationship to the plaintext block. Therefore, repeating patterns of b bits are not exposed. As with the ECB mode, the CBC mode requires that the last block be padded to a full b bits if it is a partial block. For decryption, each cipher block is passed through the decryption algorithm.

To produce the first block of ciphertext, an initialization vector (IV) is XORed with the first block of plaintext. On decryption, the IV is XORed with the output of the decryption algorithm to recover the first block of plaintext. The IV is a data block that is the same size as the cipher block. We can define CBC mode as

The IV must be known to both the sender and receiver but be unpredictable by a third party. In particular, for any given plaintext, it must not be possible to predict the IV that will be associated to the plaintext in advance of the generation of the IV. For maximum security, the IV should be protected against unauthorized changes.

Where the prime notation denotes bit complementation. This means that if an opponent can predictably change bits in IV, the corresponding bits of the received value of P1 can be changed.

Cipher Feedback Mode

For AES, DES, or any block cipher, encryption is performed on a block of b bits. In the case of DES, b = 64 and in the case of AES, b = 128. However, it is possible to convert a block cipher into a stream cipher, using one of the three modes to be discussed in this and the next two sections: cipher feedback (CFB) mode, output feedback (OFB) mode, and counter (CTR) mode.

A stream cipher eliminates the need to pad a message to be an integral number of blocks. It also can operate in real time. Thus, if a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher.

One desirable property of a stream cipher is that the ciphertext be of the same length as the plaintext. Thus, if 8-bit characters are being transmitted, each character should be encrypted to produce a ciphertext output of 8 bits. If more than 8 bits are produced, transmission capacity is wasted.

Figure 6.5 depicts the CFB scheme. In the figure, it is assumed that the unit of transmission is s bits; a common value is s = 8. As with CBC, the units of plaintext are chained together, so that the cipher text of any plaintext unit is a function of all the preceding plaintext. In this case, rather than blocks of b bits, the plaintext is divided into segments of s bits.

First, consider encryption. The input to the encryption function is a b-bit shift register that is initially set to some initialization vector (IV). The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P1 to produce the first unit of ciphertext C1, which is then transmitted. In addition, the contents of the shift register are shifted left by s bits, and C1 is placed in the rightmost (least significant) s bits of the shift register.

This process continues until all plaintext units have been encrypted. For decryption, the same scheme is used, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit. Note that it is the encryption function that is used, not the decryption function.

This is easily explained. Let MSBs(X) be defined as the most significant s bits of X.
Then

The same reasoning holds for subsequent steps in the process

Although CFB can be viewed as a stream cipher, it does not conform to the typical construction of a stream cipher. In a typical stream cipher, the cipher takes as input

some initial value and a key and generates a stream of bits, which is then XORed with the plaintext bits. In the case of CFB, the stream of bits that is XORed with the plaintext also depends on the plaintext.

In CFB encryption, like CBC encryption, the input block to each forward Cipher function (except the first) depends on the result of the previous forward cipher function; therefore, multiple forward cipher operations cannot be performed in parallel.

In CFB decryption, the required forward cipher operations can be performed in parallel if the input blocks are first constructed (in series) from the IV and the ciphertext output feedback (OFB)

Output feedback (OFB) mode

The output feedback (OFB) mode is similar in structure to that of CFB. For OFB, the output of the encryption function is fed back to become the input for encrypting the next block of plaintext (Figure 6.6). In CFB, the output of the XOR unit is fed back to become input for encrypting the next block.

Let the size of a block be b. If the last block of plaintext contains u bits (indicated by *), with u 6 b, the most significant u bits of the last output block ON are used for the XOR operation; the remaining b - u bits of the last output block are discarded.

As with CBC and CFB, the OFB mode requires an initialization vector. In the case of OFB, the IV must be a nonce; that is, the IV must be unique to each execution of the encryption operation. The reason for this is that the sequence of Encryption output blocks, Oi, depends only on the key and the IV and does not depend on the plaintext.

One advantage of the OFB method is that bit errors in transmission do not propagate.

The disadvantage of OFB is that it is more vulnerable to a message stream modification attackthan is CFB.

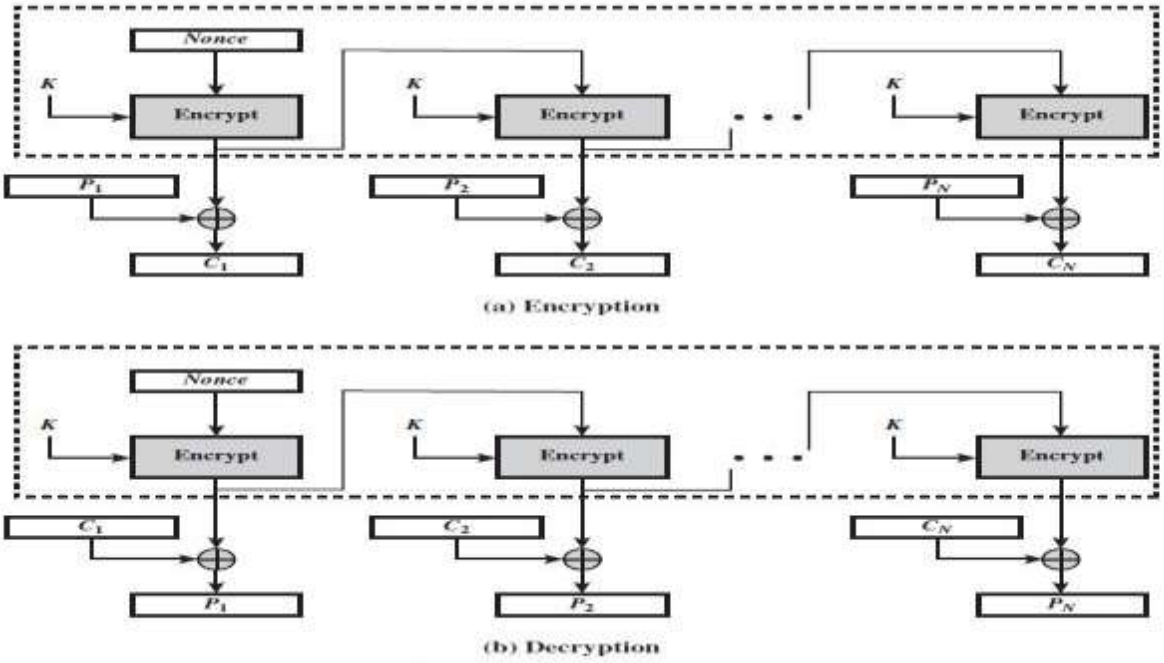


Figure 6.6 Output Feedback (OFB) Mode

Counter Mode

Although interest in the counter (CTR) mode has increased recently with applications to ATM (asynchronous transfer mode) network security and IP sec (IP security), this mode was proposed early.

Figure 6.7 depicts the CTR mode. A counter equal to the plaintext block size is used. The only requirement stated in SP 800-38A is that the counter value must be different for each plaintext block that is encrypted. Typically, the counter is initialized to some value and then incremented by 1 for each subsequent block (modulo 2b, where b is the block size).

For encryption, the counter is encrypted and then XORed with the plaintext block to produce the ciphertext block; there is no chaining.

For decryption, the same sequence of counter values is used, with each encrypted counter XORed with a ciphertext block to recover the corresponding plaintext block. Thus, the initial counter value must be made available for decryption. Given a sequence of counters T1, T2, c, TN, we can define CTR mode as follows.

For the last plaintext block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the XOR operation; the remaining b - u bits are discarded. Unlike the ECB, CBC, and CFB modes, we do not need to use padding because of the structure of the CTR mode.

As with the OFB mode, the initial counter value must be a nonce; that is, T1 must be different for all of the messages encrypted using the same key. Further, all Ti values across all messages must be unique. If, contrary to this requirement, a counter value is used multiple times, then the confidentiality of all of the plaintext blocks corresponding to that counter value may be compromised. In particular, if any plaintext block that is encrypted using a given counter value is known, then the output of the encryption function can be determined easily from the associated ciphertext block.

This output allows any other plaintext blocks that are encrypted using the same counter value to be easily recovered from their associated ciphertext blocks.

One way to ensure the uniqueness of counter values is to continue to increment the counter value by 1 across messages. That is, the first counter value of the each message is one more than the last counter value of the preceding message.

4. Explain in detail about the Advanced Encryption Standard (AES) [C02 – L2]

Introduction

It is worth examining the criteria used by NIST to evaluate potential candidates. These criteria span the range of concerns for the practical application of modern symmetric block ciphers.

In fact, two set of criteria evolved. When NIST issued its original request for candidate algorithm nominations in 1997 [NIST97], the request stated that candidate algorithms would be compared based on the factors shown in Table 5.1 (ranked in descending order of relative importance).

Finite Field Arithmetic

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: $a/b = a(b\text{-}1)$. An example of a finite field (one with a finite number of elements) is the set Zp consisting of all the integers {0, 1, c, p - 1}, where p is a prime number and in which arithmetic is carried out modulo p.

Virtually all encryption algorithms, both conventional and public-key, involve arithmetic

operations on integers. If one of the operations used in the algorithm is division, then we need to work in arithmetic defined over a field; this is because division requires that each nonzero element have a multiplicative inverse.

For convenience and for implementation efficiency, we would also like to work with integers that fit exactly into a given number of bits, with no wasted bit patterns. That is, we wish to work with integers in the range 0 through 2*n* - 1, which fit into an *n*-bit word.

Unfortunately, the set of such integers, *Z*2*n*, using modular arithmetic, is not a field.

For example, the integer 2 has no multiplicative inverse in *Z*2*n*, that is, there is no integer *b*, such that 2*b* mod 2*n* = 1. There is a way of defining a finite field containing 2*n* elements; such a field is referred to as GF(2*n*). Consider the set, *S*, of all polynomials of degree *n* - 1 or less with binary coefficients.

Where each *ai* takes on the value 0 or 1. There are a total of 2*n* different polynomials in *S*. For *n* = 3, the 23 = 8 polynomials in the set 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*).

For example, to construct the finite field GF(23), we need to choose an irreducible polynomial of degree 4. There are only two such polynomials: (*x*3 + *x*2 + 1) and (*x*3 + *x* + 1). Addition is equivalent to taking the XOR of like terms. Thus, (*x* + 1) + *x* = 1. A polynomial in GF(2*n*) can be uniquely represented by its *n* binary coefficients (*an*-1*an*-2

ca0). Therefore, every polynomial in GF(2*n*) can be represented by an *n*-bit number. Addition is performed by taking the bitwise XOR of the two *n*-bit elements. There is no simple XOR operation that will accomplish multiplication in GF(2*n*).

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(28), with the irreducible polynomial3 $m(x) = x8 + x4 + x3 + x + 1$. The developers of Rijndael give as their motivation for selecting this one of the 30 possible irreducible polynomials of degree 8 that it is the first one on the list given .

AES Structure

General Structure

Detailed Structure

General Structure

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 input to the encryption and decryption algorithms is a single 128-bit block. In FIPS PUB 197, this block is depicted as a 4 * 4 square matrix of bytes. This block is copied into the State array, which is modified at each stage of encryption or decryption. After the final stage, State is copied to an output matrix. These operations are depicted in Figure 5.2a. Similarly, the key is depicted as a square matrix of bytes.

This key is then expanded into an array of key schedule words. Figure (5.2).shows the expansion for the 128-bit key. Each word is four bytes, and the total key schedule is 44 words for the 128-bit key. Note that the ordering of bytes within a matrix is by column.

So, for example, the first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the in matrix, the second four bytes occupy the second column, and so on. Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the w matrix.

The cipher consists of *N* rounds, where the number of rounds depends on the key length:

SubBytes,

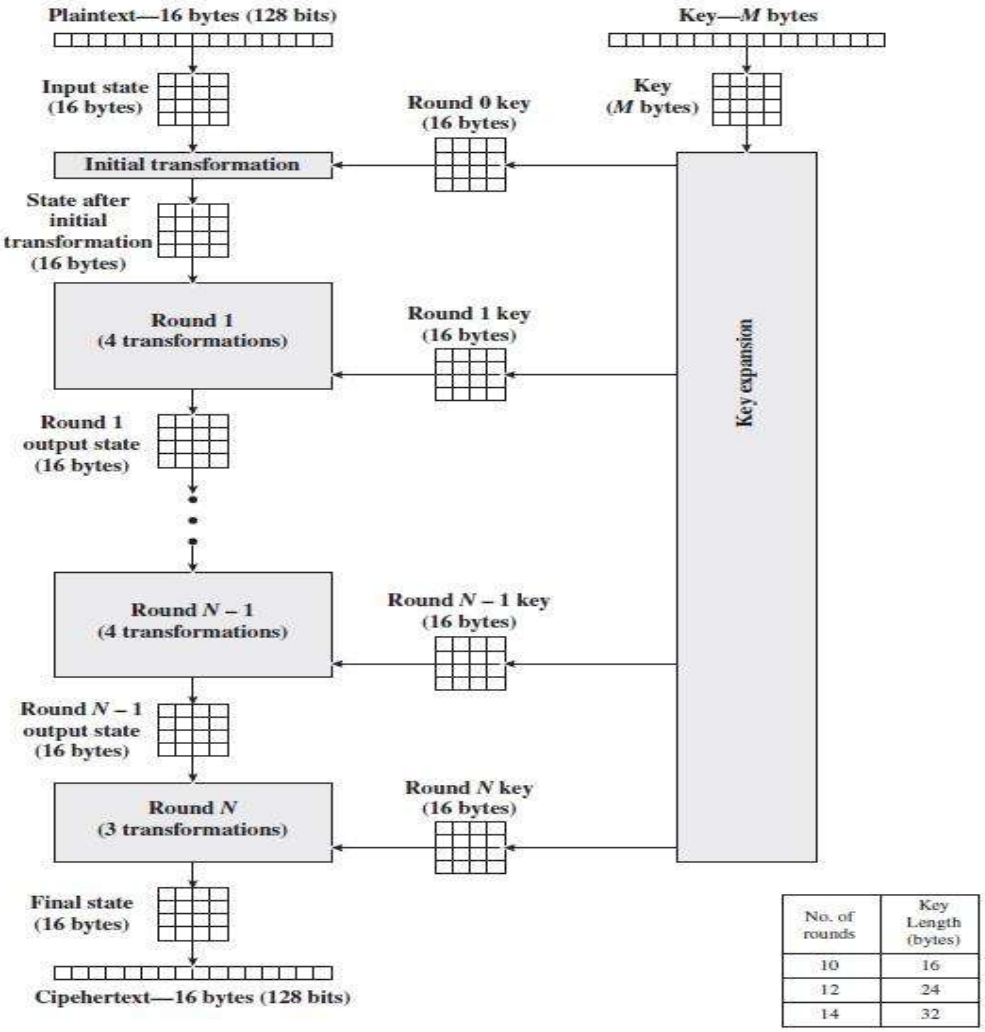
ShiftRows,

MixColumns, and

AddRoundKey,

The final round contains only three transformations, and there is a initial single transformation (AddRoundKey) before the first round, which can be considered Round

Each transformation takes one or more 4 * 4 matrices as input and produces a 4 * 4 matrix as output.Figure (5.1) shows that the output of each round is a 4 * 4 matrix, with the output of the final round being the ciphertext. Also, the key expansion function generates *N* + 1 round keys, each of which is a distinct 4 * 4 matrix. Each round key serves as one of the inputs to the AddRoundKey transformation in each round.



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

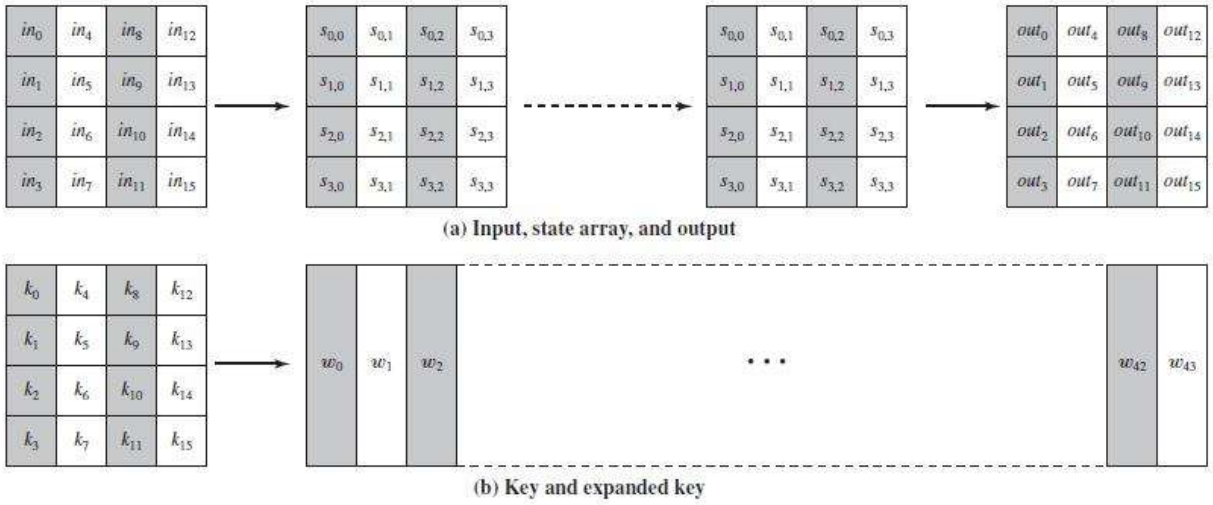


Fig (5.2) : AES Data Structures

Detailed Structure

Figure (5.1) shows the AES cipher in more detail, indicating the sequence of transformations in each round and showing the corresponding decryption function.

We can make several comments about the overall AES structure.

1. One noteworthy feature of this structure is that it is not a Feistel structure. Recall that, in the classic Feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. AES instead processes the entire data block as a single matrix during each round using substitutions and permutation.
2. The key that is provided as input is expanded into an array of forty-four 32-bit words, w[*i*]. Four
   distinct words (128 bits) serve as a round key for each round; these are indicated in Figure 5.3

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 GF(28)

• **AddRoundKey:** A simple bitwise XOR of the current block with a portion of the expanded Key

4. The structure is quite simple. For both encryption and decryption, the cipher begins with an AddRoundKey stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages. Figure (4) depicts the structure of a full encryption round.

5. Only the AddRoundKey stage makes use of the key. For this reason, the cipher begins and ends with an AddRoundKey stage. Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security.

6. The AddRoundKey stage is, in effect, a form of Vernam cipher and by itself would not be formidable. The other three stages together provide confusion, diffusion, and nonlinearity, but by themselves would provide no security because they do not use the key. We can view the cipher as alternating operations of XOR encryption (AddRoundKey) of a block, followed by scrambling of the block (the other three stages), followed by XOR encryption, and so on.This scheme is both efficient and highly secure.

7. Each stage is easily reversible. For the Substitute Byte, ShiftRows, and MixColumns stages, an inverse function is used in the decryption algorithm. For the AddRoundKey stage, the inverse is achieved by XORing the same round key to the block, using the result that identical to the encryption algorithm. This is a consequence of the particular structure of AES.

8. Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext. Figure (5.3) lays out encryption and decryption going in opposite vertical directions. At each horizontal point (e.g., the dashed

The four transformations used in AES. For each stage, we describe the forward (encryption) algorithm, the inverse (decryption) algorithm, and the rationale for the stage.

Substitute Bytes Transformation

Shift Rows Transformation

Mix Columns Transformation

AddRoundKey Transformation

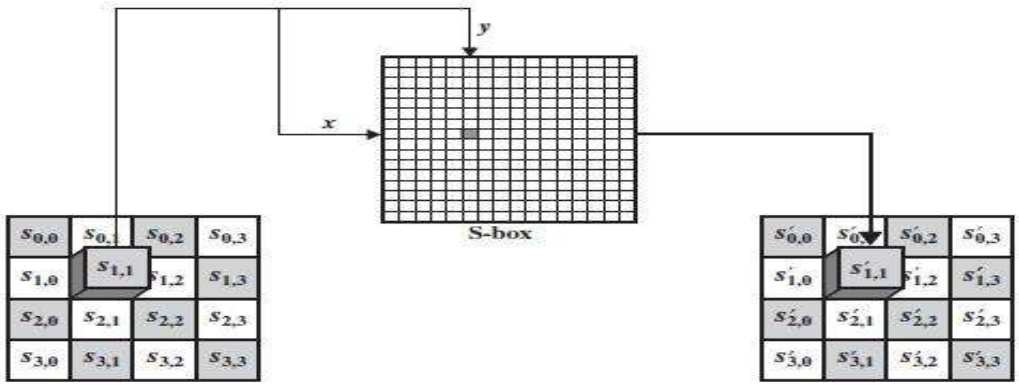
Substitute Bytes Transformation

Forward and Inverse Transformati ons The forward substitute byte transformation, called SubBytes, is a simple table lookup (Figure (5.5)a). AES defines a 16 * 16 matrix of byte values, called an S-box (Table 5.2a), that contains a permutation of all possible 256 8-bit values.

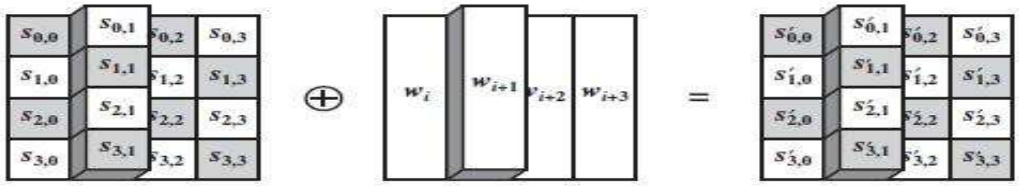
Each individual byte of State is mapped into a new byte in the following way:

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. For example, the hexadecimal value {95} references row 9, column 5 of the S-box, which contains the value {2A}. Accordingly, the value {95} is mapped into the value {2A}.

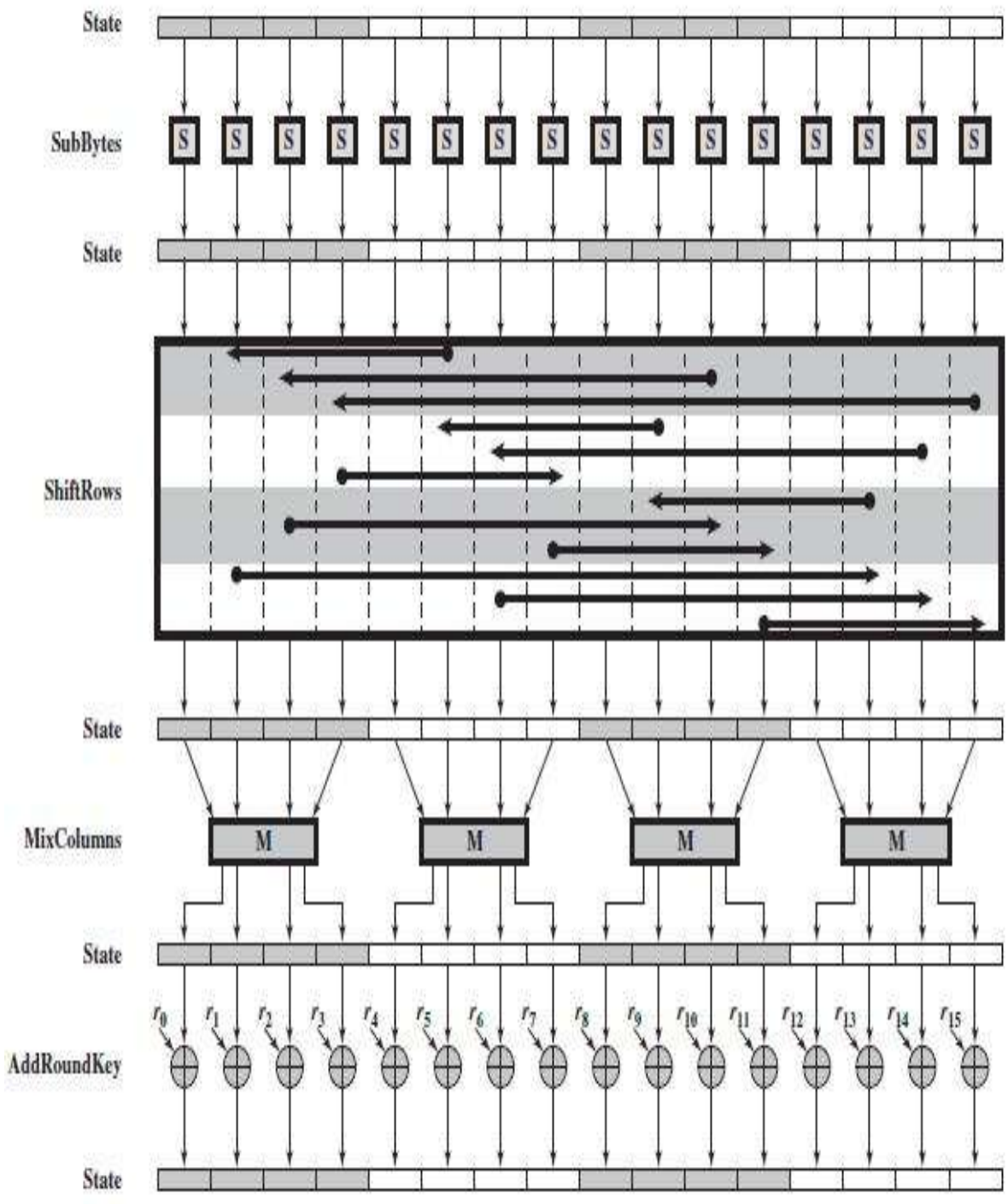


(a) Substitute byte transformation



(b) Add round key transformation

AES Byte-Level Operations



ShiftRows Transformation

Forward and Inverse Transformati ons The forward shift row transformation, called ShiftRows, is depicted in Figure 5.7a. 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. The following is an example of ShiftRows.

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

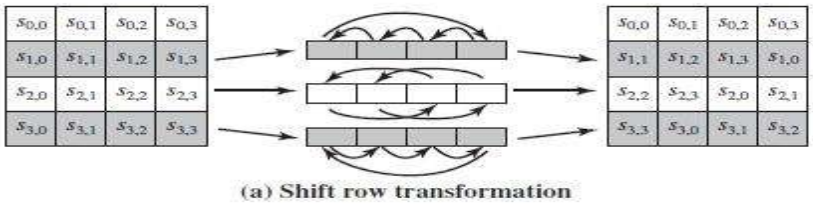
→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

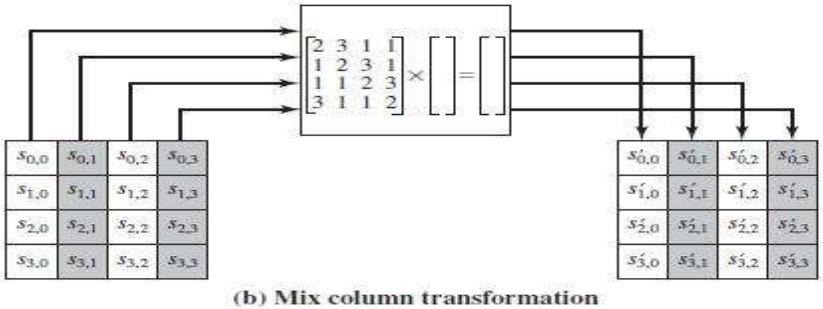
The inverse shift row transformation, called InvShiftRows, performs the circular shifts in the opposite direction for each of the last three rows, with a 1-byte circular right shift for the second row, and so on.

MixColumns Transformation

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$



(a) Shift row transformation



(b) Mix column transformation

5. Explain in details multiple encryption and Triple DES.
[C02 – L2-NOV/DEC 2012-NOV/DEC 2013-MAY/JUN 2012]

Double DES:

The simplest form of multiple encryptions has two encryption stages and two keys (Figure 6.1a). Given a plaintext P and two encryption keys K1 and K2, ciphertext C is generated as

$$C = E(K2, E(K1, P))$$

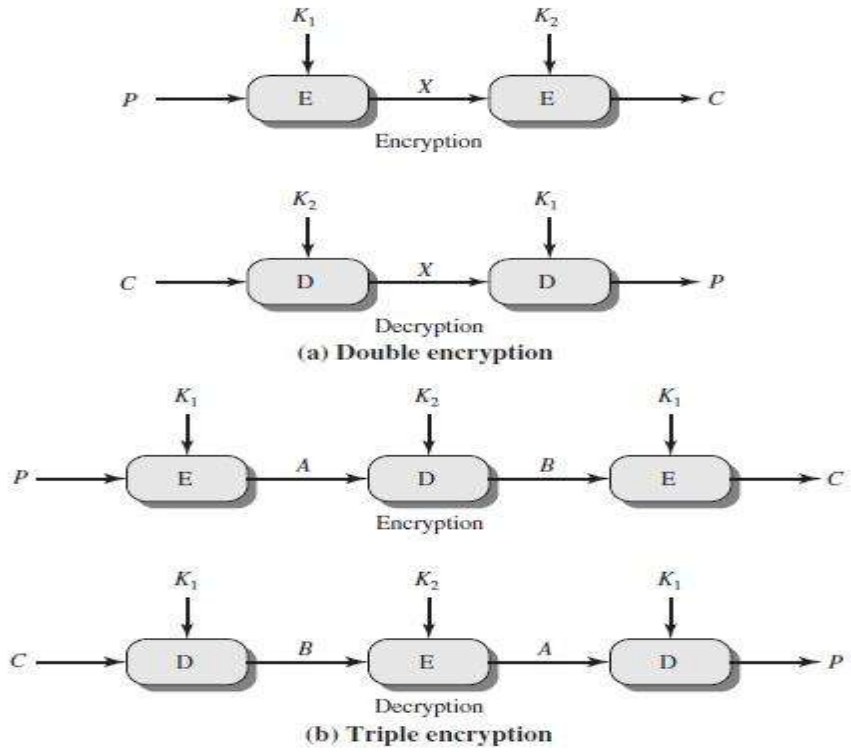


Figure 6.1 Multiple Encryption

Decryption requires that the keys be applied in reverse order:

$$P = D(K1, D(K2, C))$$

For DES, this scheme apparently involves a key length of 56 x 2 = 112 bits, of resulting in a dramatic increase in cryptographic strength. But we need to examine the algorithm more closely.

Reduction to a Single Stage:

Consider that encryption with DES is a mapping of 64-bit blocks to 64-bit blocks. In fact, the mapping can be viewed as a permutation.

That is, if we consider all 2^{64} possible input blocks, DES encryption with a specific key will map each block into a unique 64-bit block. Otherwise, if, say, two given input blocks mapped to the same output block, then decryption to recover the original plaintext would be impossible.

With 2^{64} possible inputs, how many different mappings is there that generate a permutation of the input blocks? The value is easily seen to be

$$(2^{64})! = 10^{34738000000000000000} > (10^{1020})$$

On the other hand, DES defines one mapping for each different key, for a total number of mappings:

$$2^{56} > 10^{17}$$

Therefore, it is reasonable to assume that if DES is used twice with different keys, it will produce one of the many mappings that are not defined by a single application of DES.

Although there was much supporting evidence for this assumption, it was not until 1992 that the assumption was proved [CAMP92].

Meet-in-the-Middle Attack

Thus, the use of double DES results in a mapping that is not equivalent to a single DES encryption. But there is a way to attack this scheme, one that does not depend on any particular property of DES but that will work against any block encryption cipher.

The algorithm, known as a meet-in-the-middle attack, was first described in [DIFF77]. It is based on the observation that, if we have

$$C = E(K2, E(K1, P))$$

X = E (K1, P) = D (K2, P)

Given a known pair, (P, C), the attack proceeds as follows. First, encrypt P for all 2^{56} possible values of K1 Store these results in a table and then sort the table by the values of

X. Next, decrypt C using all 2^{56} possible values of K2.

As each decryption is produced, check the result against the table for a match. If a match occurs, then test the two resulting keys against a new known plaintext-ciphertext pair. If the two keys produce the correct ciphertext, accept them as the correct keys.

For any given plaintext P, there are 2^{64} possible ciphertext values that could be produced by double DES. Double DES uses, in effect, a 112-bit key, so that there are 2^{112} possible keys.

Therefore, on average, for a given plaintext P, the number of different 112-bit keys that will produce a given ciphertext **C is $2^{112}/2^{64} = 2^{48}$.**

Thus, the foregoing procedure will produce about 2^{48} false alarms on the first (P, C) pair.

asimilar argument indicates that with an additional 64 bits of known plaintext and ciphertext, the false alarm rate is reduced to $2^{48-64} = 2^{-16}$.

Put another way, if the meet-in-the-middle attack is performed on two blocks of known plaintext-ciphertext, the probability that the correct keys are determined is $1\ 2^{-16}$.

The result is that a known plaintext attack will succeed against double DES, which has a key size of 112 bits, with an effort on the order of 2^{56}, not much more than the 2^{55} required for single DES.

Triple DES with Two Keys

An obvious counter to the meet-in-the-middle attack is to use three stages of encryption with three different keys.

This raises the cost of the known-plaintext attack to 2^{112}, which is beyond what is practical now and far into the future.

However, it has the drawback of requiring a key length of 56 x 3 = 168 bits, which may be somewhat unwieldy.

As an alternative, Tuchman proposed a triple encryption method that uses only two keys [TUCH79]. The function follows an encrypt-decrypt-encrypt (EDE) sequence

C = E (K1, D (K2, E(K1, P)))

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

C = E (K1, D (K1, E(K1, P))) = E(K1, P)

3DES with two keys is a relatively popular alternative to DES and has been adopted for use in the key management standards ANS X9.17 and ISO 8732.

The attack proceeds as follows:

1.Obtain n (P, C) pairs. This is the known plaintext. Place these in sorted on the values of P.

2.For each Pi that matches an entry in Table 1, create an entry in Table 2 consisting of the K1 value and the value of B that is produced for the (P, C) pair from Table 1, assuming that value of K1:

B = D (i, C)

At the end of this step, sort Table 2 on the values of B.

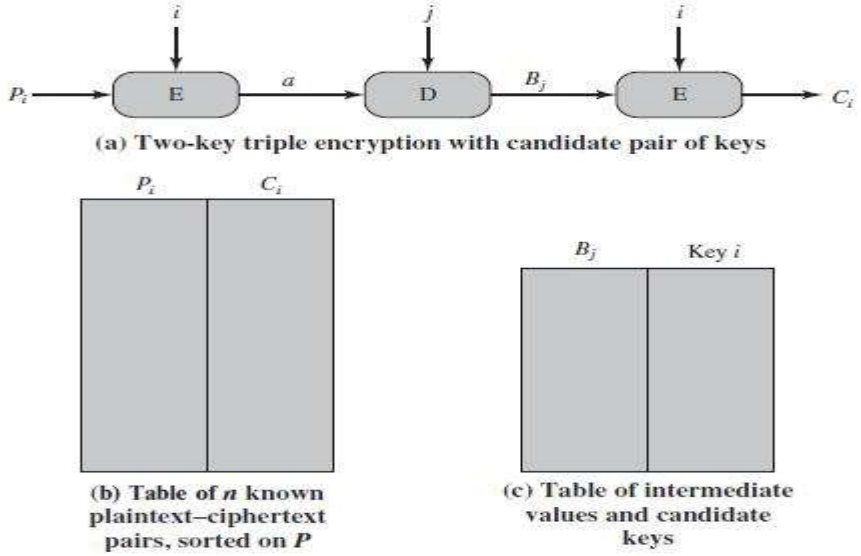


Fig : Known-Plaintext Attack on Triple DES

3. We now have a number of candidate values of K1 in Table 2 and are in a position to search for a value of K2. For each of the 2^{56}possible keys K2 = j, calculate the second intermediate value for our chosen value of a:

Bj = D (j, a)

At each step, look up Bj in Table 2. If there is a match, then the corresponding key i from Table 2 plus this value of j are candidate values for the unknown keys (K1, K2). Why? Because we have found a pair of keys (i, j) that produce a known (P, C) pair (Figure 6.2a).

4. Test each candidate pair of keys (i, j) on a few other plaintext-ciphertext pairs. If a pair of keys produces the desired ciphertext, the task is complete. If no pair succeeds, repeat from step 1 with a new value of a.

Triple DES with Three Keys

Although the attacks just described appear impractical, anyone using two-key 3DES may feel some concern.

Thus, many researchers now feel that three-key 3DES is the preferred alternative (e.g., [KALI96a]).

Three-key 3DES has an effective key length of 168 bits and is defined as follows E(K3, D(K2, E(K1, P)))Backward compatibility with DES is provided by putting K3 = K2 or K1 = K2.

6. Demonstrate that blowfish decryption is the inverse of blowfish encryption.

[**C02 – L1]**

Blowfish is a symmetric block cipher developed by Bruce. Blowfish was designed to have the following characteristics:

Fast: Blowfish encrypts data on 32-bit microprocessors at a rate of 18 clock cycles per byte

Compact. Blowfish can run in less than 5K of memory.

Simple: Blowfish's simple structure is easy to implement and eases the task of determining the strength of the algorithm.

Variably secure: The key length is variable and can be as long as 448 bits. This allows a tradeoff between higher speed and higher security.

Blowfish encrypts 64-bit blocks of plaintext into 64-bit blocks of cipher text. Blowfish is implemented in numerous products and has received a fair amount of scrutiny. So far, the security of Blowfish is unchallenged.

Subkey and S-Box generation

Blowfish makes use of a key that ranges from 32 bits to 448 bits (1 to 14 32--bit words). That key is used to generate 18 32-bit subkeys and four 8 × 32 S-boxes containing a total of 1024 32-bit entries. The total is 1042 32-bit values, or 4168 bytes.

The keys are stored in a K-array.

The subkeys are stored in the P-array:

There are four S-boxes, each with 256 32--bit entries:

The steps in generating the P-array and S-boxes are as follows:

1. Initialize first the P-array and then the four S-boxes in order using the bits of the
   fractional part of the constant π. Thus, the leftmost 32 bits of the fractional part of π become P_1, and so one For example, in hexadecimal,

2. Perform a bitwise XOR of the P-array and the K-array, reusing the words from the K-array as needed. For example, for the maximum length key (14 32-bit words),

3. a. Encrypt the 64-bit block of all zeros using the current P-array and S-arrays , replace the P1 and P2 with the output of encryption.
4. Encrypt the output of step 3 using the current P-array and S-arrays, replace P3 and P4 with the resulting ciphertext.
5. Continue this process to update all the elements of P, and then, in order, all elements of S, using at each step the output of the continuously changing Blowfish algorithm.

The update process can be summarized as follows:

Where $E_{P,S}[Y]$ is the ciphertext produced by encrypting Y using blowfish with the arrays S and P.

A total of 521 executions of the Blowfish encryption algorithm are required to produce the final S- and P arrays. Accordingly, Blowfish is not suitable for applications in which the secret key changes frequently.

Further, for rapid execution, the P- and S-arrays can be stored rather than rederived from the key each time the algorithm is used. This requires over 4 kilobytes of memory. Thus Blowfish is not appropriate for applications with limited memory, such as smart cards.

Encryption and decryption

Blowfish uses two primitive operations:

Addition: Addition of words, denoted by +, is performed modulo 2^{32}.
Bitwise exclusive—OR: This operation is denoted by $\oplus$.

The important thing about these two operations is that they do not commute. This makes cryptanalysis more difficult.

The resulting ciphertext is contained in the two variables LE_{17}; and RB_{17}. The function F is shown in Figure 4.10. The 32-bit input to F is divided into 4 bytes. If we label those bytes a, b, c, and d, then the function can be defined as follows:

Thus, each round includes the complex use of addition modulo 2^{32} and XOR, plus v substitution using S-boxes.

Decryption, shown in Figure 4.9b, is easily derived from the encryption algorithm. In this case, the 64 bits of ciphertext are initially assigned to the two one-word variables LD_0 and RD_0.

We use the variables LD_i and RD_i ; to refer to the left and right half of the data after round *i*. As with most block ciphers, Blowfish decryption involves using the subkeys in reverse order- However, unlike most block ciphers, Blowfish decryption occurs in the same algorithmic direction as encryption, rather than the reverse. The algorithm can be defined as follows:

7. In the RCS-CBC-Pad mode, there are from one to bb bytes of padding. Why not allow zero bytes of padding? That is, if the message to be encrypted IS an integer multiple of the block size, why not refrain from padding? [C02 – L2]

RC5 is a symmetric encryption algorithm developed by Ron Rivest. RC5 was designed to have the following characteristics:

- **Suitable for hardware or software:** RC5 uses only primitive computational operations commonly found on microprocessors
- **Fast:** To achieve this, RC5 is a simple algorithm and is word oriented. The basic operations work on full words of data at a time.
- **Adaptable to processors of different word lengths:** The number of bits in a word is a parameter of RC5; different word lengths yield different algorithms.
- **Variable number of rounds:** The number of rounds is a second parameter of RC5. This parameter allows a tradeoff between higher speed and higher security.
- **Variable-length key:** The key length is a third pararneter of RC5- Again, this allows a tradeoff between speed and security
- **Simple:** RCS'S simple structure is easy to implement and eases the task of determining the strength of the algorithm.
- **Low memory requirement:** A low memory requirement makes RC5 suitable for smart cards and other devices with restricted memory.
- **High security:** RC5 is intended to provide high security with suitable parameters.
- **Data-dependent rotations:** RC5 incorporates rotations (circular bit shifts) whose amount is data dependent- This appears to strengthen the algorithm against cryptanalysis.
- RC5 has been incorporated into RSA Data Security, Inc-'s major products, including BSAFE, JSAFE, and S/MAIL.

RC5 Parameters

RC5 is actually a family of encryption algorithms determined by three parameters, as follows:

Key Expansion

RC5 performs a complex set of operations on the secret key to produce a total of *t* subkeys. Two subkeys are used in each round, and two subkeys are used on an additional operation that is not part of any round, so $t = 2r + 2$. Each subkey is one Word (w bits) in length.

Figure 4-11 illustrates the technique used to generate subkeys; The subkeys are stored in a *t*-word array labeled S[0], S[1], ….,' S[t-1]. Using the parameters *r* and *w* as inputs, this array is initialized to a particular fixed pseudorandom bit pattern. Then the *b*-byte key, K[0…. *b* - 1], is converted into a *c*-word array L[0…. *c -1*]. On a little endian machine, this is accomplished by zeroing out the array *L* and copying the string *K* directly into the memory positions represented by L.

If *b* is not an integer multiple of w, then a portion of L at the right end remains zero-Finally, a mixing operation is performed that applies the contents of *L* to the initialized value of S to produce a final value for the array S.

where addition is performed modulo 2^w. The initialized array S is then mixed with the key array L to produce a final array S of subkeys. For this purpose, three passes are made through the larger of the two arrays; the smaller array may be handled more times:

Encryption

RC5 uses three primitive operations (and their inverses):

Addition: Addition of words, denoted by +, is performed modulo 2^w. The inverse operation, denoted by -, is subtraction modulo 2^w.

Bitwise exclusive-OR: This operation is denoted by $\oplus$.

Left circular rotation: The cyclic rotation of word *x* left by *y* bits is denoted by

x <<< *y*. The inverse is the right circular rotation of word *x* by *y* bits, denoted by

x >>> *y*.

Decryption

Decryption, shown in Figure 4-12b, is easily derived from the encryption algorithm. In this case, the 2*w* bits of ciphertext are initially assigned to the two one-word variables LD_r, and RD_r. We use the variables LD_i and RD_i to refer to the left and right half of the data before round *i* has begun, where the rounds are numbered from *r* down to 1.

RC5 Modes

To enhance the effectiveness of RC5 in interoperable implementations, RFC 2040 defines four different modes of operation:

RC5 block cipher: This is the raw encryption algorithm that takes a fixed—size input block (2*w* bits) and produces a ciphertext block of the same length using a transformation that depends on a key.

RCS-CBC: This is the cipher block chaining mode for RC5- CBC. CBC processes messages whose length is a multiple of the RC5 block size (multiples of 2w bits. CBC provides enhanced security compared to ECB because repeated blocks of plaintext produce different blocks of ciphertext.

RCS-CBC-Pad: This is a CBC style of algorithm that handles plaintext of any length- The ciphertext will be longer than the plaintext by at most the size of a single RC5 block.

RCS-CTS: This is the ciphertext stealing mode, which is also a CBC style of algorithm- This mode handles plaintext of any length and produces ciphertext of equal length.

8. Explain in detail principles of public key cryptography. [C02 – L2]

Principles of public key cryptography.

Public-Key Cryptosystems
Applications for Public-Key Cryptosystems
Requirement for Public-Key Cryptography
Public-Key Cryptanalysis

Key distribution under symmetric key encryption requires either (1) that two communicants already share a key, which someone has been distributed to them or (2) the use of a key distribution center.

The second problem that Diffie pondered and one that was apparanently unrelated to the first was that of “Digital signatures”.

Public key cryptosystems

characteristics:

It is computationally infeasible to determine the decryption key given only the 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 decry

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 and 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 theplaintext and the key. For a given message, two different keys will produce two different cipher texts.

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

The essential steps are the following:

Each user generates a pair of keys to be used for encryption and decryption of messages.

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.

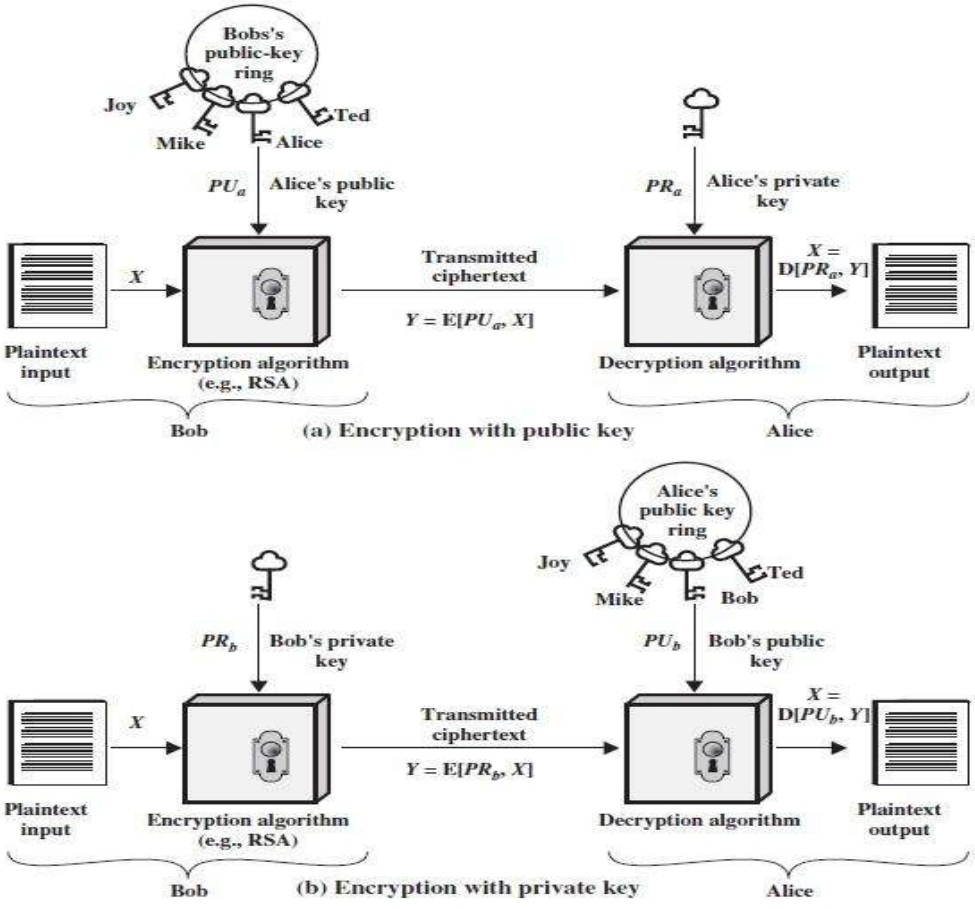
If A wishes to send a confidential message to B, A encrypts the Message

using B's public key. When B receives the message, it decrypts using its private key. No other recipient can decrypt the message because only B knows B's private key.

With this approach, all participants have access to public keys and private keys are generated locally by each participant and therefore, need not be distributed. As long as a system controls its private key, its incoming Communication is secure.

To discriminate between the two, we refer to the key used in symmetric encryption as secret key.

The two keys used for asymmetric encryption are referred as the public key and the private key



The other approach (using sender's private key for encryption and sender's public key for decryption) will provide authentication which is illustrated in the following diagram There is some source A that produces a message in plaintext, X = [X1, X2, … , XM]. TheM elements of X are letters in some finite alphabet.

The message is intended for destination B. B generates a related pair of keys: a public key, PUb, and a private key, PRb. PRb is known only to B, whereas PUb is publicly available and therefore accessible by A. With the message X and the encryption key PUb as input, A forms the ciphertext

$$Y = [Y1, Y2, \ldots, YN]:$$
$$Y = E(PUb, X)$$

The intended receiver, in possession of the matching private key, is able to invert the transformation X = D(PRb,Y) must be kept in plaintext to be used for practical purposes. A copy also must be stored in ciphertext so that the origin and contents can be verified in case of a dispute.

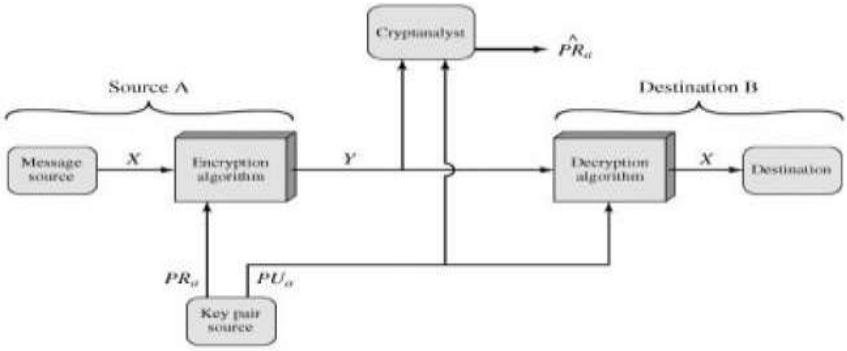
A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document. Such a block, called an authenticator, must have the property that it is infeasible to change the document without changing the authenticator.

If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin, content, and sequencing That is, the message being sent is safe from alteration but not from eavesdropping.

This is obvious in the case of a signature based on a portion of the message, because the rest of the message is transmitted in the clear.

Even in the case of complete encryption, as shown in Figure 9.3, there is no protection of confidentiality because any observer can decrypt the message by using the sender's public key.

It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme (Figure 9.4):



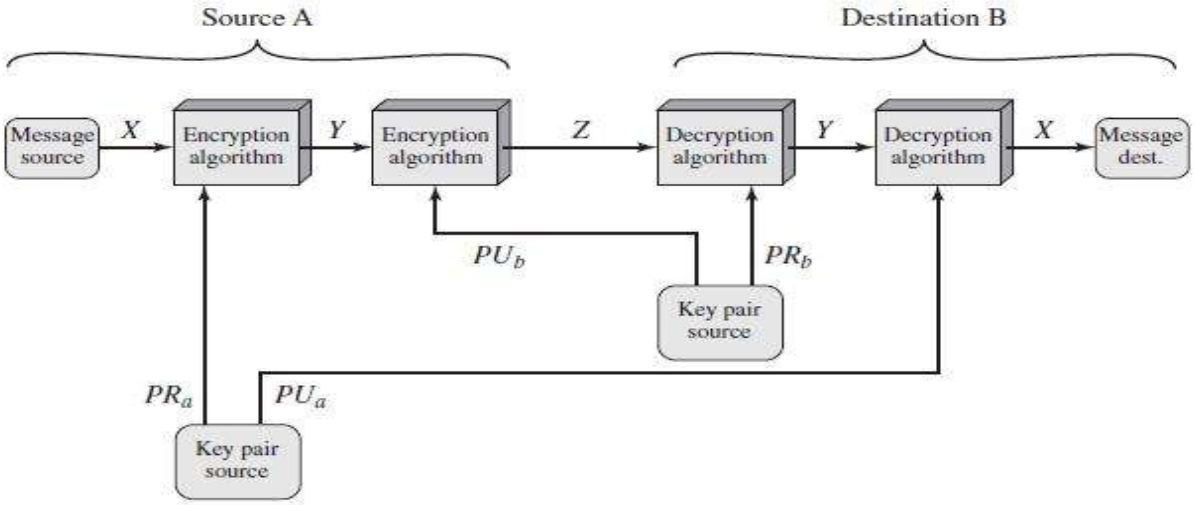
If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin, content, and sequencing That is, the message being sent is safe from alteration but not from eavesdropping.

This is obvious in the case of a signature based on a portion of the message, because the rest of the message is transmitted in the clear.

Even in the case of complete encryption, as shown in Figure 9.3, there is no protection of confidentiality because any observer can decrypt the message by using the sender's public key. It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme (Figure 9.4):

The disadvantage

This approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.



Application for Public –Key Cryptosystem:

Encryption /Decryption: the sender encrypts a message i^{th} the recipient's public key.

Digital Signature: The sender "signs" a message with its private key .Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

Key exchange: Two sides cooperate to exchange a session key .Several different approaches are possible, involving the private key of one or both parties.

Requirements for public key cryptography

1. It is computationally easy for a party B to generate a pair (public key PUb, private key PRb).

2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:

C = E(PUb, M)

3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

M = D(PRb,C) = D[PRb, E(PUb,M)]

4. It is computationally infeasible for an adversary, knowing the public key, PUb, to determine the private key, PRb.

5. It is computationally infeasible for an adversary, knowing the public key, PUb, and a ciphertext, C, to recover the original message, M. We can add a sixth requirement that, although useful, is not necessary for all public-key applications

6. The two keys can be applied in either order:

M = D[PUb, E(PRb, M)] = D[PRb, E(PUb, M)]

Public-Key Cryptanalysis

As with symmetric encryption, a public-key encryption scheme is vulnerable to a brute-force attack. The countermeasure is the same: Use large keys.

9. Explain the RSA algorithm and its key generation, encryption and decryption Operations. Or Perform encryption and decryption using RSA Alg. for the following.P=7; q=11; e=17; M=8. or Explain RSA algorithm in detail with an example.
[C02 – L2-APR/MAY-2011-NOV/DEC 2012-MAY/JUN 2013-MAY/JUN 2014 - NOV/DEC 2014]

Introduction:

It was developed by Rivest, Shamir and Adleman. This algorithm makes use of an expression with exponentials.

Plaintext is encrypted in blocks, with each block having a binary value less than some number n.

The RSA scheme is a 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 21024.

Description of the Algorithm

That is, the block size must be less than or equal to $\log_2 (n)$; in practice, the block size is k-bits, where $2^k < n < 2^{k+1}$. Encryption and decryption are of the following form, for some Plaintext block M and Cipher text block C:

$$\mathbf{C = M^e \bmod n}$$

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

$$= (M^e)^d \bmod n$$

$$= M^{ed} \bmod n$$

Both the sender and receiver 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 KU = {e, n} and a private key of KR = {d, n}.

For this algorithm to be satisfactory for public key encryption, the following requirements must be met:
It is possible to find values of e, d, n such that $M^{ed =}$ M mod n for all M<n.
It is relatively easy to calculate M^e and C^d for all values of M<n.

It is infeasible to determine d given e and n.

Let us focus on the first requirement. We need to find the relationship of the form:

$$M^{ed} = M \bmod n$$

A corollary to Euler's theorem fits the bill: Given two prime numbers p and q Integers, n and m, such that n=pq and 0<m<n, and arbitrary integer k, the following relationship holds

$$m^{k\Phi(n)+1} = m^{k(p-1)(q-1)+1} = m \bmod n$$

where Φ(n) – Euler totient function, which is the number of positive integers less than n and relatively prime to n.we can achieve the desired relationship, if

$$ed = k\Phi(n)+1$$

This is equivalent to saying:

$$ed \equiv 1 \bmod \Phi(n)$$
$$d = e^{-1} \bmod \Phi(n)$$

That is, e and d are multiplicative inverses mod Φ(n). According to the rule of modular arithmetic, this is true only if d (and therefore e) is relatively prime to Φ(n). Equivalently, gcd (Φ(n), d) = 1.

The steps involved in RSA algorithm for generating the key are

Select two prime numbers, p = 17 and q = 11.

Calculate n = p*q = 17*11 = 187

Calculate Φ (n) = (p-1) (q-1) = 16*10 = 160.

Select e such that e is relatively prime to Φ (n) = 160 and less than Φ(n); we choose e =7.

Determine d such that ed ≡ 1 mod Φ (n) and d<160. The correct value is d = 23, because 23*7 = 161 = 1 mod 160.

Computational Aspects

We now turn to the issue of the complexity of the computation required to use RSA. There are actually two issues to consider: encryption/decryption and key generation.

Let us look first at the process of encryption and decryption and then consider key generation.

The Private-Key operation $M=C^4$ mod n is implemented as follows:

1. Generate a secret random number r between 0 and n-1.
2. Compute C' = C (r^e) mod n, where e is the public exponent.
3. Compute M'= (C') d mod n with the ordinary RSA implementation.
4. Compute M=M'r^{-1} mod n.In this equitation, r^{-1} is the multiplicative inverse of r mod n.

10. Explain the Key management in detail. [C02 – L2]

Introduction

One of the major roles of public-key encryption has been to address the problem of key distribution. There are actually two distinct aspects to the use of public-key cryptography in this regard:

- The distribution of public keys
- The use of public-key encryption to distribute secret keys

Distribution of Public Keys

Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes:

- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

Public Announcement of Public Keys

On the face of it, the point of public-key encryption is that the public key is public. Thus, if there is some broadly accepted public-key algorithm, such as RSA, any participant can send his or her public key to any other participant or broadcast the key to the community at large (Figure 10.1).

For example, because of the growing popularity of PGP (pretty good privacy), which makes use of RSA, many PGP users have adopted the practice of appending their

public key to messages that they send to public forums, such as USENET newsgroups and Internet mailing lists.

Figure 10.1. Uncontrolled Public-Key Distribution

Although this approach is convenient, it has a major weakness. Anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key.

Until such time as user A discovers the forgery and alerts other participants, the forger is able to read all encrypted messages intended for A and can use the forged keys for authentication

Publicly Available Directory

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys.

Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization (Figure 10.2). Such a scheme would include the following elements:

1. The authority maintains a directory with a {name, public key} entry for each Participant.

2. Each participant registers a public key with the directory authority. Registration would
have to be in person or by some form of secure authenticated communication.

3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.

4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

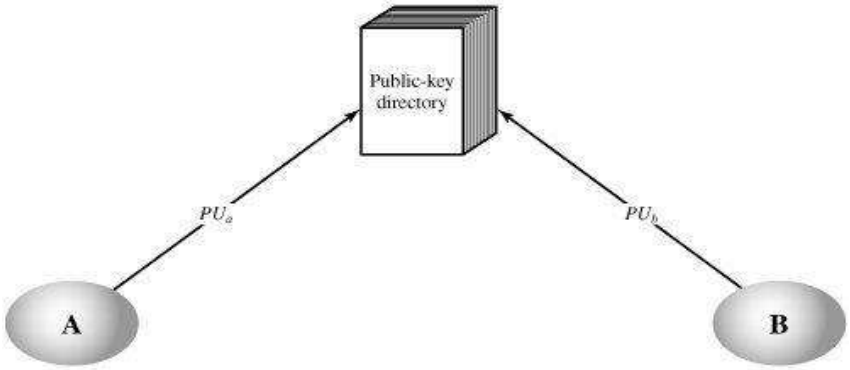


Figure 10.2. Public-Key Publication

This scheme is clearly more secure than individual public announcements but still has vulnerabilities.
If an adversary succeeds in obtaining or computing the private key of the directory authority, the adversary could authoritatively pass out counterfeit public keys and subsequently impersonate any participant and eavesdrop on messages sent to any participant.

Another way to achieve the same end is for the adversary to tamper with the records kept by the authority.

Public-Key Authority

Stronger security for public-key distribution can be achieved by providing tighter control over thedistribution of public keys from the directory.

A typical scenario is illustrated in Figure 10.3, As before, the scenario assumes that a central authority maintains a dynamic directory of public keys of all participants.

In addition, each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.

The following steps

1. A sends a timestamped message to the public-key authority containing a request for the current public key of B.
2. The authority responds with a message that is encrypted using the authority's private key, PR_{auth}

Thus, A is able to decrypt the message using the authority's public key. Therefore, A is assured that the message originated with the authority. The message includes the following:

- B's public key, $\boldsymbol{P_{Ub}}$ which A can use to encrypt messages destined for B
- The original request, to enable A to match this response with thecorresponding earlier request and to verify that the original request was not altered before reception by the authority

The original timestamp, so A can determine that this is not an old message from the authority containing a key other than B's current public key

3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (*IDA*) and a nonce (*N*1), which is used to identify this transaction uniquely.

4. B retrieves A's public key from the authority in the same manner as A retrieved B's public key.

5. At this point, public keys have been securely delivered to A and B, and they may begin their protected exchange. However, two additional steps are desirable

6. B sends a message to A encrypted with PU_a and containing A's nonce (*N*1) as well as a new nonce generated by B (*N*2) Because only B could have decrypted message (3), the presence of *N*1 in message (6) assures A that the correspondent is B.

7. A returns *N*2, encrypted using B's public key, to assure B that its correspondent is A.

Thus, a total of seven messages are required. However, the initial four messages need be used only infrequently because both A and B can save the other's public key for future use, a technique known as caching.

Periodically, a user should request fresh copies of the public keys of its correspondents to ensure currency.

11. Explain the Diffie-Hellman Key. [C02 – L2]

Introduction

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. Recall from Chapter 8 that a primitive root of a prime number p is 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 mod p, a2 mod p,c, ap-1 mod 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 K ai (mod p)** where 0 … i … (p - 1)

The Algorithm

Figure 10.1 summarizes the Diffie-Hellman key exchange algorithm. For this scheme, there are two publicly known numbers: a prime number q and an integer a that is a primitive root of q. Suppose the users A and B wish to create a shared key.

User A selects a random integer XA 6 q and computes **YA = aXA mod q.** Similarly, user B independently selects a random integer XB 6 q and computes YB = aXB mod q. Each side keeps the X value private and makes the Y value available publicly to the other side.

Thus, XA is A’s private key and YA is A’s corresponding public key, and similarly for

B. User A computes the key as K = (YB)XA mod q and user B computes the key as K = (YA)XB mod q. These two calculations produce identical results:

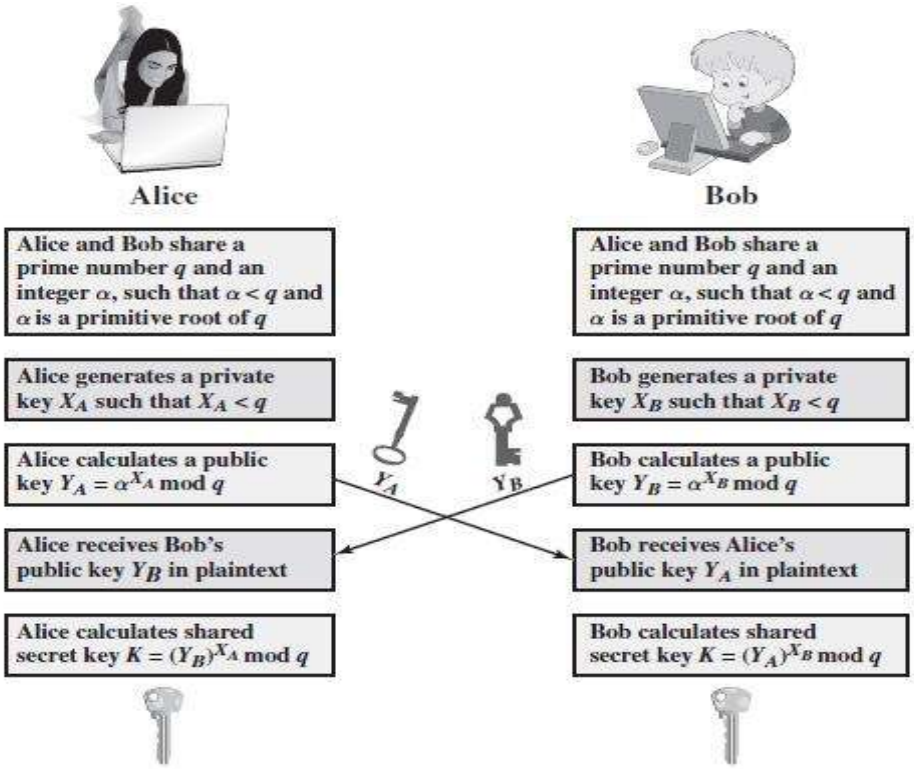


Figure 10.1 The Diffie-Hellman Key Exchange

The result is that the two sides have exchanged a secret value. Typically, this secret value is used as shared symmetric secret key. Now consider an adversary who can observe the key exchange and wishes to determine the secret key K.

Because XA and XB are private, an adversary only has the following ingredients to work with: q, a, YA, and YB. Thus, the adversary is forced to take a discrete logarithm to determine the key.

Key Exchange Protocols

Figure 10.1 shows a simple protocol that makes use of the Diffie-Hellman calculation. Suppose that user A wishes to set up a connection with user B and use a secret key to encrypt messages on that connection.

User A can generate a one-time private key XA, calculate YA, and send that to user B. User B responds by generating a private value XB, calculating YB, and sending YB to user A. Both users can now calculate the key.

12. Explain the Elliptic curve arithmetic Contents. [C02 – L2]

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.

The principal attraction of ECC, compared to RSA, is that it appears to offer equal security for a far smaller key size, thereby reducing processing overhead. On the other hand, although the theory of ECC has been around for some time, it is only recently that products have begun to appear and that there has been sustained cryptanalytic interest in probing for weaknesses. Accordingly, the confidence level in 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 book. This section and the next give some background on elliptic curves and ECC.

We begin with a brief review of the concept of abelian group. Next, we examine the concept of elliptic curves defined over the real numbers. This is followed by a look at elliptic curves defined over finite fields. Finally, we are able to examine elliptic curve ciphers.

Abelian Groups an abelian group G, sometimes denoted by {G, .}, is a set of elements with a binary operation, denoted by . , that associates to each ordered pair (a, b) of elements in G an element (a . b) in G, such that the following axioms are obeyed:3

For example, Diffie-Hellman key exchange involves multiplying pairs of nonzero integers modulo a prime number q. Keys are generated by exponentiation

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.

Before looking at this, we first look at elliptic curves in which the variables and coefficients are real numbers. This case is perhaps easier to visualize.

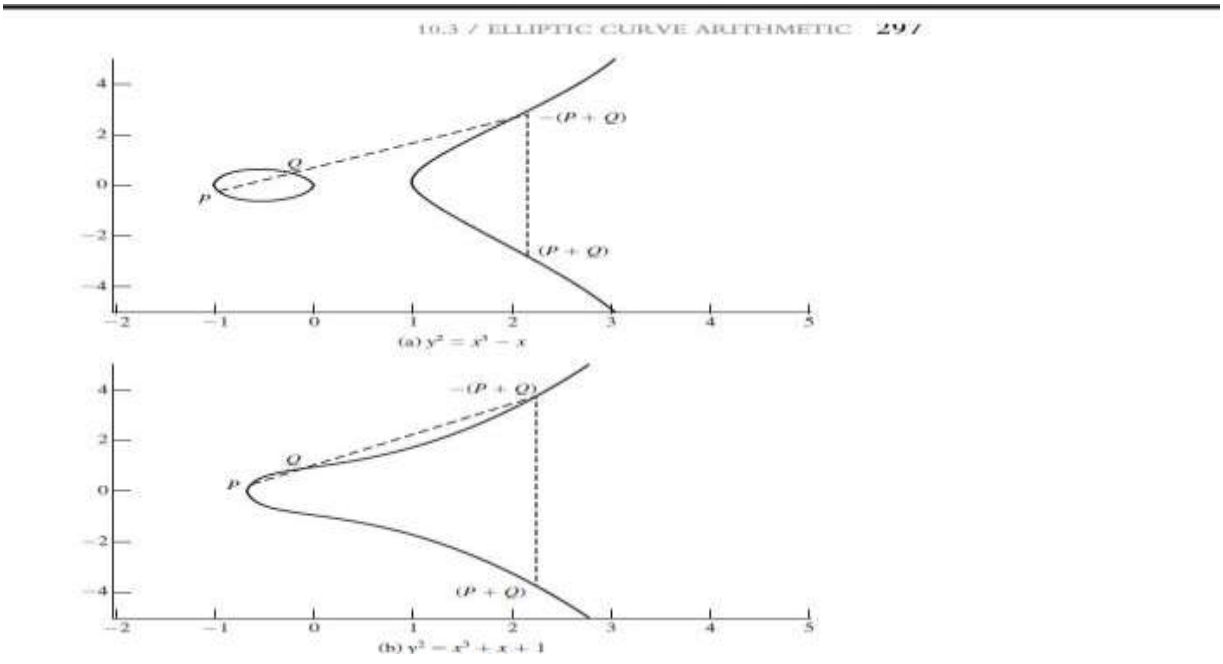
Elliptic Curves over Real Numbers

Elliptic curves are not ellipses. They are so named because they are described by cubic equations, similar to those used for calculating the circumference of an ellipse. In general, cubic equations for elliptic curves take the following form, known as a **Weierstrass equation:**

where a, b, c, d, e are real numbers and x and y take on values in the real numbers.4
For our purpose, it is sufficient to limit ourselves to equations.
Such equations are said to be cubic, or of degree 3, because the highest
Exponent they contain is a 3. Also included in the definition of an elliptic curve is a single element denoted O and called the point at infinity or the zero point, which we discuss subsequently. To plot such a curve, we need to compute

For given values of a and b, the plot consists of positive and negative values of y for each value of x. Thus, each curve is symmetric about y = 0. Figure 10.4 shows two examples of elliptic curves. As you can see, the formula sometimes produces weirdlooking curves. Now, consider the set of points E(a, b) consisting of all of the points (x, y) that satisfy Equation (10.1) together with the element O. Using a different value of the pair (a, b) results in a different set E(a, b).

Using this terminology, the two curves in Figure 10.4 depict the sets E(-1, 0) and E(1, 1), respectively.



Unit – III

Hash Functions And Digital Signatures

Part – A

1. List out the different techniques of distributing the public key.[CO3-L1]

Public announcement
Public available directory
Public key authority
Public key certificate

**2. What are the attacks that can be performed in the networks?(or)
List the authentication requirements?** [**CO3-L1-May/Jun 2014]**

Disclosure
Traffic analysis
Masquerade
Content modification
Sequence modification
Timing modification
Source repudiation
Destination repudiation

3. Mention the various ways of producing an authenticator. [CO3-L1]

Message encryption
Message Authentication Code (MAC)
Hash function

4. Differentiate Message Authentication Code and Hash function. **[CO3-L2]**

In MAC, a public function of the message and a secret key are used to produce a fixed length authenticator.

A hash function accepts a variable size message as input and produces a fixed size output (hash code) which is similar to MAC. But hash code does not use a key.

5. Define one way property, weak collision resistance and strong collision resistance of hash function. [CO3-L1]

For any given value h, it is computationally infeasible to find x such that H(x) = h - one way property.
For any given block x, it is computationally infeasible to find y ≠ x with H(y) = H(x) - weak collision resistance.
It is computationally infeasible to find any pair (x, y) such that H(x) = H(y) - strong collision property.

6. Write the purpose of hash function? [CO3-L1]

The purpose of hash function is a public function that maps a message of any length into a fixed length hash value which serves as a authenticate.

7. What is simple hash function? [CO3-L1]

Simple hash function is the bit by bit exclusive OR of every block.
This method produces a simple parity for each bit position also known as longitudinal redundancy check

8. What is the primitive root of a number? [CO3-L1-NOV/ DEC 2012]

We can define a primitive root of a number p as one whose powers generate all the integers from 1 to p-1.

That is p, if a is a primitive root of the prime number p then the numbers.

9. What is message authentication? [CO3-L1]

It is a procedure that verifies whether the received message comes from assigned source has not been altered.

It uses message authentication codes, hash algorithms to, authenticate the message.

10. Define the classes of message authentication function. [CO3-L1]

Message encryption: The entire cipher text would be used for authentication.

Message Authentication Code: It is a function of message and secret key produce a fixed length value.

Hash function: Some function that map a message of any length to fixed length which serves as authentication.

11. What are the requirements for message authentication? [CO3-L1]

The requirements for message authentication are

Disclosure

Traffic Analysis

Masquerade

Content modification

3. Sequence modification

Timing modification.

Source repudiation

Destination repudiation

12. What you meant by hash function? [CO3-L1-NOV/DEC 2013]

Hash function accept a variable size message M as input and produces a fixed size hash code H(M) called as message digest as output.

It is the variation on the message authentication code.

13. Differentiate MAC and Hash function? [CO3-L3-NOV/DEC 2013]

MAC: In Message Authentication Code, the secret key shared by sender and receiver. The MAC is appended to the message at the source at a time which the message is assumed or known to be correct.

Hash Function: The hash value is appended to the message at the source at time when the message is assumed or known to be correct. The hash function itself not considered to be secret.

14. What are the requirements of the hash function? [CO3-L1]

H can be applied to a block of data of any size.

H produces a fixed length output.

H(x) is relatively easy to compute for any given x, making both Hardware and software implementations practical.

15. What you meant by MAC? [CO3-L1]

MAC is Message Authentication Code. It is a function of message and secret key which produce a fixed length value called as MAC.

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

Where M = variable length message

K = secret key shared by sender and receiver.

C(*K, M*) = fixed length authenticator.

16. What is the meet in the middle attack? [CO3-L1]

This is the cryptanalytic attack that attempts to find the value in each of the range and domain of the composition of two functions such that the forward mapping of one through the first function is the same as the inverse image of the other through the second function-quite literally meeting in the middle of the composed function.

17. What is the role of compression function in hash function? [CO3-L1]

The hash algorithm involves repeated use of a compression function f, that takes two inputs and produce a n-bit output.

At the start of hashing the chaining variable has an initial value that is specified as part of the algorithm.

The final value of the chaining variable is the hash value usually b>n; hence the term compression.

18. Distinguish between direct and arbitrated digital signature? [CO3-L3]

Direct digital signature Arbitrated Digital Signature The direct digital signature involves only the communicating parties.

The arbiter plays a sensitive and crucial role in this digital signature.

This may be formed by encrypting the entire message with the sender's private key.

Every signed message from a sender x to receiver y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check its origin and content.

19. What requirements should a digital signature scheme should satisfy? [CO3-H1]

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

20. Mention the fundamental idea of HMAC. **[CO3-H1-MAY/JUNE 2009**]

specified as Internet standard RFC2104

uses hash function on the message:

where K+ is the key padded out to size and opad, ipad are specified padding constants overhead is just 3 more hash calculations than the message needs alone any hash function can be used

eg. MD5, SHA-1, RIPEMD-160, Whirlpool

21. Explain needha-schroder protocol? [CO3-L2]

Needham-Schroeder Protocol

Original third-party key distribution protocol for session between A B mediated by KDC

protocol overview is:

1. A->KDC: *IDA* || *IDB* || *N*

2. KDC -> A: EKa [Ks || *IDB* || *N1* || E*Kb* [*Ks*||*IDA*]]

3. A -> B: *EKb* [*Ks*||*ID]*

4. B -> A: *EKs* [*N]*

5. A -> B: *EKs* [f(*N2*)]

22. Explain denning as protocol? [CO3-H1]

Denning as presented the following:

1. A -> AS: *IDA* || *ID*

2. AS -> A: EPRas[*IDA*||PU*a*||T] || EPRas[*IDB*||PU*b*||T]

3. A -> B: EPRas[*IDA*||PU*a*||T] || EPRas[*IDB*||PU*b*||T] || EPUb[EPRas[K*s*||T]]

23. What is one way authentication? [CO3-L1-NOV/DEC 2012]

One-Way Authentication required when sender & receiver are not in communications at same time (eg. email) have header in clear so can be delivered by email system may want contents of body protected &sender authenticated

24. What are the two approaches of digital signature? [CO3-L1-NOV/DEC 2012]

1. Direct digital signature.
2. arbitrated digital signature.

25. What is Masquerade? [CO3-L1]

A masquerade occurs when one entity pretends to be an another entity.

A masquerade attack usually includes one of the forms of active attacks.

26. Why SHA is more secure than md5? **[CO3-L1-MAY/JUNE 2009]**

1. SHA requires 2^{160} attacks to find the original message, whereas in MD5 2^{128} message is enough
2. SHA requires 2^{80} messages to find 2 messages having same MD, in MD5 only 2^{64} message is enough

27. What is discrete logarithm? [CO3-L1-Apr/May 2011-MAY/JUN 2014]

Discrete logarithms are fundamental to a number of public key algorithms, including Diffie-Hellman key exchange and DSA.

28. What do u meant by one-way property in hash function? [CO3-L1-Apr/May 2011]

Performance: Easy to compute H(m)
One-way property: Given H(m) but not m, it's computationally infeasible to find m

Weak collision resistance (free): Given H(m), it's computationally infeasible to find m such that H(m') = H(m).
Strong collision resistance (free): Computationally infeasible to find m1, m2 such that
H(m1) = H(m2)

29. Write down the difference between the public key and private key? [CO3-L1-May/Jun 2012]

Public Key:
Public key cryptography (Asymmetric cryptography) uses two keys -- one is called the public key and the other the private key.
The public key and corresponding private key are mathematically related

Private Key:
Private key cryptography (symmetric cryptography) uses a single key.
If you want to send Alice a message using private key cryptography you encrypt the message with a private key (that Alice and you (but not anyone who you don't want to read the message) both have access to) and send her the cipher text.
Alice uses the same private key that was used to encrypt the message to decipher the message on her side.

30. Write the difference between MD4 and secure hash. **[CO3-L1-MAY/JUN 2013]**

MD4

- precursor to MD5
- also produces a 128-bit hash of message

- has 3 rounds of 16 steps vs 4 in MD5
- design goals:
- collision resistant (hard to find collisions)
- direct security (no dependence on "hard" problems)
- fast, simple, compact
- favours little-endian systems (eg PCs)

SECURE HASH

- SHA is developed by national institute of standards and technology along with nsa.
- SHA is modified version of MD
- Input to SHA is message less than 2^{64}
- Output is message digest 160 bits in length
- SHA is designed to be computationally infeasible
    1. Obtain original message, given its message digest.
    2. Find two messages producing same message digest

31. What are the security services provided by the Digital Signature? [CO3-L1-NOV/DEC 2014]

In a *key-only* attack, the attacker is only given the public verification key.

In a *known message* attack, the attacker is given valid signatures for a variety of messages known by the attacker but not chosen by the attacker.

In an *adaptive chosen message* attack, the attacker first learns signatures on arbitrary messages of the attacker's choice

32. What are the Birthday attacks? [CO3-L1-MAY/JUN 2014]

It is a one type of attack .it means an opponent would have try about $2^{(\text{hashcode size-1})}$ messages to find one that matches the hash code of intercepted message.

Ex: if encrypted hash code c is transmitted with corresponding un encrypted message m,then opponent need to find m, **h(m')=h(m)** to substitute another message and fool the receiver. An average opponent tries 2^{63} to find one matches of hash code

33. What are the performance difference between MD5, SHA-512? [CO3-L1-NOV/DEC 2014]

MD5

MD5 is a message digest (hash) algorithm created by Ron Rivest of MIT (the 'R' in RSA).

Its input is a message of arbitrary length and its output is a 128 bit message digest.

SHA

SHA stands for Secure Hash Algorithm. It is a hash algorithm based on MD4, a precursor to MD5, created by the National Institute of Standards and Technology in 1993.

It produces a 160-bit message digest. Because of the longer output, it is harder to produce another message that yields the same digest.

On the other hand, it requires more computational steps than MD5 (80 vs. 60), making it approximately 25% slower.

34. List the main Goals of MD4. [CO3-L1]

Security: There is the usual requirement for a hash code namely, that it be cornputationally infeasible to find two messages that have the same message digest

Speed: The algorithm should lend itself to implementations in software that executes rapidly- In particular, the algorithm is intended to be fast on 32-bit architectures.

Simplicity and compactness: The algorithm should be simple to describe and simple to program, without requiring large programs or substitution tables.

Favor little endian architecture: Some processor architectures (such as the Intel 80xxx and Pentium line) store the least significant byte of a word in the low—address byte position (little endian).

35. What are the steps involved in MD5 Logic? [CO3-L1]

Step 1: Append padding bits
Step 2: Append length
Step 3: Initialize MD buffer
Step 4: Process message in 512-bit (16-word) blocks
Step 5: Output

36. What are the steps involved SHA-512 Logic[CO3-L1]

Step 1: Append padding bits
Step 2: Append length.
Step 3: Initialize hash buffer.
Step 4: Process message in 1024-bit (128-word) blocks.
Step 5: Output.

PART B

1. Explain the message Authentication requirement. [CO3-L2]

Authentication requirement

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. This includes the creation of messages by an opponent that are purported to come from an authorized entity.

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. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed.

7. Source repudiation: Denial of transmission of message by source.

8. Destination repudiation: Denial of receipt of message by destination.

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.

2.Explain the Authentication function. [CO3-L2]

Authentication function

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.

These may be grouped into three classes

Hash function
Message Encryption
Message Authentication Code

.

• **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

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

A variation on the message authentication code is the one way hash function. As with MAC, a hash function accepts a variable size message M as input and produces affixed-size output, referred to as hash code h**=H(M).**

Unlike a MAC, a hash code does not use a key but is a function only of the input message. The hash code is also referred to as a message digest or hash value.

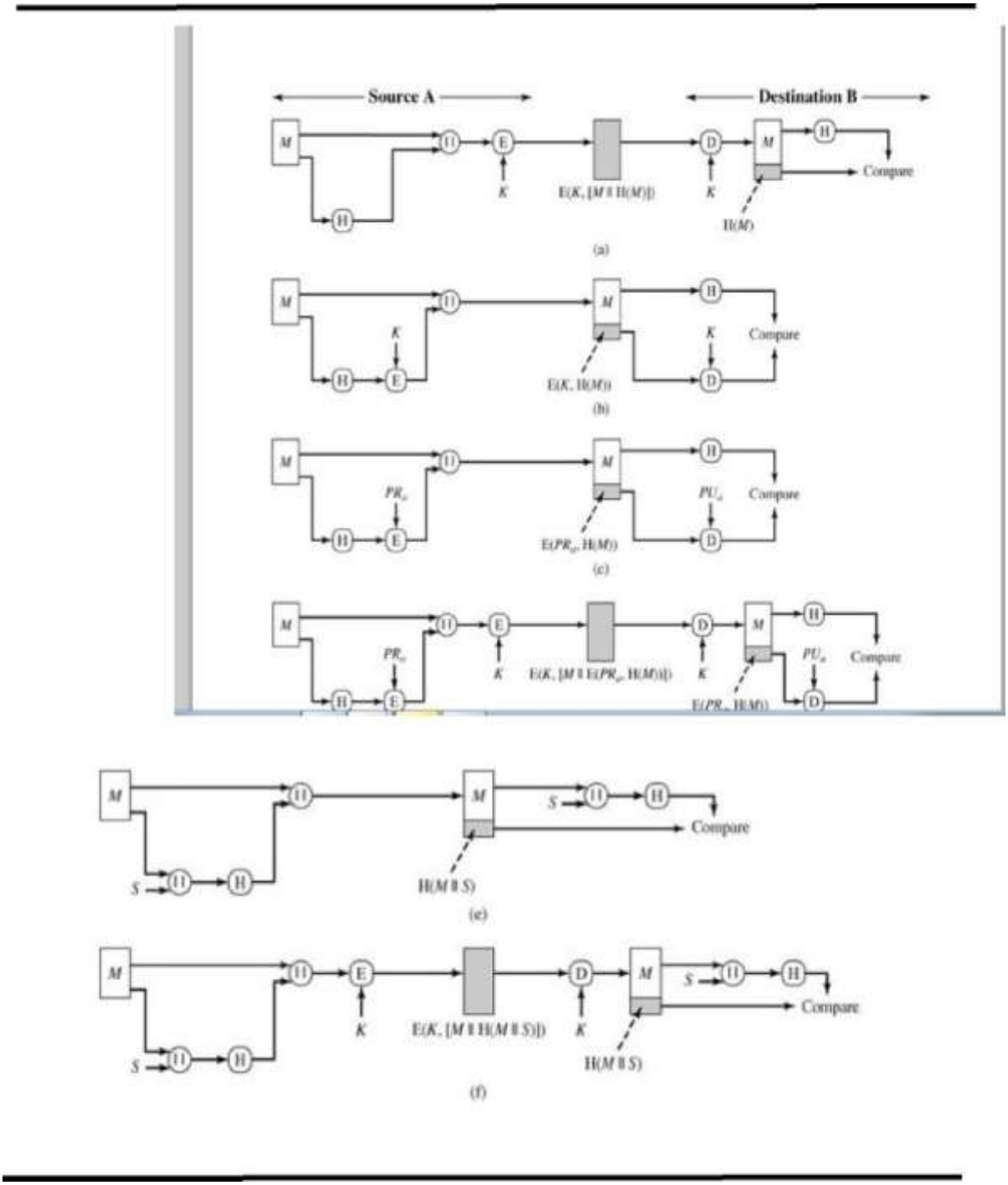


Fig .a. Encrypt message plus hash code

A Sender:

- The sender creates a message using SHA to generate a 160 bit hashcode.
- The message and hashcode is concatenated and the result is encrypted using symmetric Encryption Algorithm.

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode
- If two hashcodes are match ,the message is accepted,else it is rejected.

B Sender:

- The sender creates a message using SHA to generate a 160 bit hashcode.
- Only the hash code is encrypted using Symmetric Encryption.
- The message and hash code encrypted and result is concatented

A Receiver:

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode
- If two hashcodes are match ,the message is accepted,else it is rejected.

C Sender:

- The sender creates a message using SHA to generate a 160 bit hashcode.
- Only the hash code is encrypted using public key encryption and using the sender's private key.
- The message and hash code encrypted and result is concatented

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode
- If two hashcodes are match ,the message is accepted,else it is rejected.

D Sender:

- The sender creates a message using SHA to generate a 160 bit hashcode.
- Only the hash code is encrypted using public key encryption and using the sender's private key.
- The message and hash code encrypted and result is concatented

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode which uses the public key Encryption Algorithm of public key of sender.
- If two hashcodes are match ,the message is accepted,else it is rejected.

E Sender:

- The sender creates a message M using SHA to generate a 160 bit hashcode.
- This technique uses a hash fuction,but no encryption for message authentication
- This technique assumes that the two communicating parties share a common secret value 'S'.
- The source computes the hash value over the concatenation of M and S and appends the resulting hashvalue to M.

B Receiver:

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode which uses the public key Encryption Algorithm of public key of sender.
- The Message concatenated with hash value and 'S' is compared with receiver 's hash value.
- If two hashcodes are match ,the message is accepted,else it is rejected.

Confidentiality can be added to the previous approach by encrypting the entire message plus the hash code.

Requirements for a Hash Function

1. H can be applied to a block of data of any size.

2. H produces a fixed-length output.

3. H(x) is relatively easy to compute for any given x, making both hardware and Software implementations practical.

4. For any* given value h, it is computationally infeasible to find x such that H(x) =h. This is sometimes referred to in the literature as the one-way property.

5. For any given block x, it is computationally infeasible to find y x such that H(y) = H(x). This is
sometimes referred to as weak hash function.

6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y). This is sometimes referred to as **strong collision resistance**.

- The first three properties are requirements for the practical application of a hash function to message authentication.
- The fourth property, the one-way property, states that it is easy to generate a code given a message but virtually impossible to generate a message given a code.
- The fifth property guarantees that an alternative message hashing to the same value as a given message cannot be found.
- This prevents forgery when an encrypted hash code is used (Figures b and c).
- The sixth property refers to how resistant the hash function is to a type of attack known as the birthday attack, which we examine shortly.

Message Encryption

Message encryption by itself can provide a measure of authentication.
The analysis differs for

symmetric and public-key encryption schemes.

Symmetric Encryption

Consider the straightforward use of symmetric encryption (Figure 12.1a). A message M transmitted from source A to destination B is encrypted using a secret key K shared by

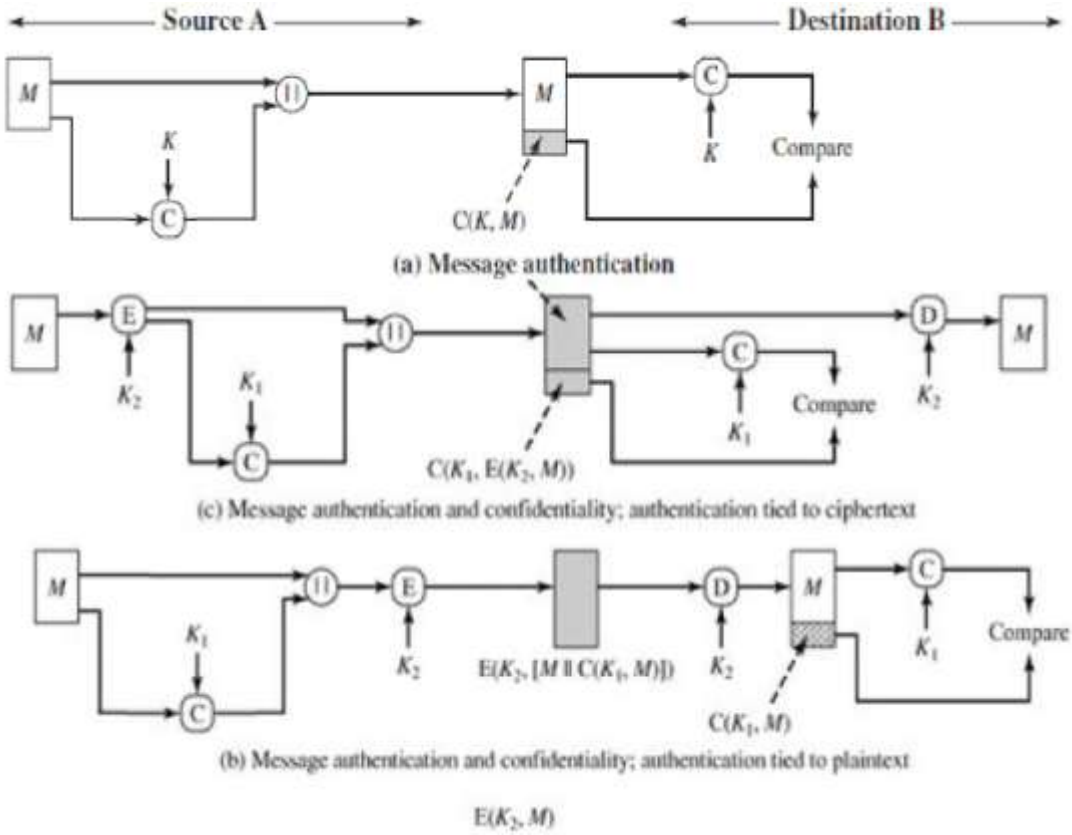
A and B. If no other party knows the key, then confidentiality is provided: No other party can recover the plaintext of the message.
n addition, B is assured that the message was generated by A. Why? The message must have come from A, because A is the only other party that possesses K and therefore the

Public-Key Encryption

The straightforward use of public-key encryption (Figure 12.1b) provides confidentiality but not authentication. The source (A) uses the public key PU_b of the destination (B) to encrypt M. Because only B has the corresponding private key PR_b, only B can decrypt the message. This scheme provides no authentication, because any opponent could also use B's public key to encrypt a message and claim to be A.

To provide authentication, A uses its private key to encrypt the message, and B uses A's public key to decrypt (Figure 12.1c). This provides authentication using the same type of reasoning as in the symmetric encryption case: The message must have come from A because A is the only party that possesses PR_a and therefore the only party with the information necessary to construct ciphertext that can be decrypted with PU_a.

Again, the same reasoning as before applies: There must be some internal structure to the plaintext so that the receiver can distinguish between well-formed plaintext and random bits.



3. Explain the Security of hash function and MAC. [CO3-L2]

Security of hash function and MAC

We can group attacks on MACs into two categories: brute-force attacks and cryptanalysis.

Brute-Force Attacks

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. Let us see why this is so. To attack a hash code, we can proceed in the following way.

The strength of a hash function against brute-force attacks depends solely on the length of the hash code produced by the algorithm. Recall from our discussion of hash functions that there are three desirable properties:

One-way: For any given code *h*, it is computationally infeasible to find *x* such that H(*x*) = *h*.
Weak collision resistance: For any given block *x*, it is computationally infeasible to find *y x* with H(*y*) = H(*x*).
Strong collision resistance: It is computationally infeasible to find any pair (*x*, *y*) such that H (*x*) = H(*y*).

Message Authentication Codes

A brute-force attack on a MAC is a more difficult undertaking because it requires known message-MAC pairs. Let us see why this is so.
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 MAC.
- To proceed, we need to state the desired security property of a MAC algorithm, which can be expressed as follows:
- Given one or more text-MAC pairs [*xi*, C(*K*, *xi*)], it is computationally infeasible to compute any text-MAC pair [*x*, C(*K*, *x*)] for any new input *x*≠ *xi*.
- The attacker would like to come up with the valid MAC code for a given message *x*.
- There are two lines of attack possible: Attack the key space and attack the MAC value
- If an attacker can determine the MAC key, then it is possible to generate a valid MAC value for any input *x*.

- Suppose the key size is *k* bits and that the attacker has one known text-MAC pair. Then the attacker can compute the *n*-bit MAC on the known text for all possible keys. At least one key is guaranteed to produce the correct MAC, namely, the valid key that was initially used to produce the known text-MAC pair. This phase of the attack takes a level of effort proportional to 2^k (that is, one operation for each of the 2^k possible key values).
- It can be shown that the level of effort drops off rapidly with each additional text-MAC pair and that the overall level of effort is roughly 2^k
- To summarize, the level of effort for brute-force attack on a MAC algorithm can be expressed as $\min(2^k, 2^n)$

Cryptanalysis

- The way to measure the resistance of a MAC algorithm to cryptanalysis is to compare its strength to the effort required for a bruteforce 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.

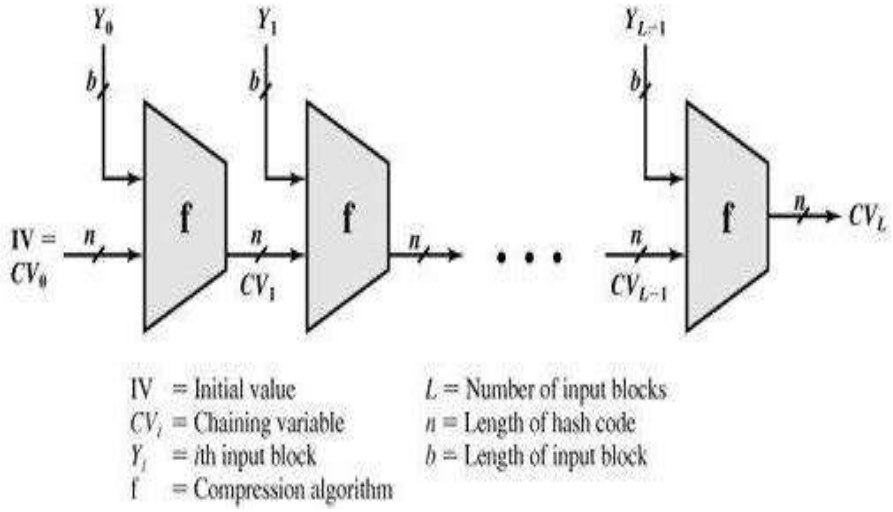


Figure 11.9. General Structure of Secure Hash Code

- The hash algorithm involves repeated use of a **compression function**, f, that takes two inputs (an *n*-bit input from the previous step, called the ***chaining variable***, and a *b*-bit block) and produces an *n*-bit output.
- At the start of hashing, the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value. Often, $b > n$; hence the term
- ***Compression***. The hash function can be summarized as follows:

4. Explain the steps in MD5 (Message Digest Algorithm) or Write the algorithm of MD5 arid explains? [CO3-L2-Nov/Dec 2013-May/Jun 2010-May/Jun 2012-DEC15]

MD5 DIGEST ALGORITHM

The MDS message-digest algorithm (RFC 1321) was developed by Ron Rivest at MIT (the "R" in the RSA [Rivest-Shamir-Adleman] public key encryption algorithm)- Until the last few years, when both bruteforce and cryptanalytic concerns have arisen, MD5 was the most widely used secure hash algorithm.

MD5 Logic

The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest- The input is processed in 512-bit blocks .Figure 9-1 depicts the overall processing of a message to produce a digest. This follows the general structure depicted in Figure 8-10. The processing consists of the following steps:

Step 1: Append padding bits- The message is padded so that its length in bits is congruent to 448 modulo 512 (length ≡ 448 mod 512). That is, the length of the padded message is 64 bits less than an integer multiple of 512 bits.

Padding is always added, even if the message is already of the desired length- For example, if the message is 448 bits long, it is padded by 512 bits to a length of 960 bits-Thus, the number of padding bits is in the range of 1 to 512- The padding consists of a single 1-bit followed by the necessary number of 0 bits.

Step 2: Append length- A 64-bit representation of the length in bits of the original message (before the padding) is appended to the result of step 1 (least significant byte,first). If the original .length is greater than 2^{64}, then only the low-order 64 bits of the length are used. Thus, the field contains the length of the original message, modulo 2^{64}.

The outcome of the first two steps yields a message that is an integer multiple of 512 bits in length- In Figure 9.1, the expanded message is represented as the sequence of 512-bit blocks $Y_0, Y_1, \ldots, Y_{L-1}$,, so that the total length of the expanded message is L × 512 bits-Equivalently, the result is a multiple of 16 32-bit words- Let M[0…..N - 1] denote the words of the resulting message, with N an integer multiple of 16- Thus, N = L × 16.

Step 3: Initialize MD buffer- A 128 bit buffer is used to hold intermediate and final results of the hash function- The buffer can be represented as four 32 bit registers (A, B, C, D)- These registers are, initialized to the following 32-bit integers (hexadecimal values):

These values are stored in little-endian format, which is the least significant byte of a word in the low address byte position- As 32 bit strings, the initialization values (in hexadecimal) appear as follows:

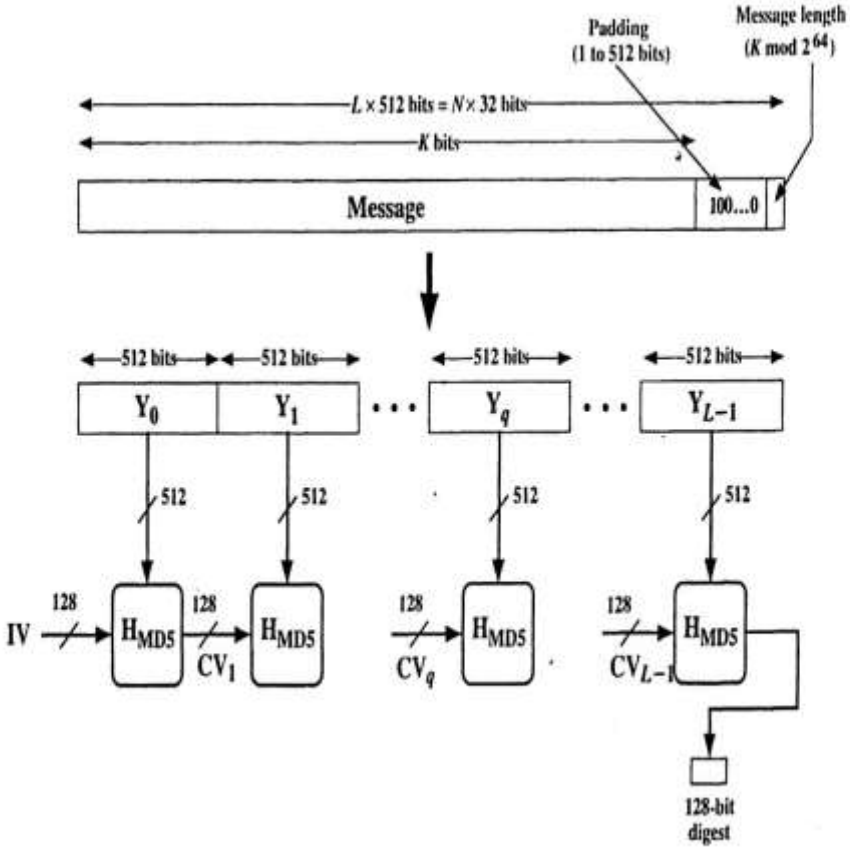


Figure 9.1 Message Digest Generation Using MD5.

Step 4: Process message in 512-bit (16-word) blocks- The heart of the algorithm is a compression function that consists of four “rounds” of processing; this module is labeled H_{MD5} in Figure 9-1, and its logic is illustrated in Figure 9-2- The four rounds have a similar structure, but each uses a different primitive logical function, referred to as F, G, H, and I in the specification.

Step 5: Output- After all L 512 bit blocks have been processed, the output from the L *th* stage is the 128-bit message digest.

We can summarize the behavior of MDS as follows:

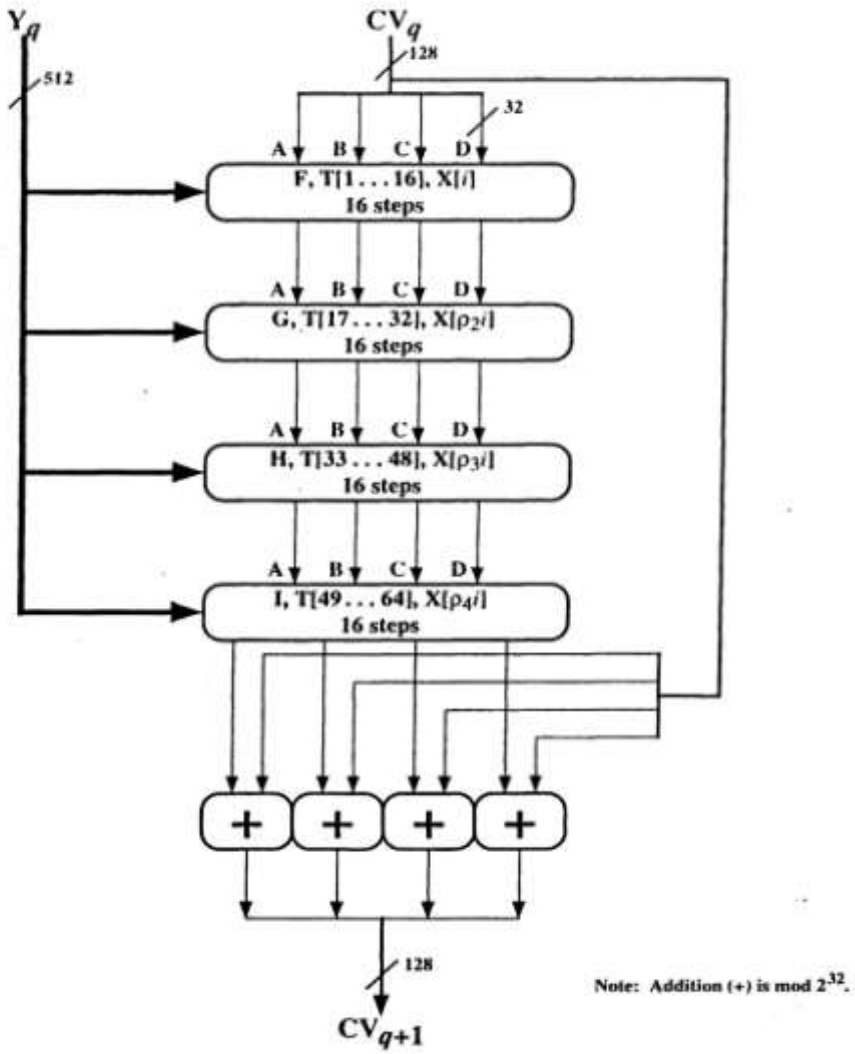


Figure 9.2 MD5 Processing of a Single 512-bit Block (MD5 compression function).

MDS Compression Function

Let us look in more detail at the logic in each of the four rounds of the processing of one 512-bit block- Each round consists of a sequence of 16 steps operating on the buffer-ABCD-Each step is of the form

- Figure 9-3 illustrates the step operation- The order in which the four words (a, b, c, d) are used produces a word level circular right shift of one word for each step- One of the four primitive logical functions is used for each of the four rounds of the algorithm- Each primitive function takes three 32 bit words as input and produces a 32-bit word output

- Each function performs a set of bitwise logical operations; that is, the nth bit of the output is a function of the nth bit of the three inputs

The functions can be summarized as follows:

Round	Primitive function g	g(b, c, d)
1	F(b, c, d)	$(b \wedge c) \vee (b \wedge d)$
2	G(b, c, d)	$(b \wedge d) \vee (c \wedge d)$
3	H(b, c, d)	$b \oplus c \oplus d$
4	I(b, c, d)	$c \oplus (b \vee d)$

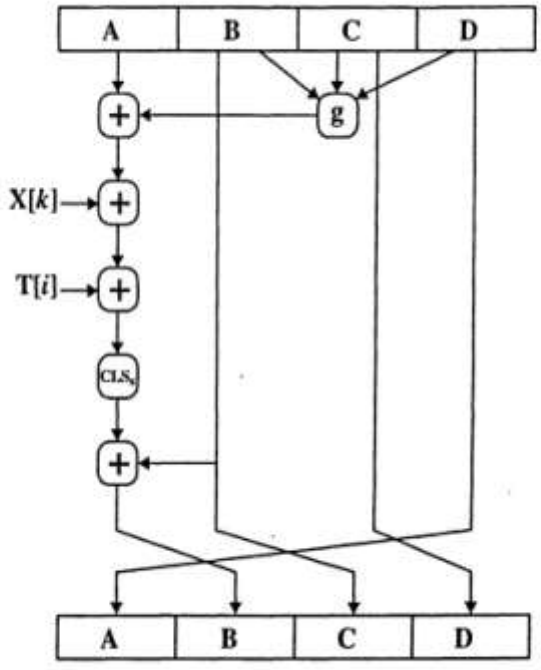


Figure 9.3 Elementary MD5 Operation (single step).

MD4

MD4 is a precusor to MD5 developed by the same designer, Ron Rivest. It was originally published as an RFC in October 1990. The following goals were listed:

- **Security:** There is the usual requirement for a hash code namely, that it be cornputationally infeasible to find two messages that have the same message digest
- **Speed:** The algorithm should lend itself to implementations in software that executes rapidly- In particular, the algorithm is intended to be fast on 32-bit architectures.
- **Simplicity and compactness:** The algorithm should be simple to describe and simple to program, without requiring large programs or substitution tables.
- **Favor little endian architecture:** Some processor architectures (such as the Intel 80xxx and Pentium line) store the least significant byte of a word in the low—address byte position (little endian).

5. Explain SHA (Secure Hash Algorithm) Or Explain the process of deriving eighty 64-bit words from the 1024 bits for processing of a single block and also discuss single round function in SHA-512algorithm. show the value of W_{16},W_{17},W_{18} and W_{19}. [CO3-L2- DEC 13]

Secure Hash Algorithm (SHA)

SHA was developed by the Nation-bits foral Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993. When weaknesses were discovered in SHA, now known as SHA-0, a revised version was issued as FIPS 180-1 in 1995 and is referred to as SHA-1.

Table 11.3 Comparison of SHA Parameters

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

The actual standards document is entitled “Secure Hash Standard.” SHA is based on the hash function MD4, and its design closely models MD4. SHA-1 produces a hash value of 160 bits.

NIST produced a revised version of the standard, FIPS 180-2, that defined three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512, respectively.

SHA-512 Logic

The algorithm takes as input a message with a maximum length of less than 2128 bits and produces as output a 512-bit message digest. The input is processed in 1024-bit blocks.

Figure 11.9 depicts the overall processing of a message to produce a digest. This follows the general structure depicted in Figure 11.8. The processing consists of the following steps.

Step 1 Append padding bits. The message is padded so that its length is congruent to 896 modulo 1024 [length K 896(mod 1024)]. Padding is always added, even if the message is already of the desired length. Thus, the number of padding bits is in the range of 1 to 1024. The padding consists of a single 1 bit followed by the necessary number of 0 bits.

Step 2 Append length. A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer (most significant byte first) and contains the length of the original message (before the padding).

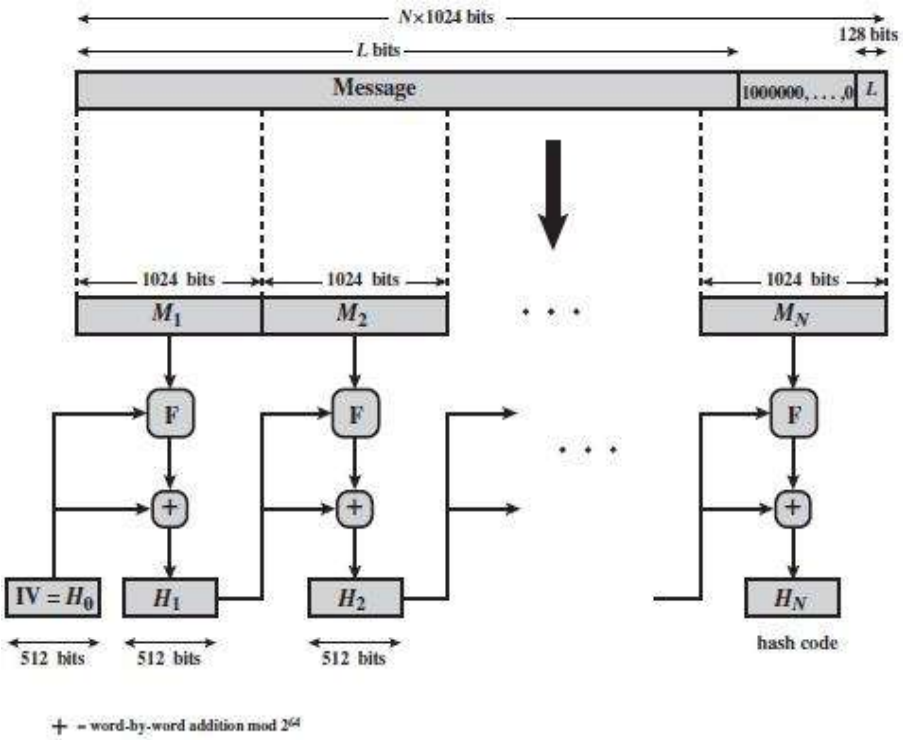


Figure 11.9 Message Digest Generation Using SHA-512

The outcome of the first two steps yields a message that is an integer multiple of 1024 bits in length. In Figure 11.9, the expanded message is represented as the sequence of 1024-bit blocks *M*1, *M*2, c, *MN*, so that the total length of the expanded message is *N* * 1024 bits.

Step 3 Initialize hash buffer. A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialized to the following 64-bit integers (hexadecimal values):

These values are stored in big-endian format, which is the most significant byte of a word in the low-address (leftmost) byte position. These words were obtained by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers.

Step 4 Process message in 1024-bit (128-word) blocks. The heart of the algorithm is a module that consists of 80 rounds; this module is labeled F in Figure 11.9. The logic is illustrated in Figure 11.10.

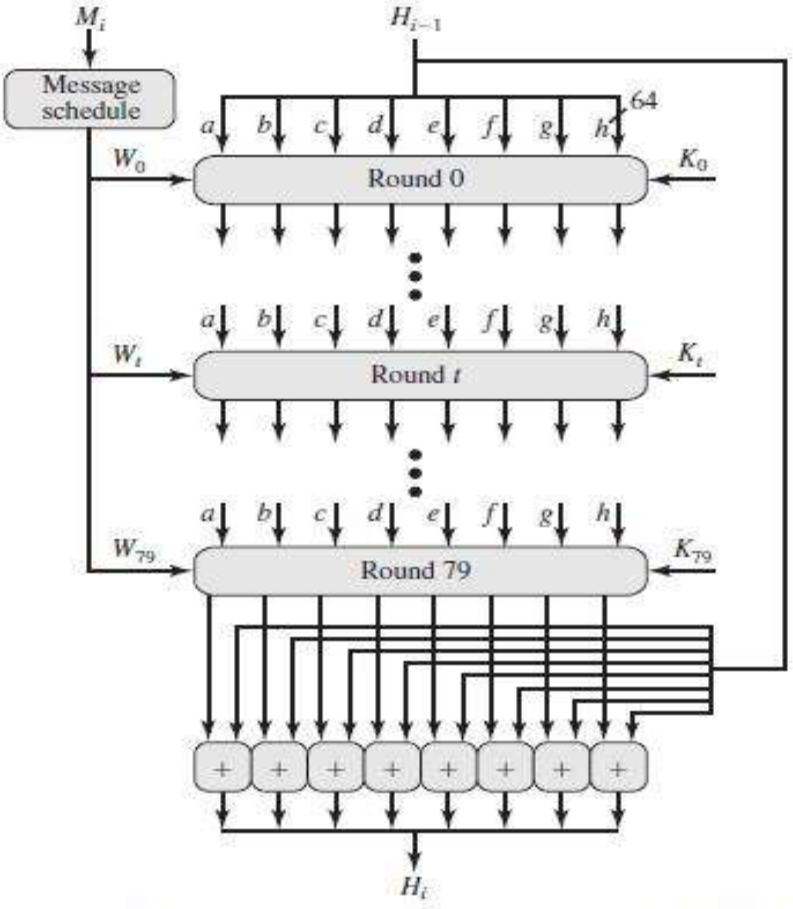


Figure 11.10 SHA-512 Processing of a Single 1024-Bit Block

Each round takes as input the 512-bit buffer value, abcdefgh, and updates the contents of the buffer. At input to the first round, the buffer has the value of the intermediate hash value, *Hi*-1.

6. Explain detail about the Hashed Message Authentication. [CO3-L2]

Hashed Message Authentication Codes(HMAC)

The IPsec authentication scheme uses a scheme called Message Authentication Codes (HMAC) , which is an encrypted message digest described in RFC 1024

HAMC uses a shared secret key between two parties than public key method for message authentication

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.

HMAC Design Objectives

RFC 2104 lists the following design objectives for HMAC.

- To use, without modifications, available hash functions.
- 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.

HMAC Algorithm

Figure 12.5 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)
Yi = *i* th block of M, 0 … *i* … (*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 Ú *n*; if key length is greater than *b*, the key is input to the hash function to produce an *n*-bit key
K+ = *K* padded with zeros on the left so that the result is *b* bits in length

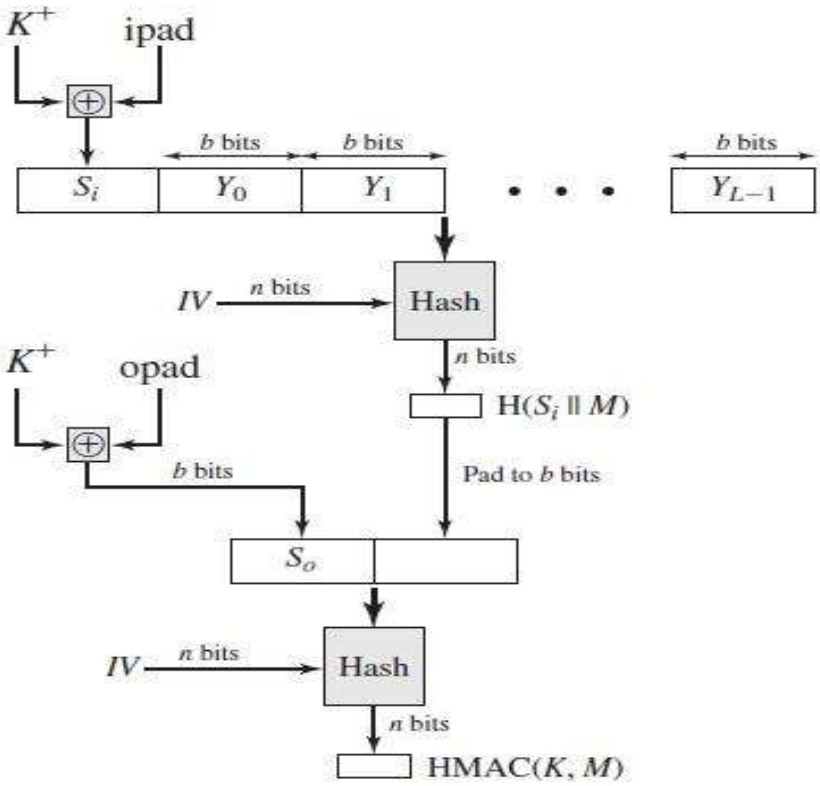


Figure 12.5 HMAC Structure

ipad = 00110110 (36 in hexadecimal) repeated *b*/8 times opad = 01011100 (5C in hexadecimal) repeated *b*/8 times

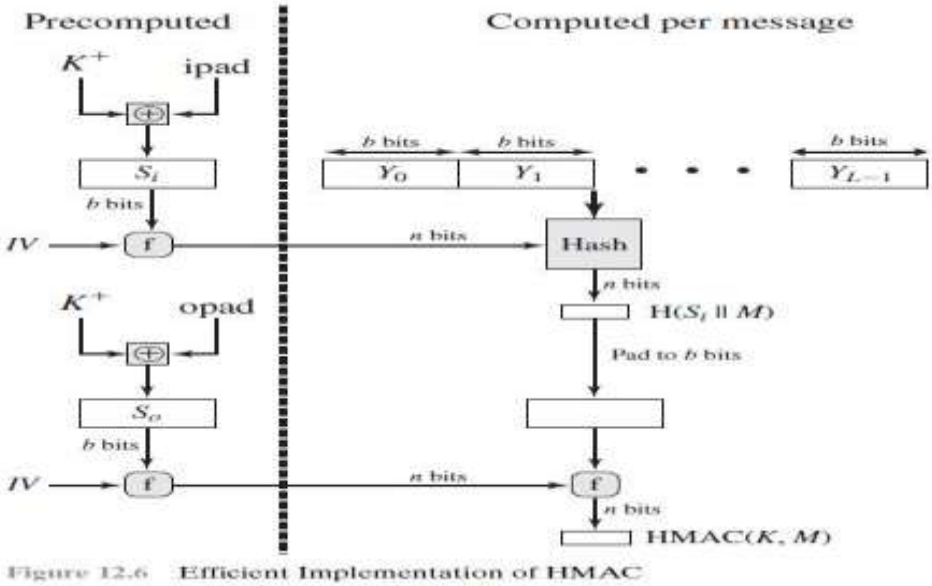


Figure 12.6 Efficient Implementation of HMAC

Security of HMAC

The security of any MAC function based on an embedded hash function depends in some way on the cryptographic strength of the underlying hash function. The appeal of HMAC is that its designers have been able to prove an exact relationship between the strength of the embedded hash function and the strength of HMAC.

1.The attacker is able to compute an output of the compression function even with an *IV* that is random, secret, and unknown to the attacker.

2.The attacker finds collisions in the hash function even when the *IV* is random and secret.

In the first attack, we can view the compression function as equivalent to the hash function applied to a message consisting of a single b-bit block.

In the second attack, the attacker is looking for two messages M and M' that produce the same hash: H(M) = H(M'). This is the birthday attack.

7. Explain the Cipher-based Message Authentication Code(CMAC). [CO3-L2]

Cipher-based Message Authentication Code(CMAC)

This MAC is secure under a reasonable set of security criteria, with the following restriction. Only messages of one fixed length of *m n* bits are processed, where n is the cipher block size and m is a fixed positive integer.

As a simple example, notice that given the CBC MAC of a one-block message X, say $T = MAC(K, X)$, the adversary immediately knows the CBC MAC for the two block message X **||** (X⊕T) since this is once again T.

This proposed construction was refined so that the two n-bit keys could be derived from the encryption key, rather than being provided separately. This refinement, adopted by NIST, is the ***Cipher-based Message Authentication Code (CMAC)*** mode of operation for use with AES and triple DES.

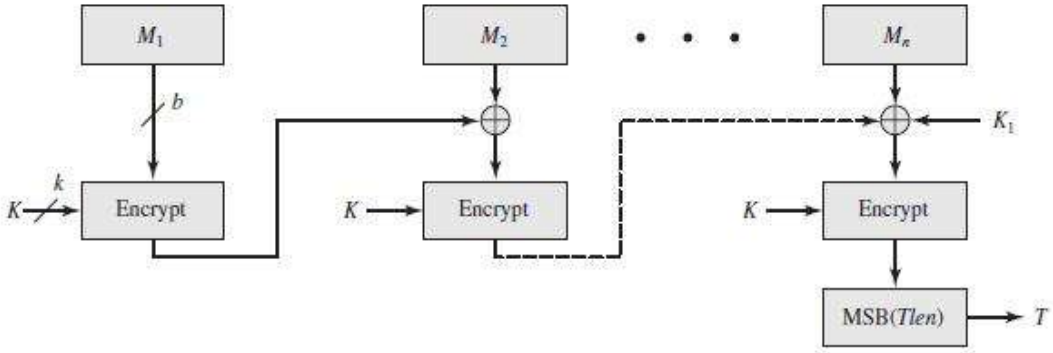
First, let us define the operation of CMAC when the message is an integer multiple n of the cipher block length b. For AES, b = 128, and for triple DES, b = 64. The message is divided into n blocks ($M_1, M_2, ..., M_n$). The algorithm makes use of a k-bit encryption key K and a b-bit constant, K_1. 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

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.

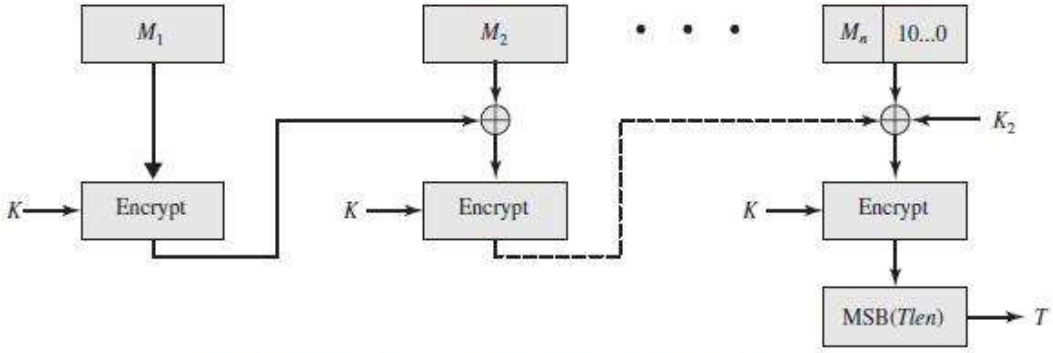
The CMAC operation then proceeds as before, except that a different b-bit key K_2 is used instead of K_1.

The two b-bit keys are derived from the k-bit encryption key as follows.

where multiplication (·) is done in the finite field GF(2^b) and x and x^2 are first and second-order polynomials that are elements of GF(2^b). Thus, the binary representation of x consists of b - 2 zeros followed by 10; the binary representation of x^2 consists of b - 3 zeros followed by 100.



(a) Message length is integer multiple of block size



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

Figure 12.8 Cipher-Based Message Authentication Code (CMAC)

8. Explain in detail about the Digital signature . [CO3-L2-May-14]

Digital Signatures

A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creators private key

Proerties

- 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.
- For example, suppose that John sends an authenticated message to Mary, using one of the schemes of Figure 12.1. Consider the following disputes that could 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.

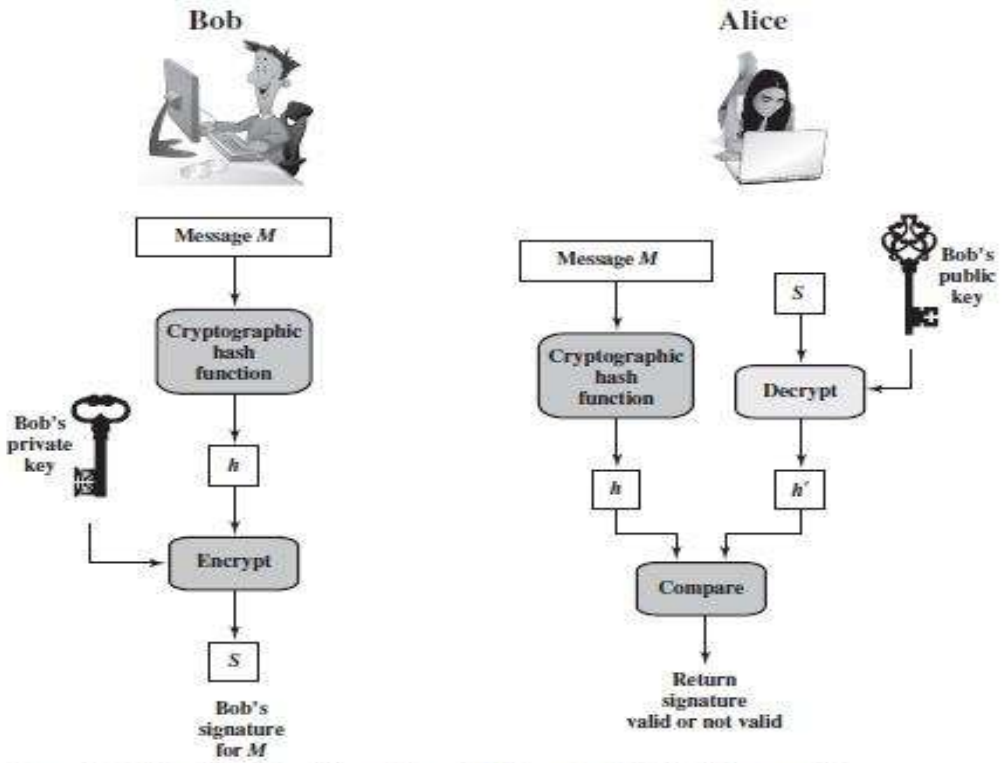


Figure 13.2 Simplified Depiction of Essential Elements of Digital Signature Process

Attacks and Forgeries

The following types of attacks, in order of increasing severity. Here A denotes the user whose signature method is being attacked, and C denotes the attacker.

- **Key-only attack:** C only knows A's public key.
- **Known message attack:** C is given access to a set of messages and their signatures.
- **Generic chosen message attack:** C chooses a list of messages before attempting to breaks A's signature scheme, independent of A's public key. C then obtains from A valid signatures for the chosen messages.
- **Directed chosen message attack:** Similar to the generic attack, except that the list of messages to be signed is chosen after C knows A's public key but before any signatures are seen.
- **Adaptive chosen message attack:** C is allowed to use A as an "oracle." This means that C may request from A signatures of messages that depend on previously obtained message-signature pairs.

[GOLD88] then defines success at breaking a signature scheme as an outcome in which C can do any of the following with a non-negligible probability:

- **Total break:** C determines A's private key.
- **Universal forgery:** C finds an efficient signing algorithm that provides an equivalent way of constructing signatures on arbitrary messages.
- **Selective forgery:** C forges a signature for a particular message chosen by C.
- **Existential forgery:** C forges a signature for at least one message. C has no control over the message. Consequently, this forgery may only be a minor nuisance to A.

On the basis of the properties and attacks just discussed, we can formulate the following 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.

Two general schemes for digital signatures

Direct Digital Signature
Arbitrated Digital Signature

Direct Digital Signature

- The term **direct digital signature** refers to a digital signature scheme that involves only the communicating parties (source, destination). It is assumed that the destination knows the public key of the source.
- Confidentiality can be provided by encrypting the entire message plus signature with a shared secret key (symmetric encryption). Note that it is important to perform the signature function first and then an outer confidentiality function.
- In case of dispute, some third party must view the message and its signature. If the signature is calculated on an encrypted message, then the third party also needs access to the decryption key to read the original message. However, if the signature is the inner operation, then the recipient can store the plaintext message and its signature for later use in dispute resolution.
- The validity of the scheme just described depends on the security of the sender's private key. If a sender later wishes to deny sending a particular message, the sender can claim that the private key was lost or stolen and that someone else forged his or her signature.
- Administrative controls relating to the security of private keys can be employed to thwart or at least weaken this ploy, but the threat is still there, at least to some degree. One example is to require every signed message to include a **timestamp** (date and time) and to require prompt reporting of compromised keys to a central authority.
- Another threat is that some private key might actually be stolen from X at time T. The opponent can then send a message signed with X's signature and stamped with a time before or equal to T.

Arbitrated Digital Signature

- The problems associated with direct digital signatures can be addressed by using an arbiter.
- As with direct signature schemes, there is a variety of arbitrated signature schemes.

- Every signed message from a sender X to a receiver Y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check its origin and content.
- The message is then dated and sent to Y with an indication that it has been verified to the satisfaction of the arbiter. The presence of A solves the problem faced by direct signature schemes: that X might disown the message.
- The arbiter plays a sensitive and crucial role in this sort of scheme, and all parties must have a great deal of trust that the arbitration mechanism is working properly.

- In the first, symmetric encryption is used. It is assumed that the sender X and the arbiter A share a secret key *Kxa* and that A and Y share secret key *Kay*. X constructs a message *M* and computes its hash value H(*M*).

  - Then X transmits the message plus a signature to A. The signature consists of an identifier *IDX* of X plus the hash value, all encrypted using *Kxa*. A decrypts the signature and checks the hash value to validate the message.
  - Then A transmits a message to Y, encrypted with *Kay*. The message includes *IDX*, the original message from X, the signature, and a timestamp. Y can decrypt this to recover the message and the signature. The timestamp informs Y that this message is timely and not a replay. Y can store *M* and the signature. In case of dispute, Y, who claims to have received *M* from X, sends the following message to A:

  - The arbiter uses *Kay* to recover *IDX*, *M*, and the signature, and then uses *Kxa* to decrypt the signature and verify the hash code. In this scheme, Y cannot directly check X's signature; the signature is there solely to settle disputes. Y considers the message from X authentic because it comes through A.

It must trust A to send E(*Kay*, [*IDX*||*M*||E(*Kxa*, [*IDX*||H(*M*)])||*T*]) only if the hash value is correct and the signature was generated by X.

9. Explain DSS (Digital Signature Standard) [CO3-L2- May-14] or Digital Signature algorithm

Digital Signatures Standard

The DSS makes use of the Secure Hash Algorithm (SHA) presents a new digital signature technique, the **Digital Signature Algorithm (DSA)**.

The DSS Approach

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.

Figure 13.3 contrasts the DSS approach for generating digital signatures to that used with RSA. In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length. This hash code is then encrypted using the sender’s private key to form the signature.

The DSS approach also makes use of a hash function. The hash code is provided as input to a signature function along with a random number generated for this particular signature.The signature function also depends on the sender’s private key and a set of

parameters known to a group of communicating principals. We can consider this set to constitute a global public key .1 The result is a signature consisting of two components, labeled *s* and *r*.

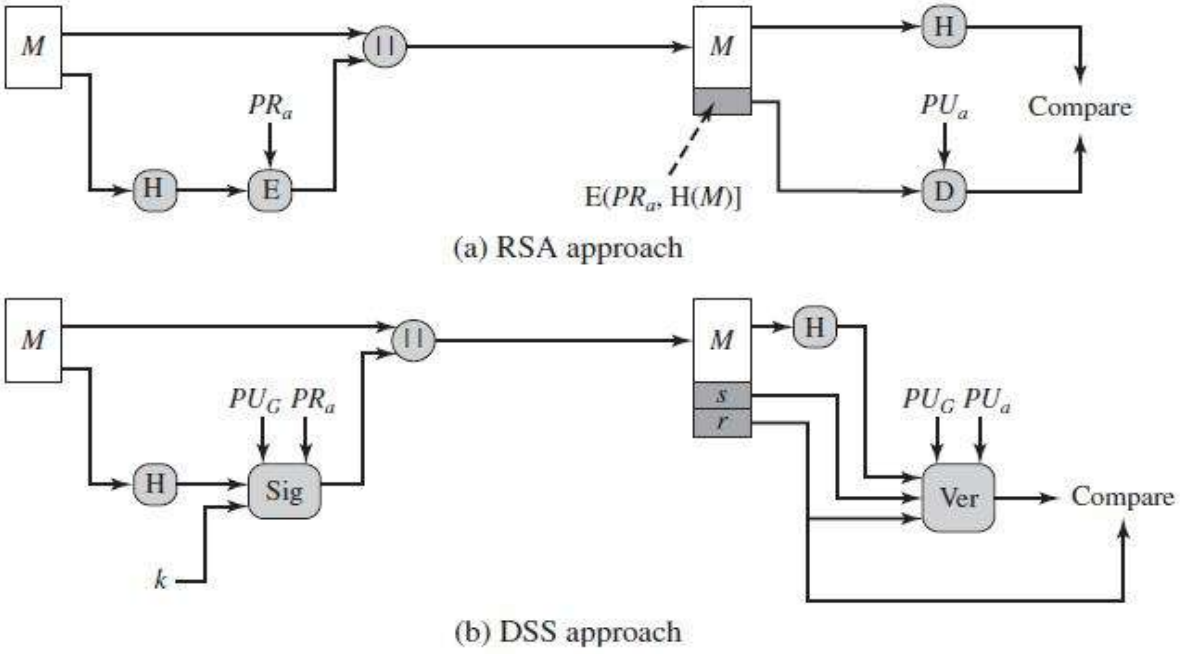


Figure 13.3 Two Approaches to Digital Signatures

At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function. The verification function also depends on the global public key as well as the sender's public key , which is paired with the sender's private key.

The output of the verification function is a value that is equal to the signature component if the signature is valid. The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature. We turn now to the details of the algorithm.

The Digital Signature Algorithm

Figure 13.4 summarizes the algorithm. There are three parameters that are public and can be common to a group of users. A 160-bit prime number is chosen. Next, a prime number is selected with a length between 512 and 1024 bits such that divides (*p* - 1).

Finally, *g* is chosen to be of the form *h*(*p*-1)/*q*mod *p*, where *h* is an*p q q r* (*PUa*) 2In number-theoretic terms, g is of order q modp; see Chapter 8. integer between 1 and with the restriction that must be greater than 1.2

The multiplicative inverse of is passed to a function that also has as inputs the message hash code and the user's private key. The structure of this function is such that the receiver can recover using the incoming message and signature, the public key of the user, and the global public key. It is certainly not obvious from Figure 13.4 or Figure 13.5 that such a scheme would work. Because this value does not depend on the message to be signed, it can be computed ahead of time.

Indeed, a user could precalculate a number of values of to be used to sign documents as needed. The only other somewhat demanding task is the determination of a multiplicative inverse, Again, a number of these values can be precalculated.

10. Explain the Authentication Protocols. [CO3-L2]

Authentication Protocols

Authentication Protocols are used to convince parties of each others identity and to exchange session keys.they may be **(mutual authentication and one-way authentication)**

Mutual Authentication

Symmetric Encryption Approaches
Public-Key Encryption Approaches

An important application area is that of mutual authentication protocols. Such protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys.

Central to the problem of authenticated key exchange are two issues: **confidentiality and timeliness.**

To prevent masquerade and to prevent compromise of session keys, essential identification and session key information must be communicated in encrypted form.

This requires the prior existence of secret or public keys that can be used for this purpose. The second issue, timeliness, is important because of the threat of message replays. Such replays, at worst, could allow an opponent to compromise a session key or successfully impersonate another party.

At minimum, a successful replay can disrupt operations by presenting parties with messages that appear genuine but are not.

Simple replay: The opponent simply copies a message and replays it later.
Repetition that can be logged: An opponent can replay a timestamped message within the valid time window.
Repetition that cannot be detected: This situation could arise because the original message could have been suppressed and thus did not arrive at its destination; only
the replay message arrives.
Backward replay without modification: This is a replay back to the message sender. This attack is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and messages received on the basis of content.

The following two general approaches is used:

Timestamps: Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time.
This approach requires that clocks among the various participants be synchronized.
Challenge/response: Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value. It can be argued.

A two-level hierarchy of symmetric encryption keys can be used to provide confidentiality for communication in a distributed environment. In general, this strategy involves the use of a trusted key distribution center (KDC).

Each party in the network shares a secret key, known as a master key, with the KDC. The KDC is responsible for generating keys to be used for a short time over a connection between two parties, known as session keys, and for distributing those keys using the master keys to protect the distribution.

In step 1.Secret keys *Ka* and *Kb* are shared between A and the KDC and B and the KDC, respectively. The purpose of the protocol is to distribute securely a session key *Ks* to A and B. A securely acquires a new session key in step 2.

The message in step 3 can be decrypted, and hence understood, only by B. Step 4 reflects B's knowledge of *Ks*, and step 5 assures B of A's knowledge of *Ks* and assures B that this is a fresh message because of the use of the nonce *N*2 that the purpose of steps 4 and 5 is to prevent a certain type of replay attack.

One way to counter suppress-replay attacks is to enforce the requirement that parties regularly check their clocks against the KDC's clock. The other alternative, which avoids the need for clock synchronization, is to rely on handshaking protocols using nonces.

This latter alternative is not vulnerable to a suppress-replay attack because the nonces the recipient will choose in the future are unpredictable to the sender. The Needham/Schroeder protocol relies on nonces only but, as we have seen, has other vulnerabilities.

Let us follow this exchange step by step.

1. A initiates the authentication exchange by generating a nonce, *Na*, and sending that plus its identifier to B in plaintext. This nonce will be returned to A in an encrypted message that includes the session key, assuring A of its timeliness.

2. B alerts the KDC that a session key is needed. Its message to the KDC includes its identifier and a nonce, *Nb* This nonce will be returned to B in an encrypted message that includes the session key, assuring B of its timeliness. B's message to the KDC also includes a block encrypted with the secret key shared by B and the KDC. This block is used to instruct the KDC to issue credentials to A; the block specifies the intended recipient of the credentials, a suggested expiration time for the credentials, and the nonce received from A.

3. The KDC passes on to A B's nonce and a block encrypted with the secret key that B shares with the KDC. The block serves as a "ticket" that can be used by A for subsequent authentications, as will be seen. The KDC also sends to A block encrypted with the secret key shared by A and the KDC. This block verifies that B has received A's initial message (*IDB*) and that this is a timely message and not a replay (*Na*) and it provides A with a session key (*Ks*) and the time limit on its use (*Tb*).

4. A transmits the ticket to B, together with the B's nonce, the latter encrypted with the session key. The ticket provides B with the secret key that is used to decrypt E(*Ks, Nb*) to recover the nonce. The fact that B's nonce is encrypted with the session key authenticates that the message came from A and is not a replay.

Public-Key Encryption Approaches

This protocol assumes that each of the two parties is in possession of the current public key of the other.

Unit – IV

Security Practice & System Security

Part – A

1. Define Kerberos. [C04-L1]

Kerberos is a centralized authentication server whose function is to authenticate users to servers and servers to users.

2. List out the requirements for Kerberos.[C04-L1-April/May2011-Apr/May 2010]

Secure
Reliable
Transparent
Scalable

3. Mention the limitations of version 4 of Kerberos. [C04-L1-Nov/Dec 2009]

a. Environmental shortcomings
    i. Encryption system dependence
    ii. Internet protocol dependence
    iii. Ticket lifetime
    iv. Inter realm authentication
b. Technical deficiencies
    i. double encryption
    ii. Propagating block chaining encryption
    iii. Session keys
    iv. Password attacks

4. What you mean by versioned certificate? [C04-L1]

Mostly used issue X.509 certificate with the product name" versioned digital id".

Each digital id contains owner's public key, owner's name and serial number of the digital id.

5. What is mean by SET? What are the features of SET? [C04-L1-Nov/Dec 2013]

Secure Electronic Transaction (SET) is an open encryption and security specification designed to protect credit card transaction on the Internet.

Features are:

1. Confidentiality of information
2. Integrity of data
3. Cardholder account authentication
4. Merchant authentication

6. What are the steps involved in SET Transaction? [C04-L1]

1. The customer opens an account
2. The customer receives a certificate
3. Merchants have their own certificate
4. The customer places an order.
5. The merchant is verified.
6. The order and payment are sent.
7. The merchant requests payment authorization.
8. The merchant confirms the order.
9. The merchant provides the goods or services.
10. The merchant requests payment.

7. In the content of Kerberos, what is realm? [C04-L3]

1. A full service Kerberos environment consisting of a Kerberos server, a no. of clients, no. of application server requires the following:
2. The Kerberos server must have user ID and hashed password of all participating users in its database.
3. The Kerberos server must share a secret key with each server. Such an environment is referred to as “Realm”.

8. What is the purpose of X.509 standard? [C04-L1]

X.509 defines framework for authentication services by the X.500 directory to its users.X.509 defines authentication protocols based on public key certificates

9. How the password files be protected? [C04-L1-May/Jun 2009]

One way encryption
Access control

10. Define firewall? [C04-L1]

Firewall is the in which protects the premises network from internet based attacks and to provide a single choke point where security and audit can be imposed.

11. What are the design goals of the firewall? [C04-L1]

All traffic from inside to outside, and vice versa, must pass through the firewall.

Only authorized traffic, as defined by the local security policy, will be allowed to pass.

It is immune to penetration.

12 List out the limitations of the firewall? [C04-L1]

It cannot protect against attacks that bypass the firewall.

The firewall does not protect against internal threats.

It cannot protect against the transfer of virus infected programs or files.

13 Classes of the intruders? [C04-H1-April/May 2011- Apr/May 2010]

Masquerader
Misfeasor
Clandestine user

14 What are the types of firewall? [C04-L1]

Packet filtering firewall
Application level gateway
Circuit level gateway

15 Define Basiton host? [C04-L1]

A Basiton host is a system identified by the firewall administrator as a critical strong point in the network security.

16 List out the firewall configurations? [C04-L1]

Screened host firewall, single homed bastion
Screened host firewall, dual homed bastion
Screened subnet firewall

17 Define the two rules for multi-level security? [C04-L1- Nov/Dec 2011]

No read up: A subject can only read an object of less or equal security level. This is referred to as **simple security property.**
No write down: A subject can only write into an object of greater or equal security level. This is referred to as ***-property (star property).** These two rules, if properly enforced, provide multilevel security.

18. Define Trojan horse attack? [C04-L1-April/May 2011]

The Torjan horse attack begins with a hostile user, named X, gain legitimate access into the system and installs both the torjan horse program and a private file to be used in the attack as a ‘back packet’.

X gives read / write permission to itself and gives Y (authorized user) write-only permission. X now indicates Y to invoke torjan horse program, by advertising it as a useful utility.
When the program detects that it is being executed by Y, it reads the sensitive character string from Y’s file and copies it into X’s back pocket file.

19. What is Access Control? [C04-L1]

i. One way to thwart a password attack is to deny the opponent access to the password file.

ii. If the encrypted password portion of the file is accessible only by a privileged user, then the opponent cannot read it without already knowing the password of a privileged user.

20. What are the Four Password Selection Strategies? [C04-L1]

User education

Computer-generated passwords

Reactive password checking

Proactive password checking

21. Describe Problems in Computer-generated passwords? [C04-L3]

Computer-generated passwords also have problems.

If the passwords are quite random in nature, users will not be able to remember them.

Even if the password is pronounceable, the user may have difficulty remembering it and so be tempted to write it down

22. What are the Firewall characteristics? [C04-L1]

All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible.

Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies.

The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system. This implies that use of a trusted system with a secure operating system.

23. What is a Trusted system? [C04-L1]

One way to enhance the ability of a system to defend against intruders and malicious programs is to implement trusted system technology.

24. What is Subject, object and Access right? [C04-L1]

Subject: An entity capable of accessing objects. Generally, the concept of subject equates with that of process.

Object: Anything to which access is controlled. Examples include files, portion of files, programs, and segments of memory.

Access right: The way in which the object is accessed by a subject. Examples are read, write and execute.

25. What is Virus Structure? [C04-L1]

A virus can be prepended or postpended to an executable program, or it can be embedded in some other fashion.

The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.

26. What are the Types of Viruses? [C04-L1]

The following categories as being among the most significant types of viruses:

Parasitic virus: The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect.

Memory-resident virus: Lodges in main memory as part of a resident system program.From that point on, the virus infects every program that executes.

Boot sector virus: Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.

Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software.

Polymorphic virus: A virus that mutates with every infection, making detection by the "signature" of the virus impossible.

Metamorphic virus: As with a polymorphic virus, a metamorphic virus mutates with every infection.

27. What is E-mail Viruses? [C04-L1]

A more recent development in malicious software is the e-mail virus.

The first rapidly spreading e-mail viruses, such as Melissa, made use of a Microsoft Word macro embedded in an attachment.

If the recipient opens the e-mail attachment, the Word macro is activated. Then

1. The e-mail virus sends itself to everyone on the mailing list in the user's e-mail pack
2. The virus does local damage.

28. What are Worms? [C04-L1-Nov/Dec 2013]

A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again.

29. Describe Virus Counter Measure? [C04-L3]

The ideal solution to the threat of viruses is prevention: The next best approach is to be able to do the following:

Detection: Once the infection has occurred, determine that it has occurred and locate the virus.

Identification: Once detection has been achieved, identify the specific virus that has infected a program.

Removal: Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that the disease cannot spread further.

30. What are the four generations of antivirus software? [C04-L1]

1. First generation: simple scanners
2. Second generation: heuristic scanners
3. Third generation: activity traps
4. Fourth generation: full-featured protection

31. What is Intruder? [C04-L1-Nov/Dec 2012]

One of the most publicized attacks to security is the intruder, generally referred to as hacker or cracker. Three classes of intruders are as follows:

a. Masquerader
b. Misfeasor
c. Clandestine user

32. What is Audit Records? [C04-L1]

A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Basically, two plans are used:

i. **Native audit records**: Virtually all multiuser operating systems include accounting software that collects information on user activity.

ii. **Detection-specific audit records**: A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems.

33. Lists the following approaches can be performed to determine whether current activity fits within acceptable limits? [C04-L1]

Mean and standard deviation
Multivariate
Markov process
Time series
Operational

34. Give the Examples of metrics that are useful for profile-based intrusion detection? [C04-H1]

Counter
Gauge
Interval timer
Resource utilization

35. What is Honey pots? [C04-L1-Apr/May 2010]

A relatively recent innovation in intrusion detection technology is the honeypot. Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems. Honeypots are designed to

1. divert an attacker from accessing critical systems
2. collect information about the attacker's activity
3. encourage the attacker to stay on the system long enough for administrators to respond.

36. List few examples of worms? [C04-L1- Nov/Dec 2012]

Electronic mail facility: A worm mails a copy of itself to other systems.

Remote execution capability: A worm executes a copy of itself on another system.

Remote login capability: A worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other.

37. Define Intrusion? [C04-L1- May/Jun 2012]

Any attempt to compromise the integrity, confidentiality and availability of information and/or resources. There are two types of intrusion, Technical and non Technical. Technical intrusion involves using technical tools and expertise to perform the intrusion. Non technical (also known as social) intrusion involves using any non-technical means to perform the intrusion.

38. Mention the two levels of hackers? [C04-L3-May/Jun 2013]

The Hacking Group
Hacktivists

39. What is logic bomb? [C04-L1-May/Jun 2013]

A **logic bomb** is a piece of code intentionally inserted into a software system that will set off a malicious function when specified conditions are met. For example, a programmer may hide a piece of code that starts deleting files (such as a salary database trigger), should they ever be terminated from the company.

40. What are ZOMBIES? [C04-L1-MAY/JUN 2014]

Program activated on an infected machine that is activated to launch attacks on other machines. It is a program that takes secretly takes over another internet-attached computer and then uses that computer to launch that are difficult to trace

41. Difference between macro virus and boot virus? [C04-L3-NOV/DEC 2014]

Boot Sector

Boot sector viruses infect the system area of a disk--that is, the boot record on floppy disks and hard disks.

All floppy disks and hard disks (including disks containing only data) contain a small program in the boot record that is run when the computer starts up.

Boot sector viruses attach themselves to this part of the disk and activate when the user attempts to start up from the infected disk.

Macro viruses

These types of viruses infect data files.

They are the most common and have cost corporations the most money and time trying to repair.

With the advent of Visual Basic in Microsoft's Office 97, a macro virus can be written that not only infects data files, but also can infect other files as well. Macro viruses infect Microsoft Office Word, Excel, PowerPoint and Access files.

42. Difference between Spyware and virus? [C04-L1- MAY/JUN 2014]

virus

This is a term that used to be generic.
Any bad software used to be a virus; however, we use the term “malware” now.

We use the word “virus” to describe a program that self-replicates after hooking itself onto something running in Windows

Spyware

Software that monitors your computer and reveals collected information to an interested party.

This can be benign when it tracks what webpages you visit; or it can be incredibly invasive when it monitors everything you do with your mouse and keyboard.

PART- B

1. What problem was Kerberos designed to address? [C04-L1]

Kerberos
Kerberos4 is an authentication service developed as part of Project Athena at MIT. The problem that Kerberos addresses is this:

Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network.

1. A user may gain access to a particular workstation and pretend to be another user operating from that workstation.
2. A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.

3. A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access.

Rather than building in elaborate authentication protocols at each server, Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.

Motivation

1. Rely on each individual client workstation to assure the identity of its user or users and rely on each server to enforce a security policy based on user identification (ID).

2. Require that client systems authenticate themselves to servers, but trust the client system concerning the identity of its user.
3. Require the user to prove his or her identity for each service invoked. Also require that servers prove their identity to clients.

The first published report on Kerberos lists the following requirements.

Secure: A network eavesdropper should not be able to obtain the necessary information to impersonate a user. More generally, Kerberos should be strong enough that a potential opponent does not find it to be the weak link.

Reliable: For all services that rely on Kerberos for access control, lack of availability of the Kerberos service means lack of availability of the supported services. Hence, Kerberos should be highly reliable and should employ distributed server architecture, with one system able to back up another.

Transparent: Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.

Scalable: The system should be capable of supporting large numbers of clients

- Version 4 of Kerberos makes use of DES, in a rather elaborate protocol, to provide the authentication service.
- Viewing the protocol as a whole, it is difficult to see the need for the many elements contained therein.
- Therefore, we adopt a strategy used by Bill Bryant of Project Athena [BRYA88] and build up to the full protocol by looking first at several hypothetical dialogues.

- Each successive dialogue adds additional complexity to counter security vulnerabilities revealed in the preceding dialogue.

A Simple Authentication Dialogue

In an unprotected network environment, any client can apply to any server for service. The obvious security risk is that of impersonation.

- An opponent can pretend to be another client and obtain unauthorized privileges on server machines.
- To counter this threat, servers must be able to confirm the identities of clients who request service.
- Each server can be required to undertake this task for each client/server interaction, but in an open environment, this places a substantial burden on each server.
- An alternative is to use an authentication server (AS) that knows the passwords of all users and stores these in a centralized database.

- In addition, the AS shares a unique secret key with each server. These keys have been distributed physically or in some other secure manner.

- Consider the following hypothetical dialogue:

A More Secure Authentication Dialogue

- Although the foregoing scenario solves some of the problems of authentication in an open network environment, problems remain.

- First, we would like to minimize the number of times that a user has to enter a password. Suppose each ticket can be used only once.
- If user C logs on to a workstation in the morning and wishes to check his or her mail at a mail server, C must supply a password to get a ticket for the mail server.
- If C wishes to check the mail several times during the day, each attempt requires reentering the password.

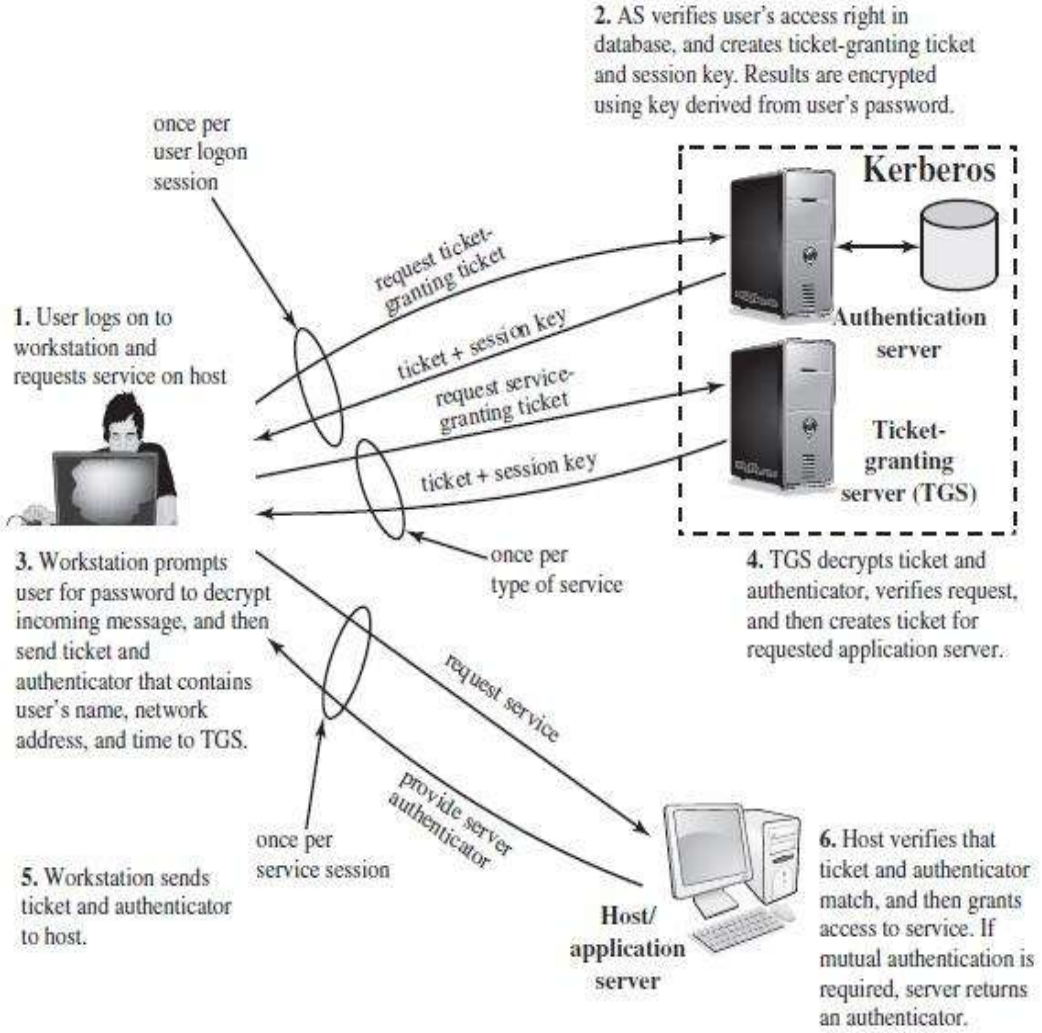


Figure 15.1 Overview of Kerberos

Kerberos Realms and Multiple Kerberi

A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires the following:

1. The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.
2. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server. Such an environment is referred to as a **Kerberos realm**.

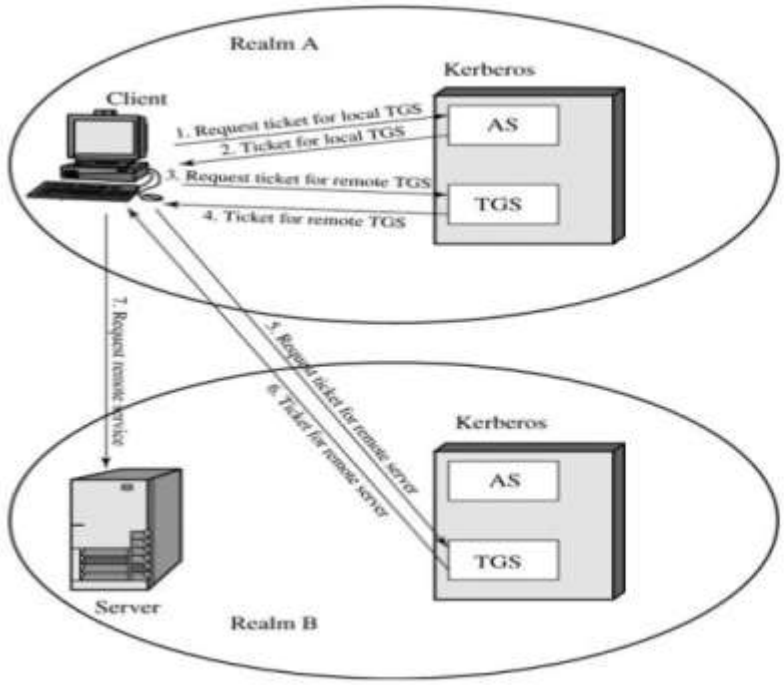
The concept of **realm** can be explained as follows. A Kerberos realm is a set of managed nodes that share the same Kerberos database.

Changing or accessing the contents of a Kerberos database requires the Kerberos master password. A related concept is that of a **Kerberos principal**, which is a service or user that is known to the Kerberos system.

Each Kerberos principal is identified by its principal name. Principal names consist of three parts: a service or user name, an instance name, and a realm name.

3. The Kerberos server in each interoperating realm shares a secret key with the server in the other realm. The two Kerberos servers are registered with each other.

The scheme requires that the Kerberos server in one realm trust the Kerberos server in the other realm to authenticate its users..



2. What is the purpose of the X.509 standard? How is an X.509 certificate revoked? or Write about X.509 authentication service? [C04-L1-Nov/Dec 2009-May/Jun 2013]

X.509 Authentication services

X.509 defines a framework for the provision of authentication services by the X.500 directory to its users. Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.

In addition, X.509 defines alternative authentication protocols based on the use of public-key certificates.

X.509 is based on the use of public-key cryptography and digital signatures. The standard does not dictate the use of a specific algorithm but recommends RSA. The digital signature scheme is assumed to require the use of a hash function.

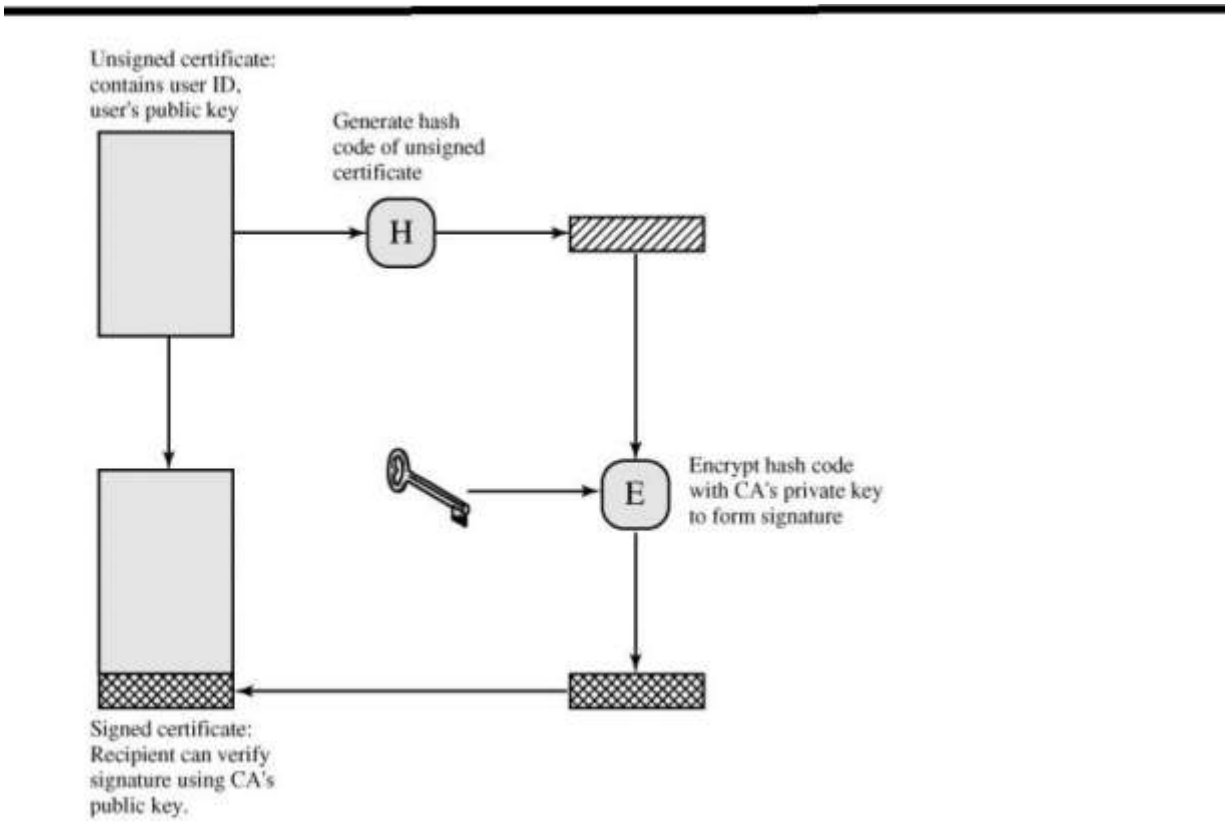


Figure 14.3. Public-Key Certificate Use

Certificates

The heart of the X.509 scheme is the public-key certificate associated with each user. These user certificates are assumed to be created by some trusted certification authority (CA) and placed in the directory by the CA or by the user.

The directory server itself is not responsible for the creation of public keys or for the certification function; it merely provides an easily accessible location for users to obtain certificates.

Figure 14.15a shows the general format of a certificate, which includes the following elements.

Version: Differentiates among successive versions of the certificate format; the default is version 1. If the Issuer Unique Identifier or Subject Unique Identifier are present, the value must be version 2. If one or more extensions are present, the version must be version 3.

Serial number: An integer value, unique within the issuing CA, that is unambiguously associated with this certificate.

Signature algorithm identifier: The algorithm used to sign the certificate, together with any associated parameters. Because this information is repeated in the Signature field at the end of the certificate, this field has little, if any, utility.

Issuer name: X.500 name of the CA that created and signed this certificate.

Period of validity: Consists of two dates: the first and last on which the certificate is valid.

Subject name: The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key.

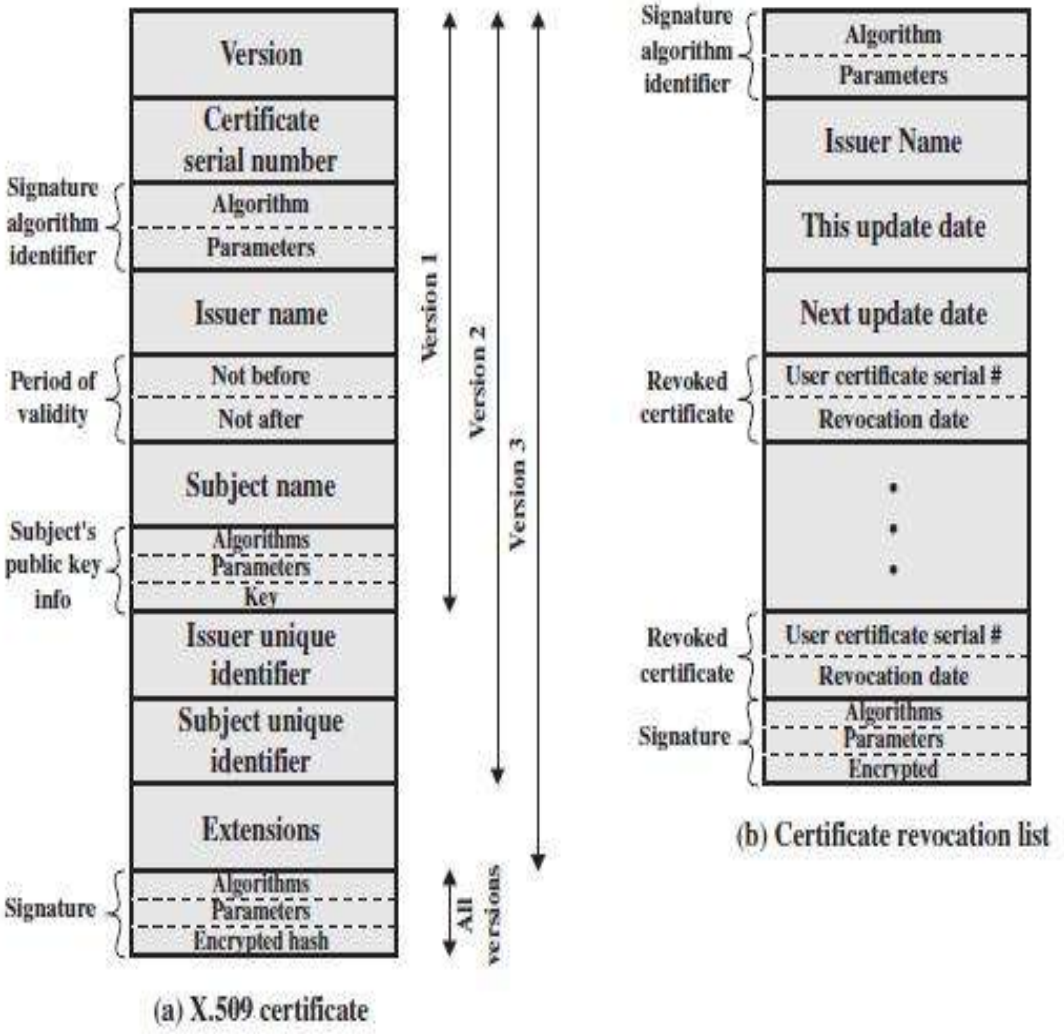
Subject's public-key information: The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.

Issuer unique identifier: An optional bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.

Subject unique identifier: An optional bit string field used to identify uniquely the subject in the event the X.500 name has been reused fordifferent entities.

Extensions: A set of one or more extension fields. Extensions were added in version 3 and are discussed later in this section.

Signature: Covers all of the other fields of the certificate; it contains the hash code of the other fields, encrypted with the CA's private key. This field includes the signature algorithm identifier.



Obtaining a User's Certificate

User certificates generated by a CA have the following characteristics:

- Any user with access to the public key of the CA can verify the user public key that was certified.
- No party other than the certification authority can modify the certificate without this being detected.
- Because certificates are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them.
- If all users subscribe to the same CA, then there is a common trust of that CA. All user certificates can be placed in the directory for access by all users. In addition, a user can transmit his or her certificate directly to other users.

- In either case, once B is in possession of A's certificate, B has confidence that messages it encrypts with A's public key will be secure from eavesdropping and that messages signed with A's private key are unforgeable.
- If there is a large community of users, it may not be practical for all users to subscribe to the same CA. Because it is the CA that signs certificates, each participating user must have a copy of the CA's own public key to verify signatures.
- This public key must be provided to each user in an absolutely secure (with respect to integrity and authenticity) way so that the user has confidence in the associated certificates. Thus, with many users, it may be more practical for there to be a number of CAs, each of which securely provides its public key to some fraction of the users.

X.509 Version 3

The X.509 version 2 format does not convey all of the information that recent design and implementation experience has shown to be needed.

1. The Subject field is inadequate to convey the identity of a key owner to a public-key user. X.509 names may be relatively short and lacking in obvious identification details that may be needed by the user.

2. The Subject field is also inadequate for many applications, which typically recognize entities by an Internet e-mail address, URL, or some other Internet-related identification.

3. Explain the Internet Firewalls for Trusted Systems Internet Firewalls for Trusted Systems.[CO4-L2]

A firewall is a device or group of devices that controls access between networks. A firewall generally consists of filters and gateway(s), varying from firewall to firewall. It is a security gateway that controls access between the public Internet and an intranet (a private internal network) and is a secure computer system placed between a trusted network and an untrusted internet.

Firewalls act as an intermediate server in handling SMTP and HTTP connections in either direction. Firewalls also require the use of an access negotiation and encapsulation protocol such as SOCKS to gain access to the Internet, the intranet, or both. Many firewalls support tri-homing, allowing use of a DMZ network. It is possible for a firewall to accommodate more than three interfaces, each attached to a different network segment.

Firewalls can be classified into three main categories: packet filters, circuit-level gateways and application-level gateways

4. What are the basic principles of Firewalls? List four techniques used by firewalls to control access and enforce a security policy and Firewall-Related Terminology.[C04-L1]

Role of Firewalls

The firewall itself must be immune to penetration.

Firewalls create checkpoints (or choke points) between an internal private network and an untrusted Internet. Once the choke points have been clearly established, the device can monitor, filter and verify all inbound and outbound traffic.

- The firewall may filter on the basis of IP source and destination addresses and TCP port number. Firewalls may block packets from the Internet side that claim a source address of a system on the intranet, or they may require the use of an access negotiation and encapsulation protocol like SOCKS to gain access to the intranet.
- The means by which access is controlled relate to using network layer or transport layer criteria such as IP subnet or TCP port number, but there is no reason that this must always be so. A growing number of firewalls control access at the application layer, using user identification as the criterion. In addition, firewalls for ATM networks may control access based on the data link layer criteria.
- The firewall also enforces logging, and provides alarm capacities as well. By placing logging services at firewalls, security administrators can monitor all access to and from the Internet. Good logging strategies are one of the most effective tools for proper network security.
- Firewalls may block TELNET or RLOGIN connections from the Internet to the intranet.
- They also block SMTP and FTP connections to the Internet from internal systems not authorised to send e-mail or to move files.
- The firewall provides protection from various kinds of IP spoofing and routing attacks. It can also serve as the platform for IPsec. Using the tunnel mode capability, the firewall can be used to implement Virtual Private Networks (VPNs).

Firewall-Related Terminology

To design and configure a firewall, some familiarity with the basic terminology is required.

.

Bastion Host

A bastion host is a publicly accessible device for the network's security, which has a direct connection to a public network such as the Internet.

The bastion host serves as a platform for any one of the three types of firewalls: packet filter, circuit-level gateway or application-level gateway.

Bastion hosts must check all incoming and outgoing traffic and enforce the rules specified in the security policy. They must be prepared for attacks from external and possibly internal sources. They should be built with the least amount of hardware and software in order for a potential hacker to have less opportunity to overcome the firewall.

The bastion host's role falls into the following three common types:

Single-homed bastion host: This is a device with only one network interface, normally used for an **application-level gateway**. The external
router is configured to send all incoming data to the bastion host, and all internal clients are configured to send all outgoing data to the host.
***Dual-homed bastion host*:** This is a firewall device with at least two network interfaces. Dual-homed bastion hosts serve as **application-level gateways**, and as **packet filters and circuit-level gateways as well**. The advantage of using such hosts is that they create a complete break between the external network and the internal network.
Multihomed bastion host: Single-purpose or internal bastion hosts can be classified as either single-homed or multihomed bastion hosts. The latter are used to allow the user to enforce strict security mechanisms.

Proxy Server

When the security policy requires all inbound and outbound traffic to be sent through a proxy server, a new proxy server should be created for the new streaming application. On the new proxy server, it is necessary to implement strict security mechanisms such as authentication.

SOCKS

The SOCKS protocol version 4 provides for unsecured firewall traversal for TCP-based client/server applications, including HTTP, TELNET and FTP.

The new protocol extends the SOCKS version 4 model to include UDP, and allows the framework to include provision for generalized strong authentication schemes, and extends the addressing scheme to encompass domain name and IPv6 addresses.

Choke Point

The most important aspect of firewall placement is to create choke points. A choke point is the point at which a public internet can access the internal network. The most comprehensive and extensive monitoring tools should be configured on the choke points.

Proper implementation requires that all traffic be funnelled through these choke points

.

5. What is the difference between a packet filtering firewall and a stateful inspection firewall? (or) What information is used by a typical packet filtering firewall? What are some weaknesses of a packet filtering firewall? What is an application-level gateway? What is a circuit-level gateway? Or Types of firewall [CO4-L1]

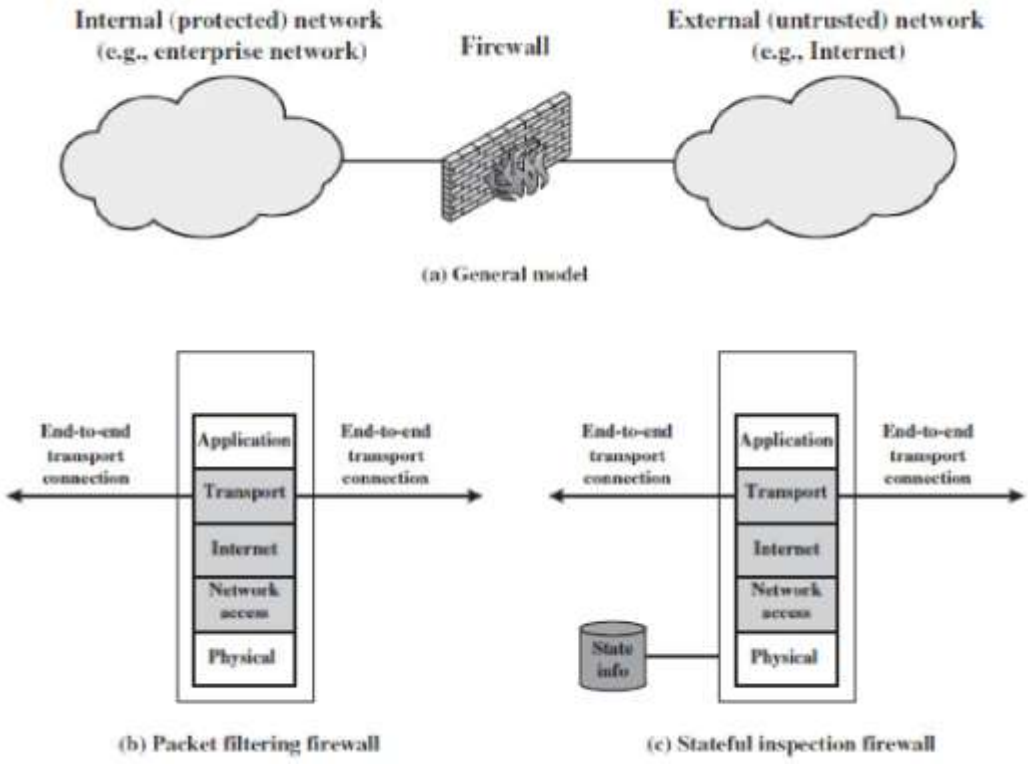
Types of Firewalls

A firewall may act as a packet filter. It can operate as a positive filter, allowing to pass only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria.

Packet Filtering Firewall

A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet (Figure 22.1b).

The firewall is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet:



Destination IP address: The IP address of the system the IP packet is trying to reach (e.g., 192.168.1.2)

Source and destination transport-level address: The transport-level (e.g.,TCP or UDP) port number, which defines applications such as SNMP or TELNET

IP protocol field: Defines the transport protocol

Interface: For a firewall with three or more ports, which interface of the firewall the packet came from or which interface of the firewall the packet is destined for

The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header.

Two default policies are possible:

1. **Default = discard:** That which is not expressly permitted is prohibited.
2. **Default = forward:** That which is not expressly prohibited is permitted.

The default discard policy is more conservative. Initially, everything is blocked, and services must be added on a case-by-case basis.

The default forward policy increases ease of use for end users but provides reduced security; the security administrator must, in essence, react to each new security threat as it becomes known.

6. What is a Firewall design and what types of systems would you expect to find on such networks? Or Firewall Configuration. [CO4-L1]

Firewall design

A security administrator must decide on the location and on the number of firewalls needed. to implement a firewall strategy. The primary step in designing a secure firewall is obviously to prevent the firewall devices from being compromised by threats.

To provide a certain level of security, the three basic firewall designs are considered: a single-homed bastion host, a dual-homed bastion host and a screened subnet firewall. The first two options are for creating a screened host firewall, and the third option contains an additional packet-filtering router to achieve another level of security.

Screened Host Firewall (Single-homed Bastion Host)

The first type of firewall is a **screened host** which uses a single-homed bastion host plus a packet-filtering router, as shown in Figure 10.4. Single-homed bastion hosts

can be configured as either circuit-level or application-level gateways. When using either of these two gateways, each of which is called a proxy server, the bastion host can hide the configuration of the internal network.

NAT is essentially needed for developing an address scheme internally. It is a critical component of any firewall strategy. It translates the internal IP addresses to IANA registered addresses to access the Internet. Hence, using NAT allows network administrators to use any internal IP address scheme.

The screened host firewall is designed such that all incoming and outgoing information is passed through the bastion host. The external screening router is configured to route all incoming traffic directly to the bastion host as indicated in Figure 10.4.

The screening router is also configured to route outgoing traffic only if it originates from the bastion host. This kind of configuration prevents internal clients from bypassing the bastion host. Thus, the bastion host is configured to restrict unacceptable traffic and proxy acceptable traffic.

A single-homed implementation may allow a hacker to modify the router not to forward packets to the bastion host. This action would bypass the bastion host and allow the hacker directly into the network.

But such a bypass usually does not happen because a network using a single-homed bastion host is normally configured to send packets only to the bastion host, and directly to the Internet.

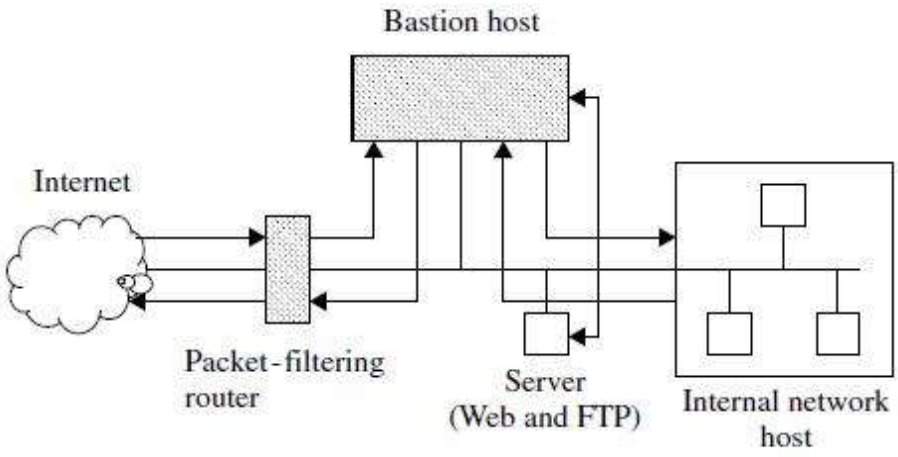


Figure 10.4 Screened host firewall system (single-homed bastion host).

Screened Host Firewall (Dual-homed Bastion Host)

The configuration of the screened host firewall using a dual-homed bastion host adds significant security, compared with a single-homed bastion host. As shown in Figure 10.5, a dual-homed bastion host has two network interfaces.

This firewall implementation is secure due to the fact that it creates a complete break between the internal network and the external Internet. As with the single-homed bastion, all external traffic is forwarded directly to the bastion host for processing.

However, a hacker may try to subvert the bastion host and the router to bypass the firewall mechanisms. Even if a hacker could defeat either the screening router or the dual-homed bastion host, the hacker would still have to penetrate the other. Nevertheless, a dual-homed bastion host removes even this possibility.

It is also possible to implement NAT for dual-homed bastion hosts.

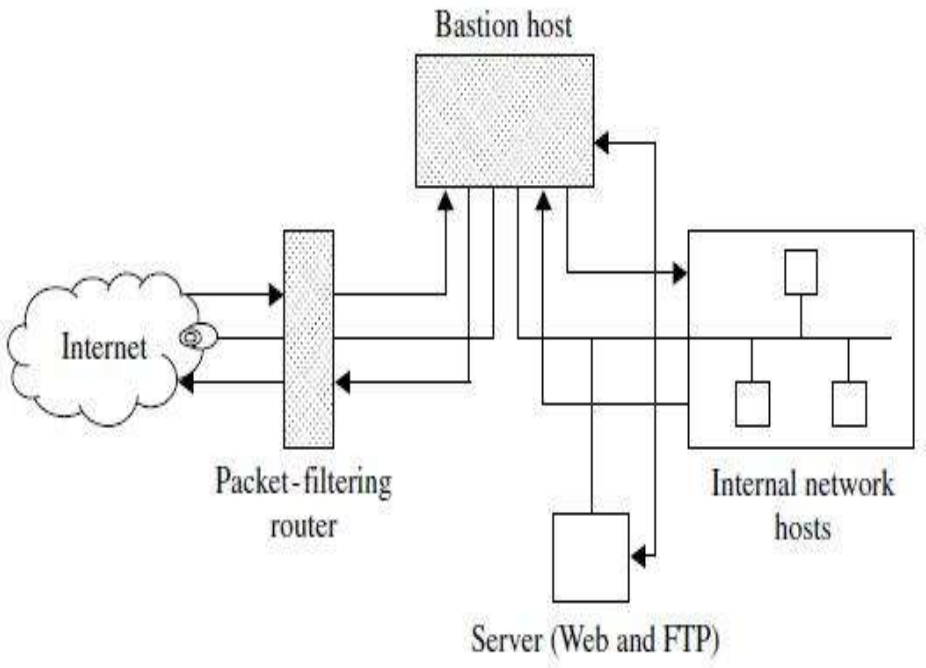


Figure 10.5 Screened host firewall system (dual-homed bastion host).

Screened Subnet Firewall

The third implementation of a firewall is the screened subnet, which is also known as a DMZ. This firewall is the most secure one among the three implementations, simply because it uses a bastion host to support both circuit- and application-level gateways. As shown in Figure 10.6, all publicly accessible devices, including modem and server, are placed inside the DMZ.

These DMZ then functions as a small isolated network positioned between the Internet and the internal network. The screened subnet firewall contains external and internal screening routers. Each is configured such that its traffic flows only to or from the bastion host. This arrangement prevents any traffic from directly traversing the DMZ subnetwork.

The external screening router uses standard filtering to restrict external access to the bastion host, and rejects any traffic that does not come from the bastion host. This router also uses filters to prevent attacks such as IP spoofing and source routing. The internal screening router also uses rules to prevent spoofing and source routing. Like its external counterpart, this internal router rejects incoming packets that do not originate from the bastion host, and sends only outgoing packets to the bastion host.

The benefits of the screened subnet firewall are based on the following facts. First, a hacker must subvert three separate tri-homed interfaces when he or she wants to access the internal network. But it is almost infeasible.

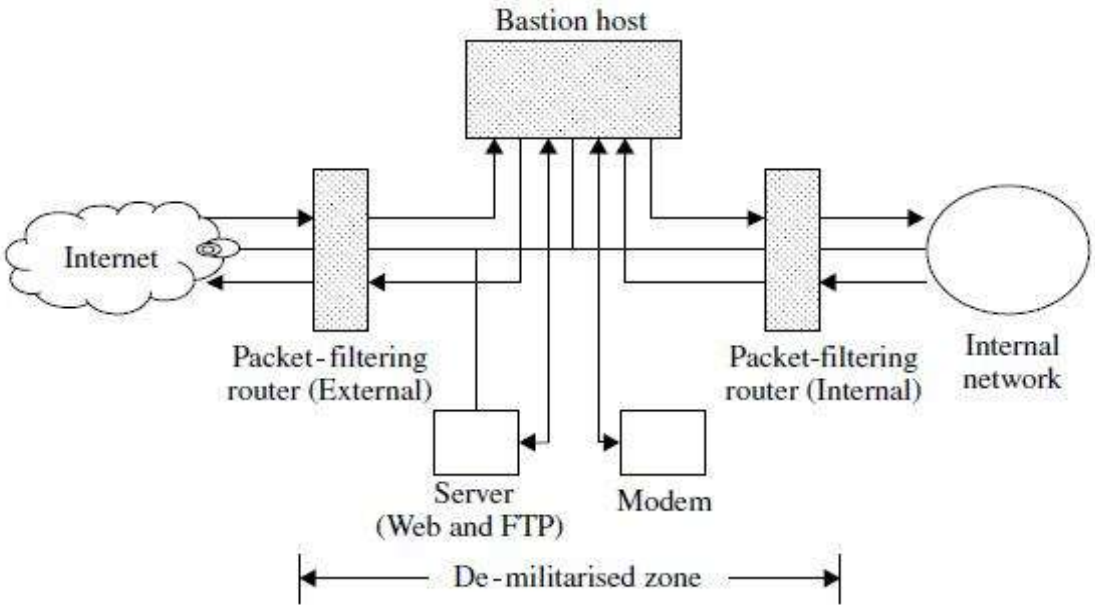


Figure 10.6 Screened subnet firewall system.

7. Mention the working principles and key features of SET for E-Commerce Transactions. [CO4-L3-Dec-15]

SET for E-commerce Transactions

The Secure Electronic Transaction (SET) is a protocol designed for protecting credit card transactions over the Internet. It is an industry-backed standard that was formed by MasterCard and Visa (acting as the governing body) in February 1996.

Business Requirements for SET

This section describes the major business requirements for credit card transactions by means of secure payment processing over the Internet.

They are listed below:

1. Confidentiality of information (provide confidentiality of payment and order information)
2. Integrity of data (ensure the integrity of all transmitted data)
3. Cardholder account authentication (provide authentication that a cardholder is a legitimate customer of a branded payment card account)
4. Merchant authentication (provide authentication that a merchant can accept credit card transactions through its relationship with an acquiring financial institution)
5. Security techniques (ensure the use of the best security practices and system design techniques to protect all legitimate parties in an electronic commerce transaction)
6. Creation of brand-new protocol (create a protocol that neither depends on transport security mechanisms nor prevents their use):
7. Interoperability (facilitate and encourage interoperability among software and network providers

SET System Participants

The participants in the SET system interactions are described in this section. A discrepancy is found between an SET transaction and a retail or mail order transaction: in a face-to face retail transaction, electronic processing begins with the merchant or the acquirer, but, in an SET transaction, the electronic processing begins with the cardholder.

Following are the participants of SET system.

• ***Cardholder***: A cardholder is an authorised holder of a payment card that has been issued by an issuer. In the cardholder's interactions, SET ensures that the payment card account information remains confidential.

• ***Issuer*****:** An issuer is a financial institution (a bank) that establishes an account for a cardholder and issues the payment card. The issuer guarantees payment for authorized transactions using the payment card.

• ***Merchant***: A merchant is a person or organisation that offers goods or services for sale to the cardholder. Typically, these goods or services are offered via a Website or by e-mail. With SET, the merchant can offer its cardholders secure electronic interactions.

• ***Acquirer***: An acquirer is the financial institution that establishes an account with a merchant and processes payment card authorization and payments.
• ***Payment gateway*****:** A payment gateway acts as the interface between a merchant and the acquirer. It carries out payment authorization services for many card brands and performs clearing services and data capture.
.• ***Certification Authority***: A CA is an entity that is trusted to issue X.509 v3 publickey certificates for cardholders, merchants and payment gateways.

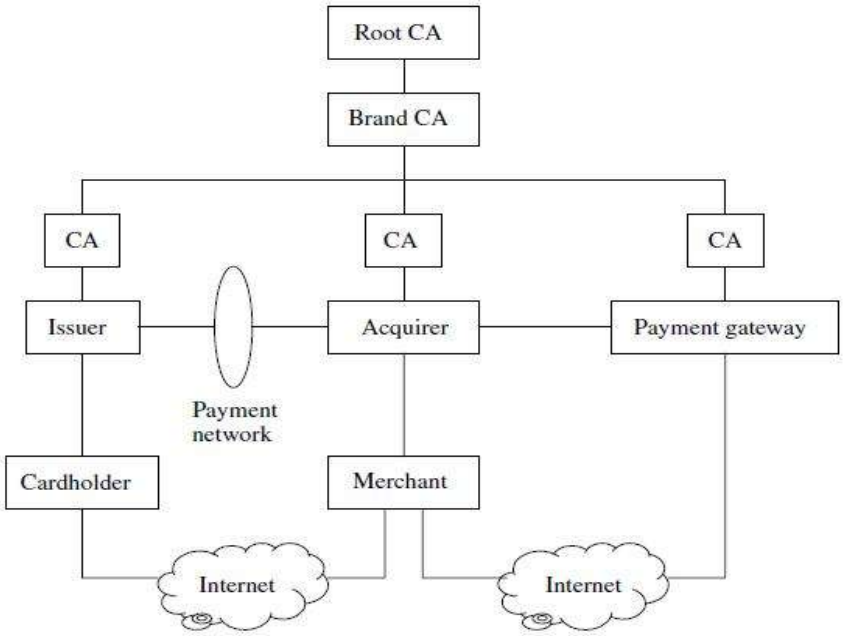


Figure 11.1 The SET hierarchy indicating the relationships between the participants.

Figure 11.1 illustrates the SET hierarchy which reflects the relationships between the participants in the SET system, described in the preceding paragraphs. In the SET environment, there exists a hierarchy of CAs. The SET protocol specifies a method of ***trust chaining*** for entity authentication.

This trust chain method entails the exchange of digital certificates and verification of the public keys by validating the digital signatures of the issuing CA. As indicated in Figure 11.1, this trust chain method continues all the way up to the root CA at the top of the hierarchy.

8. List and briefly define three classes of intruders. [CO4-L1]

Intruder

One of the two most publicized threats to security is the intruder, often referred to as a hacker or cracker.

The identified three classes of intruders:

- **Masquerader:** An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account
- **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges
- **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider.

The following are examples of intrusion:

- Performing a remote root compromise of an e-mail server
- Defacing a Web server
- Guessing and cracking passwords
- Copying a database containing credit card numbers
- Viewing sensitive data, including payroll records and medical information, without authorization
- Running a packet sniffer on a workstation to capture usernames and passwords
- Using a permission error on an anonymous FTP server to distribute pirated software and music files
- Dialing into an unsecured modem and gaining internal network access
- Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password
- Using an unattended, logged-in workstation without permission.

Intruder Behavior Patterns

The three broad examples of intruder behavior patterns are

Hackers Traditionally, those who hack into computers do so for the thrill of it or for status. The hacking community is a strong meritocracy in which status is determined by level of competence.

Criminals Organized groups of hackers have become a widespread and common threat to Internet-based systems. These groups can be in the employ of a corporation or government but often are loosely affiliated gangs of hackers.

Insider Attacks Insider attacks are among the most difficult to detect and prevent. Employees already have access and knowledge about the structure and content of corporate databases. Insider attacks can be motivated by revenge or simply a feeling of entitlement.

9. What are three benefits that can be provided by an intrusion detection system? What is the difference between statistical anomaly detection and rule-based intrusion detection? What metrics are useful for profile-based intrusion detection? What is the difference between rule-based anomaly detection and rule-based penetration identification? [CO4-L1- May 14, Dec 14,15]

Intrusion detection system

1. If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.

2. An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.

3. Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Figure 20.1 suggests, in very abstract terms, the nature of the task confronting the designer of an intrusion detection system.

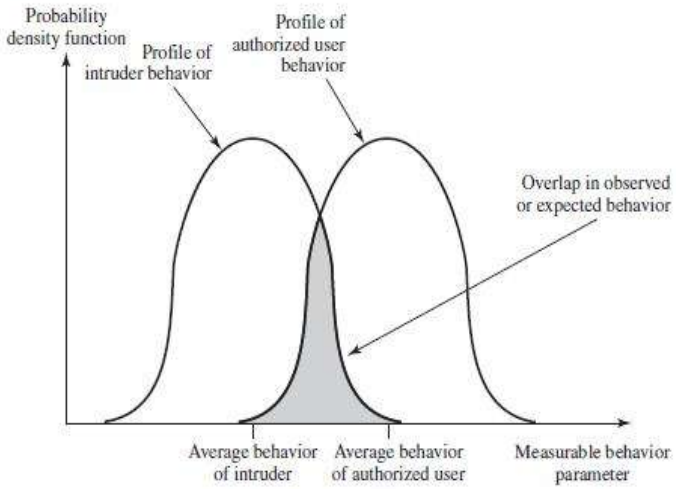


Figure 20.1 Profiles of Behavior of Intruders and Authorized Users

The following approaches to intrusion detection:

1. **Statistical anomaly detection:** Involves the collection of data relating to the behavior of legitimate users over a period of time.
   a. ***Threshold detection:*** This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.
   b. ***Profile based:*** A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.
2. **Rule-based detection:** Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.
   a. ***Anomaly detection:*** Rules are developed to detect deviation from previous usage patterns.
   b. ***Penetration identification:*** An expert system approach that searches for suspicious behavior.

Audit Records

A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Basically, two plans are used:

- **Native audit records:** Virtually all multiuser operating systems include accounting software that collects information on user activity.
- **Detection-specific audit records:** A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system.

Each audit record contains the following fields:

- **Subject:** Initiators of actions.
- **Action:** Operation performed by the subject on or with an object.
- **Object:** Receptors of actions.
- **Exception-Condition:** Denotes which, if any, exception condition is raised on return.
- **Resource-Usage:** A list of quantitative elements in which each element gives the amount used of some resource.
- **Time-Stamp:** Unique time-and-date stamp identifying when the action took place.

Statistical Anomaly Detection

Statistical anomaly detection techniques fall into two broad categories:

Threshold detection systems: Threshold detection involves counting the number of occurrences of a specific event type over an interval of time.

Profile-based systems: Profile-based anomaly detection focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant deviations.

Rule-Based Intrusion Detection

Rule-based techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious.

Rule-based anomaly detection is similar in terms of its approach and strengths to statistical anomaly detection.

With the rule-based approach, historical audit records are analyzed to identify usage patterns and to generate automatically rules that describe those patterns. Rules may represent past behavior patterns of users, programs, privileges, time slots, terminals, and so on.

Rule-based penetration identification takes a very different approach to intrusion detection. The key feature of such systems is the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses.

Rules can also be defined that identify suspicious behavior, even when the behavior is within the bounds of established patterns of usage.

The Base-Rate Fallacy

To be of practical use, an intrusion detection system should detect a substantial percentage of intrusions while keeping the false alarm rate at an acceptable level. If only a modest percentage of actual intrusions are detected, the system provides a false sense of security.

On the other hand, if the system frequently triggers an alert when there is no intrusion (a false alarm), then either system managers will begin to ignore the alarms, or much time will be wasted analyzing the false alarms.

Distributed Intrusion Detection

Until recently, work on intrusion detection systems focused on single-system standalone facilities.

Porras points out the following major issues in the design of a distributed intrusion detection system :

- A distributed intrusion detection system may need to deal with different audit record formats.
- One or more nodes in the network will serve as collection and analysis points for the data from the systems on the network.

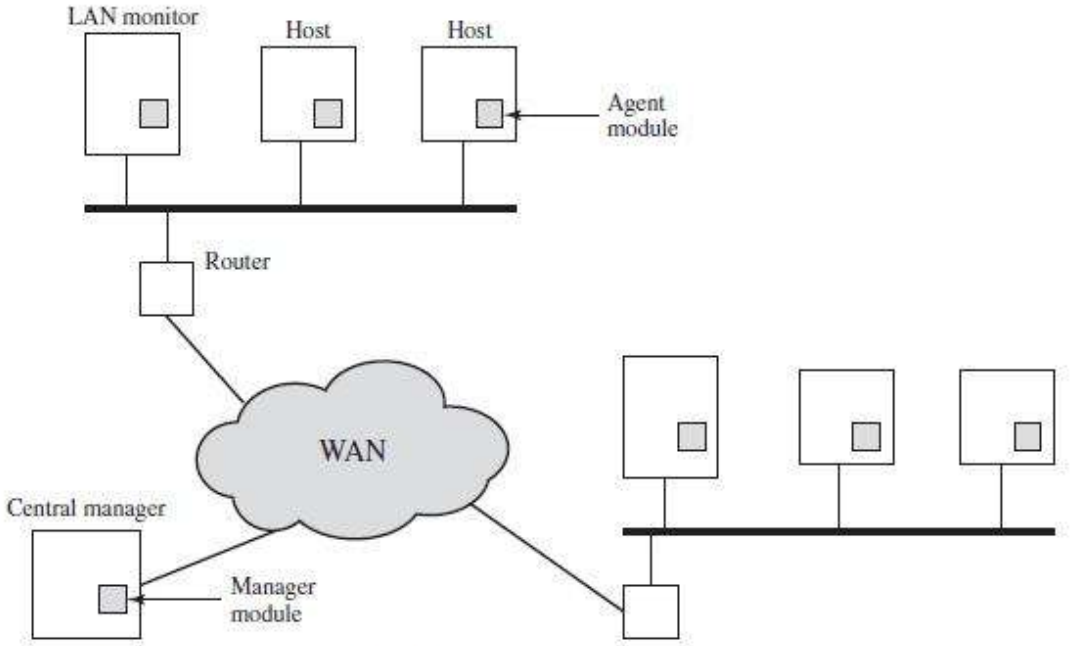


Figure 20.2 Architecture for Distributed Intrusion Detection

A good example of a distributed intrusion detection system, which consists of three main components:

- **Host agent module:** An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security related events on the host and transmit these to the central manager.
- **LAN monitor agent module:** Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager.
- **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.

Figure 20.3 shows the general approach that is taken. The agent captures each audit record produced by the native audit collection system. A filter is applied that retains only those records that are of security interest.

At the lowest level, the agent scans for notable events that are of interest independent of any past events.

At the next higher level, the agent looks for sequences of events, such as known attack patterns (signatures).

Finally, the agent looks for anomalous behavior of an individual user based on a historical profile of that user, such as number of programs executed, number of files accessed, and the like.

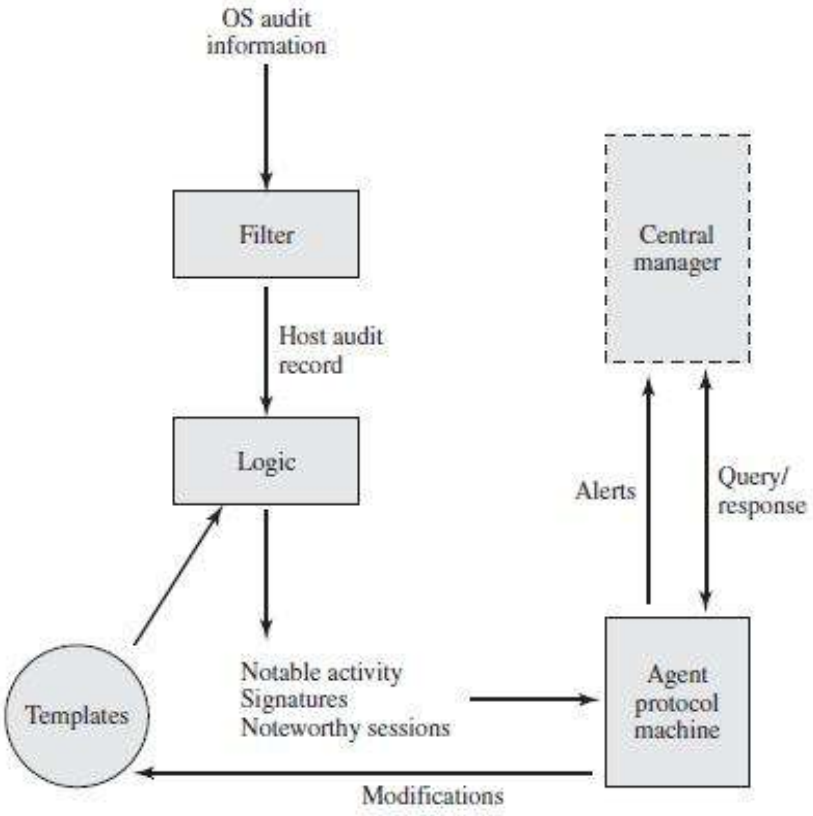


Figure 20.3 Agent Architecture

Honeypots

- A relatively recent innovation in intrusion detection technology is the honeypot. Honeypots are decoy systems that are designed to *lure a potential attacker* away from critical systems.
- Honeypots are designed to
  - divert an attacker from accessing critical systems
  - collect information about the attacker's activity
  - encourage the attacker to stay on the system long enough for administrators to respond
- Initial efforts involved a single honeypot computer with IP addresses designed to attract hackers. More recent research has focused on building entire honeypot networks that emulate an enterprise, possibly with actual or simulated traffic and data. Once hackers are within the network, administrators can observe their behavior in detail and figure out defenses.

10. What is the role of compression in the operation of a virus? What is the role of encryption in the operation of a virus? What are typical phases of operation of a virus or worm?. What is it? [CO4-L1-Dec-13]

Or

Explain about viruses and related threats? [CO4-L2-MAY/JUN 2014]

Or

Explain in detail about malicious software?

[CO4-L2-Nov/Dec 2012-May/Jun 2013-Apr/May 2011-Nov/Dec 2013]

Perhaps the most sophisticated types of threats to computer systems are presented by programs that exploit vulnerabilities in computing systems.

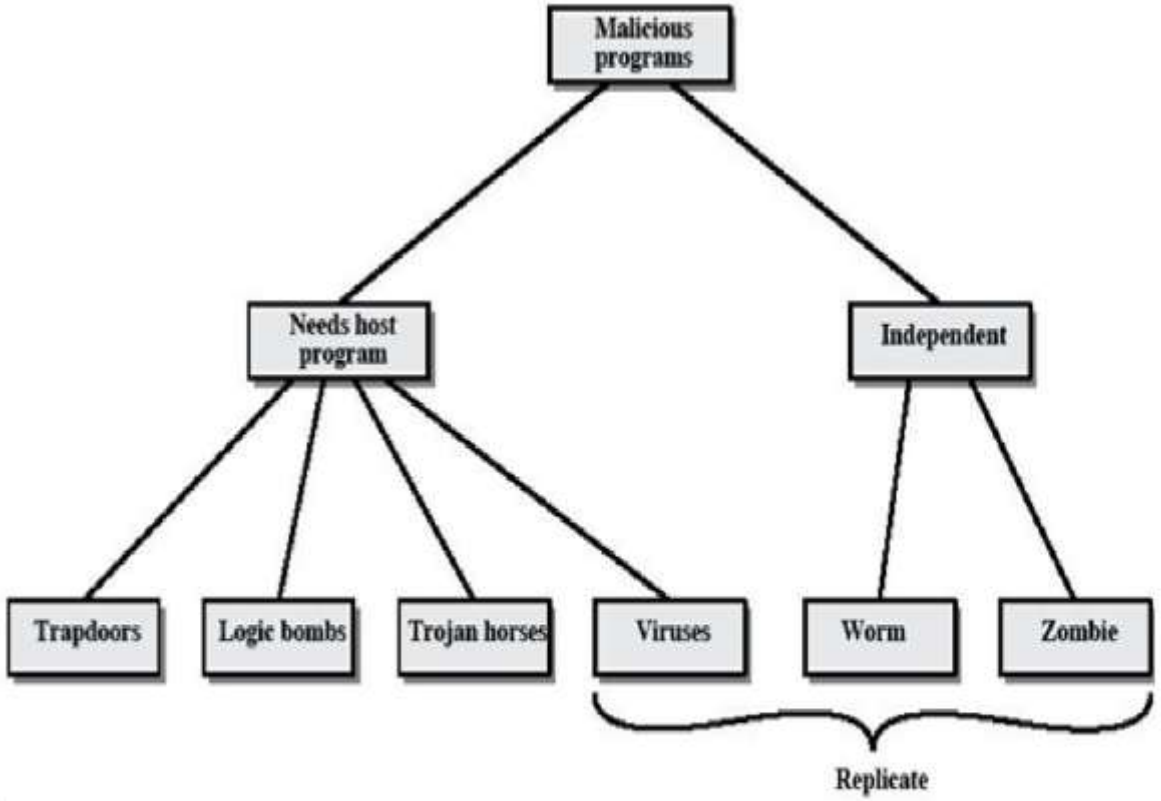


Figure 19.1 Taxonomy of Malicious Programs

Malicious Programs

Name	Description
Virus	Attaches itself to a program and propagates copies of itself to other Programs
Worm	Program that propagates copies of itself to other computers
Logic bomb	Triggers action when condition occurs

Malicious software can be divided into two categories:

Those that need a host program, and those that are independent.
The former are essentially fragments of programs that cannot exist independently of some

actual application program, utility, or system program.

Viruses, logic bombs, and backdoors are examples. The latter are self-contained programs that can be scheduled and run by the operating system. Worms and zombie programs are examples.

Trojan horse	Program that contains unexpected additional functionality
Backdoor (trapdoor)	Program modification that allows unauthorized access to Functionality
Exploits	Code specific to a single vulnerability or set of vulnerabilities
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely
Kit (virus generator)	Set of tools for generating new viruses automatically
Spamme r programs	Used to send large volumes of unwanted e-mail
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial of service (DoS) attack
Keyloggers	Captures keystrokes on a compromised system
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access
Zombie	Program activated on an infected machine that is activated to launch attacks on other machines

Backdoor:

- A backdoor, also known as a trapdoor, is a secret entry point into a program that allows someone that is aware of the backdoor to gain access without going through the usual security access procedures.
- Programmers have used backdoors legitimately for many years to debug and test programs.
- This usually is done when the programmer is developing an application that has an authentication procedure, or a long setup, requiring the user to enter many different values to run the application.
- To debug the program, the developer may wish to gain special privileges or to avoid all the necessary setup and authentication.
- The programmer may also want to ensure that there is a method of activating the program should something be wrong with the authentication procedure that is being built into the application.
- The backdoor is code that recognizes some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events.
- Backdoors become threats when unscrupulous programmers use them to gain unauthorized access.
- The backdoor was the basic idea for the vulnerability portrayed in the movie *War Games*.

Logic Bomb

- One of the oldest types of program threat, predating viruses and worms, is the logic bomb.
- The logic bomb is code embedded in some legitimate program that is set to "explode" when certain conditions are met.
- Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application.
- Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.
- A striking example of how logic bombs can be employed was the case of Tim Lloyd, who was convicted of setting a logic bomb that cost his employer, Omega Engineering, more than $10 million, derailed its corporate growth strategy, and eventually led to the layoff of 80 workers.
- Ultimately, Lloyd was sentenced to 41 months in prison and ordered to pay $2 million in restitution.

Trojan Horses

- A Trojan horse is a useful, or apparently useful, program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function.
- Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly.

Zombie

- A zombie is a program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the zombie's creator.
- Zombies are used in denial-of-service attacks, typically against targeted Web sites.
- The zombie is planted on hundreds of computers belonging to unsuspecting third parties, and then used to overwhelm the target Web site by launching an overwhelming onslaught of Internet traffic. Section 19.3 discusses zombies in the context of denial of service attacks.

The Nature of Viruses

- A computer virus is a piece of software that can **"infect"** other programs by modifying them; the modification includes injecting the original program with a routine to make copies of the virus program, which can then go on to infect other programs.
- A virus can do anything that other programs do. The difference is that a virus attaches itself to another program and executes secretly when the host program is run.
- **A computer virus has three parts:**

  • **Infection mechanism:** The means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the **infection vector**.

  • **Trigger:** The event or condition that determines when the payload is activated or delivered.

  • **Payload:** What the virus does, besides spreading. During its lifetime,

  A typical virus goes through the following four phases:
  - **Dormant phase:** The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit.
  - **Propagation phase:** The virus places a copy of itself into other programs or into certain system areas on the disk.
  - **Triggering phase:** As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
  - **Execution phase:** The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.
- **Virus Structure** A virus can be prepended or postpended to an executable program, or it can be embedded in some other fashion. In this case, the virus code,V, is prepended to infected programs, and it is assumed that the entry point to the program, when invoked, is the first line of the program.

11. Describe some worm countermeasures. What is a digital immune system? How does behavior-blocking software work? [CO4-L1-May-14]

Antivirus Approaches

The ideal solution to the threat of viruses is prevention:

- Do not allow a virus to get into the system in the first place, or
- block the ability of a virus to modify any files containing executable code or macros.

The next best approach is to be able to do the following:

• **Detection:** Once the infection has occurred, determine that it has occurred and locate the virus.
• **Identification:** Once detection has been achieved, identify the specific virus that has infected a program.
• **Removal:** Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that the virus cannot spread further.

The four generations of antivirus software:

- **First generation: simple scanners**
- **Second generation: heuristic scanners**
- **Third generation: activity traps**
- **Fourth generation: full-featured protection**

A **first-generation** scanner requires a virus signature to identify a virus. The virus may contain “wildcards” but has essentially the same structure and bit pattern in all copies.

A **second-generation** scanner does not rely on a specific signature. Rather, the scanner uses heuristic rules to search for probable virus infection.

Another second-generation approach is integrity checking. A checksum can be appended to each program. If a virus infects the program without changing the checksum, then an integrity check will catch the change.

Third-generation programs are memory-resident programs that identify a virus by its actions rather than its structure in an infected program.

Fourth-generation products are packages consisting of a variety of antivirus techniques used in conjunction.

Advanced Antivirus Techniques

More sophisticated antivirus approaches and products continue to appear. In this subsection, we highlight some of the most important.

Generic Decryption:

Generic decryption (GD) technology enables the antivirus program to easily detect even the most complex polymorphic viruses while maintaining fast scanning speeds. In order to detect such a structure, executable files are run through a GD scanner,

- **CPU emulator:** A software-based virtual computer. Instructions in an executable file are interpreted by the emulator rather than executed on the underlying processor.
- **Virus signature scanner:** A module that scans the target code looking for known virus signatures.
- **Emulation control module:** Controls the execution of the target code.

Digital Immune System:

The motivation for this development has been the rising threat of Internet-based virus propagation.

The two major trends in Internet technology are

- **Integrated mail systems:** Systems such as Lotus Notes and Microsoft Outlook make it very simple to send anything to anyone and to work with objects that are received.
- **Mobile-program systems:** Capabilities such as Java and ActiveX allow programs to move on their own from one system to another.

Figure 21.4 illustrates the typical steps in digital immune system operation:

1. A monitoring program on each PC uses a variety of heuristics based on system behavior.
2. The administrative machine encrypts the sample and sends it to a central virus analysis machine.
3. This machine creates an environment in which the infected program can be safely run for analysis.
4. The resulting prescription is sent back to the administrative machine.

5. The administrative machine forwards the prescription to the infected client.

6. The prescription is also forwarded to other clients in the organization.
7. Subscribers around the world receive regular antivirus updates that protect them from the new virus.

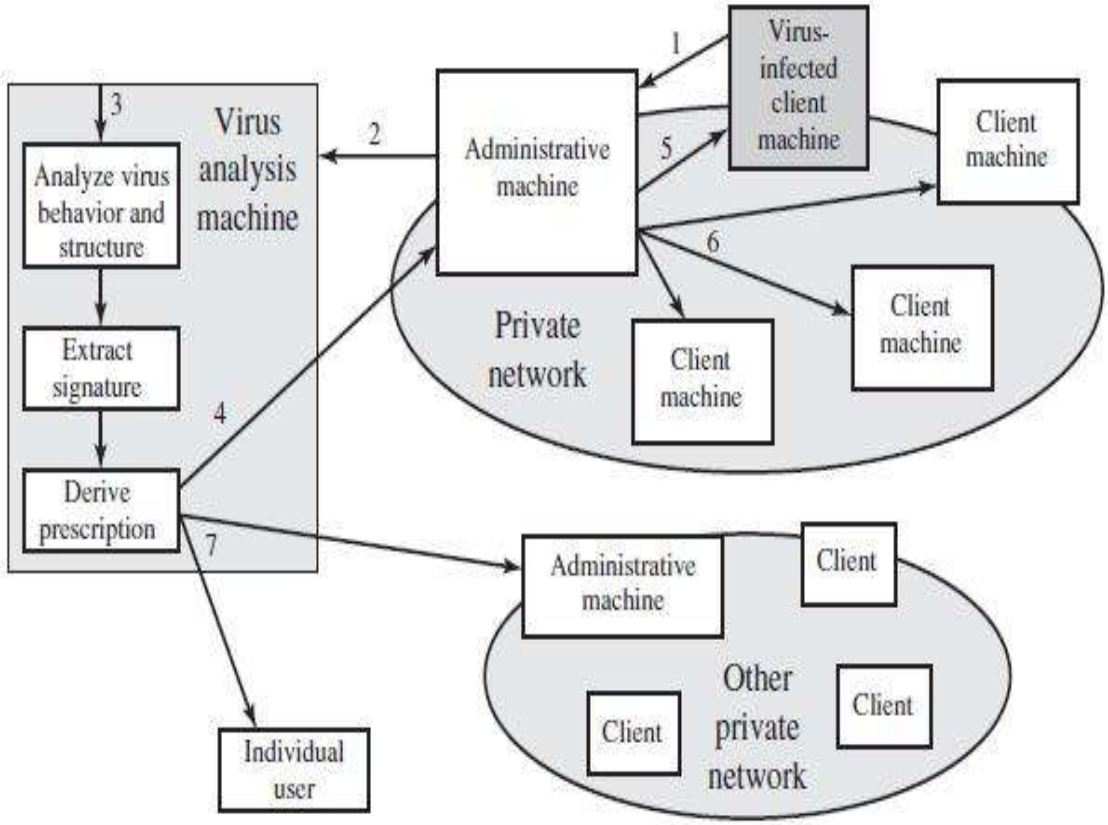


Figure 21.4 Digital Immune System

Behavior-Blocking Software:

*T*he behavior blocking software blocks potentially malicious actions before they have a chance to affect the system.

Monitored behaviors can include

o Attempts to open, view, delete, and/or modify files;
o Attempts to format disk drives and other unrecoverable disk operations; o Modifications to the logic of executable files or macros;
o Modification of critical system settings, such as start-up settings;
o Scripting of e-mail and instant messaging clients to send executable content; and o Initiation of network communications.

Figure 21.5 illustrates the operation of a behavior blocker. Behavior-blocking software runs on server and desktop computers and is instructed through policies set by the network administrator to let benign actions take place but to intercede when unauthorized or suspicious actions occur.

12. Explain the concept of Trusted systems. [CO4-L2]

Trusted systems

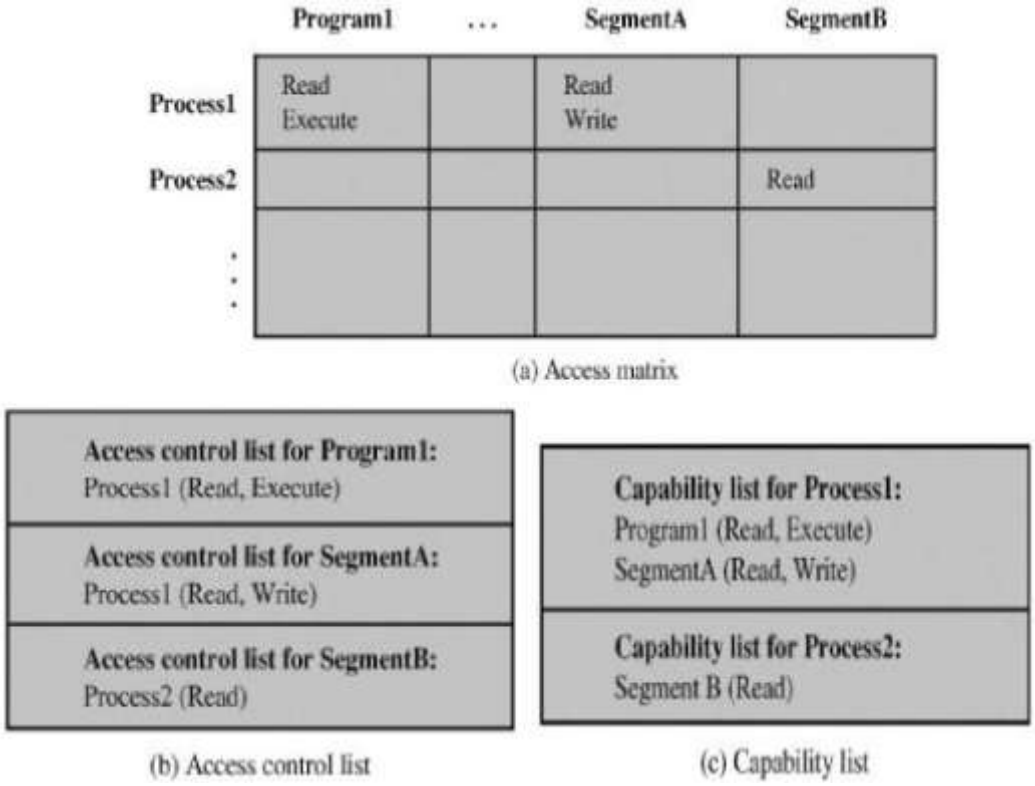
One way to enhance the ability of a system to defend against intruders and malicious programs is to implement trusted system technology.

Data Access Control

A general model of access control as exercised by a file or database management system is that of an **access matrix** (Figure 20.3a).

The basic elements of the model are as follows:

● **Subject**: An entity capable of accessing objects. Generally, the concept of subject equates with that of process.
● **Object:** Anything to which access is controlled. Examples include files, portions of files, programs, and segments of memory.
● **Access right**: The way in which an object is accessed by a subject. Examples are read, write, and execute.



A multilevel secure system must enforce the following:

- **No read up:** A subject can only read an object of less or equal security level. This is referred to in the literature as the **Simple Security Property**.
- **No write down:** A subject can only write into an object of greater or equal security level. This is referred to in the literature as the *-**Property**

The asterisk was a dummy character entered in the draft so that a text editor could rapidly find and replace all instances of its use once the property was named. No name was ever devised, and so the report was published with the "*" intact.

For a data processing system, the approach that has been taken, and has been the object of much research and development, is based on the ***reference monitor*** concept. The reference monitor is a controlling element in the hardware

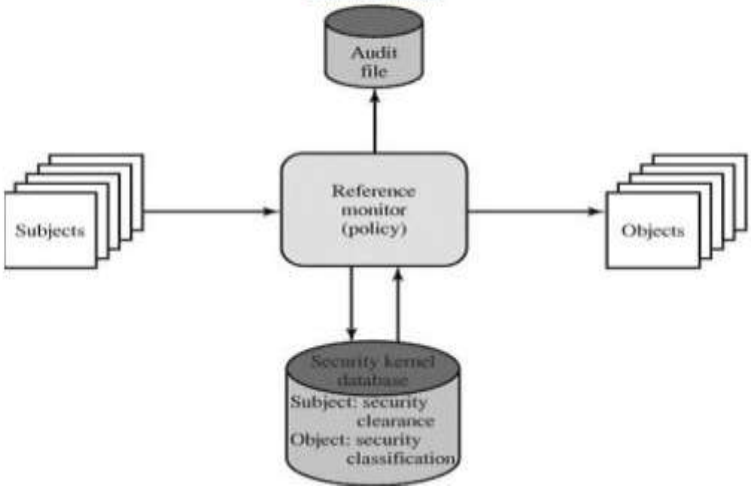
and operating system of a computer that regulates the access of subjects to objects on the basis of security parameters of the subject and object.

The reference monitor enforces the security rules (no read up, no write down) and has the following properties:

● **Complete mediation:** The security rules are enforced on every access, not just, for example, when a file is opened.
● **Isolation:** The reference monitor and database are protected from unauthorized modification.
● **Verifiability:** The reference monitor's correctness must be provable.

Figure 20.4. Reference Monitor Concept

[View full size image]



Unit – V

E-Mail, IP & Web Security

Part – A

1. Mention the services provided by the Pretty Good Privacy (PGP). [CO5-L2]

Authentication
Confidentiality
Compression
E-mail compatibility
Segmentation and reassembly

2. Signature is generated before compression in PGP. Why? [C05-L1]

There are two reasons behind it.

It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.

If one signed a compressed document, then it would be necessary either to store a compressed version of the message

For later verification or to recompress the message when verification is required.

Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty.

3. How E-mail compatibility is performed? [C05-L1]

Radix-64 is the technique which is used for E-mail compatibility. In

Radix-64, each group of 3 octets of binary data is mapped into4 ASCII characters.

4. What is the need of public key ring and private key ring? [C05-L1]

Public key ring is one of the data structures which is used to store the public keys of the other participants

Private Key ring is a data structure which is used to store the public and the private keys of the owner alone.

5.Mention the benefits of IPSec. [C05-L2]

It provides strong security that can be applied to all traffic crossing the perimeter

IPSec in a firewall is resistant to bypass.

IPSec is below the transport layer and so is transparent to applications.

IPSec is transparent to users.

6. List out the services provided by the IPSec. [C05-L1]

Access control
Connectionless integrity
Data origin authentication
Rejection of replayed packets
Confidentiality
Limited traffic flow confidentiality

7. Name the protocols that provide security in IPSec. [C05-L1]

Authentication header

Encapsulating security payload

8. What are the services provided by PGP services? [C05-L1]

Digital signature
Message encryption
Compression
E-mail compatibility
Segmentation

9. Explain the reasons for using PGP? [C05-L2]

It is available free worldwide in versions that run on a variety of platforms, including DOS/windows, UNIX, Macintosh and many more.
It is based on algorithms that have survived extensive public review and are considered extremely secure. E.g.) RSA, DSS and Diffie-Hellman for public key encryption, ST-128, IDEA, 3DES for conventional encryption, SHA-1for hash coding.
It has a wide range of applicability from corporations that wish to select and enforce a standardized scheme for encrypting files and communication.

10. Why E-mail compatibility function in PGP needed? [C05-L1]

Electronic mail systems only permit the use of blocks consisting of ASCII text. To accommodate this restriction PGP provides the service converting the row 8- bit binary stream to a stream of printable ASCII characters. The scheme used for this purpose is Radix-64 conversion.

11. Name any cryptographic keys used in PGP? [C05-L1]

One-time session conventional keys.

Public keys.

Private keys.

Pass phrase based conventional keys

12. Define key Identifier? [C05-L1]

PGP assigns a key ID to each public key that is very high probability unique with a user ID.
It is also required for the PGP digital signature.
The key ID associated with each public key consists of its least significant 64bits.

13. List the limitations of SMTP/RFC 822? [C05-L1]

SMTP cannot transmit executable files or binary objects.

It cannot transmit text data containing national language characters.

SMTP servers may reject mail message over certain size.

SMTP gateways cause problems while transmitting ASCII and EBCDIC.

SMTP gateways to X.400 E-mail network cannot handle non textual data included in X.400 messages.

14. Define S/MIME? [C05-L1]
Secure/Multipurpose Internet Mail Extension(S/MIME) is a security enhancement to the MIME Internet E-mail format standard, based on technology from RSA Data Security.

15. What are the elements of MIME? [C05-L1]

Five new message header fields are defined which may be included in an RFC 822 header.

A number of content formats are defined.

Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

16. What are the headers fields define in MME? [C05-L1]

MIME version.

Content type.

Content transfer encoding.

Content id.

Content description.

17. What is MIME content type &explain? [C05-L1]

It is used to declare general type of data. Subtype define particular format for that type of the data. It has 7 content type & 15 subtypes. They are,

Text type
Multipart type
Message type
Image type
Video type.
Audio type.
Application type

18. What are the key algorithms used in S/MIME? [C05-L1]
Digital signature standards.

Diffe- Hellman.

RSA algorithm.

19. Give the steps for preparing envelope data MIME? [C05-L2]

Generate Ks.

Encrypt Ks using recipient's public key.

RSA algorithm used for encryption.

Prepare the 'recipient info block'.

Encrypt the message using Ks.

20. What you mean by versioned certificate? [C05-L1]

Mostly used issue X.509 certificate with the product name" versioned digital id".

Each digital id contains owner's public key, owner's name and serial number of the digital id

21. What are the function areas of IP security? [C05-L1]

Authentication

Confidentiality

Key management.

22. Give the application of IP security? [C05-H1]

Provide secure communication across private & public LAN.

Secure remote access over the Internet.

Secure communication to other organization.

23. Specify the IP security services? [C05-L1]

Access control.
Connectionless interpretty.
Data origin authentication
Rejection of replayed packet.
Confidentiality.
Limited traffic for Confidentiality.

24,What do you mean by Security Association? Specify the parameters that identifies the Security Association? [C05-L1]

An association is a one-way relationship between a sender and receiver that affords security services to the traffic carried on.

A key concept that appears in both the authentication and confidentiality mechanism for ip is the security association (SA).

A security Association is uniquely identified by 3 parameters:

Security Parameter Index (SPI).

IP Destination Address.

Security Protocol Identifier.

25.What does you mean by Replay Attack? [C05-L1]

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination.

Each time a packet is sending the sequence number is incremented .

26. **Explain man in the middle attack? [C05-L1]**

If A and B exchange message, means E intercept the message and receive the B's public key and b's userId,E sends its own message with its own public key and b's userID based on the private key and Y.B compute the secret key and A compute k2 based on private key of A and Y.

27. Steps involved in SSL required protocol? [C05-H1]

SSL record protocol takes application data as input and fragments it.

Apply lossless Compression algorithm.

Compute MAC for compressed data.

MAC and compression message is encrypted using conventional algorithm.

28. What is mean by SET? What are the features of SET? [C05-L1]

a. Secure Electronic Transaction (SET) is an open encryption and security specification designed to protect credit card transaction on the Internet.

b. Features are:

i. Confidentiality of information
ii. Integrity of data
iii. Cardholder account authentication
iv. Merchant authentication

29. What are the steps involved in SET Transaction? [C05-L1]

The customer opens an account
The customer receives a certificate
Merchants have their own certificate
The customer places an order.
The merchant is verified.
The order and payment are sent.
The merchant requests payment authorization.
The merchant confirms the order.
The merchant provides the goods or services.
The merchant requests payment.

30. What is the purpose of X.509 standard? [C05-L1]

X.509 defines framework for authentication services by the X.500 directory to its users.X.509 defines authentication protocols based on public key certificates

31. What protocols comprise SSL? [C05-L1]

- SSL record protocol
- SSL change cipher spec protocol
- SSL alert protocol
- SSL hand shake protocol

32. Difference between conventional and public key encryption? [C05-L3]

Conventional encryption:

It is used in IDEA and 3DES
Kerberos use conventional encryption
It is used in RSA and DSS
It is also used in S/MIME.

33. Expand and define SPI? [C05-L1]

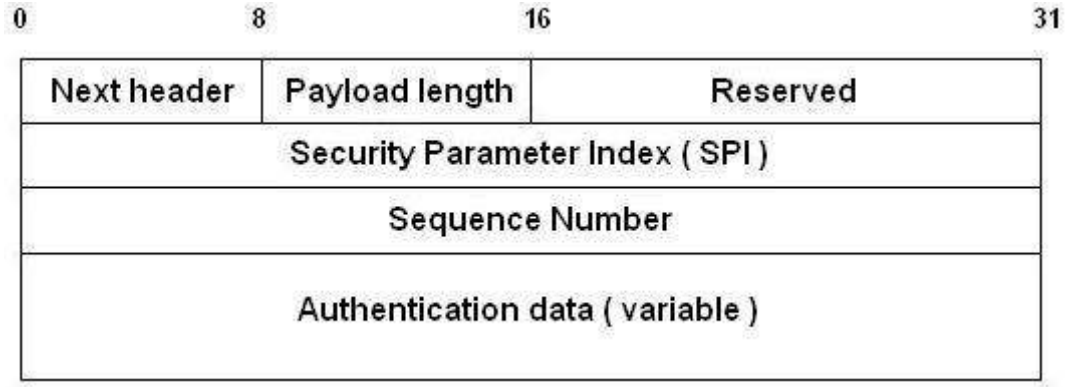
relates IP traffic to specific SAs

match subset of IP traffic to relevant SA

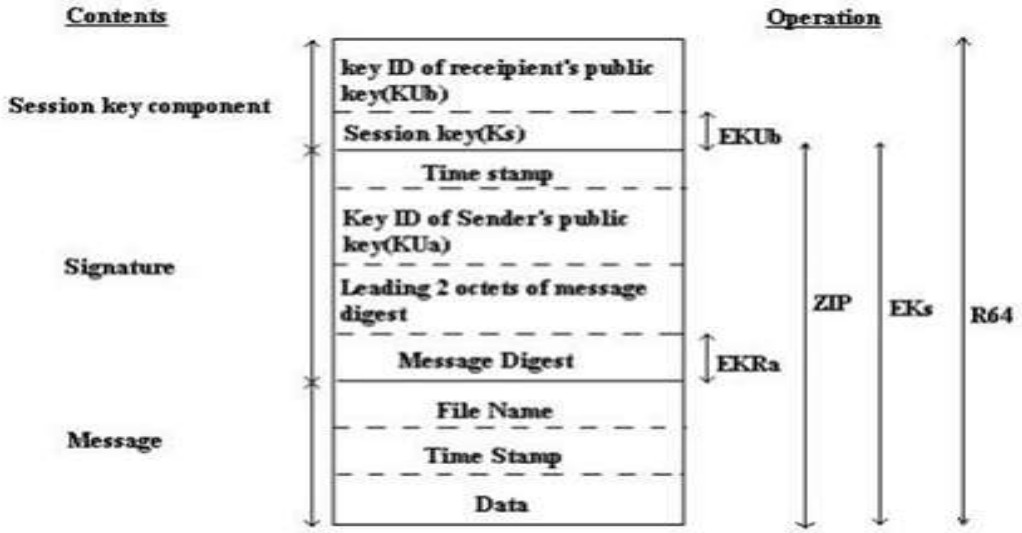
use selectors to filter outgoing traffic to map

based on: local & remote IP addresses, next layer protocol, name, local & remote ports

34. Mention the fields of IPSec authentication header? [C05-L3]



35. Sketch the general format for PGP message? [C05-L1]



36.What are the protocols used to provide IP security? [C05-L1]

Authentication header (AH) protocol.

Encapsulating Security Payload (ESP) protocol.

37.What is IPv6? [C05-L1]

Internet Protocol version 6 (**IPv6**) is the latest revision of the Internet Protocol (IP), the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet. IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion

38.What are the salient features of IPV6? [C05-L1]

New Packet Format and Header
Large Address Space
State full and Stateless IPv6 address
Multicast
Integrated

PART- B

1. Explain about the Email security or Discuss the threats faced by an e-mail and explain its security requirements to provide a secure email service Aor security service for E-mail [C05-L2-Dec-14]

Email security

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. Both are examined in this chapter andDomainKeys Identified Mail.

Pretty Good Privacy
- Notation
- Operational Description

S/MIME
- RFC 5322

Multipurpose Internet Mail Extensions
- S/MIME Functionality
- S/MIME Messages
- S/MIME Certificate Processing
- Enhanced Security Services

2. Explain pretty good privacy in detail or for what purpose Zimmerman developed PGP? Brief the various services provide a secure. [C05-L2- May -14]

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, Zimmermann has done the following:

1. Selected the best available cryptographic algorithms as building blocks.
2. Integrated these algorithms into a general-purpose application that is independent of operating system and processor and that is based on a small set of easy-to-use commands.

3. Made the package and its documentation, including the source code, freely available via the Internet, bulletin boards, and commercial networks such as AOL (America On Line).

4.Entered into an agreement with a company (Viacrypt, now Network Associates) to provide a fully compatible, low-cost commercial version of PGP.

Characteristics of PGP or PGP has grown explosively and is now widely used. A number of reasons can be 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.

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

It was not developed by, nor is it controlled by, any governmental or standards organization.

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

The algorithms used are extremely secure

Notation

Most of the notation used in this chapter has been used before, but a few terms are new. It is perhaps best to summarize those at the beginning. The following symbols are used.

} = concatenation

Z = compression using ZIP algorithm

R64 = conversion to radix 64 ASCII format1

The PGP documentation often uses the term ***secret key*** to refer to a key paired with a public key in a public-key encryption scheme.

As was mentioned earlier, this practice risks confusion with a secret key used for symmetric encryption. Hence, we use the term ***private key*** instead.

Operational Description in PGP

The actual operation of PGP, as opposed to the management of keys, consists of four services**:**

Authentication, Confidentiality,

Confidentiality and Authentication,

E-mail Compatibility.

Authentication

Figure 19.1a illustrates the digital signature service provided by PGP. This is the digital signature scheme discussed in Chapter 13 and illustrated in Figure 13.2. The sequence is as follows.

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

The combination of **SHA-1 and RSA** provides an effective digital signature scheme. Because of the strength of RSA, the recipient is assured that only the possessor of the matching private key can generate the signature.

Because of the strength of SHA-1, the recipient is assured that no one else could generate a new message that matches the hash code and, hence, the signature of the original message. As an alternative, signatures can be generated using **DSS/SHA-1**.

Although signatures normally are found attached to the message or file that they sign, this is not always the case: Detached signatures are supported.

A detached signature may be stored and transmitted separately from the message it signs. This is useful in several contexts. A user may wish to maintain a separate signature log of all messages sent or received. A detached signature of an executable program can detect subsequent virus infection.

Finally, detached signatures can be used when more than one party must sign a document, such as a legal contract. Each person's signature is independent and therefore is applied only to the document. Otherwise, signatures would have to be nested, with the second signer signing both the document and the first signature, and so on.

Confidentiality

Another basic service provided by PGP is confidentiality, which is provided by encrypting messages to be transmitted or to be stored locally as files.

In both cases, the symmetric encryption algorithm CAST-128 may be used.

Alternatively, IDEA or 3DES may be used. The 64-bit cipher feedback (CFB) mode is used.

As always, one must address the problem of key distribution. In PGP, each symmetric key is used only once.

That is, a new key is generated as a random 128-bit number for each message. Thus, although this is referred to in the documentation as a session key, it is in reality a one-time key. Because it is to be used only once, the session key is bound to the message and transmitted with it.

To protect the key, it is encrypted with the receiver's public key. Figure 19.1b illustrates the sequence,

Confidentiality and Authentication

As Figure 19.1c illustrates, both services may be used for the same message. First, a signature is generated for the plaintextmessage and prepended to the message.

Then the plaintext message plus signatureis encrypted using CAST-128 (or IDEA or 3DES), and the session key is encrypted using RSA (or ElGamal).

This sequence is preferable to the opposite: encrypting the message and then generating a signature for the encrypted message.

It is generally more convenient to store a signature with a plaintext version of a message. Furthermore, for purposes of third-party verification, if the signature is performed first, a third party need not be concerned with the symmetric key when verifying the signature.

Compression

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage.

The placement of the compression algorithm, indicated by Z for compression and Z-1 for decompression in Figure 19.1, is critical.

1. The signature is generated before compression for two reasons:

a. It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification. If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required.

b. Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and, as a result, produce different compressed forms. However, these different compression algorithms are interoperable because any version of the algorithm can correctly decompress the output of any other version. Applying the hash function and signature after compression would constrain all PGP implementations to 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 the original plaintext, cryptanalysis is more difficult.

E-mail Compatibility

When PGP is used, at least part of the block to be transmitted is encrypted. If only the signature service is used, then the message digest is encrypted (with the sender's private key). If the confidentiality service is used, the message plus signature (if present) are encrypted (with a one-time symmetric key).

3.Explain Secure / Multipurpose Internet Mail Extension (S/MIME). [C05-L2]

S/MIME

Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.

Although both PGP and S/MIME are on an IETF standards track, it appears likely that S/MIME will emerge as the industry standard for commercial and organizational use, while PGP will remain the choice for personal e-mail security for many users. S/MIME is defined in a number of documents—most importantly RFCs 3370, 3850, 3851, and 3852.

RFC 5322

RFC 5322 defines a format for text messages that are sent using electronic mail. It has been the standard for Internet-based text mail messages and remains in common use.

In the RFC 5322 context, messages are viewed as having an envelope and contents. The envelope contains whatever information is needed to accomplish transmission and delivery. The contents compose the object to be delivered to the recipient.

The RFC 5322 standard applies only to the contents. However, the content standard includes a set of header fields that may be used by the mail system to create the envelope, and the standard is intended to facilitate the acquisition of such information by programs.

The overall structure of a message that conforms to RFC 5322 is very simple. A message consists of some number of header lines **(*the header*)** followed by unrestricted text **(*the body*).**

The header is separated from the body by a blank line. Put differently, a message is ASCII text, and all lines up to the first blank line are assumed to be header lines used by the user agent part of the mail system.

A header line usually consists of a keyword, followed by a colon, followed by the keyword's arguments; the format allows a long line to be broken up into several lines. The most frequently used keywords are *From*, *To*, *Subject*, and *Date*.

Multipurpose Internet Mail Extensions

Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP), defined in RFC 821, or some other mail transfer protocol and RFC 5322 for electronic mail. [PARZ06]

Lists the following limitations of the SMTP/5322 scheme.

1. SMTP cannot transmit executable files or other binary objects. A number of schemes are in use for converting binary files into a text form that can be used by SMTP mail systems, including the popular UNIX UUencode/ UUdecode scheme. However, none of these is a standard or even a *de facto* standard.

2. SMTP cannot transmit text data that includes national language characters, because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.

3. SMTP servers may reject mail message over a certain size.

4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.

5. SMTP gateways to X.400 electronic mail networks cannot handle nontextual

6. Some SMTP implementations do not adhere completely to the SMTP

standards defined in RFC 821. Common problems include:

- Deletion, addition, or reordering of carriage return and linefeed
- Truncating or wrapping lines longer than 76 characters
- Removal of trailing white space (tab and space characters)
- Padding of lines in a message to the same length
- Conversion of tab characters into multiple

Overview the MIME specification includes the following elements.

7. Five new message header fields are defined, which may be included in an RFC 5322 header. These fields provide information about the body of the message.

8. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.

9. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

The five header fields defined in MIME are

MIME-Version: Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.

Content-Type: Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner.

S/MIME Messages

The general procedures for S/MIME message preparation

Securing a MIME Entity

S/MIME secures a MIME entity with a signature, encryption, or both.

A MIME entity may be an entire message (except for the RFC 5322 headers), or if the MIME content type is multipart, then a MIME entity is one or more of the subparts of the message. The MIME entity is prepared according to the normal rules for MIME message preparation. Then the MIME entity plus some security-related data, such as algorithm identifiers and certificates, are processed by S/MIME to produce what is known as a PKCS object.

A PKCS object is then treated as message content and wrapped in MIME (provided with appropriate MIME headers).

The message to be sent is converted to canonical form. In particular, for a given type and subtype, the appropriate canonical form is used for the message content. For a multipart message, the appropriate canonical form is used for each subpart.

1. Enveloped Data

The steps for preparing an envelopedData MIME entity are

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm (RC2/40 or triple DES).
2. For each recipient, encrypt the session key with the recipient's public RSA key.
3. For each recipient, prepare a block known as RecipientInfo that contains

an identifier of the recipient's public-key certificate,2 an identifier of the algorithm used to encrypt the session key, and the encrypted session key.

4. Encrypt the message content with the session key.

4.Explain Overview of IP Security or Draw and explain Ip security. [C05-L2]

IP Security Overview

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.

Applications of IPSec

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include the following
Secure branch office connectivity over the Internet:
A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an Internet service provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.

Establishing extranet and intranet connectivity with partners: IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.

Enhancing electronic commerce security: Even though some Web and electronic commerce applications have built-in security protocols, the use of IPSec enhances that security.

The principal feature of IPSec that enables it to support these varied applications is that it can encrypt and/or authenticate *all* traffic at the IP level.

Thus, all distributed applications, including remote logon, client/server, e-mail, file transfer, Web access, and so on, can be secured.

Figure 20.1 is a typical scenario of IPsec usage. An organization maintains LANs at dispersed locations. Nonsecure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPsec protocols are used.

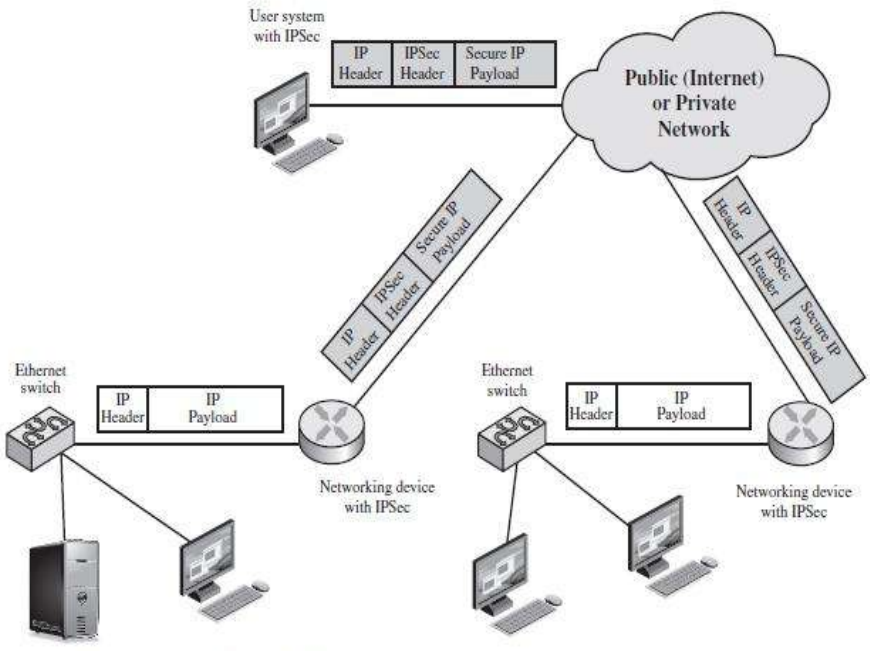


Figure 20.1 An IP Security Scenario

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.

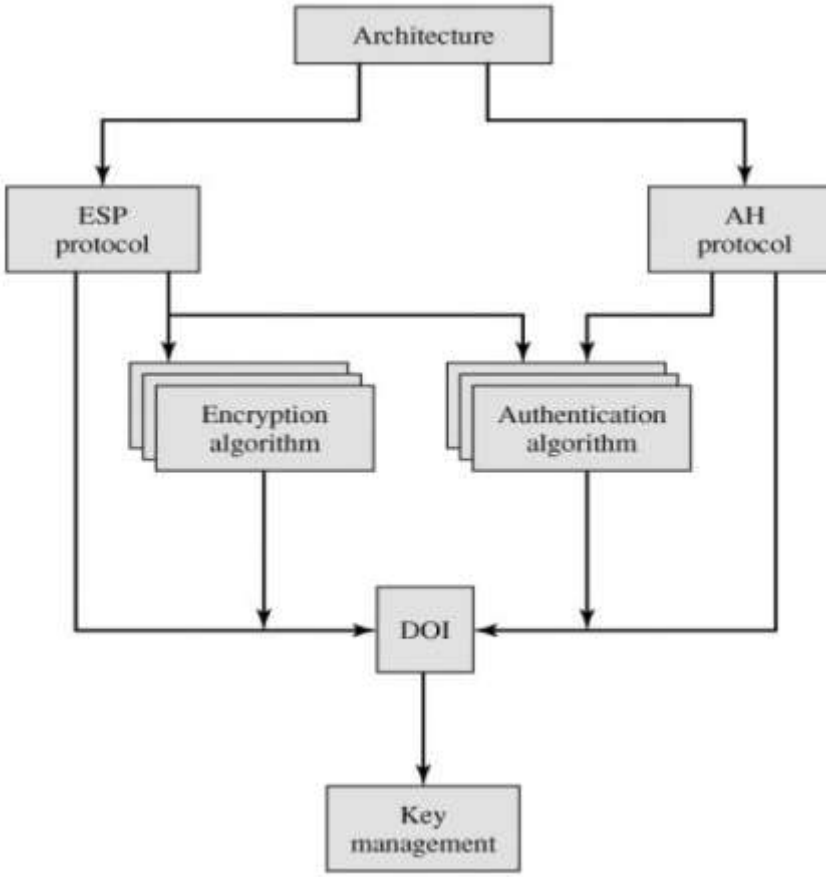


Figure 16.2. IPSec Document Overview

5. Explain the Authentication header (AH). [C05-L2]

Authentication header (AH)

The Authentication Header provides support for data integrity and authentication of IP packets. The data integrity feature ensures that undetected modification to a packet's content in transit is not possible.

The authentication feature enables an end system or network device to authenticate the user or application and filter traffic accordingly; it also prevents the address spoofing attacks

Authentication is based on the use of a message authentication code (MAC), protocol hence the two parties must share a secret key.

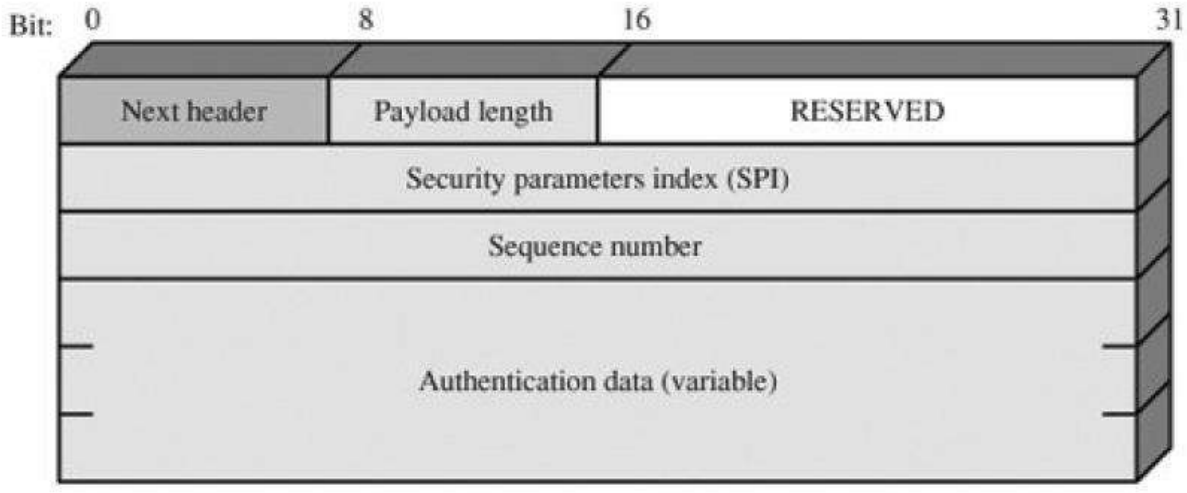
The Authentication Header consists of the following fields (Figure 16.3):

Next Header (8 bits): Identifies the type of header immediately following this header.
Payload Length (8 bits): Length of Authentication Header in 32-bit words,

Example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header, and the Payload Length field has a value of 4.
Reserved (16 bits): For future use.

Security Parameters Index (32 bits): Identifies a security association. Sequence Number (32 bits): A monotonically increasing counter value.

Authentication Data (variable): A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value (ICV), or MAC, for this packet.



Anti-Replay Service

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits

It to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The Sequence Number field is designed to thwart such attacks. First, we discuss sequence number generation by the sender, and then we look at how it is processed by the recipient.

When a new SA is established, the sender initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1. If anti-replay is enabled (the default), the sender must not allow the sequence number to cycle past 232 1 back to zero. Otherwise, there would be multiple valid packets with the same sequence number. If the limit of 232 1 is reached, the sender should terminate this SA and negotiate a new SA with a new key. Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets will be delivered.

Therefore, the IPSec authentication document dictates that the receiver should implement a window of size *W*, with a default of *W* = 64. The right edge of the window represents the highest sequence number, *N*, so far received for a valid packet. For any packet with a sequence number in the range from *N W* + 1 to *N*

1.If the received packet falls within the window and is new, the MAC is checked.

If the packet is authenticated, the corresponding slot in the window is marked.

2. If the received packet is to the right of the window and is new, the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.

3. If the received packet is to the left of the window, or if authentication fails, the packet is discarded; this is an auditable event.

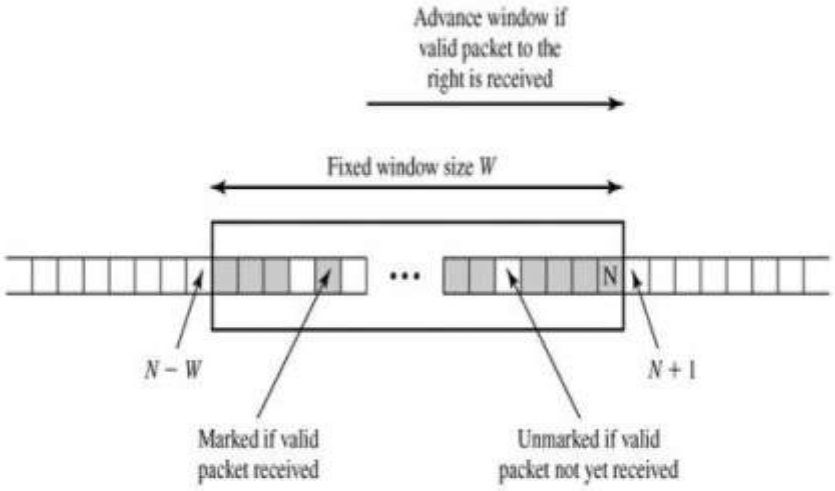


Figure 16.4. Antireplay Mechanism

Integrity Check Value

The Authentication Data field holds a value referred to as the Integrity Check Value. The ICV is a message authentication code or a truncated version of a code produced by a MAC algorithm. The current specification dictates that a compliant implementation must support

HMAC-MD5-96
HMAC-SHA-1-96

Both of these use the HMAC algorithm, the first with the MD5 hash code and the second with the SHA-1 hash code

In both cases, the full HMAC value is calculated but then truncated by using the first 96 bits, which is the default length for the Authentication Data field.

Transport and Tunnel Modes

Figure 16.5 shows two ways in which the IPSec authentication service can be used. In one case, authentication is provided directly between a server and client workstations; the workstation can be either on the same network as the server or on an external network.

As long as the workstation and the server share a protected secret key, the authentication process is secure. This case uses a transport mode SA.

In the other case, a remote workstation authenticates itself to the corporate firewall, either for access to the entire internal network or because the requested server does not support the authentication feature. This case uses a tunnel mode SA.

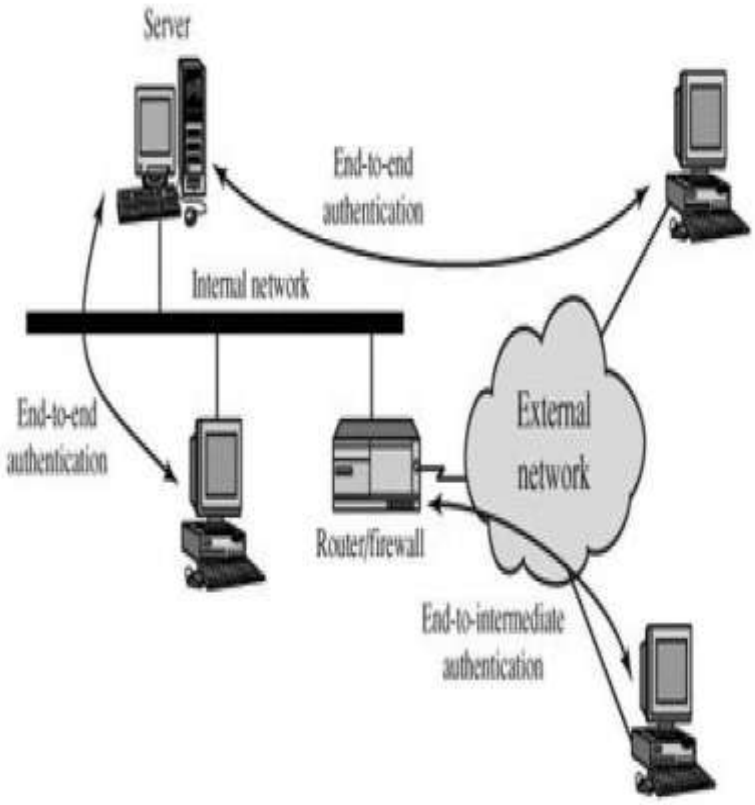


Figure 16.5. End-to-End versus End-to-Intermediate Authentication

6.Write short notes : Web security [CO5-L2-DEC13]

Web Security

Web Security Considerations

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.

As such, the security tools and approaches discussed so far in this book are relevant to the issue of Web security.

But, as pointed out in [GARF97], the Web 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 tele text, 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.

Web Security Threats

Table 17.1 provides a summary of the types of security threats faced in using the Web. 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.

Web Traffic Security Approaches

A number of approaches to providing Web security are possible. The various approaches 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 their scope of applicability and their relative location within the TCP/IP protocol stack.

The advantage of using IPSec is that it is transparent to end users and applications and provides a general-purpose solution. Further, IPSec includes a filtering capability so that only selected traffic need incur the overhead of IPSec processing.

7.Explain Secure Socket Layer Security(SSL) in Detail [CO5-L2- May-15]

Secure Socket Layer

SSL protocol is an internet protocol for secure exchange of information between a web browser and web server

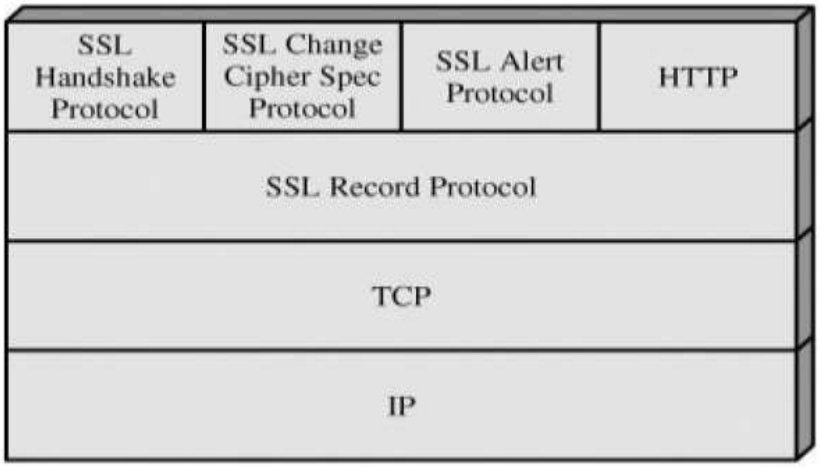
SSL provides security services between TCP and applications that use TCP. The SSL protocol is an internet protocol for secure exchange of information between a web browser and web server

Subsequently, when a consensus was reached to submit the protocol for Internet standardization, the TLS working group was formed within IETF to develop a common standard. This first published version of TLS can be viewed as essentially an SSLv3.1 and is very close to and backward compatible with SSLv3.

The bulk of this section is devoted to a discussion of SSLv3. At the end of the section, the principal differences between SSLv3 and TLS are described.

SSL Architecture

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



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.

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections:

Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.

Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

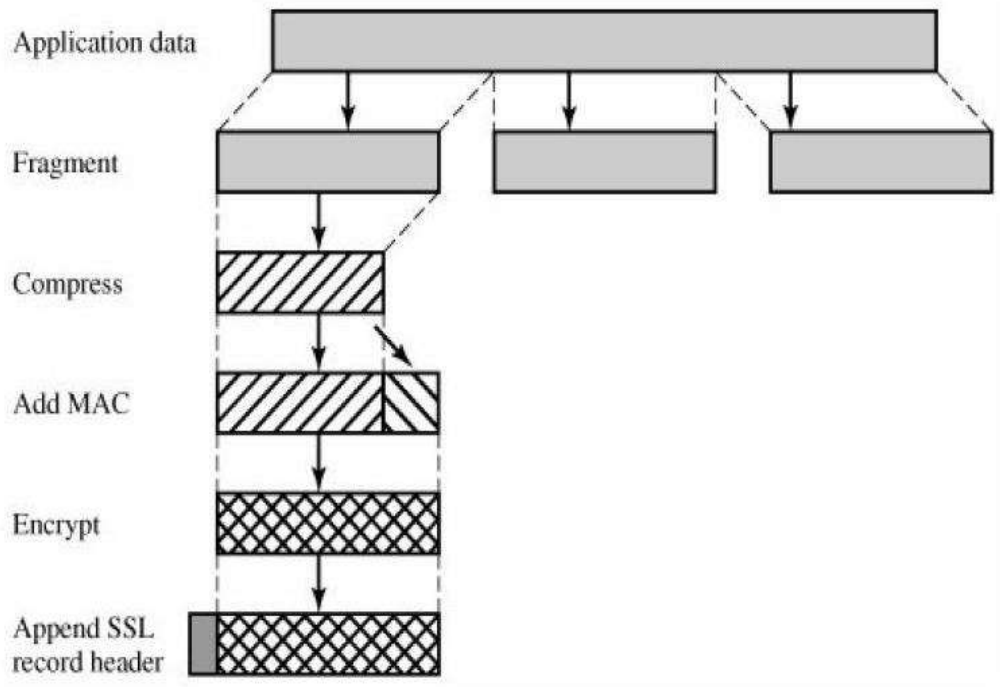


Figure 17.3 indicates the overall operation of the SSL Record Protocol. The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment.

Received data are decrypted, verified, decompressed, and reassembled and then delivered to higher-level users.

The first step is fragmentation. Each upper-layer message is fragmented into blocks of 214 bytes (16384 bytes) or less.

Next, compression is optionally applied. Compression must be lossless and may not increase the content length by more than 1024 bytes. In SSLv3 (as well as the current version of TLS), no compression algorithm is specified, so the default compression algorithm is null.

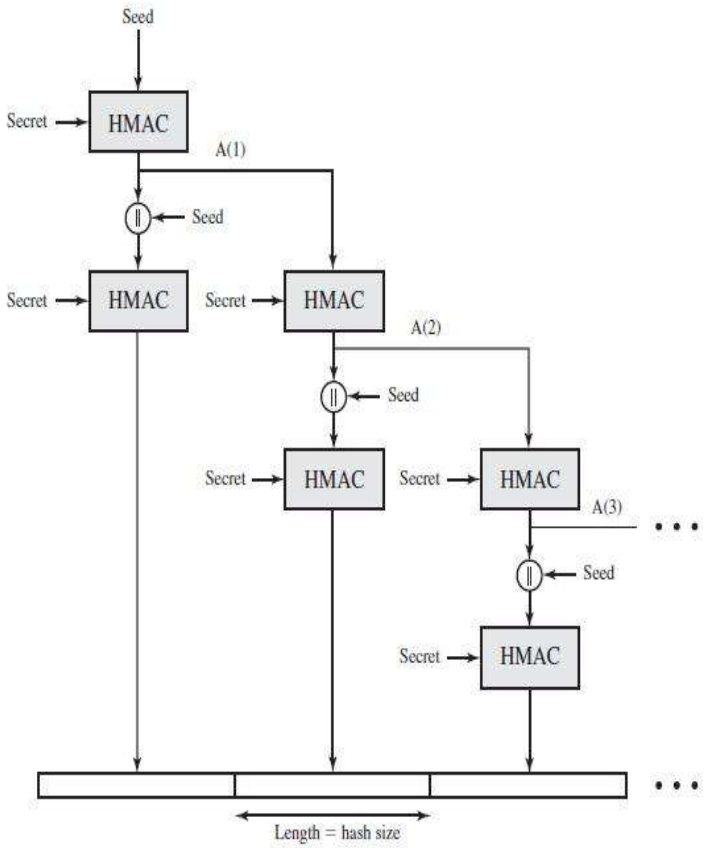


Figure 16.7 TLS Function P_hash(secret, seed)

8.Explain in detail about SET(Secure Electronic Transaction)? (or)

List out the participants of SET system, and explain in detail. [CO5-L2]

SET:

SET is an open encryption and security specification designed to protect credit card transactions on the Internet. The current version, SETv1, emerged from a call for security standards by MasterCard and Visa in February 1996.

A wide range of companies were involved in developing the initial specification, including IBM, Microsoft, Netscape, RSA, Terisa, and Verisign. Beginning in 1996, there have been numerous tests of the concept, and by 1998 the first wave of SET-compliant products was available.

SET is not itself a payment system. Rather it is a set of security protocols and formats that enables users to employ the existing credit card payment infrastructure on an open network, such as the Internet, in a secure fashion. In essence, SET provides three services:

Provides a secure communications channel among all parties involved in a transaction Provides trust by the use of X.509v3 digital certificates Ensures privacy because the information is only available to parties in a transaction when and where necessary

SET Overview

Confidentiality: all messages encrypted

Trust: all parties must have digital certificates

Privacy: information made available only when and where necessary

Key Features of SET

To meet the requirements just outlined, SET incorporates the following features:

● **Confidentiality of information:** Cardholder account and payment information is secured as it travels across the network. An interesting and important feature of SET is that it prevents the merchant from learning the cardholder's credit card number; this is only provided to the issuing bank. Conventional encryption by DES is used to provide confidentiality.

● **Integrity of data:** Payment information sent from cardholders to merchants includes order

information, personal data, and payment instructions. SET guarantees that these message contents are not altered in transit. RSA digital signatures, using SHA-1 hash codes, provide message integrity. Certain messages are also protected by HMAC using SHA-1.

Cardholder account authentication: SET enables merchants to verify that a cardholder is a legitimate user of a valid card account number. SET uses X.509v3 digital certificates with RSA signatures for this purpose.

● **Merchant authentication:** SET enables cardholders to verify that a merchant has a relationship with a financial institution allowing it to accept pay

SET Participants

● **Cardholder:** In the electronic environment, consumers and corporate purchasers interact with merchants from personal computers over the Internet. A cardholder is an authorized holder of a payment card (e.g., MasterCard, Visa) that has been issued by an issuer.

● **Merchant:** A merchant is a person or organization that has goods or services to sell to the cardholder. Typically, these goods and services are offered via a Web site or by electronic mail. A merchant that accepts payment cards must have a relationship with an acquirer.

● **Issuer**: This is a financial institution, such as a bank, that provides the cardholder with the payment card. Typically, accounts are applied for and opened by mail or in person.

● **Acquirer**:. The acquirer provides authorization to the merchant that a given card account is active and that the proposed purchase does not exceed the credit limit. The acquirer also provides electronic transfer of payments to the merchant's account.

● **Payment gateway**: This is a function operated by the acquirer or a designated third party that processes merchant payment messages. The payment gateway interfaces between SET and the existing bankcard payment networks for authorization and payment functions.

● **Certification authority (CA):** This is an entity that is trusted to issue X.509v3 public-key certificates for cardholders, merchants, and payment gateways. The success of SET will depend on the existence of a CA infrastructure available for this purpose.

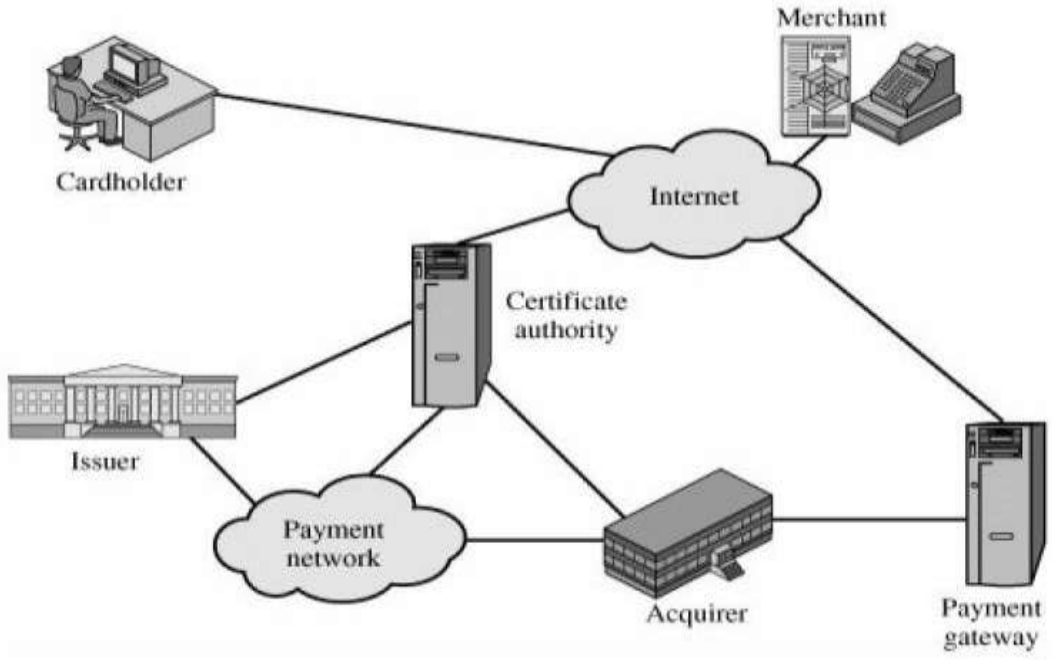


Figure 17.8. Secure Electronic Commerce Components

SET TransactionThe customer opens an account. The customer obtains a credit card account, such as MasterCard or Visa, with a bank that supports electronic payment and SET.

The customer receives a certificate. After suitable verification of identity, the customer receives an X.509v3 digital certificate, which is signed by the bank. The certificate verifies the customer's RSA public key and its expiration date. It also establishes a relationship, guaranteed by the bank, between the customer's key pair and his or her credit card.

Merchants have their own certificates. A merchant who accepts a certain brand of card must be in possession of two certificates for two public keys owned by the

merchant: one for signing messages, and one for key exchange. The merchant also needs a copy of the payment gateway's public-key certificate.

The customer places an order. This is a process that may involve the customer first browsing through the merchant's Web site to select items and determine the price. The customer then sends a list of the items to be purchased to the merchant, who returns an order form containing the list of items, their price, a total price, and an order number.
The merchant is verified. In addition to the order form, the merchant sends a copy of its certificate, so that the customer can verify that he or she is dealing with a valid store.

The order and payment are sent. The customer sends both order and payment information to the merchant, along with the customer's certificate. The order confirms the purchase of the items in the order form. The payment contains credit card details. The payment information is encrypted in such a way that it cannot be read by the merchant.

The customer's certificate enables the merchant to verify the customer.

The merchant requests payment authorization. The merchant sends the payment information to the payment gateway, requesting authorization that the customer's available credit is sufficient for this purchase.

The merchant confirms the order. The merchant sends confirmation of the order to the customer.

The merchant provides the goods or service. The merchant ships the goods or provides the service to the customer.

The merchant requests payment. This request is sent to the payment gateway, which handles all of the payment processing.

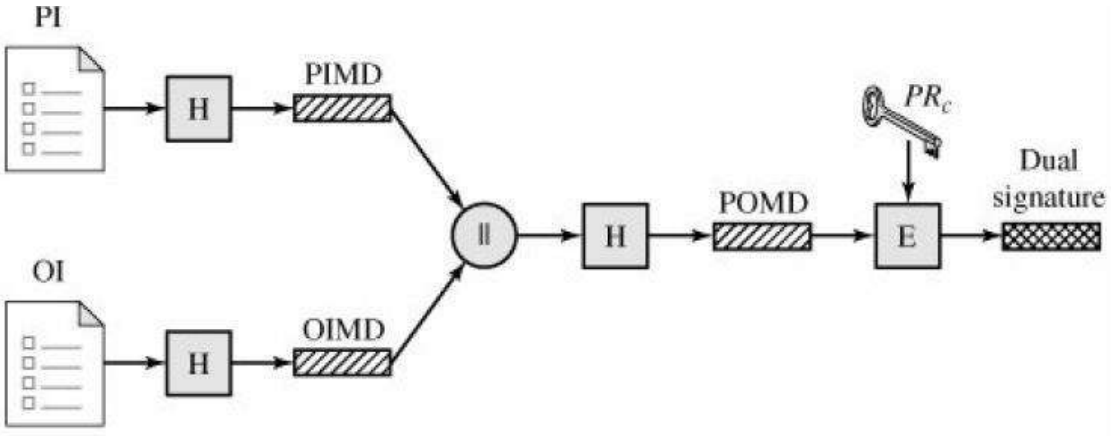
Dual Signature

- customer creates dual messages
- order information (OI) for merchant

- payment information (PI) for bank
- neither party needs details of other
- but must know they are linked
- use a dual signature for this
- signed concatenated hashes of OI & PI

where *PUc* is the customer's public signature key. If these two quantities are equal, then the merchant has verified the signature. Similarly, if the bank is in possession of DS, PI, the message digest for OI (OIMD), and the customer's public key, then the bank can compute

H(H[OI]||OIMD); D(*PUc*, DS)



Again, if these two quantities are equal, then the bank has verified the signature. In summary,

Payment Processing

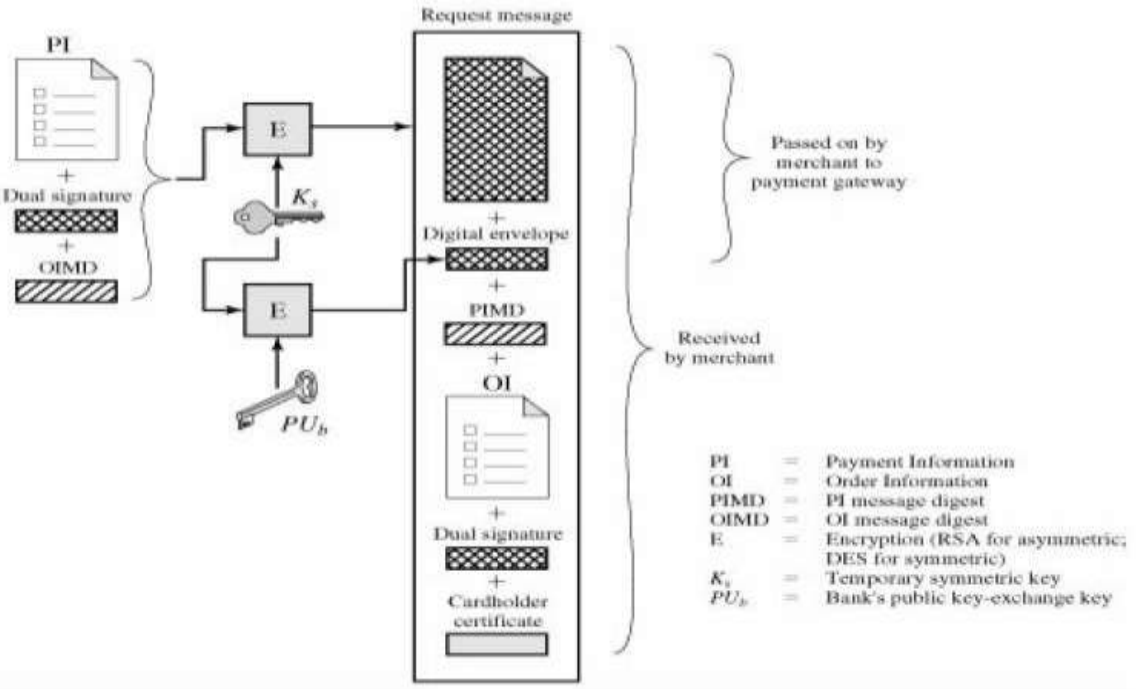
Table 17.3 lists the transaction types supported by SET. In what follows we look in some detail at the following transactions:

Purchase request

Payment authorization

Payment capture

PurchaseRequest –Customer



Payment Gateway Authorization

1. Verifies all certificates
2. Decrypts digital envelope of authorization block to obtain symmetric key & then decrypts authorization block
3. Verifies merchant's signature on authorization block