SUBJECT CODE : 3161606

As per New Syllabus

GUJARAT TECHNOLOGICAL UNIVERSITY

Semester - VI (CSE / IT)

CRYPTOGRAPHY AND NETWORK SECURITY

Vilas S. Bagad

M.E. (E&Tc), Microwaves,
M.M.S. (Information systems),
Faculty, Institute of Telecommunication Management,
Ex-Faculty Sinhgad College of Engineering,
Pune

Iresh A. Dhotre

M.E. (Information Technology)
Ex-Faculty, Sinhgad College of Engineering,
Pune.

TECHNICAL PUBLICATIONS
SINCE 1993
An Up-Thrust for Knowledge

Cryptography and Network Security

Subject Code : 3161606

Semester - VI (Computer Science and Engineering / Information Technology)

Published by :

TECHNICAL PUBLICATIONS
SINCE 1993
An Up-Thrust for Knowledge

Amit Residency, Office No.1, 412, Shaniwar Peth,
Pune - 411030, M.S. INDIA Ph.: +91-020-24495496/97
Email : sales@technicalpublications.org Website : www.technicalpublications.org

Printer :
Yogiraj Printers & Binders
Sr. No. 10/1A,
Ghule Industrial Estate, Nanded Village Road,
Tal. - Haveli, Dist. - Pune - 411041.

ISBN 978-93-90450-86-2

9 789390 450862

Course 18

PREFACE

The importance of **Cryptography and Network Security** is well known in various engineering fields. Overwhelming response to our books on various subjects inspired us to write this book. The book is structured to cover the key aspects of the subject **Cryptography and Network Security.**

The book uses plain, lucid language to explain fundamentals of this subject. The book provides logical method of explaining various complicated concepts and stepwise methods to explain the important topics. Each chapter is well supported with necessary illustrations, practical examples and solved problems. All the chapters in the book are arranged in a proper sequence that permits each topic to build upon earlier studies. All care has been taken to make students comfortable in understanding the basic concepts of the subject.

Representative questions have been added at the end of each section to help the students in picking important points from that section.

The book not only covers the entire scope of the subject but explains the philosophy of the subject. This makes the understanding of this subject more clear and makes it more interesting. The book will be very useful not only to the students but also to the subject teachers. The students have to omit nothing and possibly have to cover nothing more.

We wish to express our profound thanks to all those who helped in making this book a reality. Much needed moral support and encouragement is provided on numerous occasions by our whole family. We wish to thank the **Publisher** and the entire team of **Technical Publications** who have taken immense pain to get this book in time with quality printing.

Any suggestion for the improvement of the book will be acknowledged and well appreciated.

Authors
V. S. Bagad
I. A. Dhotre

Dedicated to God

SYLLABUS

Cryptography and Network Security - (3161606)

Credits	Examination Marks				Total Marks
C	Theory Marks		Practical Marks		
	ESE (E)	PA(M)	ESE (V)	PA (I)	
4	70	30	30	20	150

1. Introduction - Security services, security services, security mechanisms Finite fields - group, ring, fields, modular arithmetic, The Euclidean algorithm. Symmetric Cipher Model, Cryptography, Cryptanalysis and Attacks; Substitution and Transposition techniques. **(Chapter - 1)**

2. Stream ciphers and block ciphers, Block Cipher structure, Data Encryption standard (DES) with example, strength of DES, Design principles of block cipher, AES with structure, its transformation functions, key expansion, example and implementation. **(Chapter - 2)**

3. Multiple encryption and triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback mode, Output Feedback mode, Counter mode. **(Chapter - 3)**

4. Public Key Cryptosystems with Applications, Requirements and Cryptanalysis, RSA algorithm, its computational aspects and security, Diffie-Hillman Key Exchange algorithm, Man-in-Middle attack. **(Chapter - 4)**

5. Cryptographic Hash Functions, their applications, Simple hash functions, its requirements and security, Hash functions based on Cipher Block Chaining, Secure Hash Algorithm (SHA). **(Chapter - 5)**

6. Message Authentication Codes, its requirements and security, MACs based on Hash Functions, Macs based on Block Ciphers. **(Chapter - 6)**

7. Digital Signature, its properties, requirements and security, various digital signature schemes (Elgamal and Schnorr), NIST digital Signature algorithm. **(Chapter - 7)**

8. Key management and distribution, symmetric key distribution using symmetric and asymmetric encryptions, distribution of public keys, X.509 certificates, Public key infrastructure. **(Chapter - 8)**

9. Remote user authentication with symmetric and asymmetric encryption, Kerberos. **(Chapter - 9)**

10. Web Security threats and approaches, SSL architecture and protocol, Transport layer security, HTTPS and SSH. **(Chapter - 10)**

Table of Contents

Chapter - 1 Introduction (1 - 1) to (1 - 54)

Chapter - 5 Cryptographic Hash Functions (5 - 1) to (5 - 24)

Chapter - 6 Message Authentication Codes (6 - 1) to (6 - 14)

1 Introduction

Syllabus

Security services, Security services, Security mechanisms, Finite fields - group, ring, fields, modular arithmetic, The Euclidean algorithm. Symmetric Cipher Model, Cryptography, Cryptanalysis and Attacks; Substitution and Transposition techniques.

Contents

1.1 Introduction of Security

GTU : Summer-18, Winter-19

- How to protect the valuable assets ? It is necessary to keep in safe place like a bank to protect the valuable assets. But bank is not a safe place now a day. There are so many example where bank robbery in our country.
- Bank robbery is the crime of stealing from a bank during opening hours. Protecting assets was difficult and not always effective.
- Now a day, protection is easier because many factors working against the potential criminal. Very sophisticated alarm and camera systems silently protect secure places like banks.
- Traditionally information security provided by physical i.e. rugged filing cabinets with locks and administrative mechanisms i.e. personnel screening procedures during hiring process.
- Asset protection systems are designed to recover stolen cash and high value assets, apprehend criminals and deter crime. The system has the capacity to track, protect and manage critical assets in real-time.
- The techniques of criminal investigation have become so effective that a person can be identified by genetic material, voice, retinal pattern, fingerprints etc.
- Use of networks and communications links requires measures to protect data during transmission.
- **Data security** is the science and study of methods of protecting data from unauthorized disclosure and modification.
- Data and information security is about enabling collaboration while managing risk with an approach that balances availability versus the confidentiality of data.
- **Computer security :** Generic name for the collection of tools designed to protect data and to thwart hackers.
- **Network security :** Measures to protect data during their transmission.
- **Internet security :** Measures to protect data during their transmission over a collection of interconnected networks.

Protecting valuables

- Following are certain aspects for the need of security :

1. Increasing threat of attacks.
2. Fast growth of computer networking for information sharing.
3. Availability of number of tools and resources on Internet.
4. Lack of specialized resources that may be allotted for securing system.

1.1.1 Need of Security

- Security is required because the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means.
- Network security measures are needed to protect data during their transmission.
- Following are the examples of security violations.

1. User A transmits a sensitive information file to user B. The unauthorised user C is able to monitor the transmission and capture a copy of the file during its transmission.
2. A message is sent from a customer to a stockbroker with instructions for various transactions. Subsequently, the investments lose value and the customer denies sending the message.
3. While transmitting the message between two users, the unauthorised user intercepts the message, alters its contents to add or delete entries, and then forwards the message to destination user.

1.1.2 Terminology

- Basic terminology used for security purposes are as follows :

a. **Cryptography :** The art or science encompassing the principles and methods of transforming an plaintext message into one that is unintelligible and then retransforming that message back to its original form.

b. **Plaintext :** The original message.

c. **Ciphertext :** The transformed message produced as output, It depends on the plaintext and key.

d. **Cipher :** An algorithm for transforming plaintext message into one that is unintelligible by transposition and/or substitution methods.

e. **Key :** Some critical information used by the cipher, known only to the sender and receiver.

f. **Encipher (encode) :** The process of converting plaintext to ciphertext using a cipher and a key.

g. **Decipher (decode) :** The process of converting ciphertext back into plaintext using a cipher and a key.

h. **Cryptanalysis :** The study of principles and methods of transforming an unintelligible message back into an intelligible message *without* knowledge of the key. Also called **code-breaking.** Cryptanalysis is to break an encryption. Cryptanalyst can do any or all of the three different things :

1. Attempt to break a single message.
2. Attempt to recognize patterns in encrypted messages, in order to be able to break subsequent ones by applying a strainghtforward decryption algorithm.
3. Attempt to find general weakness in an encryption algorithm, without necessarily having intercepted any messages.

i. **Cryptology :** Both cryptography and cryptanalysis.

j. **Code :** An algorithm for transforming an plaintext message into an unintelligible one using a code-book.

1.1.3 Security Goals

- Security goals are as follows :

1. Confidentially
2. Integrity
3. Availability

1. Confidentiality

- Confidentiality refers to limiting information access and disclosure to authorized users and preventing access by or disclosure to unauthorized ones.
- Sensitive information should be kept secret from individuals who are not authorized to see the information.
- Underpinning the goal of confidentiality are authentication methods like user-IDs and passwords that uniquely identify a data system's users, and supporting control methods that limit each identified user's access to the data system's resources.
- Confidentiality is not only applied to storage of data but also applies to the transmission of information.
- Confidentiality means that people cannot read sensitive information, either while it is on a computer or while it is traveling across a network.
- Fig. 1.1.1 Relationship between Confidentiality Integrity and Availability.

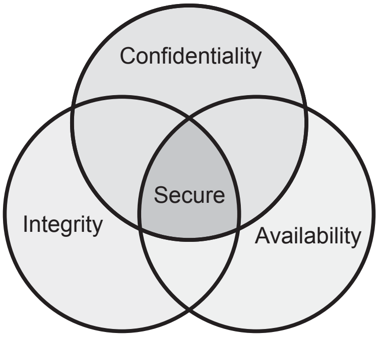


Fig. 1.1.1 Relationship between confidentiality integrity and availability

2. Integrity

- Integrity refers to the trustworthiness of information resources.
- Integrity should not be altered without detection.
- It includes the concept of "data integrity" namely, that data have not been changed inappropriately, whether by accident or deliberately malign activity.
- It also includes "origin" or "source integrity" that is, that the data actually came from the person or entity you think it did, rather than an imposter.
- Integrity ensures that information is not changed or altered in transit. Under certain attack models, an adversary may not have to power to impersonate an authenticated party or understand a confidential communication, but may have the ability to change the information being transmitted.
- On a more restrictive view, however, integrity of an information system includes only preservation without corruption of whatever was transmitted or entered into the system, right or wrong.

3. Availability

- Availability refers, to the availability of information resources. An information system that is not available when you need it is at least as bad as none at all.
- Availability means that people who are authorized to use information are not prevented from doing so. It may be much worse, depending on how reliant the organization has become on a functioning computer and communications infrastructure.
- Almost all modern organizations are highly dependent on functioning information systems. Many literally could not operate without them.
- Availability, like other aspects of security, may be affected by purely technical issues (e.g. a malfunctioning part of a computer or communications device), natural phenomena (e.g. wind or water), or human causes (accidental or deliberate).
- For example, an object or service is thought to be available if

  i. It is present in a usable form.

  ii. It has capacity enough to meet the services needs.

  iii. The service is completed an acceptable period of time.
- By combining these goals, we can construct the availability. The data item, service or system is available if

  i. There is a timely response to our request.

  ii. The service and system can be used easily.

  iii. Concurrency is controlled.

iv. It follows the fault tolerance.

v. Resources are allocated fairly.

University Questions

1. *Explain data confidentiality, data authentication and data integrity.*

   GTU : Summer-18, Marks 3

2. *Define following principles of security :*

   1. Confidentiality 2. Integrity 3. Availability

   GTU : Winter-19, Marks 3

1.2 OSI Security Architecture

- X.800 recommends sending architecture for OSI. The OSI security architecture define systematic way to asses security needs of on organization and help them to choose various security products and fields.
- The OSI security architecture mainly focuses on :

a) Security attack :

- ○ Any action which comprises the organizaion secured information.

b) Security mechanism :

- ○ A process designed to detect, prevent receiver from a security attack.

c) Security service :

- ○ The security service are intended to counter security attack by making use of the one or more security mechanism.

1.3 Security Services

- X.800 defines a security service as a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers.
- X.800 divides security services into five different categories.

  1. Authentication 2. Access control 3. Data confidentiality

  4. Data integrity 5. Nonrepudiation

1. Authentication

- Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. In public and private computer network, authentication is commonly done through the use of login passwords.
- Two specific authentication services are defined in X.800 :

  a. Peer entity authentication

b. Data origin authentication

- **Peer entity authentication** used in association with a logical connection to provide confidence in the identity of the entities connected.
- Data origin outhentication enables the recepient to verify that the message have not been tempered in transit (data integrity) and they originally from expected sender (authenticity).
- **Data origin authentication** does not provide protection against the duplication or modification of data units. This type of service supports applications like electronic mail where there are no prior interactions between the communicating entities.

2. Access control

- It is the ability to limit and control the access to host systems and applications via communications links.
- This service controls who can have access to a resource.

3. Data confidentiality

- Confidentiality is the concealment of information or resources. It is the protection of transmitted data from passive attacks.
- Confidentiality is classified into
  1. **Connection confidentiality :** The protection of all user data on a connection.
  2. **Connectionless confidentiality :** The protection of all user data in a single data block.
  3. **Selective field confidentiality :** The confidentiality of selected fields within the user data on a connection or in a single data block.
  4. **Traffic flow confidentiality :** The protection of the information that might be derived from observation of traffic flows.

4. Data integrity

- Integrity can apply to a stream of messages a single message or selected fields within a message.
- Modification causes loss of message integrity.
- Data integrity can be classified as
  1. Connection integrity with recovery
  2. Connection integrity without recovery
  3. Selective field connection integrity
  4. Connectionless integrity
  5. Selective field connectionless integrity

- 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 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.
- Connectionless integrity provides for the integrity of a single connectionless data block and may take the form of detection of data modification.

5. Nonrepudiation

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

1.4 Security Mechanism

- X.800 defined security mechanisms as follows

1. **Specific security mechanisms :** May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

a. **Encipherment :** The use of mathematical algorithms to transform data into a form that is not readily intelligible.

b. **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 the data unit and protect against forgery.

c. **Access control :** A variety of mechanisms that enforce access rights to resources.

d. **Data integrity :** A variety of mechanisms used to ensure the integrity of a data unit or stream of data units.

e. **Authentication exchange :** A mechanism intended to ensure the identity of an entity by means of information exchange.

f. **Traffic padding :** The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

g. **Notarization :** The use of a trusted third party to assure certain properties of a data exchange.

2. **Pervasive security mechanisms :** Mechanisms that are not specific to any particular OSI security service or protocol layer.

   a. **Trusted functionality :** That which is perceived to be correct with respect to some criteria.

   b. **Event detection :** Detection of security relevant events.

   c. **Security label :** The marking bound to resource that names or designates the security attributes of that resource

   d. **Security recovery :** Deals with requests from mechanisms, such as event handling and management functions and takes recovery actions.

1.5 Security Attacks

GTU : Summer-17, Winter-18, 19

- Computer based systems have three valuable components : **Hardware, software and data**.
- Securities of these components are evaluated in terms of vulnerability, threats, attacks and control.
- An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system.

Asset

- Asset means people, property and information.
- People may include employees and customers along with other invited persons such as contractors or guests.

Vulnerability

- Vulnerability refers to the security flaws in a system that allows an attack to be successful.
- Weaknesses or gaps in a security program that can be exploited by threats to gain unauthorized access to an asset. Vulnerability is a weakness or gap in our protection efforts.
- **Example :** In design, implementation or procedure, that might be exploited to cause loss or harm.

Threat

- Anything that can exploit vulnerability, intentionally or accidentally, and obtain, damage, or destroy an asset. A threat is what we're trying to protect against.
- Threat refers to the source and means of a particular type of attack.

- A threat assessment is performed to determine the best approaches to securing a system against a particular threat, or class of 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.
- Where risk assessments focus more on analyzing the potential and tendency of one's resources to fall prey to various attacks, threat assessments focus more on analyzing the attacker's resources.
- Analyzing threats can help one develop specific security policies to implement in line with policy priorities and understand the specific implementation needs for securing one's resources.
- Threats come in many forms, depending on their mode of attack. From viruses to trojans, spyware and bots, threats have evolved into sophisticated programs intended to harm computers.

Risk

- The potential for loss, damage or destruction of an asset as a result of a threat exploiting vulnerability. Risk is the intersection of assets, threats, and vulnerabilities.
- The formula used to determine risk is

  Risk = Asset + Threat + Vulnerability

  R = A + T + V
- Risk is a function of threats exploiting vulnerabilities to obtain damage or destroy assets. Thus, threats may exist, but if there are no vulnerabilities then there is little/no risk.
- Similarly, you can have vulnerability, but if you have no threat, then you have little/no risk.

Control

- Control is used as proactive measure. Control is a action, device, procedure, or technique that removes or reduces a vulnerability.
- A threat is blocked by control of vulnerability.
- Interception, interruption, modification and fabrication are the system security threats.

1.5.1 Passive Attack

- Passive attacks are those, wherein the attacker indulges in eavesdropping on, or monitoring of data transmission. A passive attack attempts to learn or make use of information from the system but does not affect system resources.
- The attacker aims to obtain information that is in transit. The term passive indicates that the attacker does not attempt to perform any modifications to the data.
- **Passive attacks** are of two types :
  1. Release of message contents 2. Traffic analysis
- **Release of message content** is shown in Fig. 1.5.1. 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 content of these transmissions.

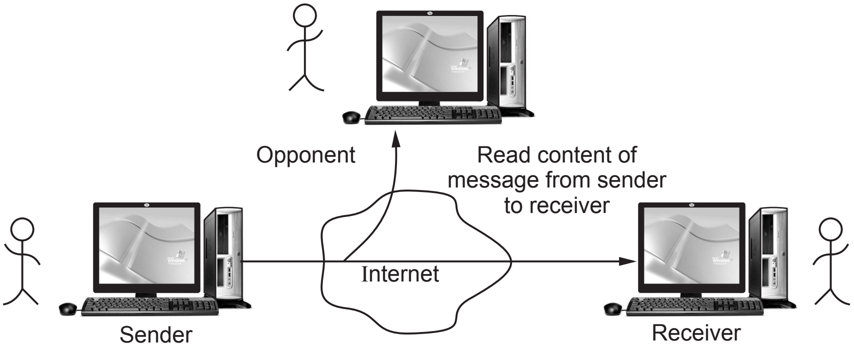


Fig. 1.5.1 Release of message contents

- **Traffic analysis :** Mask the contents of message so that opponents could not extract the information from the message. Encryption is used for masking Fig. 1.5.2 shows the traffic analysis.

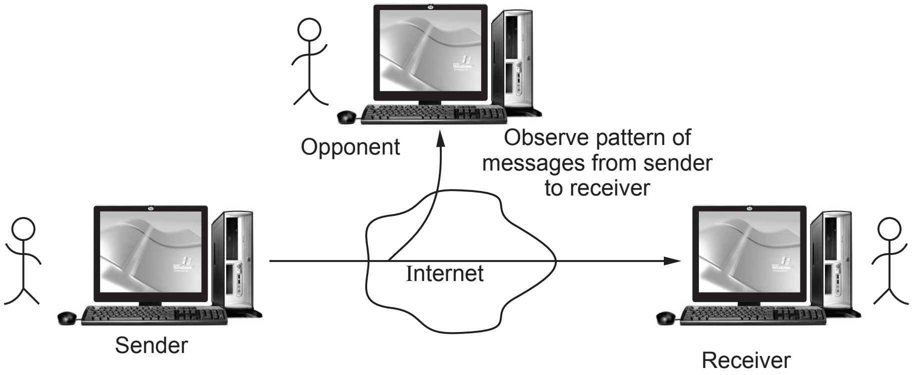


Fig. 1.5.2 Traffic analysis

- Passive attacks are very difficult to detect because they do not involve any alternation of data. It is feasible to prevent the success of attack, usually by means of encryption.

1.5.2 Active Attack

- Active attacks involve some modification of the data stream or the creation of a false stream. These attacks can not be prevented easily.
- Active attacks can be subdivided into four types :

1. Masquerade	2. Replay
3. Modification of message	4. Denial of service

1. Masquerade

- It takes place when one entity pretends to be a different entity. Fig. 1.5.3 shows masquerade.

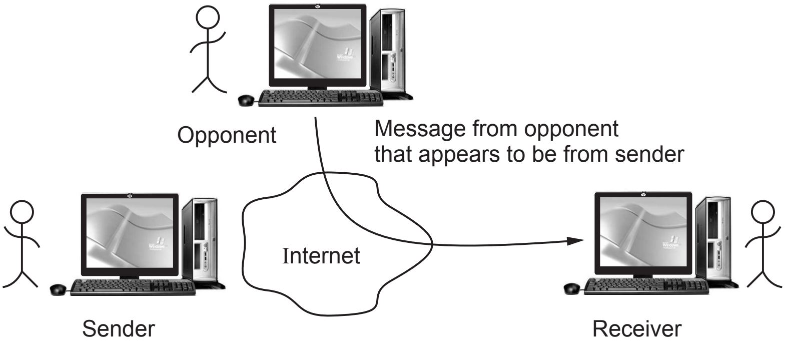


Fig. 1.5.3 Masquerade

- **For example :** Authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.
- **Interruption** attacks are called as masquerade attacks.

2. Replay

- It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.
- Fig. 1.5.4 shows replay attack.

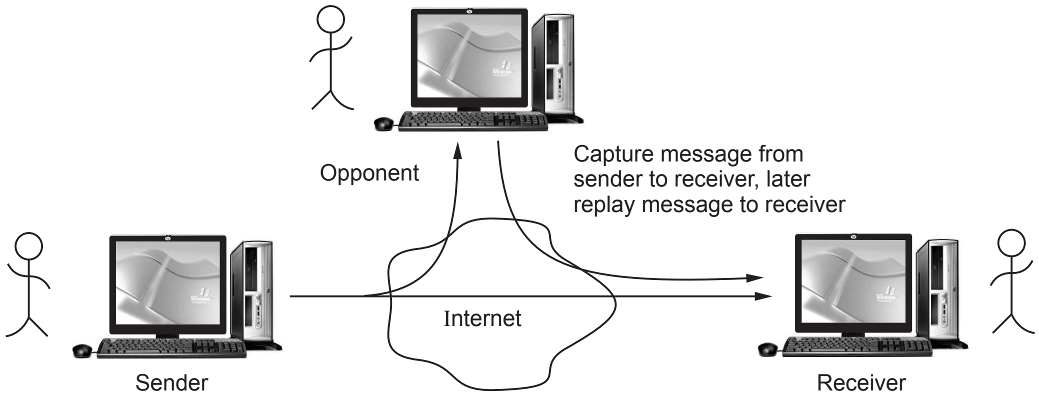

Opponent
Capture message from sender to receiver, later replay message to receiver
Internet
Sender
Receiver

Fig. 1.5.4 Replay

3. Modification of message

- It involves some change to the original message. It produces an unauthorized effect. Fig. 1.5.5 shows the modification of message.

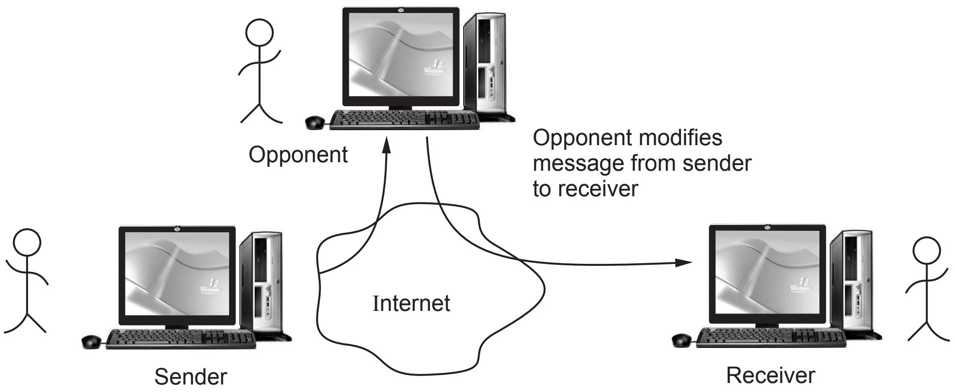


Fig. 1.5.5 Modification of message

- For example, a message meaning "Allow Rupali Dhotre to read confidential file accounts " is modified to mean "Allow Mahesh Awati to read confidential file accounts".

4. Denial of service

- Fabrication causes Denial Of Service (DOS) attacks.
- DOS prevents the normal use or management of communications facilities.
- Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.
- Fig. 1.5.6 shows denial of service attack.

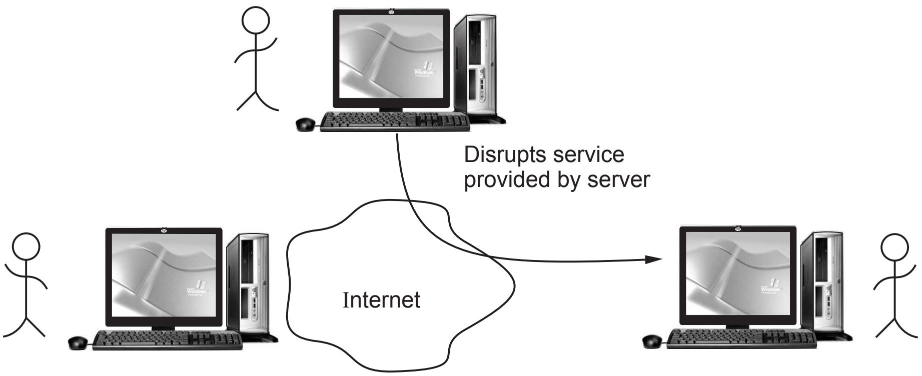


Fig. 1.5.6 Denial of service

- It is difficult to prevent active attack because of the wide variety of potential physical, software and network vulnerabilities.
- The first type of DOS attacks were single source attacks, meaning that a single system was used to attack another system and cause something on that system to fail. SYN flood is the most widely used DOS attack.
- Fig. 1.5.7 shows the SYN flood DOS attack.

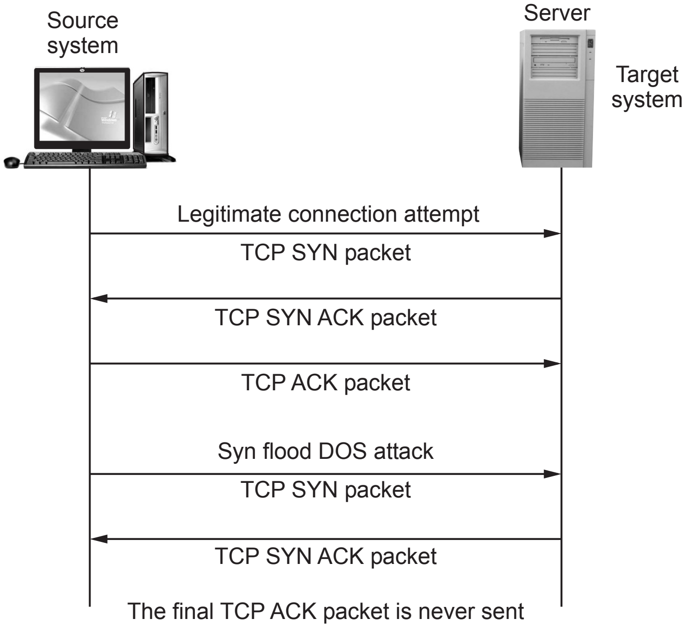


Fig. 1.5.7 SYN flood DOS attack

- Source system sends a large number of TCP SYN packets to the target system. The SYN packets are used to begin a new TCP connection.

- When the target receives a SYN packet, it replies with TCP SYN ACK packet, which acknowledges the SYN packet and sends connection setup information back to the source of the SYN.
- The target also places the new connection information into a pending connection buffer.
- For a real TCP connection, the source would send a final TCP ACK packet when it receives the SYN ACK.
- However, for this attack, the source ignores the SYN ACK and continues to send SYN packets. Eventually, the target's pending connection buffer fills up and it can no longer respond to new connection requests.

1.5.3 Difference between Passive and Active Attack

Sr. No.	Passive attacks	Active attacks
1.	Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions.	Active attacks involve some modification of the data stream or the creation of a false stream.
2.	**Types :** Release of message contents and traffic analysis	**Types :** Masquerade, replay, modification of message and denial of service.
3.	Very difficult to detect.	Easy to detect.
4.	The emphasis in dealing with passive attacks is on prevention rather than detection.	It is quite difficult to prevent active attacks absolutely.
5.	It does not affect the system.	It affects the system.

1.5.4 Man-in-the-Middle Attack

- In cryptography, a **Man-In-The-Middle (MITM) attack** is an attack in which an attacker is able to read, insert and modify at will, meassages between two parties without either party knowing that the link between them has been compromised.
- The attacker must be able to observe and intercept messages going between the two victims. The MITM attack can work against public-key cryptography and is also particularly applicable to the original Diffie-Hellman key exchange protocol, when used without authentication.
- The MITM attack may include one or more of
  1. Eavesdropping, including traffic analysis and possibly a known-plaintext attack.

2. Chosen ciphertext attack, depending on what the receiver does with a message that it decrypts.
3. Substitution attack
4. Replay attacks
5. Denial of service attack. The attacker may for instance jam all communications before attacking one of the parties. The defense is for both parties to periodically send authenticated status messages and to treat their disappearance with paranoia.

- MITM is typically used to refer to active manipulation of the meassages, rather than passively eavesdropping.

Example of a successful MITM attack against public-key encryption

- Suppose Alice wishes to communicate with Bob and that Mallory wishes to eavesdrop on the conversation, or possibly deliver a false message to Bob. To get started. Alice must ask Bob for his public key. If Bob sends his public key to Alice, but Mallory is able to intercept it, a man-in-the-middle attack can begin.
- Mallory can simply send Alice a public key for which she has the private, matching, key. Alice, believing this public key to be Bob's, then encrypts her message with Mallory's key and sends the enciphered message back to Bob.
- Mallory again intercepts, deciphers the message, keeps a copy, and reenciphers it using the public key Bob originally sent to Alice. When Bob receives the newly enciphered message, he will believe it came from Alice.
- This example shows the need for Alice and Bob to have some way to ensure that they are truly using the correct public keys of each other. Otherwise, such attacks are generally possible in principle, against any message sent using public-key technology.

Defenses against the attack

- The possibility of a man-in-the-middle attack remains a serious security problem even for many public-key based cryptosystems. Various defenses against MITM attacks use authentication techniques that are based on :
  1. Public keys
  2. Stronger mutual authentication
  3. Secret keys (high information entropy secrets)
  4. Passwords (low information entropy secrets)
  5. Other criteria, such as voice recognition or other biometrics
- The integrity of public keys must generally be assured in some manner, but need not be secret, whereas passwords and shared secret keys have the additional

secrecy requirement. Public keys can be verified by a Certificate Authority, whose public key is distributed through a secure channel.

University Questions

1. *Briefly explain any two active security attacks.* **GTU : Summer-17, Marks 4**
2. *Discuss man in middle attack.* **GTU : Winter-18, Marks 4**
3. *Explain different type of attacks on crypto system.* **GTU : Winter-18, Marks 4**
4. *Explain cryptanalytic attacks with example of any encryption algorithm.* **GTU : Winter-19, Marks 7**

1.6 Finite Fields

- A group G is a nonempty set together with a *binary operation* (*) such that the following three properties are satisfied :
  1. **Associativity :** (a*b)*c = a*(b*c). For all a, b, c ε G.
  2. **Identity :** There is an element e ε G such that a*e = e*a. For all a ε G.
  3. **Inverses :** For each element a ε G, there is an element b ε G such that a*b = b*a = e.
- Order of a Group G is the number of elements it contains (denoted |G|). Order of an element g ε G is the smallest positive integer n such that $g^n = e$ (denoted |g|). Here $g^n = g*g*\ldots*g$ n (times). In a *finite* group, the order of each element of the group divides the order of the group.

Properties of Groups

- For all g ε G, $g^0 = e$.
- For all n, m ≥ 1, g ε G,
  1. $g^n = g^{n-1} * g$
  2. $g^n * g^m = g^{n+m}$
  3. $(g^n)^{-1} = g^{-n} = (g^{-1})^n$
  4. $(g^m)^n = g^{mn}$
- If G is a group and for all a, b, ε G we have a*b = b*a (commutativity) then G is called an **Abelian Group.**
- In an Abelian group G, for all a, b ε G, then $(a * b)^{-1} = b^{-1} * a^{-1} = a^{-1} * b^{-1}$

1.7 Modular Arithmetic

- Much of modern number theory and many practical problems (including problems in cryptography), are concerned with *modular arithmetic*. In arithmetic modulo N, we are concerned with arithmetic on the integers, where we identify all numbers which differ by an exact multiple of N. That is,

$$x \equiv y \bmod N \text{ if } x = y + mN \qquad \text{for some integer m.}$$

- This identification divides all the integers into N equivalence classes. We usually denote these by their "simplest" members, that is, the numbers 0, 1, ... , N – 1.
- If a is an integer and n is a positive integer, define a mod n to be the remainder when a is divided by n. Then, $a = [a/n] \times n + (a \bmod n)$;
- Example : 11 mod 7 = 4; – 11 and 7 = 3.

Theorem : ≡ n is an equivalence relation on the integers. An equivalence class consists of those integers which have the same remainder on division by n. The equivalence classes are also known as congruence classes modulo n. Rather than say the integers a and b are equivalent we say that they are congruent modulo n.

Definition :

The set of all integers congruent to a modulo n is called the residue class [a].

Example : Residue classes mod 3 :

$$[0] = \{..., -6, -3, 0, 3, 6, ...\}$$

$$[1] = \{..., -5, -2, 1, 4, 7, ...\}$$

$$[2] = \{..., -4, -1, 2, 5, 8, ...\}$$

- The modulo operator has the following properties :
  1. $a \equiv b \bmod n$ if $n | (a - b)$.
  2. $(a \bmod n) = (b \bmod n)$ implies $a \equiv b \bmod n$.
  3. $a \equiv b \bmod n$ implies $b \equiv a \bmod n$.
  4. $a \equiv b \bmod n$ and $b \equiv c \bmod n$ imply $a \equiv c \bmod n$.
- Properties of modular arithmetic operations :
  1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
  2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
  3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$
- Proof of property 1 :

Define (a mod n) = r_a and (b mod n) = r_b. Then $a = r_a + jn$ and $b = r_b + kn$ for some integers j and k. Then,

$$\begin{aligned}(a + b) \bmod n &= (r_a + jn + r_b + kn) \bmod n \\ &= (r_a + r_b + (j + k)\, n) \bmod n \\ &= (r_a + r_b) \bmod n \\ &= [(a \bmod n) + (b \bmod n)] \bmod n\end{aligned}$$

- Examples for the above three properties

  11 mod 8 = 3; 15 mod 8 = 7

  [(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 = 2

  (11 + 15) mod 8 = 26 mod 8 = 2

  [(11 mod 8) – (15 mod 8)] mod 8 = – 4 mod 8 = 4

  (11 – 15) mod 8 = – 4 mod 8 = 4

  [(11 mod 8) × (15 mod 8)] mod 8 = 21 mod 8 = 5

  (11 × 15) mod 8 = 165 mod 8 = 5

- Properties of modular arithmetic

Let, $Z_n = \{0, 1, 2, \ldots , (n - 1)\}$ be the **set of residues modulo n.**

Property		Expression
Commutative laws	1.	(w + x) mod n = (x + w) mod n
	2.	(w × x) mod n = (x × w) mod n
Associative laws	1.	[(w + x) + y] mod n = [w + (x + y)] mod n
	2.	[(w × x) × y] mod n = [w × (x × y)] mod n
Distributive law		[w × (x + y)] mod n = [(w × x) + (w × y)] mod n
Identities		(0 + w) mod n = w mod n
		(1 × w) mod n = w mod n
Additive inverse (– w)		For each $w \in Z_n$, there exists a z such that w + Z ≡ 0 mod n

- If (a + b) ≡ (a + c) mod n, then b ≡ c mod n (due to the existence of an additive inverse)
- If (a × b) ≡ (a × c) mod n, then b ≡ c mod n (only if a is relatively prime to n; due to the possible absence of a multiplicative inverse).

e.g. $6 \times 3 = 18 \equiv 12 \bmod 8$ and

$6 \times 7 = 42 \equiv 2 \bmod 8$ but

$3 \neq 7 \bmod 8$ (6 is not relatively prime to 8)

- If n is prime then the property of multiplicative inverse holds (from a ring to a field).
- Following table provides modular addition and multiplication modulo 7.

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

(a) Addition modulo 7

*	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

(b) Multiplication modulo 7

w	– w	w^–1
0	0	---
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

(c) Additive and multiplicative inverses modulo 7

Table 1.7.1 Arithmetic modulo 7

1.7.1 Modular Exponentiation

- Modular exponentiation is a type of exponentiation performed over a modulus. Doing a modular exponentiation means calculating the remainder when dividing by a positive integer m (called the modulus) a positive integer b (called the base) raised to the e-th power (e is called the exponent).
- In other words, problems take the form where given base b, exponent e, and modulus m, one wishes to calculate c.
- Many public-key encryption algorithms use modular exponentiation - raising a number a (base) to some power b (exponent) mod p.
- $c = ab = a \cdot a \ldots a \bmod p$

Example 1.7.1 *To find 11^{13} mod 53*

Solution : $13 = 8 + 4 + 1$ so $11^{13} = 11^{8+4+1} = 11^8 * 11^4 * 11^1$

We can compute successive squares of 11 to obtain, $11, 11^2, 11^4, 11^8$ and then multiply together $11^1 * 11^4 * 11^8$ to get the answer 11^{13}.

Because we are working mod 53, we will "take mods" at every stage of the calculation.

Thus we have

$$11 \bmod 53 = 11$$

$$11^2 = 121, \; 121 \bmod 53 = 121 - 2*53 = 15$$

$$11^4 = (11^2)^2 = 15^2 \bmod 53 = 225 \bmod 53 = 225 - 4*53 = 13$$

$$11^8 = (11^4)^2 = 13^2 \bmod 53 = 169 \bmod 53 = 169 - 3*53 = 10$$

Therefore $11^{13} \bmod 53 = 11 * 13 * 10 = 1430 \bmod 53 = 1430 = 26*53 + 52$

The answer is $11^{13} \bmod 53 = 52$.

1.8 Euclidean Algorithm

- The Euclidean algorithm is an algorithm for finding the greatest common divisor of two positive integers.
- The greatest common divisor of two integers is defined as : An integer c is called the gcd(a, b) (read as the greatest common divisor of integers a and b) if the following 2 conditions hold :

  1) $c \mid a \cup c \mid b$

  2) For any common divisor d of a and b, $d \mid c$.

- Rule 2 ensures that the divisor c is the greatest of all the common divisors of a and b.
- One way we could find the gcd of two integers is by trial and error. Another way is that we could prime factorize each integer and from the prime factorization, see which factors are common between the two integers. However, both of these become very time consuming as soon as the integers are relatively large.
- However, Euclid devised a fairly simple and efficient algorithm to determine the gcd of two integers. The algorithm basically makes use of the division algorithm repeatedly.
- Let's say you are trying to find the gcd(a, b), where a and b are integers with $a^{3} b > 0$.
- Euclid's algorithm says to write out the following :

$$\begin{aligned}
a &= q_1 b + r_1, & \text{where } & 0 < r < b \\
b &= q_2 r_1 + r_2, & \text{where } & 0 < r_2 < r_1 \\
r_1 &= q_3 r_2 + r_3, & \text{where } & 0 < r_3 < r_2 \\
&. \\
&. \\
r_i &= q_{i+2} r_{i+1} + r_{i+2}, & \text{where } & 0 < r_{i+2} < r_{i+1} \\
&. \\
&. \\
r_{k-1} &= q_{k+1} r_k
\end{aligned}$$

- Euclid's algorithm says that the gcd(a, b) = $\mathbf{r_k}$
- Consider computing gcd(125, 87)

$$\begin{aligned}
125 &= 1 * 87 + 38 \\
87 &= 2 * 38 + 11 \\
38 &= 3 * 11 + 5 \\
11 &= 2 * 5 + 1 \\
5 &= 5 * 1
\end{aligned}$$

Thus, we find gcd(125, 87) = 1

Example 1.8.1 *Find gcd(125, 20)*

Solution :

$$\begin{aligned}
125 &= 6 * 20 + 5 \\
20 &= 4 * 5,
\end{aligned}$$

Thus, the gcd(125, 20) = 5

1.8.1 Extended Euclidean Algorithm

- One of the consequences of the Euclidean algorithm is as follows :

  Given integers a and b, there is always an integral solution to the equation $ax + by = \gcd(a,b)$.

- Furthermore, the Extended Euclidean Algorithm can be used to find values of x and y to satisfy the equation above. The algorithm will look similar to the proof in some manner.

- Consider writing down the steps of Euclid's algorithm :

$$\begin{aligned}
a &= q_1 b + r_1, && \text{where } 0 < r < b \\
b &= q_2 r_1 + r_2, && \text{where } 0 < r_2 < r_1 \\
r_1 &= q_3 r_2 + r_3, && \text{where } 0 < r_3 < r_2 \\
&\;\cdot \\
&\;\cdot \\
r_i &= q_{i+2} r_{i+1} + r_{i+2}, && \text{where } 0 < r_{i+2} < r_{i+1} \\
&\;\cdot \\
r_{k-2} &= q_k r_{k-1} + r_k, && \text{where } 0 < r_k < r_{k-1} \\
r_{k-1} &= q_{k+1} r_k
\end{aligned}$$

- Consider solving the second to last equation for r_k. You get

$$\begin{aligned}
r_k &= r_{k-2} - q_k r_{k-1}, \text{ or} \\
\gcd(a,b) &= r_{k-2} - q_k r_{k-1}
\end{aligned}$$

Now, solve the previous equation for r_{k-1} :

$$r_{k-1} = r_{k-3} - q_{k-1} r_{k-2},$$

and substitute this value into the previous derived equation :

$$\begin{aligned}
\gcd(a,b) &= r_{k-2} - q_k (r_{k-3} - q_{k-1} r_{k-2}) \\
\gcd(a,b) &= (1 + q_k q_{k-1}) r_{k-2} - q_k r_{k-3}
\end{aligned}$$

- Now we have expressed gcd(a, b) as a linear combination of r_{k-2} and r_{k-3}. Next we can substitute for of r_{k-2} in terms of r_{k-3} and r_{k-4}, so that the gcd(a, b) can be expressed as the linear combination of r_{k-3} and r_{k-4}. Eventually, by continuing this process, gcd(a, b) will be expressed as a linear combination of a and b as desired.

- Find integers x and y such that : $135x + 50y = 5$.

- Use Euclid's algorithm to compute gcd(135, 50) :

$$\begin{aligned}
135 &= 2 * 50 + 35 \\
50 &= 1 * 35 + 15 \\
35 &= 2 * 15 + 5 \\
15 &= 3 * 5
\end{aligned}$$

- Now, let's use the Extended Euclidean algorithm to solve the problem :
  $5 = 35 - 2 * 15$, from the second to last equation $35 = 2 * 15 + 5$.
- But, we have that
  $15 = 50 - 35$, from the third to last equation $50 = 1 * 35 + 5$.
- Now, substitute this value into the previously derived equation :

$$5 = 35 - 2 * (50 - 35)$$

$$5 = 3 * 35 - 2 * 50$$

- Now, finally use the first equation to determine an expression for 35 as a linear combination of 135 and 50 :

$$35 = 135 - 2 * 50.$$

- Plug this into our last equation :

$$5 = 3 *(135 - 2 * 50) - 2 * 50$$

$$5 = 3 * 135 - 8 * 50$$

So, a set of solutions to the equation is $x = 3$, $y = -8$.

Example 1.8.2 *Using Euclidean algorithm calculate gcd (16, 20) and gcd (50, 60).*

Solution : gcd (16, 20)

Step 1 : $a_1 = 20, \quad b_1 = 16$ **Step 2 :** $a_2 = 16 \quad b_2 = 4$

$20 = 16 \times 1 + 4$ $16 = 4 \times 4 + 0$

Here $r_2 = 0$ and so the last non-zero reminder is $r_2 = 4$.

Thus gcd (16, 20) = 4

gcd (50, 60)

$$a_1 = 60, \quad b_1 = 50$$

$$a_1 = b_1 q_1 + r_1 = 50 \times 1 + 10$$

$$a_2 = 50, \quad b_2 = 10 = b_2 q_2 + r_2 = 10 \times 5 + 0$$

Here $r_2 = 0$ and so the last non-zero remainder is $r_2 = 10$. Thus gcd (50, 60) = 10

Example 1.8.3 *Using Euclidean algorithm calculate GCD (48, 30) and GCD (105, 80).*

Solution : Using Euclidean algorithm calculate GCD :

GCD(48, 30)

$$48 = 1 \times 30 + 18 \text{ gcd}(30, 18)$$

$$30 = 1 \times 18 + 12 \text{ gcd } (18, 12)$$

$$18 = 1 \times 12 + 6 \text{ gcd}(12, 6)$$

$$12 = 2 \times 6 + 0 \text{ gcd}(6, 0)$$

Therefore, GCD(48, 30) = 6

GCD(105, 80)

$$\begin{aligned} 105 &= 1 \quad 80 + 25 \text{ gcd}(80, 25) \\ 80 &= 3 \quad 25 + 5 \text{ gcd } (25, 5) \\ 25 &= 5 \quad 5 + 0 \text{ gcd}(5, 0) \end{aligned}$$

Therefore, GCD(105, 80) = 5

1.8.2 Greatest Common Divisor

- Definition. A positive integer d is called the greatest common divisor of the nonzero integers a and b if
  i) d is a divisor of both a and b, and
  ii) Any divisor of both a and b is also a divisor of d.
- We will use the notation gcd(a, b), or simply (a, b), for the greatest common divisor of a and b.
- Greatest Common Divisor gcd(a,b) is the largest number that divides both a and b.
- If a and b share no common factors, they are called relatively prime.

Example 1.8.4 *Find gcd(1403, 1081).*

Solution :

$$\begin{aligned} 1403 &= 1081.1 + 322 \\ 1081 &= 322.3 + 115 \\ 322 &= 115.2 + 92 \\ 115 &= 92.1 + 23 \\ 92 &= 23.4 + 0 \end{aligned}$$

The last nonzero remainder is 23, so gcd(1403, 1081) = 23.

Example 1.8.5 *Find gcd (120 , 70).*

Solution :

$$\begin{aligned} 120 &= 70 + 50 \\ 70 &= 50 + 20 \\ 50 &= 20 \times 2 + 10 \\ 20 &= 10 \times 2 + 0 \end{aligned}$$

Therefore gcd (120,70) = 10.

- It is always possible to write gcd(a, b) as a linear combinations of a and b. That is, there exist integers x and y such that gcd(a, b) = ax+by (x or y may be negative).

- In fact, though we have not proved it, gcd(a, b) is the smallest positive linear combination of a and b. Once we use the Euclidean algorithm to find gcd(a, b) we can then retrace our steps to write gcd(a, b) in the form ax+by.

1.9 Conventional Cryptosystem

- A message is to be transferred from source to destination across some sort of internet. Both the sides must cooperate for the exchange of the data.
- A logical information channel is established by defining a route through the internet from source to destination.
- All the techniques for providing security have two components :
  1. A security related transformation on the information to be sent.
  2. Some secret information shared by the two principles, it is hoped, unknown to the opponent.
- Fig. 1.9.1 shows the network security model.

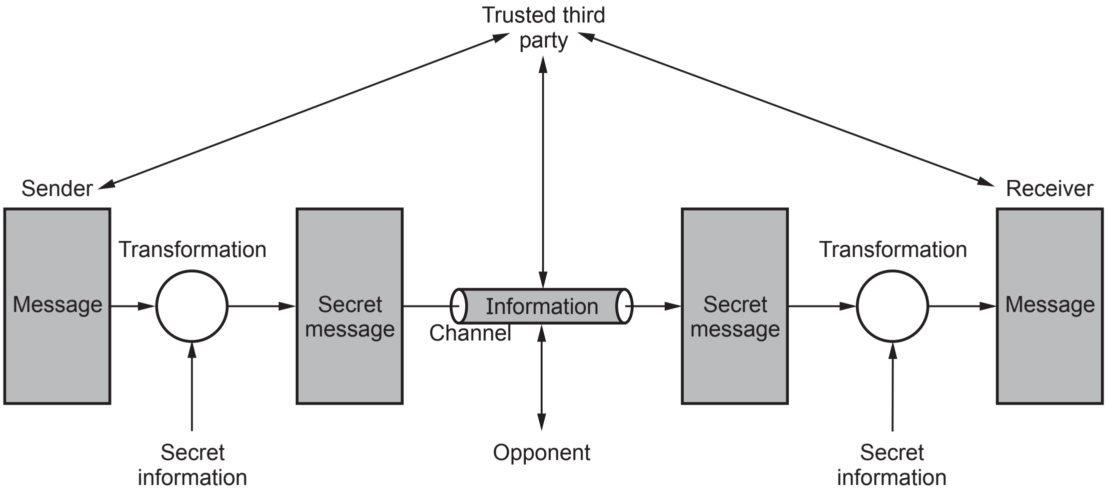


Fig. 1.9.1 Network security model

- A trusted third party is needed to achieve secure transmission.
- Basic tasks in designing a particular security service.
  1. Design an algorithm for performing the security related transformation.
  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 principles that makes use of the security algorithm and the secret information to achieve a particular security service.

1.10 Symmetric Cipher Model

- A symmetric encryption model has five ingredients : Plaintext, Encryption algorithm, Secret key, Ciphertext and Decryption algorithm.
- Fig. 1.10.1 shows the conventional encryption model.

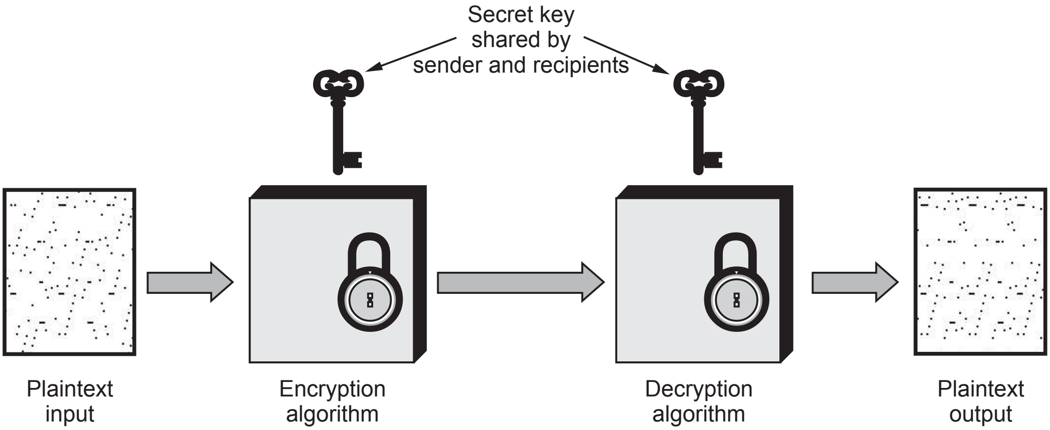


Fig. 1.10.1 Conventional encryption model

- Plaintext is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm** performs various substitutions and transformations on the plaintext.
- **Secret key** is a value independent of the plaintext and of the algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext** is the scrambled message produced as output. It depends on the plaintext and the secret key.
- **Decryption algorithm** takes the ciphertext and the secret key and produces the original plaintext.
- The original intelligible message, referred to as plaintext is converted into random nonsense, referred to as ciphertext. The science and art of manipulating messages to make them secure is called **cryptography.**
- An original message to be transformed is called the plaintext and the resulting message after the transformation is called the ciphertext.
- The process of converting the plaintext into ciphertext is called encryption. The reverse process is called decryption. The encryption process consists of an algorithm and a key. The key controls the algorithm.
- The objective is to design an encryption technique so that it would be very difficult or impossible for an unauthorized party to understand the contents of the ciphertext.

- A user can recover the original message only by decrypting the ciphertext using the secret key. Depending upon the secret key used, the algorithm will produce a different output. If the secret key changes, the output of the algorithm also changes.
- The security of the conventional encryption depends on the several factors. The encryption algorithm must be powerful. Decryption message must be difficult. The algorithm depend on the secrecy of the key only. The algorithm is upon to all but only key is to keep secret. As shown in the diagram, the message source is the plaintext i.e. X with the message X and encryption key K as input and ciphertext Y, we can write this as,

$$Y = E(K, X) \qquad \ldots(1.10.1)$$

- Y is to be produced by using encryption algorithm E as a function of the plaintext X. The intended receiver in possession of the key, is able to invert the transformation.

$$X = D(K, Y) \qquad \ldots(1.10.2)$$

- An opponent, observing Y but not having access to K or X, must attempt to recover X and K or both X and K. It is assumed that the opponent does have knowledge of the encryption (E) and decryption (D) algorithms.

1.10.1 Advantages of Symmetric Key Cryptography

1. High rates of data throughput.
2. Keys for symmetric-key ciphers are relatively short.
3. Symmetric-key ciphers can be used as primitives to construct various cryptographic mechanisms (i.e. pseudorandom number generators).
4. Symmetric-key ciphers can be composed to produce stronger ciphers.
5. Symmetric-key encryption is perceived to have an extensive history.

1.10.2 Disadvantages of Symmetric Key Cryptography

1. Key must remain secret at both ends.
2. In large networks, there are many keys pairs to be managed
3. Sound cryptographic practices dictates that the key be changed frequently
4. Digital signature mechanisms arising from symmetric-key encryption typically require either large keys or the use of third trusted parties.

University Question

1. *What are the essential ingredients of a symmetric cipher ?* **GTU : Winter-17, Marks 4**

1.11 Cryptography

- Cryptography is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents. The term is derived from the Greek word kryptos, which means hidden.
- Cryptography is the science of secret writing that brings numerous techniques to safeguard information that is present in an unreadable format.
- By using cryptographic systems, the sender can first encrypt a message and then pass on it through the network. The receiver on the other hand can decrypt the message and restore its original content.

Characteristics of cryptography :

1. The types of operations used for transforming plaintext to ciphertext.
2. The number of keys used.
3. The way in which the plaintext is processed.

- Cryptography is an art or science encompassing the principles and methods of transforming a plaintext message into one that is unintelligible, and then that message back to its original form.
- Cryptanalysis : The study of methods for obtaining the meaning of encrypted information without accessing the secret information.
- Cryptography is where security engineering meets mathematics
- Cryptology = Cryptography + cryptanalysis
- Some Major Applications :
  1. To protect privacy, confidentiality
  2. Insuring data integrity for detecting and preventing unauthorized data manipulation
  3. Authentication, the means by which two parties can positively identify each other.
  4. Non-repudiation - to hold people responsible for their actions.
- There are a number of cryptographic primitive's basic building blocks, such as block ciphers, stream ciphers, and hash functions. Block ciphers may either have one key for both encryption and decryption, in which case they're called shared key or have separate keys for encryption and decryption, in which case they're

called public key or asymmetric. A digital signature scheme is a special type of asymmetric crypto primitive.

- Perhaps the most straightforward example is to be found using the simple mono-alphabetic substitution cipher method, in which each letter in the alphabet is shifted by an integer value. The key supplied to this algorithm would be that integer value. However, a problem arises in sharing the key with the intended recipient without letting it be discovered by others.

University Questions

1. *Discuss the following terms in brief :*
   - Brute force attack
   - Cryptography **GTU : Summer-17, Marks 3**
2. *Discuss the following terms in brief : Passive attack, Cryptanalysis.* **GTU : Winter-17, Marks 3**

1.12 Cryptanalysis

- The process of trying to break any cipher text message to obtain the original plain text message itself is called as cryptanalysis.
- Cryptanalysis is the art of deciphering encrypted communications without knowing the proper keys
- Cryptanalysis is the breaking of codes. The person attempting a cryptanalysis is called as a cryptanalyst.
- Brute force attack : The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained.
- Types of Attacks on Encrypted Messages :

Sr. No.	Types of attack	Known to cryptanalyst
1.	Ciphertext only	1.Encryption algorithm 2.Cipher text
2.	Known plaintext	1.Encryption algorithm 2.Cipher text 3.One or more plaintext ciphertext pairs formed with the secret key.

3.	Chosen plaintext	1.Encryption algorithm 2.Ciphertext 3.Plain text message chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key.
4.	Chosen ciphertext	1. Encryption algorithm 2.Cipher text 3.Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key.
5.	Chosen text	1.Encryption algorithim 2.Cipher text 3.Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key. 4.Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key.

1.12.1 Cryptanalysis Attacks

There are four general types of cryptanalytic attacks. Each of them assumes that the cryptanalyst has complete knowledge of the encryption algorithm used.

1. Ciphertext-only attack
2. Known-plaintext attack
3. Chosen-plaintext attack
4. Adaptive chosen plaintext attack.

1. Ciphertext - only attack :

- The cryptanalyst has the ciphertext of several messages, of all of which have been encrypted using the same encryption algorithm.
- The analyst may be able to capture one or more plaintext messages as well as their encryptions.
- Better yet to deduce the key used to encrypt the messages, in order to decrypt other messages encrypted with the same keys.

Given : $C_1 = E_k(P_1)$, $C_2 = E_k(P_2)$..........$C_i = E_k(P_i)$

Deduce : Either P_1, P_2,........P_i, K or an algorithm to infer P_{i+1} from

$C_{i+1} = E_k(P_{i+1})$

2. Known-plaintext attack

- The cryptanalyst has access not only to the ciphertext of several messages, but also to the plaintext of those messages.
- Job is to deduce the key used to encrypt the messages.
- OR an algorithm to decrypt any new messages encrypted with the same key.
- It is also referred to as a probable word attack.

Given : $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2)\ldots\ldots\ldots P_i, C_i = E_k(P_i)$

Deduce : Either K or an algorithm to infer P_{i+1} from $C_{i+1} = E_k(P_{i+1})$

3. Chosen-plaintext attack

- This is more powerful than a known plaintext attack because the cryptanalyst can choose specific plaintext blocks to encrypt.
- The cryptanalyst not only has access to the ciphertext and associated plaintext for several messages, but he also chooses the plaintext that gets encrypted.

Given : $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2)\ldots\ldots\ldots P_i, C_i = E_k(P_i)$

where the cryptanalyst gets to choose $P_1, P_2, \ldots\ldots P_i$

Deduce : Either K or an algorithm to infer P_{i+1} from $C_{i+1} = E_k(P_{i+1})$

4. Adaptive chosen plaintext attack

- Not only can the cryptanalyst choose the plaintext that is encrypted, but he can also modify his choice based on the result of previous encryption.
- A cryptanalyst might just be able to choose one large block of plaintext to be encrypted - **in chosen plaintext attack**.

Example 1.12.1 *What is the objective of attacking an encryption system ? Write the two approaches to attack a conventional encryption scheme.* **GTU : Summer-12, Marks 7**

Solution : The objective of attacking an encryption system is to recover the key in use rather then simply to recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme :

1. **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. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
2. **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.

If either type of attack succeeds in deducing the key, the effect is catastrophic : All future and past messages encrypted with that key are compromised.

1.13 Vulnerability and Threat

Computer based systems have three valuable components : **Hardware, software and data.**

- Securities of these components are evaluated in terms of vulnerability, threats, attacks and control.
- An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt to avade security services and violate the security policy of a system.

Asset :

- Asset means people, property and information.
- People may include employees and customers along with invited persons such as contractors or guests.

Vulnerability :

- Vulnerability refers to the security flaws in a system that allows an attack to be successful.
- Weaknesses or gaps in a security program that can be exploited by threats to gain unauthorized access to an asset. Vulnerability is a weakness or gap in our protection efforts.
- **Example :** In design, implementation or procedure, that might be exploited to cause loss or harm.

Threat :

- Anything that can exploit vulnerability, intentionally or accidentally and obtain, damage or destroy an asset. A threat is what we're trying to protect against.
- Threat refers to the source and means of a particular type of attack.
- A threat assessment is performed to determine the best approaches to securing a system against a particular threat or class of threat.
- A potential for violation of security, which exists when there is a circumstance, capability, action or event that could breach security and case harm. That is, a threat is a possible danger that might exploit vulnerability.
- Where risk assessments focus more on analyzing the potential and tendency of one's resources to fall prey to various attacks, threat assessments focus more on analyzing the attacker's resources.

- Analyzing threats can help one develop specific security policies to implement in line with policy priorities and understand and specific implementation needs for securing one's resources.
- Threats come in many forms, depending on their mode of attack. From viruses to trojans, spyware and bots, threats have evolved into sophisticated programs intended to harm computers.

Risk :

- The potential for loss, damage or destruction of an asset as a result of a threat exploiting vulnerability. Risk is the intersection of assets, threats and vulnerabilities.
- The formula used to determine risk is

$$\textbf{Risk} = \textbf{Asset + Threat + vulnerability}$$

$$\mathbf{R = A + T + V}$$

- Risk is a function of threats exploiting vulnerabilities to obtain damage or destroy assets. Thus, threats may exist, but if there are no vulnerabilities then there is little / no risk.
- Similarly, you can have vulnerability, but if you have no threat, then you have little / no risk.

Control :

- Control is used as proactive measure. Control is a action, device, procedure or technique that removes or reduces a vulnerability.
- A threat is blocked by control of vulnerability.
- Interception, interruption, modification and fabrication are the system security threats.

Brute force attack :

- The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained.
- Brute force attack is an automated process of trial and error used to guess a person's user name, password, credit-card number of cryptographic keys.

1.14 Substitution Techniques

- A substitution cipher changes characters in the plaintext to produce to ciphertext. 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.

1.14.1 Caesar Cipher

- Caesar cipher is a special case of substitution techniques wherein each alphabet in a message is replaced by an alphabet three places down the line.
- Caesar cipher is susceptible to a statistical ciphertext only attack.
- For example,

Plaintext	hellow world
Ciphertext	KHOOR ZRUOG

- List of all possible combination of letters.

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
Cipher	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V

Plain	t	u	v	w	x	y	z
Cipher	W	X	Y	Z	A	B	C

- Numerical equivalent to each letter is given below.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- The algorithm can be expressed as follows. For each plaintext letter P, substitute the ciphertext letter C :

$$C = E(3, P) = (P + 3) \bmod 26$$

- A shift may be of any amount, so that the general Caesar algorithm is

$$C = E(K, P) = (P + K) \bmod 26$$

where K = Values from 1 to 25

- The decryption algorithm is simply

$$P = D(K, C) = (C - K) \bmod 26$$

- If it is known that a given ciphertext is a Caesar cipher, then a brute force cryptanalysis is easily performed : Simply try all the 25 possible keys.
- **Demerits :**
  1. The encryption and decryption algorithms are known.
  2. There are only 25 keys to try.
  3. The language of the plaintext is known and easily recognizable.

1.14.2 Monoalphabetic Cipher

- Monoalphabetic cipher substitutes one letter of the alphabet with another letter of the alphabet. However, rather than substituting according to a regular pattern, any letter can be substituted for any other letter, as long as each letter has a unique substitute left and vice versa.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	m	n	b	v	c	x	z	a	s	d	f	g	h

Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	j	k	l	p	o	i	u	y	t	r	e	w	q

For example

Plaintext message : hello how are you

Ciphertext message : acggk akr moc wky

- Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.

Homophonic Substitution Cipher

- It provides multiple substitutes for a single letter. For example, A can be replaced by D, H, P, R; B can be replaced by E, Q, S, T etc.

1.14.3 Playfair Cipher

- The playfair algorithm is based on the use of a 5 × 5 matrix of letters constructed using a keyword.
- For example : Monarchy is the keyword.

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

- 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 in alphabetic order.
- The letters I and J count as one letter.

1.14.4 Hill Cipher

- The encryption algorithm takes m successive plaintext letters and substitutor 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 = 2, ... z = 25), the system can be described as follows :

$$C_1 = (K_{11} P_1 + K_{12} P_2 + K_{13} P_3) \bmod 26$$

$$C_2 = (K_{21} P_1 + K_{22} P_2 + K_{23} P_3) \bmod 26$$

$$C_3 = (K_{31} P_1 + K_{32} P_2 + K_{33} P_3) \bmod 26$$

- This can be expressed in term of column vectors and matrices :

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \bmod 26$$

or $C = KP \bmod 26$

Where C and P are column vectors of length 3, representing the plaintext and ciphertext.

- K is a 3 × 3 matrix, representing the encrypting key.
- For example :

Plaintext = Paymoremoney

$$\text{Key (K)} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

The first three letters of the plaintext are represented by the vector.

$$C = KP \bmod 26 = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} \bmod 26 = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \text{LNS}$$

For plaintext pay, ciphertext is LNS.

The entire ciphertext is **LNSHDLEWMTRW**

- Decryption requires using the inverse of the matrix K.
- The general terms in Hill cipher is

$$\text{Cipher } C = E(K, P) = KP \bmod 26$$

$$\text{Plaintext } P = D(K, P) = K^{-1} C \bmod 26 = K^{-1} KP = P$$

Advantages

1. It completely hides single letter frequency.
2. Hill cipher is strong against a ciphertext only attack.
3. By using larger matrix, more frequency information hiding is possible.

Disadvantage

1. Easily broken with a known plaintext attack.

1.14.5 Polyalphabetic Substitution

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

Fig. 1.14.1

- In polyalphabetic substitution, each occurrence of a character can have a different substitute. The relationship between a character in the plaintext to a character in the ciphertext is one to many.
- An example of polyalphabetic substitution is the **Vigenere cipher.**
- The Vigenere cipher chooses a sequence of keys, represented by a string. The key letters are applied to successive plaintext characters, and when the end of the key is reached, the key start over.
- Fig. 1.14.1 shows a tableall or table to implement this cipher efficiently,
  (See Fig. 1.14.1 on previous page)
- For example : Let the message be THE BOY HAS THE BAG and let the key be VIG.

  Key = VIG VIG VIG VIG VIG

  Plaintext = THE BOY HAS THE BAG

  Ciphertext = OPKWWECIYOPKWIM
- The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword.

1.14.6 One Time Pad

- The key string is chosen at random and at least as long as the message, so it does not repeat.
- Each new message requires a new key of the same length as the new message. It produces random output that bears no statistical relationship to the plaintext.
- **Vernam cipher** uses a one time pad, which is discarded after a single use, and therefore is suitable only for short messages.
- For example :

Plaintext :	c	o	m	e	t	o	d	a	y
	2	14	12	4	19	14	3	0	24
Key	N	C	B	T	Z	Q	A	R	X
	13	2	1	19	25	16	0	17	23
Total	15	16	13	23	44	30	3	17	47
Subtract 26 if > 25	15	16	13	23	18	04	3	17	21
Ciphertext	P	Q	N	X	S	E	D	R	V

- The one time pad offers complete security but, in practice, has two fundamental difficulties.
  1. There is the practical problem of making large quantities of random keys.
  2. Key distribution and protection is also major problem with one time pad.

1.14.7 Comparison between Monoalphabetic and Polyalphabetic Cipher

Sr. No.	Monoalphabetic Cipher	Polyalphabetic Cipher
1.	Once a key is chosen, each alphabetic character of a plaintext is mapped onto a **unique** alphabetic character of a ciphertext.	Each alphabetic character of a plaintext can be mapped onto "**m**" alphabetic characters of a ciphertext.
2.	The relationship between a character in the plaintext and the characters in the ciphertext is one-to-one.	The relationship between a character in the plaintext and the characters in the ciphertext is one-to-many.
3.	A stream cipher is a monoalphabetic cipher if the value of *k*i does not depend on the position of the plaintext character in the plaintext stream.	A stream cipher is a polyalphabetic cipher if the value of *k*i does depend on the position of the plaintext character in the plaintext stream.
4.	Monoalphabetic cipher includes additive, multiplicative, affine and monoalphabetic substitution cipher.	Polyalphabetic cipher includes autokey, Playfair, Vigenere, Hill, one-time pad, rotor and Enigma cipher.

Example 1.14.1 *Construct a playfair matrix with the key "engineering". And encrypt the message "test this process".* **GTU : Summer-12, Marks 4**

Solution : Using playfair cipher

Message : te st is pr oc es xs

Cipher : PI XY PM GT EU LF PG TY

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

Solved Examples

Example 1.14.2 *Explain playfair cipher substitution technique in detail. Find out cipher text for the following given key and plaintext.*

Key = ENGINEERING

Plaintext = COMPUTER. **GTU : Summer-17, Marks 7**

Solution : key = ENGINEERING plaintext = COMPUTER

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

Plaintext = COMPUTER → CO MP UT ER

Cipher text = FLHTNI

Example 1.14.3 *Explain Playfair Cipher in detail. Find out cipher text for the following given plain text and key.*

Key = GOVERNMENT

Plaintext = PLAYFAIR **GTU : Winter-17, Marks 7**

Solution :

key = GOVERNMENT plaintext = PLAYFAIR

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

Plaintext = PLAYFAIR → PL AY FA IR

Cipher text = KQYETHBS

Example 1.14.4 *Explain Playfair Cipher in detail. Find out cipher text for the following given plain text and key.*

Key = GOVERNMENT

Plaintext = PLAYFAIR **GTU : Winter-18, Marks 4**

Solution : Key = GUJAR

Plain text = Surgical Strike

G	**U**	**J**	**A**	**R**
B	C	D	E	F
H	K	L	M	N

O	P	Q	S	T
V	W	X	Y	Z

Plain text = Su rg ic al St ri ke

Cipher text = PA UJ UD JM OP GU MC

Example 1.14.5 *Encrypt the message "meet me at the usual place" using the Hill cipher with the key* $\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$

GTU : Winter-18, Marks 7

Solution : Ciphertext = Key × Plaintext mod 26

$$C = KP \bmod 26$$

1st pair from plain text "me" => $\begin{pmatrix} 12 \\ 4 \end{pmatrix}$

$$\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}\begin{pmatrix} 12 \\ 4 \end{pmatrix} \Rightarrow \begin{pmatrix} 9\times12 + 4\times4 \\ 5\times12 + 7\times4 \end{pmatrix} = \begin{pmatrix} 124 \\ 88 \end{pmatrix} \Rightarrow \text{mod } 26 \Rightarrow \begin{pmatrix} 20 \\ 10 \end{pmatrix} \Rightarrow \begin{pmatrix} u \\ k \end{pmatrix}$$

2nd pair from plain text "et"

$$\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}\begin{pmatrix} 4 \\ 19 \end{pmatrix} \Rightarrow \begin{pmatrix} 9\times4 + 4\times19 \\ 5\times4 + 7\times19 \end{pmatrix} = \begin{pmatrix} 112 \\ 153 \end{pmatrix} \Rightarrow \text{mod } 26 \Rightarrow \begin{pmatrix} 8 \\ 23 \end{pmatrix} \Rightarrow \begin{pmatrix} i \\ x \end{pmatrix}$$

Cipher text for "meet" is "ukix".

To get plain text from cipher text, we need to find the inverse of K

$$|A| = (9 \times 7 - 5 \times 4) \Rightarrow 43$$

$$\text{Adj (A)} \Rightarrow \begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \Rightarrow \frac{1}{43}\begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \Rightarrow \frac{1}{17}\begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} (\because 43 \ \% \ 26 = 17)$$

Find the multiplier for 17, using 17 × X = 1 mod 26 => X = 23

$$\begin{pmatrix} 161 & -92 \\ -115 & 207 \end{pmatrix} \Rightarrow \text{Mod } 26 \Rightarrow \begin{pmatrix} 5 & -14 \\ -11 & 25 \end{pmatrix} \Rightarrow \begin{pmatrix} 5 & 12 \\ 15 & 25 \end{pmatrix} (\because \text{Add 26 for } - \text{ve values})$$

$P = CK^{-1}$ = > For the cipher text of "uk"

$$\begin{pmatrix} 5 & 12 \\ 15 & 25 \end{pmatrix}\begin{pmatrix} 20 \\ 10 \end{pmatrix} = \begin{pmatrix} 5\times20 + 12\times10 \\ 15\times20 + 25\times10 \end{pmatrix} \Rightarrow \begin{pmatrix} 220 \\ 550 \end{pmatrix} \text{mod } 26 \Rightarrow \begin{pmatrix} 12 \\ 4 \end{pmatrix} = \begin{pmatrix} m \\ e \end{pmatrix}$$

Hence the plain text is "me".

Corresponding Cipher :

M	E	E	T
20	10	8	23
U	**K**	**I**	**X**

M	E
20	10
U	**K**

A	T
24	3
Y	**D**

T	H	E
17	14	12
R	**O**	**M**

U	S	U	A	L
4	8	22	18	25
E	**I**	**W**	**S**	**Z**

P	L	A	C	E
23	22	8	14	10
X	**W**	**I**	**O**	**K**

A	T
20	13
U	**N**

T	H	E	N
20	1	11	3
U	**B**	**L**	**D**

R	A	T	H	E	R
2	24	3	1	11	21
C	**Y**	**D**	**B**	**L**	**V**

T	H	A	N
10	11	9	3
K	**L**	**J**	**D**

E	I	G	H	T
15	18	4	9	12
P	**S**	**E**	**J**	**M**

O	C	L	O	C	K
4	6	25	23	6	2
E	**G**	**Z**	**X**	**G**	**C**

Example 1.14.6 *Given key*

$$K = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$$

and plaintext = 'ney'. Find out the ciphertext applying Hill Cipher. Is Hill cipher strong against ciphertext only attack or known plaintext attack ? Justify the answer.

GTU : Summer-19, Marks 7

Solution : Key K $= \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$ and ney

$$\begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}\begin{bmatrix} 13 \\ 4 \\ 24 \end{bmatrix} = \begin{bmatrix} 17\times 13+17\times 4+17\times 24 \\ 21\times 13+18\times 4+21\times 24 \\ 2\times 13+2\times 4+19\times 24 \end{bmatrix} = \begin{bmatrix} 697 \\ 849 \\ 490 \end{bmatrix} \bmod 26 = \begin{bmatrix} 21 \\ 17 \\ 22 \end{bmatrix} = \begin{bmatrix} \text{v} \\ \text{r} \\ \text{w} \end{bmatrix}$$

Plaintext = ney

Ciphertext = vrw

Example 1.14.7 *How cryptanalyst can exploit the regularities of the language ? How diagrams can solve this problem ? Use the key "hidden" and encrypt the message "Message" using playfair cipher.* **GTU : Summer-19, Marks 7**

Solution :

- If the cryptanalyst knows the nature of the plaintext (e.g., noncompressed English text), then the analyst can exploit the regularities of the language.
- To see how such a cryptanalysis might proceed. The ciphertext to be solved is
  UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
  VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
  EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English.
- If the message were long enough, this technique alone might be sufficient, but because this is a relatively short message, we cannot expect an exact match.
- In any case, the relative frequencies of the letters in the ciphertext (in percentages) are as follows :

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00	Z 11.67	D 5.00	W 3.33	G 1.67
K 0.00	S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00	U 8.33	V 4.17	T 2.50
I 0.83	N 0.00	O 7.50	X 4.17	A 1.67	J 0.83	R 0.00	M 6.67	

- It seems likely that cipher letters P and Z are the equivalents of plain letters e and t, but it is not certain which is which.
- The letters S, U, O, M and H are all of relatively high frequency and probably correspond to plain letters from the set {a, h, i, n, o, r, s}.
- The letters with the lowest frequencies (namely, A, B, G, Y, I, J) are likely included in the set {b, j, k, q, v, x, z}. There are a number of ways to proceed at this point.
- We could make some tentative assignments and start to fill in the plaintext to see if it looks like a reasonable "skeleton" of a message. A more systematic approach is to look for other regularities.
- For example, certain words may be known to be in the text. Or we could look for repeating sequences of cipher letters and try to deduce their plaintext equivalents. A powerful tool is to look at the frequency of two-letter combinations, known as digrams.

  Key = hidden

  Message = message

Using playfair cipher :

h	i	d	e	n
a	b	c	f	g
k	l	m	o	p
q	r	s	t	u
v	w	x	y	z

Plaintext = me sx sa ge

Ciphertext = od xd qc fn

Example 1.14.8 *Perform encryption in playfair cipher algorithm with plain text as "INFORMATION AND NETWORK SECURITY", keyword is "MONARCHY".*

*(Note : 1. Put j and i both combine as a single field in 5 * 5 matrix).*

GTU : Winter-19, Marks 7

Solution :

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

Plaintext : INFORMATION AND NETWORK SECURITY

IN FO RM AT IO NA ND NE TW OR KS EC UR IT Y**Z**

Ciphertext : GA PH MO RS FA RM RY GM QZ MN IT LE ZM SK WD

University Questions

1. *Write a brief note on hill cipher.* **GTU : Winter-17, Marks 3**
2. *Describe monoalphabetic cipher.* **GTU : Summer-18, Marks 4**
3. *Explain playfair cipher with example.* **GTU : Summer-18, Marks 7**
4. *Explain one time pad cipher with example.* **GTU : Summer-18, Marks 3**
5. *Explain the VERNAM cypher method.* **GTU : Winter-18, Marks 3**
6. *Explain the rail fence cipher. Why a pure transposition cipher is easily recognized ?* **GTU : Summer-19, Marks 3**
7. *Explain one time pad algorithm with example and mention its strength and weakness.* **GTU : Winter-19, Marks 3**
8. *What is the difference between a monoalphabetic cipher and a polyalphabetic cipher ?* **GTU : Winter-19, Marks 4**

1.15 Transposition Techniques

- A transposition cipher rearranges the characters in the plaintext to form the ciphertext. The letters are not changed.
- The rail fence cipher is composed by writing the plaintext in two rows, proceeding down, then across and reading the ciphertext across, then down.

- For example, to enciphere the message “meet me after this party” with a rail fence of depth 2, we write the following :

m	e	m	a	t	r	h	s	a	t
e	t	e	f	e	t	i	p	r	y

- The ciphertext is

  MEMATRHSATETEFETIPRY

- Attacking a transposition cipher requires rearrangement of the letters of the ciphertext.

- A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext.

Plaintext : The book is suitable for self study.

Key : 5 6 4 1 3 2

Key :	5	6	4	1	3	2
Plaintext :	t	h	e	b	o	o
	k	i	s	s	u	i
	t	a	b	l	e	f
	o	r	s	e	l	f
	s	t	u	d	y	

Ciphertext : BSLEDOIFFOUELYESBSUTKTOSHIART.

1.15.1 Difference Between Substitution Cipher and Transposition Cipher

Parameters	Substitution Cipher	Transportation Cipher
Definition	A substitution technique is one in which the letters of plain text are replaced by other letters or number or symbols.	Transposition cipher does not substitute one symbol for another instead it changes the location of the symbols.
Type	Monoalphabetic and Polyalphabetic substitution cipher.	Keyless and keyed transportation cipher.
Changes	Each letter retains its position changes its identity.	Each letter retains its identity but changes its position.
Disadvantages	The last letters of the alphabet which are mostly low frequency tend to stay at the end.	Keys very close to the correct key will reveal long sections of legible plaintext.
Example	Ceaser cipher	Rail fence cipher

Solved Examples

Example 1.15.1 *Encrypt the message "GTU Examination" using the Hill cipher algorithm with the key matrix* $\begin{pmatrix} 5 & 17 \\ 4 & 15 \end{pmatrix}$. *Show your calculations and the result.*

GTU : Winter-19, Marks 7

Solution :

$$\begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}\begin{bmatrix} 6 \\ 19 \end{bmatrix} = \begin{bmatrix} 5\times6+17\times19 \\ 4\times6+15\times19 \end{bmatrix} = \begin{bmatrix} 353 \\ 309 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} 15 \\ 23 \end{bmatrix} = \begin{bmatrix} \text{P} \\ \text{X} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}\begin{bmatrix} 20 \\ 4 \end{bmatrix} = \begin{bmatrix} 5\times20+17\times4 \\ 4\times20+15\times4 \end{bmatrix} = \begin{bmatrix} 168 \\ 140 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} 12 \\ 10 \end{bmatrix} = \begin{bmatrix} \text{M} \\ \text{K} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}\begin{bmatrix} 23 \\ 0 \end{bmatrix} = \begin{bmatrix} 5\times23+17\times0 \\ 4\times23+15\times0 \end{bmatrix} = \begin{bmatrix} 115 \\ 92 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} 11 \\ 14 \end{bmatrix} = \begin{bmatrix} \text{L} \\ \text{O} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}\begin{bmatrix} 12 \\ 8 \end{bmatrix} = \begin{bmatrix} 5\times12+17\times8 \\ 4\times12+15\times8 \end{bmatrix} = \begin{bmatrix} 196 \\ 168 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} 14 \\ 12 \end{bmatrix} = \begin{bmatrix} \text{O} \\ \text{M} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}\begin{bmatrix} 13 \\ 0 \end{bmatrix} = \begin{bmatrix} 5\times13+17\times0 \\ 4\times13+15\times0 \end{bmatrix} = \begin{bmatrix} 65 \\ 52 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} 13 \\ 0 \end{bmatrix} = \begin{bmatrix} \text{N} \\ \text{A} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}\begin{bmatrix} 19 \\ 8 \end{bmatrix} = \begin{bmatrix} 5\times19+17\times8 \\ 4\times19+15\times8 \end{bmatrix} = \begin{bmatrix} 231 \\ 196 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} 23 \\ 14 \end{bmatrix} = \begin{bmatrix} \text{X} \\ \text{O} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}\begin{bmatrix} 14 \\ 13 \end{bmatrix} = \begin{bmatrix} 5\times14+17\times13 \\ 4\times14+15\times13 \end{bmatrix} = \begin{bmatrix} 286 \\ 251 \end{bmatrix} \text{mod } 26 = \begin{bmatrix} 0 \\ 17 \end{bmatrix} = \begin{bmatrix} \text{A} \\ \text{R} \end{bmatrix}$$

Plain text : GTU Examination

Cipher text : PX MK LO OM NA XO AR

University Questions

1. *Write differences between substitution techniques and transportation techniques.* **GTU : Summer-17, Marks 3**
2. *Explain transposition techniques with appropriate example.* **GTU : Winter-17, Marks 7**
3. *Explain columnar transposition cipher technique.* **GTU : Summer-18, Marks 4**

1.16 Short Questions and Answers

Q.1 What is meant by threat ?

Ans. : 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.

Q.2 What is encipherment ?

Ans. : 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.

Q.3 What is a passive attack ?

Ans. : Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. Two types of passive attacks are release of message contents and traffic analysis.

Q.4 What is an active attack ?

Ans. : An active attack involves some modification of the data stream or the creation of a false.

Q.5 What are the aspects of information security ?

Ans. : There are three aspects of the information security. i.e. security attack, security mechanism, security service.

Q.6 Define an attack.

Ans. : An attack on system security that derives from an intelligent threat : that is an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system.

Q.7 What are the essential ingradients of a symmetric cipher ?

Ans. : A symmetric encryption scheme has five ingradients : Plaintext, Encryption algorithm, Secret key, Ciphertext, Decryption algorithm.

Q.8 Define the monoalphabetic cipher.

Ans. : 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.

Q.9 Define the playfair cipher.

Ans. : The playfair cipher treats the diagrams in the plaintext as single units and translates these units into ciphertext diagrams. This algorithm is based on the use of a 5 by 5 matrix of letters constructed using keyword.

Q.10 What is product cipher ?

Ans. : Product cipher has the performance of two or more basic ciphers in sequence is such a way that the final result or product is cryptographically stronger than any of the component ciphers.

Q.11 List out the problems of one time pad ?

Ans. : Problem with one time pad is that of making large quantities of random keys. It also makes the problem of key distribution and protection.

Q.12 Define symmetric encryption.

Ans. : In symmetric encryption, sender and receiver use same key for encryption and decryption.

1.17 Multiple Choice Questions

Q.1 The original message is called as __________.

a Ciphertext b Plaintext c Cryptography d encryption

Q.2 The process of converting plaintext to ciphertext is called as __________.

a Encryption b Decryption c Substitution d Transposition

Q.3 Interception, interruption, __________ and fabrication are the system security threats.

a Traffic analysis b Masquerade c Replay d modification

Q.4 Which of the following is NOT types of active attack.

a Masquerade b Replay

c Traffic analysis d Modification of message

Q.5 __________ attacks are called as masquerade attacks.

a Interception b interruption c modification d fabrication

Q.6 __________ attacks are very difficult to detect because they do not involve any alternation of data.

a Active b Passive

c Active and passive d None of these

Q.7 The process of trying to break any cipher text message to obtain the original plain text message itself is called as __________.

a Cryptanalyst b Cryptography

c Cryptology d cryptanalysis

Q.8 The process of converting the ciphertext into plaintext is called __________.

a Encryption
b Decryption
c Substitution
d Transposition

Q.9 A symmetric encryption model has __________ ingredients.

a four
b three
c five
d six

Q.10 The one time pad is susceptible to a __________.

a Chosen plain text attack
b known plain text attack
c Known cipher text attack
d None of these

Q.11 Secret key cryptography is also as __________.

a Symmetric key cryptography
b asymmetric key cryptography
c private key cryptography
d quantum cryptography

Q.12 In cryptography, what is cipher?

a Encrypted message
b Decrypted message
c algorithm for performing encryption and decryption
d both (a) and (b)

Q.13 Security ------------ is a process that is designed to detect, prevent, or recover from a security attack.

a Attack
b mechanism
c service
d All of these

Q.14 Security ------- is any action that compromises the security of information owned by an organization.

a Service
b mechanism
c attack
d All of these

Q.15 ------------- attack attempts to alter system resources or affect their operation.

a Passive
b active
c both (a) and (b)
d None

Q.16 Two types of passive attacks are the ----------------, ------------------.

a Masquerade, replay
b Masquerade, traffic analysis
c Release of message contents, Replay
d Release of message contents, traffic analysis

Q.17 Which of the following is NOT an active attack ?

a Masquerade
b Replay
c Traffic analysis
d denial of service

Q.18 -------- involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

a Masquerade
b replay
c traffic analysis
d denial of service

Q.19 Access control means ---------------------.

a The protection of all user data on a connection
b The assurance that the communicating entity is the one that it claims to be.
c The protection of data from unauthorized disclosure.
d The prevention of unauthorized use of a resource

Q.20 A cyclic group is always --------- and may be finite or infinite.

a abelian
b finite
c order
d None

Q.21 A ring is a set of elements with two binary operations, called ----------- and ------------------.

a addition and subtraction
b addition and division
c multiplication and division
d addition, multiplication

Q.22 If a is an integer and n is a positive integer, we define a mod n to be the remainder when a is divided by n. The integer n is called the -------------

a ring
b modulus
c graph
d fields

Q.23 ------------- simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect.

a Denial of service
b Masquerade
c Replay
d Modification of messages

Q.24 A loss of ------------ is the unauthorized disclosure of information.

a integrity
b availability
c authentication
d confidentiality

Q.25 Which of the following are the security requirements triad ?

a Confidentiality
b Integrity
c Availability
d All of these

Q.26 The ----------- are based on modification of the original message in some manner, or on creation of a false message.

a passive attack
b release of message content attack
c traffic analysis attack
d active attack

Q.27 Which one of the following is active attack ?

a Masquerade
b Traffic analysis
c Eavesdropping
d Shoulder surfing

Q.28 Which of the following is passive attack?

a Relay attack
b Masquerade
c Traffic analysis
d Denial of Service

Q.29 -------- attacks are very difficult to detect because they do not involve any alternation of data.

a Active
b Passive
c Active and passive
d None of these

Q.30 -------------- attacks are called as masquerade attacks.

a Interception
b Interruption
c Modification
d Fabrication

Q.31 Interception, interruption, --------------- and fabrication are the system security threats.

a traffic analysis
b masquerade
c replay
d modification

Q.32 --------------- prevents either sender or receiver from denying a transmitted message.

a Nonrepudiation
b Replay
c Fabrication
d Masquerade

Answer Keys for Multiple Choice Questions

Q.1	b	**Q.2**	a	**Q.3**	d
Q.4	c	**Q.5**	b	**Q.6**	b
Q.7	d	**Q.8**	b	**Q.9**	c
Q.10	d	**Q.11**	a	**Q.12**	c
Q.13	b	**Q.14**	c	**Q.15**	b
Q.16	d	**Q.17**	c	**Q.18**	b
Q.19	d	**Q.20**	a	**Q.21**	d
Q.22	b	**Q.23**	d	**Q.24**	d
Q.25	d	**Q.26**	d	**Q.27**	a
Q.28	c	**Q.29**	b	**Q.30**	b
Q.31	d	**Q.32**	a		

❑❑❑

Notes

2 Stream Ciphers and Block Ciphers

Syllabus

Stream ciphers and block ciphers, Block Cipher structure, Data Encryption standard (DES) with example, strength of DES, Design principles of block cipher, AES with structure, its transformation functions, key expansion, example and implementation.

Contents

2.1 Stream Ciphers

- A block cipher operates on blocks of data.
- Algorithm breaks the plaintext into blocks and operates on each block independently.
- Usually blocks are 8 or 16 bytes long.
- Security of block ciphers depends on the design of the encryption function.
- Software implementations of block ciphers run faster than software implementation of the stream ciphers.
- Errors in transmitting one block generally do not affect other blocks.
- Each block is enciphered independently, using the same key, identical plaintext blocks produce identical ciphertext blocks.
- Suppose that plaintext is 227 bytes long and the cipher you are using operates on 16-byte blocks.
- Algorithm grabs the first 16-bytes of data, encrypts them using the key table.
- Algorithm produces 16-bytes of ciphertext.
- After first block, algorithm takes next block.
- The key table does not change from block to block.

$$\text{Plaintext} = 227 \text{ bytes}$$

$$\text{Block size} = 16 \text{ bytes} = \frac{227}{16} = 14 \text{ blocks plus } 3 \text{ bytes}$$

- Algorithm encrypts 14 blocks and 3 bytes remain.
- For encrypting last 3 bytes data padding is used.
- Extra bytes are added to make the last block size to 16 bytes.
- Whoever decrypts the ciphertext must be able to recognize the padding.
- One problem with block ciphers is that if the same block of plaintext appears in two places, it encrypts to the same ciphertext.
- To avoid having these kinds of copies in the ciphertext, feedback modes are used.
- Cipher block chaining does not require the extra information to occupy bit spaces, so every bit in the block is part of the message.
- Before a plaintext block is enciphered, that block is XOR'ed with preceding ciphertext block.
- In addition to the key, this technique requires an initialization vector to XOR the initial plaintext block.

- For decrypting the data, copy a block of ciphertext, decrypt it and XOR the result with the preceding block of ciphertext.
- Taking E_K to be the encipherment algorithm with key K and I to be the initialization vector, the cipher block chaining technique is

$$C_o = E_K(m_o \oplus I)$$

$$C_i = E_K(m_i \oplus C_{i-1}) \quad \text{for } i > 0$$

2.1.1 Advantages and Disadvantage of Block Cipher

Advantages :

1. High diffusion
2. Immunity to insertation of symbols.

Disadvantages :

1. Slowness of encryption
2. Error propagation.

2.2 Block Ciphers

GTU : Winter-14, 17, 18, 19

- Stream cipher algorithms are designed to accept a crypto key and a stream of plaintext to produce a stream of ciphertext.
- Fig. 2.2.1 shows the stream cipher.
- Stream cipher is similar to a one time pad.
- A stream cipher encrypts smaller block of data, typically bits or bytes.
- A key stream generator outputs a stream of bits $K_1, K_2, K_3 \ldots\ldots K_i$.
- This key stream is XORed with a stream of plaintext bits $P_1, P_2, P_3 \ldots\ldots P_i$ to produce the stream of ciphertext bits.

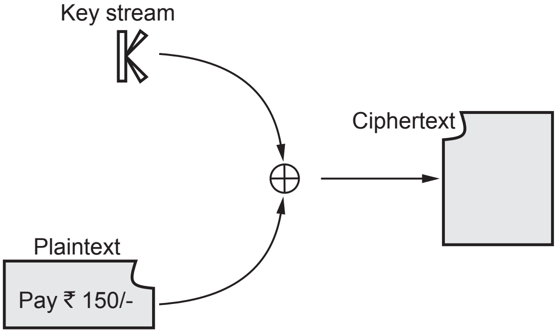


Fig. 2.2.1 Stream cipher

$$C_i = P_i \oplus K_i$$

- At the description end, the ciphertext bits are XORed with an identical key stream to recover the plaintext bits.

$$P_i = C_i \oplus K_i$$

- The system security depends entirely on the insides of the keystream generator.

2.2.1 Advantages and Disadvantages of Stream Cipher

Advantages :

1. Speed of transformation
2. Low error propagation.

Disadvantages :

1. Low diffusion
2. Susceptibility to malicious insertation and modifications.

2.2.2 Comparison between Stream and Block Cipher

Sr. No.	Stream cipher	Block cipher
1.	Stream ciphers operate on smaller units of plaintext.	Block ciphers operate on larger block of data.
2.	Faster than block cipher.	Slower than stream cipher.
3.	Stream cipher processes the input element continuously producing output one element at a time.	Block cipher processes the input one block of element at a time, producing an output block for each input block.
4.	Requires less code.	Requires more code.
5.	Only one time of key use.	Reuse of key is possible.
6.	Ex. - One time pad	Ex. - DES
7.	Application - SSL (secure connections on the web.)	Application - Database, file encryption.
8.	Stream cipher is more suitable for hardware implementation.	Easier to implement in software.

University Questions

1. *Define block cipher. Explain design principles of block cipher.* **GTU : Winter-14, Marks 7**
2. *What are the differences between stream cipher and block cipher ?* **GTU : Winter-17, Marks 3**
3. *Differentiate block cipher and a stream cipher.* **GTU : Winter-18, Marks 3**
4. *Differentiate block cipher and stream cipher algorithm with example.* **GTU : Winter-19, Marks 4**

2.3 Block Cipher Structure : Feistel Cipher

- A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on

one half of the data being processed, followed by a permutation that interchanges the two halves.

- The original key is expanded so that a different key is used for each round. Many symmetric block encryption algorithms in current use are based on a structure referred toas a Feistel block cipher

2.3.1 Feistel Cipher

- Fig. 2.3.1 shows the classical Feistel network. The inputs to the encryption algorithm are a plaintext block of length 2w bits and a key K. The plaintext block is divided into two halves i.e. Left (L_0) and Right (R_0).

Parameters and design features

Following parameters are considered :

1. Block size
2. Key size
3. Number of rounds
4. Subkey generation algorithms
5. Round function
6. Fast software encryption / decryption.
7. Ease of analysis

1. Security depends upon the block size. Larger **block size** gives greater security but encryption / decryption speed is reduced normal. Block size is 64-bit and AES uses 128-bit block size.
2. Greater security is achieved by using longer **key size**. Because of longer key size, again speed of algorithm decreases. Key sizes of 64 bits or less are now widely considered to be inadequate and 128 bits have become a common size.
3. **Number of rounds** are 16 in most of the algorithm. In Feistel cipher, single round offers insufficient security and multiple rounds offer greater security.
4. In **subkey generation algorithm**, greater complexity leads to greater difficulty of cryptanalysis.
5. **Round function** is again greater complexity for greater resistance to cryptanalysis.
6. **Fast software encryption / decryption** : The speed of execution of the algorithm becomes a concern.
7. **Ease of analysis** : There is great benefit in making the algorithm easy to analysis.

Decryption Algorithm

- Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order.

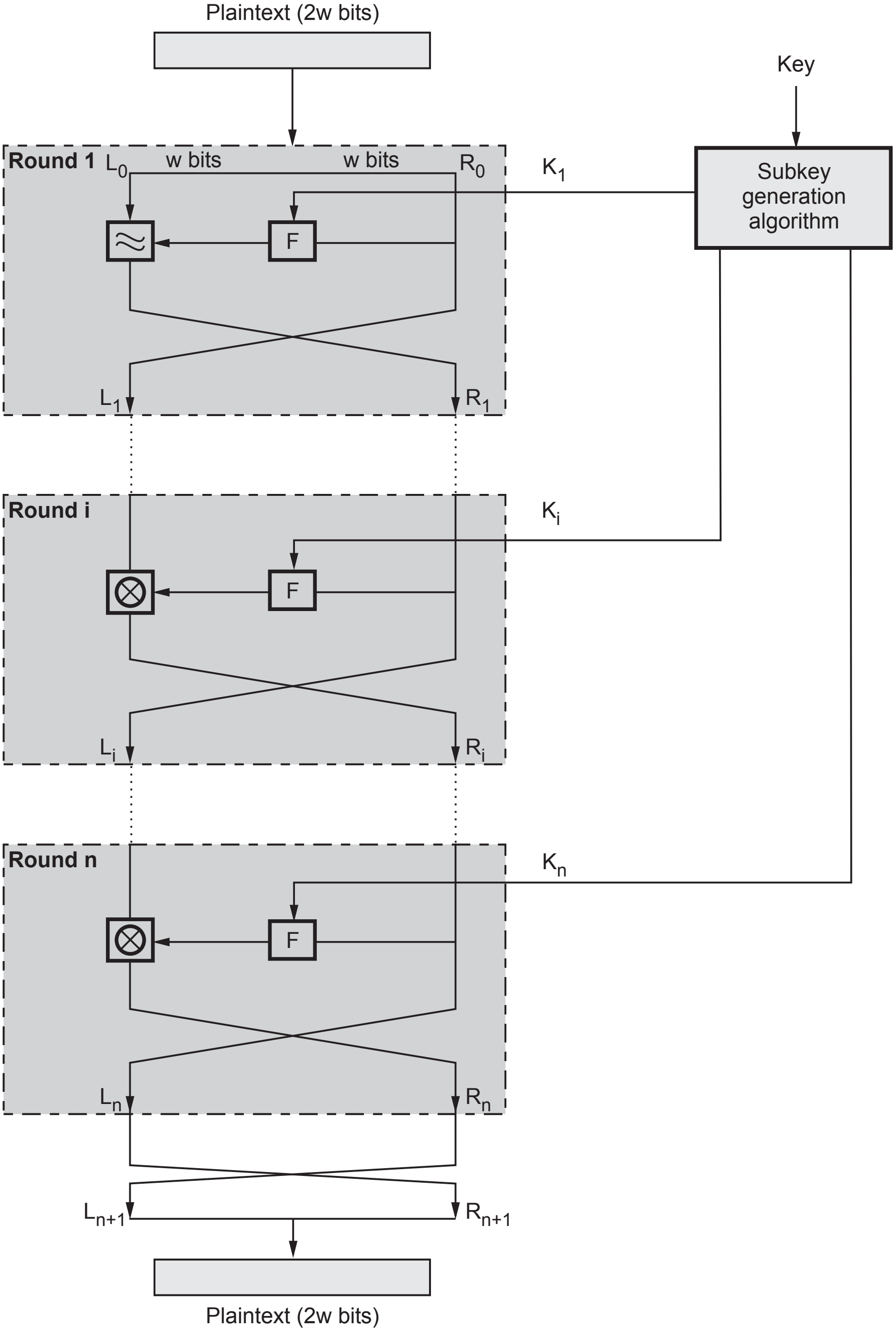


Fig. 2.3.1 Classical feistel network

- The output of the first round of the decryption process is equal to a 32 bit swap of the input to the 16th round of the encryption process.

- Consider the encryption process :

$$LE_{16} = RE_{15}$$

$$RE_{16} = LE_{15} \times F(RE_{15}, K_{16})$$

- On the decryption side

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$

$$RD_1 = LD_0 \times F(RD_0, K_{16}) = RE_{16} \times F(RE_{15}, K_{16})$$

$$= [(LE_{15} \times F(RE_{15}, K_{16})] \times F(RE_{15}, K_{16})$$

∴ We have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$

- For the i^{th} iteration of the encryption algorithm,

$$LE_i = RE_{i-1}$$

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

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

2.4 Simple DES

- Takes an 8-bit block plaintext, a 10-bit key and produces an 8-bit block of cipher-text.
- Decryption takes the 8-bit block of cipher-text, the same 10-bit key and produces the original 8-bit block of plaintext.
- It was designed as a test block cipher for learning about modern cryptanalytic techniques such as linear cryptanalysis, differential cryptanalysis and linear-differential cryptanalysis.
- The same key is used for encryption and decryption. Though, the schedule of addressing the key bits is altered so that the decryption is the reverse of encryption.
- An input block to be encrypted is subjected to an initial permutation IP. Then, it is applied to two rounds of key-dependent computation. Finally, it is applied to a permutation which is the inverse of the initial permutation.

$$\textbf{plaintext} = \mathbf{b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8}$$

$$\textbf{key} = \mathbf{k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9 k_{10}}$$

Subkey generation

- First, produce two subkeys K_1 and K_2:

$$\mathbf{K_1 = P8(LS_1(P10(key)))}$$

$$K_2 = P8(LS_2(LS_1(P10(key))))$$

where P8, P10, LS_1 and LS_2 are bit substitution operators.

- For example, P10 takes 10 bits and returns the same 10 **bits in a different order :**

$$P10(k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9 k_{10}) = k_3 k_5 k_2 k_7 k_4 k_{10} k_1 k_9 k_8 k_6$$

It's convenient to write such bit substitution operators in this notation :

P10 : (10 bits to 10 bits)

3	5	2	7	4	10	1	9	8	6

P8 : (10 bits to 8 bits)

6	3	7	4	8	5	10	9

LS_1 ("left shift 1 bit" on 5 bit words) : 10 bits to 10 bits

2	3	4	5	1	7	8	9	10	6

LS_2 ("left shift 2 bit" on 5 bit words) : 10 bits to 10 bits

3	4	5	1	2	8	9	10	6	7

Encryption

- The plain text is split into 8-bit blocks; each block is encrypted separately. Given a plaintext block, the cipher text is defined using the two subkeys K_1 and K_2, as follows:

$$\textbf{Ciphertext} = IP^{-1}(\ f_{K_2}(\ SW(f_{K_1}(\ IP(\ plaintext\ )\ )\ )\ )\ )$$

where :

Initial Permutation (IP) : 8 bits to 8 bits

2	6	3	1	4	8	5	7

IP^{-1} (8 bits to 8 bits)

4	1	3	5	7	2	8	6

Switch (SW) : 8 bits to 8 bits

5	6	7	8	1	2	3	4

and $f_K(\ )$ is computed as follows.

We write exclusive-or (XOR) as +.

$$\mathbf{f_K(\ L,\ R\ )\ \ =\ \ (\ L\ +\ F_K(R)\ ,\ R\ )}$$

$$\mathbf{F_K(R)\ \ =\ \ P4\ (\ \ S0(\ lhs(\ EP(R)+K\ ))\ \ ,\ \ S1(\ rhs(EP(R)+K\ ))\ \ )}$$

4 bits to 8 bits

4	1	2	3	2	3	4	1

P4 (4 bits to 4 bits)

2	4	3	1

lhs (8 bits to 4 bits)

1	2	3	4

rhs (8 bits to 4 bits)

5	6	7	8

S0($b_1 b_2\ b_3 b_4$) = The [$b_1 b_4, b_2 b_3$] cell from the "S-box" S0 below, and similarly for S1.

S0

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	0	3

S1

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	1	1	0	3

- **Algorithm :**

The block of 12 bits is written in the form L_0R_0, where L_0 consists of the first 6 bits and R_0 consists of the last 6 bits. The i^{th} round of the algorithm transforms an input $L_{i-1}\ R_{i-1}$ to the output L_iR_i using an 8-bit K_i derived from K.

- Fig. 2.4.1 shows one round of a Feistel system.

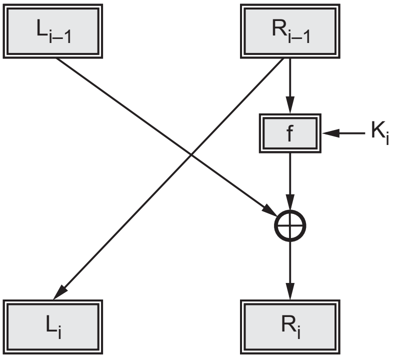


Fig. 2.4.1 One round of a Feistel system

- The output for the i^{th} round is found as follows :

  $\boldsymbol{L_i = R_{i-1}}$ and $\boldsymbol{R_i = L_{i-1} \oplus f(R_{i-1}, K_i)}$
- This operation is performed for a certain number of rounds, say n, and produces $L_n R_n$.
- The ciphertext will be $R_n L_n$.
- Encryption and decryption are done the same way except the keys are selected in the reverse order.
- The keys for encryption will be K_1, K_2. ...... K_n and for decryption will be K_n, ... K_{n-1} ...... K_1.
- **Function $\mathbf{f(R_{i-1}, K_i)}$:** The function $f(R_{i-1}, K_i)$, depicted in the Fig. 2.4.2 below, is described in following steps.

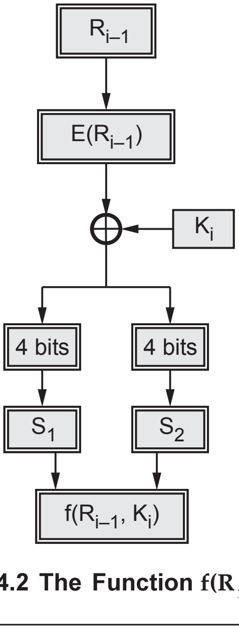


Fig. 2.4.2 The Function $\mathbf{f(R_{i-1}, K_i)}$

1. The 6-bits are expanded using the following expansion function. The expansion function takes 6-bit input and produces an 8-bit output. This output is the input for the two S-boxes.

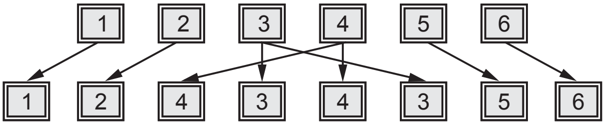


Fig. 2.4.3 The expansion function, E(R_{i-1})

2. The 8-bit output from the previous step is Exclusive-ORed with the key K_i
3. The 8-bit output is divided into two blocks. The first block consists of the first 4 bits and the last four bits make the second block. The first block is the input for the first S-box (S1) and the second block is the input for the second S-box (S2).
4. The S-boxes take 4-bits as input and produce 3-bits of output. The first bit of the input is used to select the row from the S-box, 0 for the first row and 1 for the second row. The last 3 bits are used to select the column.
5. The output from the S-boxes is combined to form a single block of 6-bits. These 6 bits will be the output of the function $f(R_{i-1}, K_i)$.

Example : Let the output from the expander function be 11010010.

Solution : 1101 will be the input for the S1 box and 0010 will be the input for the S2 box. The output from the S1 box will be 111, the first of the input is 1 so select the second row and 101 will select the 6th column. Similarly the output from the S2 box will be 110. In above example we have the S1 output 111 and S2 output 110. So the output for the function

$f(R_{i-1}, K_i)$ will be 111110, the S1 output followed by the S2 output.

2.5 Data Encryption Standard

GTU : Summar-18, Winter-18,19

- DES Encryption standard (DES) is **a symmetric key block cipher** published by the National Institute of Standards and Technology (NIST).
- It encrypts data in 64-bit block.
- DES is symmetric key algorithm : The same algorithm and key is used for both encryption and decryption.
- Key size is 56-bit.
- The encryption process is made of two permutations i.e. P-boxes, which is called initial and final permutation.

- DES uses both transposition and substitution and for that reason is sometimes referred to as a **product cipher**. Its input, output and key are each 64-bits long. The sets of 64-bits are referred to as **blocks**.
- The cipher consists of 16 rounds or iterations. Each rounds uses a separate key of 48-bits.
- Fig. 2.5.1 shows DES encryption algorithm. First, the 64-bit plaintext passes through an Initial Permutation (IP) that rearranges the bits to produce the permuted input.

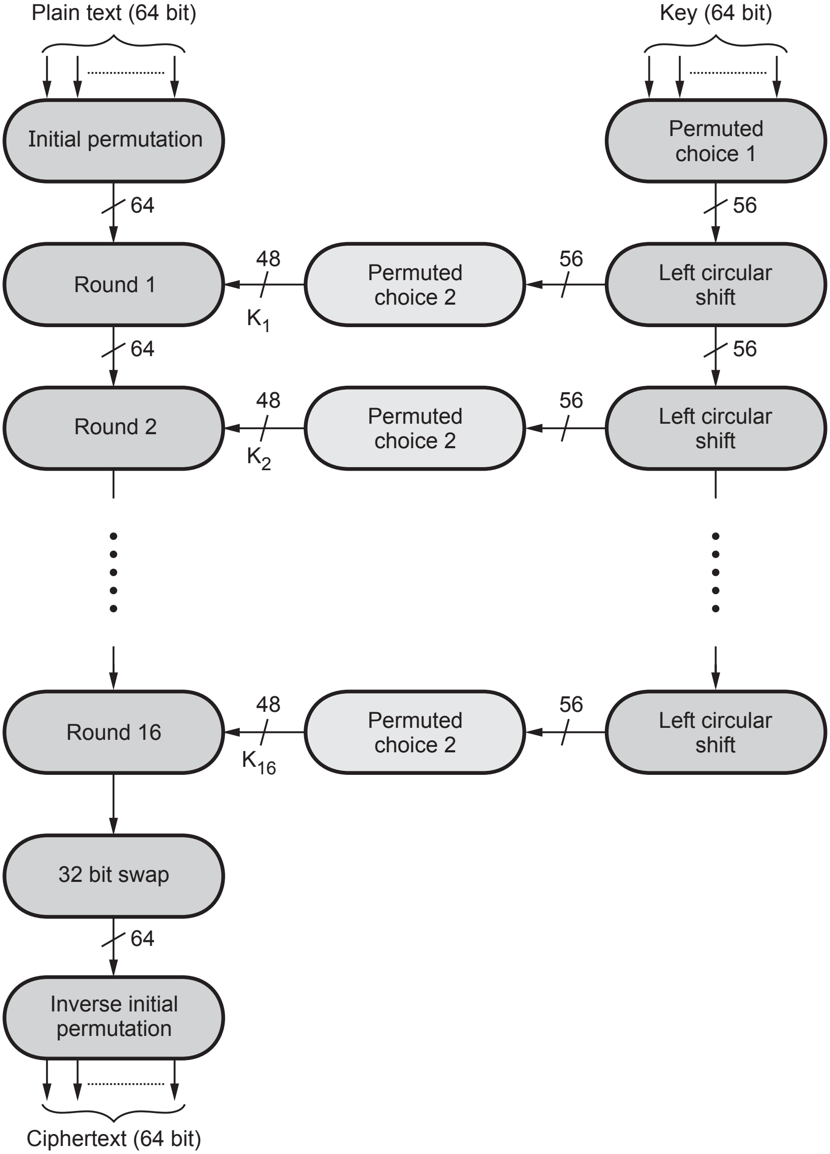


Fig. 2.5.1 DES encryption algorithm

- Then there is a phase consisting of 16 rounds of the same function, which involves both permutation and substitution functions.
- The output of the 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. At last, the pre-output is passed through a permutation (IP^{-1}) that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

Initial permutation

- Table shows the initial permutation and its inverse. The input to a table consist of 64-bits numbered from 1 to 64.
- The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64. Each entry in the permutation table indicates the positon of a numbered input bit in the output, which also consists of 64-bits.

Initial Permutation (IP) table

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	2.1	13	5
63	55	47	39	31	23	15	7

Inverse Initial Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

2.5.1 Details of Single Round

- Fig. 2.5.2 shows single round of DES algorithm. The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L and R.
- The overall processing at each round can be summarised in the following formulae :

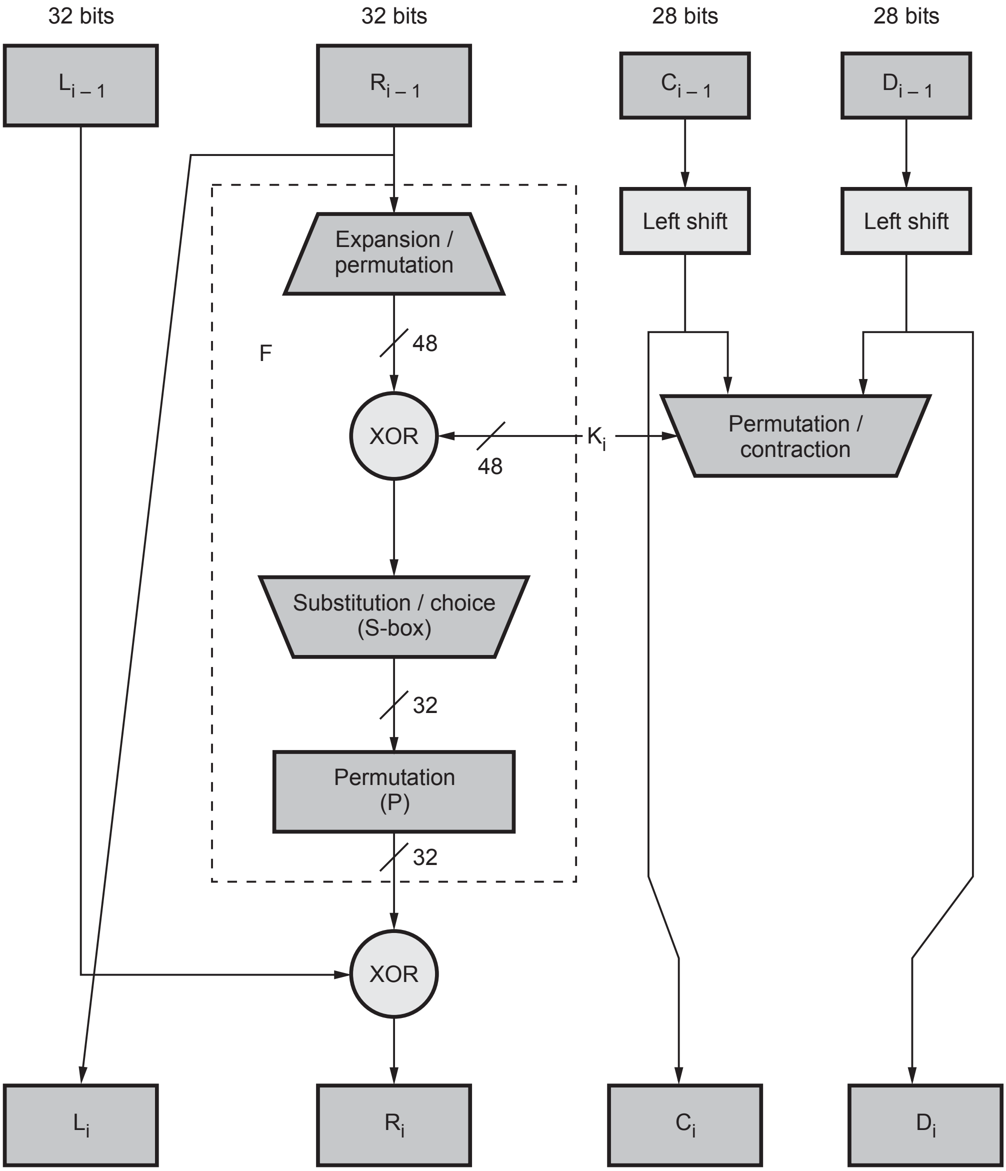


Fig. 2.5.2 Single round of DES algorithm

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

$$R_i = L_{i-1} \text{ X } F(R_{i-1}; K_i)$$

The left output (L_i) is simply copy of the right input (R_{i-1}). The right output (R_i) is the XOR of left input (L_{i-1}) and right input (R_{i-1}) and key for this stage is K_i. In this stage, the substitution and permutation both functions are used.

- Fig. 2.5.3 shows role of S-boxes in the function F. It consists of set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.

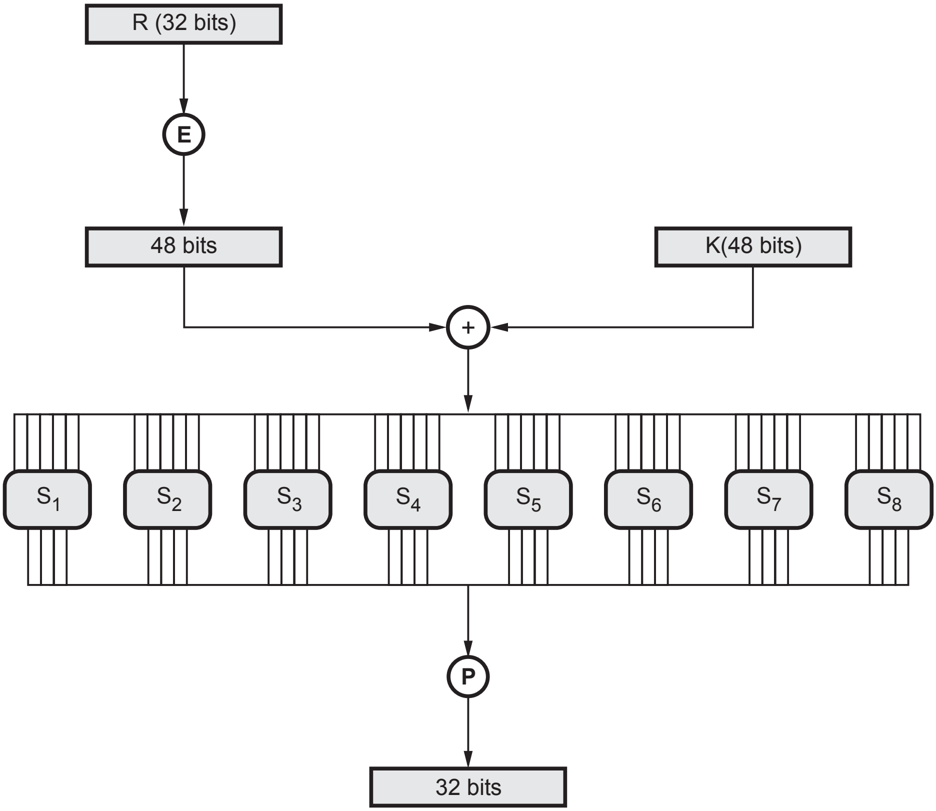


Fig. 2.5.3 S-boxes in the function (F)

- The 48 bit input block is divided into 8 subblocks and each subblock is given to a S-box. The S-box transforms the 6 bit input into a 4 bit output.
- First and last bits of the input to box S_i form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for S_i. Two bits can store any decimal number between 0 and 3. This specifies the row number. The middle four bits select one of the sixteen columns.

- Following table gives the S-box value for DES

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

- Fig. 2.5.4 shows the selection of an entry in a S-box based n the 6-bit input. For example, in S_2, for input 101101, the row is 11 and the column is 0110. The value in row 3, column 6 which select row 3 and column 6 of S_2 box. The output is 4.

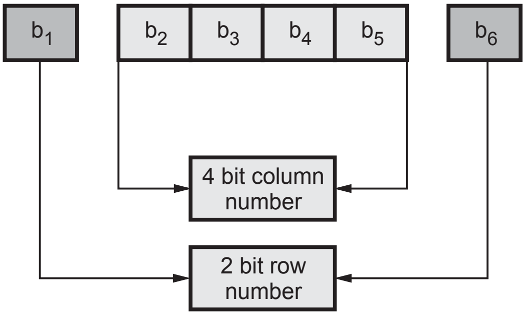


Fig. 2.5.4 Selecting entry in S-box

2.5.2 Key Generation

- 64-bit key is used as input to the algorithm. The initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key.
- From 56-bit key, a different 48-bit subkey is generated during each round using a process called as key transformation.
- The resulting 56-bit key is then treated as two 28-bit quantities, labeled C_0 and D_0. At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift, or rotation, of 1 or 2-bits.
- These shifted values serve as input to the next round. They also serve as input to Permuted choice Two, which produces a 48-bit output that serves as input to the function $F(R_{i-1}, K_i)$.

2.5.3 DES Encryption

- A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is inverse of the initial permutation IP.
- The key-dependent computation can be simply defined in terms of a function *f*, called the cipher function, and a function KS, called the key schedule.
- Given two blocks L and R of bits, LR denotes the block consisting of the bits of L followed by the bits of R.
  1. **Initial permutation :** The 64-bits of the input block to be enciphered are first subjected to the permutation, called the initial permutation.
  2. **Key dependent computation :** The computation which uses the permuted input block as its input to produce the pre-output block consists. Cipher function *f* which operates on two blocks, one of 32-bits and one of 48-bits, and produces a block of 32-bits. Let the 64 bits of the input block in an iteration consist of a 32-bit block L followed by a 32-bit block R. Using the notation defined in the introduction the input block is then LR. Let K be a block of 48 bits chosen from the 64-bit key. Then the output L' R' of an iteration with input LR is defined by :

$$\left.\begin{aligned} L' &= R \\ R' &= L\ (+)\ f(R, K) \end{aligned}\right\} \quad \text{... (2.5.1)}$$

where (+) denotes bit-by-bit addition modulo 2.

As before, let the permuted input block be LR. Finally, let L_0 and R_0 be respectively L and R and let L_n and R_n be respectively L' and R' of equation (2.4.1) hence L and R are respectively L_{n-1} and R_{n-1} and K is K_n i.e. when *n* is in the range from 1 to 16,

$$\text{Then } L_n = R_{n-1}$$

$$R_n = L_{n-1}\ (+)\ f(R_{n-1}, K_n)T$$

The pre-output block is then $R_{16}L_{16}$.

3. **Key schedule :** Key generation techniques is shown in the Fig. 2.5.5 (See Fig. 2.5.5 on next page).

The input of the first iteration of the calculation is the permuted input block. If L' R' is the output of the 16th iteration then R'L' is the pre-output block. At each iteration a different block K of key bits is chosen from the 64-bit key designated by KEY. Let KS be a function which takes a integer *n* in the range from 1 to 16 and a 64-bit block KEY as input and yields as output a 48-bit block K_n which is a permuted selection of bits from KEY i.e.

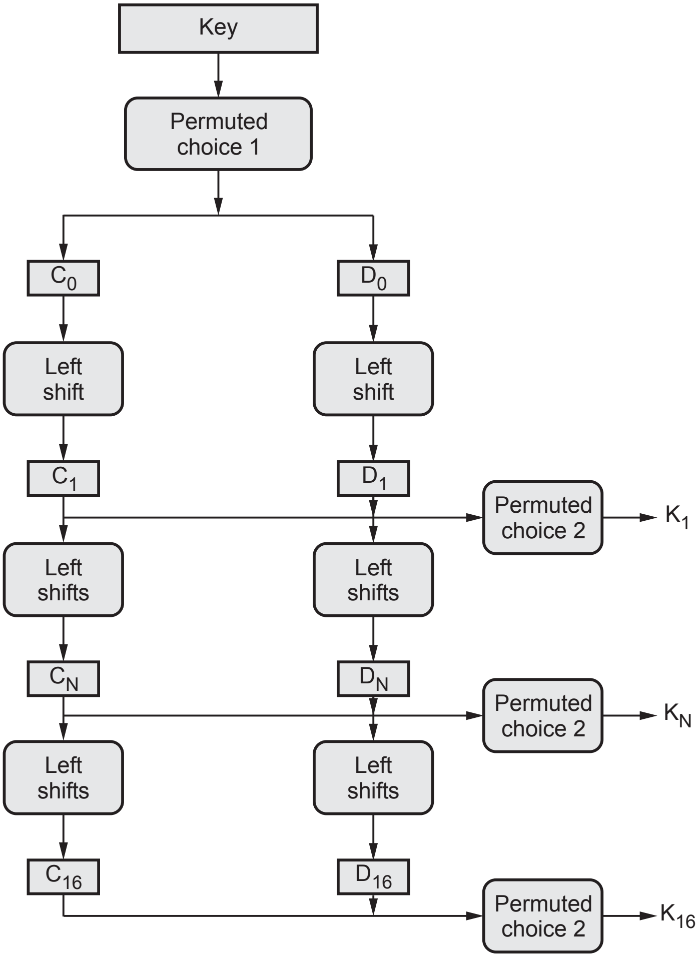


Fig. 2.5.5 Key generation techniques

$$K_n = KS(n, KEY)$$

with K_n determined by the bits in 48 distinct bit positions of KEY. KS is called the key schedule.

2.5.4 DES Decryption

The permutation IP^{-1} applied to the pre-output block is the inverse of the initial permutation IP applied to the input. Consequently, to decipher it is only necessary to apply the very same algorithm to an enciphered message block, taking care that at each iteration of the computation the same block of key bits K is used during decipherment as was used during the encipherment of the block only in a reverse order. For the decipherment calculation with $R_{10}L_{10}$ as the permuted input, K_{10} is used in the first iteration, K_{10} in the second, and so on, with K, used in the 16th iteration.

2.5.5 DES Weak Keys

- With many block ciphers there are some keys that should be avoided, because of reduced cipher complexity.
- These keys are such that the same sub-key is generated in more than one round, and they include :
  1. **Weak keys :** The same sub-key is generated for every round and DES has 4 weak keys.
  2. **Semi-weak keys :** Only two sub-keys are generated on alternate rounds and DES has 12 of these (in 6 pairs).
  3. **Demi-semi weak keys :** Have four sub-keys generated.
- None of these cause a problem since they are a tiny fraction of all available keys however they MUST be avoided by any key generation program.

2.5.6 Advantages of DES

1. As 56-bit keys are used there are 70 quadrillion possible key values and hence a specific key cannot be identified easily.
2. As the length of the key is increased the security provided by the algorithm also increases.
3. The security of the DES algorithm resides in the key.

2.5.7 Disadvantages of DES

1. As it is a symmetric algorithm both sender and receiver must have same key, there is a possibility that the key is intercepted.
2. The design of S boxes makes it susceptible to linear cryptanalysis attack.
3. It is susceptible to differential cryptanalysis attack and brute force attack taking advantage of which DES crackers have been designed.
4. It has certain weak keys which generate the same key for all cycles of the algorithm like when all key bits are either 0s or 1s or if one half of the key bits are 0s or 1s. They are 0000000 0000000, 0000000 fffffff, fffffff 0000000, fffffff fffffff.
5. Some initial keys produce only two subkeys while some produce only four. They are called possible weak keys.

Possible techniques for improving DES

- Multiple enciphering with DES
- Extending DES to 128-bit data paths and 112-bit keys
- Extending the key expansion calculation.

2.5.8 Block Cipher Design Principles

The criteria for the **S-boxes** are as follows :

1. No output bit of any S-box should be too close a linear function of the input bits.
2. Each row of an S-box should include all 16 possible output bit combinations.
3. If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.
4. If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.
5. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.
6. For any non zero 6-bit difference betwen inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference.

Criteria for **permutation P** are as follows.

1. The four output bits from each S-box at round i are distributed so that two of them affect middle bits of round (i + 1) and the other two affect end bits.
2. The four output bits from each S-box affect six different S-boxes on the next round, and no two affect the same S-box.
3. For two S-boxes j, k, if an output bit from S_j affects a middlle bits of S_{tock} on the next round, then an output bit from S_k cannot affect a middle bit of S_j.

University Questions

1. *Write a short note on DES.* **GTU : Summer-18, Marks 7**
2. *Discuss in detail encryption and decryption process of DES.* **GTU : Winter-18, Marks 7**
3. *Draw block diagram to show broad level steps in DES and also give steps of one round in DES with another diagram.* **GTU : Winter-19, Marks 7**

2.6 Confusion and Diffusion

GTU : Winter-18, Summer-19

Diffusion

- Diffusion is making output dependent on previous input (plain/cipher-text). Ideally, each output bit is influenced by every previous input bit.
- These are measures to thwart cryptanalysis based on statistical analysis. In diffusion, the statistical structure of the plaintext is dissipated into long range statistics of the cipher-text.

- This is achieved by having each plaintext letter affect the value of many cipher-text digits, which is equivalent to saying that each cipher-text digit is affected by many plaintext digits.
- The letter frequencies in the cipher-text will be more nearly equal than in the plaintext.

Confusion

- In Shannon's original definitions, confusion makes the relation between the key and the cipher-text as complex as possible. Confusion is making the output dependent on the key. Ideally, every key bit influences every output bit. Confusion tries to hide the connection between the cipher-text and the secret key.
- Confusion seeks to make the relationship between the statistics of the cipher-text and the value of the encryption key as complex as possible. This is achieved by the use of a complex substitution algorithm. These operations became the cornerstone of modern block cipher design.

2.6.1 Distinguish between Diffusion and Confusion

No.	Diffusion	Confusion
1.	Diffusion hides the relation between the ciphertext and the plaintext.	Confusion hides the relation between the ciphertext and key.
2.	If a single symbol in the plaintext is changed, several or all symbols in the ciphertext will also be changed.	If a single bit in the key is changed, most or all bits in the ciphertext will also be changed.
3.	In diffusion, the statistical structure of the plain text is dissipated into long-range statistics of the cipher text. This is achieved by permutation.	In confusion, the relationship between the statistics of the cipher text and the value of the encryption key is made complex. It is achieved by substitution.

University Questions

1. *Explain the difference between diffusion and confusion.* **GTU : Winter-18, Marks 4**
2. *Which two methods are used to frustrate statistical cryptanalysis ?* **GTU : Summer-19, Marks 3**

2.7 AES with Structure

GTU : Summer-17,19, Winter-17,18,19

- Advanced Encryption Standard (AES) is a block cipher with a block length of 128 bits. AES allows for three different key lengths: 128, 192, or 256 bits.
- AES is a non-Feistel cipher that encrypts and decrypts a data block of 128-bits.

- The data to be sent is encrypted using a substitution permutation network, which means the data is first broken into blocks, in 4×4 rows, with each byte being substituted for a new one in line with the encryption key.
- The key features of AES :
  1. Symmetric key symmetric block cipher
  2. Data of 128 bits
  3. Compared to triple-DES it tends to be faster and stronger
  4. Design details and specifications are complete
  5. Resistance against all known attacks.
  6. Speed and code compactness on a wide range of platforms.
  7. Design simplicity.
- Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are identical.
- Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption.
- Fig. 2.7.1 shows AES encryption and decryption process.
  (See Fig. 2.7.1 on next page)
- To appreciate the processing steps used in a single round, it is best to think of a 128-bit block as consisting of a 4 × 4 array of bytes, arranged as follows:

$$\begin{bmatrix} \text{byte}_0 & \text{byte}_4 & \text{byte}_8 & \text{byte}_{12} \\ \text{byte}_1 & \text{byte}_5 & \text{byte}_9 & \text{byte}_{13} \\ \text{byte}_2 & \text{byte}_6 & \text{byte}_{10} & \text{byte}_{14} \\ \text{byte}_3 & \text{byte}_7 & \text{byte}_{11} & \text{byte}_{15} \end{bmatrix}$$

- Notice that the first four bytes of a 128-bit input block occupy the first column in the 4 × 4 array of bytes. The next four bytes occupy the second column, and so on. The 4 × 4 array of bytes shown above is referred to as the state array in AES.
- In Advanced Encryption Standard, the process goes through several rounds:
  1. **Key Adding :** The encryption key is added to the data, fusing them together.
  2. **Substitution :** Each byte of the cipher block is substituted for a new one, according to the cipher schedule.
  3. **Row Shifting :** The rows of bytes shift around to different positions.
  4. **Column Mixing :** The columns of bytes are further complicated through mathematical equations.

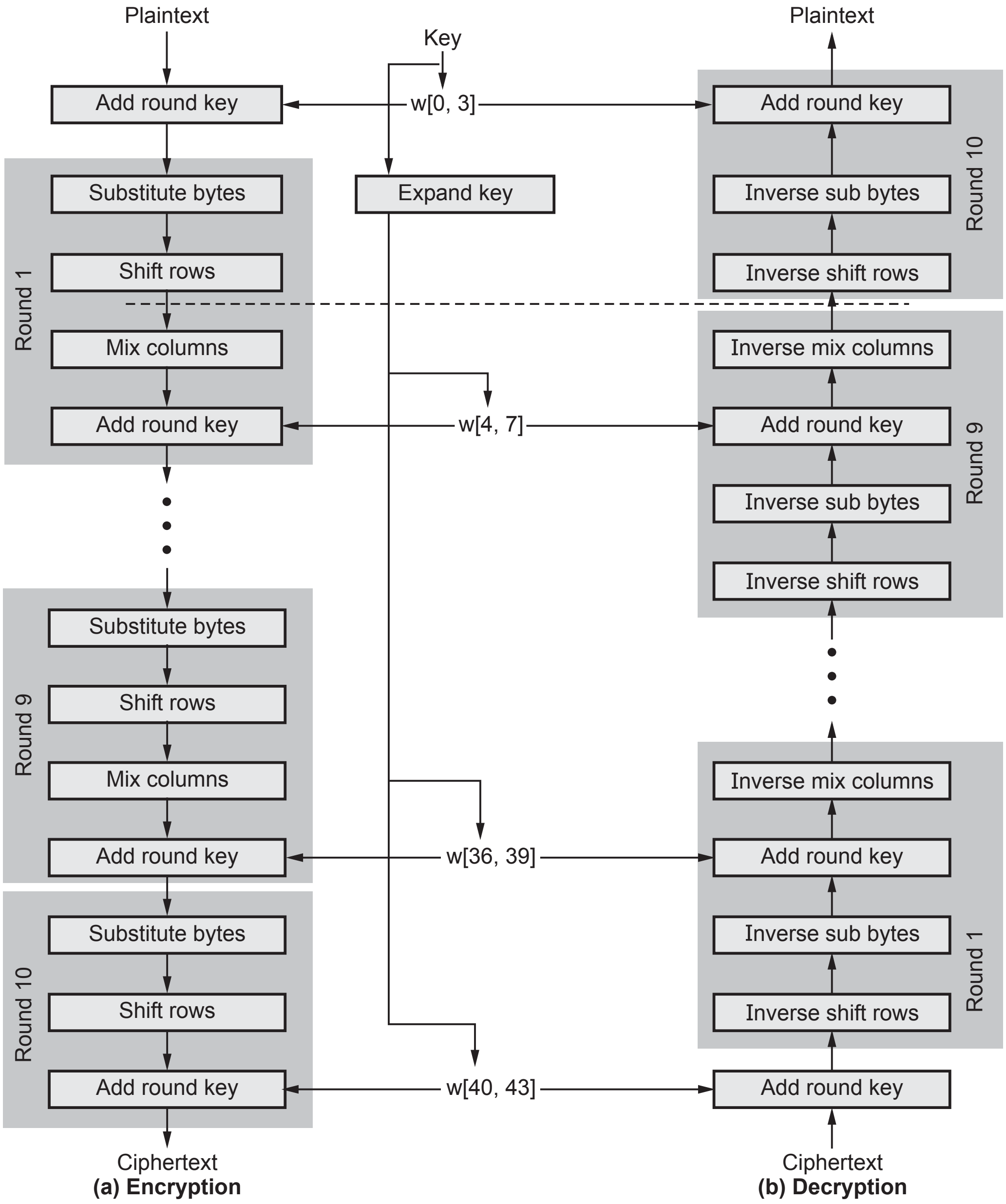


Fig. 2.7.1 AES encryption and decryption

1. **STEP 1 :** It is called subbytes for byte-by-byte substitution during the forward process. The corresponding substitution step used during decryption is called InvSubBytes. This step consists of using a 16 × 16 lookup table to find a replacement byte for a given byte in the input state array. The entries in the lookup table are created by using the notions of multiplicative inverses in GF(28) and bit scrambling to destroy the bit-level correlations inside each byte.

2. **STEP 2 :** It is called ShiftRows for shifting the rows of the state array during the forward process. The corresponding transformation during decryption is denoted InvShiftRows for Inverse Shift-Row Transformation. Fig. 2.7.2 shows one round of encryption and one round of decryption process.

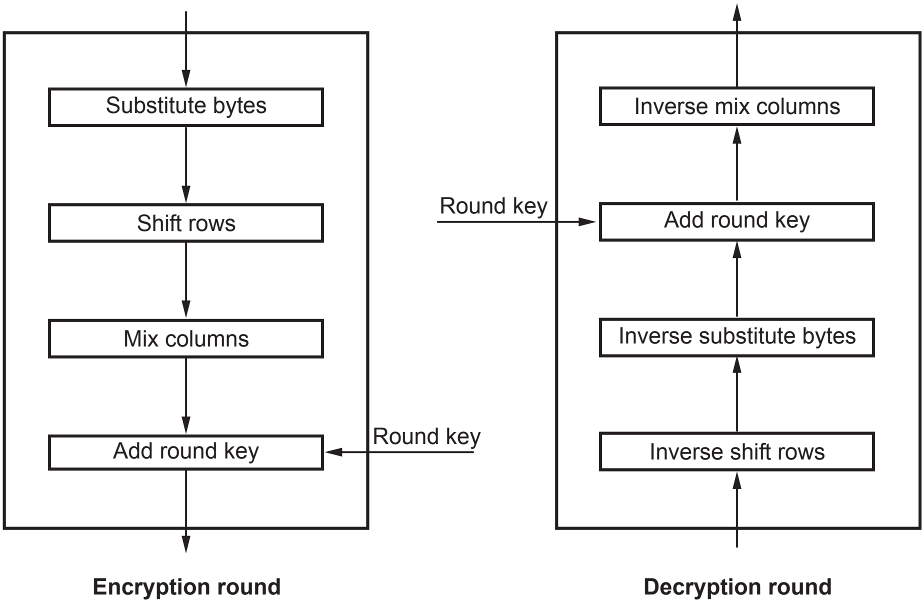


Fig. 2.7.2 One round of encryption and one round of decryption process

3. **STEP 3 :** it is called MixColumns for mixing up of the bytes in each column separately during the forward process. The corresponding transformation during decryption is denoted InvMixColumns and stands for inverse mix column transformation. The goal is here is to further scramble up the 128-bit input block.

- The shift-rows step along with the mix-column step causes each bit of the ciphertext to depend on every bit of the plaintext after 10 rounds of processing.
- **STEP 4 :** this round is called AddRoundKey for adding the round key to the output of the previous step during the forward process. The corresponding step during decryption is denoted InvAddRoundKey for inverse add round key transformation
- **Decryption Process :** The decryption process is very similar to the encryption process but this works in the reverse of the same process.
- Hence each round consists of the four processes carried out in the reverse order : **Add round key**, **Mix columns**, **Shift rows** and **Byte substitution**.
- For Advanced Encryption Standard (AES) cipher the encryption and the decryption have to be separately applied and implemented.

- The last round for encryption does not involve the "Mix columns" step. The last round for decryption does not involve the "Inverse mix columns" step.

Comments about the AES structure :

1. AES structure is not a Feistel structure.
2. The key that is provided as input is expanded into an array of forty-four 32-bit words, w(i).
3. Four different stages are used, one of permutation and three of substitution.
4. 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.
5. Only the AddRoundkey stage make use of the key.
6. The AddRoundkey stage is, in effect, a form of Vernam Cipher and by itself would not be formidable.
7. Each stage is easily reversible.
8. The decryption algorithm makes use of the expanded key in reverse order.
9. Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext.
10. The final round of both encryption and decryption consists of only three stages.

2.7.1 Advantages of AES

1. Implies to be a very robust protocol since this can be applied to both hardware and software.
2. It is also very robust for hackers because of its large key sizes. The key sizes used here are very higher as like 128, 192 and 256 bits for encryption.
3. A large set of applications such as e-business, data storage in an encrypted format and wireless communication make use of these AES protocols in a large extent.
4. Commercially this cipher protocol is among the most widely used ones all around the world.

2.7.2 Evaluation Criteria for AES

- NIST evaluation criteria for AES are
  1. Security
  2. Cost
  3. Algorithm and implementation characteristics.

1. Security

- This refers to the effort required to cryptanalyse an algorithm. Following parameters are also consider for evaluation.
  a. **Actual security** compared to other submitted algorithms.
  b. **Randomness :** The extent to which the algorithm output is indistinguishable from a random permutation on the input block.
  c. **Soundness** of the mathematical basis for the algorithm's security.
  d. Other security factors raised by the public during the evaluation process.

2. Cost

a. Licensing requirements : When the AES is issued, the algorithm specified in the AES shall be available on a worldwide, non-exclusive, royalty free basis.

b. Computational efficiency : The evaluation of computational efficiency will be applicable to both hardware and software implementations.

c. Memory requirements : The memory requirement for implementing the algorithm in hardware and software will be considered.

3. Algorithm and Implementation Characteristics

This category includes a variety of considerations, including flexibility, suitability for a variety of hardware and software implementations; and simplicity, which will make an analysis of security more straight forward.

The following criteria were used in the final evaluation :

1. **General security :** NIST relied on the public security analysis conducted by the cryptographic community.
2. **Software implementations :** It includes execution speed, performs across a variety of platforms and variation of speed with key size.
3. Restricted space environments.
4. Hardware implementations.
5. Attacks on implementations.
6. Encryption versus decryptions.
7. Key agility.
8. Other versatility and flexibility.
9. Potential for instruction level parallelism.

2.7.3 Comparison between AES and DES

Sr. No.	Parameters	AES	DES
1	Block size	128-bits	64-bits
2	Key length	128, 192, 256-bits	56-bits (effective length)
3	Encryption primitives	Substitution, shift, bit mixing	Substitution, Permutation
4	Cryptographic primitives	Confusion, Diffusion	Confusion, Diffusion
5	Design rationale	Closed	Open

University Questions

1. *Elaborate AES encryption with neat sketches.* **GTU : Summer-17, Marks 7**
2. *Explain avalanache effect in DES and discuss strength of DES in brief.* **GTU : Summer-17, Marks 4**
3. *Explain AES encryption in detail.* **GTU : Winter-17, Marks 7**
4. *Describe various steps of AES.* **GTU : Summer-18, Marks 7**
5. *Discuss in detail encryption and decryption process of AES.* **GTU : Winter-18, Marks 7**
6. *Explain four different stages of AES (advance encryption standard) structure.* **GTU : Summer-19, Marks 7**
7. *Briefly describe mix columns and add round key in AES algorithm.* **GTU : Winter-19, Marks 7**

2.8 Short Questions and Answers

Q.1 Explain the avalanche effect.

Ans. : 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 ciphertext. In particular, a change I one of the plaintext or one bit of the key should produce a change in many bits of the ciphertext.

Q.2 What is a brute force attack ?

Ans. : A brute force attack consists of trying every possible code, combination or password until you find the right one.

Q.3 What is DES ?

Ans. : DES is a symmetric cipher defined in Federal Information Processing (FIPS) Standard Number 46 in 1977 as the federal government approved encryption algorithm for sensitive but non-classified information. DES utilizes a 56-bit key. This key size is vulnerable to a brute force attack using current technology.

Q.4 Define : Diffusion.

Ans. : Diffusion is the statistical structure of the plain text is dissipated into long-range statistics of the cipher text. This is possible by using permutation.

Q.5 List out the parameters of AES.

Ans. : Parameters of AES are security, cost and algorithm and implementation characteristics.

Q.6 What is AES cipher ?

Ans. : Advanced Encryption Standard (AES) is a symmetric key block cipher. AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. The key size can be 128,192 or 256 bits. It depends on number of rounds. The number of rounds:10 rounds for 128 bits,12 rounds for 192 bits, and 14 rounds for 256 bits.

2.9 Multiple Choice Questions

Q.1 Substitution box provides ________.

a Confusion b diffusion

c both confusion and diffusion d None of these

Q.2 DES encrypts data in block size of ______ bits each.

a 32 b 56 c 64 d 128

Q.3 DES consists of _______ rounds to perform the substitution and transposition techniques.

a 16 b 18 c 21 d 25

Q.4 DES uses a key generator to generate sixteen _______ round keys.

a 32-bit b 42-bit c 48-bit d 56-bit

Q.5 ________ is the first step in DES.

a Key transformation b Expansion permutation

c S-box substitution d P-box substitution.

Q.6 DES has ______ weak keys.

a 2 b 4 c 6 d 8

Q.7 The input block length in AES is ______.

a 56 bits b 64 bits c 112 bits d 128 bit

Q.8 Differential attack is a ________.

a Chosen text b chosen-plaintext attack

c Cipher text only
d known plaintext

Q.9 DES Encryption standard (DES) is a symmetric key block cipher published by the NIST .

a asymmetric key block
b symmetric key stream
c asymmetric key stream
d Symmentric key block

Q.10 AES is a ________ cipher with a block length of 128 bits.

a stream
b block
c hybrid
d None

Q.11 Differential attack is a ________ attack.

a ciphertext-only
b known-plaintext
c chosen-plaintext
d adaptive chosen plaintext

Q.12 Block ciphers in counter mode use ________ numbers as the input to the algorithm.

a random
b binary
c sequence
d all of these

Q.13 AES stands for ________ .

a Advanced Encryption Substitution
b Active Encryption Standard
c Advanced Email Standard
d Advanced Encryption Standard

Q.14 ________ is a bit-oriented cipher.

a AES
b DES
c AES and DES
d None

Q.15 AES is a ________ oriented cipher.

a byte
b bit
c number
d all of these

Q.16 The 4×4 byte matrices in the AES algorithm are called ________ .

a permutations
b words
c transitions
d states

Q.17 Advanced encryption standard uses the algorithm ________ .

a Twofish algorithm
b Blowfish

c Rijndael algorithm d Kryptotal algorithm

Q.18 AES has ________ different configuration.

a 2 b 3

c 4 d 5

Answer Keys for Multiple Choice Questions

Q.1	a	**Q.2**	c	**Q.3**	a	**Q.4**	c
Q.5	a	**Q.6**	b	**Q.7**	d	**Q.8**	b
Q.9	d	**Q.10**	b	**Q.11**	c	**Q.12**	c
Q.13	d	**Q.14**	b	**Q.15**	a	**Q.16**	d
Q.17	c	**Q.18**	b				

❑❑❑

Notes

3 Multiple Encryption and Triple DES

Syllabus

Multiple encryption and triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback mode, Output Feedback mode, Counter mode.

Contents

3.1 Double DES

GTU : Winter-17, 19, Summer-19

- Double DES has a 112-bit key and enciphers blocks of 64 bits.
- Double DES uses two keys to say K_1 and K_2 in this algorithm. It first performs DES on the original plain text using K_1 to get the encrypted text in cryptography.
- Here, it again performs DES on the encrypted text but this time with the other key K_2 in this algorithm.
- Firstly, the final output is the encryption of encrypted text with the original plain text encrypted twice with two different keys shown in the structure as given below :

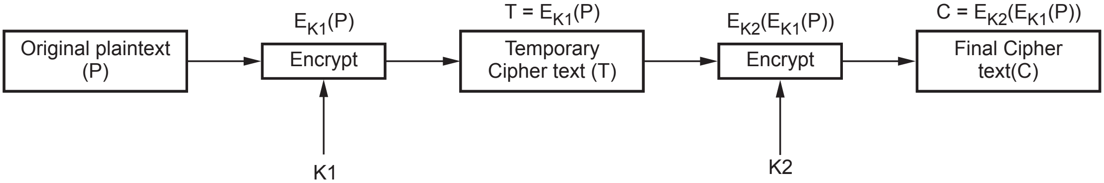


Fig. 3.1.1

- Using two encryption stages and two keys.
  A) The plain text to ciphertext is as follows,

  $$C = E_{K_2}(E_{K_1}(P)) \text{ where } K_1 \text{ and } K_2 \text{ are the key.}$$

  B) Ciphertext to plain text is as follows,

  $$P = D_{K_1}(D_{K_2}(C))$$

- Meet-in-the-middle attack is the drawback of double DES in this. Mainly, this attack involves encryption from one end, decryption from the other and matching the results in the middle hence the name in the message.
- Meet-in-the-middle attack was first introduced by Diffie and Hellman in for cryptanalysis of DES and it is a generic method to analyze high-level structures of cryptographic algorithms.
- Its fundamental idea is that if the target algorithm can be decomposed into two smaller parts and the computation of each part only involves portions of master keys, then we can investigate the security level of each part separately and finally combine the results from both sides.
- This attack requires knowing some plaintext/ciphertext pairs. Let's assume that we have a plaintext/ciphertext pair; i.e., we know the plaintext p and the corresponding ciphertext C.

- Attacks on DES have typically been brute force attacks. Here is the double encryption :

$$p \rightarrow E(K_1, p) \rightarrow E(K_2, E(K_1, p)) = C$$

- Encrypt p using all 2^{56} possible keys and store the results. The stored results will include all possible encryptions $p \rightarrow E(K_1, p)$.
- Then decrypt C using all possible keys.

$$D(K_2, C) = D(K_2, E(K_2, E(K_1, p))) \rightarrow E(K_1, p)$$

- After decrypting with each key, check for a match with the stored outputs of the 2^{56} possible encryptions. When we have a match, we have located a possibly correct pair of keys. Now, perhaps more than one pair of keys will result in a match, but the number of pairs of keys that return matches should be small.

University Questions

1. *What is meant by meet - in - the middle attack in double DES ? Explain the same in brief.* **GTU : Winter-17, Marks 4**
2. *How meet in the middle attack is performed on double DES ?* **GTU : Summer-19, Marks 4**
3. *What is a meet-in-the-middle attack in double DES ?* **GTU : Winter-19, Marks 4**

3.2 Triple DES

GTU : Summer-17, 18

- Triple DES is simply another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits.
- The procedure for encryption is exactly the same as regular DES, but it is repeated three times. Hence the name triple DES.
- Triple DES uses 2 or 3 keys.
- The data is encrypted with the first key (K_1), decrypted with the second key (K_2), and finally encrypted again with the third key (K_3).
- Triple DES with three keys is used quite extensively in many products including PGP and S/MIME.
- Brute force search impossible on Triple DES.
- Meet-in-middle attacks need 256 Plaintext-Ciphertext pairs per key.
- Cipher text is produced as $C = E_{K_3}[D_{K_2}[E_{K_1}[P]]]$.
- Fig. 3.2.1 shows the 3DES method with three key.
- Triple DES runs three times slower than standard DES, but is much more secure if used properly.

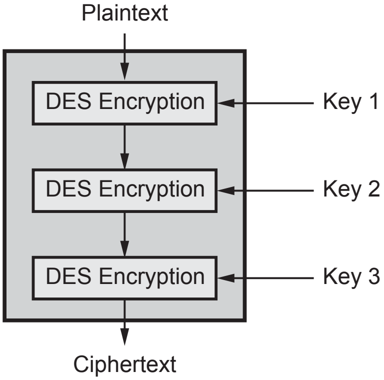


Fig. 3.2.1 3DES with three key method

- The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse.
- Like DES, data is encrypted and decrypted in 64-bit chunks.
- There are some weak keys that one should be aware of : If all three keys, the first and second keys, or the second and third keys are the same, then the encryption procedure is essentially the same as standard DES. This situation is to be avoided because it is the same as using a really slow version of regular DES.
- The input key for DES is 64-bits long; the actual key used by DES is only 56-bits in length. The least significant (right-most) bit in each byte is a parity bit, and should be set so that there are always an odd number of 1s in every byte. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56-bits. This means that the effective key strength for Triple DES is actually 168-bits because each of the three keys contains 8 parity bits that are not used during the encryption process.

University Questions

1. *Explain triple DES with two keys.* **GTU : Summer-17, Marks 4**
2. *Explain double and triple DES.* **GTU : Summer-18, Marks 4**

3.3 Block Cipher Mode Operation **GTU : Summer-17, 18, 19, Winter-17, 19**

- The modes of operation of block ciphers are configuration methods that allow those ciphers to work with large data streams, without the risk of compromising the provided security.
- There are five types of operations in block cipher modes, ECB (Electronic Code Block) mode, CBC (Cipher Block Chaining) mode, CFB (Cipher Feedback) mode, OFB (Output Feedback) mode and CTR (Counter) mode.

- Where ECB and CBC mode works on block ciphers, and CFB and OFB mode works on block ciphers acting as stream ciphers.
- ECB is used for transmitting a single value in secure manner, CBC is used for encrypting blocks of text authentication, CFB is used for transmitting encrypted stream of data authentication, OFB is used for transmitting encrypted stream of data, CTR is used for transmitting block-oriented applications.
- Modes of operation enable the repeated and secure use of a block cipher under a single key. A block cipher by itself allows encryption only of a single data block of the cipher's block length.
- When targeting a variable-length message, the data must first be partitioned into separate cipher blocks. Typically, the last block must also be extended to match the cipher's block length using a suitable padding scheme.
- Modes of operation have primarily been defined for encryption and authentication. While modes of operation are commonly associated with symmetric encryption, they may also be applied to public-key encryption primitives such as RSA in principle.

3.3.1 Electronic Code Book (ECB)

- A block of plaintext encrypts into a block of Ciphertext. Block size is 64-bits. Each block is encrypted independently.
- Plaintext patterns are not concealed since identical blocks of plaintext give identical blocks of ciphertext. It is not necessary to encrypt the file linearly.
- User can encrypt the 10 blocks in the middle first, then the blocks at the end, and finally the blocks in the beginning. Because of this, encrypted files are accessed randomly like a data base.
- It is very easy to parallelize the process. Pad the last block with some regular pattern i.e. zeros, ones to make it a complete block.
- End of file character is used to denote the final plaintext byte before padding.
- ECB method is ideal for a short amount of data, such as an encryption key.
- Fig. 3.3.1 shows ECB mode.
- In this mode, the plain text is divided into a block where each block is of 64 bits. Then each block is encrypted separately. The same key is used for the encryption of all blocks. Each block is encrypted using the key and makes the block of ciphertext.

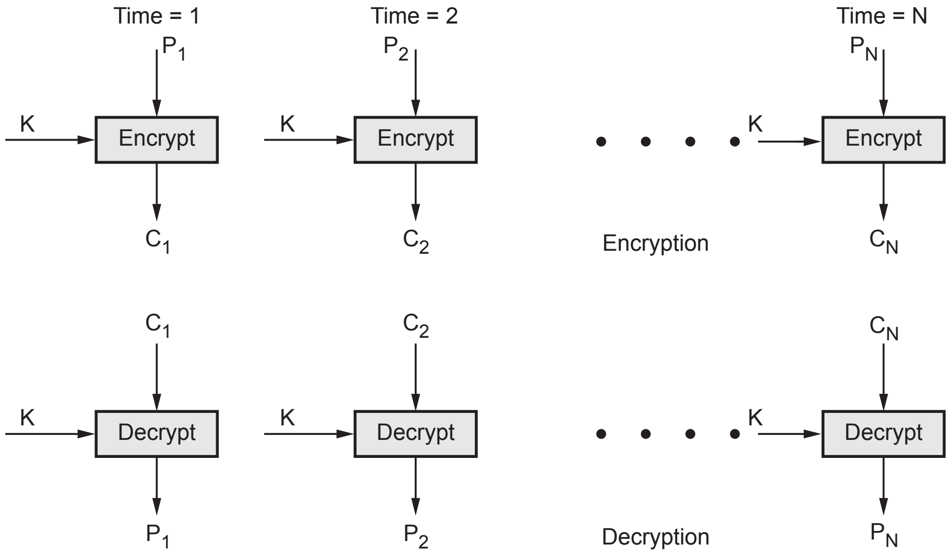


Fig. 3.3.1 ECB mode

- At the receiver side, the data is divided into a block, each of 64 bits. The same key which is used for encryption is used for decryption. It takes the 64-bit ciphertext and by using the key convert the ciphertext into the plain text.
- For lengthy messages, the ECB mode may not be secure.
- Used in secure transmission of single values i.e. an encryption key.
- ECB has security problems that limit its usability.
- Patterns in the plaintext can yield patterns in the ciphertext.
- It is also easy to modify a ciphertext message by adding, removing or switching encrypted blocks.
- Synchronization error is unrecoverable.

3.3.2 Cipher Block Chaining Mode (CBC)

- Cipher block Mode at the sender side, the plain text is divided into blocks. In this mode IV(Initialization Vector) is used which can be a random block of text. IV is used to make the ciphertext of each block unique.
- The first block of plain text and IV is combined using the XOR operation and then encrypted the resultant message using the key and form the first block of ciphertext. the first block of ciphertext is used as IV for the second block of plain text. the same procedure will be followed for all blocks of plain text.
- At the receiver side, the ciphertext is divided into blocks. The first block ciphertext is decrypted using the same key which is used for encryption. The decrypted result will be XOR with the IV and form the first block of plain text. The second

block of ciphertext is also decrypted using the same key and the result of the decryption will be XOR with the first block of ciphertext and form the second block of plain text. The same procedure is used for all the blocks.

- The plaintext is XORed with the previous ciphertext block before it is encrypted.
- The CBC mode is iterative mode.
- After a plaintext block is encrypted, the resulting ciphertext is also stored in a feedback register.
- Before the next plaintext block is encrypted, it is XORed with feedback register to become the next input to the encrypting routine.
- The encryption of each block depends on all the previous blocks.
- A ciphertext block is decrypted normally and also saved in a feedback register.
- After the next block is decrypted, it is XORed with the results of the feedback register.
- Mathematically it is

$$C_i = E_k(P_i \oplus C_{i-1})$$

$$P_i = C_{i-1} \oplus D_k(C_i)$$

- It hides patterns in the plaintext.
- In order to guarantee that there is always some random looking ciphertext to apply to the actual plaintext, the process is started with a block of random bits called the Initialization Vector (IV).
- Fig. 3.3.2 shows cipher block chaining mode.

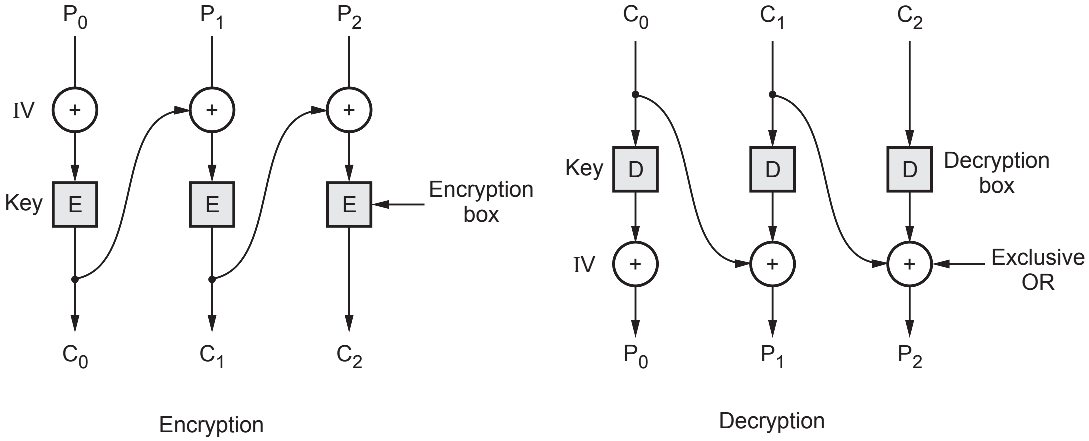


Fig. 3.3.2 CBC

- When used in networking messages, most CBC implementations add the IV to the beginning of the message in plaintext.

- A single bit error in a plaintext block will affect that ciphertext block and all subsequent ciphertext blocks.
- CBC mode is self recovering.
- Two blocks are affected by an error, but the system recovers and continues to work correctly for all subsequent blocks. Synchronization error is unrecoverable.
- Encryption is not parallelizable.
- Decryption is parallelizable and has a random access property.

3.3.3 Cipher Feedback Mode (CFB)

- Data is encrypted in units that are smaller than a defined block size.
- It is possible to convert the DES into stream cipher using cipher feedback mode.
- In this mode, the data is encrypted in the form of units where each unit is of 8 bits.
- Like cipher block chaining mode, IV is initialized. the IV is kept in the shift register. It is encrypted using the key and form the ciphertext.
- Fig. 3.3.3 shows CFB encryption and decryption process.

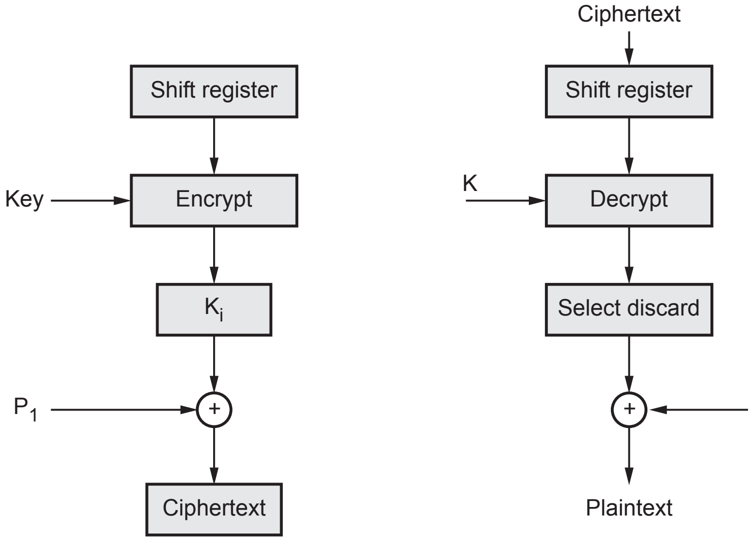


Fig. 3.3.3 CFB Modes

- More than one message can be encrypted with the same key, provided that a different initialization vector is used.
- CFB speed is the same as the block cipher.
- Encryption is not parallelizable, decryption is parallelizable and has a random access property.

- CFB is self recovering with respect to synchronization errors as well.

Advantages :

1. Simplicity
2. Need not be used on a byte boundary.
3. Input to the block cipher is randomized.
4. Ciphertext size is the same size as the plaintext size.

Disadvantages :

1. Encryption is not parallelizable.
2. Plaintext is somewhat difficult to manipulate.

3.3.4 Output Feedback Mode

- The output feedback (OFB) mode is similar in structure to that of CFB. Fig. 3.3.4 shows output feedback mode.

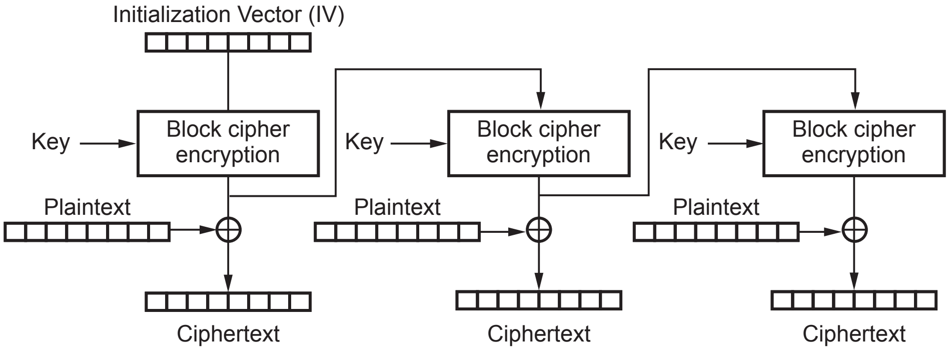
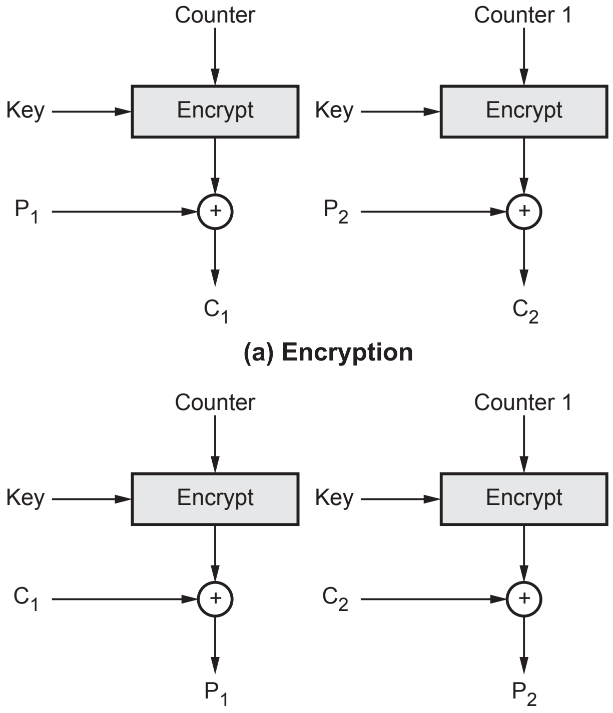


Fig. 3.3.4 Output feedback (OFB) mode encryption

- It is the output of the encryption function that is fed back to the shift register in OFB, whereas in CFB, the ciphertext unit is fed back to the shift register.
- The other difference is that the OFB mode operates on full blocks of plaintext and ciphertext, not on an s-bit subset.
- **Advantages and Limitations of OFB**
  1. Needs an Initialization vector which is unique for each use
  2. Bit errors do not propagate
  3. More vulnerable to message stream modification
  4. Sender & receiver must remain in sync
  5. Only use with full block feedback

3.3.5 Counter Mode

- Block ciphers in counter mode use sequence numbers as the input to the algorithm.
- More than one message can be encrypted with the same key, provided that a different initialise vector is used.
- Plaintext is very easy to manipulate, any change in ciphertext directly affects the plaintext. Fig. 3.3.5 shows counter mode.



(a) Encryption

(b) Decryption

Fig. 3.3.5 Counter mode

- Synchronization error is unrecoverable.
- A ciphertext error affects only the corresponding bit of plaintext.
- Encryption : The counter is encrypted and then XORed with the plaintext block to produce the ciphertext block.

Advantages

1. Simple to implement.
2. It provides confidentiality.
3. Random access of block is possible.
4. Efficiency is same as block cipher.

University Questions

1. *Discuss electronic code book and cipher feedback mode with neat diagrams.* **GTU : Summer-17, Marks 7**
2. *Discuss the following block cipher modes of operation in detail with neat sketches :*
   - Cipher block chaining mode
   - Counter mode **GTU : Winter-17, Marks 7**
3. *Explain cipher feedback mode of DES operation.* **GTU : Summer-18, Marks 4**
4. *Explain counter mode of DES operation.* **GTU : Summer-18, Marks 4**
5. *Explain working of ECB. Why ECB (electronic code book) is rarely used to encrypt message ?* **GTU : Summer-19, Marks 4**
6. *Why CFB (cipher feedback mode) encrypted messages are less subject to tampering than OFB (output feedback mode) ?* **GTU : Summer-19, Marks 3**
7. *Explain CFB algorithm mode with diagram.* **GTU : Winter-19, Marks 3**
8. *Explain Counter (CTR) algorithm mode with diagram.* **GTU : Winter-19, Marks 3**

3.4 Short Questions and Answers

Q.1 What is triple encryption ?

Ans. : The function follows an encrypt - decrypt - encrypt (EDE) sequence. There is no cryptographic significance to the use of decryption for the second stage.

Q.2 How many keys are used in triple encryption ?

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

Q.3 Why is the middle portion of 3DES a decryption rather than an encryption ?

Ans. : Decryption requires that the keys be applied in reverse order: P=Dk1[Ek1[P]]. This results in a dramatic increase in cryptographic strength.

Q.4 Why ECB mode is not secure for lengthy message ?

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

3.5 Multiple Choice Questions

Q.1 Which is the largest disadvantage of the symmetric encryption ?

a More complex and therefore more time-consuming calculations.

b Problem of the secure transmission of the Secret Key.

c Less secure encryption function.

d Isn't used any more.

Q.2 ________ DES was designed to increase the size of the DES key.

a Double b Triple c Quadruple d None of the above

Q.3 ECB and CBC are ________ ciphers.

a block b stream c field d product

Q.4 The AES key expansion algorithm takes as input a ________ key and produces a linear array of 156 bytes.

a 8-byte b 12-byte c 16-byte d 24-byte

Q.5 ECB mode stands for ------------------- Mode

a Electronic code block b Electronic cyber block

c Encryption code block d Electronic counter block

Q.6 How many keys does the Triple DES algorithm use ?

a 2 b 3

c 2 or 3 d 3 or 4

Q.7 Which one of the following DES operating modes can be used for large messages with the assurance that an error early in the encryption/decryption process won't spoil results throughout the communication ?

a Cipher Block Chaining (CBC) b Electronic Codebook (ECB)

c Cipher Feedback (CFB) d Output Feedback (OFB)

Q.8 What is the minimum number of cryptographic keys required to achieve a higher level of security than DES with the Triple DES algorithm ?

a 1 b 2 c 3 d 4

Q.9 Double DES has a --------- key and enciphers blocks of 64 bits.

a 32-bit b 56-bit c 112-bit d 128-bit

Answer Keys for Multiple Choice Quesions

Q.1	b	**Q.2**	b	**Q.3**	a
Q.4	c	**Q.5**	a	**Q.6**	c
Q.7	d	**Q.8**	b	**Q.9**	c

4 Public Key Cryptosystems with Appliations

Syllabus

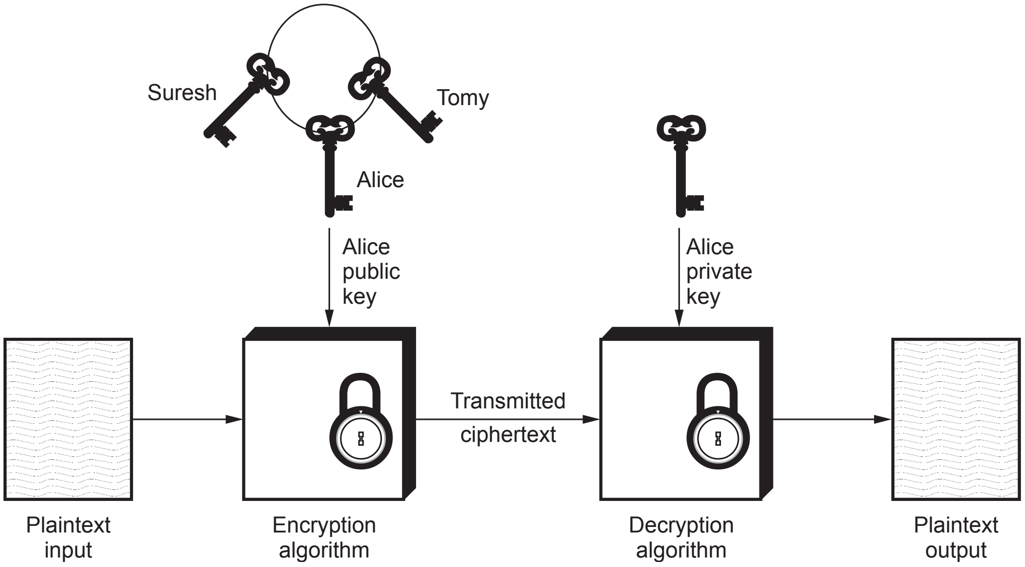
Public Key Cryptosystems with Applications, Requirements and Cryptanalysis, RSA algorithm, its computational aspects and security, Diffie-Hillman Key Exchange algorithm, Man-in-Middle attack

Contents

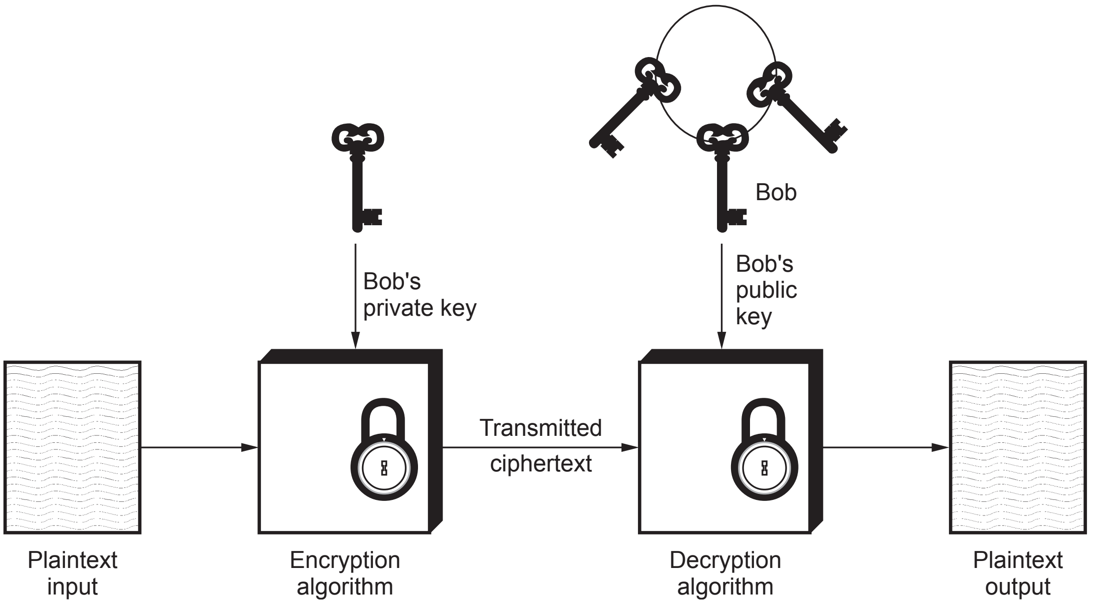
4.1 Public Key Cryptosystem

GTU : Summer-17, Winter-17, 19

- Public key cryptography, is a method of encrypting data with two different keys and making one of the keys, the public key, available for anyone to use. The other key is known as the private key.
- Data encrypted with the public key can only be decrypted with the private key, and data encrypted with the private key can only be decrypted with the public key. Public key encryption is also known as asymmetric encryption.
- In public key cryptography, there are two keys. Suppose Alice wishes to receive encrypted messages; she publishes one of the keys, the public key, and anyone, say Bob, can use it to encrypt a message and send it to her.
- When Alice gets the encrypted message, she uses the private key to decrypt it and read the original message. If Alice needs to reply to Bob, Bob will publish his own public key, and Alice can use it to encrypt her reply.
- These algorithms have the following important characteristic.
  1. It must be computationally easy to encipher or decipher a message given the appropriate key.
  2. It must be computationally infeasible to derive the private key from the public key.
  3. It must be computationally infeasible to determine the private key from a chosen plaintext attack.
- Fig. 4.1.1 shows public key cryptosystem.



(a) Encryption



(b) Authentication

Fig. 4.1.1 Public key cryptography

- Public key cryptographic algorithm has six elements as follow :
  1. **Plain Text :** This is a readable message which is given as input to the algorithm. In a public key algorithm, the plain text is encrypted in blocks.
  2. **Encryption Algorithm :** The encryption algorithm is implemented on the plain text which performs several transformations on plain text.
  3. **Public and Private keys :** These are the set of keys among which if one is used for encryption the other would be used for decryption. The transformation of plain text by encryption algorithm depends on the key chosen from the set to encrypt the plain text.
  4. **Cipher Text :** This is the output of encryption algorithm. The generated cipher text totally depends on the key selected from the set of the public and private key. Both of these keys, one at a time with plain text would produce different cipher texts.
  5. **Decryption Algorithm :** This would accept the output of the encryption algorithm i.e. the cipher text and will apply the related key to produce the original plain text.
- The essential steps are the following :
  1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
  2. Each user places one of the two keys in a public register. This is the public key. The companion key is kept private.

3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.
4. Alice decrypts the message using her private key.

- The public key is accessed to all participants and private key is generated locally by each participant.

4.1.1 Requirement of Public Key Cryptography

1. It is computationally easy for a party B to generate a pair.
2. It is computationally easy for a sender A, to generate the corresponding ciphertext :

$$C = E(PU_b, M)$$

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

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

4. It is computationally infeasible for an adversary, knowing the public key (PU_b) to determine the private key PR_b.
5. It is computationally infeasible for an adversary, knowing the public key (PU_b) and a ciphertext (C) to recover the original message (M).

4.1.2 Advantages and Disadvantages

- **Advantages of public key algorithm**
  1. Only the private key must be kept secret.
  2. The administration of keys on a network requires the presence of only a functional trusted TTP as opposed to an unconditionally trusted TTP.
  3. A private/public key pair remains unchanged for considerable long periods of time.
  4. There are many relatively efficient digital signature mechanisms as a result of asymmetric-key schemes.
  5. In a large network the number of keys necessary may be considerably smaller than in the symmetric-key scenario.
- **Disadvantages of public key algorithm**
  1. Slower throughput rates than the best known symmetric-key schemes.
  2. Large key size.
  3. No asymmetric-key scheme has been proven to be secure.
  4. Lack of extensive history.

4.1.3 Comparison between Public Key and Private Key

Public Key	Private Key
Public key encryption is also known as asymmetric key encryption.	Private key encryption is also known as symmetric key encryption.
One key for encryption and other key for decryption.	Same key is used for encryption and decryption.
Slower.	Very fast.
Key exchange is not a problem.	Key exchange is big problem.
Also called public key encryption.	Also called secret key encryption.
One of the two keys must be kept secret.	The key must be kept secret.
Public keys enable users to encrypt a message to other individuals on the system.	Private keys enable, user can decrypt a message secured by your public key.

University Questions

1. *What are the principal elements of public - key cryptosystem ? Explain in brief.* **GTU : Summer-17, Marks 3**
2. *List the requirements of Public Key Cryptography.* **GTU : Winter-17, Marks 3**
3. *What are the principal elements of a public-key cryptosystem ?* **GTU : Winter-19, Marks 3**

4.2 RSA Algorithm

GTU : Summer-11, 17, 18, 19, Winter-17, 18, 19

- RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n – 1 for some n.
- A typical size for n is 1024 bits.
- The RSA algorithm developed in 1977 by Rivest, Shamir, Adleman (RSA) at MIT. RSA algorithm is public key encryption type algorithm. In this algorithm, one user uses a public key and other user uses a secret (private key) key.
- In the RSA algorithm each station independently and randomly chooses two large primes p and q number, and multiplies them to produce $n = pq$ which is the modulus used in the arithmetic calculations of the algorithm.
- The details of the RSA algorithm are described as follows :
- **Key generation :**
  1) Pick two large prime numbers p and q, $p \neq q$;
  2) Calculate $n = p \times q$;

3) Calculate $\phi(n) = (p-1)(q-1)$;

4) Pick e, so that gcd $(e, \phi(n)) = 1$, $1 < e < \phi(n)$;

5) Calculate d, so that $d \cdot e \bmod \phi(n) = 1$, i.e. d is the multiplicative inverse of e in mod $\phi(n)$;

6) Get public key as $K_U = \{e, n\}$;

7) Get private key as $K_R = \{d, n\}$.

- **Encryption :**

  For plaintext block $P < n$, its ciphertext $C = P^e \bmod n$.

- **Decryption :**

  For ciphertext block C, its plaintext is $P = C^d \bmod n$.

Why RSA works :

- As we have seen from the RSA design, RSA algorithm uses modular exponentiation operation. For $n = p \cdot q$, e which is relatively prime to $\phi(n)$, has exponential inverse in mod n.
- Its exponential inverse d can be calculated as the multiplicative inverse of e in mod $\phi(n)$. The reason is illustrated as follows :

  Based on Euler's theorem, for y which satisfies $y \bmod \phi(n) = 1$, the following equation holds :

  $$x^y \bmod n = x \bmod n$$

  AS $d \cdot e \bmod \phi(n) = 1$, we have that $p^{ed} \equiv P \bmod n$. So the correctness of RSA cryptosystem is shown as follows :

- **Encryption :** $C = P^e \bmod n$;
- **Decryption :** $P = C^d \bmod n = (P^e)^d \bmod n = P^{ed} \bmod n = P \bmod n = P$.

Why RSA is secure :

- The premise behind RSA's security is the assumption that factoring a big number (n into p and q) is hard. And thus it is difficult to determine $\phi(n)$. Without the knowledge of $\phi(n)$, it would be hard to derive d based on the knowledge of e.

Advantages

1. RSA can be used both for encryption as well as for digital signatures.
2. Trapdoor in RSA is in knowing value of n but not knowing the primes that are factors of n.

Disadvantages

1. If any one of p, q, m, d is known, then the other values can be calculated. So secrecy is important.

2. To protect the encryption, the minimum number of bits in n should be 2048.

4.2.1 Attacks on RSA

Attacks on RSA algorithm are as follows :

1. **Brute force :** This involves trying all possible private keys.
2. **Mathematical attacks :** This involves the factoring the product of two primes.
3. **Timing attacks :** These depends on the running time of the description algorithm.
4. **Chosen ciphertext attacks :** This type of attack exploits properties of the RSA algorithm.

4.2.1.1 Computing $\phi(n)$

- Computing $\phi(n)$ is no easier that factoring n. For, if n and $\phi(n)$ are known, and n is the product of two primes p, q, then n can be easily factored, by solving the two equations.

$$n = pq \qquad \text{... (4.2.1)}$$

$$\phi(n) = (p-1)(q-1) \qquad \text{... (4.2.2)}$$

for the two unknowns p and q.

- If we substitute q = n/p into the equation (4.2.2), we obtain a quadratic equation in the unknown value p :

$$p^2 - (n - \phi(n) + 1)p + n = 0 \qquad \text{... (4.2.3)}$$

- The two roots of equation (4.2.3) will be p and q, the factors of n. If a cryptanalyst can learn the value of $\phi(n)$, then he can factor 'n' and break the system.

4.2.1.2 Timing Attacks

- Kocher described a new attack on RSA in 1995.
- If the attacker Eve knows Alice's hardware in sufficient detail and is able to measure the decryption times for several known cipher-texts, she can deduce the decryption key (d) quickly. This attack can also be applied against the RSA signature scheme.
- In 2003, Boneh and Brumley demonstrated a more practical attack capable of recovering RSA factorizations over a network connection. This attack takes advantage of information leaked by the Chinese remainder theorem optimization used by many RSA implementations.

- One way to thwart these attacks is to ensure that the decryption operation takes a constant amount of time for every cipher-text. However, this approach can significantly reduce performance.
- There is simple counter-measures against timing attacks :
  1. **Constant exponentiation time :** Ensure that all exponentiations take the same time, but this will degrade performance.
  2. **Random delay :** Better performance could be achieved by adding a random delay to the exponentiation algorithm to confuse the timing attack.
  3. **Blinding :** Multiply the cipher-text by a random number before performing exponentiation. This process prevents the attacker from knowing what cipher-text bits are being processed inside the computer and therefore prevents the bit-by-bit analysis essential to the timing attack. RSA data security reports a 2 % to 10 % performance penalty for blinding.

4.2.1.3 Mathematical Attacks

- We can identify three approaches to attacking RSA mathematically :
  1. Factor n into two prime factors, this enables calculation of ϕ **(n)** = **(p – 1) (q – 1), which in turn, enables determination of d = e^{-1} mod ϕ (n).**
  2. Determine ϕ (n) directly, without first determining p and q.
  3. Determine d directly, without first determining ϕ (n).
- Most discussions of cryptanalysis of RSA have focused on the task of factoring n into its two prime numbers. Determining ϕ (n) given n is equivalent to factoring n.
- With presently known algorithms, determining d given e and n appears to at least as time consuming as the factoring problem.

4.2.1.4 Adaptive Chosen Cipher-text Attacks

- In 1998, Daniel Bleichenbacher described the first practical adaptive chosen cipher-text attack, against RSA-encrypted messages using the PKCS#1 v1 padding scheme.
- Due to flaws with the PKCS#1 scheme, Bleichenbacher was able to mount a practical attack against RSA implementations of the Secure Socket Layer protocol and to recover session keys.
- As a result of this work, cryptographers now recommend the use of provably secure padding schemes such as Optimal Asymmetric Encryption padding and RSA laboratories has released new versions of PKCS#1 that are not vulnerable to these attacks.

Example 4.2.1 *Perform encryption and decryption using RSA algorithm for $p = 17$, $q = 11$, $e = 7$ and $M = 2$.*

Solution : $P = 17 \quad q = 31$ and $e = 7$

$$n = p \times q = 17 \times 31 = 527$$

$$\phi(n) = (p-1)(q-1) = (17-1)(31-1) = 480$$

$$d = (1 + k\,\phi(n)) / e = (1 + 480k) / 7$$

$$= -959 / 7 = -137 \qquad \text{(for } k = -2)$$

$$d = -137 \pmod{480} = 343$$

Encryption $(C) = M^e \pmod n = 2^7 \pmod{527} = 128$

Decryption $M = C^d \pmod n = 128^{343} \pmod{527} = 2$

Example 4.2.2 *For the given values $p = 19$, $q = 23$ and $e = 3$ find n, $\phi(n)$ and d using RSA algorithm.*

Solution :

$$n = p * q$$

$$n = 19 \times 23$$

$$\mathbf{n = 437}$$

$$\phi(n) = (p-1) * (q-1)$$

$$\phi(n) = 18 \times 22$$

$$\phi(n) = \mathbf{396}$$

$$e.d. = 1 \bmod \phi(n)$$

$$3d = 1 \bmod 396$$

$$d = \frac{1}{3}$$

Example 4.2.3 *Using the RSA algorithm, encrypt the following :*

i) $p = 3$, $q = 11$, $e = 7$, $M = 12$

ii) $p = 7$, $q = 11$, $e = 17$, $M = 25$

iii) Find the corresponding ds for i) and ii) and decrypt the ciphertext.

Solution : i)

$$n = p * q$$

$$n = 3 * 11 = 33$$

$$\phi(n) = (p-1)(q-1)$$

$$\phi(n) = 2 * 10 = 20$$

$$e \cdot d = 1 \bmod \phi(n)$$

$$7 \cdot d = 1 \bmod 20$$

$$d = 3$$

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

$$= 12^7 \bmod 33$$

$$C = 12$$

ii) $n = p * q = 7 * 11 = 77$

$$\phi(n) = (p - 1) * (q - 1) = 6 \times 10 = 60$$

$$e \cdot d = 1 \bmod \phi(n) \Rightarrow 17\ d = 1 \bmod 77$$

$$d = 3$$

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

$$= 25^{17} \bmod 77 \Rightarrow 77 \Rightarrow c = 9$$

$$C = 12$$

iii) Decryption :

$$M = c^d \bmod n$$

In case (i) $M = 12^3 \bmod 33 = 12$

In case (ii) $M = 9^{57} \bmod 77 = 25$

Example 4.2.4 *In RSA system the public key of a given user is e = 7 and n = 187*

i) What is the private key of this user ?

ii) If the intercepted ciphertext is c = 11 and sent to a user whose public key is e = 7 and n = 187. What is the plaintext ?

iii) What are the possible approaches to defeating the RSA algorithm ?

Solution : i) $n = p * q$

$$n = 11 \times 17 \Rightarrow 187$$

$$\phi(n) = (p - 1)(q - 1)$$

$$= (17 - 1)(11 - 1) = 16 \times 10 = 160$$

$$e \cdot d = 1 \bmod \phi(n)$$

$$7\ d = 1 \bmod 160$$

$$7 \times 23 = 1 \bmod 160$$

Public key PU	(e, n) =	7, 187
Private key PR	(d, n) =	23, 187

ii) $c = 11, \quad e = 7, \quad n = 187$

Plaintext $p = c^d \bmod n$

$= 11^{23} \bmod 187$

$= 79720245 \bmod 187$

$\therefore$ Plaintext $= 88$

iii) Refer sections 2.16 and 2.15.

Example 4.2.5 *Explain about the RSA algorithm with example as : $p = 11$, $q = 5$, $e = 3$ and $PT = 9$*

Solution : $p = 11, \ q = 5$

$$n = p * q = 11 \times 5 = 55$$

$$\phi(n) = (p-1) * (q-1) = 10 * 4 = 40$$

$$e = 3 \text{ and } m = 9$$

$$\gcd(\phi(n), (e)) = \gcd(40, 3) = 1$$

$$d \equiv e^{-1} (\bmod \phi(n))$$

$$d \times e^{-1} (\bmod \phi(n)) = 1$$

$$3d \bmod 40 = 1$$

$$d = 27$$

public key $pu = \{e, n\} = \{3, 55\}$

private key $pr = \{d, n\} = \{27, 55\}$

Encryption : $C = M^e \bmod n = 9^3 \bmod 55 = 14$

decryption : $M = c^d \bmod n$

$M = 14^{27} \bmod 55 = 9$

Example 4.2.6 *In a public key system using RSA, the ciphertext intercepted is $C = 10$ which is sent to the user whose public key is $e = 5$, $n = 35$. What is the plaintext M ?*

GTU : Summer-11, Marks 4

Solution : Given data : $C = 10$, $e = 5$, $n = 35$

Find plaintext M

First calculate d : $e \cdot d = 1 \bmod \phi(n)$

$5d = 1 \bmod (24)$

$$5 \times 5 \bmod (24) = 1$$

$$d = 5$$

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

$$= 10^5 \bmod (35)$$

$$= 100000 \bmod 35$$

$$\text{Plaintext } M = 5$$

Example 4.2.7 *Perform encryption and decryption using the RSA algorithm for p = 3, q = 11, e = 7, M = 5.* **GTU : Summer-11, Marks 3**

Solution :

Given data : $p = 3, \ q = 11, \ e = 7, \ M = 5$

$$\text{Calculate } n = p \times q = 3 \times 11$$

$$n = 33$$

$$\phi(n) = (p-1)(q-1) = 2 \times 10$$

$$\phi(n) = 20$$

First calculate d : $e \cdot d = 1 \bmod \phi(n)$

$$7d = 1 \bmod 20$$

$$7 \times 3 \bmod 20 = 1$$

$$d = 3$$

To encrypt message m(<n), computers : $C = M^e \bmod n = 5^7 \bmod 33$

$$= 78125 \bmod 33 = 14$$

To decrypt received bit pattern, c, compute : $M = C^d$ mode $(n) = 14^3 \bmod(33)$

$$= 2744 \bmod 33 = 5$$

Example 4.2.8 *In a public key cryptosystem using RSA algorithm, user uses two prime numbers 5 and 7. He chooses 11 as encryption key, find out decryption key. What will be the ciphertext, if the plaintext is 2 ?* **GTU : Summer-11, Marks 3**

Ans. : Given data : $P = 5; \ Q = 7$

$$N = P \times Q = 5 \times 7$$

$$N = 35$$

$$\phi(N) = (P-1) \times (Q-1)$$

$$= (5-1) \times (7-1)$$

$$\phi(N) = 24$$

$$e \times d = 1 \bmod (\phi(N))$$

$$11 \times d = 1 \bmod(24)$$

$$d = 11$$

$$\text{Cipher text } C = M^e \bmod N = 2^{11} \bmod 35$$

$$C = 18$$

University Questions

1. *Explain RSA algorithm in detail with suitable example.* **GTU : Summer-17, Marks 7**
2. *Discuss RSA Algorithm with suitable example.* **GTU : Winter-17, Marks 7**
3. *Explain key pair generation using RSA algorithm.* **GTU : Summer-18, Marks 4**
4. *Explain encryption and decryption using RSA.* **GTU : Summer-18, Marks 4**
5. *Distinguish between symmetric encryption and asymmetric encryption using suitable example.* **GTU : Winter-18, Marks 3**
6. *Explain in detail RSA algorithm, highlighting its security aspect.* **GTU : Winter-18, Marks 7**
7. *Explain the three approaches to attack RSA mathematically.* **GTU : Summer-19, Marks 3**
8. *Explain process of encryption in RSA algorithm with suitable example.*
   (Prime number P, Q and encryption key E is given for reference) P = 7, Q = 17, E = 7. **GTU : Winter-19, Marks 7**

4.3 Diffie-Hillman Key Exchange Algorithm

GTU : Winter-17, 18, 19, Summer-17, 19

- The Diffie-Hellman key agreement protocol was developed by Diffie and Hellman in 1976. This protocol allows two users to exchange a secret key over an insecure medium without any prior secrets.
- The protocol has two system parameters p and g. They are both public and may be used by all the users in a system.
- Parameter p is a prime number and parameter g is an integer less than p, with the following property :

  1. For every number n between 1 and p – 1 inclusive.

  2. There is a power k of g such that $n = g^k \bmod p$.
- The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $k = g^{ab} \bmod p$ given the two public values $g^a \bmod p$ and $g^b \bmod p$ when the prime p is sufficiently large.

- The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack. This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants. Possible solutions include the use of digital signatures and other protocol variants.
- Suppose Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol. They proceed as follows :
  1. First, Alice generates a random private value a and Bob generates a random private value b.
  2. Both a and b are drawn from the set of integers. They derive their public values using parameters p and g and their private values.
  3. Alice's public value is g^a mod p and Bob's public value is g^b mod p.
  4. They then exchange their public values.
  5. Finally, Alice computes $g^{ab} = (g^b)^a$ mod p.
  6. Bob computes $g^{ba} = (g^a)^b$ mod p.
  7. Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key k.

Algorithm :

- Select two numbers (1) prime number q (2) α an integer that is a primitive root of q.
- Suppose the users A and B wish to exchange a key.
  1. User A select a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A}$ mod q.
  2. User B selects a random integer $X_B < q$ and compute $Y_B = \alpha^{X_B}$ mod q.
  3. Both side keeps the X value private and makes the Y value available publicly to the other side.
  4. User A computes the key as $K = (Y_B)^{X_B}$ mod q.
  5. User B computes the key as $K = (Y_A)^{X_B}$ mod q.
- Both side gets same results :

$$\begin{aligned} K &= (Y_B)^{X_A} \bmod q = (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\ &= (\alpha^{X_B})^{X_A} \bmod q = \alpha^{X_B X_A} \bmod q \\ &= (\alpha^{X_A} \bmod q)^{X_B} \bmod q = (Y_A)^{X_B} \bmod q \end{aligned}$$

Example

- Key exchange is based on the use of the prime number and a primitive root of prime number.

- Prime number $q = 353$
  Primitive root $\alpha = 3$
- A and B select secret keys.
  $X_A = 97 \qquad X_B = 233$
- Calculates the public keys

$$\text{A computes } Y_A = \alpha^{X_A} \bmod q$$

$$= (3)^{97} \bmod 353 = (1.9080 \times 10^{97}) \bmod 353 = 40$$

$$\text{B computes } Y_B = \alpha^{X_B} \bmod q$$

$$= (3)^{233} \bmod 353 = (1.4765 \times 10^{111}) \bmod 353 = 248$$

- After they exchange public keys, each can compute the common *secret key*.

$$\text{A computes K} = (Y_B)^{X_A} \bmod q = (248)^{97} \bmod 353$$

$$= (1.8273 \times 10^{232}) \bmod 353 = 160$$

$$\text{B computes K} = (Y_A)^{X_B} \bmod q = (40)^{233} \bmod 353$$

$$= (1.9053 \times 10^{373}) \bmod 353 = 160$$

Problems

Example 4.3.1 *User A and B use the Diffie-Hellman key exchange technique with a common prime q = 71 and a primitive root α = 7.*

a) If user A has private key $X_A = 5$, what is A's public key Y_A ?

b) If user B has private key $X_B = 12$, what is B's public key Y_B ?

c) What is the shared secret key ?

Solution :

a) A's public key Y_A

$$Y_A = \alpha^{X_A} \bmod q = (7)^5 \bmod 71 = 16807 \bmod 71 = 51$$

b) B's public key Y_B

$$Y_B = \alpha^{X_B} \bmod q = (7)^{12} \bmod 71 = 13841287201 \bmod 71 = 4$$

c) Shared secret key

i) At user A $\quad K = (Y_B)^{X_A} \bmod q$

$$= (4)^5 \bmod 71 = 1024 \bmod 71$$

$$K = 30$$

The man in middle attack can work against the Diffie-Hellman key exchange algorithm, causing it to fail.

Advantages

1. Any user can choose a random x and publish g^x in a public database such as a phone book.
2. Phone book must be maintained by a TTP.
3. Other users can look up the database and get the public key for the individual and use it to encrypt the message.
4. Ideal for use with emails.

Disadvantages

1. Does not protect against man-in-the-middle attacks.
2. Even can intercept all traffic between Alice and Bob and generate separate keys for communication with them.
3. If Allice sends an encrypted message for Bob with his public key, Eve simply forwards it.
4. For large prime p, (p – 1) is an even number and so Z_p^* will have an subgroup of order 2.

Example 4.3.2 *If generator g = 2 and n or P = 11, using Diffie-Hellman algorithm solve the following :*

i) Show that 2 is a primitive root of 11.

ii) If A has a public key '9' what is A's private key ?

iii) If B has a public key '3' what is B's private key ?

iv) Calculate the shared secret key.

Solution : i)

$$2^1 \bmod 11 = 2$$

$$2^2 \bmod 11 = 4$$

$$2^3 \bmod 11 = 8$$

$$2^4 \bmod 11 = 5$$

$$2^5 \bmod 11 = 10$$

$$2^6 \bmod 11 = 9$$

$$2^7 \bmod 11 = 7$$

$2^8 \text{ mod } 11 = 3$

$2^9 \text{ mod } 11 = 6$

Using 2 as integer, we get all the integer values between 1 to 11. So 2 is a primitive root of 11.

ii) Public key = 9

$2^6 \text{ mod } 11 = 9$

$X_A = 6$

iii) $Y_B = (11)^6 \text{ mod } 9$

$Y_B = 1$

iv) Shared secret key :

$K = (Y_B)^{XA} \text{ mod } q$

$K = 3^6 \text{ mod } 11$

$K = 3$

Example 4.3.3 *Calculate the shared secret (K_A and K_B) key using Diffie Hellman Key Exchange Algorithm with suitable example. Take q = 23, α = 5, X_A = 6 and X_B = 15.*

GTU : Winter-17, Marks 4

Solution : A's public key Y_A

$$Y_A = \alpha^{X_A} \text{ mod } q = (5)^6 \text{ mod } 23 = 8$$

B's public key Y_B

$$Y_B = \alpha^{X_B} \text{ mod } q = (5)^{15} \text{ mod } 23 = 19$$

For shared secret key (K_A and K_B)

A compute $K = (Y_B)^{X_A} \text{ mod } q = (19)^6 \text{ mod } 23 = 2$

A compute $K = (Y_A)^{X_B} \text{ mod } q = (8)^{15} \text{ mod } 23 = 2$

University Questions

1. *Discuss Diffie - Hillman key exchange algorithm in detail.*

GTU : Summer-17, Winter-17, Marks 7

2. *Explain man in middle attack in Diffie Hellman key exchange.*

GTU : Winter-17, Marks 4

3. *Briefly explain Diffie Hellman Key exchange with an example.* **GTU : Winter-18, Marks 7**
4. *For Diffi-Hellman algorithm, two publicaly known numbers are prime number 353 and primitive root of it is 3. A selects the random integer 97 and B selects 233. Compute the public key of A and B. Also compute common secret key.* **GTU : Summer-19, Marks 4**
5. *Describe the Diffie Hellman key exchange algorithm with example.* **GTU : Winter-19, Marks 4**

4.4 Short Questions and Answers

Q.1 What is a prime number ?

Ans. : A prime number is an integer than can only be divided without remainder by positive and negative values of itself and 1

Q.2 List out the ingredients of public key encryption scheme.

Ans. : Ingredients of public key encryptions are :

a) Plaintext b) Encryption algorithm

c) Public key d) Private key

e) Cipher-text f) Decryption algorithm

Q.3 Name any two methods for testing prime numbers.

Ans. : Testing prime numbers methods are divisibility algorithm and probabilistic algorithms.

Q.4 Define : Replay attack.

Ans. : 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 send the sequence number is incremented.

Q.5 Define : Primality test.

Ans. : A primality test is a test to determine whether or not a given number is prime, as opposed to actually decomposing the number into its constituent prime factors. Primality tests come in two varieties : deterministic and probabilistic.

Q.6 What is primitive root.

Ans. : A primitive root of a prime p is an integer g such that g (mod p) has multiplicative order p^{-1}

Q.7 What is weak collision resistance ? What is the use of it ?

Ans. : Weak collision-resistance : Given an x and h(x), it is infeasible to find x` such that h(x) = h(x`). This implies that given h(x), it is infeasible to find any x` such that h(x) = h(x`).

Q.8 Why random numbers are used in network security ?

Ans. : Most encryption algorithms require source of random data. Random numbers are necessary not only for generating cryptographic keys but are also needed in steps of cryptographic algorithms or protocols.

Q.9 What is the Diffie-Hellman key exchange ?

Ans. : The purpose for this algorithm is to enable two users to exchange a key securely that can then be used for subsequent encryption of messages. It depends for its effectiveness on the difficulty of computing discrete logarithms.

Q.10 Mention any one technique of attacking RSA.

Ans. : Techniques are

1. Brute force
2. Mathematical attacks
3. Timing attacks
4. Chosen ciphertext attacks.

Q.11 State few applications of RC4 algorithm.

Ans. : RC4 is used in SSL/TLS. It is also used in WEP, the IEEE 802.11 wireless networking security standard. It can also be found in a number of other applications including email encryption products.

4.5 Multiple Choice Questions

Q.1 In public key cryptosystems, the private key is kept by

a sender

b receiver

c sender and receiver

d all the connected devices to the network

Q.2 User A, if wanting to send an authenticated message to user B, it would encrypt the message with A's —— private key.

a public key

b private key

c both key

d third party key

Q.3 In RAS algorithm, the public key pair is ————-.

a [d, n]

b [p, q]

c [e, n]

d [p, n]

Q.4 In RAS algorithm, the private key pair is ————.

a [d, n] b [p q]

c [e, n] d [p, n]

Q.5 Using the RSA algorithm for p = 3,q=11. What is value of n ?

a 8 b 4

c 33 d 20

Q.6 Perform encryption using the RSA algorithm for p = 3, q = 11, e = 7, M = 5.

a 33 b 20

c 35 d 14

Q.7 RSA is a ——— cipher in which the plaintext and ciphertext are integers between 0 and n - 1 for some n.

a block b stream

c private d None

Q.8 Public key encryption is also known as ———— key encryption.

a symmetric b private

c asymmetric d None

Q.9 One commonly used public-key cryptography method is the ______ algorithm.

a RSS b RAS

c RSA d RAA

Q.10 The ———— is the message after transformation.

a ciphertext b plaintext

c secret-text d none

Q.11 The ———— method provides a one-time session key for two parties.

a RSA b Diffie-Hellman

c DES d AES

Answer Keys for Multiple Choice Questions

Q.1	b	**Q.2**	b	**Q.3**	c	Q.4	a
Q.5	c	**Q.6**	d	**Q.7**	a	Q.8	c
Q.9	c	**Q.10**	a	**Q.11**	b		

❑❑❑

5 Cryptographic Hash Functions

Syllabus

Cryptographic Hash Functions, their applications, Simple hash functions, its requirements and security, Hash functions based on Cipher Block Chaining, Secure Hash Algorithm (SHA).

Contents

5.1 Hash Function

GTU : Winter-17, 18, 19, Summer-17, 19

- A hash function takes an input m, and computes a fixed size string known as a hash. Unlike a MAC, a hash code does not use a key but is a function only of the input message.
- **Definition :** A hash function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-values.
- The data to be encoded is often called the "message", and the hash value is sometimes called the message digest or simply digests.
- A hash function maps a variable-length input into a fixed-length output. This hash function output can be treated as a fingerprint of the input data. A very simple example of hash function is modulo operation. Hash functions have been used in many fields of computer science such as hash table in data structure, checksum algorithms for error detection, digital signature in information security etc.
- The most common cryptographic uses of hash functions are with digital signatures and for data integrity.
- When hash functions are used to detect whether the message input has been altered, they are called Modification Detection Codes (MDC).
- There is another category of hash functions that involve a secret key and provide data origin authentication, as well as data integrity; these are called Message Authentication Codes (MACs).
- A hash value h is generated by a function H of the form.

$$h = H(M)$$

where M = Variable – Length message

$H(M)$ = Fixed – Length hash value.

- Hash code is also referred to as a message digest or hash value. A change to any bit or bits in the message results in a change to the hash code.
- Fig. 5.1.1 (a) shows the basic uses of hash function.

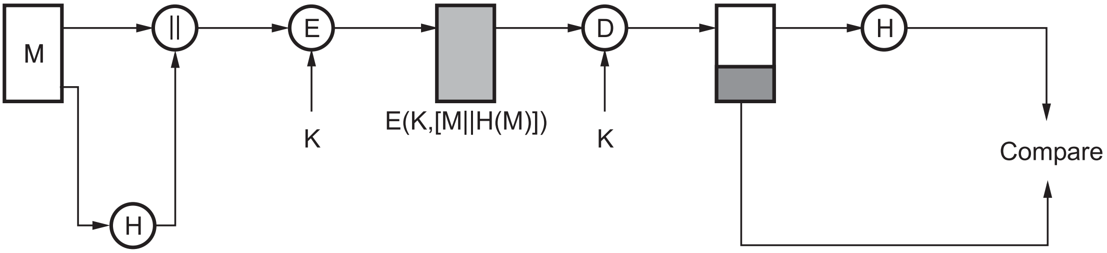


Fig. 5.1.1 (a) Encrypt message plus hash code

1. Encrypt message plus hash code.

Provide confidentiality : Only A and B share K.

Provides authentication : H(M) is cryptographically protected.

2. Encrypt hash code - shared secret key

Only the hash code is encrypted, using symmetric encryption.

Reduces the processing burden for those applications that do not require confidentiality.

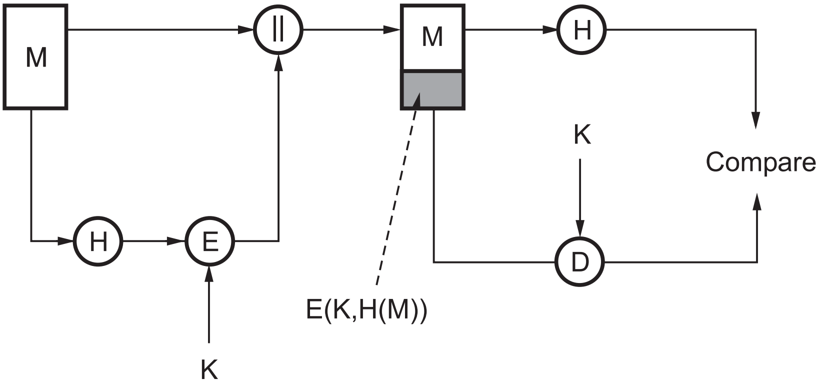


Fig. 5.1.1 (b) Encrypt hash code - shared secret key

3. Encrypt hash code - sender's private key

- It provides authentication and digital signature.

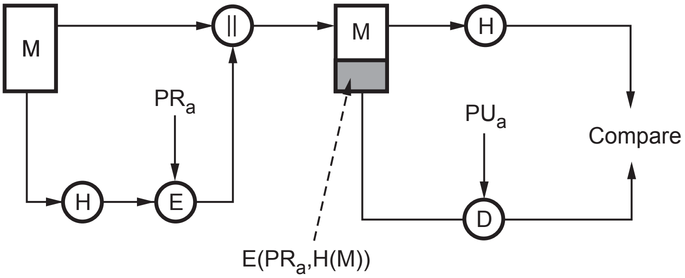


Fig. 5.1.1 (c) Encrypt hash code - sender's private key

5.1.1 Requirements for a Hash Functions

- The purpose of a hash function is to produce a fingerprint of a file, message or other block of data.

Properties

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 called 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 called as **weak collision** resistance.
6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y). This is called as **strong collision** resistance.

5.1.2 One-way Hash Function

- A one-way hash function is also known as a message digest, fingerprint or compression function. It is a mathematical function and takes a variable-length input string and converts it into a fixed-length binary sequence.
- One-way hash function is designed in such a way that it is hard to reverse the process. A good hash function also makes it hard to find two strings that would produce the same hash value. All modern hash algorithms produce hash values of 128 bits and higher.
- Even a slight change in an input string should cause the hash value to change drastically. Even if 1 bit is flipped in the input string, at least half of the bits in the hash value will flip as a result. This is called an **avalanche effect**.
- A common way for one-way hash functions to deal with the variable length input problem is called a compression function. Compression functions work by viewing the data being hashed as a sequence of n fixed-length blocks.
- To compute the hash value of a given block, the algorithm needs two things : The data in the block and an input seed.
- The input seed is set to some constant value, c, and the algorithm computes the hash value h_1 of the first block. Next, the hash value of the first block, h_1 is used as the seed for the second block.
- The function proceeds to compute the hash value of the second block based on the data in the second block and the hash value of the first block, h_1. So, the hash value for block n is related to the data in block n and the hash value h_{n-1} (for n > 1). The hash value of the entire input stream is the hash value of the last block.

5.1.3 Application of Hash Function

- A typical use of a cryptographic hash would be as follows :
  1. Alice poses a tough math problem to Bob, and claims she has solved it. Bob would like to try it himself, but would yet like to be sure that Alice is not

bluffing. Therefore, Alice writes down her solution, appends a random nonce, computes its hash and tells Bob the hash value. This way, when Bob comes up with the solution himself a few days later, Alice can prove that she had the solution earlier by revealing the nonce to Bob.

2. Second application of secure hashes is verification of message integrity. Determining whether any changes have been made to a message, for example, can be accomplished by comparing message digests calculated before, and after, transmission. A message digest can also serve as a means of reliably identifying a file; several source code management systems, including Git, Mercurial and Monotone, use the sha1sum of various types of content (file content, directory trees, ancestry information, etc) to uniquely identify them.
3. A related application is password verification. Passwords are usually not stored in clear text, for obvious reasons, but instead in digest form. To authenticate a user, the password presented by the user is hashed and compared with the stored hash. This is sometimes referred to as one-way encryption.
4. Hash functions can also be used in the generation of pseudorandom bits. Hashes are used to identify files on peer-to-peer file sharing networks. For example, in an ed2k link, an MD4-variant hash is combined with the file size, providing sufficient information for locating file sources, downloading the file and verifying its contents. Magnet links are another example. Such file hashes are often the top hash of a hash list or a hash tree which allows for additional benefits.

5.1.4 Birthday Attack

- A birthday attack refers to a class of brute-force attacks.
- The attack is named after the statistical property of birthday duplication - you only need 23 people to have a larger than 50 % chance that they are born on the same day of the year.
- This is due to the fact that each time you adding one person to the set of people you are looking for duplicates in, you are looking for duplicates against all the people already in the set, not just one of them.
- The same technique can be used to look for conflicts in one-way functions. Instead of taking one output of the one-way function, you create or acquire a set of values (let us call this a) that have a some property and then create another set of other values that have different properties (let us call this b) and try to find any value that is in both a and b. This is a much smaller problem that finding a value that match a particular value in a.

- The properties in a and b might for instance be
  1. a contains secure hashes of an innocent message and b contains one of a less innocent message, so the attacker can substitute the messages at a later date.
  2. a is the password hashes of a system the attacker wants to get an account on, and b is a set of password hashes that the attacker knows the passwords for.
  3. a is the set of public keys from a Discrete Logarithms based cryptosystem where g and p are static, while b is the set of g^e mod p functions that the attacker knows e for.
- Birthday attacks are often used to find collisions of hash functions. To avoid this attack, the output length of the hash function used for a signature scheme can be chosen large enough so that the birthday attack becomes computationally infeasible.
- Resistance against this attack is why the Unix password hashes use a salt.

5.1.5 Attack on Collision Resistance

- Weak collision resistance : for any x, it is hard to find $x' \neq x$ such that $h(x)=h(x')$.
- Strong collision resistance : it is hard to find any x, x' for which $h(x) = h(x')$.
- It's easier to find collisions. Therefore strong collision resistance is a stronger assumption.
- Real world hash functions : MD5, SHA-1, SHA-256.
- The weak collision property refers guarantees that an alternative message yielding the same code cannot be found. This prevents forgery when an encrypted hash code is used.
- The strong collision property refers to how resistant the hash function is to a class of attacks known as the birthday attack.

5.1.6 Requirements and Security

- Attacks are of two types.

  1. Brute-force attack   2. Cryptanalysis

Brute - force attacks

1. Hash functions

- The strength of a hash function against brute-force attacks depends solely on the length of the hash code produced by the algorithm.

- **Desirable properties**
  a. **One way :** For any given code h, it is computationally infeasible to find x such that H(x) = h.

  b. **Weak collision resistance :** For any given block x, it is computationally infeasible to find y ≠ x with H(y) = H(x).

  c. **Strong collision resistance :** It is computationally infeasible to find any pair (x, y) such that H(x) = H(y).

- For a hash code of length n, the level of effort required, as we have seen is proportional to the following :

One way	2^n
Weak collision resistance	2^n
Strong collision resistance	$2^{n/2}$

2. Message authentication codes

- Given one or more text MAC pair [x_i, C(K, x_i)] it is computationally infeasible to compute any text MAC pair [x, C(K, x)] for any new input $x \neq x_i$.
- 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.
- An attacker can also work on the MAC value without attempting to recover the key. Here, the objective is to generate a valid MAC value for a given message or to find a message that matches a given MAC value.
- The level of effort for brute-force attack on a MAC algorithm can be expressed as min (2^k, 2^n).

Cryptanalysis

Hash functions

- The hash algorithm involves repeated use of a compression function (f), that takes two inputs and produces an n-bit output.
- Cryptanalysis of hash functions focuses on the internal structure of f and is based on attempts to find efficient techniques for producing collisions for a single execution of f.

Example 5.1.1 *What is the role of a compression function in a hash function ?*

GTU : Winter-19, Marks 3

Solution :

- A compression function takes a fixed length input and returns a shorter, fixed-length output.
- A typical hash function uses a compression function as a basic building block, and involves repeated application of the compression function.

University Questions

1. *Give differences between hash function and message authentication codes.* **GTU : Summer-17, Marks 3**
2. *State the basic difference(s) between message authentication code and hash function.* **GTU : Winter-17, Marks 3**
3. *Enlist the practical applications of hashing.* **GTU : Winter-17, Marks 4**
4. *Discuss HASH function and its application in Crypto System.* **GTU : Winter-18, Marks 3**
5. *What is the difference between weak and strong collision resistance ? Consider the hash functions based on cipher block chaining, what kind of attack can occur on this ?* **GTU : Summer-19, Marks 4**

5.2 Applications of Cryptographic Hash Functions

- Message Authentication and Digital Signatures are the two main application of the cryptographic hash function.

Message authentication

- Message authentication is an alternative technique which uses secret key. This technique assumes that two communicating parties, share a common secret key K. When A has a message to send to B, it calculates the MAC.

MAC = C(K, M)

where,
M = Input message
C = MAC function
K = shared secret key
MAC = Message authentication code.

- Calculated MAC and message are transmitted to the receiver. The receiver performs the same calculation on the received message.
- Received MAC is compared with the calculated MAC. If both are matches, then
  1. The receiver is assured that the message has not been altered.

2. The receiver is assured that the message is from the alleged sender.
3. If the message includes a sequence number, then the receiver can be assumed of the proper sequence because an attacker cannot successfully alter the sequence number.

- Fig. 5.2.1 shows the message authentication.

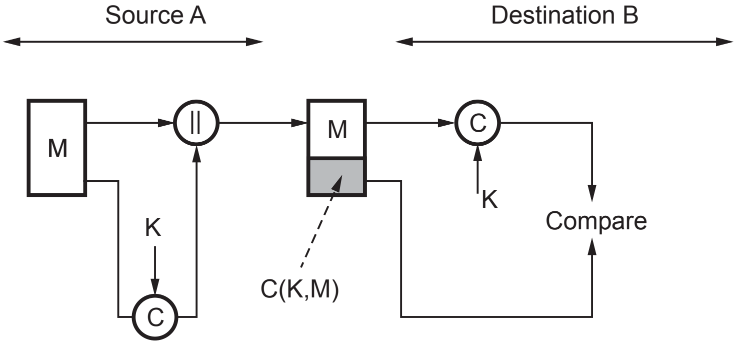


Fig. 5.2.1 Message authentication

- Above figure provides authentication but not confidentiality. Confidentiality can be provided by performing message encryption either after or before the MAC algorithm.
- Fig. 5.2.2 shows encryption after the MAC.

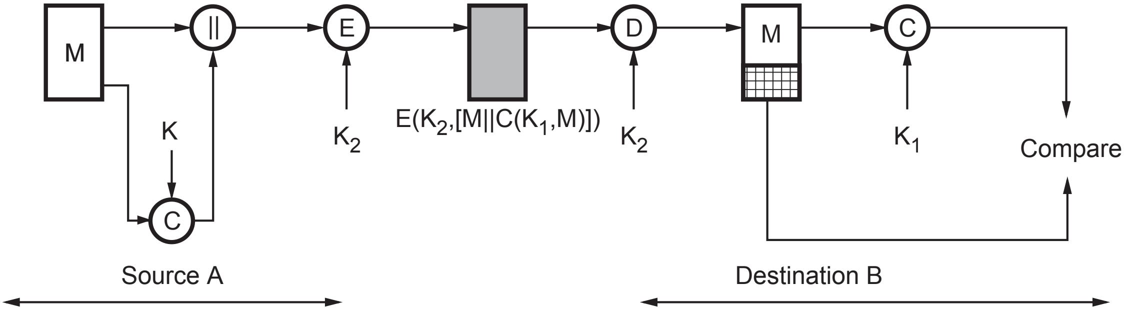


Fig. 5.2.2 Message authentication and confidentiality

- Two separate keys are needed, each of which is shared by the sender and the receiver. Here MAC is calculated with the message input and is then concatenated to the message. The entire block is then encrypted.
- When a hash function is used to provide message authentication, the hash function value is often referred to as a **message digest.**
- Message authentication is achieved using a Message Authentication Code (MAC), also known as a keyed hash function.

Digital Signatures

- In the digital signature, the hash value of a message is encrypted with a user's private key. Anyone who knows the user's public key can verify the integrity of the message that is associated with the digital signature.
- If confidentiality as well as a digital signature is desired, then the message plus the private-key-encrypted hash code can be encrypted using a symmetric secret key.

Other Applications

- One-way password file is created by hash function. Hash functions can be used for intrusion detection and virus detection.
- A cryptographic hash function can be used to construct a pseudorandom function (PRF) or a pseudorandom number generator. A common application for a hash-based PRF is for the generation of symmetric keys.

5.3 Simple Hash Functions

- For a hash function, the input is viewed as a sequence of n-bit blocks. The input is processed one block at a time in an iterative fashion to produce an n-bit hash function.
- One of the simplest hash functions is the bit-by-bit exclusive-OR of every block. This can be expressed as follows :

$$C_i = b_{i1} \oplus b_{i2} \oplus b_{i3} \oplus \ldots\ldots\ldots \oplus b_{im}$$

where C_i = i^{th} bit of the hash code, $1 \leq i \leq n$.

m = number of n-bit blocks in the input

b_{ij} = i^{th} bit in j^{th} block

$\oplus$ = XOR operation

- Fig. 5.3.1 shows two types of hash functions. (See Fig. 5.3.1 on next page)
- A simple way to improve matters is to perform a one bit circular shift or rotation, on the hash value after each block is processed. The procedure is as follows :
  1. Initially set the n-bit hash value to zero.
  2. Process each successive n-bit block of data as follows.
     a. Rotate the current hash value to the left by one bit.
     b. XOR the block into the hash value.

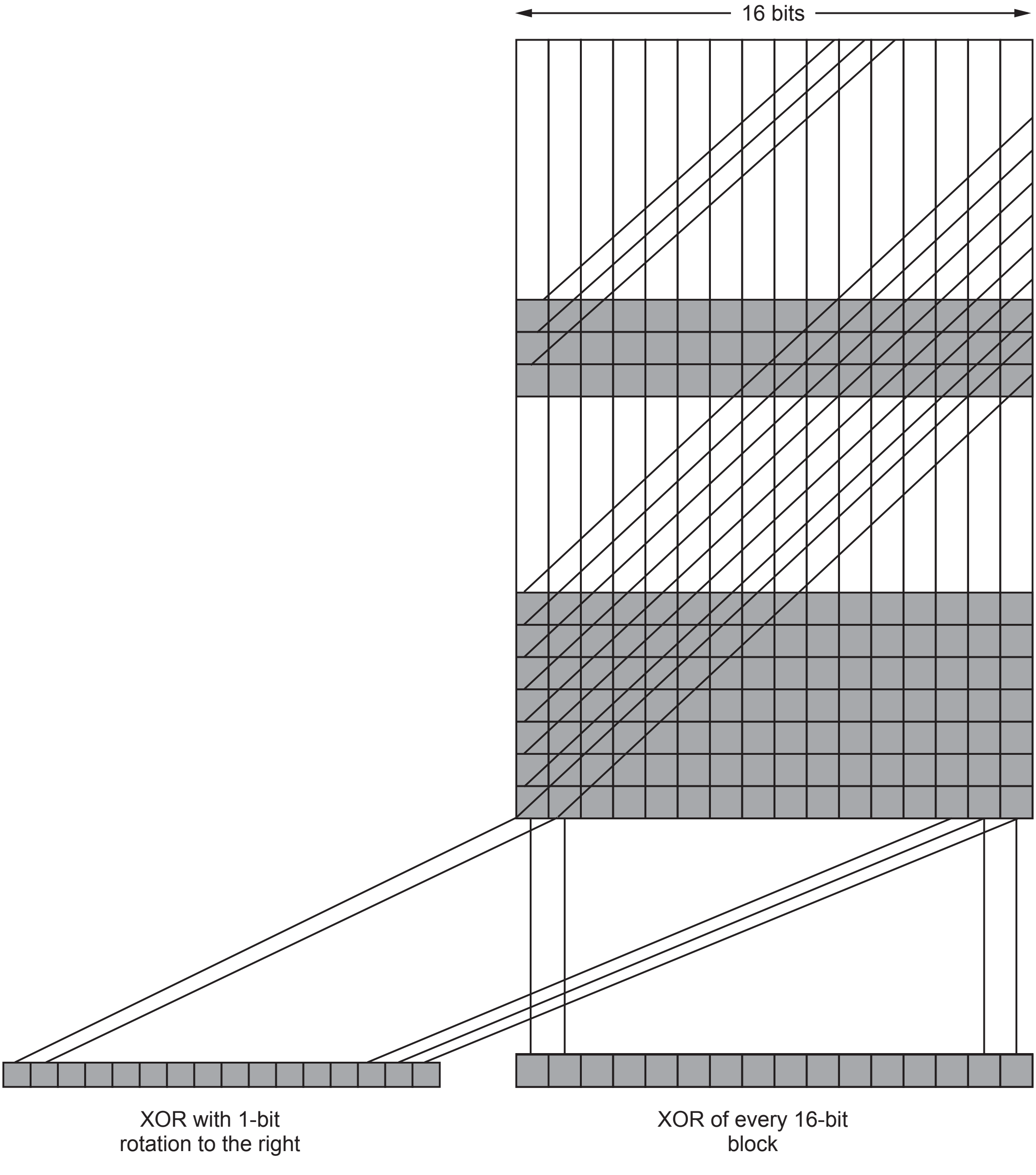

16 bits
XOR with 1-bit rotation to the right
XOR of every 16-bit block

Fig. 5.3.1 Two simple hash functions

University Question

1. *Write requirements for hash function and briefly explain simple hash function.*

GTU : Summer-17, Marks 7

5.4 Hash Functions Based on Cipher Block Chaining GTU : Summer-17

- Two major categories of hash functions are : **dedicated hash functions and block** cipher-based hash functions.
- Block cipher is a popular encryption-decryption primitive. To encrypt, the block cipher accepts a key K and a plaintext block x as input and produces a cipher text block c = E(K, x), also written as $c = E_K(x)$.
- Given a message M consisting of a sequence of 64-bit blocks $P_1; P_2; : : : ; P_N$, define the hash code h = H(M) as the block-by-block XOR of all blocks and append the hash code as the final block :

$$h = P_{N+1} = P_1 \oplus P_2 \oplus ...... \oplus P_N$$

- Encrypt the entire message plus the hash code using CBC mode to produce the encrypted message $C_1; C_2; : : ; C_{N+1}$. There are several ways the ciphertext can be manipulated in such a way that it is not detectable by the hash code.
- By the definition of CBC :

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

So we have;

$$P_1 = IV \oplus D(K, C_1)$$

$$P_i = C_{i-1} \oplus D(K, C_i)$$

$$P_{N+1} = C_N \oplus D(K, C_{N+1})$$

- But, P_{N+1} has the hash code.

$$P_{N+1} = P_1 \oplus P_2 \oplus ...... \oplus P_N$$

$$= [IV \oplus D(K, C_1)] \oplus [C_1 \oplus D(K, C_2)]. \oplus ... \oplus [Y_{C+1} \oplus D(K, C_N)]$$

- Because the terms in the preceding equation can be XOR'ed in any order, it follows that the hash code would not change if the ciphertext blocks were permuted.

5.5 Secure Hash Algorithm (SHA) GTU : Winter-17, 19

- The Secure Hash Algorithm (SHA) was developed by National Institute of Standards and Technology (NIST). It is based on the MD4 algorithm. Based on different digest lengths, SHA includes algorithms such as SHA-1, SHA-256, SHA-384, and SHA-512.
- Unlike encryption, given a variable length meassge **x**, a secure hash algorithm computes a function **h(x)** which has a fixed and often smaller number of bits. When a message of any length is less than 2^{64} bits is input, the SHA-1 produces a 160-bit output called message digest.

- SHA-1 called secure bacause it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest.
- There are a number of attacks on SHA-1, all relating to what is known as collision resistance. For examples, if you are using SHA-1 for the storage of passwards, there are no passoword recovery attacks as at December 2011 that make use of the collision attacks on SHA-1.
- The most commonly used hash function from the SHA family is SHA-1. It is used in many applications and protocols that require secure and authenticated communications. SHA-1 is used in SSL/TLS, PGP, SSH, S/MIME, and IPSec.

Features of SHA-1 :

1. The SHA-1 is used to compute a message digest for a message or data file that is provided as input.
2. The message or data file should be considered to be a bit string.
3. The length of the message is the number of bits in the message (the empty message has length 0).
4. If the number of bits in a message is a multiple of 8, for compactness we can represent the message in hex.
5. The purpose of message padding is to make the total length of a padded message a multiple of 512.
6. The SHA-1 sequentially processes blocks of 512 bits when computing the message digest.
7. The 64-bit integer is 1, the length of the original message.
8. The padded message is then processed by the SHA-1 as n 512-bit block.

- SHA-1 was cracked in the year 2005 by two different research groups. In one of these two demonstrations, Xiaoyun Wang, Yigun Lisa Yin, and Hongbo Yu demonstrated that it was possible to come up with a collosion for SHA-1 within a space of size only 2^{69}, which was far fewer that the security level of 2^{80} that is associated with this hash function.
- New hash function SHA-512 is introduced to overcome problem of SHA-1.

5.5.1 Secure Hash Algorithm (SHA-512)

- The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST). SHA-1 produces a hash value of 160 bits. In 2002, NIST produced a revised version of the standard, FIPS 180-2, that defined three new version of SHA, with hash value lengths of 256,384 and 512 bits, known as SHA-256, SHA-384 and SHA-512.

- Comparison of SHA parameters

Sr. No.	Parameters	SHA-1	SHA-256	SHA-384	SHA-512
1.	Message digest size	160	256	384	512
2.	Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
3.	Block size	512	512	1024	1024
4.	Word size	32	32	64	64
5.	Number of steps	80	64	80	80
6.	Security	80	128	192	256

- For both SHA-1 and SHA-256, one begins by converting the message to a unique representation of the message that is a multiple of 512 bits in length, without loss of information about its exact original length in bits, as follows : Append a 1 to the message. Then add as many zeroes as necessary to reach the target length, which is the next possible length that is 64-bits less than a whole multiple of 512 bits. Finally, as a 64-bit binary number, append the original length of the message in bits.

Description of SHA-1

- Expand each block of 512, when it is time to use it, into a source of 80 32-bit subkeys as follows : The first 16 subkeys are the block itself. All remaining subkeys are generated as follows : Subkey N is the exclusive OR of subkeys N-3, N-8, N-14 and N-16, subjected to a circular left shift of one place. Starting from the 160-bit block value (in hexadecimal).

  67452301 EFCDAB89 98BADCFE 10325476 C3D2E1F0

  As input for the processing of the *first* 512-bit block of the modified message, for each message block, do the following -

- Encipher the starting value using the 80 sub keys for the current message block. Add each of the 32-bit pieces of the cipher text result to the starting value, modulo 2^32, of course and use that result as the starting value for handling the next message block. The starting value created at the end of handling the last block is the hash value, which is 160 bits long.

The SHA "block cipher" component

- The main calculation in SHA enciphers a 160-bit block using 80 32-bit subkeys in 80 rounds. This calculation is somewhat similar to a series of Feistel rounds, except that instead of dividing the block into two halves, it is divided into five pieces. An F-function is calculated from four of the five pieces, although it is really the XOR of a function of three of the pieces and a circular left shift of a fourth, and XORed with one piece, which is also modified by being XORed with the current round's subkey and a constant. The same constant is used over each group of 20 rounds. One of the other blocks is also altered by undergoing a circular left shift, and then the (160-bit) blocks are rotated.
- The F-function, as well as the constant, is changed every 20 rounds. Calling the five pieces of the 160-bit block being "encrypted" a, b, c, d and e, the rounds of the SHA "block cipher" component proceed as follows
- Change a by adding the current constant to it. The constants are, in hexadecimal
  - For rounds 1 to 20 : 5A827999
  - For rounds 21 to 40 : 6ED9EBA1
  - For rounds 41 to 60 : 8F1BBCDC
  - For rounds 61 to 80 : CA62C1D6
- Change a by adding the appropriate subkey for this round to it.
- Change a by adding e, circular left-shifted 5 places to it.
- Change a by adding the main f-function of b, c and d to it, calculated as follows :
  - For rounds 1 to 20, it is (b AND c) OR (NOT b) AND (d).
  - For rounds 21 to 40, it is b XOR c XOR d.
  - For rounds 41 to 60, it is (b AND c) OR (b AND d) OR (c AND d).
  - For rounds 61 to 80, it is again b XOR c XOR d.
- Change d by giving it a circular *right* shift of 2 positions (or, for consistency, a circular left shift of 30 places.)
- Then swap pieces, by moving each piece to the next earlier one, except that the old a value is moved to e.

There are various types in SHA such as SHA-256, SHA-384, and SHA-512.

SHA-512 logic

- The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces as output a 512-bit message digets. The input is processed in 1024-bit blocks.

Fig. 5.5.1 shows message digest generation using SHA-512.

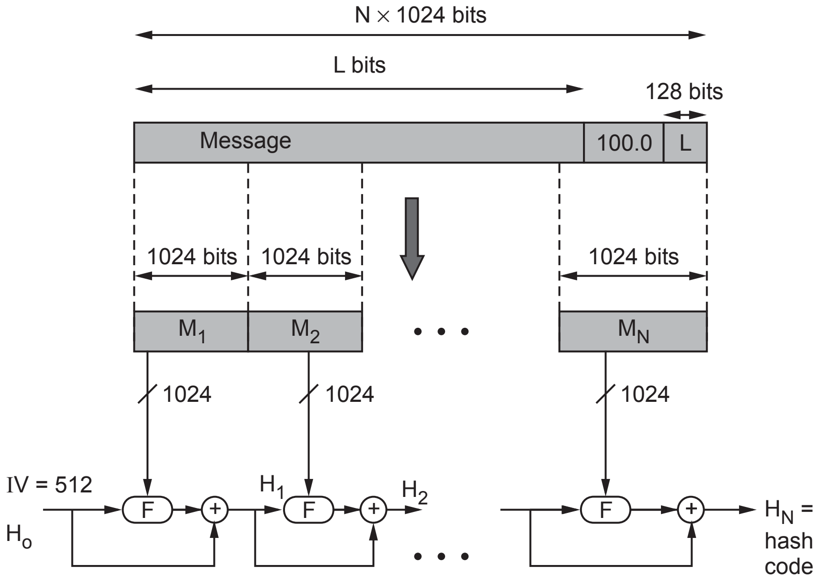


Fig. 5.5.1 Message digest using SHA-512

Steps

1. **Append padding bits :** The message is padded so that its length is congruent to 896 modulo 1024. Padding consists of a single 1-bit followed by the necessary number of 0-bits.
2. **Append length :** A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer that contains the length of the original message (before the padding).
3. **Initialize has 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 initialised to the following 64-bit integers (hexadecimal values)

Sr. No.	Register	Values
1.	a	6A09E667F3BCC908
2.	b	BB67AE8584CAA73B
3.	c	3C6EF372FE94F82B
4.	d	A54FF53A5F1D36F1
5.	e	S10E527FADE682D1
6.	f	9B05688C2B3E6C1E
7.	g	1F83D9ABFB41BD6B
8.	h	5BE0CDI9137E2179

4. **Process message in 1024-bit blocks :** It consist of 80 rounds. Each round takes as input the 512-bit buffer value abcdefgh and updates the contents of the buffer. Each round t makes use of a 64-bit value W_t. The output of the last round is added to the input to the first round (H_{i-1}) to produce H_i.

- Fig. 5.5.2 shows the processing of a single 1024-bit block.

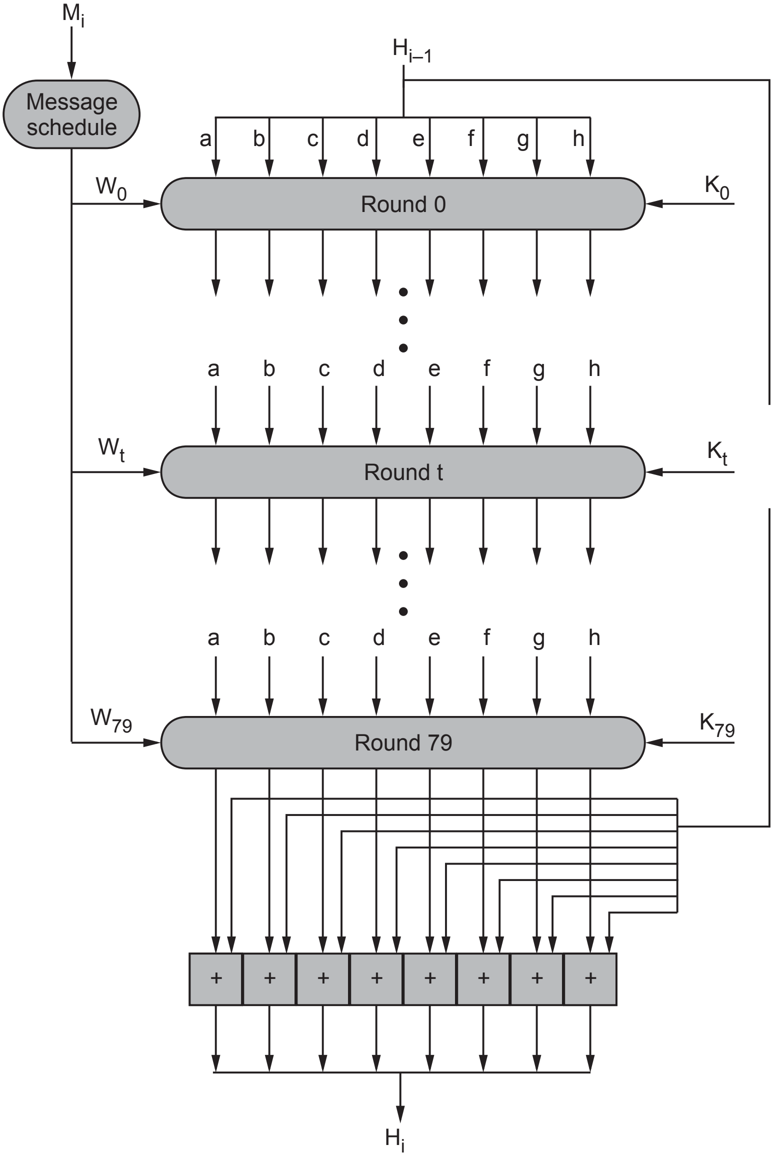


Fig. 5.5.2 SHA-512 processing of a single 1024-bit block

5. **Output :** The output from the N^{th} stage is the 512-bit message digest.

- The behaviour of SHA-512 is as follows

$$H_0 = IV$$

$$H_i = SUM_{64}\ (H_{i-1}, \text{abcdefghj})$$

$$MD = H_N$$

where IV = Initial value of the abcdefgh buffer.

abcdefgh_i = The output of the last round of processing of the i^{th} message block.

N = The number of blocks in the message.

SUM_{64} = Addition modulo 2^{64} performed separately on each word of the pair of inputs.

MD = Final message digest value

SHA-512 round function

Each round is defined by the following set of equations.

$$T_1 = h + ch(e, f, g) + \left(\sum\nolimits_1^{512} e\right) + W_t + K_t$$

$$T_2 = \left(\sum\nolimits_0^{512} a\right) + Maj(a, b, c)$$

$$a = T_1 + T_2$$

$$b = a$$

$$c = b$$

$$d = c$$

$$e = d + T_1$$

$$f = e$$

$$g = f$$

$$h = g$$

Fig. 5.5.3 shows single round operation.

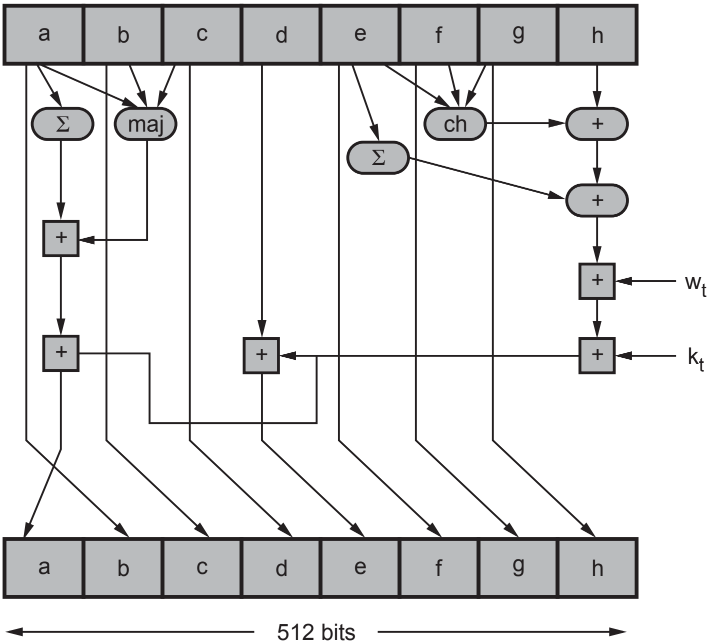


Fig. 5.5.3 Single round operation

University Questions

1. *Write a detailed note on Secure Hash Algorithm.* **GTU : Winter-17, Marks 7**
2. *Explain working of secure hash algorithm, with basic arithmetical and logical functions used in SHA.* **GTU : Winter-19, Marks 7**

5.6 Message Digest

GTU : Winter-11, Summer-13, 15, Winter-14, 15

- A message-digest algorithm is also called **a hash function** or a cryptographic hash function.
- It accepts a message as input and generates a fixed-length output, which is generally less than the length of the input message. The output is called a **hash value**, a fingerprint or a message digest.
- Message Digest 5 (MD5) processes the input text in 512-bit blocks. These blocks are further divided into 16 32-bit sub blocks.
- MD5 is a 128-bit hash.
- The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private key under a public-key cryptosystem such as RSA.

5.6.1 MD5 Description

- Suppose if we have b-bit message as input, and that we wish to find its message digest. Here b is an arbitrary non-negative integer; b may be zero, it need not be a multiple of eight, and it may be arbitrarily large. The bits of the message written down as follows :

  m_0 m_1 ... m_{b-1}

- The following five steps are performed to compute the message digest of the message.

Step 1 : Append Padding Bits

- The message is "padded" so that its length is congruent to 448, modulo 512. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512.
- Padding is performed as follows : a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

Step 2 : Append Length

- A 64-bit representation of b is appended to the result of the previous step. In the unlikely event that b is greater than 2^{64}, and then only the low-order 64 bits of b are used.
- At this point the resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16 (32-bit) words.
- Let M[0 ... N-1] denote the words of the resulting message, where N is a multiple of 16.

Step 3 : Initialize MD Buffer

- A four-word buffer (A, B, C, and D) is used to compute the message digest. Here each of A, B, C, D is a 32-bit register. These registers are initialized to the following values in hexadecimal :

  Word A : 01 23 45 67

  Word B : 89 ab cd ef

  Word C : fe dc ba 98

  Word D : 76 54 32 10

Step 4 : Process Message in 16-Word Blocks

- We first define four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word.

F(X,Y,Z) = XY v not(X) Z

G(X,Y,Z) = XZ v Y not(Z)

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X v not(Z))

- In each bit position F acts as a conditional: if X then Y else Z. The function F could have been defined using + instead of v since XY and not(X)Z will never have 1's in the same bit position.
- It is interesting to note that if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X, Y, Z) will be independent and unbiased.
- The functions G, H, and I are similar to the function F, in that they act in "bitwise parallel" to produce their output from the bits of X, Y, and Z, in such a manner that if the corresponding bits of X, Y, and Z are independent and unbiased, then each bit of G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased.
- This step uses a 64-element table T[1 ... 64] constructed from the sine function. Let T[i] denote the i-th element of the table, which is equal to the integer part of 4294967296 times abs(sin(i)), where i is in radians.

Step 5 : Output

- The message digest produced as output is A, B, C, and D. That is, we begin with the low-order byte of A, and end with the high-order byte of D.

5.6.2 Differences between MD4 and MD5

The following are the differences between MD4 and MD5 :

1. A fourth round has been added.
2. Each step now has a unique additive constant.
3. The function g in round 2 was changed from (XY v XZ v YZ) to (XZ v Y not(Z)) to make g less symmetric.
4. Each step now adds in the result of the previous step. This promotes a faster "avalanche effect".
5. The order in which input words are accessed in rounds 2 and 3 is changed, to make these patterns less like each other.
6. The shift amounts in each round have been approximately optimized, to yield a faster "avalanche effect." The shifts in different rounds are distinct.

5.6.3 Comparison between MD5 and SHA

Sr. No.	MD5	SHA
1.	MD length is 128-bits	Length is 160-bits
2.	Speed is faster than SHA	Slower than MD5
3.	Number of iteration is 64	Number of iteration is 80
4.	Buffer space is 128-bits	Buffer space is 160-bits
5.	MD5 is vulnerable to cryptanalytic attacks	SHA-1 appears not to be vulnerable to cryptanalytic attack
6.	MD5 uses a little endian scheme	SHA-1 uses a big endian scheme
7.	Simple to implement and do not need any large programs or complex table	Simple to implement and do not need any large programs or complex table.
8.	No limit on maximum message size.	Maximum message size is $2^{64} - 1$ bits.

University Questions

1. *Explain MD5 hash algorithm.* **GTU : Dec.-11, Marks 7**
2. *Explain four passes of MD5 message digest algorithm.* **GTU : Summer-13, Marks 7**
3. *Explain MD5 algorithm.* **GTU : Winter-14, Marks 7**
4. *Write MD5 algorithm.* **GTU : Summer-15, Marks 7**
5. *Describe MD5 message digest algorithm.* **GTU : Winter-15, Marks 7**

5.7 Short Questions and Answers

Q.1 What is a Hash function ?

Ans. : A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h(that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties. The basic requirements for a cryptographic hash function are :

- The input can be of any length,
- The outupt has a fixed length,
- H(x) is relatively easy to compute for any given x,
- H(x) is one-way,
- H(x) is collision-free.

Q.2 What types of attacks are addressed by message authentication ?

Ans. :

- **Content modification :** Changes to the contents of the message.
- **Sequence modification :** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
- **Timing modification :** Delay or replay of messages.

Q.3 What is the function of a compression function in a hash function ?

Ans. : The hash function involves repeated use of a compression function. The motivation is that if the compression function is collision resistant, then the hash function is also collision resistant function. So a secure hash function can be produced.

Q.4 What is the use of digital signature ?

Ans. : Data appended to, or 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.

Q.5 What is a birthday attack ?

Ans. : A birthday attack is a name used to refer to class of brute-force attacks. It gets its name from the surprising result that the probability that two or more people in a group of 23 share the same birthday is greater than 1/2; such a result is called a birthday paradox.

Q.6 What do you mean by one-way property in hash function ?

Ans. : For any given value h, it is computationally infeasible to find x such that H(x) = h.

Q.7 What is one-way property ?

Ans. : A function that maps an arbitrary length message to a fixed length message digest is a one-way hash function if it is a one-way function.

5.8 Multiple Choice Questions

Q.1 A hash function maps a _______ input into a _______ output.

a Fixed length, variable length b Variable length, fixed length

c Fixed length, fixed length d Variable length, variable length

Q.2 Message authentication is achieved using a message authentication code, also known as a ---- hash function.

a simple b secure c keyed d collision

Q.3 Birthday attack can be prevented by, ______.

a using a non cryptographic hash function

b using keyed hash

c padding the vulnerable message

d increasing the width of the hash value

Q.4 Which of the following hash functions is still considered secure against collision attacks ?

a SHA-1 b SHA-256 c MD-4 d MD-5

Q.5 When a hash function is used to provide message authentication, the hash function value is often referred to as a _______.

a Digital signature b SHA c message digest d All of these

Q.6 MD5 is a _______ bit hash.

a 64 b 128 c 256 d All of these

Q.7 MD5 processes the input text in _______ bit blocks.

a 128 b 256 c 512 d 1024

Q.8 Number of iteration is MD5 is _______.

a 16 b 24 c 32 d 64

Answer Keys for Multiple Choice Quesions

Q.1	b	**Q.2**	c	**Q.3**	d
Q.4	b	**Q.5**	c	**Q.6**	b
Q.7	c	**Q.8**	d		

❑❑❑

6 Message Authentication Codes

Syllabus

Message Authentication Codes, its requirements and security, MACs based on Hash Functions, Macs based on Block Ciphers.

Contents

6.1 Message Authentication Codes (MAC) GTU : Summer-17, Winter-19

- Message authentication is a mechanism or service used to verify the integrity of a message. Message integrity guarantees that the message has not been changed. Message authentication guarantees that the sender of the message is authentic.
- A MAC algorithm, sometimes called a keyed hash function accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC. The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers to detect any changes to the message content.

Properties of Message Authentication Codes

1. Cryptographic checksum : A MAC generates a cryptographically secure authentication tag for a given message.
2. Symmetric : MACs are based on secret symmetric keys. The signing and verifying parties must share a secret key.
3. Arbitrary message size : MACs accept messages of arbitrary length.
4. Fixed output length : MACs generate fixed-size authentication tags.
5. Message integrity : MACs provide message integrity: Any manipulations of a message during transit will be detected by the receiver.
6. Message authentication : The receiving party is assured of the origin of the message.
7. No non-repudiation : Since MACs are based on symmetric principles, they do not provide non-repudiation.

- MACs provide two security services, message integrity and message authentication, using symmetric ciphers. MACs are widely used in protocols. Both of these services are also provided by digital signatures, but MACs are much faster.
- MACs do not provide non-repudiation.
- In practice, MACs are either based on block ciphers or on hash functions.
- HMAC is a popular MAC used in many practical protocols such as Transport Layer Security (TLS) indicated by a small lock in the browser.

Applications of MAC

- Following are the situations in which MAC used.
  1. Application in which the same message is broadcast to a number of destinations.
  2. Authentication of a computer program in plaintext is an attractive service.

3. Another scenario is an exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages.

- Message Authentication Codes (MAC) also known as a cryptographic check. The MAC is generated by a function C.

$$\text{MAC} = C(K, M)$$

where M = Variable length message

K = Secret key shared only by sender and receiver.

$C(K, M)$ = Fixed length authenticator

- Security of the MAC generally depends on the bit length of the key. Weakness of the algorithm is the brute force attack.
- For a ciphertext - only attack, the opponent, given ciphertext C, would perform $P_i = D(K_i, C)$ for all possible key values K_i until a P_i was produced that matched the form of acceptable plaintext.

Suppose the key size is greater than the MAC size :

- **Round 1**

Given : M_1, $\text{MAC}_1 = C(K_1 M_1)$

Compute $\text{MAC}_i = C(K_i, M_1)$ for all 2^k keys

Number of matches $\approx 2^{(k-n)}$

- **Round 2**

Given : M_2, $\text{MAC}_2 = C(K, M_2)$

Compute $\text{MAC}_i = C(K_i, M_2)$ for all $2^{(k-n)}$ keys resulting from Round 1

Number of matches $\approx 2^{(k-2\times n)}$

- On average, α rounds will be needed if $K = \alpha \times n$

For example : If the key size is 80-bit and MAC is 32 bits long, then the first round will produce about 2^{48} possible keys.

Key length is less than or equal to MAC length

- First round will produce a single match.
- It is possible that more than one key will produce such a match, in which case the opponent would need to perform the same test on a new (message, MAC) pair.

Consider the following MAC algorithm.

- Let $M = (X_1 \,||\, X_2 \,||\, \ldots\ldots\ldots \,||\, X_m)$ be a message that is treated as a concatenation of 64-bit blocks X_i. Then define

$$\Delta(M) = X_1 \oplus X_2 \oplus X_3 \oplus \ldots\ldots\oplus X_m$$

$$C(K, M) = E(K, \Delta(M))$$

Where ⊕ is the exclusive-OR (XOR) and the encryption algorithm is DES in electronic codebook mode.

- Key length = 56 bits
  MAC length = 64 bits
- If an opponent observes {M || C(K, M)}, a brute force attempt to determine. K will require at least 2^{56} encryptions.
- Assume that an opponent knows the MAC function C but does not know K. Then the MAC function should satisfy the following requirements :

1. If an opponent observes M and C(K, M), it should be computationally infeasible for the opponent to construct 0 message M′ such that
   C(K, M′) = C(K, M).
2. C(K, M) should be uniformly distributed in the sense that for randomly chosen messages, M and M′, the probability that C(K, M) = C(K, M′) is 2^{-n}, where n is the number of bits in the MAC.
3. Let M′ be equal to some known transformation on M. That is, M′ = f(M).

Message authentication code based on DES

- The data authentication algorithm based on DES, has been one of the most widely used MAC for a number of years. The algorithm can be defined as using the cipher block chaining mode of operation of DES with an initialization vector of zero.
- Fig. 6.1.1 shows the data authentication algorithm.

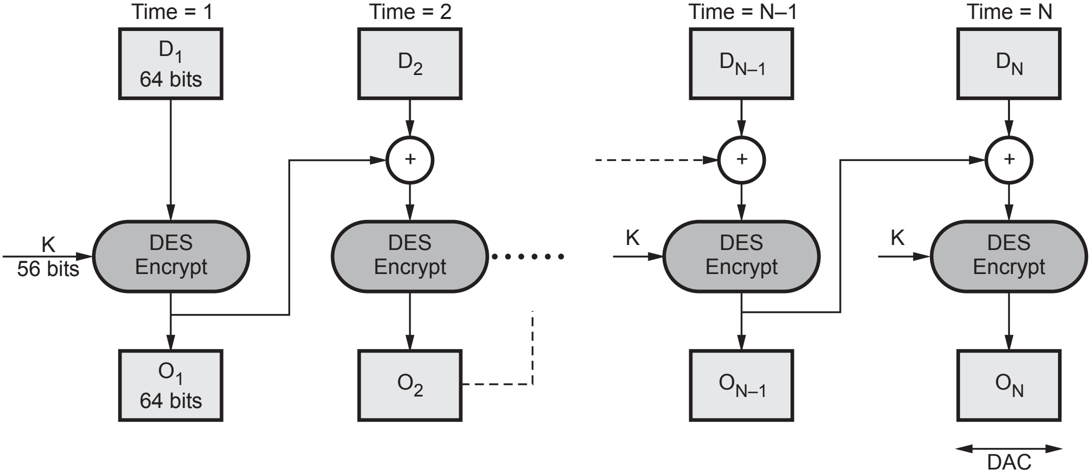


Fig. 6.1.1 Data authentication algorithm

- The algorithm can be defined as using the cipher block chaining mode of operation of DES. The data to be authenticated are grouped into contiguous 64-bit blocks : D_1, D_2, D_3, …………, D_N.

- Using the DES encryption algorithm (E) and a secret key (K), a data authentication code (DAC) is calculated as follows

$$
\begin{aligned}
O_1 &= E(K, D_1) \\
O_2 &= E(K, [D_2 \oplus O_1]) \\
O_3 &= E(K, [D_3 \oplus O_2]) \\
&\vdots \\
&\vdots \\
O_N &= E(K, [D_N \oplus O_{N-1}])
\end{aligned}
$$

The DAC consists of either the entire block O_N or the leftmost M bits of the block, with $16 \leq M \leq 64$.

6.1.1 Authentication Requirements

- Attacks can be identified as follows :
  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.
  3. **Masquerade :** Insertion of messages into the network from a fraudulent source.
  4. **Sequence modification :** Any modification to a sequence of messages between parties, including insertion, deletion and reordering.
  5. **Content modification :** Changes to the contents of a message, including insertion, deletion, transposition and modification.
  6. **Timing modification :** Delay or replay of messages.
  7. **Source repudiation :** Denial of transmission of message by source.
  8. **Destination repudiation :** Denial of receipt of message by destination.
- Message authentication is a procedure to verify that received messages come form the alleged source and have not been altered.
- Digital signature is an authentication technique that also includes measures to counter repudiation by the source.

University Questions

1. *Write a note on: Message Authentication Codes.* **GTU : Summer-17, Marks 7**

2 *Describe MAC with it's security implications.* **GTU : Winter-19, Marks 7**

6.2 MAC Based on Hash Functions

GTU : Summer-19

- The IPsec authentication scheme uses a scheme called Hashed Message Authentication Codes (HMAC), which is an encrypted message digest described in RFC 1024.
- HMAC uses a shared secret key between two parties rather than public key methods for message authentication.

Objectives for HMAC

1. To use, without modifications, available hash function.
2. To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required.
3. To use and handle keys in a simple way.
4. To preserve the original performance of the hash function without incurring a significant degradation.
5. To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions about the embedded hash function.

HMAC algorithm

- Fig. 6.2.1 shows HMAC structure.
- Define the following terms :

H = Embedded hash function

IV = Initial value input to hash function

M = Message input to HMAC

Y_i = i^{th} block of M, $0 \leq i \leq (L - 1)$

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 $\geq n$

K^+ = K padded with zeros on the left so that the result is b bits in length.

ipad = 00110110 (36 in hexadecimal) repeated b/8 times

opad = 01011100 (5C in hexadecimal) repeated b/8 times.

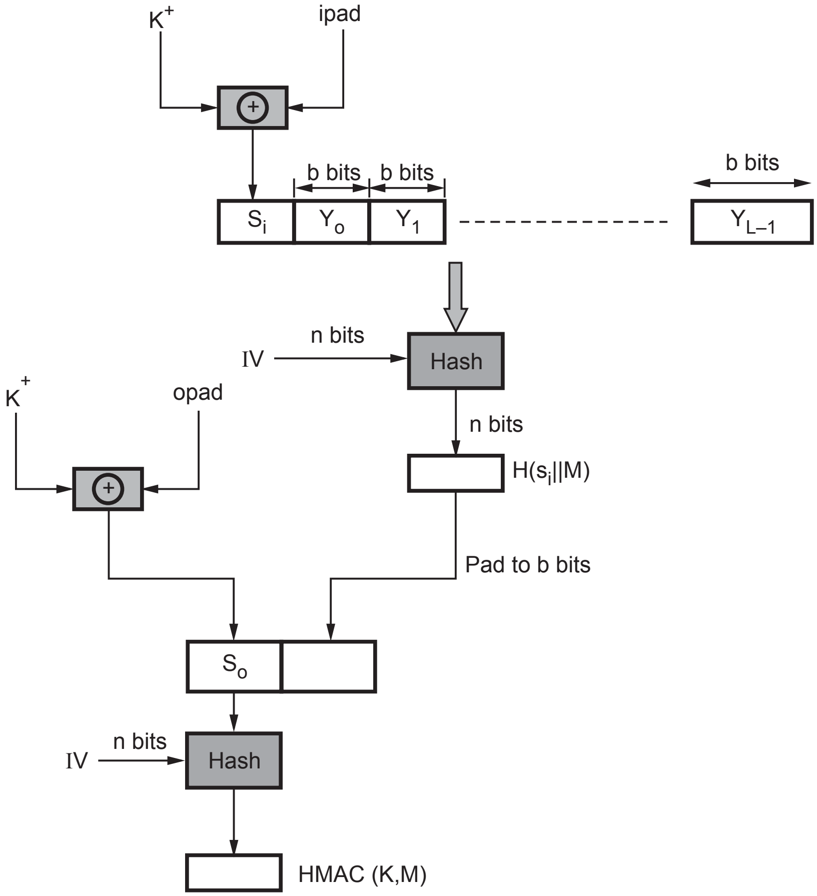


Fig. 6.2.1 HMAC structure

Then HMAC can be expressed as follows :

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

1. Append zeros to the left end of K to create a b-bit string K^+.
2. XOR K^+ with ipad to produce the b-bit block S_i.
3. Append M to S_i.
4. Apply H to the stream generated in step 3.
5. XOR K^+ with opad to produce the b-bit block S_o.
6. Append the hash result from step 4 to S_o.
7. Apply H to the stream generated in step 6 and output the result.

- A more efficient implementation is possible, as shown in Fig. 6.2.2. Two quantities are precomputed :

  $f(IV, (K^+ \oplus ipad))$

  $f(IV, (K^+ \oplus opad))$

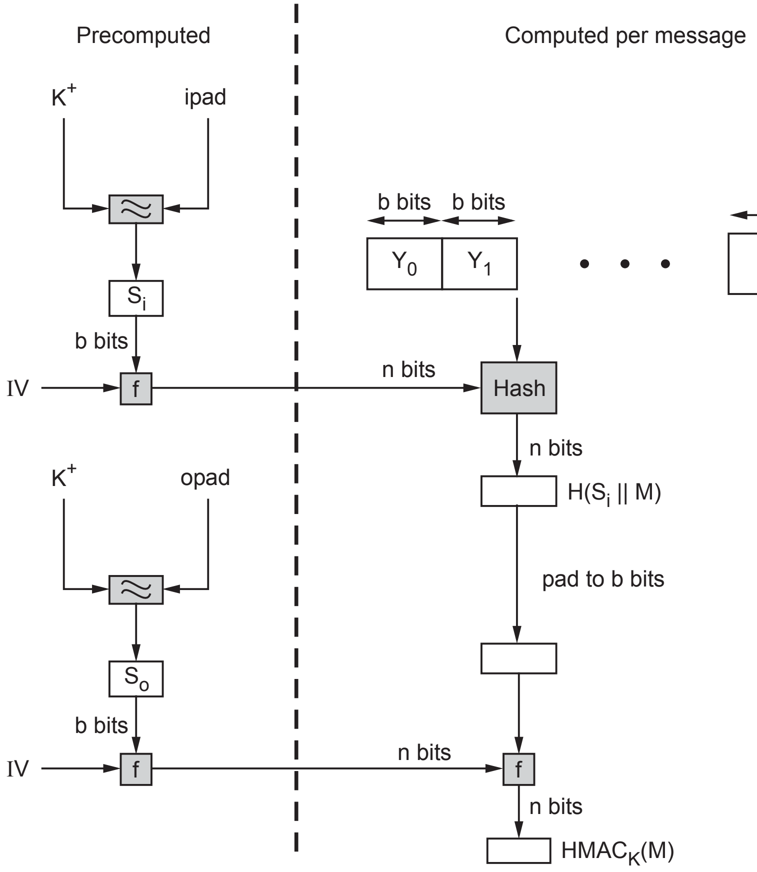


Fig. 6.2.2 Efficient implementation of HMAC

Where f(CV, block) is the compression function for the hash function.

HMAC security

- Know that the security of HMAC relates to that of the underlying hash algorithm.
- Attacking HMAC requires either :
  a) Brute-force attack on key used. This in order of 2n where n is the chaining variable bit-width.

  b) Birthday attack (but since keyed would need to observe a very large number of messages). Like MD5 this is in order of 2n/2 for a hash length of n.

- Choose hash function used based on speed versus security constraints.
- Note that HMAC is more secure than MD5 for birthday attack.
  a) In MD5 the attacker can choose any set of messages to find a collision (i.e. H(M) = H(M′).
  b) In HMAC since the attacker does not know K, he cannot generate messages offline. For a hash code of 128 bits, this requires 264 observed blocks (i.e. 264 ∗ 29 = 273 bits) generated using the same key. On a 1 Gbps line, this requires monitoring stream of messages with no change of the key for 250,000 years (quite infeasible !!).

University Question

1. *Write the algorithm for message authentication code (MACs) based on HASH functions.*

GTU : Summer-19, Marks 7

6.3 Macs Based on Block Ciphers

GTU : Summer-18

- Cipher-based Message Authentication Code (CMAC) is a block cipher-based message authentication code algorithm. CMAC mode of operation is used with AES and triple DES.

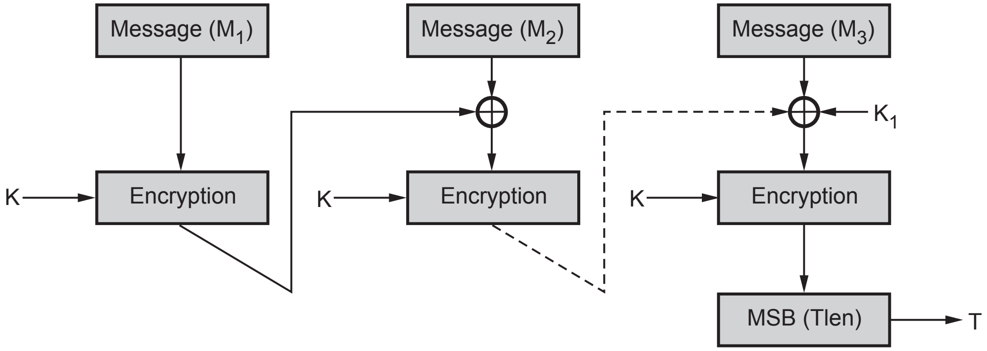


Fig. 6.3.1 Message length is integer multiple of block size

- The CMAC on a message is constructed by splitting it into blocks of size equal to the block size of the underlying cipher, for instance, 128 bits in the case of the AES, Cipher Block Chaining (CBC)-encrypting the message and retaining the result of the last block encryption as the computed MAC value.
- To avoid certain classes of attack, the last block is subjected, before ciphering, to an exclusive disjunction (XORing) with one of two possible "subkey" values, usually denoted as K1 or K2.

- The choice of which subkey to use is determined by whether the last message block contains padding or not. The subkey values can only be computed by parties knowing the cipher key in use.
- Fig. 6.3.1 shows calculation of CMAC.

$$
\begin{aligned}
C_1 &= E(K, M_1)\\
C_2 &= E(K,[M_2 \oplus C_1])\\
C_3 &= E(K,[M_3 \oplus C_2])\\
&\vdots\\
C_n &= E(K,[M_n \oplus C_{n-1} \oplus K_1])\\
T &= MSB_{tlen}(C_n)
\end{aligned}
$$

where

$$
\begin{aligned}
T &= \text{message authentication code}\\
\text{Tlen} &= \text{bit length of T}\\
\text{MSBs (X)} &= \text{the s left most bits of the bit string X}
\end{aligned}
$$

6.3.1 Data Authentication Algorithm

- The data authentication algorithm based on DES, has been one of the most widely used MAC for a number of years. The algorithm can be defined as using the cipher block chaining mode of operation of DES with an initialization vector of zero.
- Fig. 6.3.2 shows the data authentication algorithm.

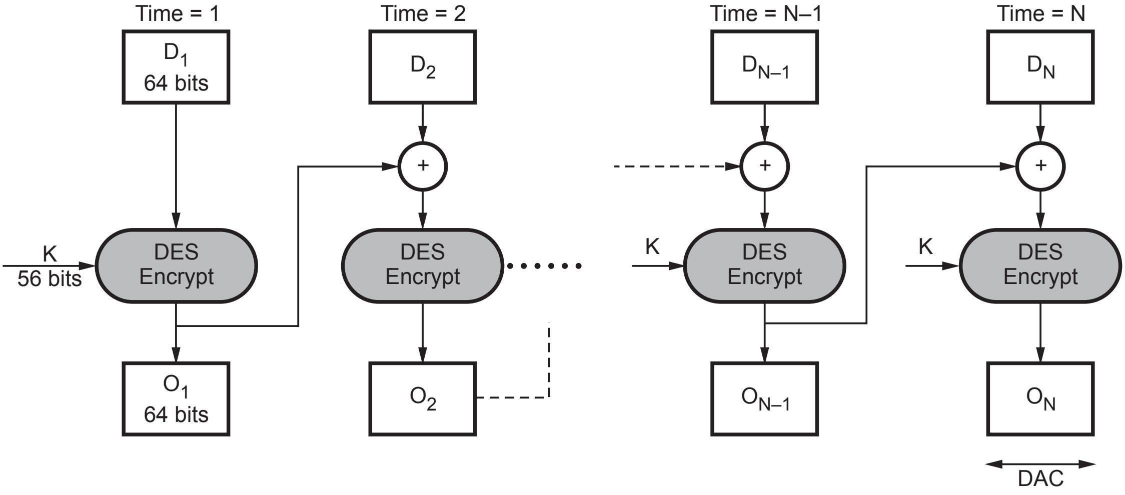


Fig. 6.3.2 Data authentication algorithm

- The algorithm can be defined as using the cipher block chaining mode of operation of DES. The data to be authenticated are grouped into contiguous 64-bit blocks : $D_1, D_2, D_3, \ldots\ldots\ldots\ldots, D_N$.
- Using the DES encryption algorithm (E) and a secret key (K), a data authentication code (DAC) is calculated as follows

$$\begin{aligned} O_1 &= E(K, D_1) \\ O_2 &= E(K, [D_2 \oplus O_1]) \\ O_3 &= E(K, [D_3 \oplus O_2]) \\ &: \\ &: \\ O_N &= E(K, [D_N \oplus O_{N-1}]) \end{aligned}$$

- The DAC consists of either the entire block O_N or the leftmost M bits of the block, with $16 \le M \le 64$.

University Question

1. *Explain MAC code generation using block cipher.* **GTU : Summer-18, Marks 3**

6.4 Short Questions and Answers

Q.1 Define sequence modification.

Ans. : Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.

Q.2 Define message authentication code.

Ans. : A function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

Q.3 What is bit length of the key?

Ans. : When an entire message is encrypted for confidentiality, using either symmetric or asymmetric encryption, the security of the scheme generally depends on the bit length of the key.

Q.4 What is data authentication algorithm?

Ans. : The data authentication algorithm based on DES, has been one of the most widely used MAC for a number of years. The algorithm can be defined as using the cipher block chaining mode of operation of DES with an initialization vector of zero.

Q.5 Define message authentication.

Ans. : Message authentication is a procedure to verify that received messages come form the alleged source and have not been altered.

Q.6 Give differences between hash function and message authentication codes.

GTU : Summer-17, Winter-17, Marks 3

Ans. : A MAC is an algorithm that requires the use of a secret key. A MAC takes a variable length message and a secret key as input and produces an authentication code. A recipient in possession of the secret key can generate an authentication code to verify the integrity of the message.

A hash function maps a variable-length message into a fixed length hash value or message digest. For message authentication, a secure hash function must be combined in some fashion with a secret key.

Q.7 What is the role of a compression function in a hash function ?

GTU : Winter-19, Marks 3

Ans. :

- A compression function takes a fixed length input and returns a shorter, fixed-lengths output.
- A typical hash function uses a compression function as a basic building block and involves repeated application of the compression function.

6.5 Multiple Choice Questions

Q.1 An authentication technique which makes use of a secret key to generate a small fixed-size block of data, known as a ————————

a Hash function b message authentication code

c MD5 d None of these

Q.2 MAC provides

a Message confidentiality

b Non-repudiation

c Message authentication and integrity

d All of these

Q.3 A(n) _____ can be used to preserve the integrity of a document or a message.

a message digest b message summary

c message confidentiality d none of the above

Q.4 When an entire message is encrypted for ————-, using either sym-metric or asymmetric encryption, the security of the scheme generally depends on the bit length of the key.

a Integrity b non-repudiation

c availability d confidentiality

Q.5 Source repudiation means

a Denial of receipt of message by destination

b Denial of transmission of message by destination

c Denial of transmission of message by source

d Denial of receipt of message by source

Q.6 Destination repudiation means

a Denial of receipt of message by destination

b Denial of transmission of message by destination

c Denial of transmission of message by source

d Denial of receipt of message by source

Answer Keys for Multiple Choice Questions

Q.1	b	**Q.2**	c	**Q.3**	a
Q.4	d	**Q.5**	c	**Q.6**	a

Notes

7 Digital Signature

Syllabus

Digital Signature, its properties, requirements and security, various digital signature schemes (Elgamal and Schnorr), NIST digital Signature algorithm.

Contents

7.1 Digital Signature

GTU : Winter-17, 19, Summer-18, 19

- 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 creator's private key.

Requirements

- Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other.
- In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature is analogous to the handwritten signature.
- It must have the following properties
  1. It must verify the author and the date and time of the signature.
  2. It must to authenticate the contents at the time of the signature.
  3. It must be verifiable by third parties, to resolve disputes.
- The digital signature function includes the authentication function. On the basis of these properties, we can formulate the following requirements for a digital signature.
- Must be a bit pattern depending on the message being signed.
- Signature must use some information unique to the sender to prevent forgery and denial.
- Computationally easy to produce a signature.
- Computationally easy to recognize and verify the signature.
- Computationally infeasible to forge a digital signature.
  a) either by constructing a new message for an existing digital signature.
  b) or by constructing a fraudulent digital signature for given message.
- Practical to retain a copy of the digital signature in storage.

Two general schemes for digital signatures

1) Direct
2) Arbitrated

7.1.1 Arbitrated Digital Signatures

- Every signed message from A to B goes to an arbiter BB (Big Brother) that everybody trusts.
- BB checks the signature and the timestamp, origin, content, etc.
- BB dates the message and sends it to B with an indication that it has been verified and it is legitimate.

  e.g. Every user shares a secret key with the arbiter
- A sends to BB in an encrypted form the plaintext P together with B's id, a timestamp and a random number RA.
- BB decrypts the message and thus makes sure it comes from A; it also checks the timestamp to protect against replays.
- BB then sends B the message P, A's id, the timestamp and the random number RA; he also sends a message encrypted with his own private key (that nobody knows) containing A's id, timestamp t and the plaintext P (or a hash).
- B cannot check the signature but trusts it because it comes from BB-he knows that because the entire communication was encrypted with KB.
- B will not accept the messages or messages containing the same RA to protect against replay.
- In case of dispute, B will show the signature he got from BB (only B may have produced it) and BB will decrypt it.

7.1.2 Direct Digital Signature

- This involves only the communicating parties and it is based on public keys.
- The sender knows the public key of the receiver.
- Digital signature : Encrypt the entire message (or just a hash code of the message) with the sender's private key.
- If confidentiality is required : Apply the receiver's public key or encrypt using a shared secret key.
- In case of a dispute the receiver B will produce the plaintext P and the signature E(KRA, P) - the judge will apply KUA and decrypt P and check the match : B does not know KRA and cannot have produced the signature himself.

Weaknesses

- The scheme only works as long as KRA remains secret : If it is disclosed (or A discloses it herself), then the argument of the judge does not hold : anybody can produce the signature.

- **Attack :** To deny the signature right after signing, simply claim that the private key has been lost-similar to claims of credit card misuse.

  i.e. If A changes her public-private keys (she can do that often) the judge will apply the wrong public key to check the signature.

- **Attack :** To deny the signature change your public-private key pair-this should not work if a PKI is used because they may keep trace of old public keys.

  i.e. A should protect her private key even after she changes the key.

- **Attack :** Eve could get hold of an old private key and sign a document with an old timestamp.

7.1.3 Digital Signature Standard

- The Digital Signature Standard (DSS) makes use of the Secure Hash Algorithm (SHA) and presents a new digital signature technique, the Digital Signature Algorithm (DSA). DSS cannot be used for encryption or key exchange. Fig. 7.1.1 shows the DSS approach.
- It uses a hash function. The hash code is provided as input to a signature function along with a random number K generated for this particular signature.
- The signature function also depends on the sender's private key (PR_a) and a set of parameters known to a group of communicating principles.
- The result is a signature consisting of two components, labeled s and r.
- At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function.

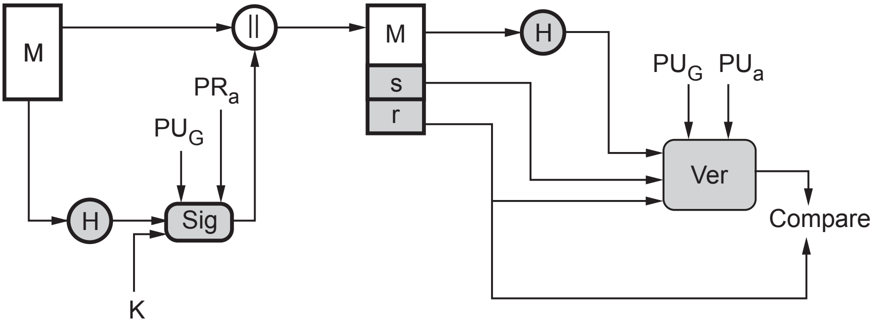


Fig. 7.1.1 DSS approach

- Fig. 7.1.2 shows the RSA approach. In the RSA approach, the message to be signed is input to a hash function that producs a secure hash code of fixed length. This hash code is then encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted.

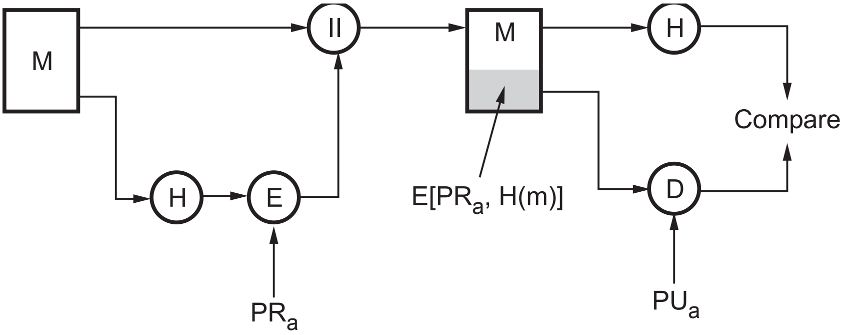


Fig. 7.1.2 RSA approach

- The recipient takes the message and produces a hash code. The recipient also decrypts the signature using the sender's public key. If the calculated hash code matches the decrypted signature, the signature is accepted as valid.

7.1.4 Digital Signature Algorithm

- There are three parameters that are public and can be common to a group of users. **Prime number q** is chosen and it is **160-bit**. A **prime number p** is selected with a length between **512** and **1024 bits** such that q divides (P – 1).
- g is chosen to be of the form $h^{(P - 1)/q}$ mod p where h is an integer between 1 and (P – 1)
- With these number, user selects a private key and generate a public key. The private key x must be a number from 1 to (q – 1) and should be chosen randomly or pseudorandomly.
- The public key is calculated from the private key as $y = g^x$ mod p.
- To create a signature, a user calculates two quantities, **rands**, that are functions of
  - i) Public key components (p, q, g)
  - ii) User's private key (x)
  - iii) Hash code of the message H(M)
  - iv) An additional integer (K)
- **At the receiving end**, verification is performed. The receiver generates a quantity V that is a function of the public key components, the sender's public key and the hash code of the incoming message. If this quantity matches the r components of the signature, then the signature is validated.
- Fig. 7.1.3 shows the functions of signing and verifying.

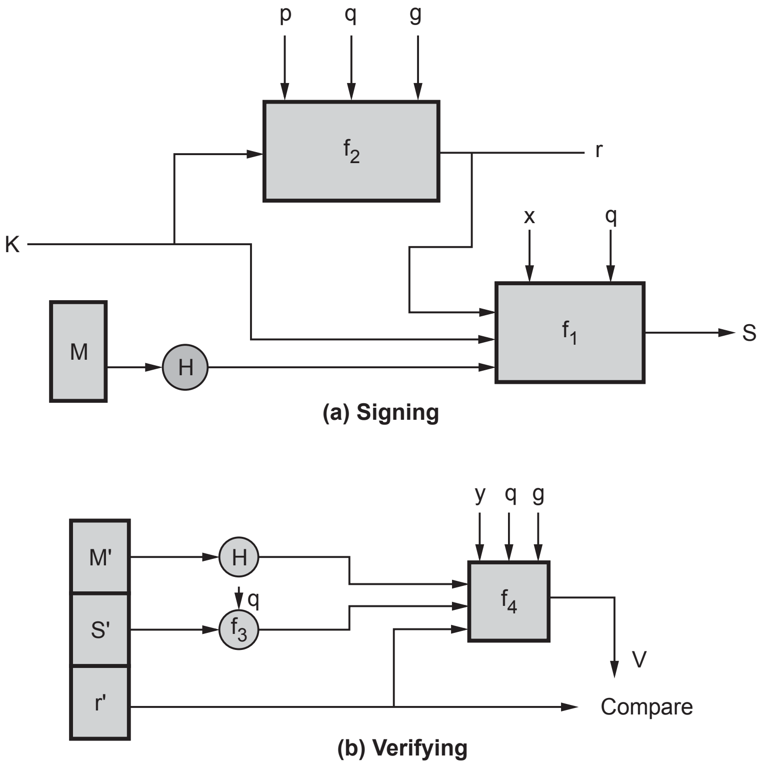


Fig. 7.1.3 Signing and verifying

University Questions

1. *Elaborate any one approach to Digital Signatures.* **GTU : Winter-17, Marks 7**
2. *Explain DSA (Digital Signature Algorithm).* **GTU : Summer-18, Marks 3**
3. *What is the principle of digital signature algorithm (DSA)? How a user can create a signature using DSA? Explain the signing and verifying function in DSA.* **GTU : Summer-19, Marks 7**
4. *Draw generic model of digital signature process.* **GTU : Winter-19, Marks 3**

7.2 Digital Certificate

- A data structure that securely binds an individual or entity to a public key used in cryptographic operations such as digital signatures or asymmetric encryption.
- To obtain digital certificate an organization must apply to a certification authority which is responsible for validating and ensuring the authenticity of requesting organization. The certificate will identify the name of the organization, a serial number, the validity date and the organization's public key where encryption to / from that organization is required.

- In addition, the digital certificate will also contain the digital signature of the certification authority to allow any recipient to confirm the authenticity of the digital certificate.
- A digital certificate is an ID that is carried with a file. To validate a signature, a certifying authority validates information about the software developers and then issues them digital certificates. The digital certificate contains information about the person to whom the certificate was issued, as well as information about the certifying authority that issued it. When a digital certificate is used to sign programs, ActiveX controls, and documents, this ID is stored with the signed item in a secure and verifiable form so that it can be displayed to a user to establish a trust relationship.
- A digital certificate allows unique identification of an entity; it is essentially an electronic ID card, issued by a trusted third party. Digital certificates form part of the ISO authentication framework, also known as the X.509 protocol. This framework provides for authentication across networks.
- A digital certificate serves two purposes : It establishes the owner's identity, and it makes the owner's public key available. A digital certificate is issued by a Certification Authority (CA). It is issued for only a limited time and when its expiry date has passed, it must be replaced.
- A digital certificate consists of :
  1. The public key of the person being certified
  2. The name and address of the person being certified, also known as the Distinguished Name (DN)
  3. The digital signature of the CA
  4. The issue date
  5. The expiry date
- The Distinguished Name is the name and address of a person or organization. You enter your Distinguished Name as part of requesting a certificate. The digitally-signed certificate includes not only your own Distinguished Name, but the Distinguished Name of the CA, which allows verification of the CA.
- To communicate securely, the receiver in a transmission must trust the CA that issued the certificate that the sender is using. This means that when a sender signs a message, the receiver must have the corresponding CA's signer certificate and public key designated as a trusted root key. For example, your web browser has a default list of signer certificates for trusted CAs. If you want to trust certificates from another CA, you must receive a certificate from that CA and designate it as a trusted root key.

- If you send your digital certificate containing your public key to someone else, what keeps that person from misusing your digital certificate and posing as you? The answer is: your private key.
- A digital certificate alone is not proof of anyone's identity. The digital certificate allows verification only of the owner's identity, by providing the public key needed to check the owner's digital signature. Therefore, the digital certificate owner must protect the private key that belongs with the public key in the digital certificate. If the private key were stolen, anyone could pose as the legitimate owner of the digital certificate.

7.3 ElGamal

- The ElGamal algorithm provides an alternative to the RSA for public key encryption.
  1. Security of the RSA depends on the difficult of factoring large integers.
  2. Security of the ElGamal algorithm depends on the difficulty of computing discrete logs in a large prime modulus.
- ElGamal has the disadvantage that the ciphertext is twice as long as the plaintext.
- It has the advantages the same plaintext gives a different ciphertext each time it is encrypted.
- Like RSA, the ElGamal system is a public key algorithm so it has one set of key numbers that are published and another secret number that is used for deciphering.
  1. The keys are generated by selecting a large prime number p. It is recommended that $p - 1$ be divisible by another large prime.
  2. Compute a generator number g and select a random integer "a" less than $p - 1$.
  3. With these numbers compute $b = g^{a} \pmod p$.
  4. The public key consists of the three number (p, g, b) and the secret key is the number a.
  5. To find "a" given the public key, an attacker must be able to solve the discrete logarithm problem.

Encryption :

- If Bob wants to send a message to Alice he begins by looking up her public key (p, g, b) and representing the message as an integer m in the range 0 to $p - 1$.
- He then selects a random key, k that is less than $p - 1$.

- Using these numbers, Bob computes two numbers :

$$c_1 = g^k \quad \text{and} \quad c_2 = mb^k$$

- He sends (c_1, c_2) to Alice.

Decryption :

- When Alice receives the cipher-text, she will recover the plaintext using her secret key, "a" to compute :

$$m = c_2 c_1^{-a} \bmod p$$

- This works because :

$$c_2 c_1^{-a} = mb^k (g^k)^{-a}\; mb^k\; (g^a)^k\; (g^k)^{-a}$$

$$= mg^{ak}\; g^{-ak} = m \pmod p$$

- Bob should choose a different random integer k for each message he sends to Alice. If M is a longer message, so it is divided into blocks, he should choose a different k for each block.
- Say he encrypts two messagers (or blocks) M_1 and M_2, using the same k, producing cipher-texts.
- Eve intercepts both cipher-text messages and discovers one plaintext message M_1, she can compute the other plaintext message M_2.

Example : Alice selected her initial prime number p = 11, found the primitive element g = 7 and selected her random secret key a = 2, then her public key is :

$$b = 7^2 \bmod 11 = 5$$

- She would publish her public key : (11, 7, 5)
- Bob wants to send the letter "a" to Alice
  1. He first breaks it up into a set of numbers where each number is less than 11 (the value of p).
  2. Since the ASCII representation of "a" is 01100001, he might break it up into four messages (01 10 00 01) or in decimal (1, 2, 0, 1).
  3. Next, he would select a random number k = 3 and then compute and send to Alice :

m	c_1	c_2
1	$7^3 \bmod 11 = 2$	$1 \times 5^3 \bmod 11 = 4$
2	$7^3 \bmod 11 = 2$	$2 \times 5^3 \bmod 11 = 8$

0	7^3 mod 11 = 2	0×5^3 mod 11 = 0
1	7^3 mod 11 = 2	1×5^3 mod 11 = 4

- The cipher-text is ((2, 4), (2, 8), (2, 0), (2, 4)).

Deciphering a Message

- When Alice receives this message from Bob, she uses her secrete key a = 2 as follows :

  (2, 4) : m = 4(2) – 2 = 4(4) – 1 = 12 mod 11 = 1 (4 and 3 are inverse mod 11)

  (2, 8) : m = 8(2) – 2 = 8(4) – 1 = 24 mod 11 = 2

  (2, 0) : m = 0(2) – 2 = 0(4) – 1 = 0 mod 11 = 0

  (2, 4) : m = 1

  Alice reassembles the message into the letter "a".

University Questions

1. *Describe Elgamal digital signature.* **GTU : Summer-19, Marks 7**
2. *Explain Elgamal digital signature scheme.* **GTU : Winter-19, Marks 4**

7.4 Schnorr Signature

- The Schnorr signature scheme is derived from Schnorr's identification protocol using the Fiat-Shamir heuristic. The resulting digital signature scheme is related to the Digital Signature Standard (DSS). As in DSS, the system works in a subgroup of the group Z^*_p for some prime number p. The resulting signatures have the same length as DSS signatures
- Its security has been analyzed in the Random Oracle Model (ROM) under the Discrete Logarithm (DL) assumption.
- The Schnorr scheme minimizes the message dependent amount of computation required to generate a signature. The main work for signature generation does not depend on the message and can be done during the idle time of the processor. The message dependent part of the signature generation requires multiplying a 2n-bit integer with an *n-bit* integer.
- The first part of this scheme is the generation of a private/public key pair, which consists of the following steps :
  1. Choose primes p and q, such that q is a prime factor of p - 1.
  2. Choose an integer a such that a^q = 1 mod p. The values a, p, and q comprise a global public key that can be common to a group of users.
  3. Choose a random integer s with 0 < s < q. This is the user's private key.

4. Calculate $v = a^{-s} \bmod p$. This is the user's public key.

- A user with public key s and private key v generates a signature as follows:
  1. Choose a random integer r with $0 < r < q$ and compute $x = a^{r} \bmod p$. This is independent of any message M, hence can be pre-computed.
  2. Concatenate message with x and hash result to compute: $e = H(M \,||\, x)$
  3. Compute $y = (r + se) \bmod q$. The signature consists of the pair (e, y).
  4. Any other user can verify the signature as follows :
  5. Compute $x' = a^{y}v^{e} \bmod p$.
  6. Verify that $e = H(M \,||\, x')$.

Example 7.4.1 *Using the Schnorr scheme, let q = 83, p = 997, and d = 23. Find values for e_1 and e_2. Choose r = 11, if M = 400 and h(400) = 100. Find value of S_1, S_2*

Solution : Given data : q = 83, p = 997, and d = 23.

Take $e_0 = 7$

Then
$$e_1 = e_0^{(p-1)/q} \bmod p$$
$$e_1 = (7^{(997-1)/83}) \bmod 997$$
$$= (7^{12}) \bmod 997$$
$$e_1 = 9$$
$$e_2 = (e_1^{d}) \bmod p$$
$$= (9^{23}) \bmod 997$$
$$= 521$$

Calculate S_1 and S_2 in mod q.

So
$$h(40067) = 81$$
$$S_1 = h\,(M \mid e_1^{r} \bmod p)$$
$$= h\,(400 \mid 9^{11} \bmod p)$$
$$= h\,(400 \mid 67)$$
$$= h\,(40067)$$
$$= 81$$
$$S^2 = r + ds_1 \bmod q$$
$$= 11 + 23 \times 81 \bmod 83$$
$$= 48$$

University Questions

1. *Explain digital signature schemes Elgamal and Schnorr.* **GTU : Winter-18, Marks 7**
2. *Explain Schnorr digital signature scheme.* **GTU : Winter-19, Marks 4**

7.5 NIST Digital Signature Algorithm

- The National Institute of Standards and Technology (NIST) requests comments on Federal Information Processing Standard. The NIST has announced proposed changes to a standard that specifies how to implement digital signatures, which can be used to ensure the integrity of electronic documents, such as wills and contracts, as well as the identity of the signer.
- These proposed changes to the Federal Information Processing Standard (FIPS) 186-3, known as the Digital Signature Standard, were posted for public comment on April 10, 2012. First published in 1994 and revised several times since then, the standard provides a means of guaranteeing authenticity in the digital world by means of operations based on complex math that are all but impossible to "forge". Updates to the standard are still necessary as technology changes.
- A Digital Signature is a function provided by Public Key Infrastructure (PKI). The process entails transforming a message or data and some secret information held by the sender into a tag called a signature. It provides proof of the source and verification of the integrity of the data.
- The sender generates a digital signature using his/her private key. The recipient verifies the sender's identity using the sender's public key.
- The purpose of a digital signature is to provide a means for an entity to bind its identity to data, and to detect unauthorized modifications to data.
- There are three algorithms suitable for digital signature generation and verification:
  1. Digital Signature Algorithm (DSA)
  2. Rivest-Shamir-Adleman, a reversible Digital Signature Algorithm (RSA)
  3. Elliptic Curve Digital Signature Algorithm (ECDSA)

Benefits:

1. Digital signatures eliminate the need for transmitting passwords for authentication, which reduces the threat of their compromise
2. Using a private key to generate digital signatures for authentication prevents an attacker from using the same information to masquerade as another entity and authenticate repeatedly.

3. Digital signatures provide security for electronic mail, Electronic Funds Transfer (EFT), Electronic Data Interchange (EDI), software distribution, data storage, and other applications that require data integrity assurance and data origin authentication.

7.6 Short Questions and Answers

Q.1 What is the use of digital signature ?

Ans. : Data appended to, or 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.

Q.2 What is the utility of a detached signature ?

Ans. : 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 signature can be used when more than one party must sign a document, such as legal contract.

Q.3 What is digital signature ?

Ans. : Digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.

Q.4 What is one-way property ?

Ans. : A function that maps an arbitrary length message to a fixed length message digest is a one-way hash function if it is a one-way function.

Q.5 What are the two approaches of digital signature ?

Ans. : Two approaches of digital signature are RSA approach and DSS approaches.

Q.6 List any two requirements of digital signature.

Ans. : Requirements :

(a) It must be relatively easy to produce the digital signature.

(b) It must be relatively easy to recognize and verify the digital signature.

Q.7 How confidentially is provided in direct digital signature ?

Ans. : Confidentiality can be provided by encrypting the entire message plus signature with a shared secret key (symmetric encryption).

Q.8 What is time stamped signatures ?

Ans. : Sometimes a signed document needs to be time stamped to prevent it from being replayed by an adversary. This is called time-stamped digital signature scheme.

7.6 Multiple Choice Questions

Q.1 A digital signature is an ————- mechanism that enables the creator of a message to attach a code that acts as a signature.

a authorization

b authentication

c both authorization & authentication

d none of these

Q.2 Digital signature standard cannot be used for encryption ——————-.

a encryption

b decryption

c authorization

d all of them

Q.3 ————- helps in ensuring non-fraudulent transactions on the web.

a Certificate authority

b Digital authority

c Dual authority

d Digital signature.

Q.4 Digital signature envelope is decrypted by using ——————-.

a merchant private key

b payment public key

c merchant's public key.

d payment's private key.

Q.5 The recipient verifies the sender's identity using the sender's ————- key.

a private key

b public key

c both key

d none of these

Q.6 For a digital signature, there is a —————— relationship between a signature and a message.

a One to many

b many to one

c many to many

d one to one

Answer Keys for Multiple Choice Questions

Q.1	b	**Q.2**	a	**Q.3**	a
Q.4	d	**Q.5**	b	**Q.6**	d

❑❑❑

8 Key Management and Distribution

Syllabus

Key management and distribution, symmetric key distribution using symmetric and asymmetric encryptions, distribution of public keys, X.509 certificates, Public key infrastructure.

Contents

8.1 Key Management

GTU : Summer-17,18,19, Winter-17,18

- The purpose of public key cryptography is
  1. The distribution of public keys.
  2. The use of public key encryption to distribute secret keys.

8.1.1 Distribution of Public Keys

- Different methods have been proposed for the distribution of public keys. There are
  1. Public announcement.
  2. Publicly available directory.
  3. Public key authority.
  4. Public key certificates.

1. Public announcement

- In public key algorithm, any participant can **send** his or her public key to any other participant or **broadcast** the key to the community at large.
- Fig. 8.1.1 shows the public key distribution.

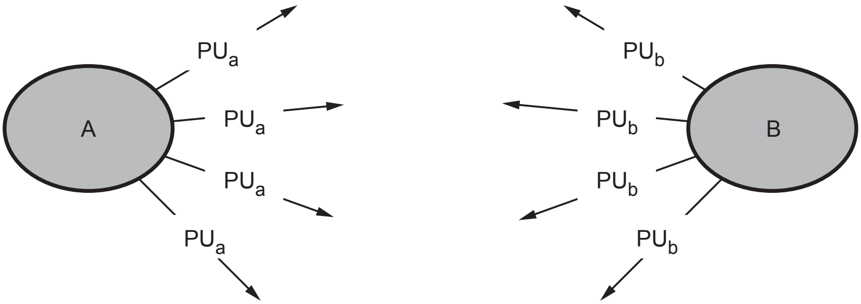


Fig. 8.1.1 Public key distribution

- Because of the growing popularity of PGP, 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 newgroups and Internet mailing lists.
- The disadvantage is that, 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.

2. Public available directory

- 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.
- Fig. 8.1.2 shows public key publication.

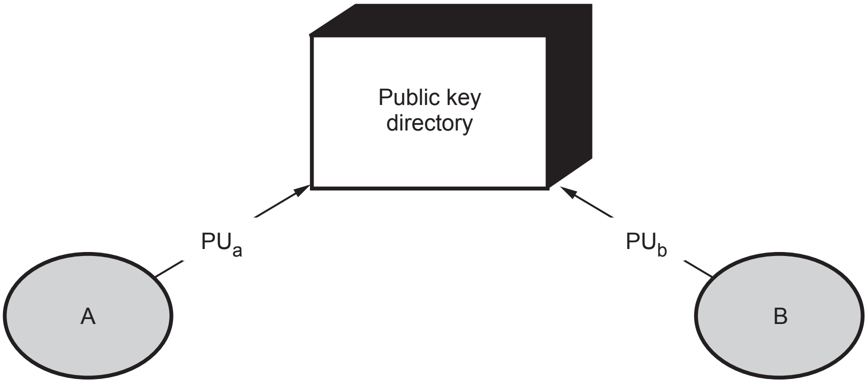


Fig. 8.1.2 Public key publication

- 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.
  4. Participants could also access the directory electronically.

3. Public key authority

- Fig. 8.1.3 shows public key distribution scenario

(Refer Fig. 8.1.3 on next page.).

- Following steps occur in public key distribution.
  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$_{auth}$. The message also contains B's public key (PU$_b$), original request and timestamp.

3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) 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. Public keys have been securely delivered to A and B and they may begin their protected exchange.
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).
7. A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.

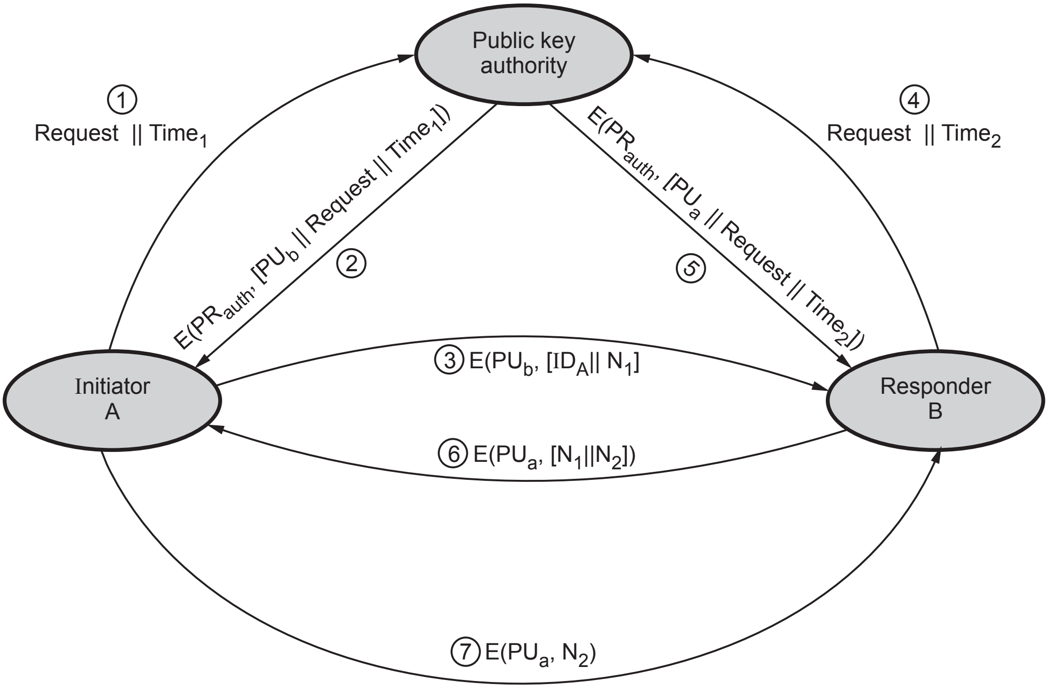

Public key authority
Initiator A
Responder B
① Request || Time$_1$
② E(PR$_{auth}$, [PU$_b$ || Request || Time$_1$])
③ E(PU$_b$, [ID$_A$|| N$_1$]
④ Request || Time$_2$
⑤ E(PR$_{auth}$, [PU$_a$ || Request || Time$_2$])
⑥ E(PU$_a$, [N$_1$||N$_2$])
⑦ E(PU$_a$, N$_2$)

Fig. 8.1.3 Public key distribution scenario

Drawback

Public key authority could be somewhat of a bottleneck in the system. The directory of name and public keys maintained by the authority is vulnerable to tampering.

4. Public key certificates

- Certificates can be used by participants to exchange keys without contacting a public key authority. Certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party.

- The third party is a certificate authority, such as government agency or a financial institution, that is trusted by the user community.
- A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate.
- Requirements on this scheme :
  1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
  2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
  3. Only the certificate authority can create and update certificates.
  4. Any participant can verify the currency of the certificate.
- A certificate scheme is illustrated in Fig. 8.1.4. Each participant applies to the certificate authority, supplying a public key and requesting a certificate.

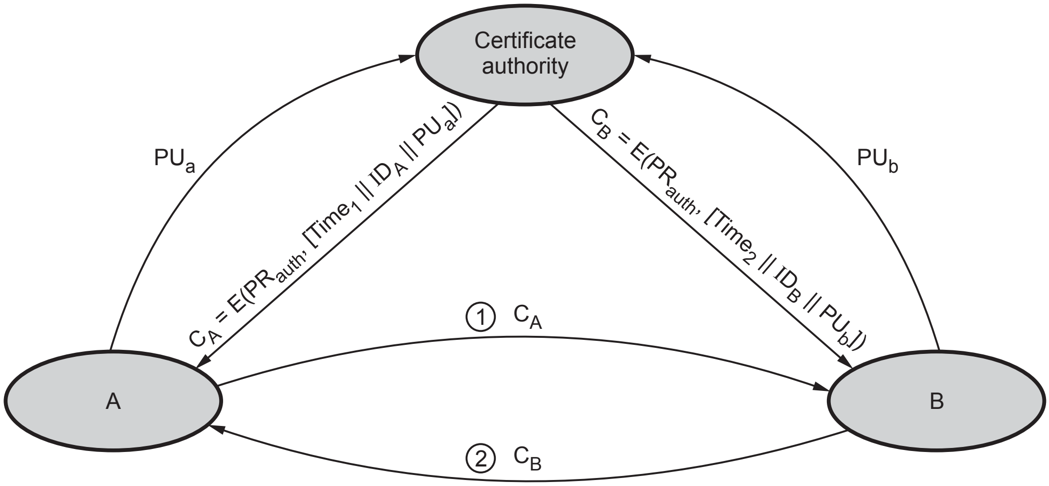

Certificate authority
PU_a
PU_b
$C_A = E(PR_{auth}, [Time_1 || ID_A || PU_a])$
$C_B = E(PR_{auth}, [Time_2 || ID_B || PU_b])$
① C_A
② C_B
A
B

Fig. 8.1.4 Exchange of public key certificates

- For participant A, the authority provides a certificate of the form

$$C_A = E\ (PR_{auth,}\ [T\ ||\ ID_A\ ||\ PU_a])$$

- where PR_{auth} is the private key used by the authority and T is a timestamp.

8.1.2 Distribution of Secret Keys using Public Key Cryptography

- Public key encryption provides for the distribution of secret key to be used for conventional encryption.

Simple secret key distribution

If user A wishes to communicate with user B, the following procedure is employed :

1. User A generates a public/private key pair {PU_a, PR_a} and transmits a message to user B consisting of PU_a and an identifier of A, ID_A.
2. User B generates a secret key (K_s) and transmits it to user A, encrypted with A's public key.
3. User A computes D(PR_a, E(PU_a, K_s)) to recover the secret key. Because only A can decrypt the message, only user A and user B know the identity of K_s.
4. User A discards PU_a and PR_a and user B discards PU_a.
5. Fig. 8.1.5 shows use of public key encryption.

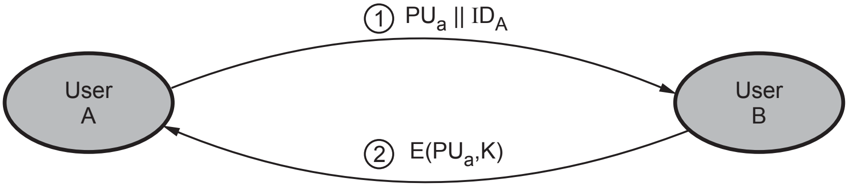
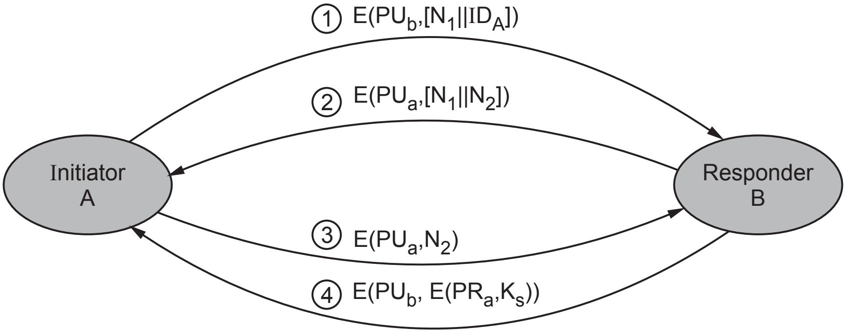


Fig. 8.1.5 Use of public key encryption

- User A and B can now securely communicate using conventional encryption and the session key K_s. At the completion of the exchange, both user A and B discard K_s.
- The protocol discussed above is insecure against an adversary who can intercept messages and then either relay the intercepted message or substitute another message. Such an attack is known as a **man in middle attack.**

Secret key distribution with confidentiality and authentication

- Fig. 8.1.6 shows the public key distribution of secret keys.



① E(PU_b,[N_1||ID_A])
② E(PU_a,[N_1||N_2])
Initiator A
Responder B
③ E(PU_a,N_2)
④ E(PU_b, E(PR_a,K_s))

Fig. 8.1.6 Public key distribution of secret keys

- It provides protection against both passive and active attacks.

1. A uses B's public key to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.
2. 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 $B(N_2)$.
3. A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.
4. A selects a secret key K_s and sends $M = E(PU_b, E(PR_a, K_s))$ to B.
5. B computes $D(PU_a, D(PR_b, M))$ to recover the secret key.

8.1.3 Key Distribution and Certification

- Management and handling of the pieces of secret information is generally referred to as **key management.**
- Activities of key management includes selection, exchange, storage, certification, revocation, changing, expiration and transmission of the key.
- Key management is the set of processes and mechanisms which support key establishment and maintenance of ongoing keying relationship between parties, including replacing older key with new keys.
- Two major issues in key management are :
  1. Key life time
  2. Key exposure

Key life time - limit of use which can be measured as a duration of time.

Issue related to key :

1. Users must be able to obtain securely a key pair suited to their efficiency and security needs.
2. Keys need to be valid only until a specified expiration date.
3. The expiration date must be chosen properly and publicized securely.
4. User must be able to store their private keys securely.
5. Certificates must be unforgettable, obtainable in a secure manner.

1. Public Key Infrastructure

- Public Key Infrastructure (PKI) is a well-known technology that can be used to establish identities, encrypt information and digitally sign documents.
- PKI identifies and manages relationships of parties in an electronic exchange, serving a wide array of security needs including access control, confidentiality, integrity, authentication and non-repudiation.

- PKI also uses unique Digital Certificates (DC) to secure eCommerce, email, data exchange and Virtual Private Networks (VPN) and intranets and is also used to verify the identity and privileges of each user.
- The Certificate Authority (CA) provides a full life-cycle management of public keys and certificates, including issuance, authentication, storage, retrieval, backup, recovery, updating and revocation to the PKI.
- All users of PKI must have a registered identity, which is stored in a digital certificate issued by a CA.
- Remote users and sites using public private keys and public key certificates can authenticate each other with a high degree of confidence.
- Authentication is dependent on three conditions :
    1. It must be established that each party have a private key that has not been stolen or copied from the owner.
    2. The certificate must be issued to the owner in accord with the stated policy of the certificate issuer.
    3. The policies of the certificate issuer must be satisfactory to the parties so as to verify identity.

Benefits of PKI

1. Confidential communication : Only intended recipients can read files.
2. Data integrity : Guarantees files are unaltered during transmission.
3. Authentication : Ensures that parties involved are who they claim to be.
4. Non-repudiation : Prevents individuals from denying.

Limitation of PKI

The problems encountered deploying a PKI can be categorized as follows :

1. Public key infrastructure is new
2. Lack of standards
3. Shortage of trained personnel
4. Public key infrastructure is mostly about policies.

2. Certificate

- **Certificates** are digital documents that are used for secure authentication of communicating parties.
- A certificate binds identity information about an entity to the entity's public key for a certain validity period.

- A certificate is digitally signed by a Trusted Third Party (TTP) who has verified that the key pair actually belongs to the entity.
- Certificates can be thought of as analogous to passports that guarantee the identity of their bearers.
- **Authorities :** The trusted party who issues certificates to the identified end entities is called a **Certification Authority (CA).**
- Certification authorities can be thought of as being analogous to governments issuing passports for their citizens.
- A certification authority can be managed by an external certification service provider or the CA can belong to the same organization as the end entities.
- CAs can also issue certificates to other (sub) CAs. This leads to a tree-like **certification hierarchy.**
- The highest trusted CA in the tree is called a root CA.
- In large organizations, it may be appropriate to delegate the responsibility for issuing certificates to several different certificate authorities.

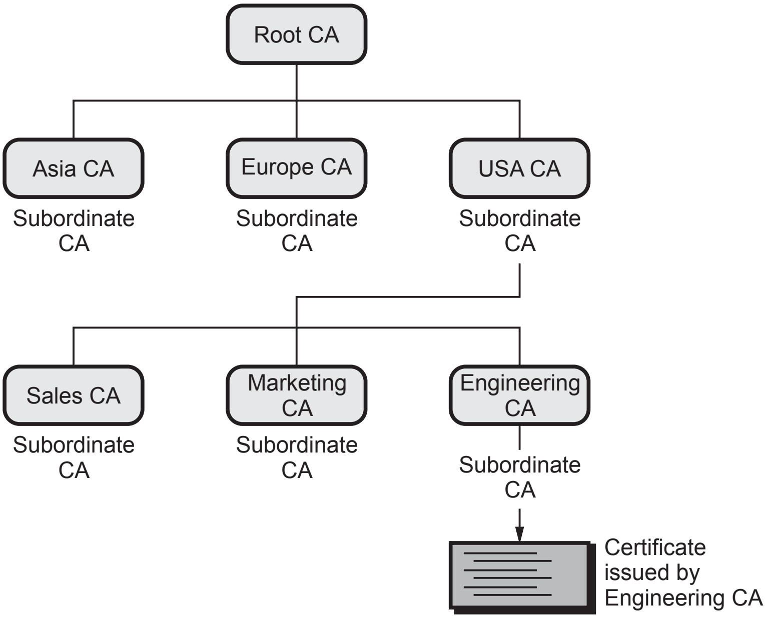


Fig. 8.1.7 Hierarchy of CA

- For example, the number of certificates required may be too large for a single CA to maintain; different organizational units may have different policy requirements; or it may be important for a CA to be physically located in the same geographic area as the people to whom it is issuing certificates.

- The X.509 standard includes a model for setting up a hierarchy of the Certification Authority.
- Fig. 8.1.7 shows the hierarchy of certificate authorities.
- In the Fig. 8.1.7, the root CA is at the top of the hierarchy. The root CA's certificate is a self-signed certificate : that is, the certificate is digitally signed by the same entity.
- The CAs, that are directly subordinate to the root CA, have CA certificates signed by the root CA. CAs under the subordinate CAs in the hierarchy have their CA certificates signed by the higher-level subordinate CAs.
- Organizations have a great deal of flexibility in terms of the way they set up their CA hierarchies.
- **Certificate chains :** Certificate chain is series of certificates issued by successive CAs.
- In some cases, a CA can delegate the actual identification of end entities as well as some other administrative tasks to a separate entity, the **Registration Authority (RA).**

Verifying certificates

- When authentication is required, the entity presents a signatures it has generated from authentication data using its private key, and a certificate corresponding to that key.
- The receiving entity can verify the signature with the public key of the sender contained in the certificate.
- Next the receiving entity must verify the certificate itself by checking the validity time of the certificate and the signature of the CA in the certificate.
- If the CA is a sub CA, the receiving entity must also verify the signatures of all higher-level CAs up to the root CA.
- The list of certificates needed for verification is called a **certification path.**
- If all signatures are valid and the receiving entity trusts the root CA, the first entity will be authenticated successfully.
- If a private key of an end entity is compromised or the right to authenticate with a certificate is lost before its natural expiration date, the CA must revoke the certificate and inform all PKI users about this.
- The CA will periodically publish a **certificate revocation list (CRL)**.
- The CRL is a list identifying the revoked certificates and it is signed by the CA.

- The end entities should check the latest CRL whenever they are verifying a validity of a certificate.

3. Key Length and Encryption Strength

- The strength of encryption depends on both the cipher used and the length of the key.
- Encryption strength is often described in terms of the size of the keys used to perform the encryption : in general, longer keys provide stronger encryption.
- Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher.
- Roughly speaking, 128-bit RC4 encryption is 3×10^{26} times stronger than 40-bit RC4 encryption.
- Different ciphers may require different key lengths to achieve the same level of encryption strength.
- The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based.
- Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values.
- Thus a 128-bit key for use with a symmetric key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher.

8.1.4 Key Distribution

- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others. **Key distribution** refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key.
- For two parties A and B, key distribution can be achieved in a number of ways, as follows.
  1. User A can select a key and physically deliver it to user B.
  2. A third party can select the key and physically deliver it to user A and user B.
  3. If user A and user B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
  4. If user A and user B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to user A and user B.

- For manual delivery of key, **options 1 and 2** are used. These options are suitable for **link encryption.**
- Option 3 is suitable for link encryption or end-to-end encryption.
- For end-to-end encryption, some variation on option 4 has been widely adopted.
- The use of a key distribution center is based on the use of a hierarchy of keys. Minimum two levels of keys are used. Fig. 8.1.8 shows the use of a key hierarchy.
- Communication between end systems is encrypted using a temporary key, often referred to as a **session key**. The **session key** is used for the duration of a logical connection, such as a frame relay connection, or transport connection and then discarded.
- Session keys are transmitted in encrypted form, using a **master key** that is shared by the key distribution center and an end system or user. For each end user, there is a unique master key that it shares with the key distribution center.

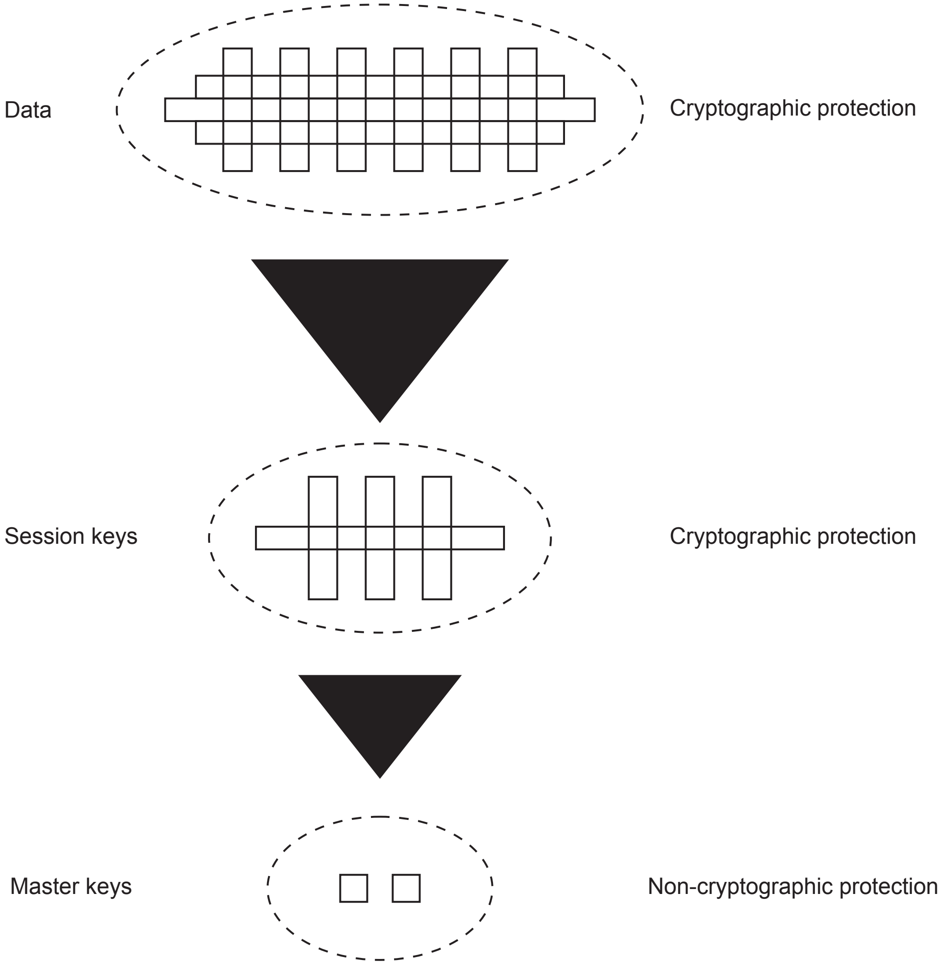


Fig. 8.1.8 Use of a key hierarchy

A key distribution scenario

- User A wishes to establish a logical connection with user B and requires a one time session key to protect the data transmitted over the connection. User A has a master key (K_a), known only to itself and the KDC. User B shares the master key K_b with the KDC.

The following steps occur :

1. A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier (N_1) for this transaction.
2. KDC responds with a message encrypted using K_a.
3. A stores the session key for use in the upcoming session and forward to B the information that originated at the KDC for B :
4. User B sends a nonce N_2 to A.

- Fig. 8.1.9 shows the key distribution scenario.

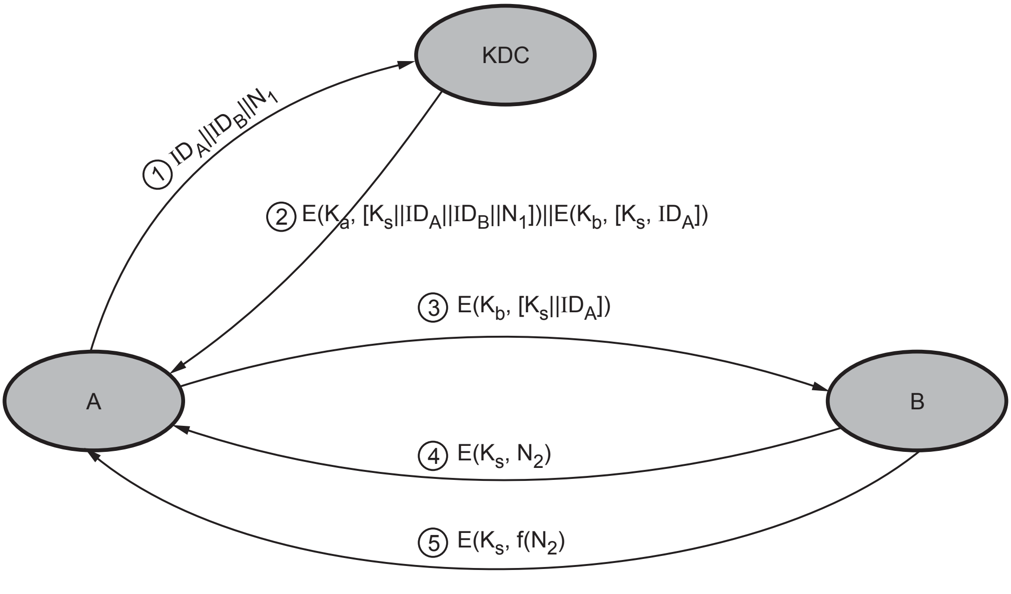


Fig. 8.1.9 Key distribution scenario

- Steps 1, 2 are used for key distribution and steps 3, 4, 5 for authentication.

Session key lifetime

1. For connection-oriented protocol

- Use the same session key for the length of time that the connection is open. Use new session key for each new session.
- For long lifetime, change the session key periodically.

2. For connectionless protocol

- The most secure approach is to use a new session key for each exchange. For connectionless protocol, such as a transaction - oriented protocol, there is no explicit connection initiation or termination.

Transparent key control scheme

- Fig. 8.1.10 shows automatic key distribution for connection - oriented protocol.

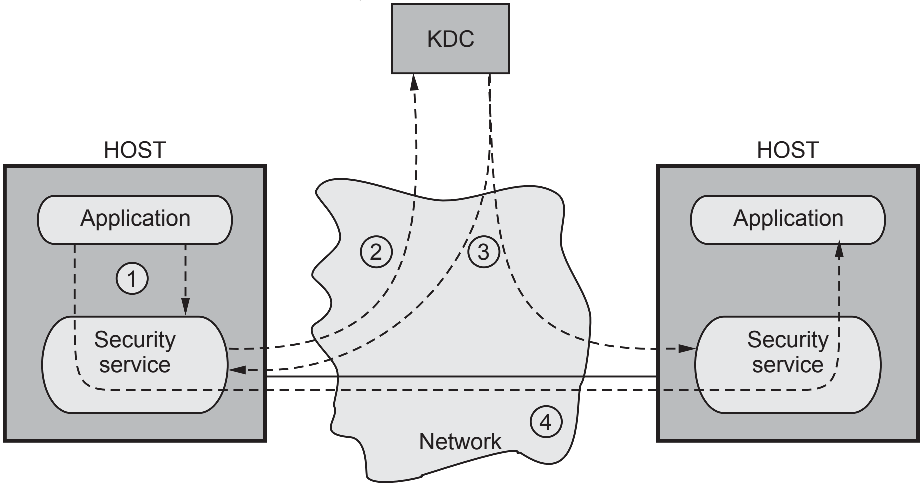


Fig. 8.1.10 Automatic key distribution for connection - oriented protocol

- Assume that communication make use of a connection-oriented end-to-end protocol, such as TCP.
- Following steps occurs :
  1. Host sends packet requesting connection.
  2. Session Security Module (SSM) saves that packet and applies to the KDC for permission to establish the connection.
  3. KDC distributes session key to both hosts.
  4. The requesting SSM can now release the connection request packet, and a connection is set up between the two end systems.

Decentralized key control

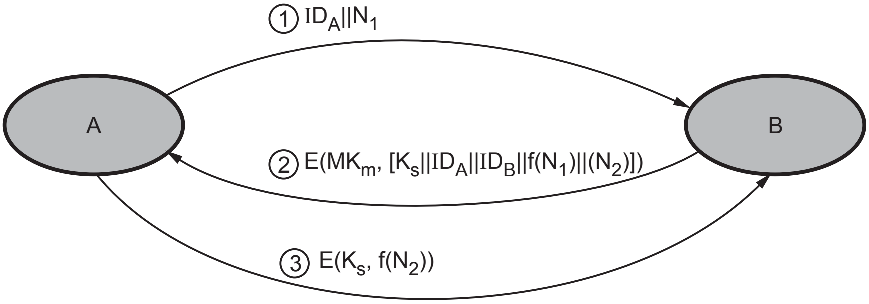


Fig. 8.1.11 Decentralized key distribution

- Decentralized approach requires that each end system be able to communicate in a secure manner with all potential parter end systems for purposes of session key distribution.
- A session key may be established with the following sequence of steps.
  1. A issues a request to B for a session key and includes a none, N_1.
  2. B responds with a message that is encryped using the shared master key.
  3. Using the new session key, A returns f (N_2) to B.

University Questions

1. *Explain various general categories of schemes for the distribution of public keys.* **GTU : Summer-17, Winter-17, Marks 7**
2. *Explain various public key distribution techniques.* **GTU : Summer-18, Marks 4**
3. *What is KDC ? List the duties of a KDC.* **GTU : Winter-18, Marks 4**
4. *What is difference between a session key and a master key ? List four categories of schemes for the distribution of public keys.* **GTU : Summer-19, Marks 4**

8.2 X.509 Certificates

GTU : Summer-17,19, Winter-18,19

- X.509 is part of X.500 recommandations for directory service i.e. set of servers which maintains a database of information about users and other attributes.
- X.509 defines authentication services e.g. certificate structure and authentication protocols. Also X.509 also defines alternative authentication protocols base on use of public-key certificates. The X.509 certificate format is emplied in S/MIME, IP security, SET and SSL/TLS.
- X.509 standard uses RSA algorithm and hash function for digital signature. Fig. 8.2.1 shows generation of public key certificate.

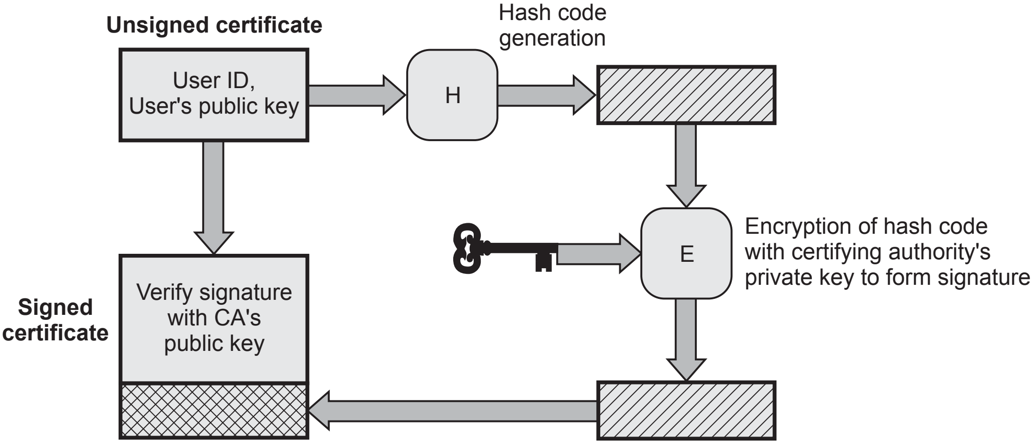

Unsigned certificate
Hash code generation
User ID, User's public key
H
E
Encryption of hash code with certifying authority's private key to form signature
Signed certificate
Verify signature with CA's public key

Fig. 8.2.1 Public key certificate

8.2.1 X.509 Format of Certificate

- The current version of the standard is version 3, called as X.509V3. The general format of digital certificate X.509V3 is shown in Fig. 8.2.2.

1	Version
2	Certificate Serial Number
3	Signature Algorithm Identifier
4	Issuer Name
5	Period of Validity
6	Subject Name
7	Subject's Public Key Info.
8	Issuer Unique Identifier
9	Subject Unique Identifier
10	Extensions
11	Signature

Fig. 8.2.2 X.509 Digital certificate format version 3

1. **Version** : Identifies successive versions of certificate format the default is version.
2. **Certificate Serial Number** : It contains an unique integer number, which is generated by Certification Authority (CA).
3. **Signature Algorithm Identifier** : Identifies the algorithm used by the CA to sign the certificate.
4. **Issuer Name** : Identifies the distinguished name of the CA that created and signed this certificate.
5. **Period of Validity** : Consists of two date-time values (not before and not after) within which the certificate is valid.
6. **Subject Name** : It specifies the name of the user to whom this certificate is issued.
7. **Subject's Public Key Information** : It contains public key of the subject and algorithms related to that key.
8. **Issuer Unique Identifier** : It is an optional field which helps to identify a CA uniquely if two or more CAs have used the same Issuer Name.
9. **Subject Unique Identifier** : It is an optional field which helps to identify a subject uniquely if two or more subjects have used the same Subject Name.

10. **Extensions** : One or more fields used in version 3. These extensions convey additional information about the subject and issuer keys.
11. **Signature :** It contains hash code of the fields, encrypted with the CA's private key. It includes the signature algorithm identifier.

Standard notations for defining a certificate

CA<<A>> = CA{V, SN, AI, CA, T_AA, A_P}

where,

CA<<A>> indicates the certificate of user A issued by certification authority CA.

CA{V ……… A_P} indicates signing of V……A_P by CA.

8.2.2 Obtaining User's Certificate

- The characteristics of user certificate are -
  1. Any user who can access public key of CA can verify user public key.
  2. Only certification Authority (CA) can modify the certificate.
- All user certificates are placed in a directory for access of other users. The public key provided by CA is absolutely secure (w.r.t. integrity and authenticity).
- If user A has obtained a certificate from CA X_1 and user B has obtained a certificate from CA X_2. If A don't know the public key of X_2, then B's certificate (issued by X_2) is useless to A. The user A can read B's certificate but A can not verify the signature. This problem can be resolved by securely exchanging the public keys by two CAs.

8.2.3 Revocation of Certificates

- The certificate should be revoked before expiry because of following reasons :
  1. User's private key is compromised.
  2. User is not certified by CA.
  3. CA's certificate is compromised.
- Each CA has a list of all revoked but not expired certificates. The Certificate Revocation List (CRL) is posted in directory signed by issuer and includes issuer's name, date of creation, date of next CRL. Fig. 8.2.3 Certificate revocation list. Each certificate has unique serial number of identify the certificate.

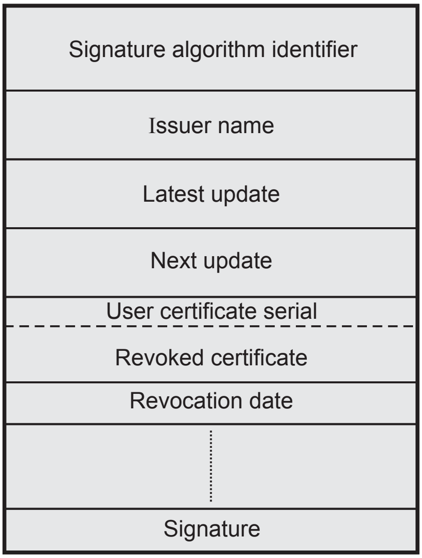


Fig. 8.2.3 Certificate revocation list

8.2.4 Authentication Procedures

- X.509 supports three types of authenticating using public key signatures. The types of authentication are
  1. One-way authentication
  2. Two-way authentication
  3. Three-way authentication

1. One-way authentication

- It involves single transfer of information from one user to other as shown in Fig. 8.2.4.

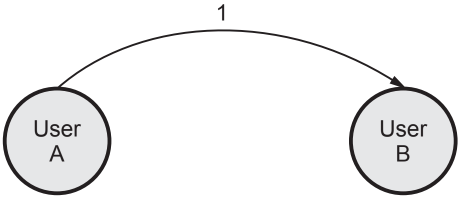


Fig. 8.2.4 One way authentication

2. Two-way authentication

- Two-way authentication allows both parties to communicate and verify the identity of the user.

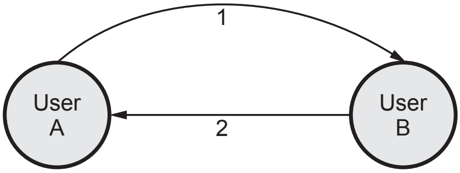


Fig. 8.2.5 Two-way authentication

3. Three-way authentication

Three-way authentication is used where synchronized clockes are not available Fig. 8.2.6 shows three-way authentication.

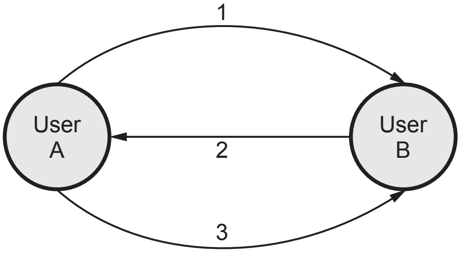


Fig. 8.2.6 Three-way authentication

University Questions

1. *Write a note on : X.509 Certificate Format.* **GTU : Summer-17, Marks 7**
2. *Describe briefly the authentication process covered by X.509.* **GTU : Winter-18, Marks 7**
3. *What is the purpose of X.509 standard ? How is an X.509 certificate revoked ?* **GTU : Summer-19, Marks 7**
4. *Explain use of public-key certificate with diagram and draw X.509 certificate format.* **GTU : Winter-19, Marks 7**

8.3 Public Key Infrastructure **GTU : Summer-18**

- Management and handling of the pieces of secret information is generally referred to as **key management.**
- Activities of key management includes selection, exchange, storage, certification, revocation, changing, expiration and transmission of the key.
- Key management is the set of processes and mechanisms which support key establishment and maintenance of ongoing keying relationship between parties, including replacing older key with new keys.
- Two major issues in key management are :

  1. Key life time 2. Key exposure

Key life time - limit of use which can be measured as a duration of time.

Issue related to key :

1. Users must be able to obtain securely a key pair suited to their efficiency and security needs.
2. Keys need to be valid only until a specified expiration date.
3. The expiration date must be chosen properly and publicized securely.
4. User must be able to store their private keys securely.
5. Certificates must be unforgettable, obtainable in a secure manner.

1. Public Key Infrastructure

- Public Key Infrastructure (PKI) is a well-known technology that can be used to establish identities, encrypt information and digitally sign documents.
- PKI identifies and manages relationships of parties in an electronic exchange, serving a wide array of security needs including access control, confidentiality, integrity, authentication and non-repudiation.
- PKI also uses unique Digital Certificates (DC) to secure eCommerce, email, data exchange and Virtual Private Networks (VPN) and intranets and is also used to verify the identity and privileges of each user.
- The Certificate Authority (CA) provides a full life-cycle management of public keys and certificates, including issuance, authentication, storage, retrieval, backup, recovery, updating and revocation to the PKI.
- All users of PKI must have a registered identity, which is stored in a digital certificate issued by a CA.
- Remote users and sites using public private keys and public key certificates can authenticate each other with a high degree of confidence.
- Authentication is dependent on three conditions :
  1. It must be established that each party have a private key that has not been stolen or copied from the owner.
  2. The certificate must be issued to the owner in accord with the stated policy of the certificate issuer.
  3. The policies of the certificate issuer must be satisfactory to the parties so as to verify identity.

Benefits of PKI

1. Confidential communication : Only intended recipients can read files.
2. Data integrity : Guarantees files are unaltered during transmission.
3. Authentication : Ensures that parties involved are who they claim to be.
4. Non-repudiation : Prevents individuals from denying.

Limitation of PKI

The problems encountered deploying a PKI can be categorized as follows :

1. Public key infrastructure is new
2. Lack of standards
3. Shortage of trained personnel
4. Public key infrastructure is mostly about policies.

2. Certificate

- **Certificates** are digital documents that are used for secure authentication of communicating parties.
- A certificate binds identity information about an entity to the entity's public key for a certain validity period.
- A certificate is digitally signed by a Trusted Third Party (TTP) who has verified that the key pair actually belongs to the entity.
- Certificates can be thought of as analogous to passports that guarantee the identity of their bearers.
- **Authorities :** The trusted party who issues certificates to the identified end entities is called a **Certification Authority (CA).**
- Certification authorities can be thought of as being analogous to governments issuing passports for their citizens.
- A certification authority can be managed by an external certification service provider or the CA can belong to the same organization as the end entities.
- CAs can also issue certificates to other (sub) CAs. This leads to a tree-like **certification hierarchy.**
- The highest trusted CA in the tree is called a root CA.
- In large organizations, it may be appropriate to delegate the responsibility for issuing certificates to several different certificate authorities.
- For example, the number of certificates required may be too large for a single CA to maintain; different organizational units may have different policy requirements; or it may be important for a CA to be physically located in the same geographic area as the people to whom it is issuing certificates.
- The X.509 standard includes a model for setting up a hierarchy of the Certification Authority.
- Fig. 8.3.1 shows the hierarchy of certificate authorities.

- In the Fig. 8.3.1, the root CA is at the top of the hierarchy. The root CA's certificate is a self-signed certificate : that is, the certificate is digitally signed by the same entity -- the root CA.

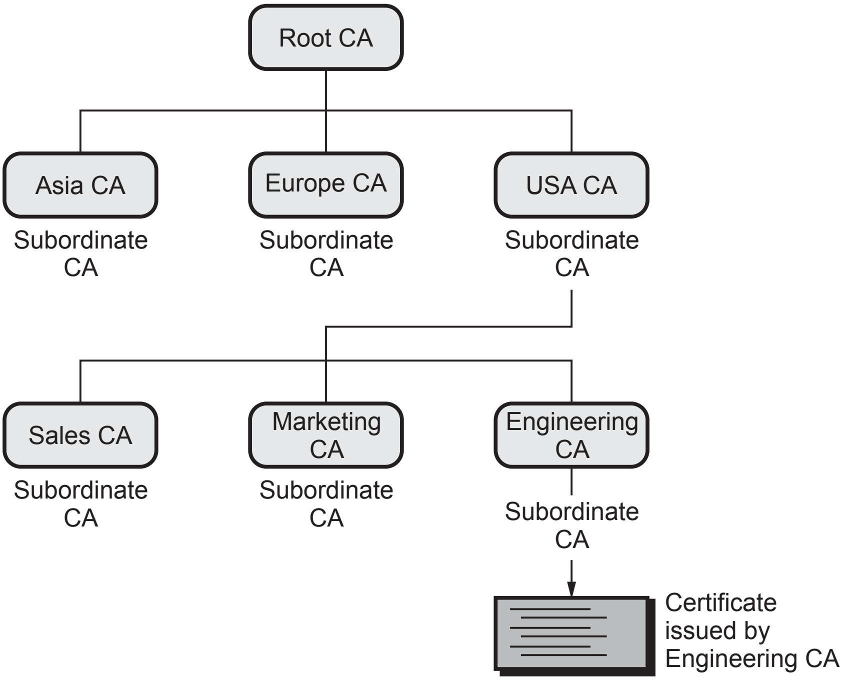


Fig. 8.3.1 Hierarchy of CA

- The CAs, that are directly subordinate to the root CA, have CA certificates signed by the root CA. CAs under the subordinate CAs in the hierarchy have their CA certificates signed by the higher-level subordinate CAs.
- Organizations have a great deal of flexibility in terms of the way they set up their CA hierarchies.
- **Certificate chains :** Certificate chain is series of certificates issued by successive CAs.
- In some cases, a CA can delegate the actual identification of end entities as well as some other administrative tasks to a separate entity, the **Registration Authority (RA).**

Verifying certificates

- When authentication is required, the entity presents a signature it has generated from authentication data using its private key, and a certificate corresponding to that key.
- The receiving entity can verify the signature with the public key of the sender contained in the certificate.

- Next the receiving entity must verify the certificate itself by checking the validity time of the certificate and the signature of the CA in the certificate.
- If the CA is a sub CA, the receiving entity must also verify the signatures of all higher-level CAs up to the root CA.
- The list of certificates needed for verification is called a **certification path.**
- If all signatures are valid and the receiving entity trusts the root CA, the first entity will be authenticated successfully.
- If a private key of an end entity is compromised or the right to authenticate with a certificate is lost before its natural expiration date, the CA must revoke the certificate and inform all PKI users about this.
- The CA will periodically publish a **Certificate Revocation List (CRL)**.
- The CRL is a list identifying the revoked certificates and it is signed by the CA.
- The end entities should check the latest CRL whenever they are verifying a validity of a certificate.

3. Key Length and Encryption Strength

- The strength of encryption depends on both the cipher used and the length of the key.
- Encryption strength is often described in terms of the size of the keys used to perform the encryption : in general, longer keys provide stronger encryption.
- Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher.
- Roughly speaking, 128-bit RC4 encryption is 3×10^{26} times stronger than 40-bit RC4 encryption.
- Different ciphers may require different key lengths to achieve the same level of encryption strength.
- The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based.
- Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values.
- Thus a 128-bit key for use with a symmetric key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher.

University Question

1. *Write a short note on public key infrastructure.* **GTU : Summer-18, Marks 7**

8.4 Short Questions and Answers

Q.1 What is key management?

Ans. : Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties.

Q.2 What is master key?

Ans. : Session keys are transmitted in encrypted form, using a master key that is shared by the key distribution center and an end system or user.

Q.3 Define session key.

Ans. : Communication between end systems is encrypted using a temporary key, often referred to as a session key.

Q.4 What is PKI?

Ans. : A Public-Key Infrastructure (PKI) is defined as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.

Q.5 What is key distribution?

Ans. : Key distribution is the function that delivers a key to two parties who wish to exchange secure encrypted data. Some sort of mechanism or protocol is needed to provide for the secure distribution of keys.

Q.6 What is digital certificate?

Ans. : Certificates are digital documents that are used for secure authentication of communicating parties.

Q.7 What is Certification Authority?

Ans. : The trusted party who issues certificates to the identified end entities is called a Certification Authority.

Q.8 What is a nonce?

Ans. : A random value to be repeated in message to assure that the response is fresh and has not been replayed by an opponent.

8.5 Multiple Choice Questions

Q.1 The trusted party who issues certificates to the identified end entities is called a ________.

a certification hierarchy

b Registration Authority

c Certification Authority

d All of these

Q.2 The ________ standard includes a model for setting up a hierarchy of the Certification Authority.

a X.501 b X.509

c X.521 d X.599

Q.3 X.509 is based on the use of _________ cryptography and ______ .

a Private key, digital certificate b Public key, digital certificate

c Private key, digital signature d Public key, digital signature

Q.4 Following is NOT a method of distribution of public keys.

a Public announcement. b Publicly available directory.

c Private key authority d Public key certificates.

Q.5 Communication between end systems is encrypted using a temporary key, often referred to as a _________ .

a Master key b private key c public key d session key

Q.6 Session keys are transmitted in encrypted form, using a ————- that is shared by the key distribution center and an end system or user.

a Master key b private key c public key d session key

Answer Keys for Multiple Choice Questions

Q.1	c	**Q.2**	b	**Q.3**	d
Q.4	c	**Q.5**	d	**Q.6**	a

❑❑❑

Notes

9 User Authentication

Syllabus

Remote user authentication with symmetric and asymmetric encryption, Kerberos.

Contents

9.1 Remote User Authentication Principle GTU : Summer-19

- Authentication Protocols are used to convince parties of each other's identity and to exchange session keys. They may be one-way or mutual.
- Authentication techniques are used to verify identity. The authentication of authorized users prevents unauthorized users from gaining access to corporate information systems. Authentication method is of validating the identity of user, service or application. The use of authentication mechanisms can also prevent authorized users from accessing information that they are not authorized to view.
- User is authenticated by four ways :
  1. Something the individual knows : Examples include a password, a personal identification number (PIN).
  2. Something the individual possesses : Examples include cryptographic keys, electronic keycards, smart cards and physical keys.
  3. Something the individual is : Examples include recognition by fingerprint, retina and face.
  4. Something the individual does : Examples include recognition by voice pattern, handwriting characteristics etc.

9.1.1 Mutual Authentication

- Mutual authentication allows for both ends to know that they truly know whom they are communicating with.
- 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. Timeline prevent the replay attacks.
- 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. Timeline prevent the replay attacks.

Examples of replay attacks

1. Simply replay : The opponent simply copies a message and replays it later.
2. Repetition that can be logged : Replay time stamped message within valid time.

3. Repetition that cannot be detected : Original message suppressed and only reply message arrives.

4. Backward replay without modification.

Replay attack countermeasures

- Replay Attacks are where a valid signed message is copied and later resent. 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.
- Possible countermeasures include the use of :
  1. Sequence numbers : Generally impractical since must remember last number used with every communicating party.
  2. Timestamps : Needs synchronized clocks amongst all parties involved, which can be problematic.
  3. Challenge/response : Using unique, random, unpredictable nonce, but not suitable for connectionless applications because of handshake overhead.

9.1.2 One Way Authentication

- It involves single transfer of information from one user to other.
- Client authenticates itself to the server. The server may or may not be authenticated to the client. This is referred to as one way authentication.

9.1.2.1 Password based Authentication

- Password is a front line protection against the unauthorized access (intruder) to the system. A password authenticate the identifier (ID) and provides security to the system. Therefore almost all systems are password protected.

1] Password vulnerability

Passwords are extremely common. Passwords can often be guessed. Use of mechanisms to keep passwords secret does not guarantee that the system security can not be broken. It only says that it is difficult to obtain passwords. The intruder can always use a trial and error method. A test of only a limited set of potential strings tends to reveal most passwords because there is a strong tendency for people to choose relatively short and simple passwords that they can remember. Some techniques that may be used to make the task of guessing a password difficult are as follows

1. Longer passwords.

2. Salting the password table.

3. System assistance in password selection.

The length of a password determines the ease with which a password can be found by exhaustion. For example, 3-digit password provides 1000 variations whereas a four digit passwords provides 10,000 variations. Second method is the system assistance. A password can be either system generated or user selected. User selected passwords are often easy to guess. A system can be designed to assist users in using passwords that are difficult to guess.

2] Encrypted passwords

Instead of storing the names and passwords in plain text form, they are encrypted and stored in cipher text form in the table. In this case, instead of directly using a user specified name and password for table lookup, they are first encrypted and then the results are used for table lookup. If the stored encoded password is seen, it can not be loaded, so the password cannot be determined. The password file does not need to be kept secret.

3] One time passwords

Set of paired passwords solve the problem of password sniffing. When a session begins, the system randomly selects and presents one part of a password pair; user must supply the other part. In this, user is challenged and must respond with the correct answer to that challenge. In this method, the password is different in each instance. One time passwords are among the only ways to prevent improper authentication due to password exposure. Commercial implementations of one time password system such as secur ID, use hardware calculators.

Password selection strategies

- Too short password is too easy to guess. If the passward is 8 random character, it is impossible to crack the password. In order to eliminate gaussable passwords four basic techniques are suggested.
  1. User education
  2. Computer generated password
  3. Reactive password checking
  4. Proactive password checking

University Question

1. *List three approaches to secure user authentication in a distributed environment.*

GTU : Summer-19, Marks 4

9.2 Remote User Authentication with Symmetric

1. Mutual Authentication :

- The Needham Schroeder protocol refers to two methods of communication protocols through an insecure network.
  1. Needham Schroeder symmetric key protocol, which is based on symmetric encryption algorithm to establish a session key between two parties in a network.
  2. Needham Schroeder public-key protocol, based on the public key cryptography to provide mutual authentication between two communication parties over a network.

Needham Schroeder public key authentication protocol

- The Needham Schroeder public key authentication protocol aims to provide a mutual authentication between two parties Alice (A) and Bob (B).
- Both parties want to insure each other identity before starting to communicate.
- The protocol is as follows :
  a. K_A and K_B are Alice's public key and Bob's public key respectively,

  b. N_A and N_B are nonces generated by A and B respectively.

  1. $A \rightarrow B : \{N_A , A\}_{K_B}$ (Init)

Alice generates a nonce N_A and sends it to Bob with her identity. Everything is encrypted using Bob's public key.

2. $B \rightarrow A : \{N_A , N_B\}_{K_A}$ (Challenge)

Bob generates a nonce N_B, and sends it to Alice with N_A he has just received. It is a way to prove that he is really the owner of the private key corresponding to K_B. In other word, this mechanism is implemented in order to authenticate Bob. Sending back to Alice N_A is also a way to avoid a replay of this message.

3. $A \rightarrow A : \{N_B\}_{K_B}$ (Response)

Alice decrypts the message and check if it contains the right value of N_A. Then, she sends back N_B to Bob to prove her ability to decrypt with her private key and so to authenticate herself.

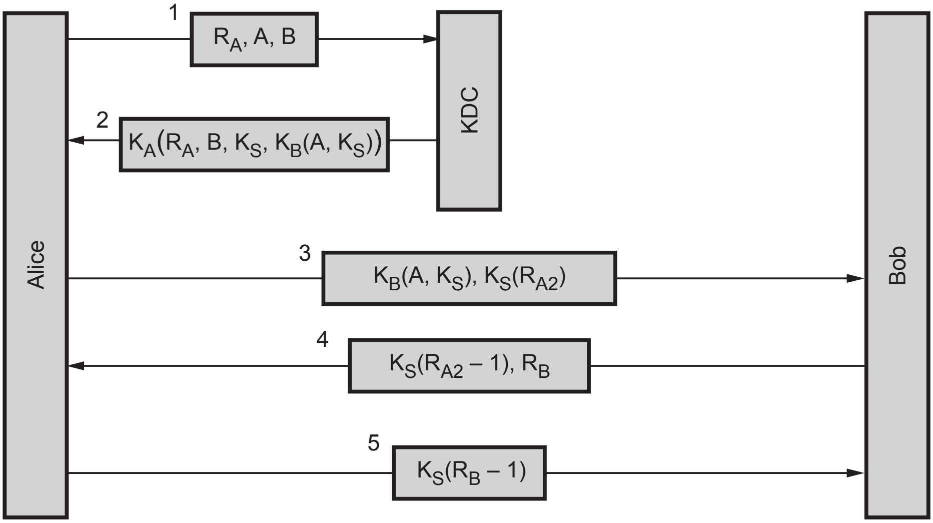


Fig. 9.2.1 Needham Schroeder authentication protocol

2. Based on Shared Secret Key :

In this protocol, a secret key is shared with both party. i.e. source and destination. One party sends random number to the other, other side transforms it in a special way and then returns a result. This type of protocols are called challenge-response protocols. The working of this protocol is as follows.

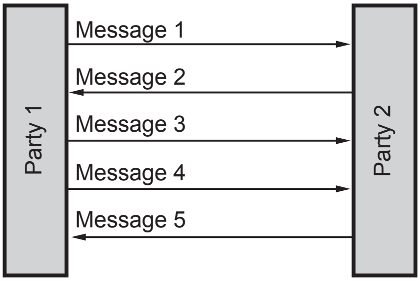


Fig. 9.2.2 Two way authentication using a challenge-response protocol

First the party 1 sends a message 1 to party 2 i.e. identification of party 1. The party 2 needs to find out the message which it received is from party 1 or any other third party. Party 2 sends a large random number to party 1 in plaintext. The party 1 then encrypts the message with the key which shares with party 2 and sends the ciphertext back in message 3. When party 2 receives this message, they know that message is from party 1 because of the shared secret key. Uptill now party 2 is sure only about communication, but party 1 is not sure about the communication between him and party 2. The party 1 sends a random number to party 2 as plaintext in message 4. When party 2 responds with secret key, party 1 knows they are communicating with party 2. This protocol has some disadvantages. It is slower and contains extra messages. These can be eliminated by combining information.

9.3 Remote User Authentication with Asymmetric

1. Mutual Authentication :

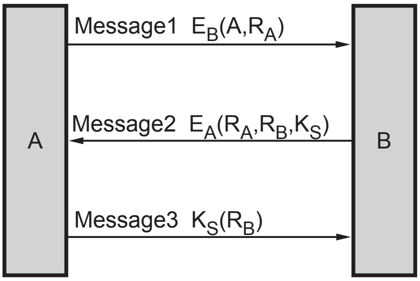


Fig. 9.3.1 Authentication using public key cryptography

In this method, A sends a random number R_A and identity by encrypting. A uses B's public-key E_B for sending message. When B receives this messages, B sends A back a message containing A's random number R_A and his own random number R_B and a proposed session key, K_s. When A gets message 2, A decrypts it using private key. After examining the message 2, A finds out the random number R_A. A knows that message 2 is from B only. Then A agrees to the session by sending back message 3 to B. When B reads R_B encrypted with the session key which is generated by B, B knows that A got message 2 and verified R_A.

This protocol does has some disadvantage. It assumes that both user (A and B) already know each others public keys.

2. Certificate Based Authentication :

- Client have a public key certificate. Fig. 9.3.2 shows the certificate based authentication.

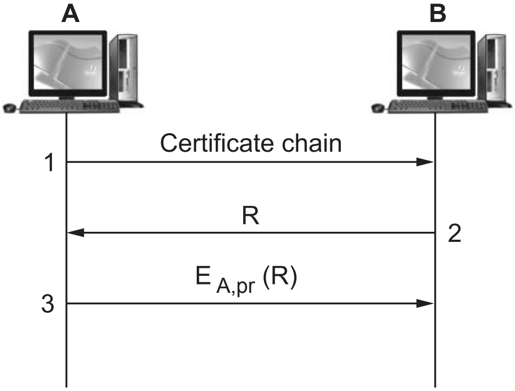


Fig. 9.3.2 Certificate based authentication

- A sends his certificate in message 1.
- B performs certain checks which includes principal name, validity period, certificate authority etc.
- B then sends his challenge i.e a nonce R.
- A responds by encrypting the challenge with his private key.
- When B receives $E_{A,pr}$(R), he decrypts it with A's public key and compares it with the nonce he transmitted in message 2.
- It they match, he concludes that A has used the private key corresponding to the public key in his certificate.

9.4 Kerberos

GTU : Summer-17, 18, 19, Winter-17, 19

- Kerberos is an **authentication protocol**. It provides a way to authenticate clients to services to each other through a trusted third party.
- Kerberos makes the assumption that the connection between a client and service is insecure. Passwords are encrypted to prevent others from reading them. Clients only have to authenticate once during a pre-defined lifetime.
- Kerberos was designed and developed at MIT by Project Athena. Currently, Kerberos is upto Version 5. Version 4 being the first version to be released outside of MIT.
- Kerberos has been adopted by several private companies as well as added to several operating systems.
- Its creation was inspired by client-server model replacing time-sharing model. Kerberos is a network authentication protocol designed to allow users, clients and servers, authenticate themselves to each other.
- This **mutual authentication** is done using **secret-key cryptography** with parties proving to each other their identity across an insecure network connection.
- Communication between the client and the server can be secure after the client and server have used Kerberos to prove their identity.
- From this point on, subsequent communication between the two can be encrypted to assure privacy and data integrity.

Requirement of Kerberos

- Kerberos client/server authentication requirements are :
  1. **Security :** That Kerberos is strong enough to stop potential eavesdroppers from finding it to be a weak link.
  2. **Reliability :** That Kerberos is highly reliable employing a distributed server architecture where one server is able to back up another. This means that Kerberos systems are fail safe, meaning graceful degradation, if it happens.
  3. **Transparency :** That user is not aware that authentication is taking place beyond providing passwords.
  4. **Scalability :** Kerberos systems accept and support new clients and servers.
- To meet these requirements, Kerberos designers proposed a third-party trusted authentication service to arbitrate between the client and server in their mutual authentication.

9.4.1 Kerberos Terminology

- Kerberos has its own terminology to define various aspects of the service.
  1. **Authentication Server (AS) :** A server that issues tickets for a desired service which are in turn given to users for access to the service.
  2. **Client :** An entity on the network that can receive a ticket from Kerberos.
  3. **Credentials :** A temporary set of electronic credentials that verify the identity of a client for a particular service. It also called a ticket.
  4. **Credential cache or ticket file :** A file which contains the keys for encrypting communications between a user and various network services.
  5. **Crypt hash :** A one-way hash used to authenticate users.
  6. **Key :** Data used when encrypting or decrypting other data.
  7. **Key Distribution Center (KDC) :** A service that issue Kerberos tickets and which usually run on the same host as the Ticket-Granting Server (TGS).
  8. **Realm :** A network that uses Kerberos composed of one or more servers called KDCs and a potentially large number of clients.
  9. **Ticket-Granting Server (TGS) :** A server that issues tickets for a desired service which are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.
  10. **Ticket-Granting Ticket (TGT) :** A special ticket that allows the client to obtain additional tickets without applying for them from the KDC.

9.4.2 Kerberos Version 4

- Kerberos version 4 uses DES for providing authentication service. Some aspect of version 4 are

  A) Simple Authentication Dialogue. B) More Secure Authentication Dialogue.

9.4.2.1 Simple Authentication Dialogue

- For a secure transaction, server should confirm the client and its request. In unprotected network it creates burden on server, therefore an authentication server (AS) is used. The authentication server (AS) maintains passward of all users in centralized database. Also the authentication server shares a unique secret key with each server.
- Let

  Client is represented as C

  Authentication server is represented as AS

  Server is represented as V

Identifier of user on C is represented as ID_C
Identifier of V is represented as ID_V
Passward of user on C is P_C

Network address of C is represented as AD_C

Secret encryption key shared by AS and V is K_V

Then consider a hypothetical dialogue.

	Sender and receiver		Contents of message
1.	C → AS	:	$ID_C \| P_C \| ID_V$
2.	AS → C	:	Ticket
3.	C → V	:	$ID_C \|$ Ticket
4.	Ticket	=	$E [K_V , (ID_C \| AD_C \| ID_V)]$

Explanation

1. **Client Clogs on to workstation requesting to access to server V :** The workstation requests user's passward and sends message to AS including user ID + server ID + user passward. The AS checks this message with database and verifies it.
2. **AS issues ticket :** On verifying the tests. AS issues ticket containing user ID + server ID + network address.
3. **Client C applies server V :** With this ticket, client C asks server V for access. Server V decrypts the ticket and verify the aunthenticity of data then grants the requested service. In above hypothetical dialogue, symbol || represents concatenation.

9.4.2.2 Secure Authentication Dialogue

- Kerberos version 4 protocol ensures secure authentication dialogue involving three sessions.
  i] Authentication Service - Exchange to obtain ticket-granting ticket.
  ii] Ticket-granting Service - Exchange to obtain service granting ticket.
  iii] Client/server authentication - Exchange to obtain service.
- Each of the above session has two steps, as shown in table below

Session	Step	Sender-Receiver
[i]	1. 2.	C → AS AS → C
[ii]	3. 4.	C → TGS (Ticket-granting server) TGS → C

[iii]	5.	C → V V → C

- Fig. 9.4.1 shows how the steps are executed in Kerberos version 4.

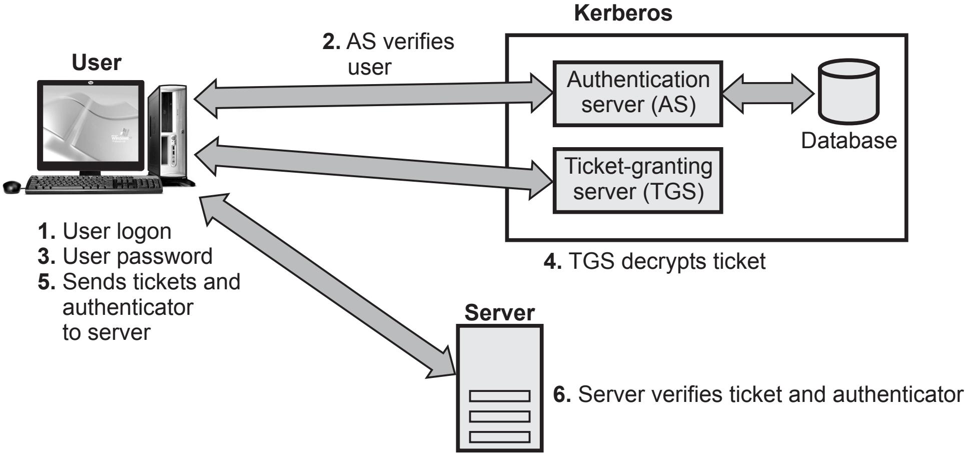

Kerberos
User
2. AS verifies user
Authentication server (AS)
Database
Ticket-granting server (TGS)
1. User logon
3. User password
5. Sends tickets and authenticator to server
4. TGS decrypts ticket
Server
6. Server verifies ticket and authenticator

Fig. 9.4.1 Overview of kerberos

9.4.2.3 Kerberos Realms

- The constituents of a full-service Kerberos environment are
  a) A Kerberos server
  b) Clients
  c) Number of application server.
- Requirements of Kerberos sever :
  a) Kerberos server should have user ID.
  b) Hashed password for all users.
  c) All users should be registered with Kerberos server.
  d) Kerberos server should have secret key with each server.
  e) All servers should be registered with Kerberos server.
- A **Kerberos realm** is referred as is the environment where
  - all nodes share same secured database.
  - changing and accessing the Kerberos database requires Kerberos master password.
  - a read only copy of Kerberos database resides in computer system.
- Networks have different realms under different administrative organizations. The users of one realm may access the servers in other realm provided the users are authenticated. The interoperating Kerberos shares a secret key with the server in other realm.

9.4.3 Kerberos Version 5

- Version 4 of Kerberos have some environmental shortcoming and technical deficiencies.

Environmental shortcomings of version 4

1. Encryption system dependence
2. Internet protocol dependence
3. Message byte ordering
4. Ticket lifetime
5. Authentication forwarding
6. Inter realm authentication.

Technical deficiencies of version 4

1. Double encryption
2. PCBC (Propagating Cipher Block Chaining) encryption
3. Session keys
4. Password attacks

9.4.3.1 Version 5 Authentication Dialogue

- The Kerberos version 5 message exchange involves three session, these are
  1. Authentication Service Exchange
  2. Ticket - Granting Exchnage
  3. Client/Server Authentication Exchange
- Each session has two steps. Table 9.4.1 summarizes session, steps and their functions.

	Session	Step	Function
[i]	Application Service Exchange	C → AS AS → C	To obtain ticket-granting ticket.
[ii]	Ticket-Granting Service Exchange	C → TGS TGS → C	To obtain service-granting ticket.
[iii]	Client/Server Authentication Exchange	C → V V → C	To obtain service.

Table 9.4.1

- The flags field is expanded in ticket in version 5 of Kerberos. Various flags that may be included in a ticket, are

  i) INITIAL

  ii) PRE - AUTHENT

  iii) HW - AUTHENT

  iv) RENEWABLE

  v) MAY-POSTDATE

  vi) POSTDATED

  vii) INVALID

  viii) PROXIABLE

  ix) PROXY

  x) FORWARDABLE

  xi) FORWARDED

University Questions

1. *Write a detailed note on : Kerberos.* **GTU : Summer-17, Marks 7**
2. *Explain the concept of Realm in Kerberos in brief.* **GTU : Winter-17, Marks 3**
3. *Explain authentication mechanism of Kerberos.* **GTU : Summer-18, Marks 7**
4. *What problem was Kerberos designed to address? What are the three threats associated with user authentication over a network or internet.* **GTU : Summer-19, Marks 4**
5. *What problem was Kerberos designed to address ?* **GTU : Winter-19, Marks 3**

9.5 Short Questions and Answers

Q.1 What is key management ?

Ans. : Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties.

Q.2 What is master key ?

Ans. : Session keys are transmitted in encrypted form, using a master key that is shared by the key distribution center and an end system or user.

Q.3 Define session key.

Ans. : Communication between end systems is encrypted using a temporary key, often referred to as a session key.

Q.4 What is PKI ?

Ans. : A Public-Key Infrastructure (PKI) is defined as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.

Q.5 What is key distribution ?

Ans. : Key distribution is the function that delivers a key to two parties who wish to exchange secure encrypted data. Some sort of mechanism or protocol is needed to provide for the secure distribution of keys.

9.6 Multiple Choice Questions

Q.1 Password based authentication can be divided into two broad categories : _____ and ______.

a Fixed, Variable
b Time stamped, fixed
c Fixed, one-time
d None of the above **[Ans. : c]**

Q.2 Challenge-response authentication can be done using ____

a symmetric-key ciphers
b asymmetric-key ciphers
c keyed-hash functions
d all of the above **[Ans. : d]**

Q.3 __________ is a popular session key creator protocol that requires an authentication server and a ticket granting server.

a KDC
b Kerberos
c CA
d none of the above **[Ans. : b]**

Q.4 For each _______ the Kerberos Key Distribution Center (KDC) maintains a database of the realm's principal and the principal's associated "secret keys".

a key
b realm
c document
d none of the mentioned **[Ans. : b]**

Answer Keys for Multiple Choice Questions

Q.1	c	**Q.2**	d
Q.3	b	**Q.4**	b

❑❑❑

10 Web Security

Syllabus

Web Security threats and approaches, SSL architecture and protocol, Transport layer security, HTTPS and SSH.

Contents

10.1 Web Security Threats and Approaches

- A birthday attack refers to a class of brute-force attacks.
- The attack is named after the statistical property of birthday duplication - you only need 23 people to have a larger than 50 % chance that they are born on the same day of the year.
- This is due to the fact that each time you adding one person to the set of people you are looking for duplicates in, you are looking for duplicates against all the people already in the set, not just one of them.
- The same technique can be used to look for conflicts in one-way functions. Instead of taking one output of the one-way function, you create or acquire a set of values (let us call this a) that have a some property and then create another set of other values that have different properties (let us call this b) and try to find any value that is in both a and b. This is a much smaller problem that finding a value that match a particular value in a.
- The properties in a and b might for instance be
  1. a contains secure hashes of an innocent message and b contains one of a less innocent message, so the attacker can substitute the messages at a later date.
  2. a is the password hashes of a system the attacker wants to get an account on, and b is a set of password hashes that the attacker knows the passwords for.
  3. a is the set of public keys from a Discrete Logarithms based cryptosystem where g and p are static, while b is the set of g^e mod p functions that the attacker knows e for.
- Brithday attacks are often used to find collisions of hash functions. To avoid this attack, the output length of the hash function used for a signature scheme can be chosen large enough so that the birthday attack becomes computationally infeasible.

Resistance against this attack is why the Unix password hashes use a salt.

University Questions

1. *Briefly explain web security Threats.* **GTU : Summer-17, Winter-17, Marks 3**
2. *List out the various web security threats.* **GTU : Winter-18, Marks 3**
3. *Which types of security threats are faced by user while using the web?* **GTU : Summer-19, Marks 3**

10.2 SSL

- SSL protocol is an internet protocol for secure exchange of information between a web browser and a web server.
- SSL is designed to make use of TCP to provide a reliable end to end secure service.
- Secure Socket Layer (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 a web server.

Features of SSL

1. SSL server authentication, allowing a user to confirm a server's identity.
2. SSL client authentication, allowing a server to confirm a user's identity.
3. An encrypted SSL session, in which all information sent between browser and server is encrypted by a sending software and decrypted by the receiving software.
4. SSL supports multiple cryptographic algorithms.

10.2.1 SSL Architecture

- SSL uses TCP to provide reliable end-to-end secure service. SSL consists of two subprotocols, one for establishing a secure connection and other for using it. Fig. 10.2.1 shows SSL protocol stack.

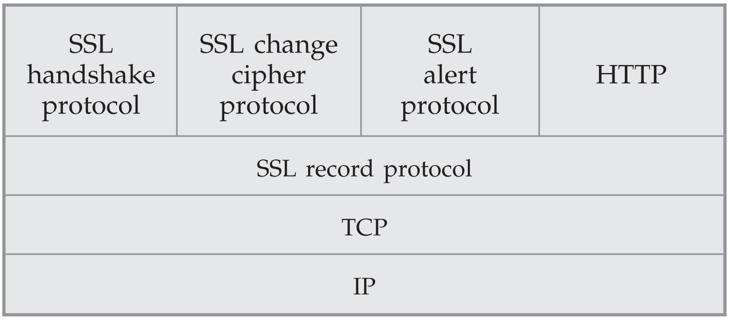


Fig. 10.2.1 SSL protocol stack

- SSL record protocol : It provides basic security services to various higher layer protocols.

HTTP	:	Provides the transfer service for web client/server interaction.
SSL Handshake protocol, SSL Change cipher protocol, SSL Alert protocol.	:	Management of SSL exchanges.

10.2.2 SSL Record Protocol

- The SSL Record protocol provides two services for SSL connection.
  1. **Confidentiality** - Handshake protocol for encryption of SSL payload.
  2. **Message integrity** - Handshake protocol for Message Authentication Code (MAC).

SSL record protocol operation is shown in Fig. 10.2.2.

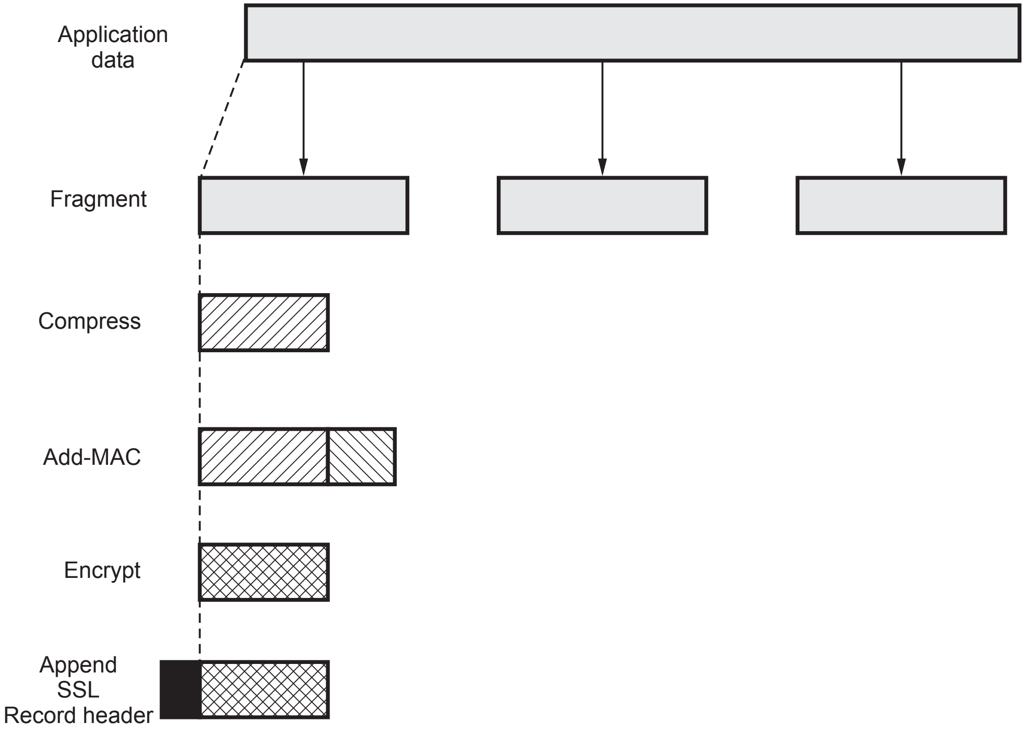


Fig. 10.2.2 SSL Record protocol operation

- The record protocol takes application message to transmit, fragments the data, compress, applies MAC, encrypts, adds a header and transmits the TCP segment.

10.2.3 Handshake Protocol

- Handshake protocol allows the server and client to authenticate each other and to negotiate an encryption before transmitting application data various massages are used in protocol. Table 10.2.1 enlist these messages and there associated function.

Phase	Message type	Function
1.	Hello - request Client - hello Server - hellow	Null Version, session id, cipher, compression Version, session id, cipher, compression.

2.	Certificate Server - key - exchange Certificate - request Server - done	Chain of X.509 V3 certificates. Parameters, signature. Type, authorities. Null
3.	Certificate - verify	Signature
4.	Client - key - exchange finished.	Parameters, signature hash value.

Table 10.2.1 SSL handshake protocol message types

- Fig. 10.2.3 shows handshake protocol action

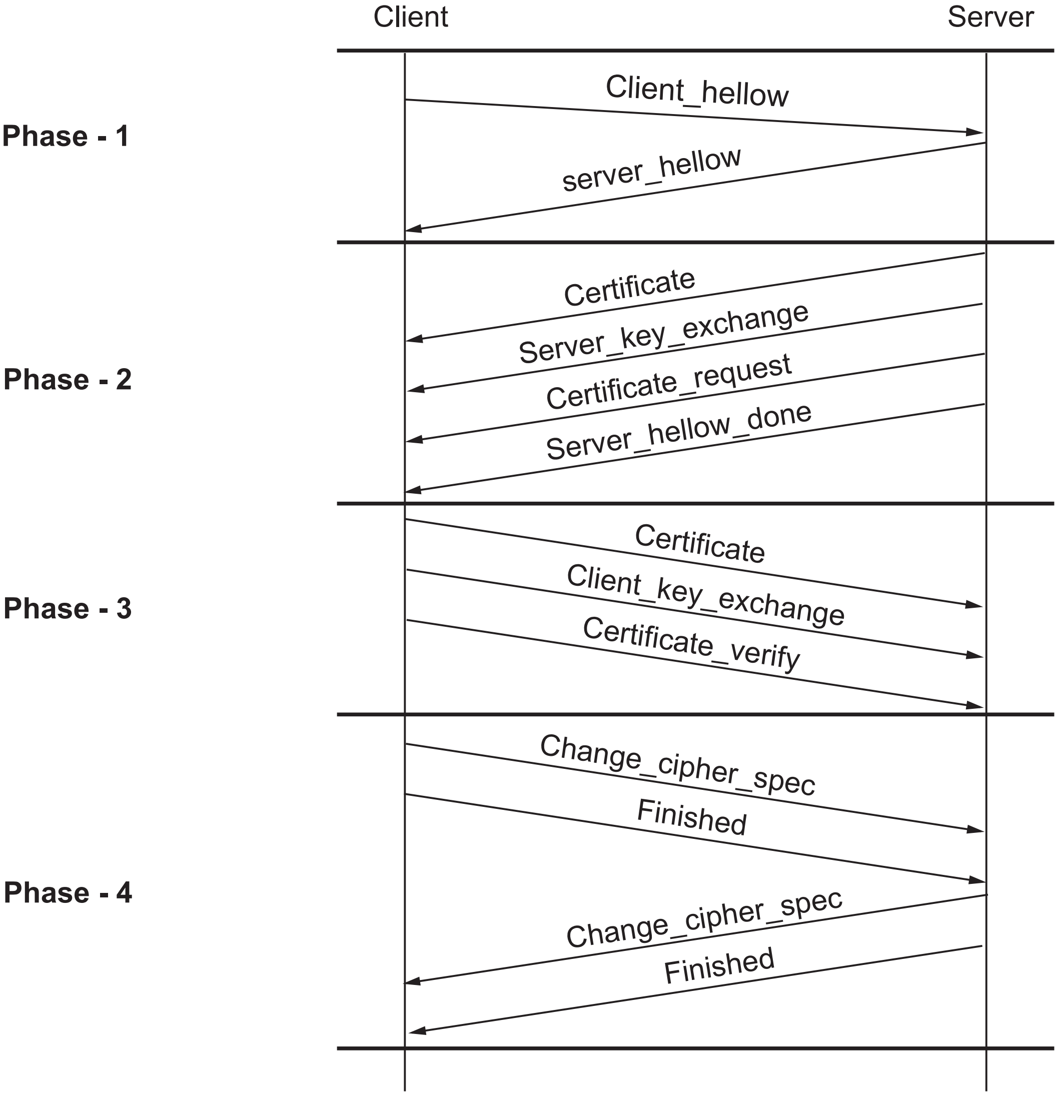


Fig. 10.2.3 Handshake protocol action

10.2.4 Comparison between IPSec and SSL

Sr. No.	Parameters	IPSec	SSL
1.	Position in the OSI model	Internet Layer	Between transport and application layers
2.	Configuration	Complex	Simple
3.	NAT	Problematic	No Problem
4.	Software Location	Kernel Area	User Area
5.	Firewall	Not Friendly	Friendly
6.	Installation	Vender Non-specific	Vender Specific
7.	Interoperability	Yes	No
8.	Deploy	More expensive to deploy, support and maintain.	Less costly to deploy and maintain

10.2.5 Comparison of SSL and TLS

SSL	TLS
In SSL the minor version is 0 and the major version is 3.	In TLS, the major version is 3 and the minor version is 1.
SSL use HMAC algorithm except that the padding bytes concatenation.	TLS makes use of the same algorithm the padding bytes concatenation.
SSL supports 12 various alert codes.	TLS supports all of the alert codes defined in SSL 3 with the exception of no certificate.

University Questions

1. *Discuss SSL architecture in brief.* **GTU : Summer-17, Marks 4**
2. *Briefly discuss the working of SSL Record Protocol.* **GTU : Winter-17, Marks 4**
3. *Write a short note on SSL.* **GTU : Summer-18, Marks 7**
4. *Explain HAND SHAKE protocol in SSL.* **GTU : Winter-18, Marks 3**
5. *Define the parameters that define SSL session state and session connection.* **GTU : Summer-19, Marks 3**

10.3 Transport Layer Security

- Transport Layer Security (TLS) is a feature of mail servers designed to secure the transmission of electronic mail from one server to another using encryption technology. TLS can reduce the risk of eavesdropping tampering and message forgery mail communications.

- TLS is a security protocol from the Internet Engineering Task Force (IETF) that is based on the Secure Sockets Layer (SSL) 3.0 protocol developed by Netscape.
- TLS was designed to provide security at the transport layer. TLS is a non-proprietary version of SSL. For transactions on Internet, a browser needs :

1. Make sure that server belongs to the actual vendor.
2. Contents of message are not modified during transition.
3. Make sure that the imposter does not interpret sensitive information such as credit card number.

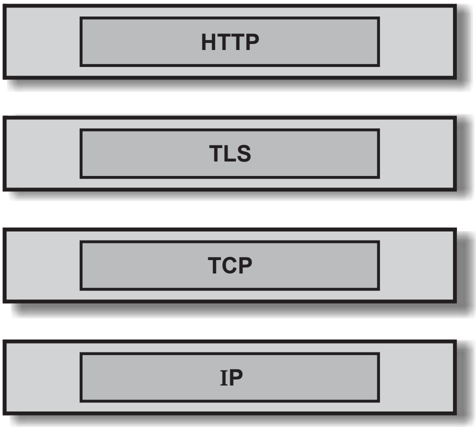


Fig. 10.3.1

- Fig. 10.3.1 shows the position of TLS in the protocol.
- TLS has two protocols : Handshake and data exchange protocol
  1. **Handshake :** Responsible for negotiating security, authenticating the server to the browser and (optionally) defining other communication parameters. The TLS handshake protocol allows authentication between the server and client and the negotiation of an encryption algorithm and cryptographic keys before the application protocol transmits or receives any data.
  2. **Data exchange (record) protocol :** Data exchange (record) protocol uses the secret key to encrypt the data for secrecy and to encrypt the message digest for integrity. The TLS record protocol is designed to protect confidentiality by using symmetric data encryption.

Handshake protocol

- Fig. 10.3.2 shows the TLS handshake protocol.

1. Browser sends a hello message that includes TLS version and some preferences.
2. Server sends a certificate message that includes the public key of the server. The public key is certified by some certification authority, which means that the public key is encrypted by a CA private key. Browser has a list of CAs and their public keys. It uses the corresponding key to decrypt the certification and finds the server public key. This also authenticates the server because the public key is certified by the CA.

3. Browser sends a secret key, encrypts it with the server public key and sends it to the server.

4. Browser sends a message, encrypted by the secret key to inform the server that handshaking is terminating from the browser key.

5. Server decrypts the secret key using it private key and decrypts the message using the secret key. It then sends a message, encrypted by the secret key, to inform the browser that handshaking is terminating from the server side.

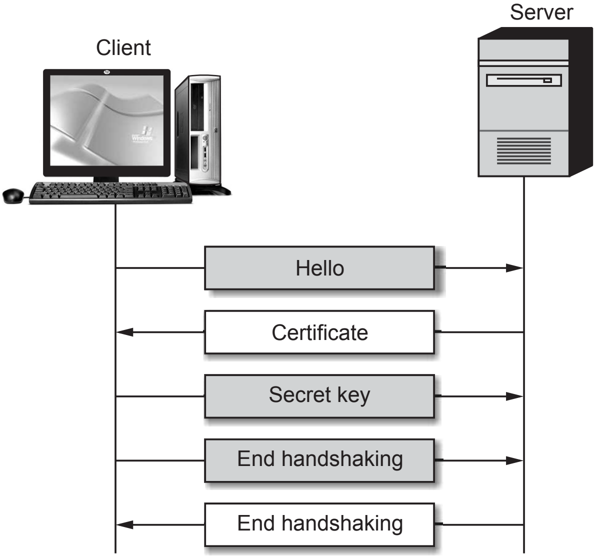


Fig. 10.3.2 TLS handshake protocol

10.4 HTTPS

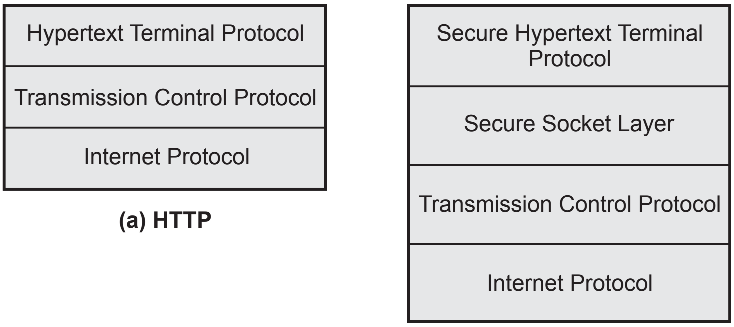
- Secure HTTP is an extension to the Hypertext Transfer Protocol (HTTP) that allows the secure exchange of files on the World Wide Web.
- In HTTP basic authentication, the client sends his username and password in clear text as part of the HTTP request.
- In all subsequent HTTP requests for content from subdirectories of the original request, these credentials will be automatically resent.
- In HTTP Digest authentication, no passwords are sent in the clear. Instead, a cryptographic hash value containing the username, password, and additional security-relevant data, will be transmitted from the client to the server.

HTTP Problems :

1. HTTP Basic authentication is vulnerable to passive eavesdropping. Moreover, it provides no mechanism for explicit session expiration (i.e. logout).
2. HTTP Digest authentication cannot guarantee sufficient support on all client platforms.
3. Both mechanisms do not provide session tracking, but only authentication.

- Secure HTTP indicates to user that page contents were not viewed or modified by a network attacker.

- Each S-HTTP file is either encrypted, contains a digital certificate, or both. For a given document, S-HTTP is an alternative to another well-known security protocol, Secure Sockets Layer (SSL).
- A major difference is that S-HTTP allows the client to send a certificate to authenticate the user whereas, using SSL, only the server can be authenticated. S-HTTP is more likely to be used in situations where the server represents a bank and requires authentication from the user that is more secure than a user-id and password.
- Fig. 10.4.1 shows position of HTTPS.


Hypertext Terminal Protocol
Transmission Control Protocol
Internet Protocol
Secure Hypertext Terminal Protocol
Secure Socket Layer
Transmission Control Protocol
Internet Protocol

(a) HTTP

(b) Secure HTTP

Fig. 10.4.1 Position of secure HTTP

- Secure HTTP layer is above the SSL and TCP protocol. HTTPS communicates over port 443 by default.
- Fig. 10.4.2 shows secure HTTP transactions.

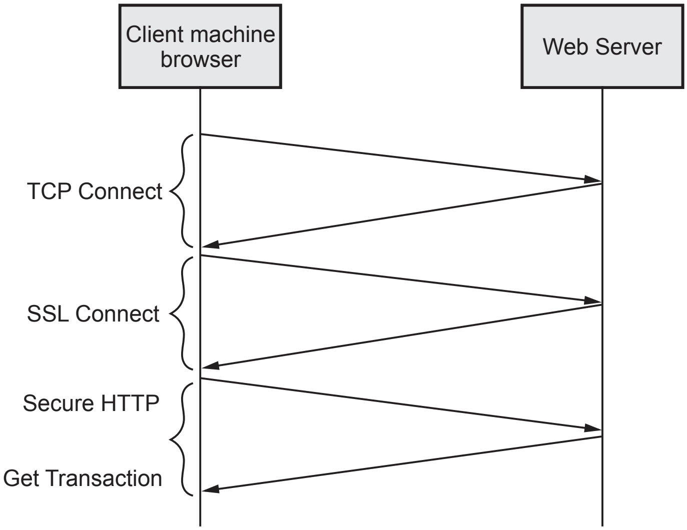


Fig. 10.4.2 HTTPS transaction

- When a client makes a request over HTTPS, it first tries to locate a certificate on the server. If the cert is found, it attempts to verify it against its known list of certificate authorities (CA). If it is not one of the listed CAs, it might show a dialog to the user warning about the website's certificate. Once the certificate is verified, the SSL handshake is complete and secure transmission is in effect.
- Benefits of a HTTPS certificate are :
  1. Customer information like credit card numbers, bank account numbers is encrypted and cannot be intercepted
  2. Visitors can verify
  3. Customers are more likely to trust and complete purchases from sites that use HTTPS
- S-HTTP provides a wide variety of mechanisms to provide for confidentiality, authentication and integrity. It also includes header definitions to provide key transfer, certificate transfer and similar administrative functions.
- S-HTTP does not use any a particular key certification scheme. It includes support for RSA, in-band, out-of-band and Kerberos key exchange. Like SSL, client public keys are not required.

University Questions

1. *Explain HTTPS in brief.* **GTU : Winter-17, Marks 3**
2. *What is the main difference HTTP and HTTP's protocol. When HTTP;s is used, which elements of the communication are encrypted ?* **GTU : Winter-19, Marks 4**

10.5 SSH

- Secure Shell (SSH) is a protocol for secure remote login and other secure network services over an insecure network. Secure Shell is a powerful, software-based approach to network security that provides a secure channel for data transmission through a network.
- It Supports secure remote logins, secure remote command execution, secure file transfers. SH connections provide highly secure authentication, encryption, and data integrity to combat password theft and other security threats.
- Features of SSH :

1. Privacy : via strong end-to-end encryption- DES, IDEA, Blowfish
2. Integrity : via 32 bit Cyclic Redundancy Check (CRC-32)
3. Authentication : server via server's host key, client usually via password or public key

4. Authorization : controlled at a server wide level or per account basis
5. Forwarding : encapsulating another TCP based service such as Telnet within an SSH session

- Secure Shell client and server applications are widely available for most popular operating systems. Secure Shell provides three main capabilities:

1. Secure command-shell
2. Secure file transfer
3. Port forwarding

- Fig 10.5.1 shows SSH protocol stack. SSH is organized as three protocols that typically run on top of TCP.

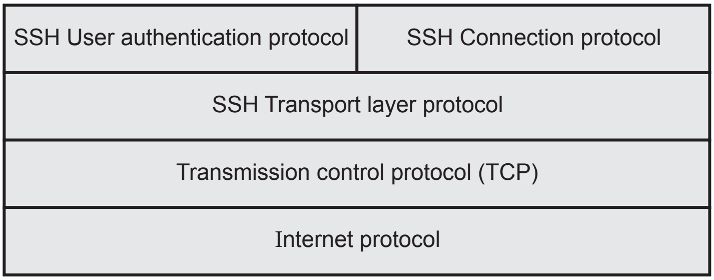

SSH User authentication protocol
SSH Connection protocol
SSH Transport layer protocol
Transmission control protocol (TCP)
Internet protocol

Fig. 10.5.1 SSH protocol stack

Functions of SSH protocol stack

- Transport Layer Protocol provides server authentication, data confidentiality, and data integrity with forward secrecy. The transport layer may optionally provide compression.
- User Authentication Protocol: Authenticates the user to the server.
- Connection Protocol: Multiplexes multiple logical communications channels over a single, underlying SSH connection.

SSH Transport layer protocol

- Server authentication is based on the server's public/private key pair.
  1. Host Keys : one host may have many, or many hosts could share one
  2. Client must have the server's public key in advance.
- Two alternative trust models defined in RFC4251.
  1. The client has a local DB associates each host name with public key.
  2. The host name to key association is certified by CA. The client only knows CA's public key and can verify all host keys certified by CA.

Reasons to use SSH

1. Designed to be a secure replacement for rsh, rlogin, rcp, rdist, and telnet.

2. Strong authentication. Closes several security holes (e.g., IP, routing, and DNS spoofing).
3. Improved privacy. All communications are automatically and transparently encrypted.
4. Arbitrary TCP/IP ports can be redirected through the encrypted channel in both directions
5. The software can be installed and used (with restricted functionality) even without root privileges.
6. Optional compression of all data with gzip (including forwarded X11 and TCP/IP port data), which may result in significant speedups on slow connections.

Components of Secure Shell

1. SSHD Server : A program that allows incoming SSH connections to a machine, handling authentication, authorization.
2. Clients : A program that connects to SSH servers and makes requests for service
3. Session : An ongoing connection between a client and a server. It begins after the client successfully authenticates to a server and ends when the connection terminates.

- When SSHD is started, it starts listening on port22 for a socket.
- When a socket get connected the secure shell daemon spawns a child process. Which in turn generates an host key e g. RSA.
- After key is generated the secure shell daemon is ready for the local client to connect to another secure shell daemon or waits for a connection from remote host.

SSH Package exchange

- Fig. 10.5.2 show the sequence of events in the SSH Transport Layer Protocol.
- The client initiates the connection by sending a request to the TCP port of the SSH server.
- Server reveals it's SSH protocol version to the client.
- If the client and server decide their versions are compatible, the connection proceeds.

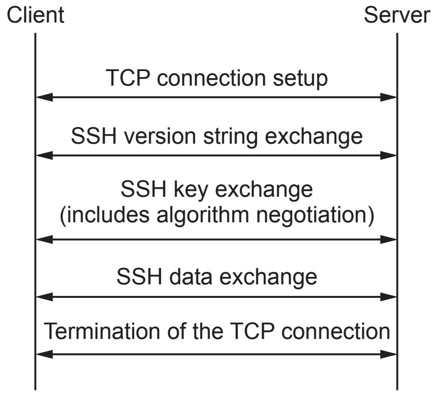

Client
Server
TCP connection setup
SSH version string exchange
SSH key exchange
(includes algorithm negotiation)
SSH data exchange
Termination of the TCP connection

Fig. 10.5.2 Sequence of events in the SSH Transport Layer Protocol

- SSH server sends the information to the client - host key, the server key, a list of supported encryption, compression and authentication methods, and a sequence of eight random bytes.
- Client checks identity of server by using the host key against known hosts database.
- Client generates a session key and double encrypts it using the host key & server key.
- Client sends encrypted session key along with check bytes and acceptable algorithm.

Authentication

- Server then decrypts the encrypted session key it received.
- Server sends a confirmation encrypted with this session key.
- Client receives confirmation, confirms server authentication.
- Client Authentication usually either by Password Authentication or Public key Authentication.
- Server confirms client authorization.
- Generates a 256 bit random challenge, encrypts it with clients public key, and sends to client.
- Client decrypts challenge, generates a hash value with a session identifier (commonly generated random string at beginning of session), and sends to server.
- Server generates hash, if both match, session is authenticated.
- Fig. 10.5.3 shows binary packet protocol.
  1. Packet length : length of the packet not including the MAC and the packet length field.
  2. Padding length : length of padding.
  3. Payload : it is useful contents and might be compressed. Maximum payload size is 32768.
  4. Random padding : it is 4 - 255 bytes. The total length of packet not including the MAC must be multiple of max(8, cipher block size) even if a stream cipher is used.
  5. MAC : it is computed over the clear packet and an implicit sequence number.

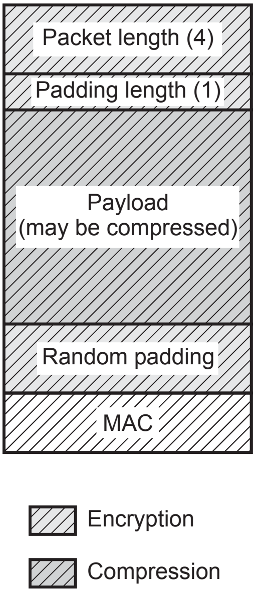


Fig. 10.5.3 Binary packet protocol

University Questions

1. *Explain HTTPS and SSH.* **GTU : Summer-18, Marks 7**
2. *For what purpose Secure Shell (SSH) is useful? Briefly define SSH protocol.* **GTU : Summer-19, Marks 3**

10.6 Short Questions and Answers

Q.1 What is SSH ?

Ans. : SSH is a protocol for secure remote login and other secure network services over an insecure network

Q.2 What is SSL session ?

Ans. : 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.

Q.3 Which of two services provided by SSL Record Protocol tor SSL connections ?

Ans. : 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 MAC.

Q.4 What is the purpose of HTTPS ?

Ans. : HTTPS refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server.

Q.5 What is the difference between an SSL connection and an SSL session ?

Ans. : A connection is a transport that provides a suitable type of service. For SSL, such connections are peer-to-per relationships. The conections are transient. 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.

10.7 Multiple Choice Questions

Q.1 SSL is designed to make use of ------ to provide a reliable end-to-end secure service.

a UDP b TCP c IP d HTTP

Q.2 HTTS uses port number __________

a 25 b 80 c 443 d 1024

Q.3 SSH stand for ______

a Secure socket shell

b secure socket layer

c secure session handshake

d Secure shell

Q.4 SSH connection protocol runs on the top of SSH ______.

a Internet protocol

b transport layer protocol

c application layer protocol

d None of these

Q.5 The user initiates an SSH connection. SSH attempts to connect to port ______ on the remote host.

a 22 b 25 c 80 d 110

Q.6 Which of the following three users authentication method supported by SSH.

a Private key, password, hostbased

b Public key, login, password

c Public key, password, hostbased

d Private key, password, serverbased

Answer Keys for Multiple Choice Questions

Q.1	b	**Q.2**	c	**Q.3**	d
Q.4	b	**Q.5**	a	**Q.6**	c

❑❑❑

Notes

SOLVED MODEL QUESTION PAPER

(As Per 2018 Pattern)

Cryptography and Network Security

Semester - VI (CSE / IT)

Time : $2\frac{1}{2}$ Hours] **[Total Marks : 70**

Instructions

1) Attempt all questions.

2) Make suitable assumptions wherever necessary.

3) Figures to the right indicate full marks.

Q.1 a) *What is difference between passive and active attack ?* **[Refer section 1.5.3]** **[3]**

b) *Describe monoalphabetic cipher.* **[Refer section 1.14.2]** **[4]**

c) *What is security services ? Explain various security services.* **[Refer section 1.3]** **[7]**

Q.2 a) *What is AES ? List its advantages.* **[Refer section 2.7]** **[3]**

b) *What is block cipher ? Differentiate block cipher and stream cipher.* **[Refer section 2.2]** **[4]**

c) *Discuss electronic code book and cipher feedback mode with diagram.* **[Refer section 3.3]** **[7]**

OR

c) *Explain double DES and triple DES.* **[Refer sections 3.1 and 3.2]** **[7]**

Q.3 a) *For what purpose Secure Shell (SSH) is useful ? Briefly define SSH protocol.* **[Refer section 10.5]** **[3]**

b) *What is the main difference between HTTP and HTTPS protocol. When HTTPS is used, which elements of the communication are encrypted ?* **[Refer section 10.4]** **[4]**

c) *What is hash function ? List the requirement of hash function. Also explain one-way hash function.* **[Refer section 5.1]** **[7]**

OR

Q.3 a) *What is the role of a compression function in a hash function ?* **[Refer Q.7 of section 6.4]** **[3]**

b) *Explain various public key distribution techniques.* **[Refer section 8.1]** **[4]**

c) *Describe MAC with it's security implications.* **[Refer section 6.1]** **[7]**

Q.4 a) *What is KDC ? List the duties of a KDC.* **[Refer section 8.1]** **[3]**

b) *Explain Schnorr digital signature scheme.* **[Refer section 7.4]** **[4]**

c) *Explain HTTPS and SSH.* **[Refer section 10.5]** **[7]**

OR

Q.4 a) *List three approaches to secure user authentication in a distributed environment.* **[Refer section 9.1]** **[3]**

b) *What problem was Kerberos designed to address ? What are the three threats associated with user authentication over a network or internet.* **[Refer section 9.4]** **[4]**

c) *Explain about the RSA algorithm with an example as p=11, q=5, e=3 and plaintext = 9* **[Refer section 4.2 and example 4.2.5]** **[7]**

Q.5 a) *Explain the concept of Realm in Kerberos in brief.* **[Refer section 9.4]** **[3]**

b) *Explain MD5 algorithm.* **[Refer section 5.6]** **[4]**

c) *What is the principle of Digital Signature Algorithm (DSA) ? How a user can create a signature using DSA ? Explain the signing and verifying function in DSA.* **[Refer section 7.1]** **[7]**

OR

Q.5 a) *List the requirement of public key cryptography.* **[Refer section 4.1]** **[3]**

b) *Briefly discuss the working of SSL record protocol.* **[Refer section 10.2]** **[4]**

c) *Explain use of public-key certificate with diagram and draw X.509 certificate format.* **[Refer section 8.2]** **[7]**

❑❑❑

Notes

Notes