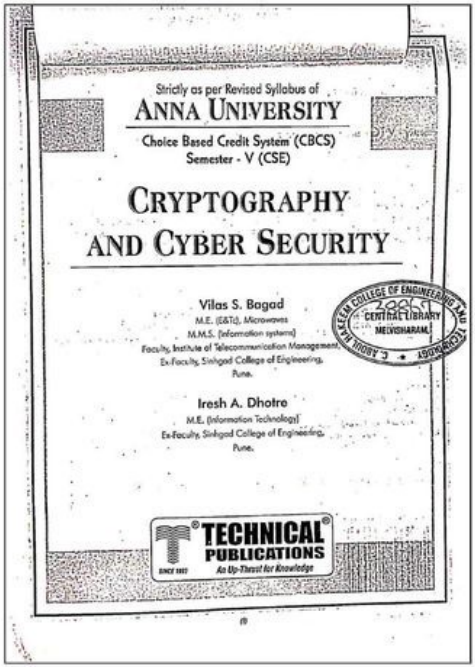
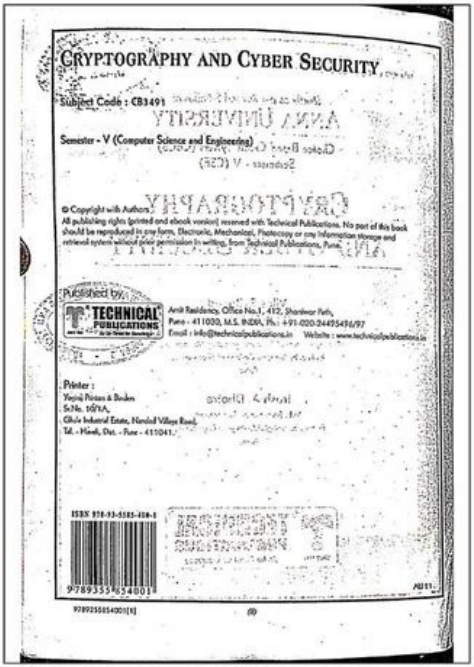

Strictly as per Revised Syllabus of
ANNA UNIVERSITY
Choice Based Credit System (CBCS)
Semester - V (CSE)
CRYPTOGRAPHY
AND CYBER SECURITY
Vilas S. Bagad
M.E. (E&TC), Microwaves
M.M.S. (Information systems)
Faculty, Institute of Telecommunication Management,
Ex-Faculty, Sinhgad College of Engineering,
Pune.
Iresh A. Dhotre
M.E. (Information Technology)
Ex-Faculty, Sinhgad College of Engineering,
Pune.
TECHNICAL PUBLICATIONS
An Up-Thrust for Knowledge
SINCE 1993
CENTRAL LIBRARY
MEVISHARAM


CRYPTOGRAPHY AND CYBER SECURITY
Subject Code : CB3491
Semester - V (Computer Science and Engineering)
© Copyright with Authors
All publishing rights (printed and ebook version) reserved with Technical Publications. No part of this book should be reproduced in any form, Electronic, Mechanical, Photocopy or any information storage and retrieval system without prior permission in writing, from Technical Publications, Pune.
Published by:
TECHNICAL PUBLICATIONS
Amit Residency, Office No.1, 412, Shaniwar Peth,
Pune - 411030, M.S. INDIA, Ph. : +91-020-24495496/97
Email : info@technicalpublications.in
Website : www.technicalpublications.in
Printer :
Yogiraj Printers & Binders
Sr.No. 10/1A,
Ghule Industrial Estate, Nanded Village Road,
Tal. - Haveli, Dist. - Pune - 411041.
ISBN 978-93-5585-408-1
9789355854081

SYLLABUS

Cryptography and Cyber Security - (CB3491)

UNIT I INTRODUCTION TO SECURITY

Computer Security Concepts - The OSI Security Architecture - Security Attacks - Security Services and Mechanisms - A Model for Network Security - Classical encryption techniques : Substitution techniques, Transposition techniques, Steganography - Foundations of modern cryptography : Perfect security - Information Theory - Product Cryptosystem - Cryptanalysis. (Chapter - 1)

UNIT II SYMMETRIC CIPHERS

Number theory - Algebraic Structures - Modular Arithmetic - Euclid's algorithm - Congruence and matrices - Group, Rings, Fields, Finite Fields SYMMETRIC KEY CIPHERS : SDES - Block Ciphers - DES, Strength of DES - Differential and linear cryptanalysis - Block cipher design principles - Block cipher mode of operation - Evaluation criteria for AES - Pseudorandom Number Generators - RC4 - Key distribution. (Chapter - 2)

UNIT III ASYMMETRIC CRYPTOGRAPHY

MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY: Primes - Primality Testing - Factorization - Euler's totient function, Fermat's and Euler's Theorem - Chinese Remainder Theorem - Exponentiation and logarithm.

ASYMMETRIC KEY CIPHERS : RSA cryptosystem - Key distribution - Key management - Diffie Hellman key exchange - Elliptic curve arithmetic - Elliptic curve cryptography. (Chapter - 3)

UNIT IV INTEGRITY AND AUTHENTICATION ALGORITHMS

Authentication requirement - Authentication function - MAC - Hash function - Security of hash function - HMAC, CMAC - SHA - Digital signature and authentication protocols - DSS - Schnorr Digital Signature Scheme - ElGamal cryptosystem - Entity Authentication : Biometrics, Passwords, Challenge Response protocols - Authentication applications - Kerberos.

MUTUAL TRUST : Key management and distribution - Symmetric key distribution using symmetric and asymmetric encryption - Distribution of public keys - X.509 Certificates. (Chapter - 4)

UNIT V CYBER CRIMES AND CYBER SECURITY

Cyber Crime and Information Security - classifications of Cyber Crimes - Tools and Methods - Password Cracking, Keyloggers, Spywares, SQL Injection - Network Access Control - Cloud Security - Web Security - Wireless Security. (Chapter - 5)

PREFACE

The importance of **Cryptography and Cyber 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 Cyber 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 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
D. A. Dhotre

Dedicated to God.

TABLE OF CONTENTS

UNIT I

UNIT II

UNIT III

UNIT IV

UNIT V

UNIT I

1 Introduction to Security

Syllabus

Computer Security Concepts - The OSI Security Architecture - Security Attacks - Security Services and Mechanisms - A Model for Network Security - Classical encryption techniques: Substitution techniques, Transposition techniques, Steganography - Foundations of modern cryptography : Perfect security - Information Theory - Product Cryptosystem - Cryptanalysis.

Contents

1.1 Computer Security Concepts

AU : Dec.-20

- The history of information security begins with computer security.
- Network security, to protect networking components, connections and contents.
- Information security to protect the confidentiality, integrity and availability of information assets, whether in storage, processing or transmission.
- Physical security consists of all mechanisms used to ensure that physical access to the computer systems and networks is restricted to only authorize users.
- 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.
- 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 unauthorized 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.

1.1.1 Basic Terminologies in Security

- 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 straightforward 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.

1.1.2 Categories

- Various categories of computer security are :
  1. Cryptography 2. Data security
  3. Computer security 4. Network security
- Cryptography is data encryption and decryption.
- Data security is ensuring safe data from modification and corruption.
- Computer security is formal description of security policies. It includes protection, preventation and detection of unauthorized use of computer.
- Network security is protection of data on the network during transmission or sharing.

1.1.3 Techniques

- Commonly used security techniques are as follows :
  1. Encryption : Used to protect information and data. It is cryptography techniques. Different types of encryption are used for providing security.
  2. Access control : Access to data or computer is controlled by using some mechanism. Access control is a security technique that regulates who or what can view or use resources in a computing environment. It is a fundamental concept in security that minimizes risk to the business or organization.
  3. Data backup : Data backup refers to saving additional copies of your data in separate physical or virtual locations from data files in storage. If you lose your data, recovery could be slow, costly or impossible. It is important that you secure, store and backup your data on a regular basis.
  4. Firewall : Firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.
  5. Antivirus software : Many antivirus software programs include real-time threat detection and protection to guard against potential vulnerabilities as they happen, as well as system scans that monitor device and system files looking for possible risks.
  6. Intrusion detection systems : IDS can offer protection from external users and internal attackers. It also automatically monitors the Internet to search for any of the latest threats which could result in a future attack.
  7. Series of confidence : It ensure that all software use has been authentic.

1.1.4 Elements of Information Security

- 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.

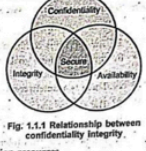


Fig. 1.1.1 Relationship between confidentiality Integrity

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.

1.1.5 Threats and Vulnerability

Threat

- The term "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.
- Penetration testing exercises are substantially focused on assessing threat profiles, to help one develop effective countermeasures against the types of attacks represented by a given threat. 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.

Vulnerability

- The term "vulnerability" refers to the security flaws in a system that allows an attack to be successful.
- Vulnerability testing should be performed on an ongoing basis by the parties responsible for resolving such vulnerabilities and helps to provide data used to identify unexpected dangers to security that need to be addressed.
- Such vulnerabilities are not particular to technology - they can also apply to social factors such as individual authentication and authorization policies.
- Testing for vulnerabilities is useful for maintaining ongoing security, allowing the people responsible for the security of one's resources to respond effectively to new dangers as they arise. It is also invaluable for policy and technology development.

and as part of a technology selection process; selecting the right technology early on can ensure significant savings in time, money and other business costs further down the line.

- Understanding the proper use of such terms is important not only to sound like you know what you're talking about, nor even just to facilitate communication. It also helps develop and employ good policies.
- The specificity of technical jargon reflects the way experts have identified clear distinctions between practical realities of their fields of expertise and can help clarify even for oneself how one should address the challenges that arise.
- Other examples of vulnerability include these :
  1. A weakness in a firewall that lets hackers get into a computer network.
  2. Unlocked doors at businesses.
  3. Lack of security cameras.

Review Question

1. Discuss examples from real life, where the following security objectives are needed :
i) Confidentiality
ii) Integrity
iii) Non-repudiation
Suggest suitable security mechanisms to achieve them. **AU : Dec.-20, Marks 5 + 5 + 5**

1.2 The OSI Security Architecture

AU : Dec.-20

- The international telecommunication union telecommunication standardization sector recommendation X.800 security architecture for OSI. It is useful to managers as a way of organizing the task of providing security.
- To assess effectively the security needs of an organization and to evaluate and choose various security products and policies, we need some systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. The OSI security architecture focuses on three essential parts : Security attacks, security mechanisms and security services.
- It focuses on security attacks, mechanisms and services. These can be defined below :
  1. **Security attack** : Any action that compromises the security of information owned by an organization.
  2. **Security mechanism** : A process that is designed to detect, prevent or recover from a security attack.

3. **Security service** : A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization.

1.2.1 Vulnerabilities in OSI Model

- How does the security framework work ? The security architecture is mapped to the customer's enterprise architecture using the Open Systems Interconnect (OSI) networking model. The security framework has security solutions for all the pieces of the enterprise infrastructure that supports the goals of the organization.
- The security framework operates and protects that infrastructure at each of the operational levels of the OSI model. As transactions take place from end-to-end of the enterprise architecture, these transactions utilize technologies that operates at all the levels of the OSI model as well. Since security extends into policies and procedures and supports business driven goals, the security framework has added two additional layers to the model, the financial and political layers. These layers began as a tongue-in-cheek joke at the National Security Agency in the mid-nineties.
- However, security of information systems really does have to match the budget and the business objectives of an organization and these layers have achieved legitimacy in their own right.

1. Physical Layer

- Layer one of the OSI model is the physical layer where the wire over which electronic impulses run to create the magic. At this layer, the security framework protects the cable plant, the wiring and telecommunications infrastructure.
- The physical layer is protected by redundant power and WAN connection. It also means protecting the physical hardware in network closets, server farms and systems in raised floor spaces. Protecting the physical layer entails locks, alarms on entrances, climate controls and access to data centers.

2. Data Link and Network Layers

- At the data link and network layers, the security framework protects systems with a number of technologies. VPNs protect information by encrypting it and sending it through encrypted tunnels through networks or the internet. Network intrusion detection systems or NIDS watch traffic flowing over the wires looking for bit stream patterns that could indicate attacks or malicious intent.
- Host intrusion detection systems monitor bit streams entering the host machines at the Network Interface Card (NIC) level, also looking for suspicious patterns. Virus scanning at this level looks for patterns that indicate malicious code that fits signatures for known viruses.

3. Network and Transport Layers

- At the network and transport layer, the security framework uses firewalls to do stateful inspection of packets entering and leaving the network. Routers, using Access Control Lists or ACLs filter IP packets, preventing traffic from going to systems that have no need for it.
- Utilizing IP address schemes, network engineers can plan and implement routing tables that protect networks with router ACLs, making firewall rules easier to write and deploy and thwarting attacks such as address spoofing. At the network and transport layers, virus scanning software opens attachments in messaging packets such as e-mail, looking for embedded viruses or malicious code.

4. Session, Presentation and Application Layers

- At the session, presentation and application layers, the security framework uses a number of techniques and tools to secure systems. Some of these techniques are policies for system management such as hardening the operating systems, keeping patch levels and OS revisions up to date, running with only the services needed to support the business processes and turning off all other process, running processes with limited system privileges, etc.
- All of these management techniques contribute to security at the session, presentation and application layer and are the kind of system controls that automatically enforce security policies.

5. Presentation and Application Layers

- At the presentation and application layers, the security framework utilizes user account management to control access to the network, systems and applications. The security framework relies on system managers to control access to their machines, network administrators to manage user access to their networks and application managers such as Data Base Administrators (DBA) to control access to applications and data.
- At this level, the security framework includes virus scanning applications to scan hard drives and system memory for malicious code, updating scan engines and virus signatures. Host Intrusion Detection Systems (HIDS), active in the lower levels of the model, work at the presentation and application layer to watch for changes to critical system files and other system behaviour that might indicate an attacker trying to gain control of the system.
- The security framework can also control user access centrally using a Role/Rule-Based Access Control (RBAC) engine, that uses a directory such as LDAP or Active Directory that contains information about users and the systems and resources to which the users are authorized access. PKI and digital certificates

can be used at this level to provide digital signatures, encryption and non-repudiation at the application level.

Review Question

1. *Explain OSI security architecture model with neat diagram.* **AU : Dec.-20, Marks 7**

1.3 Security Attacks

AU : Dec.-13,19

- An attempt to gain unauthorized access to information resource or services, or to cause harm or damage to information systems.
- Security attacks are of two types : Passive attack and active attack

Fig. 1.3.1

1.3.1 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. 1.3.2. 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.

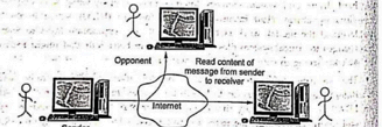


Fig. 1.3.2 Release of message contents

- **Traffic analysis :** Mask the contents of message so that opponents could not extract the information from the message. Encryption is used for masking. Fig. 1.3.3 shows the 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.

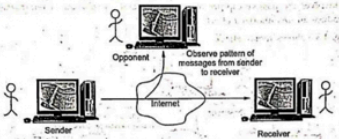


Fig. 1.3.3 Traffic analysis

1.3.2 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. 1.3.4 shows masquerade.

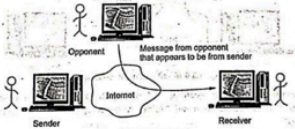


Fig. 1.3.4 Masquerade

- **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. 1.3.5 shows replay attack.

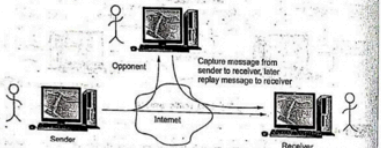


Fig. 1.3.5 Replay

3. Modification of message

- It involves some change to the original message. It produces an unauthorized effect. Fig. 1.3.6 shows the modification of message.

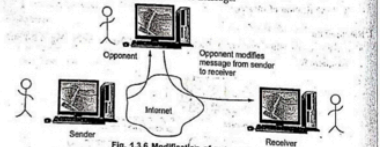


Fig. 1.3.6 Modification of message

- 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. 1.3.7 shows denial of service attack.

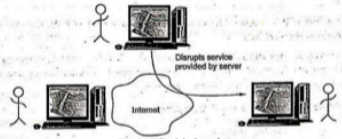


Fig. 1.3.7 Denial of service

- 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. 1.3.8 shows the SYN flood DOS attack.

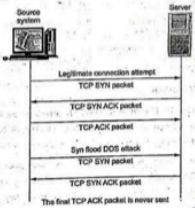


Fig. 1.3.8 SYN flood DOS attack

- 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.

1.3.2.1 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.

1.3.3 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, messages 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 messages, 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.

Review Questions

1. *What are the different types of attacks ? Explain.* **AU : Dec.-13. Marks 8**
2. *Write a note on different types of security attacks and services in detail.* **AU : Dec.-19. Marks 13**

1.4 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 authentication 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.

1.5 Security Mechanism

AU : Dec.-20

- A mechanism that is designed to detect, prevent, or recover from a security attack.
- Security mechanisms are technical tools and techniques that are used to implement security services. A mechanism might operate by itself, or with others, to provide a particular service.
- Security mechanisms defined by X.800 are given below :

Security Mechanism

- X.800 defined security mechanisms as follows :

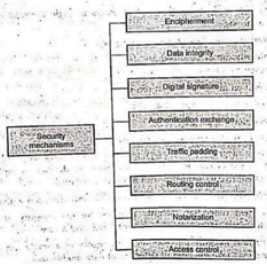


Fig. 1.5.1 Security mechanisms

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.

Review Question

1. *Describe the various security mechanisms.* **AU : Dec.-20, Marks 8**

1.6 A Model for Network Security

AU : May-19, Dec.-22

- 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. 1.6.1 shows the network security model.

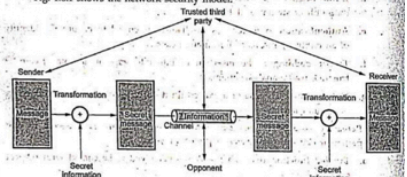


Fig. 1.6.1 Network security model

- 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.

Review Question

1. *Explain the network security model and its important parameters with a neat block diagram.* **AU : May-19, Dec.-22, Marks 13**

1.7 Cryptography

- Cryptography is the science of writing in secret code and is an ancient art. Cryptography is not only protects data from theft or alteration, but can also be used for user authentication.
- The term is derived from the Greek word kryptos, which means hidden.
- In cryptography, we start with the unencrypted data, referred to as plaintext. Plaintext is encrypted into ciphertext, which will in turn (usually) be decrypted back into usable plaintext.
- Fig 1.7.1 shows cryptography.

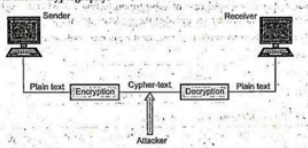


Fig. 1.7.1 Cryptography

- Cryptography provides secure communication in the presence of malicious third parties.
- Encryption is the process of encoding a plain text message into non-readable form. Decryption is a process of transferring an encrypted message back into its normal form.
- Algorithms are considered secure if an attacker cannot determine any properties of the plaintext or key, given the ciphertext.
- An attacker should not be able to determine anything about a key given a large number of plaintext/ciphertext combinations which used the key.

Advantages of Cryptography

1. It provides security to on line network communication.
2. It provides security to email, credit/debit card information etc.
3. Cryptography hides the contents of a secret message from a malicious people.
4. Cryptography can also provide authentication for verifying the identity of someone or something.

1.8 Classical Encryption Techniques : Substitution Techniques

AU : Dec.-14,15,17,20,21. May-15,16,17,18,19

- 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.

1.8.1 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
Cipher	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U

Plain	s	t	u	v	w	x	y	z
Cipher	V	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.

1.8.2 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	c	z	a	s	d	f	g	h		

Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext				o									q

For example

Plaintext message : hello how are you

Ciphertext message : aeggk 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.

1.8.3 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	Q	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.

1.8.4 Hill Cipher

- The encryption algorithm takes m successive plaintext letters and substitutes for them m ciphertext letters.
- The substitution is determined by m linear equations in which each character is assigned a numerical value (a = 0, b = 1, c = 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

Cipher C = E(K, P) = KP mod 26

Plaintext P = D(K, P) = K^{-1} C mod 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.

1.8.5 Polyalphabetic Substitution

- 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. 1.8.1 shows a tableall or table to implement this cipher efficiently.

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

Key

Fig. 1.8.1

- **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.

1.8.6 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	t	h	e	y
	2	14	12	4	19	14	3	0	24	
Key	N	C	B	U	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.
  3. Only possible attack to such a cipher is a brute force attack.

1.8.7 Feistel Cipher

- Fig. 1.8.2 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).

(See Fig. 1.8.2 on next page)

Parameters and design features

Following parameters are considered :

1. Block size
2. Key size

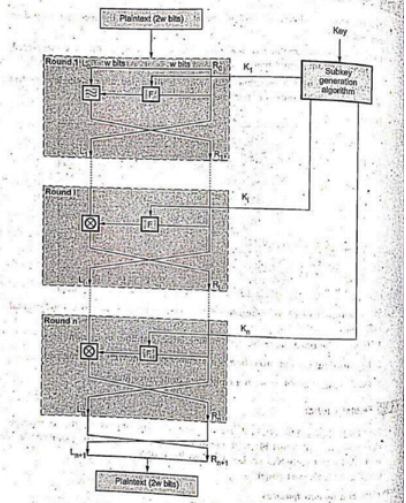


Fig. 1.8.2 Classical feistel network

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.
- 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

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

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

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

$\therefore$ 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.

1.8.8 Comparison between Monoalphabetic and Polyalphabetic Cipher

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 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 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 1.8.1 *Encrypt the message "PAY" using Hill cipher with the following key matrix and show the decryption to get the original plain text.*

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

AU : Dec.-14, Marks 16

Solution :

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

The letters PAY of the plaintext are represented by the vector :

$$\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \text{LNS}$$

Ciphertext = LNS

Example 1.8.2 *Encrypt the following using play fair cipher using the keyword MONARCHY. "SWARAJ IS MY BIRTH RIGHT". Use X for blank spaces.*

AU : Dec.-17, Marks 16

Solution :

Key Square :

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

Plain text : SWARAJ IS MY BIRTH RIGHT

SW AR AJ IS MY BI RT HR IG HT

Cipher Text /Encryption Result = QX RM BS XA NC SX ZR OD KE DP

Example 1.8.3 *Convert "COMPUTER SECUIRTY" using Caesar cipher.*

Solution : Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.

Plain text : COMPUTER SECUIRTY

Cipher Text : FRPSXWHU VHFXLUWB

Example 1.8.4 *Convert plain text into cipher text by using simple columnar techniques of the following sentence: 'ALL IS WELL FOR YOUR EXAM'.*

Solution : Consider the six columns. Therefore, we write the message in the rectangle row-by-row :

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
A	L	L	I	S	W
E	L	F	O	R	Y
O	U	R	E	X	A
M					

- Now let us decide the order of columns at some random order, say 5, 2, 4, 1, 6, 3. Then read the text in the order of these columns.

Ciphertext = SOELLOMIFRAEYAWRXLLU

Example 1.8.5 *Find out cipher text using Playfair cipher for following given plain text and key.*

Key = GOVERNMENT, Plain text = PLAYFAIR

Solution :

Key = GOVERNMENT

Plain text = PLAYFAIR

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

Plain text = PLAYFAIR ⇒ PL AY FA IR

Cipher text = PQHETHBZ

Example 1.8.6 *Find out cipher text using playfair cipher for following given plain text and key.*

Key = ENGINEERING, Plain text = COMPUTER

Solution :

Key = ENGINEERING

Plain text = COMPUTER

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

Plain text = COMPUTER ⇒ CO MP UT ER

Cipher text = FLTHUPEN

Example 1.8.7 *Consider the following :*

Plaintext : "PROTOCOL" Secret key : "NETWORK"

What is the corresponding cipher text using play fair cipher method ?

AU : May-16, Marks 8

Solution. : Corresponding cipher text using play fair cipher method :

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

Plaintext : PR OT OC OL

Ciphertext : LA NW NR NS

Example 1.8.8 *Using Playfair cipher encrypt message. "We live in a world full of beauty" use key "ANOTHER".* **AU : May-17, Marks 5**

Solution : Example :

Plaintext : We live in a world full of beauty.

Keyword : Another

Step 1 : Preparing plain text

The plain text matrix is :

we	li	ve	in	aw
or	ld	fu	lx	lo
fb	ea	ut	yz	

Step 2 : Preparing key matrix

The key matrix is :

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

Step 3 : Encryption

By following the above rules for encryption of plain text the cipher text is :

VRFKAFGONVNBULLMIZIHIEFESHZY

Example 1.8.9 *Use polyalphabetic ciphers to encrypt plain text "SHE IS VERY HAPPY AND BEAUTIFUL GIRL" use key 'ANOTHER'.*

Solution. :

Keyword	anoth	erano	thera	nothe	ranot	heran
Plaintext	sheis	veryh	appya	ndbea	utifu	lgirl
Ciphertext	SUSBZ	ZVRLV	TWTPA	ARULE	LTVTN	SKZRY

Example 1.8.10 *Use play fair cipher to encrypt the following message "This is a columnar transposition" use key - APPLE.*

Solution : Message = This is a columnar transportation

Key = APPLE

Encryption :

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

Message = Th is is ac ol um na rt ra ns po si ti on

Ciphertext = UG MQ MQ BH MB SO IE SU MT BK QM NQ KN

Example 1.8.11 *Encrypt the plain text 'COE' using hill cipher, use keyword 'ANOTHERBZ'.*

Solution. : Plain text = COE

Key = ANOTHERBZ

For plaintext COE, here C = 2, O = 14, E = 4

$$\text{Therefore, } P = \begin{pmatrix} 2 \\ 14 \\ 4 \end{pmatrix}$$

For key ANOTHERBZ the numbers are 0, 13, 14, 19, 6, 4, 17, 1, 25

The numbers in the matrix form :

$$K = \begin{pmatrix} 0 & 13 & 14 \\ 19 & 6 & 4 \\ 17 & 1 & 25 \end{pmatrix}$$

Ciphertext = (Key × Plaintext) Mod 26

Encryption is as follows :

$$C = \begin{pmatrix} 0 & 13 & 14 \\ 19 & 6 & 4 \\ 17 & 1 & 25 \end{pmatrix} \begin{pmatrix} 2 \\ 14 \\ 4 \end{pmatrix} \text{mod } 26 = \begin{pmatrix} 238 \\ 138 \\ 148 \end{pmatrix} \text{mod } 26 = \begin{pmatrix} 4 \\ 8 \\ 18 \end{pmatrix}$$

Ciphertext : 4 = E, 8 = I and 18 = S

Ciphertext = EIS

Example 1.8.12 *Encrypt the message "PAY" using Hill cipher with the following key matrix and show the decryption to get the original plaintext.*

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

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

The letters PAY of the plaintext are represented by the vector :

$$\begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} = \begin{bmatrix} 375 \\ 819 \\ 486 \end{bmatrix} \bmod 26 = \begin{bmatrix} 11 \\ 13 \\ 18 \end{bmatrix} = \text{LNS}$$

Ciphertext = LNS

Example 1.8.13 *Using hill cipher encrypt the message 'ESSENTIAL'. The key for encryption is 'ANOTHERBZ'.*

Solution :

$$\text{Key matrix } K = \begin{bmatrix} 0 & 13 & 14 \\ 19 & 6 & 4 \\ 17 & 1 & 25 \end{bmatrix}$$

$$\text{Plaintext matrix } P = \begin{bmatrix} 4 & 4 & 8 \\ 18 & 13 & 0 \\ 18 & 19 & 11 \end{bmatrix}$$

Ciphertext matrix $C = K \times P \bmod 26$

$$C = \begin{bmatrix} 0 & 13 & 14 \\ 19 & 6 & 4 \\ 17 & 1 & 25 \end{bmatrix} \begin{bmatrix} 4 & 4 & 8 \\ 18 & 13 & 0 \\ 18 & 19 & 11 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 486 & 435 & 154 \\ 256 & 230 & 196 \\ 536 & 556 & 411 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 & 19 & 24 \\ 22 & 22 & 14 \\ 16 & 10 & 21 \end{bmatrix} \bmod 26$$

$$C = \begin{bmatrix} S & T & Y \\ W & W & D \\ Q & K & V \end{bmatrix} \bmod 26$$

Ciphertext = SWQTWKYDV

Example 1.8.14 *Use transposition cipher to encrypt plain text 'I Love my India' and use the key 'HEAVEN'. [Use single columnar transposition]*

Solution :

KEY	→	H	E	A	V	E	N
Key number	→	4	2	1	6	3	5
Plaintext	→	I	L	O	V	E	M
		Y	I	N	D	I	A
Arrange the key number as per ascending order							
KEY	→	A	E	E	H	N	V
Key number	→	1	2	3	4	5	6
Plaintext	→	O	L	E	I	M	V
		N	I	I	Y	A	D

Ciphertext = ONLIEIIYMAVD

Example 1.8.15 *Encrypt the message "THIS IS AN EXERCISE" using playfair cipher with key = DOLLARS.*

Key = DOLLARS

Message = THIS IS AN EXERCISE

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

To encipher a message, divide it into pairs of letters : TH IS IS AN EX ER CI SE

Ciphertext = PKGC GC DT CYFAIQBF

Review Questions

1. *Explain the substitution encryption techniques in detail.* **AU : May-15, Marks 16**
2. *Illustrate the hill cipher technique with an example.* **AU : Dec.-15, Marks 16**
3. *Describe : (i) Playfair cipher (ii) Vigenere cipher.* **AU : May-17, Marks 16**
4. *Explain classical encryption techniques with symmetric cipher and hill cipher model.* **AU : May-18, Marks 16**
5. *What is monoalphabetic cipher ? Examine how it differs from ceasar cipher.* **AU : May-19, Marks 6, Dec.-20, Marks 7**
6. *Encrypt the message "this is an exercise", using additive cipher with key = 20. Ignore the space between words. Decrypt the message to get the original plaintext.* **AU : Dec.-20, Marks 7**
7. *Let message = "graduate", Key = "word", find ciphertext using playfair cipher.* **AU : Dec.-21, Marks 8**
8. *Demonstrate encryption and decryption process in hill cipher. Consider m = "sh" and key = hill".* **AU : Dec.-21, Marks 4 + 9**

1.9 Transposition Techniques **AU : Dec.-22**

- 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
  e t e f e r 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
		u	d	y			

Ciphertext : BSLEDOUFOUELYESBSUTKTOSHIART.

1.9.1 Rail Fence Cipher

- The Rail Fence Cipher is a transposition cipher. It rearranges the plaintext letters by drawing them in a way that they form a shape of the rails of an imaginary fence.
- To encrypt the message, the letters should be written in a zigzag pattern, going downwards and upwards between the levels of the top and bottom imaginary rails. The shape that is formed by the letters is similar to the shape of the top edge of the rail fence.
- Next, all the letters should be read off and concatenated, to produce one line of ciphertext. The letters should be read in rows, usually from the top row down to the bottom one.
- The secret key is the number of levels in the rail. It is also a number of rows of letters that are created during encryption. This number cannot be very big, so the number of possible keys is quite limited.
- Suppose we want to encrypt the message "buy your books in August" using a rail fence cipher with encryption key 3.

  a) Arrange the plaintext characters in an array with 3 rows (the key determines the number of rows), forming a zig-zag pattern :

Fig. 1.9.1

  b) Then concatenate the non-empty characters from the rows to obtain the ciphertext : **BOOIGUYUBOSNUUTYRKAS.**

1.9.2 Difference between Substitution Techniques and Transposition Techniques

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 but changes its identity	Each letter retains its identity but changes its position.
Disadvantage	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

Example 1.9.1 *Solve the following example using rail fence technique. "COMPUTER SECURITY IS IMPORTANT".*

Solution. :

Plain text : COMPUTER SECURITY IS IMPORTANT

Arrange the plaintext characters in an array with 3 rows.

C - - - U - - - S - - - R - - - I - - - P - - - A - -

- O - P - T - R - E - U - I - Y - S - M - O - T - N

- - M - - - E - - - C - - - T - - - I - - - R - - - T

Ciphertext : **CUSRIPAOPTREUIYSMOTNMECTIRT**

Example 1.9.2 *Decipher a message : "TSACT SGCEB HISRM SELNV ISEEE AVITP" using a Rail fence using 10 Columns and 3 rails and retrieve original message.*

Solution. : The number of columns in rail fence cipher remains equal to the length of plaintext message. Hence, rail matrix can be constructed accordingly.

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

Original Message : - THIS IS A SECRET MESSAGE VCLIEVT BNP

Example 1.9.3 *Convert plain text to cipher text using Rail Fence technique "COMPUTER ENGINEERING".*

Solution. : Plain text = COMPUTER ENGINEERING

Step 1 : Write down Plain text as sequence of diagonal.

C				U				E				N				I		
	O		P		T		R		N		I		E		R		N
		M				E				G				E				G

Ciphertext : CUENIOPTRNIERNMEGEG

Example 1.9.4 *Convert plain text into cipher text by using simple columnas techniques of the following sentence: "ALL IS WELL FOR YOUR EXAM".*

Solution. : Consider the six columns. Therefore, we write the message in the rectangle row-by-row :

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
A	L	L	I	S	W
E	L	L	F	O	R
Y	O	U	R	E	X
A	M				

Now, let us decide the order of columns at some random order, say 5,2,4,1,6,3. Then read the text in the order of these columns.

Ciphertext = SOELLOMIFRAEYAWRXLLU

Example 1.9.5 *Convert plain text to cipher text using Rail Fence technique "COMPUTER SECUIRTY".*

Solution : Plain text : COMPUTER SECURITY

Arrange the plaintext characters in an array with 3 rows.

C				U				E				R		
	O		P		T		R		S		U		I	
		M				E				C				T	

Ciphertext : **CUSROPTREUIYMECT**

Example 1.9.6 Encrypt "Computer Security Technology" using rail fence technique.

Solution : Plain text : Computer Security Technology

Arrange the plaintext characters in an array with 3 rows.

C - - - u - - - S - - - r - - - T - - - n - - - g -
- o - p - t - r - e - u - i - y - e - h - o - o - y
- - m - - - e - - - c - - - t - - - c - - - l - - -

Ciphertext : CuSrtngoptreuiyehooymectcl

Review Question

1. *Compare substitution and transposition techniques with examples.* **AU : Dec.-22, Marks 5**

1.10 Steganography

AU : May - 19

- Steganography is derived from the Greek for covered writing and essentially means "to hide in plain sight".
- As defined as it is the art and science of communicating in such a way that the presence of a message cannot be detected. Simple steganographic techniques have been in use for hundreds of years, but with the increasing use of files in an electronic format new techniques for information hiding have become possible.
- The other major area of steganography is copyright marking, where the message to be inserted is used to assert copyright over a document. This can be further divided into watermarking and fingerprinting.
- Steganography and encryption are both used to ensure data confidentiality. However, the main difference between them is that with encryption anybody can see that both parties are communicating in secret.
- Steganography hides the existence of a secret message and in the best case nobody can see that both parties are communicating in secret. This makes staganography suitable for some tasks for which encryption aren't, such as copyright marking.
- Adding encrypted copyright information to a file could be easy to remove but embedding it within the contents of the file itself can prevent it being easily identified and removed.
- Steganography provides a means of secret communication which cannot be removed without significantly altering the data in which it is embedded. The embedded data will be confidential unless an attacker can find a way to detect it.

- The following is a list of main requirements that staganography techniques must satisfy :
  1. The integrity of the hidden information after it has been embedded inside the stego object must be correct.
  2. The stego object must remain unchanged or almost unchanged to the naked eye. If the stego object changes significantly and can be noticed, a third party may see that information is being hidden and therefore could attempt to extract or to destroy it.
  3. In watermarking, changes in the stego object must have no effect on the watermark.
  4. Finally, we always assume that the attacker knows that there is hidden information inside the stego object.
- Fig. 1.10.1 shows a simple process in steganography. In this example, a secret image is being embedded inside a cover image to produce the stego image.

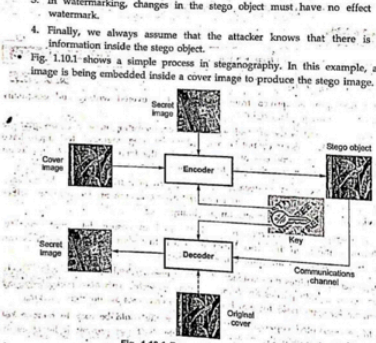


Fig. 1.10.1 Process in steganography

- The first step in embedding and hiding information is to pass both the secret message and the cover message into the encoder. Inside the encoder, one or several protocols will be implemented to embed the secret information into the cover message. The type of protocol will depend on what information you are trying to embed and what you are embedding it in.

- A key is often needed in the embedding process. This can be in the form of a public or private key so you can encode the secret message with your private key and the recipient can decode it using your public key.
- In embedding the information this way, you can reduce the chance of a third party attacker getting hold of the stego object and decoding it to find out the secret information.
- In general the embedding process inserts a mark, M, in an object, I. A key, K, usually produced by a random number generator is used in the embedding process and the resulting marked object, $\tilde{I}$, is generated by the mapping : $I \times K \times M \rightarrow \tilde{I}$.
- Having passed through the encoder, a stego object will be produced. A stego object is the original cover object with the secret information embedded inside. This object should look almost identical to the cover object as otherwise a third party attacker can see embedded information.
- Having produced the stego object, it will then be sent off via some communications channel, such as email, to the intended recipient for decoding.
- The recipient must decode the stego object in order for them to view the secret information. The decoding process is simply the reverse of the encoding process. It is the extraction of secret data from a stego object.
- In the decoding process, the stego object is fed into the system. The public or private key that can decode the original key that is used inside the encoding process is also needed so that the secret information can be decoded.
- Depending on the encoding technique, sometimes the original cover object is also needed in the decoding process. Otherwise, there may be no way of extracting the secret information from the stego object.
- After the decoding process is completed, the secret information embedded in the stego object can then be extracted and viewed.
- The generic decoding process again requires a key, K, this time along with a potentially marked object, $\tilde{I}'$. Also required is either the mark, M, which is being checked for or the original object, I and the result will be either the retrieved mark from the object or indication of the likelihood of M being present in $\tilde{I}'$. Different types of robust marking systems use different inputs and outputs.

Steganographic Techniques

1. **Genome steganography :** Encoding a hidden message in a strand of human DNA.
2. **Hiding in text :** Information hidden in documents by manipulating the positions of lines and words, hiding the data in html files.

3. **Hiding in the disk space :** Hiding the data in unused or reserved space.
4. **Hiding data in software and circuitry :** Data can be hidden in the layout of the code distributed in a program or the layout of electronic circuits on a board.
5. **Information hiding in images :** Ranges from least significant bit insertion to masking and filtering to applying more sophisticated image processing algorithms.
6. **Hiding in network packets :** Hidden in packets transmitted through the Internet.

Limitations

- With encryption, Bob can be reasonably sure that he has received a secret message when a seemingly meaningless file arrives. It has either been corrupted or is encrypted. It is not so clear with hidden data; Bob simply receives an image, for example and needs to know that there is a hidden message and how to locate it.
- Another limitation is due to the size of the medium being used to hide the data. In order for steganography to be useful the message should be hidden without any major changes to the object it is being embedded in. This leaves limited room to embed a message without noticeably changing the original object.
- This is most obvious in compressed files where many of the obvious candidates for embedding data are lost. What is left is likely to be the most perceptually significant portions of the file and although hiding data is still possible it may be difficult to avoid changing the file.

1.10.1 Requirements of Steganography Technique

- The following is a list of main requirements that steganography techniques must satisfy :

1. The integrity of the hidden information after it has been embedded inside the stego object must be correct.
2. The stego object must remain unchanged or almost unchanged to the naked eye. If the stego object changes significantly and can be noticed, a third party may see that information is being hidden and therefore could attempt to extract or to destroy it.
3. In watermarking, changes in the stego object must have no effect on the watermark.
4. Finally, we always assume that the attacker knows that there is hidden information inside the stego object.

1.10.2 Difference between Steganography and Cryptography

Sr. No.	Cryptography	Steganography
1.	It is a technique to convert the secret message into other than human readable form.	It is a technique to hide the existence of the communication.
2.	It is a kind of known communication.	It is a kind of hidden communication.
3.	Cryptography alters the overall structure of the data.	Steganography does not alter the overall structure of the data.
4.	The final result obtained is known as cipher text.	The final result obtained is known as stego media.
5.	Once it has been discovered no one can easily get the secret data.	Once it has been discovered anyone can get the secret data.

Review Question

1. *What is steganography ? Describe the various techniques used in steganography.*
   AU : May-19, Marks 7

1.11 Foundations of Modern Cryptography AU : Dec.-22

- Cryptography : The scientific study of techniques for securing digital information, transaction, and distributed computations.
- Classical cryptography was restricted to military. Modern cryptography is influences almost everyone. Classical cryptography was mostly about secret communication. With modern cryptography the scope has expanded. It now deals with digital signatures, digital cash, secure voting....
- Modern cryptography breaks out of the "design-break-design" cycle model of classical cryptography. The security is not based on the secrecy of the protocol details but based on sound mathematical and computational principles.
- Provable security: It is now possible to formally argue about the security of protocols. It can be proven that breaking the cryptosystem requires solving some other problem in mathematics, which is believed to be difficult.

1.11.1 Perfect Security

- A ciphertext maintains perfect secrecy if the attacker's knowledge of the contents of the message is the same both before and after the adversary inspects the

ciphertext, attacking it with unlimited resources. That is, the message gives the adversary precisely no information about the message contents.

- What is a "secure" cipher? Intuitively, the answer is that a secure cipher is one for which an encrypted message remains "well hidden," even after seeing its encryption.
- If Alice encrypts a message (m) under a key (k) and an eavesdropping adversary obtains the cipher text (c), Alice only has a hope of keeping message secret if the key is hard to guess, and that means, at the very least, that the key should be chosen at random from a large key space.
- Messages come from some distribution; let D be a random variable for sampling the messages from the message space M. Distribution D is known to the adversary. This captures a priori information about the messages.
- The ciphertext c = Enc (m; k) depends on:
  m chosen according to D; k is chosen randomly; Enc may also use some randomness
- These induce a distribution C over the ciphertexts.
- Knowledge about m before observing the output of C is captured by: D
- Knowledge about m after observing the output of C is captured by: D|C
- Shannon secrecy: distribution D and D|C must be identical.
- Intuitively, this means that : C contains no NEW information about m, in the standard sense of information theory.
- **Definition of Shannon Secrecy :** A cipher (M; K; KG; Enc; Dec) is Shannon secure w.r.t a distribution D over M if for all m' ∈ M and for all c,
  $Pr [m \leftarrow D : m = m'] = Pr [k \leftarrow KG, m \leftarrow D : m = m' | Enc(m, k) = C]$
- **It is Shannon secure** if it is Shannon secure w.r.t. all distributions D over M.
- Suppose we have two messages : $m_1 \in M$ and $m_2 \in M$.
- What is the distribution of ciphertexts for m_1 ?
- Likewise, for m_2, the ciphertext distribution is :
- **Perfect secrecy :** C_1 and C_2 must be identical for every pair of m_1; m_2.
- **Ciphertexts are independent of the plaintext.**
- Definition of Perfect Secrecy : Scheme (M; K; KG; Enc; Dec) is perfectly secure for every pair of messages m_1; m_2 in M and for all c,
  $Pr [k \leftarrow KG : Enc (m_1, k) = c] = Pr [k \leftarrow KG : Enc (m_2, k) = c]$

- A private-key encryption scheme is perfectly secure if and only if it is Shannon secure.

1.11.2 Information Theory

- An information theory has mainly been used in cryptography to prove lower bounds on the size of the secret key required to achieve a certain level of security in secrecy and authentication systems. In order to prove the security of a cryptographic system, a definition of security or, alternatively, of breaking the system must be given.
- Whether a system with provable security is satisfactory from a theoretical and practical viewpoint depends in a crucial manner on three aspects :
  a) On the acceptability and generality of the definition of security;
  b) On how realistic the two assumptions are; and
  c) On the practicality of the system.
- For instance, it is trivial to "prove" the security of a cipher if we define security to mean that an adversary is unable to square a circle with straightedge and compass. It is similarly trivial to prove that an adversary cannot obtain any information about the plaintext for a system in which the legitimate receiver cannot either, or if one assumes that the adversary is unable to even receive the ciphertext.
- There are two possible types of assumptions about the adversary's computing power :
  1. A system is called computationally-secure if it is secure against an adversary with reasonably bounded computational resources and it is called information-theoretically secure if it is secure even against adversaries with infinite computing power
  2. The second type of assumption, namely that an adversary has infinite computing power, implies no restriction whatsoever and therefore anticipates all arguments about model's of computation and realistic estimates of an opponent's computing power.
- The role of information theory in cryptography can be characterized as that of deriving results on the provable security of a system, even in presence of adversaries with infinite computing power.
- According to Shannon, the entropy of an information source S is defined as :

$$H(s) = \sum i\,pi \log_2(1/pi)$$

where pi is the probability that symbol Si in S will occur.

- Information theory has been used in cryptography primarily to derive pessimistic results, i.e., lower bounds on the size of the secret key necessary to achieve a certain level of security.

1.11.3 Product Cryptosystem

- A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. A cryptosystem is also referred to as a cipher system.
- A cryptosystem is a structure or scheme consisting of a set of algorithms that converts plaintext to ciphertext to encode or decode messages securely. The term "cryptosystem" is shorthand for "cryptographic system" and refers to a computer system that employs cryptography, a method of protecting information and communications through the use of codes so that only those for whom the information is intended can read and process it.
- To help keep data secure, cryptosystems incorporate the algorithms for key generation, encryption and decryption techniques. Cryptosystems are used for sending messages in a secure manner over the internet, such as credit card information and other private data.
- Components of cryptosystems are plaintext, encryption algorithm, cipher text, decryption algorithm and encryption & decryption key.
- Types of cryptosystems : Cryptosystems are categorized by the method they use to encrypt data, either symmetrically or asymmetrically.
- Symmetric key encryption is when the cryptosystem uses the same key for both encryption and decryption. Asymmetric key encryption is when the cryptosystem uses different keys for encryption and decryption.

1.11.4 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 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.
- The goal of a cryptanalyst is to find some weakness or insecurity in a cryptographic scheme to prevent obtaining plaintext without keys.
- There are at least five main sections that best describe the types of attacks within cryptanalysis.

1. Ciphertext-only attacks, where you just have access to the ciphertext. This type is rather hard to implement in more recent cryptosystems, since they are better protected.
2. Known-plaintext attacks, where you have access to the ciphertext and a plaintext-ciphertext pair.
3. Chosen-plaintext attacks, where you may choose a plaintext and learn its ciphertext in regards to how it was encrypted.
4. Chosen-ciphertext attacks, the same as the previous one except that here you may pick ciphertexts and find out the appropriate plaintexts.
5. Chosen-text attacks, where you have access to the plaintext and ciphertext, where the purpose of it is to find out the key.

Cryptology :

- Cryptology is the art and science that deals with both cryptography and cryptanalysis.
- Today we need cryptology because of the everyday use of computers and the Internet. It is important for businesses to be able to protect the information in their computers.
- If you decide to buy a CD from Amazon.com/flipkart.com using your credit card, it is important that no one but Amazon has the ability to read the file where your credit card number is stored. Electronic fund transfers have made privacy a great concern.

1.11.4.1 Cryptanalysis Attack Types

- There are numerous techniques for performing cryptanalysis, depending on what access the cryptanalyst has to the plaintext, ciphertext, or other aspects of the cryptosystem. Below are some of the most common types of attacks :
- Cryptanalysis attack types include :
  1. Known-Plaintext Analysis (KPA) : Attacker decrypt ciphertexts with known partial plaintext.
  2. Chosen-Plaintext Analysis (CPA) : Attacker uses ciphertext that matches arbitrarily selected plaintext via the same algorithm technique.
  3. Ciphertext-Only Analysis (COA) : Attacker uses known ciphertext collections.
  4. Man-In-The-Middle (MITM) Attack : Attack occurs when two parties use message or key sharing for communication via a channel that appears secure but is actually compromised. Attacker employs this attack for the interception

of messages that pass through the communications channel. Hash functions prevent MITM attacks.

5. Adaptive Chosen-Plaintext Attack (ACPA) : Similar to a CPA, this attack uses chosen plaintext and ciphertext based on data learned from past encryptions.

Review Question

1. *Outline any four types of cryptanalysis attack and explain with neat sketches. How this attack is made possible ?* AU : Dec.-22, Marks 13

1.12 Two Marks Questions with Answers

Q.1 Distinguish active and passive attack with example. AU : Dec.-22, 20, 16, May-15, 11

Ans. : Difference between passive and active attacks :

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.

Q.2 What are the key principle of security ? AU : Dec-22

Ans. : Key principle of security is Confidentiality, integrity, and availability. Confidentiality means protecting information from unofficial broadcasting and unauthorized access to people. Data integrity aims to maintain the information's consistency, accuracy, and authenticity. Availability is to provide data, technological infrastructure, and applications when the organisation needs them.

Q.3 What is meant by denial of service attack ? It Active Attack or Passive Attack ? AU : Dec.-21

Ans. : Fabrication causes Denial of service attacks. DOS prevents the normal use or management of communication facilities. It is active attack.

Q.4 Encrypt the plaintext tobeomottobe using the vigenere cipher for the key value Now. AU : Dec.-20

Ans. :

Key	Now Now Now Now
Plaintext	tob eor not tob e
Ciphertext	gcx rcn ncp gcx r

Q.5 Let message = "Anna", and k = 3, find the cipher text using Caesar. AU : Dec.-21

Ans. : Message = "Anna" key = 3 Cipher text = Dqqd

Q.6 What is a security mechanism ?

Ans. : A security mechanism is any process that is designed to detect, prevent or recover from a security attack.

Q.7 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.8 List some examples of security attacks.

Ans. : 1) Gain unauthorized access to information.

2) Disallow responsibility or liability for information the cheater did originate.

3) Enlarge cheater's legitimate license.

4) Prevent the function of software, typically by adding a convert function.

5) Cause others to violate a protocol by means of introducing incorrect information.

Q.9 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.10 What is an active attack ?

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

Q.11 Categorize passive and active attack. AU : Dec.-17

Ans. : Active attacks can be subdivided into four types :

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

Passive attacks are of two types : 1. Release of message contents 2. Traffic analysis

Q.12 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.13 What is a threat ? List their types. AU : May-18

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.14 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.15 List the classical encryption techniques. AU : Dec.-18

Ans. : Classical encryption techniques are : Caesar Cipher, Monoalphabetic Cipher, Playfair Cipher, Hill Cipher, Polyalphabetic Substitution, One Time Pad and Feistel Cipher.

Q.16 Define symmetric encryption. AU : Dec.-15

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

Q.17 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.18 What are the two basic functions used in the encryption algorithm ? AU : Dec.-14

Ans. : All the encryption algorithms are based on two general principles :

- **Substitution :** In which each element in the plaintext is mapped into another element.
- **Transposition :** In which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost.

Q.19 How many keys are required for two people to communicate via a cipher ?

Ans. : If both sender and receiver use the same key, the system is referred as symmetric, single-key, secret-key or conventional encryption. If both sender and receiver use a different key, the system is referred as asymmetric, two-key or public key encryption.

Q.20 Why is asymmetric cryptography bad for huge data ? Specify the reason. AU : May-18

Ans. : Asymmetric encryption limits the maximum size of the plaintext. In practice, block modes don't get used with asymmetric encryption, because encrypting many blocks with an asymmetric scheme would be really slow.

Q.21 What are the two general approaches to attacking a cipher ?

Ans. : The two general approaches for attacking a cipher.

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 samples plaintext-cipher text pairs.
2. **Brute-force attack :** The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained.

Q.22 Define the caesar cipher.

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

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

The general Caesar cipher algorithm is

$$C = E(p) = (p + k) \text{ mode } (26)$$

Where k takes the value in the range 1 to 25

The decryption algorithm is

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

Q.23 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.24 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.25 What is the difference between a monoalphabetic cipher and a polyalphabetic cipher ? AU : Dec.-12, CSE/IT

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

Q.26 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.27 Define steganography. AU : May-13, CSE/IT

Ans. : Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity.

Q.28 Why modular arithmetic has been used in cryptography ? AU : Dec.-13, CSE/IT

Ans. : Applications of modular are given to divisibility tests and to block ciphers in cryptography. Modular arithmatic directly underpins public key system such as RSA and Diffie-Hell man as well as providing finite fields which underlie elliptic curves and is used in a variety of symmetric key algorithms including AES, IDEA and RC4.

Q.29 List out the problems of one time pad ? AU : Dec.-13, CSE/IT

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.30 Distinguish between attack and Threat. AU : Dec.-13

Ans. :

- The main difference between threat and attack is a threat can be either intentional or unintentional where as an attack is intentional.
- Threat is a circumstance that has potential to cause loss or damage whereas attack is attempted to cause damage.
- Threat to the information system doesn't mean information was altered or damaged but attack on the information system means there might be chance to alter, damage, or obtain information when attack was successful.
- A security threat is the expressed potential for the occurrence of an attack.
- A security attack is an action taken against a target with the intention of doing harm.

Q.31 Specify the components of encryption algorithm or What are the ingredients of a symmetric cipher ? AU : May-19

Ans. : Components of encryption algorithm :

Plaintext - original message

Ciphertext - coded message

Cipher - algorithm for transforming plaintext to ciphertext

Key - info used in cipher known only to sender / receiver

Enciphering (encryption) - converting plaintext to ciphertext

Deciphering (decryption) - recovering ciphertext from plaintext

Q.32 List the entities that are to be kept secret in conventional encryption techniques. AU : Dec.-19

Ans. : Secret key and an encryption algorithm.

□□□

UNIT II

2 Symmetric Ciphers

Syllabus

Number theory - Algebraic Structures - Modular Arithmetic - Euclid's algorithm - Congruence and matrices - Group, Rings, Fields, Finite Fields SYMMETRIC KEY CIPHERS: SDES - Block Ciphers - DES, Strength of DES - Differential and linear cryptanalysis - Block cipher design principles - Block cipher mode of operation - Evaluation criteria for AES - Pseudorandom Number Generators - RC4 - Key distribution.

Contents

2.1 Number Theory

AU : Dec.-22

- Number theory is the study of the set of positive whole numbers 1,2,3,4,5,6,7,8,9, : : ; which are often called the set of natural numbers. Number theory is about integers and their properties.
- Number theory is the study of the integers.
- In modern cryptographic system, the messages are represented by numerical values prior to being encrypted and transmitted. The encryption processes are mathematical operations that turn the input numerical values into output numerical values.
- Mathematical tools are required for building, analyzing and attacking the cryptosystems.

2.1.1 Divisibility

Definition : *Given two integers 'a' and 'b', we say 'a' divides 'b' if there is an integer c such that b = ac. If a divides b, we write a | b.*

For example : 7 / 63 because $7 \times 9 = 63$

- A consequence of this definition is that every number divides zero. Since a.0 = 0 for every integer a. If a divides b, then b is a *multiple* of a. For example, 63 is a multiple of 7.
- The following statements about divisibility hold.
  1. If a | b, then a | bc for all c.
  2. If a | b and b | c, then a | c;
  3. If a | b and a | c, then a | sb + tc for all s and t.
  4. For all c ≠ 0, a | b if and only if ca | cb.

Example 2.1.1 *Which of the following is true ?*

1. 77 | 7 2. 7 | 77 3. 24 | 24 4. 0 | 24 5. 24 | 0

Solution :

1. 77 | 7 : False bigger number can't divide smaller positive number
2. 7 | 77 : True because 77 = 7 · 11
3. 24 | 24 : True because 24 = 24 · 1
4. 0 | 24 : False; only 0 is divisible by 0
5. 24 | 0 : True, 0 is divisible by every number (0 = 24 · 0)

2.1.2 Prime Number

- A prime number is an integer that can only be divided without remainder by positive and negative values of itself and 1.
- Any integer a > 1 can be factored in a unique way as

$$a = p_1^{a_1} p_2^{a_2} p_3^{a_3} \ldots\ldots p_t^{a_t}$$

Where $p_1 < p_2 < \ldots\ldots < p_t$ are prime numbers and where each a_i is a positive integer. This is known as the fundamental theorem of arithmetic.

- If p is the set of all prime numbers then any positive integer a can be written uniquely in the following form :

$$a = \prod_{p \in P} p^{a_p} \text{ where each } a_p \geq 0$$

2.1.2.1 Relatively Prime Numbers

- **Definition :** Two integers a and b are **relatively prime** if gcd (a, b) = 1.
- The integers a1, a2, ..., an are **pair-wise relatively prime** if gcd (ai, aj) = 1 whenever $1 \leq i < j \leq n$.
- **Example 1 :** Are 15, 17 and 27 pair-wise relatively prime ? *No, because gcd(15, 27)= 3.*
- **Example 2 :** Are 15, 17 and 28 pair-wise relatively prime ? *Yes, because gcd(15, 17)= 1, gcd (15, 28) = 1 and gcd (17, 28) = 1.*
- Number that is relatively prime to another number means that the GCD of the two numbers is 1. Therefore, it does not mean that either of the numbers has to be prime.
- The method for calculating the number of relatively prime numbers less than a given number involves prime factorization, which can be reviewed in positive integral divisors.
  1. Find the exponential prime factorization of the number.
  2. Taking each term separately, change the term to 2 numbers :
     a. Subtract 1 from the base for the first number.
     b. Subtract 1 from the exponent and evaluate the expression for the second number.
  3. Multiply all the numbers together found in step 2.

Example : How many numbers less than 20 are relatively prime to 20 ?

- The prime factorization of 20 is : $2^2 \times 5^1$
- Taking 2^2 first, we get : $2-1=1$ and $2^2-1=2$

- Taking 5^1 we get : $5-1=4$ and $5^1-1=1$
- Multiplying all of them together we get : (1) (2) (4) (1) or 8.
- The answer is 8. The numbers which are relatively prime are 1, 3, 7, 9, 11, 13, 17 and 19. So indeed there are 8.

Example 2.1.2 *Is 97 a prime ?*

Solution : The floor of $\sqrt{97} = 9$. The primes less than 9 are 2, 3, 5, and 7. We need to see if 97 is divisible by any of these numbers. It is not, so 97 is a prime.

2.1.3 Algebraic Structures

- Cryptography requires sets of integers and specific operations that are defined for those sets. The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure.
- Algebra is about operations on sets. Some of the operations are as follows :
  a) Addition and multiplication of numbers;
  b) Modular arithmetic;
  c) Addition and multiplication of polynomials;
  d) Addition and multiplication of matrices;
  e) Union and intersection of sets;
  f) Composition of permutations.
- Common algebraic structures are rings, groups and field.

Reveiw Question

1. *Discuss properties that are satisfied by groups, rings and fields.*

AU : Dec.-22, Marks 13

2.2 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,

  $x \equiv y \bmod N$ if $x = y + mN$ 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 : $11 \bmod 7 = 4$; $-11 \bmod 7 = 3$.

Theorem : $\equiv_n$ 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 $n \mid (a - b)$.
  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 $(a \bmod n) = r_a$ and $(b \bmod n) = r_b$. 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, ..., (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 ∈ 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 × 3 = 18 ≡ 12 mod 8 and

6 × 7 = 42 ≡ 2 mod 8 but

3 ≠ 7 mod 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 2.2.1 Arithmetic modulo 7

2.2.1 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 2.2.1 *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 $11^{13} \bmod 53 = 52$.

2.3 Euclid's 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 :

$$a = q_1 b + r_1, \quad \text{where } 0 < r < b$$

$$b = q_2 r_1 + r_2, \quad \text{where } 0 < r_2 < r_1$$

$$r_1 = q_3 r_2 + r_3, \quad \text{where } 0 < r_3 < r_2$$

$$r_i = q_{i+2} r_{i+1} + r_{i+2}, \quad \text{where } 0 < r_{i+2} < r_{i+1}$$

$$r_{k-1} = q_{k+1} r_k$$

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

$125 = 1 * 87 + 38$

$87 = 2 * 38 + 11$

$38 = 3 * 11 + 5$

$11 = 2 * 5 + 1$

$5 = 5 * 1$

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

Example 2.3.1 *Find gcd (125, 20)*

Solution : $125 = 6 * 20 + 5$

$20 = 4 * 5$,

Thus, the gcd(125, 20) = 5

2.3.1 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 :

$$a = q_1 b + r_1, \quad \text{where } 0 < r < b$$
$$b = q_2 r_1 + r_2, \quad \text{where } 0 < r_2 < r_1$$
$$r_1 = q_3 r_2 + r_3, \quad \text{where } 0 < r_3 < r_2$$

$$r_i = q_{i+2} r_{i+1} + r_{i+2}, \quad \text{where } 0 < r_{i+2} < r_{i+1}$$

$$r_{k-2} = q_k r_{k-1} + r_k, \quad \text{where } 0 < r_k < r_{k-1}$$
$$r_{k-1} = q_{k+1} r_k$$

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

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

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 :

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

- 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) :

$$135 = 2 * 50 + 35$$
$$50 = 1 * 35 + 15$$

$$35 = 2 * 15 + 5$$
$$15 = 3 * 5$$

- 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 2.3.2 *Using Euclidean algorithm calculate gcd (16, 20) and gcd (50, 60).*

Solution : gcd (16, 20)

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

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

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

Thus gcd (16, 20) = 4

gcd (50, 60)

$a_1 = 60$, $b_1 = 50$

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

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

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

Example 2.3.3 *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 x 18 + 12 gcd (18, 12)

18 = 1 x 12 + 6 gcd(12, 6)

12 = 2 x 6 + 0 gcd(6, 0)

Therefore, GCD(48, 30) = 6

GCD(105, 80)

105 = 1 . 80 + 25 gcd(80, 25)

80 = 3 25 + 5 gcd (25, 5)

25 = 5 5 + 0 gcd(5, 0)

Therefore, GCD(105, 80) = 5

2.4 Finite Fields

- A field is a set of elements on which two arithmetic operations i.e. addition and multiplication, have been defined and which has the properties of abstract algebra arithmetic, such as closure, associativity, commutativity, distributivity and having both additive and multiplicate inverses.
- A finite field is simply a field with a finite number of elements. It can be shown that the order of a finite field must be a power of a prime p^n, where n is a positive integer.
- Finite field of order p can be defined using arithmetic mod p.
- **Properties**
  1. It can be shown that finite fields have order p^n, where p is a prime.
  2. It can be shown that for each prime p and each positive integer 'n', there is upto isomorphism, a unique finite field of order p^n.
  3. Let $GF(p^n)$ represent a finite field of order p^n. GF stands for Galois field.

Construction of finite fields

- To construct $GF(p^n)$ first find an irreducible polynomial I of degree n, with coefficients in Z_p.
- Let $GF(p^n) = \{a_{n-1} X^{n-1} + a_{n-2} X^{n-2} + ... + a_1 x + a_0 \mid a_i \varepsilon Z_p\}$
- (Note that here addition is done modulo Z_p while multiplication is done modulo I).
- Example GF (16) = $GF(2^4)$ we want polynomial of degree 4 with coefficients in $Z_2 = \{ax_3 + bx_2 + cx + d \mid a, b, c, d, \varepsilon Z_2\}$.

- Here addition is done as in Z_2 [x], while multiplication is done modulo $x^4 + x + 1$.

Properties of GF (p^n)

- It can be shown that for each positive integer n there exists an irreducible polynomial of degree n over GF(p) for any p.
- It can be shown that for each divisor m of n, $GF(p^n)$ has a unique sub field of order p^m. Moreover, these are the only subfields of $GF(p^n)$.

Primitive Element

- A nonzero element a ε GF(q) is called a **Primitive Element** if $h^1, h^2 ... h^{q-1}$, are precisely all the nonzero elements of GF(q) (i.e. the multiplicative order of a is (q – 1)).
  1. Generator of the multiplicative group of nonzero elements.
  2. Used to simplify multiplication.
- It can be shown that every $GF(p^n)$ contains a primitive element.

2.4.1 Groups

- 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 $g^n = g*g*...*g$ 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 ∈ G, then $(a * b)^{-1} = b^{-1} * a^{-1} = a^{-1} * b^{-1}$

2.4.2 Ring with Unity

- A Ring R is a nonempty set with two binary operations, addition (denoted by a + b) and multiplication (denoted ab), such that for all a, b, c ∈ R :
  1. R is an abelian group under addition.
  2. a(bc) = (ab)c (associativity)
  3. a(b + c) = ab + ac and (b + c)a = bc + ca.
- A Unity in a ring is a nonzero element that is the identity under multiplication.
- A Commutative Ring R is ring such that for all a, b, c ∈ R.
  1. a(b + c) = ab + ac = (b + c)a (commutativity)
- A Unit is a nonzero element of a Commutative Ring with Unity that has a multiplicative inverse.
- A Zero-Divisor is a nonzero element a ∈ R, R is a commutative ring, such that there is a nonzero element b ∈ R with ab = 0.
- An Integral Domain is a commutative Ring with unity and no zero-divisors.

Polynominal Rings

- A polynominal over a commutative ring R is an expression of the form
  $f(x) = a_n x^n + \ldots + a_1 x + a_0$
  where the coefficients a_i, $0 \le i \le n$, are elements of R and x is a variable with indeterminate meaning. The set of all such expressions is denoted by R[x].
- The polynomial $0x^{m+n} + \ldots + 0x^{n+1} + a_n x^n + \ldots + a_1 x + a_0$ is regarded as the same polynomial as f(x). If $a_n \neq 0$, then n is called the degree of f(x), denoted by deg f(x). In this case $a_n = 1$ c (f(x)) is called the leading coefficient of f(x).
- Let $g(x) = b_m x^m + \ldots + b_1 x + b_0$ be a polynomial in R[x]. Addition of polynomials is defined by
  $f(x) + g(x) = b_m x^m + \ldots + b_{n+1} x^{n+1} + (a_n + b_n) x^n + \ldots + (a_1 + b_1) x + (a_0 + b_0)$
  where we assumed without loss of generality that m ≥ n. The multiplication of polynomials is defined by
  $f(x)g(x) = c_{m+n} x^{m+n} + \ldots + c_2 x^2 + c_1 x + c_0$, where $c_k = \sum_{i+j=k} a_i b_j$.

2.5 Symmetric Ciphers

- A symmetric encryption model has five ingredients.
  1. Plaintext
  2. Encryption algorithm
  3. Secret key
  4. Ciphertext
  5. Decryption algorithm

- Fig. 2.5.1 shows the conventional encryption model.

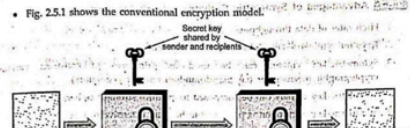


Fig. 2.5.1 Conventional encryption model

- **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 message 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.

2.5.1 Advantages of Symmetric Ciphers

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.

2.5.2 Disadvantages of Symmetric Ciphers

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.

2.6 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.

plaintext = $b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8$

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

Subkey generation

- First, produce two subkeys K_1 and K_2:

$$K_1 = P8(LS_1(P10(key)))$$

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

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 :

$$P10(k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10}) = k_3k_5k_2k_7k_4k_{10}k_1k_9k_8k_6$$

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 :

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

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 $+$.

$f_K(L, R) = (L + F_K(R), R)$

$F_K(R) = P4 (S0(lhs(EP(R)+K)) , S1(rhs(EP(R)+K)))$

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	2	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. 2.6.1 shows one round of a Feistel system.

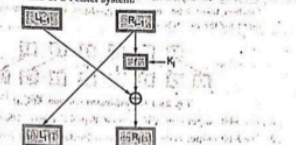


Fig. 2.6.1 One round of a Feistel system

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

$$L_i = R_{i-1} \text{ and } R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

- This operation is performed for a certain number of rounds, say n, and produces L_nR_n.
- The ciphertext will be R_nL_n.
- Encryption and decryption are done the same way except the keys are selected in the reverse order.
- The key... K_i.
- **Function $f(R_{i-1}, K_i)$:** The function $f(R_{i-1}, K_i)$, depicted in the Fig. 2.6.2 below, is described in folls for encryption will be $K_1, K_2, \ldots K_n$ and for decryption will be $K_n, \ldots K_{n-1}$ ...owing steps.

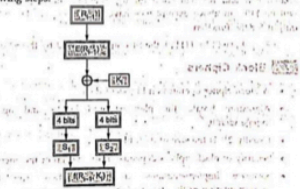


Fig. 2.6.2 The Function $f(R_{i-1}, K_i)$

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.

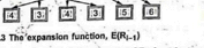


Fig. 2.6.3 The expansion function, $E(R_{i-1})$

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.

2.7 Block Ciphers

- A block cipher operates on blocks of data.
- Algorithm breaks the plaintext into blocks and operates on each block independently.
- Usually 2^n is the size of each block.
- 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.

Plaintext = 227 bytes

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

$$C_0 = E_K(m_0 \oplus I)$$

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

2.7.1 Advantages and Disadvantage of Block Cipher

Advantages :

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

Disadvantages :

1. Slowness of encryption
2. Error propagation.

2.8 DES

AU : Dec.-21

- 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. 2.8.1 shows DES encryption algorithm. First, the 64-bit plaintext passes through an Initial Permutation (IP) that rearranges the bits to produce the permuted input.
- 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.

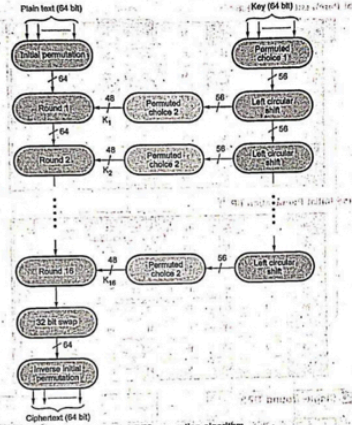


Fig. 2.8.1 DES encryption algorithm

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	21	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

2.8.1 Single Round DES

- Fig. 2.8.2 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 :

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

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

- 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.

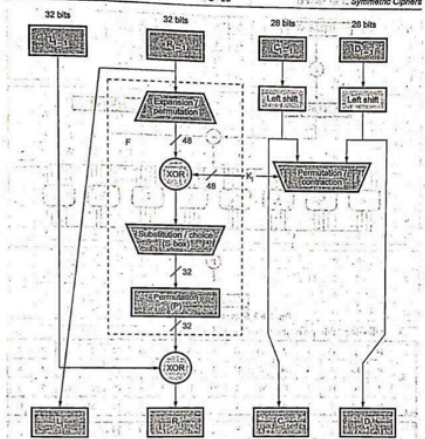


Fig. 2.8.2 Single round of DES algorithm

- Fig. 2.8.3 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.
- 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.

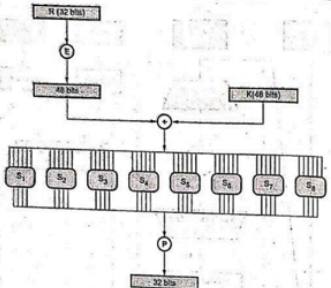


Fig. 2.8.3 S-boxes in the function (F)

- 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. 2.8.4 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. 2.8.4 Selecting entry in S-box

2.8.2 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 $F(R_{i-1}, K_i)$.

2.8.3 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 :

$$L' = R$$
$$R' = L(+) f(R,K) \quad \ldots (2.8.1)$$

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 (2.8.1) 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.

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. 2.8.5.

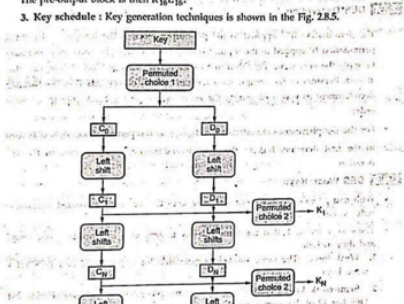


Fig. 2.8.5 Key generation techniques

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.

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

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

2.8.4 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 $R_{16}L_{16}$ as the permuted input, K_{16} is used in the first iteration, K_{15} in the second, and so on, with K_1 used in the 16th iteration.

2.8.5 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.

2.8.6 Avalanche Effect in DES

- 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 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 in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext.
- This is referred to as the avalanche effect. If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched.

2.8.7 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.

2.8.8 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.

2.8.9 S-Box Design Criteria

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 between 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 middle bits of S_{block} on the next round, then an output bit from S_k cannot affect a middle bit of S_j.

2.8.10 Double DES

- Using two encryption stages and two keys.

A) The plain text to ciphertext is as follows,

$C = E_{K_2}(E_{K_1}(P))$ where k_1 and k_2 are the key.

B) Ciphertext to plain text is as follows,

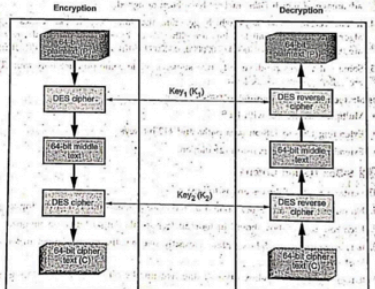


Fig. 2.8.6 Meet-in-the-middle attack for double DES

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

- Double DES suffers from Meet-in-the-Middle Attack.
- Meet-in-the-Middle Attack is as follows,
  1. Assume $C = E_{K_2}(E_{K_1}(P))$
  2. Given the plaintext P and ciphertext C
  3. Encrypt P using all possible keys K_1
  4. Decrypt C using all possible keys K_2

Fig. 2.8.6 shows the meet-in-the-middle attack for double DES.

2.8.11 Triple DES

- 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. 2.8.7 shows the 3DES method with three key.
- Triple DES runs three times slower than standard DES, but is much more secure if used properly.
- 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

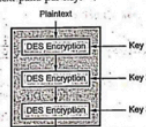


Fig. 2.8.7 3DES with three key method

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.

2.8.12 Triple DES with Two Keys

- In triple DES with two keys there are only two keys K1 used by first and third stage and K2 used in second stage.
- First the plain text is encrypted with key K1 then the output of step one is decrypted with K2 and final the output second step is encrypted again with key K1.
- The function follows an encrypt-decrypt-encrypt (EDE) sequence :

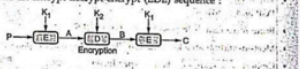


Fig. 2.8.8

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

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

- Fig 2.8.8 shows 3DES with two keys.
- There is no cryptographic significance to the use of decryption for the second stage. Its only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single

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

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

Review Question

1. *Draw the functionality diagram (functionality in one round) of DES with number of bits in each flow of data.* **AU : Dec.-21, Marks 8**

2.9 Differential Cryptanalysis

- Differential cryptanalysis is an approach to cryptanalysis whereby differences in inputs are mapped to differences in outputs and patterns in the mappings of plaintext edits to ciphertext variation are used to reverse engineer a key.
- Differential cryptanalysis aims to map bitwise differences in inputs to differences in the output in order to reverse engineer the action of the encryption algorithm. It is again aiming to approximate the encryption algorithm looking to find a maximum likelihood estimator of the true encryption action by altering plaintexts and analyzing the impact of changes to the plaintext to the resulting ciphertext. Differential cryptanalysis is therefore a chosen plaintext attack.
- The main difference from linear attack is that differential attack involves comparing the XOR of two inputs to the XOR of the corresponding output.
- Differential attack is a **chosen-plaintext attack.**
- This is a chosen plaintext attack, assumes than an attacker knows (plaintext, ciphertext) pairs.
- Difference $\Delta P = P_1 \oplus P_2$, $\Delta C = C_1 \oplus C_2$.
- Distribution of ΔC's given ΔP may reveal information about the key.
- After finding several bits, use brute-force for the rest of the bits to find the key.
- Surprisingly ...DES was resistant to differential cryptanalysis.
- At the time DES was designed, the authors knew about differential cryptanalysis. S-boxes were designed to resist differential cryptanalysis.
- Against 8-round DES, attack requires 2^{28} known plaintext-ciphertext pairs.
- Against 16-round DES, attack required 2^{47} chosen plaintexts.
- Differential cryptanalysis not effective against DES !!!

2.9.1 Linear Cryptanalysis

- Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, "ciphertext" bits and subkey bits. It is a known plaintext attack.

- Performing linear cryptanalysis on a block cipher usually consists of three steps :
  1. Find linear approximations of the non-linear parts of the encryption algorithm (usually only the substitution boxes, known as S-boxes).
  2. Combine linear approximations of S-boxes with the rest of the (linear) operations done in the encryption algorithm, to obtain a linear approximation of the entire encryption algorithm. This linear approximation is a function which relates the plaintext bits, the ciphertext bits, and the bits of the private key.
  3. Use the linear approximation as a guide for which keys to try first. This leads to substantial computational savings over trying all possible values of the key. Multiple linear approximations may be used to further cut down the number of keys that need to be tried.

2.9.2 Difference between Differential and Linear Cryptanalysis

Sr. No.	Linear cryptanalysis	Differential cryptanalysis
1.	Linear cryptanalysis focus on statistical analysis against one round of decrypted ciphertext.	Differential analysis focuses on the statistical analysis of two inputs and two outputs of a cryptographic algorithm.
2.	Linear cryptanalysis is one of the two most widely used attacks on block ciphers.	Differential cryptanalysis is usually a chosen plaintext attack, meaning that the attacker must be able to obtain encrypted cipher-texts for some set of plaintexts of choosing.
3.	Linear cryptanalysis is a general form of cryptanalysis based on finding affine approximations to the action of a cipher.	Differential cryptanalysis is a general form of cryptanalysis applicable primarily to block ciphers, but also to stream ciphers and cryptographic hash functions.
4.	linear cryptanalysis "only" requires known plaintext	Differential cryptanalysis requires chosen plaintext

2.10 Block Cipher Mode of Operation AU : Dec.-22

Different types of cipher block modes are discussed here.

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.

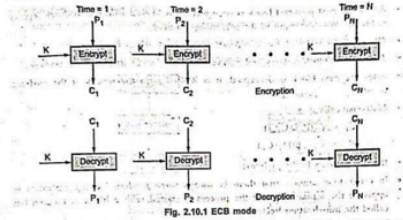


Fig. 2.10.1 ECB mode

- 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.
- 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.

2. Cipher Block Chaining Mode (CBC)

- 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 the 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).

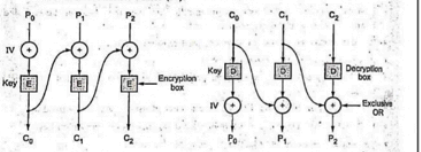


Fig. 2.10.2 CBC

- 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. 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.
- Fig. 2.10.3 shows encryption and decryption process. (See Fig. 2.10.3 on next page).

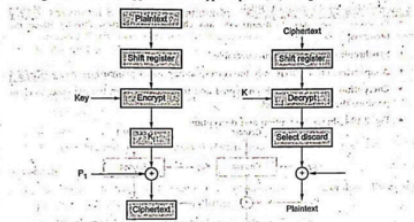


Fig. 2.10.3 CFB modes

- 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.

4. 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.
- 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.
- Fig. 2.10.4 shows encryption and decryption.

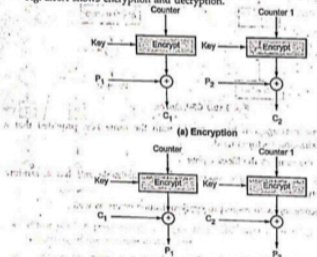


Fig. 2.10.4 Counter mode

Advantages

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

University Question

1. Explain in detail on the design principles of block cipher and the modes of operation. **AU : Dec.-22, Marks 13**

2.11 Advanced Encryption Standards AU : Dec.-16,17,20,21, May-17,18,19

- Advanced Encryption Standard (AES) is a symmetric key block cipher published by the NIST in December 2001.

2.11.1 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.

2.11.2 AES Cipher

AU : May-18

- 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.

Characteristics

1. Resistance against all known attacks.
2. Speed and code compactness on a wide range of platforms.
3. Design simplicity.

- For 128-bits AES, each round contains four steps :
  - i. Byte substitution
  - ii. Row shift
  - iii. Column mixing
  - iv. Round key addition
- The input to the encryption and decryption algorithms is a single 128-bit block. The block is represented as a row of matrix of 16 bytes.
- Fig. 2.11.1 shows the overall structure of AES. (See Fig. 2.11.1 on next page.)
- AES use several rounds in which each round is made of several stages. Data block is transformed from one stage to another.

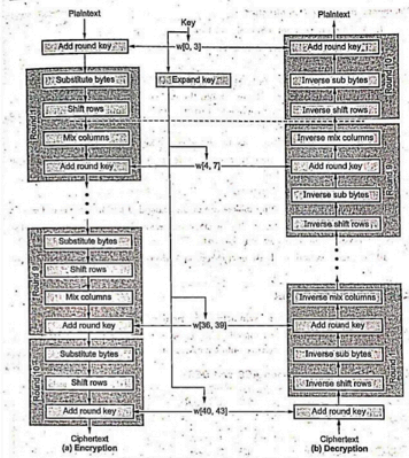


Fig. 2.11.1 AES encryption and decryption

- Data block is referred to as state. Block is copied into state array which is modified at each stage of encryption or decryption. After the final stage, state is copied to an output matrix.

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.

2.11.3 Applications of AES

1. AES can be used anywhere, symmetric key cryptography is needed.
2. There is no particular list of applications of AES, but many banking systems use AES-128 and AES-256 to secure online banking or internet banking.

2.11.4 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

Example 2.11.1 *For each of the following elements of DES, indicate the comparable element in AES or explain why it is not needed in AES.*

a. XOR of subkey material with the input to the function f

b. XOR of the f function output with left side of the block

c. The f function d. Permutation P

e. Swapping of halves of the block. **AU : Dec.-17, Marks 16**

Solution :

a. **XOR of subkey material with the input to the function f :** The similar element in AES for XOR of subkey with the input to the function (that passes different stages before XORing) is the added round key stage in all the 10 rounds.

b. **XOR of the f function output with left side of the block :** There is no similar element in AES for XOR the f function output with left half side of the block, this is because AES structure is not a feistel structure. The entire block is processed in parallel (No two halves are using one half to modify the other half).

c. **The f function :** There is no single element that is similar to f function, but the four stages (Substitution bytes, shift rows, mix columns, added roundly) in each round do the same as f function.

d. **Permutation P :** The similar element for P is the shift rows in each of the 10 rounds.

e. **Swapping of halves of the block :** No similar element in AES this is because that AES structure not a feistel structure and no need to swap halves since work in parallel (No half needs to modify the other half).

Review Questions

1. *Explain AES algorithm with all its round functions in detail.* **AU : Dec.-16, Marks 16**
2. *What do you mean by AES ? Diagrammatically illustrate the structure of AES and describe the steps in AES encryption process with example.* **AU : May-18, Marks 16**
3. *Describe in detail the key generation in AES algorithm and its expansion format.* **AU : May-19, Marks 7**
4. *For each of the following elements of DES, indicate the comparable element in AES if available :*
   a) XOR of subkey material with the input to the function.
   b) f function. **AU : Dec.-20, Marks 4 + 4**
5. *What do you mean by AES ? Diagrammatically illustrate the structure of AES and describe the steps in AES encryption process with example.* **AU : Dec.-20, Marks 13**
6. *Explain with sample data : Four transformations in AES.* **AU : Dec.-21, Marks 5**

2.12 Stream Cipher

- Stream cipher algorithms are designed to accept a crypto key and a stream of plaintext to produce a stream of ciphertext.
- Fig. 2.12.1 shows the stream cipher.

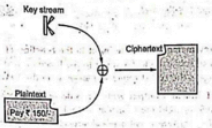


Fig. 2.12.1 Stream cipher

- 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.

$$C_i = P_i \oplus K_i$$

- 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.

2.12.1 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.

2.12.2 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.

2.13 Pseudorandom Number Generators

- Random numbers play an important role in the use of encryption for various network security applications. Getting good random numbers is important, but difficult.
- The random number generation can be divided into two categories: true random number generation and pseudo random number generation.
- With true random number generation the next random number generated is not known, and the sequence of random numbers cannot be re-generated.
- With pseudo random number generation, a sequence of "random numbers" is generated using a known algorithm and the exact same sequence can be regenerated; hence the classification of pseudo.
- Cryptographic applications typically make use of deterministic algorithmic techniques for random number generation, producing sequences of numbers that are not statistically random, but if the algorithm is good, the resulting sequences will pass many reasonable tests of randomness. Such numbers are referred to as pseudorandom numbers, created by "Pseudorandom Number Generators (PRNGs)".

- Computers may generate a set of numbers that are close to random, but are not exactly random; we call such numbers pseudo-random. Economists, statisticians and scientists use pseudo-random numbers all the time.
- When a computer generates a good pseudo-random sequence, the cryptanalyst often has to rely on brute force. Even brute forcing might not give him the correct message, but rather a set of messages with a high probability that it matches the original text.
- Some programs require a large number of random numbers, we can greatly speed up the program by using a faster, more efficient random number generator. The method of this random number generation by linear congruential method, works by computing each successive random number from the previous.

Cryptographically Generated Random Numbers :

- We use the encryption logic to produce random number. Three representative examples are
  1. Cyclic encryption
  2. DES output feedback mode
  3. ANSI X9.17 PRNG

1. Cyclic Encryption :

- It generates session keys from a master key. A counter with period N provides input to the encryption logic. If 56-bit DES keys are to be produced, then a counter with period 2^{56} can be used. After each key is produced, the counter is incremented by one.
- Fig. 2.13.1 shows the pseudorandom number generation from a counter.

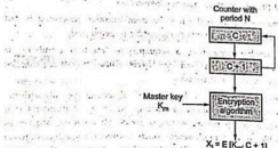


Fig. 2.13.1 Pseudorandom number generation from a counter

- The pseudorandom numbers produced by this scheme cycle through a full period.

Cryptography and Cyber Security 2 - 48 Symmetric Ciphers
Computers may generate a set of numbers that are close to random, but are not exactly random; we call such numbers pseudo-random. Economists, statisticians and scientists use pseudo-random numbers all the time.
When a computer generates a good pseudo-random sequence, the cryptanalyst often has to rely on brute force. Even brute forcing might not give him the correct message, but rather a set of messages with a high probability that it matches the original text.
Some programs require a large number of random numbers, we can greatly speed up the program by using a faster, more efficient random number generator. The method of this random number generation by linear congruential method, works by computing each successive random number from the previous.
Cryptographically Generated Random Numbers :
We use the encryption logic to produce random number. Three representative examples are
1. Cyclic encryption
2. DES output feedback mode
3. ANSI X9.17 PRNG
1. Cyclic Encryption :
It generates session keys from a master key. A counter with period N provides input to the encryption logic. If 56-bit DES keys are to be produced, then a counter with period 2^{56} can be used. After each key is produced, the counter is incremented by one.
Fig. 2.13.1 shows the pseudorandom number generation from a counter.
Fig. 2.13.1 Pseudorandom number generation from a counter
The pseudorandom numbers produced by this scheme cycle through a full period.
TECHNICAL PUBLICATIONS® - an up-thrust for knowledge

- Each of the outputs $X_0, X_1, \ldots, X_{N-1}$ is based on a different counter value and therefore $X_0 \neq X_1 \neq \ldots \neq X_{N-1}$.
- It is not computationally feasible to deduce any of the session keys through knowledge of one or earlier session keys.
- If this is possible, it means the encryption algorithm is broken in the same way.
- So if the encryption algorithm is safe, the session keys cannot be deduced.

2. DES output feedback mode

- DES output feedback mode is used for key generation as well as for stream encryption.
- Successive 64-bit outputs constitute a sequence of pseudorandom numbers with good statistical properties.

3. ANSI X9.17 PRNG

- Fig. 2.13.2 shows ANSI X9.17 PRNG method.

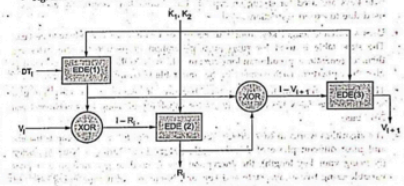


Fig. 2.13.2 ANSI X9.17 PRNG method

- It consists of iterations where each iteration uses triple DES.
- For the I^{th} iteration.

1. **Input :** Two 64-bit pseudorandom numbers.
   - DT_I : Current data and time
   - V_I : A seed generated in the previous iteration.
2. **Output :** Two 64-bit pseudorandom numbers.
   - R_I : pseudorandom number
   - V_{I+1} : The seed for the next iteration

3. **Key :** It makes use of three triple DES encryption modules.

$$R_I = EDE_{K_1,K_2}[V_I \oplus EDE_{K_1,K_2}[DT_I]]$$

$$V_{I+1} = EDE_{K_1,K_2}[R_I \oplus EDE_{K_1,K_2}[DT_I]]$$

- K_1, K_2 : Two 56-bit DES keys.
- Even if $R_1, R_{i+1}, \ldots, R_{i+j}$ is known, it is difficult to deduce R_{i+j+1} because DT_{i+j+1} and V_{i+j} are unknown.

2.14 RC4

AU : Dec.-21

- RC4 is an encryption algorithm that was created by Ronald Rivest of RSA Security. It is used in WEP and WPA, which are encryption protocols commonly used on wireless routers. The algorithm is based on the use of a random permutation.
- RC4 was originally very widely used due to its simplicity and speed. Typically, 16 byte keys are used for strong encryption, but shorter key lengths are also widely used due to export restrictions.
- Uses a variable length key from 1 to 256 bytes to initialize a 256-byte state table. The state table is used for subsequent generation of pseudo-random bytes and then to generate a pseudo-random stream which is XORed with the plaintext to give the ciphertext. Each element in the state table is swapped at least once.
- The key is often limited to 40 bits, because of export restrictions but it is sometimes used as a 128 bit key. It has the capability of using keys between 1 and 2048 bits.
- The algorithm works in two phases, key setup and ciphering. Key setup is the first and most difficult phase of this encryption algorithm. During a N-bit key setup (N being your key length), the encryption key is used to generate an encrypting variable using two arrays, state and key, and N-number of mixing operations.
- These mixing operations consist of swapping bytes, modulo operations, and other formulas. A modulo operation is the process of yielding a remainder from division. For example, 11/4 is 2 remainder 3; therefore eleven mod four would be equal to three.
- Once the encrypting variable is produced from the key setup, it enters the ciphering phase, where it is XORed with the plain text message to create an encrypted message. XOR is the logical operation of comparing two binary bits. If the bits are different, the result is 1. If the bits are the same, the result is 0.

- To generate the key stream, the cipher makes use of a secret internal state which consists of two parts :
  1. A permutation of all 256 possible bytes (denoted "S" below).
  2. Two 8-bit index-pointers (denoted "i" and "j").
- The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the Key-Scheduling Algorithm (KSA). Then the stream of bits is generated by a pseudo-random generation algorithm.

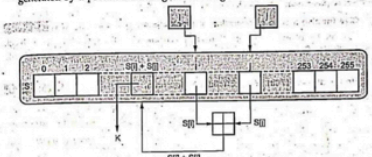


Fig. 2.14.1 Algorithm

RC4 Algorithm Strengths :

1. The difficulty of knowing where any value is in the table.
2. The difficulty of knowing which location in the table is used to select each value in the sequence.
3. A particular RC4 algorithm key can be used only once.
4. Encryption is about 10 times faster than DES.

RC4 Algorithm Weakness :

1. The algorithm is vulnerable to analytic attacks of the state table.
2. One in every 256 keys can be a weak key.

2.14.1 Uses of RC4

- RC4 has become part of some commonly used encryption protocols and standards such as WEP, WPA, TLS, Kerberos and SASL mechanism Digest MD5.

Review Question

1. *Explain the bitwise XOR operation which involved in RC4.* **AU : Dec.-21. Marks 5**

2.15 Two Marks Questions with Answers

Q.1 If a bit error occurs in plain text block b1, how far does the error propagate in CBC mode of DES? AU : Dec.-21

Ans. : If a bit of a plain text block b1 is in error the entire cipher text block will be effected and will be erroneous. All subsequent cipher blocks will also be effected such cipher text block is fed to next stage and XOR with next plain text block. However, at the receiver, only the block b1 of plain text recovered reproduces the same bit error. All the subsequent plain text blocks are reproduced correctly.

Q.2 Give the five modes of operation of block cipher. AU : Dec.-21

Ans. : Five modes of mode of operation are Electronic CodeBook (ECB), Cipher Block Chaining (CBC), Cipher feedback (CFB), Output Feedback (OFB) and Counter Mode (CTR).

Q.3 Find gcd(2740, 1760) using Euclidean algorithm. AU : Dec.-20, 21

Ans. :

q	r1	r2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

GCD = 20

Q.4 Find gcd (56, 86) using Euclid's algorithm. AU : Dec.-21

Ans. :

q	r1	r2	r
1	86	56	30
1	56	30	26
1	30	26	4
6	26	4	2
	4	2	0

Q.5 Define field and ring in number theory. AU : Dec.-10

Ans. :

- A ring R, sometimes denoted by {R, +, ×}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in R the following axioms are obeyed.
- A field F, sometimes denoted by {F, +, ×}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in F the following axioms are obeyed.

Q.6 What is an avalanche effect ?

Ans. : Avalanche effect is that a small change in either the plaintext or the key should produce a significant change in the cipher text.

Q.7 What do you mean by differential cryptanalysis ?

Ans. : Differential cryptanalysis is a method for breaking certain classes of cryptosystems. Differential cryptanalysis is efficient when the cryptanalyst can choose plaintexts and obtain ciphertexts.

Q.8 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.9 What are the ECB and CBC modes ?

Ans. : When we use a block cipher to encrypt a message of arbitrary length, we use techniques that are known as modes of operation for the block cipher. In ECB mode, each plaintext block is encrypted independently with the block cipher. ECB mode is as secure as the underlying block cipher. In CBC mode, each plaintext block is exclusive-ORed with previous ciphertext block, then encrypted.

Q.10 What are the CFB and OFB modes ?

Ans. : The Cipher Feedback (CFB) mode and the Output Feedback (OFB) mode are two more standard modes of operation for a block cipher. In CFB mode, the previous ciphertext block is encrypted and the output produced is combined with the plaintext block using exclusive-or to produce the current ciphertext block. It is possible to define CFB mode so that it uses feedback that is less than one full data block. OFB mode is similar to the CFB mode except that the quantity exclusive-ORed with each plaintext block is generated independently of both the plaintext and ciphertext. The encryption of a plaintext block is derived by takint the exclusive-OR of the plaintext block with the relevant data block.

Q.11 What is the difference between statistical randomness and unpredictability ? AU : May-10, Oct.

Ans. : In applications such as reciprocal authentication and session key generation the requirement is not so much that the sequence of numbers be statistically random but that the successive numbers of the sequence are unpredictable. With true random sequences each number is statistically independent of other numbers in the sequence and therefore unpredictable.

Q.12 What are the different modes of operation in DES ? AU : May-11

Ans. : DES modes of operation :

1. Electronic Codebook Book (ECB) : Message is broken into independent blocks of 64 bits.
2. Cipher Block Chaining (CBC) : Message is broken in independent blocks of 64 bits, but next input depends on previous output.
3. Cipher FeedBack (CFB) : The message is XORed with the feedback of encrypting the previous block.
4. Output Feedback : The feedback is independent of the message

Q.13 What types of attacks are addressed by DES algorithm ?

Ans. : **Timing attacks :** Attacks actual implementation of cipher. Use knowledge of consequences of implementation to derive knowledge of some/all sub key bits. **Analytic attacks :** These utilize some deep structure of the cipher by gathering information about encryptions. It can eventually recover some/all of the sub-key bits, and if necessary then exhaustively search for the rest.

Q.14 Write down the purpose of the S-boxes in DES ?

Ans. : In S-box, each row defines a general reversible substitution. It consists of a set of 8 S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.

Q.15 List out the parameters of AES.

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

Q.16 Distinguish between differential and linear cryptanalysis.

Ans. : In differential cryptanalysis, it breaks the DES in less 2^{55} complexities. In cryptanalysis, it finds the DES key given 2^{47} plaintexts.

Q.17 What is the disadvantages with ECB mode of operation ? AU : May-13, Oct.

Ans. : Disadvantages :

a. Synchronization error is unrecoverable

b. Not suitable for lengthy messages.

Q.18 What are the modes of DES ? AU : Dec.-13, CSE/IT

Ans. : Five standard modes of operation :

1. Electronic Code Book (ECB) 2. Cipher Block Chaining (CBC)
3. Cipher Feedback (CFB) 4. Output Feed (OFB) 5. Counter (CTR)

Q.19 List the uses of RC4. AU : Dec.-13, CSE/IT

Ans. : RC4 has become part of some commonly used encryption protocols and standards such as WEP, WPA, TLS, Kerberos and SASL mechanism Digest MD5.

Q.20 Why random numbers are used in network security ? AU : May-14

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.21 State few applications of RC4 algorithm. AU : May-15

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.

Q.22 What is AES cipher ? AU : Dec.-15

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.

Q.23 Brief the strengths of triple DES. AU : Dec.-16, CSE/IT

Ans. : a) Strength for triple DES is actually 168 bits.
b) Brute force search impossible on triple DES.
c) It uses 2 or 3 keys.

Q.24 Give the five modes of operation of block cipher. AU : May-17

Ans. : Block cipher modes of operations are electronic code book, cipher block chaining mode, cipher feedback mode, counter mode and output feedback mode.

Q.25 Compare DES and AES. AU : Dec-18

Ans. :

AES	DES
AES stands for Advanced Encryption Standard.	DES stands for Data Encryption Standard.
Key length can be of 128-bits, 192-bits and 256-bits.	Key length is 56 bits in DES.
Number of rounds depends on key length : 10(128-bits), 12(192-bits) or, 14(256-bits).	DES involves 16 rounds of identical operations.

The structure is based on substitution permutation network.	The structure is based on feistal network.
AES is more secure than the DES chiphers and is the de facto world standard.	DES can be broken easily, as it has known vulnerabilities. 3DES (Triple DES) is a variation of DES which is secure than the usual DES.
The rounds in AES are : Byte Substitution, Shift Row, Mix Column and Key Addition.	The rounds in DES are : Expansion, XOR operation with round key.

Q.26 Define field and ring in number theory. AU : Dec-21

Ans. :

- **Field** - A field, denoted by F = <{...}, o, > is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation has no inverse.
- A Field supports two pairs of operations: addition/subtraction andmultiplication/division, except that the division by zero is not allowed.
- **Ring** - A ring, R = <{...}, o, >, is an algebraic structure with two operations. The second operation must be distributed over the second.

UNIT III

3 Asymmetric Cryptography

Syllabus

MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY : Primes - Primality Testing - Factorization - Euler's totient function, Fermat's and Euler's Theorem - Chinese Remainder Theorem - Exponentiation and logarithm ASYMMETRIC KEY CIPHERS : RSA cryptosystem - Key distribution - Key management - Diffie Hellman key exchange - Elliptic curve arithmetic - Elliptic curve cryptography.

Contents

3.1 Mathematics of Asymmetric Key Cryptography

3.1.1 Primes

- A prime number is an integer that can only be divided without remainder by positive and negative values of itself and 1.
- Any integer a > 1 can be factored in a unique way as

$$a = p_1^{a1} p_2^{a2} p_3^{a3} \ldots\ldots p_t^{at}$$

Where $p_1 < p_2 < \ldots\ldots < p_t$ are prime numbers and where each a_i is a positive integer. This is known as the fundamental theorem of arithmetic.

- If p is the set of all prime numbers then any positive integer a can be written uniquely in the following form :

$$a = \prod_{\pi \in P} p^{a_p} \text{ where each } a_p \geq 0$$

3.1.1.1 Relatively Prime Numbers

- **Definition :** Two integers a and b are relatively prime if gcd (a, b) = 1.
- The integers a1, a2, ..., an are pair-wise relatively prime if gcd $(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.
- **Example 1 :** Are 15, 17 and 27 pair-wise relatively prime ? *No, because gcd(15, 27) = 3.*
- **Example 2 :** Are 15, 17 and 28 pair-wise relatively prime ? *Yes, because gcd(15, 17) = 1, gcd (15, 28) = 1 and gcd (17, 28) = 1.*
- Number that is relatively prime to another number means that the GCD of the two numbers is 1. Therefore, it does not mean that either of the numbers has to be prime.
- The method for calculating the number of relatively prime numbers less than a given number involves prime factorization, which can be reviewed in positive integral divisors.
  1. Find the exponential prime factorization of the number.
  2. Taking each term separately, change the term to 2 numbers :
     a. Subtract 1 from the base for the first number.
     b. Subtract 1 from the exponent and evaluate the expression for the second number.
  3. Multiply all the numbers together found in step 2.

Example : How many numbers less than 20 are relatively prime to 20 ?

- The prime factorization of 20 is : $2^2 \times 5^1$
- Taking 2^2 first, we get : $2-1=1$ and $2^2-1=2$
- Taking 5^1 we get : $5-1=4$ and $5^1-1=1$
- Multiplying all of them together we get : (1) (2) (4) (1) or 8.
- The answer is 8. The numbers which are relatively prime are 1, 3, 7, 9, 11, 13, 17 and 19. So indeed there are 8.

Example 3.1.1 *Is 97 a prime ?*

Solution : The floor of $\sqrt{97} = 9$. The primes less than 9 are 2, 3, 5, and 7. We need to see if 97 is divisible by any of these numbers. It is not, so 97 is a prime.

3.1.2 Primality Testing

Two properties of prime numbers

1. If p is prime and a is a positive integer less than p, then $a^2 \bmod p = 1$ if an only if either a mod p = 1 or mod p = − 1 and p = p − 1. By the rules of modular arithmetic (a mod p) (a mod p) = a^2 mod p.
   Thus if either a mod p = 1 or a mod p = −1 then a^2 mod p = 1. Conversely, if a^2 mod p = 1, then (a mod p)^2 = 1 which is true only for a mod p = 1 or a mod p = −1.
2. Let p be a prime number greater than 2. We can then write $p - 1 = 2^k q$, with k > 0, q odd. Let a be any integer in the range $1 < a < p - 1$. Then one of the following conditions is true.
   a) a^q is congruent to 1 modulo p. That is $a^q \bmod p = 1$ or equivalently $a^q \equiv 1 \pmod p$.
   b) One of the numbers $a^q, a^{2q}, a^{4q}, \ldots, a^{2^{k-1}q}$ is congruent to − 1 modulo p. That is, there is some number j in the range $(1 \le j \le k)$ such that $a^{2^{j-1}q} \bmod p = -1 \bmod p = p - 1$ or equivalently, $a^{2^{j-1}q} \equiv -1 \pmod p$.

3.1.3 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 3.1.2 *Find gcd(1403, 1081)*

Solution :

$1403 = 1081 \cdot 1 + 322$

$1081 = 322 \cdot 3 + 115$

$322 = 115 \cdot 2 + 92$

$115 = 92 \cdot 1 + 23$

$92 = 23 \cdot 4 + 0$

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

Example 3.1.3 *Find gcd (120, 70)*

Solution :

$120 = 70 + 50$

$70 = 50 + 20$

$50 = 20 \times 2 + 10$

$20 = 10 \times 2 + 0$

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.

3.2 Euler's Totient Function

AU : Dec.-21

- Euler's totient function (also called the Phi function) counts the number of positive integers less than n that are coprime to n. That is, ϕ(n) is the number of $m \in N$ such that $1 \le m < n$ and gcd(m, n) = 1.
- The totient function appears in many applications of elementary number theory, including Euler's theorem, primitive roots of unity, cyclotomic polynomials, and constructible numbers in geometry.

- The following table shows the function values for the first several natural numbers :

n	$\phi(n)$	Numbers coprime to n
1	1	1
2	1	1
3	2	1,2
4	2	1,3
5	4	1,2,3,4
6	2	1,5
7	6	1,2,3,4,5,6
8	4	1,3,5,7
9	6	1,2,4,5,7,8
10	4	1,3,7,9
11	10	1,2,3,4,5,6,7,8,9,10
12	4	1,5,7,11
13	12	1,2,3,4,5,6,7,8,9,10,11,12
14	6	1,3,5,9,11,13
15	8	1,2,4,7,8,11,13,14

- Can you find some relationships between n and $\phi(n)$? One thing you may have noticed is that :
  when n is a prime number (e.g. 2, 3, 5, 7, 11, 13), $\phi(n) = n - 1$.
- But how about the composite numbers ? You may also have noticed that, for example, $15 = 3*5$ and $\phi(15) = \phi(3) \times \phi(5) = 2 \times 4 = 8$. This is also true for 14, 12, 10 and 6.
- However, it does not hold for 4, 8, 9. For example, $9 = 3 \times 3$, but $\phi(9) = 6 \neq \phi(3) \times \phi(3) = 2 \times 2 = 4$. In fact, this multiplicative relationship is conditional :
  when m and n are coprime, $\phi(m \times n) = \phi(m) \times \phi(n)$.

Review Question

1. Prove that Euler's Totient value of any prime number (p) is $p - 1$ and the Euler's Totient value of the non-prime number (n) is $(p - 1) \times (q - 1)$ where $p \times q$ are prime factor of n. **SPPU : Dec.-21, Marks 5**

3.3 Fermat's and Euler's Theorem

Fermat's theorem

If p is prime and a is a positive integer not divisible by p, then $a^{p-1} \equiv 1 \bmod(p)$.

Proof : Consider the set of positive integers less than p : {1, 2, ... p − 1} and multiply each element by a, modulo p, to get the set $X = \{a \bmod p, 2a \bmod p, \ldots\ldots (p-1)a \bmod p\}$. None of the elements of X is equal to zero because p does not divide a. Further more no two of the integers in X are equal.

Euler's theorem

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

$$a^{\phi(n)} \equiv 1 \pmod n \qquad \ldots (3.3.1)$$

Proof : Equation (3.3.1) is true if n is prime because in that case $\phi(n) = (n - 1)$ and Fermat's theorem holds. It also holds for any integer n. Recall that $\phi(n)$ is the number of positive integers less than n that are relatively prime to n. Consider the set of such integers, labelled as follows :

$$R = \{x_1, x_2, \ldots x_{\phi(n)}\}$$

That is, each element x_i of R is a unique positive integer less than n with gcd $(x_i, n) = 1$. Now multiply each element by a, modulation n :

$$S = \{(ax_1 \bmod n), (ax_2 \bmod n) \ldots\ldots (ax_{\phi(n)} \bmod n)\}$$

The set S is a permutation of R.

Therefore,

$$\prod_{i=1}^{\phi(n)} (ax_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$\prod_{i=1}^{\phi(n)} ax_i \equiv \prod_{i=1}^{\phi(n)} x_i \pmod n$$

$$a^{\phi(n)} \times \left[\prod_{i=1}^{\phi(n)} x_i\right] \equiv \prod_{i=1}^{\phi(n)} x_i \pmod n$$

$$a^{\phi(n)+1} \equiv a \pmod n$$

Example 3.3.1 *Explain Fermat's little theorem and solve the following using the same :*
i) $15^{18} \bmod 17$ ii) $5^{37} \bmod 13$.

Solution : Fermat's little theorem

1. If p is prime and a is an integer not divisible by p, then . . .

$$a^{p-1} \equiv 1 \pmod{p}.$$

2. And for every integer a :

$$a^{p-1} \equiv 1 \pmod{p}.$$

3. This theorem is useful in public key (RSA).

Fermat's theorem

If p is prime and a is a positive integer not divisible by p, $a^{p-1} \equiv 1$ a^{p-1} mod(p).

Proof : Consider the set of positive integers less than p : [1, 2, ... p – 1] and multiply each element by a, modulo p, to get the set X = [a mod p, 2a mod p, ... (p – 1) a mod p]. None of the elements of X is equal to zero because p does not divide a. Further more no two of the integers in X are equal.

i) $15^{18} \bmod 17$

$= [(15 \bmod 17) \times (15^{17} \bmod 17)] \bmod 17$

$= [(-2 \bmod 17) \times (-2 \bmod 17)] \bmod 17$

$= 4 \bmod 17$

ii) $5^{27} \bmod 13$

$= [(5^{14} \bmod 13) \times (5^{13} \bmod 13)] \bmod 13$

$= [(12 \bmod 13) \times (5 \bmod 13)] \bmod 13$

$= 8 \bmod 13$

3.4 Chinese Remainder Theorem AU : Dec.-16,20.21

- Find a number x such that have remainders of 1 when divided by 3, 2 when divided by 5 and 3 when divided by 7. i.e.
  1. $x = 1 \bmod 3$
  2. $x = 2 \bmod 5$
  3. $x = 3 \bmod 7$
- Integers can be represented by their residues modulo a set of pair-wise relatively prime moduli. For example : In Z_{36}, integer 8 can be represented by the residues of the 2 relatively prime factors of 10 (2 and 5) as a tuple (0, 3).
- Let $M = m_1 \times m_2 \times m_3 \times \ldots \times m_k$, where m_i's are pairwise relatively prime, i.e. $\gcd(m_i, m_j) = 1, 1 \le i \ne j \le k$

- Assertion.
  1. $A \leftrightarrow (a_1, a_2, \ldots a_k)$ where $A \in Z_m$, $a_i \in Z_{mi}$ and $a_i = A \bmod m_i$ for $1 \le i \le k$
  a) One to one correspondance (bijection) between Z_M and the Cartesian product $Z_{m1} \times Z_{m2} \times \ldots \times Z_{mk}$.
  b) For every integer A such that $0 \le A < M$, there is 'n unique k-tuple $(a_1, a_2, \ldots a_k)$ with $0 \le a_i < m_i$.
  c) For every such k-tuple $(a_1, a_2, \ldots a_k)$, there is a unque A in Z_M.
  d) Transformation from A to $(a_1, a_2, \ldots a_k)$ is unique
  e) Computing A from $(a_1, a_2, \ldots a_k)$ is done as follows :
  1. Let $M_i = M/m_i$ for $1 \le i \le k$; i.e. $M_i = m_1 \times m_2 \times \ldots \times m_{i-1} \times \ldots \times m_k$
  2. Note that $M_i = 0 \pmod{m_j}$ for all $j \ne i$
  3. Let $c_i = M_i \times (M_i^{-1} \bmod m_i)$ for $1 \le i \le k$
  4. Then $A = (a_1c_1 + a_2c_2 + \ldots + a_kc_k) \bmod M$
  5. $\leftarrow a_i = A \bmod m_i$, since $c_i \equiv M_j \equiv 0 \pmod{m_i}$ if $j \ne i$ and $c_i \equiv 1 \pmod{m_i}$
- Operations performed on the elements of Z_M can be equivalently performed on the corresponding k-tuples by performing the operation independently in each co-ordinate position.

  Example : $A \leftrightarrow (a_1, a_2, \ldots a_k)$ $B \leftrightarrow (b_1, b_2, \ldots b_k)$

  $(A + B) \bmod M \leftrightarrow ((a_1 + b_1) \bmod m_1, \ldots (a_k + b_k) \bmod m_k)$

  $(A - B) \bmod M \leftrightarrow ((a_1 - b_1) \bmod m_1, \ldots (a_k - b_k) \bmod m_k)$

  $(A \times B) \bmod M \leftrightarrow ((a_1 \times b_1) \bmod m_1, \ldots (a_k \times b_k) \bmod m_k)$
- CRT provides a way to manipulate (potentially large) numbers mod M in terms of tuple of smaller numbers.

Chinese remainder theorem :

Suppose gcd (m, n) = 1. given a and b, there exists exactly one solution x (mod mn) to the simultaneous congruence under certain conditions.

$x \equiv a \pmod m$, $x \equiv b \pmod n$.

Proof :

- There exist integers s, t such that $ms + nt = 1$. Then $ms \equiv 1 \pmod n$ and $nt \equiv 1 \pmod m$ Let $x = bms + ant$. Then $x \equiv ant \equiv a \pmod m$ and $x \equiv bms \equiv b \pmod n$ as desired.

- Suppose x_1 is another solution. Then $x \equiv x_1 \pmod{m}$ and $x \equiv x_1 \pmod{n}$ so $x-x_1$ is a multiple of both m and n.

Lemma :

Let m, n be integers with gcd (m, n) = 1. If an integer c is a multiple of both m and n, then c is a multiple of mn.

Proof :

Let $c = mk = nl$. Write $ms + nt = 1$ with integers s, t. Multiply by c to obtain $c = cms + cnt = mnls + mnkt = mn(ls + kt)$.

- To finish the proof of the theorem, let $c = x - x_1$ in the lemma to find that $x - x_1$ is a multiple of mn. Therefore, $x \equiv x_1 \pmod{mn}$. This means that any two solutions x to the system of congruences are congruent mod mn, as claimed.

Example 3.4.1 *Solve $x \equiv 3 \pmod 7$, $x \equiv 5 \pmod{15}$.*

Solution : $x \equiv 80 \pmod{105}$ (Note : 105 = 7·15). Since $80 \equiv 3 \pmod 7$ and $80 \equiv 5 \pmod{15}$, 80 is a solution. The theorem guarantees that such a solution exists, and says that it is uniquely determined mod the product mn, which is 105 in the present example.

How to solve :

- One way, which works with small numbers m and n, is to list the numbers congruent to b (mod n) until you find one that is congruent to a (mod m).
- For example, the numbers congruent to 5 (mod 15) are

  5, 20, 35, 50, 65, 80, 95, ...

  Mod 7, there are 5, 6, 0, 1, 2, 3, 4, ... since we want 3 (mod 7), we choose 80.
- For slightly larger numbers m and n, making a list would be inefficient. However, a similar idea works. The numbers congruent to b (mod n) are of the form $b + nk$ with k an integer, so we need to solve $b + nk \equiv a \pmod m$. This is the same as

  $$nk \equiv a - b \pmod m.$$

- Since gcd (m, n) = 1 by assumption, there is a multiplicative inverse i for n (mod m). Multiplication by 1 gives

  $$k \equiv (a - b)\, i \pmod m.$$

  Substituting back into $x = b + nk$, then reducing mod mn, gives the answer.

Example 3.4.2 *Solve $x \equiv 7 \pmod{12345}$, $x \equiv 3 \pmod{11111}$.*

Solution : First, we know from our calculations in section that the inverse of 11111 (mod 12345) is i = 2471.

Therefore $k \equiv 2471(7 - 3) \equiv 9884 \pmod{12345}$. This yields $x = 3 + 11111 \cdot 9884 = 109821127 \pmod{(11111.12345)}$.

Example 3.4.3 *In a Chinese remainder theorem, let $n = 210$ and let $n_1 = 5, n_2 = 6, n_3 = 7$. Compute $f^{-1}(3, 5, 2)$, i.e. given $x_1 = 3, x_2 = 5, x_3 = 3$, compute x.*

Solution :

$N_1 = n_2 \times n_3 = 42$

$N_2 = n_1 \times n_3 = 35$

$N_3 = n_1 \times n_2 = 30$

$v_1 = (N_1)^{-1} = 42^{-1} = 2^{-1} = 3 \pmod 5$

$v_2 = (N_2)^{-1} = 35^{-1} = 5^{-1} = 5 \pmod 6$

$v_3 = (N_3)^{-1} = 30^{-1} = 2^{-1} = 4 \pmod 7$

$x = a_1v_1N_1 + a_2v_2N_2 + a_{133}N_3$

$= 126 + 875 + 360$

$= 1361$

$x \equiv 101 \pmod{210}$

Example 3.4.4 *State Chinese Remainder theorem and find X for the given set of congruent equations using CRT.*

X = 2 (mod 3)

X = 3 (mod 5)

X = 2 (mod 7) **AU : Dec.-16. Marks 8**

Solution : The Chinese Remainder Theorem (CRT) tells us that since 3, 5 and 7 are co-prime in pairs then there is a unique solution modulo $3 \times 5 \times 7 = 105$.

$n_1 = 3, \quad n_2 = 5, \quad n_3 = 7$

$N = n_1 \times n_2 \times n_3 = 3 \times 5 \times 7 = 105.$

$c_1 = 2, \quad c_2 = 3, \quad c_3 = 2$

Now $N_1 = N/n_1$

$= 105/3$

$N_1 = 35$ and so $d_1 = 35^{-1} \pmod 3 = 2$,

$N_2 = N/n_2 = 105/5 = 21$ and so $d_2 = 21^{-1} \pmod 5 = 1$, and

$N_3 = N/n_3 = 105/7 = 15$ and so $d_3 = 15^{-1} \pmod 7 = 1$.

Hence $x = (2 \times 35 \times 2) + (3 \times 21 \times 1) + (2 \times 15 \times 1)$

$= 233$

$= 233 \pmod{105} = 23$

The solution is $x = 23$. You can check that by noting that the relations

$23 = 7 \times 3 + 2 = 2 \pmod 3$

$23 = 4 \times 5 + 3 = 3 \pmod 5$

$23 = 3 \times 7 + 2 = 2 \pmod 7$

are all satisfied for this value of x.

Review Questions

1. *State Chinese Remainder theorem and find the value of X for the given set of congruent equations using Chinese Remainder theorem.*
   $X = 1 \pmod 5$
   $X = 2 \pmod 7$
   $X = 3 \pmod 9$
   AU : Dec.-20, Marks 13
2. *A Box contains gold coins. If the coins are equally divided among three friends, two coins are left over. If the coins are equally divided among five friends, three coins are left over If the coins are equally divided among seven friends, two coins are left over. If the box holds smallest number of coins that meets these conditions, how many coins are there ? (Hint : Use Chinese Remainder Theorem).*
   AU : Dec.-21, Marks 15

3.5 Exponentiation and Logarithm

- The exponential function, written exp(x) or e^x, is the function whose derivative is equal to its equation. In other words :

If $y = e^x$ $\quad \frac{dy}{dx} = e^x$

If $y = e^{kx}$ $\quad \frac{dy}{dx} = ke^{kx}$ where-k is a constant

Because of this special property, the exponential function is very important in mathematics and crops up frequently.

Like most functions you are likely to come across, the exponential has an inverse function, which is $\log_e x$, often written ln x (pronounced 'log x').

3.5.1 Logarithms

- Discrete logarithms are fundamental to a number of public key algorithms, including Diffie-Hellman key exchange and the DSA.

The powers of an integer, modulo n.

- Every a and n that are relatively prime :

$$a^{\phi(n)} = 1 \pmod n \quad \text{... (3.5.1)}$$

where $\phi(n)$ Euler's quotient function, is the number of positive integers less than 'n' and relatively prime to 'n'.

- For more general expression :

$$a^m = 1 \pmod n \quad \text{... (3.5.2)}$$

If a and n are relatively prime, then there is at least one integer m that satisfies equation (3.5.2), namely $m = \phi(n)$

- Consider the powers of 7, modulo 19 :

$7^1 = 7 \bmod (19)$

$7^2 = 49 = 2 \times 19 + 11 = 11 \pmod{19}$

$7^3 = 343 = 18 \times 19 + 1 = 1 \pmod{19}$

$7^4 = 2401 = 126 \times 19 + 7 = 7 \pmod{19}$

$7^5 = 16807 = 884 \times 19 + 11 = 11 \pmod{19}$

- The sequence is periodic and the length of the period is the smallest positive exponent 'm' such that $7^m = 1 \pmod{19}$.
- The logarithm of a number is defined to be the power to which some positive base must be raised in order to equal the number. For base x and for a value y :

$$y = x^{\log_x(y)}$$

- The properties of logarithms include the following :

a. $\log_x(1) = 0$

b. $\log_x(x) = 1$

c. $\log_x(yz) = \log_x(y) + \log_x(z)$ ... (3.5.3)

d. $\log_x(y^r) = r \times \log_x(y)$ ... (3.5.4)

- The power of 'a' (primitive root) from 1 through $(p - 1)$ produce each integer from 1 through $(p - 1)$ exactly once.
- We also know that any integer 'b' satisfies

$b = r \pmod p$ for some r, where

$0 \leq r \leq (p - 1)$

by the definition of modulo arithmetic.

- It follow that for any integer 'b' and a primitive root 'a' of prime number 'p', we can find a unique exponent 'i' such that

$$b \equiv a^i \pmod p \qquad \text{where } 0 \le i \le (p-1)$$

This exponent is referred to as the discrete logarithm of the number 'b' for the base a (mod p). We denote this value as $d\log_{a,p}(b)$

$d\log_{a,p}(1) = 0$, because $a^0 \bmod p = 1 \bmod p = 1$

$d\log_{a,p}(a) = 1$, because $a^1 \bmod p = a$

3.5.2 Computing Discrete Logarithm

- We want to find a unique integer x such that $\alpha^x = \beta \pmod n$
- We can find x by solving : $x = \log_\alpha \beta \pmod n$
- Logarithms in real numbers are easy to calculate, partially because the log function is continuous and monotonically increasing.
- Discrete logarithms do not have either of these properties. For example, in a (mod 5) system, the powers of 2 are 1, 2, 4, 3.
- This wraparound makes the discrete log function significantly harder to compute than the ordinary log function.
- **Multiplicative group :** A set of congruence classes that is relatively prime to the modulus. We used the group Z_p, where the modulus is a prime number and the group is cyclic (the values repeat).
- **Order of a group :** The number of elements in a group, which can be found using Euler's totient function.
  1. For Z_p, this is $p - 1$.
  2. For Z_p^k, it is $(p - 1)\, p^{k-1}$
- **Generators and primitive elements :** An element that produces the other elements of the group when raised to various powers, primitive elements are also generators.
- **Problem :** We have a multiplicative group $(G, *)$, α is a generator of G having order n and β is an element generated by α. Remember, we want to find a unique integer x such that $\alpha^x = \beta \pmod n$ by solving $x = \log_\alpha \beta \pmod n$
- Computing $\alpha^x = \beta$ for a given x is simple and efficient using the square and multiply algorithm for exponentiation.

- Computing $a = \log_\alpha \beta$ is difficult and can consume a large amount of time and memory for large values, such as those used in cryptography.
- This property makes discrete logs ideal for cryptographic applications because one function is easy, but the inverse function is difficult.
- There is a class of public-key cryptosystems that use the discrete logarithm problem for key generation and encryption/decryption.

3.6 Asymmetric Key Ciphers

AU : May-19

Diffie and Hellman proposed a new type of cryptography that distinguished between encryption and decryption keys. One of the keys would be publicly known; the other would be kept private by its owner.

- 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.
- A public key encryption scheme has six ingredients. Fig. 3.7.1 shows public key cryptography.
  1. **Plaintext :** It is input to algorithm and in a readable message or data.
  2. **Encryption algorithm :** It performs various transformations on the plaintext.
  3. **Public and private keys :** One key is used for encryption and other is used for decryption.
  4. **Ciphertext :** This is the scrambled message produced as output. It depends on the plaintext and the key.
  5. **Decryption algorithm :** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.
- The essential steps are the following :
  1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
  2. Each user places one of the two keys in a public register. 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.

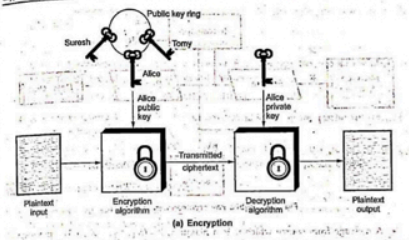


Fig. 3.6.1 Public key cryptography

- The public key is accessed to all participants and private key is generated locally by each participant.
- System controls its private key. At any time, a system can change its private key. Fig. 3.6.2 shows the process of public key algorithm.

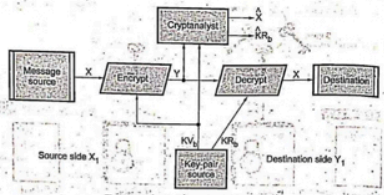


Fig. 3.6.2 Public key cryptosystem secrecy

- A message from source which is in a plaintext, $X = (X_1, X_2, \ldots X_m)$ The message is intended for destination which generates a related pair of keys a public key KU_b, and a private key KR_b.
- Private key is secret key and known only to Y_1. With the message X and encryption key KU_b as input, X_1 forms the ciphertext.

$$Y = (Y_1, Y_2, Y_3 \ldots Y_n)$$
$$Y = E_{KU_b}(X)$$

- The intended receiver, in possession of the matching private key is able to invert the transformation.

$$X = D_{KR_b}(Y)$$

- An opponent, observing Y and having access to public key (KU_b), but not having access to private key (KR_b), must attempt to recover X. It is assumed that the opponent does have knowledge of the encryption (E) and decryption algorithms (D).
- Public key cryptography requires each user to have two keys : A public key used by anyone for encrypting messages to be sent to that user and a private key, which the user needs for decrypting messages.

Requirements for 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).

3.6.1 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.

3.6.2 Comparison between Public Key and Private Key Algorithm

Sr. No.	Symmetric key cryptography	Asymmetric key cryptography
1.	Same key is used for encryption and decryption.	One key for encryption and other key for decryption.
2.	Very fast.	Slower.

3.	Key exchange is big problem.	Key exchange is not a problem.
4.	Also called secret key encryption.	Also called public key encryption.
5.	The key must be kept secret.	One of the two keys must be kept secret.
6.	The sender and receiver must share the algorithm and the key.	The sender and receiver must each have one of the matched pair of keys.
7.	Size of the resulting encrypted text is usually same as or less than the original clear text size.	Size of the resulting encrypted text is more than the original clear text size.
8.	Cannot be used for digital signatures.	Can be used for digital signature.

Review Question

1. *Explain public key cryptography and when it is preferred ?* AU : May-19, Marks 8

3.7 RSA Cryptosystem

AU : Dec.-13,18,21,22, May-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 $P^{ed} = P \bmod n$. 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.

3.7.1 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.

3.7.1.1 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.

$$n = pq \quad \text{... (3.7.1)}$$

$$\phi(n) = (p - 1)(q - 1) \quad \text{... (3.7.2)}$$

  for the two unknowns p and q.
- If we substitute $q = n/p$ into the equation (3.8.2), we obtain a quadratic equation in the unknown value p :

$$p^2 - (n - \phi(n) + 1)p + n = 0 \quad \text{... (3.7.3)}$$

- The two roots of equation (3.8.3) 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.

3.7.1.2 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.

3.7.3 Mathematical Attacks

- We can identify three approaches to attacking RSA mathematically :
  1. Factor n into two prime factors, this enables calculation of $\phi(n) = (p-1)(q-1)$, which in turn, enables determination of $d = e^{-1} \bmod \phi(n)$.
  2. Determine $\phi(n)$ directly, without first determining p and q.
  3. Determine d directly, without first determining $\phi(n)$
- Most discussions of cryptanalysis of RSA have focused on the task of factoring n into its two prime numbers. Determining $\phi(n)$ 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.

3.7.4 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 3.7.1 *For the given values p = 19, q = 23 and e = 3 find n, $\phi(n)$ and d using RSA algorithm.*

Solution :

$n = p * q$

$n = 19 \times 23$

$n = 437$

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

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

$\phi(n) = 396$

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

$3d = 1 \bmod 396$

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

Example 3.7.2 *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 its for i) and ii) and decrypt the ciphertext.

Solution : i)

$n = p * q$

$n = 3 * 11 = 33$

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

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

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

$7 \cdot d = 1 \bmod 20$

$d = 3$

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

$= 12^7 \bmod 33$

$C = 12$

ii)

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

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

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

$d = 3$

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 3.7.3 *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) $n = p * q$

$n = 11 \times 17 = 187$

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

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

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

$7 \cdot d = 1 \bmod 160$

$7 \times 23 = 1 \bmod 160$

Public key PU $(e, n) = 7, 187$

Private key PR $(d, n) = 23, 187$

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

Plaintext $p = c^d \bmod n$

$= 11^{23} \bmod 187$

$= 79720245 \bmod 187$

∴ Plaintext = 88

iii) Refer sections 3.7 and 2.15.

Example 3.7.4 *Explain about the RSA algorithm with example as : p = 11, q = 5, e = 3 and PT = 9* **AU : Dec.-13, Marks 16**

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

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

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

$e = 3$ and $m = 9$

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

$d = e^{-1} (\text{mod } \phi(n))$

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

$3d \bmod 40 = 1$

$d = 27$

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 3.7.5 *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 3.7.6 *Perform encryption and decryption using RSA algorithm for p = 17, q = 11, e = 7 and M = 88.* **AU : Dec.-18, Marks 13**

Solution : Given data : $p = 17, q = 11$

$N = p \times q = 17 \times 11$

$N = 187$

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

$= (17 - 1) \times (11 - 1)$

$\phi(N) = 160$

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

$7 \times d = 1 \bmod (160)$

$d = 23$

Encryption $C = M^e (\text{mod } N) = 88^7 \bmod 187$

$C = 11$

Description $M = C^d (\text{mod } N) = 11^{23} \bmod 187$

$M = 88$

Example 3.7.7 Perform encryption and decryption using RSA algorithm for $p = 7$, $q = 11$, $e = 7$ and $M = 9$. **AU : May-19, Marks 5**

Solution : Given data : $p = 7$, $q = 11$

$N = p \times q = 7 \times 11$

$N = 77$

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

$= (7 - 1) \times (11 - 1)$

$\phi(N) = 60$

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

$7 \times d = 1 \bmod (60)$

$d = 43$

Encryption $C = M^e (\text{mod } N) = 9^7 \bmod 77$

$C = 37$

Description $M = C^d (\text{mod } N) = 37^{43} \bmod 77$

$M = 9$

Review Questions

1. *In a public-key system using RSA, you intercept the ciphertext C = 20 sent to a user whose public key is e = 13, n = 77. What is the plaintext M ?* **AU : Dec.-21, Marks 7**
2. *In an RSA system, the public key of a given user is e = 65, n = 2881, What is the private key of this user ?* **AU : Dec.-21, Marks 6**
3. *Alice chooses 173 and 149 as two prime numbers and 3 as public key in RSA. Check whether the chosen prime numbers are valid or not ?* **AU : Dec.-21, Marks 5**
4. *Mr. Ram chooses RSA for encryption, and he chooses 3 and 7 are two prime numbers. He encrypt the given message (message given in English alphabets) by mapping A = 1, B = 2, C = 3 ..., Z = 26. Find atleast two problems in his implementation.* **AU : Dec.-21, Marks 5**
5. *Identify the possible threats for RSA algorithm and list their counter measures. Perform decryption and encryption using RSA algorithm with p = 3, q = 11, e = 7 and N = 5.* **AU : Dec.-22, Marks 13**

3.8 Key Distribution and Key Management

- Key management plays a fundamental role in cryptography as the basis for securing cryptographic techniques providing confidentiality, entity authentication, data origin authentication, data integrity, and digital signatures.
- The goal of a good cryptographic design is to reduce more complex problems to the proper management and safe-keeping of a small number of cryptographic keys, ultimately secured through trust in hardware or software by physical isolation or procedural controls.
- Reliance on physical and procedural security (e.g., secured rooms with isolated equipment), tamper-resistant hardware, and trust in a large number of individuals is minimized by concentrating trust in a small number of easily monitored, controlled, and trustworthy elements.
- Key management follows a lifecycle of operations which are needed to ensure the key is created, stored, used, and rotated securely. Most cryptographic keys follow a lifecycle which involves key : Generation, Distribution, Use, Storage, Rotation, Backup/Recovery, Revocation and Destruction.
- Key generators, AES encryption algorithms, or random number generators tend to be used for secure key generation. Keys should be distributed to the required user via a secure TLS or SSL connection, to maintain the security of the keys being distributed.
- The key should only be used by authorized users, to make certain the key is not misused, copied, etc. When the key is used to encrypt data, it must then be stored for later decryption. The most secure method is via a Hardware Security Module (HSM).
- If an HSM is not used, then the keys can either be securely stored on the client's side, or, if the keys are used on the Cloud, then the Cloud Service Provider's Key Management Service can be used.
- Once a key's crypto-period, or time period the key is usable, passes, the key must be rotated. When the key of an encrypted set of data expires, the key is retired and replaced with a new key.
- Revoking a key means the key can no longer be used to encrypt or decrypt data, even if its cryptoperiod is still valid. Destroying a key, whether that is due to compromise or due to it no longer being used, deletes the key permanently from any key manager database or other storage method.

3.9 Diffie-Hellman Key Exchange

AU : Dec.-17,20,21,22

- 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 $g^a \bmod p$ and Bob's public value is $g^b \bmod p$.
  4. They then exchange their public values.
  5. Finally, Alice computes $g^{ab} = (g^b)^a \bmod p$.
  6. Bob computes $g^{ba} = (g^a)^b \bmod p$.
  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 $Y_A = \alpha^{X_A} \bmod q$.
2. User B selects a random integer $X_B < q$ and compute $Y_B = \alpha^{X_B} \bmod q$.
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 $K = (Y_B)^{X_B} \bmod q$.
5. User B computes the key as $K = (Y_A)^{X_B} \bmod q$.

- Both side gets same results :

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

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.
  $X_A = 97$ $X_B = 233$
- Calculates the public keys

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

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

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.

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

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

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

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

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.

Problems

Example 3.9.1 *User A and B use the Diffie-Hellman key exchange technique with a common prime q = 71 and a primitive root $\alpha = 7$.*

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 $K = (Y_B)^{X_A} \bmod q$

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

$$K = 30$$

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

Example 3.9.2 *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)

$2^1 \bmod 11 = 2$

$2^2 \bmod 11 = 4$

$2^3 \bmod 11 = 8$

$2^4 \bmod 11 = 5$

$2^5 \bmod 11 = 10$

$2^6 \bmod 11 = 9$

$2^7 \bmod 11 = 7$

$2^8 \bmod 11 = 3$

$2^9 \bmod 11 = 6$

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

$2^6 \bmod 11 = 9$

$X_A = 6$

iii) $Y_B = (11)^6 \bmod 9$

$Y_B = 1$

iv) Shared secret key :

$$K = (Y_B)^{X_A} \bmod q$$

$$K = 3^6 \bmod 11$$

$$K = 5$$

Example 3.9.3 *Users Alice and Bob use the Diffie - Hellman key exchange technique with a common prime q = 83 and a primitive root α = 5.*

i) If Alice has a private key X_A = 6, what is Alice's public key Y_A ?

ii) If Bob has a private key X_B = 10, what is Bob's public key Y_B ?

iii) What is the shared secret key ? **AU : Dec.-17, Marks 16**

Solution : A's public key Y_A

$$Y_A = \alpha^{X_A} \bmod q$$

$$= (5^6) \bmod 83 = 21$$

B's public key Y_B

$$Y_B = \alpha^{X_B} \bmod q$$

$$= (5^{10}) \bmod 83 = 11$$

Shared secret key :

i) At user A :

$$K = (Y_B)^{X_A} \bmod q = (11^6) \bmod 83 = 9$$

Review Questions

1. *Alice and Bob use the Diffie – Hellman key exchange technique with a common prime number 11 and a primitive root of 2. If Alice and Bob choose distinct secret integers as 9 and 3, respectively, then compute the shared secret key.* **AU : Dec.-20, Marks 5**
2. *Diffie-Hellman key agreement is not limited to negotiating a key shared by only two participants. Any number of users can take part in an agreement by performing iterations of the agreement protocol and exchanging intermediate, Write the steps and formulas to be followed for DH key exchange between Alice, Bob, and Carol.* **AU : Dec.-21, Marks 6**
3. *Users A and B use the Diffie-Hellmann key exchange technique, a common prime q = 11 and a permitive root alpha = 7.*

   i) If user A has private key XA = 3. What is A's public key YA ?

   ii) If user B has private key XB = 6. What is B's public key YB ?

   iii) What is the shared secret key ? Write the Algorithm. **AU : Dec.-22, Marks 4 + 4 + 5**

3.10 Elliptic Curve Arithmetic

- Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks. They are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

3.10.1 Abelian Groups

- Abelian group G is denoted by {G, • }. It is set of elements with a binary operation.
- Each ordered pair (a, b) of elements in G an element (a • b) in G, such that the following axioms are obeyed :

(A1) Closure : If a and b belong to G, then a•b is also in G.

(A2) Associative : a•(b•c)=(a•b)•c for all a, b, c in G.

(A3) Identity element : There is an element e in G such that a•e = e•a = a for all a in G.

(A4) Inverse element : For each a in G there is an element a' in G such that

$$a \bullet a' = a' \bullet a = e.$$

(A5) Commutative : a•b = b•a for all a, b in G.

- A number of public-key ciphers are based on the use of an abelian group. For example, Diffie-Hellman key exchange involves multiplying pairs of nonzero integers modulo a prime number q.
- Keys are generated by exponentiation over the group, with exponentiation defined as repeated multiplication.
- For elliptic curve cryptography, an operation over elliptic curves, called addition, is used. Multiplication is defined by repeated addition. For example,

$$a \times k = \underbrace{(a + a + \cdots + a)}_{k \text{ times}}$$

where the addition is performed over an elliptic curve. Cryptanalysis involves determining k given a and $(a \bullet k)$.

3.10.2 Elliptic Curves over Real Numbers

- An elliptic curve over real numbers may be defined as the set of points (x, y) which satisfy an elliptic curve equation of the form :

$$y^2 = x^3 + ax + b,$$ where x, y, a and b are real numbers.

- Each choice of the numbers a and b yields a different elliptic curve. For example, a = – 4 and b = 0.67 gives the elliptic curve with equation $y^2 = x^3 - 4x + 0.67$; the graph of this curve is shown below :

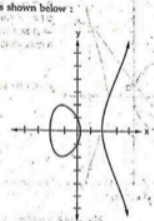


Fig. 3.10.1

- If $x^3 + ax + b$ contains no repeated factors or equivalently if $4a^3 + 27b^2$ is not 0, then the elliptic curve $y^2 = x^3 + ax + b$ can be used to form a group. An elliptic curve group over real numbers consists of the points on the corresponding elliptic curve, together with a special point O called the point at infinity.

  P + Q = R is the additive property defined geometrically.

- Elliptic curve groups are additive groups; that is, their basic function is addition. The addition of two points in an elliptic curve is defined geometrically.
- The negative of a point P = (x_P, y_P) is its reflection in the x-axis : the point-P is (x_P, – y_P). Notice that for each point P on an elliptic curve, the point-P is also on the curve.
- **Adding distinct points P and Q :** Suppose that P and Q are two distinct points on an elliptic curve and the P is not – Q. To add the points P and Q, a line is drawn through the two points. This line will intersect the elliptic curve in exactly one more point, call – R. The point – R is reflected in the x-axis to the point R. The law for addition in an elliptic curve group is P + Q = R.

- For example :

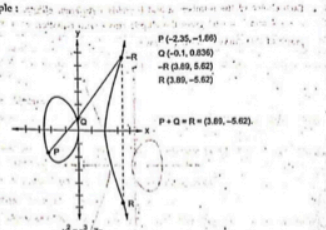


Fig. 3.10.2

3.11 Elliptic Curve Cryptography

AU : Dec.-20

- An elliptic curve is a set of points on the cooridnate plane satisfying an equation of the form $y^2 + axy + by = x^3 + cx^2 + dx + e$. In order to use elliptic curves for say, Diffie-Hellman, there needs to be some mathematical operation on two points in the set that will always produce a point also in the set.
- ECC can be done with at least two types of arithmetic, each of which gives different definitions of multiplication. The two types of arithmetic are
  1. Zp arithmetic,
  2. GF(2^n) arithmetic, which can be done with shifts and ⊕ s.
- To form a cryptographic system using elliptic curves, we need to find a hard problem corresponding to factoring the product of two primes or taking the discrete logarithm.
- Consider the equation Q = KP where Q, P$\in E_p(a, b)$ and K < P. It is relatively easy to calculate Q given K and P, but it is relatively hard to determine K given Q and P. This is called the **discrete logarithm problem for elliptic curves.**

Example 3.11.1 *Consider the group $E_{23}(9, 17)$. This is the group defined by the equation $y^2 \bmod 23 = (x^3 + 9x + 17) \bmod 23$. What is the discrete logarithm K of Q = (4, 5) to the base P = (16, 5) ?*

Solution : The brute-force method is to compute multiples of P until Q is found.

Thus,

$P = (16, 5)$

$2P = (20, 20)$

$3P = (14, 14)$

$4P = (19, 20)$

$5P = (13, 10)$

$6P = (7, 3)$

$7P = (8, 7)$

$8P = (12, 17)$

$9P = (4, 5)$

Because $9P = (4, 5) = Q$, the discrete logarithm $Q = (4, 5)$ to the base $P = (16, 5)$ is $K = 9$.

Analog of Diffie-Hellman key exchange

A key exchange between users A and B can be accomplished as follows :

1. A selects an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A \times G$; the public key is a point in E_q (a, b)
2. B similarly selects a private key n_B and computes a public key P_B.
3. A generates the secret key $K = n_A \times P_B$.

   B generates the secret key $K = n_B \times P_A$.

The two calculations in step 3 produce the same result because

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G)$$
$$= n_B \times P_A$$

Elliptic curve encryption and decryption

- For an encryption/decryption, system requires a point G and an elliptic group $E_q(a, b)$ as parameters. Each user A selects a private key n_A and generates a public key $P_A = n_A \times G$.
- To encrypt and send message P_m to user B, A chooses a random positive integer K and produces the ciphertext C_m consisting of the pair of points

$$C_m = \{KG, P_m + KP_B\}$$

- To decrypt the ciphertext, B multiplies the first point in the pair by B's secret key and subtracts the result from the second point

$$P_m + KP_B - n_B(KG)$$
$$= P_m + K(n_B G) - n_B(KG) = P_m$$

Review Questions

1. *With a neat sketch, explain the Elliptic curve cryptography with an example.*

AU : Dec.-20, Marks 8

3.12 Two Marks Questions with Answers

Q.1 **User X and Y exchange the key using Diffie-Hellmann algorithm. Assume a = 5, q = 11, XA = 2, XB = 3. Find the value of YA, YB and k.** *(Refer section 3.9)*

AU : Dec.-22

Q.2 **Draw functional diagram of RSA based digital signature.** *(Refer section 3.7)*

AU : Dec.-21

Q.3 **Find atleast two points lies in the elliptic curve $y^2 = x^3 + 2x + 3 (mod 5)$.**

AU : Dec.-21

Ans. : For x = 1, we have y^2 = 6 = 1 (mod 5). We can again check all the residues modulo 5 to find the square roots of 1. We have :

0^2 = 0 (mod 5)

1^2 = 1 (mod 5)

2^2 = 4 (mod 5)

3^2 = 4 (mod 5)

4^2 = 1 (mod 5)

Therefore, we have two solutions for y when x = 1: (1, 2) and (1, 3).

Hence, two points on the elliptic curve are (1, 2) and (1, 3).

Q.4 **Using Fermat's theorem, check whether 19 is prime or not.? Consider a is 7.**

AU : Dec.-21

Ans. : Fermat's theorem : a^(p-1) = 1 (mod p)

In this case, we have a = 7 and p = 19. Therefore, we can check whether 19 is prime or not by verifying whether:

7^(19-1) = 1 (mod 19)

Using the exponentiation by squaring method, we can calculate that: 7^18 = 133024179815834904B1

Since 133024179815834904 81 = 1 (mod 19), we can conclude that 19 is prime according to Fermat's Theorem.

Q.5 For p = 11 and q = 19 and choose d = 17. Apply RSA algorithm where Cipher message = 80 and thus find the plain text. (Refer section 3.7) AU : Dec-20

Q.6 What is the use of Fermat's theorem ? AU : May-11, IT

Ans. : Fermat's theorem sometimes is helpful for quickly finding a solution to some exponentiations and multiplicative inverses when the modulus is a prime.

Q.7 State Euler's theorem with example. AU : May-11, 14, 18, IT

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

$$a^{\phi(n)} = 1 \bmod n$$

Proof : Let Z_n be the set of integers that are less than n and are relatively prime to n.

If $a \in Z_n$ we have

1. $(ax \bmod n) \in Z_n$
2. If $x, y \in Z_n$ and $x \neq y$ then $(ax \bmod p) \neq (ay \bmod p)$

So, $$\prod_{x \in Z_n} x = \prod_{x \in Z_n} (ax \bmod p) = \left(a^{\phi-1} \bmod p\right) \prod_{x \in Z_n} x$$

Q.8 Define : Finite field. AU : May-12, CSE/IT

Ans. : Finite field is a field that contains a finite number of elements. The finite fields are classified by size; there is exactly one finite field up to isomorphism of size p^k for each prime p and positive integer k.

Q.9 Find GCD (21,300) using Euclid's algorithm. AU : May-13, CSE/IT

Ans. : GCD (21, 300)

$300 = 14 \times 21 + 6$

$21 = 3 \times 6 + 3$

$6 = 2 \times 3 + 0$

Therefore GCD = 3

Q.10 Find 11^7 mod 13. AU : May-15

Ans. :

Step 1 : $11^2 = 121 \equiv 4 \pmod{13}$

Step 2 : $11^4 = (11^2)^2 \equiv 42 \equiv 3 \pmod{13}$

Step 3 : $11^7 = 11 \times 4 \times 3 = 132 \equiv 2 \pmod{13}$

Q.11 What is discrete logarithm ? AU : May-11, 13 CSE/IT

Ans. : The logarithm of a number is defined to be the power to which some positive base (except 1) must be raised in order to equal that number. If working with modulo arithmetic, and the base is a primitive root, then an integral discrete logarithm exists for any residue.

Q.12 What is discrete logarithm problem ? AU : May-16

Ans. : Discrete Logarithm problem (DLP) is easy to perform and hard to reverse. The strength of one way function is based on time needed to reverse it.

Let be a cyclic finite group and g ∈ G be a generator of G. The DLP in G is following :

Given an element h ∈ G, find the smallest positive integer x such that

$h = [x]g$ (additive group)

$h = g^x$ (multiplicative group)

x is denoted as :

$x = D \log(h)$

Q.13 Find gcd (1970, 1066) using Euclid's algorithm. AU : Dec-16, CS/IT

Ans. :

$1970 = 1 \times 1066 + 904$	gcd(1066, 904)
$1066 = 1 \times 904 + 162$	gcd(904, 162)
$904 = 5 \times 162 + 94$	gcd(162, 94)
$162 = 1 \times 94 + 68$	gcd(94, 68)
$94 = 1 \times 68 + 26$	gcd(68, 26)
$68 = 2 \times 26 + 16$	gcd(26, 16)
$26 = 1 \times 16 + 10$	gcd(16, 10)
$16 = 1 \times 10 + 6$	gcd(10, 6)
$10 = 1 \times 6 + 4$	gcd(6, 4)
$6 = 1 \times 4 + 2$	gcd(4, 2)
$4 = 2 \times 2 + 0$	gcd(2, 0)

In this case GCD (1970,1066) = 2.

Q.14 State Fermat's theorem. AU : May-17, Dec.-17

Ans. : If p is prime and a is positive integer not divisible by p, then $a^{p-1} \equiv 1 \bmod (p)$.

Proof :

Consider the set of positive integers less than p : [1, 2, ... , p - 1] and multiply each element by a, modulo p, to get the set X = [a mod p, 2a mod p,..... (p - 1)a mod p]. None of the elements of X is equal to zero because p does not divide a. Further more no two of the integers in X are equal.

Q.15 Determine the gcd (24140, 16762) using Euclid's algorithm. AU : May-17

Ans. : gcd (24140, 16762)

24140 =	1 × 16762 + 7378	gcd(16762, 7378)
16762 =	2 × 7378 + 2006	gcd(7378, 2006)
7378 =	3 × 2006 + 1360	gcd(2006, 1360)
2006 =	1 × 1360 + 646	gcd(1360, 646)
1360 =	2 × 646 + 68	gcd(646, 68)
646 =	9 × 68 + 34	gcd(68, 34)
68 =	2 × 34 + 0	gcd(34, 0)

In this case gcd (24140, 16762) = 34

Q.16 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.17 What is key distribution center ? AU : Dec.-15

Ans. : A key distribution center is responsible for distributing keys to pairs of users such as hosts, processes, applications. Each user must share a unique key with the key distribution center for purposes of key distribution.

Q.18 State the difference between conventional encryption and public-key encryption. AU : Dec.-11, 13, CSE/IT

Ans. : **Conventional encryption :** Same algorithm and same key is used for encryption and decryption. Sender and receiver must share the algorithm and key. Key must be kept secret. **For public-key encryption :** one algorithm is used for encryption and decryption with pair of keys. The sender and receiver must each have one of the matched pair of keys. One of two keys must be kept secret.

Q.19 What are Elliptic curve cryptosystems ?

Ans. : Elliptic curve cryptosystems are analogs of public-key cryptosystems such as RSA and ElGamal in which modular multiplication is replaced by the elliptic curve addition operation. The curves used in elliptic curve analogs of discrete logarithm cryptosystems are normally of the form $y^2 = x^3 + ax + b \pmod p$, where p is prime.

Q.20 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.21 What is an elliptic curve ? AU : Dec.-16, CSE/IT

Ans. : An elliptic curve over k is a nonsingular projective algebraic curve E of genus 1 over k with a chosen base point O ∈ E. For elliptic curve cryptography, an operation over elliptic curve, called addition is used. Multiplication is defined by repeated addition.

Q.22 Give the significance of hierarchical key control. AU : Dec.17

Ans. : A hierarchical scheme minimizes the effort involved in master key distribution, because most master keys are those shared by a local key distribution centre with its local entities. Furthermore, such a scheme limits the damage of a faulty or subverted key distribution centre to its local area only.

Q.23 Perform encryption and decryption using RSA algorithm for the following. P= 7; q = 11, e = 17; M=8. AU : May-18

Ans. : $n = p \times q = 7 \times 11 = 77$

$\phi(n) = (p-1)(q-1) = 60$

$e.d = 1 \bmod \phi(n) \rightarrow 17\ d = 1 \bmod (77)$

$d = 3$

encryption $C = M^e \bmod n = (8)^{17} \bmod 77 = 57$

decryption $M = C^d \bmod n = (57)^3 \bmod 77 = 8$

Q.24 List advantages and disadvantages of RSA.

Ans. : 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 :

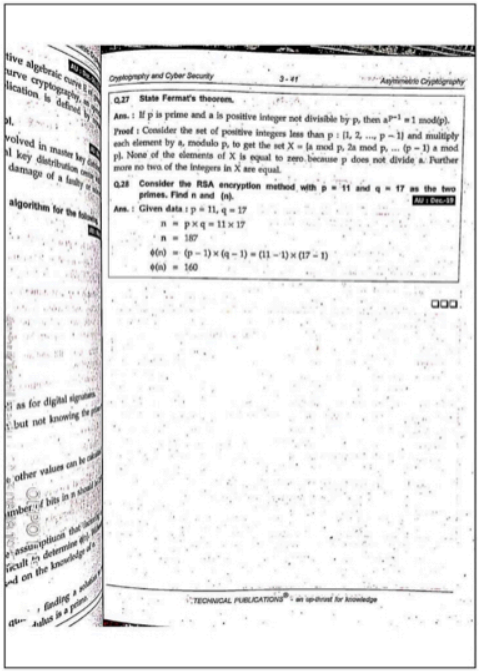
1. If any one of p, q, m, d is known, them the other values can be calculated. So secrecy is important.
2. To protect the encryption, the minimum number of bits in n should be 2048.

Q.25 Why RSA is secure ?

Ans. : • The premise behind RSA's security is the assumptiuon 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.

Q.26 What is the use of Fermat's theorem ?

Ans. : Fermat's theorem sometimes is helpful for quickly finding a solution to some exponentiations and multiplicative inverses when the modulus is a prime.



Q.27 State Fermat's theorem.

Ans. : If p is prime and a is positive integer not divisible by p, then $a^{p-1} \equiv 1 \bmod(p)$.

Proof : Consider the set of positive integers less than p : {1, 2, ..., p – 1} and multiply each element by a, modulo p, to get the set X = {a mod p, 2a mod p, ... (p – 1) a mod p}. None of the elements of X is equal to zero because p does not divide a. Further more no two of the integers in X are equal.

Q.28 Consider the RSA encryption method with p = 11 and q = 17 as the two primes. Find n and {n). AU : Dec.-15

Ans. : Given data : p = 11, q = 17

$n = p \times q = 11 \times 17$

$n = 187$

$\phi(n) = (p - 1) \times (q - 1) = (11 - 1) \times (17 - 1)$

$\phi(n) = 160$

□□□

UNIT IV

4 Integrity and Authentication Algorithms

Syllabus

Authentication requirement - Authentication function - MAC - Hash function - Security of hash function: HMAC, CMAC - SHA - Digital signature and authentication protocols - DSS - Schnorr Digital Signature Scheme - ElGamal cryptosystem - Entity Authentication : Biometrics, Passwords, Challenge Response protocols - Authentication applications - Kerberos MUTUAL TRUST : Key management and distribution - Symmetric key distribution using symmetric and asymmetric encryption - Distribution of public keys - X.509 Certificates.

Contents

4.1 Authentication and Authorization

Authentication

- 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.
- Data authentication means providing data integrity as well as that the data have been received from the individual who claimed to supply this information.

In authentication :

a. A 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 key.

b. Insufficient authentication occurs when a website permits an attacker to access sensitive content or functionality without having to properly authenticate.

c. Weak password recovery validation is when a website permits an attacker to illegally obtain, change or recover another user's password.

Authorization

- Authorization is a procedure of controlling the access of authenticated users to the system resources. An authorization system provides each user with exactly those rights granted to them by the administrator.
- Besides providing users with access rights to files, directories, printers etc, an authorization system might control user privileges, such as local access to the server, setting the system time, creating backup copies of the data and server shutdown.

In authorization :

a. Credential/session prediction is a method of hijacking or impersonating a website user.

b. Insufficient authorization is when a website permits access to sensitive content or functionality that should require increased access control restrictions.

c. Insufficient session expiration is when a website permits an attacker to reuse old session credentials or session IDs for authorization.

4.1.1 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.

4.1.2 Authentication Function

- Functions are at two levels in message authentication. At the lower level, function that produces an authenticator. These value is used to authenticate a message. The lower level function is used in the higher level authentication protocol. The higher level authentication protocol enables a receiver to verify the authenticity of message.
- Following are the some types of functions that may be used to produce an authenticator. They may be grouped into three classes.
  1. Message encryption. 2. Message Authentication Code (MAC)
  3. Hash function.

1) Message encryption

- Ciphertext of the entire message serves as its authenticator. Message encryption by itself can provide a measure of authentication.

Symmetric encryption

- Fig. 4.1.1 shows the uses of message encryption in symmetric encryption.
- A message M transmitted from source A to destination B is encrypted using a secret key K shared by A and B. If no other party knows the key, then confidentiality is provided.

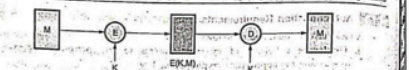


Fig. 4.1.1 Symmetric encryption (confidentiality and authentication)

- Destination B is assured that the message was generated by A. Because of secret key used by both party, it provides authentication as well as confidentiality.
- Given a decryption function D and a secret key K, the destination will accept any input X and produce output Y = D(K, X).
- If X is the ciphertext of a legitimate message M produced by the corresponding encryption function, then Y is some plaintext message M. Otherwise, Y will likely be a meaningless sequence of bits.
- For example, suppose that we are transmitting English language message using a caesar cipher with a shift of two A sends the following legitimate ciphertext :

  nbsftfbupbutboeptftfbupbutboemjuumfmbnct

  B decrypt to produce the following plaintext :

  lzqdrdzsnzsrzmcmddzsnzsrzmckhskdkalar
- If an opponent generates the following random sequences of letters :

  zuvrsoevgqxlzwigamdvnmhpmccxiuureosfbceb

  This decrypts to :

  Which does not fit the profile of ordinary English.

Public key encryption

- Public key encryption provides confidentiality but not authentication. Fig. 4.1.2 shows public key encryption with confidentiality in message encryption.
- Source A uses the public key PU_b of the destination B to encrypt message M. Because only B has the corresponding private key PR_b, only B can decrypt the message.

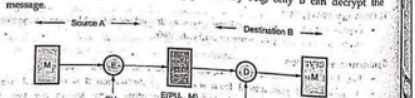


Fig. 4.1.2 Public key encryption (Confidentiality)

- This method provides no authentication because any opponent could also use B's public key to encrypt a message, claiming to be A.
- Fig. 4.1.3 shows the message encryption in public key encryption with authentication and signature.

Fig. 4.1.3 Public key encryption (Authentication)

- A uses its private key to encrypt the message, and B uses A's public key to decrypt.
- It provides authentication. The message must have come from A because A is the only party that possesses PR_a.
- It also provides digital signature. Only A could have constructed the ciphertext because only A possesses PR_a. Not even B, the recipient could have constructed the ciphertext.
- To provide both confidentiality and authentication, A can encrypt M first using its private key, which provides the digital signature and then using B's public key, which provides confidentiality.
- Fig. 4.1.4 shows confidentiality, authentication and signature for public key encryption.

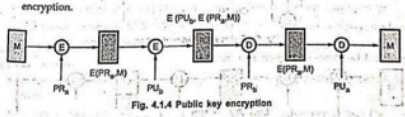


Fig. 4.1.4 Public key encryption

- It provides confidentiality because of PU_b.
- Provides authentication and signature because of PR_a.

2) Message Authentication Code (MAC)

- MAC 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 assured of the proper sequence because an attacker cannot successfully alter the sequence number.
- Fig. 4.1.5 shows the message authentication.
- Fig. 4.1.5 provides authentication but not confidentiality. Confidentiality can be provided by performing message encryption either after or before the MAC algorithm.

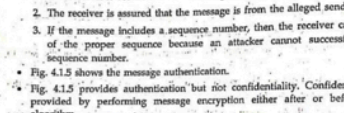


Fig. 4.1.5 Message authentication

- Fig. 4.1.6 shows encryption after the MAC.

Fig. 4.1.6 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.
- Fig. 4.1.7 shows the message authentication and confidentiality with encryption.

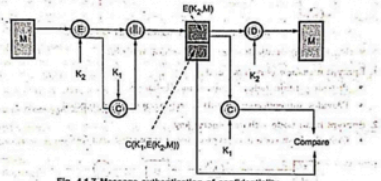


Fig. 4.1.7 Message authentication of confidentiality (authentication tied to ciphertext)

- Here also two separate keys are needed. The message is encrypted first. Then the MAC is calculated using the resulting ciphertext and is concatenated to the ciphertext to form the transmitted block.

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.

3) Hash function

- 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.
- 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. 4.1.8 (a) Encrypt message plus hash code

- Fig. 4.1.8 (a) shows th e basic uses of hash function.

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.

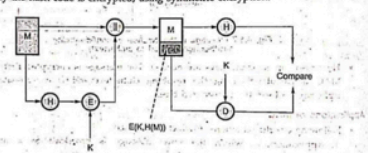


Fig. 4.1.8 (b) Encrypt hash code - shared secret key

- Reduces the processing burden for those applications that do not require confidentiality.

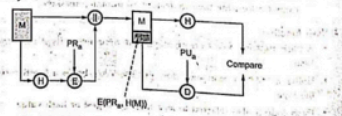


Fig. 4.1.8 (c) Encrypt hash code - sender's private key

3. Encrypt hash code - sender's private key.
   - Provides authentication and digital signature.

4.2 MAC

AU : Dec.-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.

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

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, $MAC_1 = C(K_1, M_1)$

Compute $MAC_i = C(K_i, M_1)$ for all 2^k keys

Number of matches = $2^{(k-n)}$

- **Round 2**

Given : M_2, $MAC_2 = C(K, M_2)$

Compute $MAC_i = C(K_i, M_2)$ for all $2^{(k-n)}$ keys resulting from Round 1

Number of matches = $2^{(k-2\times n)}$

- 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. 4.2.1 shows the data authentication algorithm.

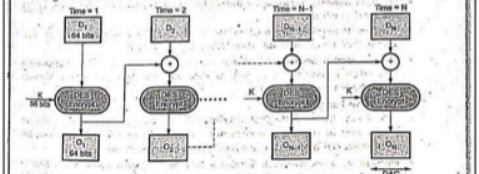


Fig. 4.2.1 Data authentication algorithm

- 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 : $D_1, D_2, D_3, \ldots\ldots\ldots, D_N$.

- Using the DES encryption algorithm (E) and a secret key (K), a data authentication code (DAC) is calculated as follows

$$O_1 = E(K, D_1)$$
$$O_2 = E(K, [D_2 \oplus O_1])$$
$$O_3 = E(K, [D_3 \oplus O_2])$$
$$\vdots$$
$$O_N = E(K, [D_N \oplus O_{N-1}])$$

- The DAC consists of either the entire block O_N or the leftmost M bits of the block with $16 \le M \le 64$.

Review Question

1. *Compare the uses of MAC and Hash function. Represent them using appropriate diagrams.* AU : Dec.-19, Marks 8

4.3 Hash Function

AU : May-18

- **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 digest.
- 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).

One - way Hash Function

- A one-way hash function, also known as a message digest, fingerprint or compression function, is a mathematical function which takes a variable-length input string and converts it into a fixed-length binary sequence.
- Furthermore, a one-way hash function is designed in such a way that it is hard to reverse the process, that is, to find a string that hashes to a given value (hence the name one-way.)
- 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.

Hash Function

- 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.

4.3.1 Requirements of 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 $y \neq x$ 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.**

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

$$C_i = b_{i1} \oplus b_{i2} \oplus b_{i3} \oplus \ldots\ldots \oplus b_{im}$$

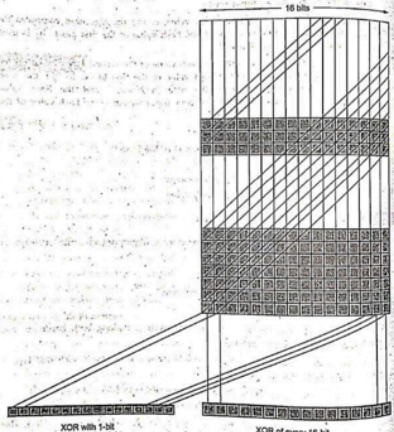


Fig. 4.3.1 Two simple hash functions

where C_i = i^{th} bit of the hash code, $1 \le i \le n$.

m = number of n-bit blocks in the input

b_{ij} = i^{th} bit in j^{th} block

$\oplus$ = XOR operation

- 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.

Fig. 4.3.1 shows two types of hash functions. (Refer Fig. 4.3.1 on previous page)

4.3.2 Applications 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.

4.3.3 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.

4.3.4 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.

4.3.5 Secure of Hash Function and HMAC

- 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 $y \neq x$ 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

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.

Review Question

1. How Hash function algorithm is designed ? Explain their features and properties.

AU : May-18, Marks 10

4.4 HMAC

AU : Dec.-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. 4.4.1 shows HMAC structure.

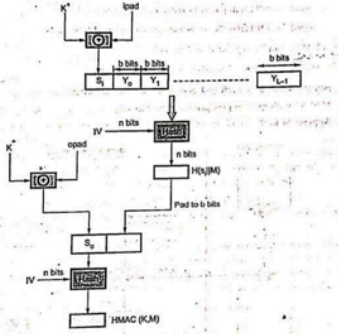


Fig. 4.4.1 HMAC structure

- 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, $0 \le i \le (L - 1)$
  - L = Number of blocks in M

- b = Number of bits in a block
- n = Length of hash code produced by embedded hash function.
- K = Secret key recommended length is ≥ 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.

Then HMAC can be expressed as follows :

HMAC (K, M) = H [($K^+ \oplus$ opad) || H[($K^+ \oplus$ ipad) || 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.

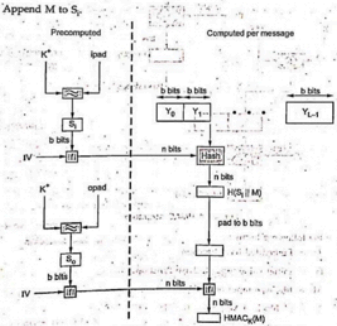


Fig. 4.4.2 Efficient implementation of HMAC

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. 4.4.2. Two quantities are precomputed :
  f(IV, (K+ ⊕ ipad))
  f(IV, (K+ ⊕ opad))
  Where f(CV, block) is the compression function for the hash function. (Refer Fig. 4.4.2 on previous page)

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 !!).

Review Question

1. *List the design objectives of HMAC and explain the algorithm in detail.* AU : Dec.-19, Marks 13

4.5 CMAC

- 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.
- 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. 4.5.1 shows calculation of CMAC.

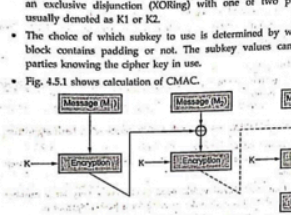


Fig. 4.5.1 Message length is integer multiple of block size

$C_1 = E(K, M_1)$

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

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

$\vdots$

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

$T = MSB_{Tlen}(C_n)$

where

T = message authentication code

Tlen = bit length of T

MSBs (X) = the s left most bits of the bit string X

4.6 SHA

AU : May-17

- 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 message 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 because 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 passiword 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 l, 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.

4.6.1 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

- Fig. 4.6.1 shows message digest generation using SHA-512.
- 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 digest. The input is processed in 1024-bit blocks.

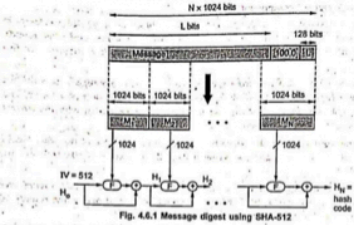


Fig. 4.6.1 Message digest using SHA-512

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 register (a, b, c, d, e, f, g, h). These registers are initialized to the following 64-bit integer (hexadecimal values)

Sr. No.	Register	Values
1.	a	6A09E667F3BCC908
2.	b	BB67AE8584CAA73B
3.	c	3C6EF372FE94F82B
4.	d	A54FF53A5F1D36F1
5.	e	510E527FADE682D1
6.	f	9B05688C2B3E6C1F
7.	g	1F83D9ABFB41BD6B
8.	h	5BE0CD19137E2179

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. 4.6.2 shows the processing of a single 1024-bit block.

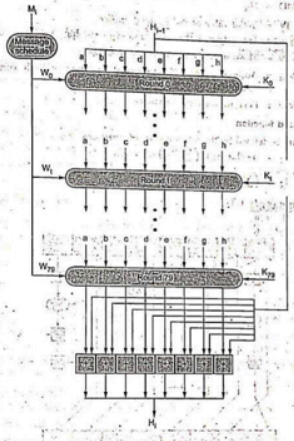


Fig. 4.6.2 SHA-512 processing of a single 1024-bit block

5. **Output :** The output from the N^{th} stage is the 512-bit message digest.
- The behaviour of SHA-512 is as-follows

$$H_0 = IV$$

$$H_i = SUM_{64}(H_{i-1}, abcdefgh_i)$$

$$MD = H_N$$

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.

$$T_1 = h + ch(e,f,g) + \left(\sum_1^{512} e\right) + W_t + K_t$$

$$T_2 = \left(\sum_0^{512} a\right) + Maj(a,b,c)$$

$a = T_1 + T_2$

$b = a$

$c = b$

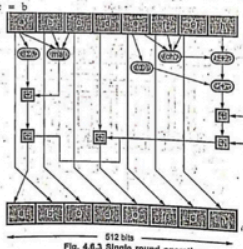


Fig. 4.6.3 Single round operation

d = c
e = d + T_i
f = e
g = f
h = g

Fig. 4.6.3 shows single round operation. [Refer Fig. 4.6.3 on previous page]

Example 4.6.1 *Compare the performance of RIPEMD - 160 algorithm and SHA - 1 algorithm.* **AU : May-17, Marks 16**

Solution : RIPEMD-160 verses SHA-1 :

- brute force attack harder (160 like SHA-1 vs 128 bits for MD5)
- not vulnerable to known attacks, like SHA-1 though stronger
- RIPEMD-160 is slower than SHA-1
- RIPEMD-160 is more secure than SHA-1 all designed as simple and compact
- SHA-1 optimised for big endian CPU's vs RIPEMD-160 optimised for little endian CPU's

4.7 Digital Signature **AU : Dec.-19,20,22**

- 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

4.7.1 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.

4.7.2 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.

4.7.3 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. 4.7.1 shows the DSS approach.

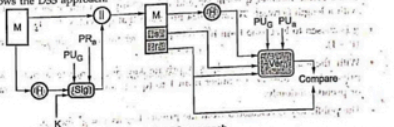


Fig. 4.7.1 DSS approach

- 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. 4.7.2 shows the RSA approach. In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length. This hash code is then encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted.

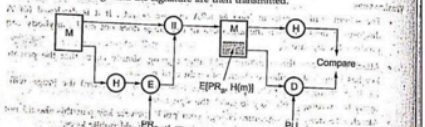


Fig. 4.7.2 RSA approach

- 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.

4.7.4 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 $h^{(P-1)/q} \bmod p$ 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 $y = g^x \bmod p$.

- 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. 4.7.3 shows the functions of signing and verifying.

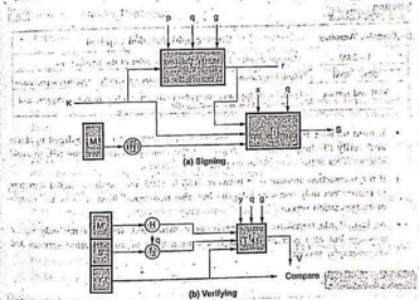


Fig. 4.7.3 Signing and verifying

Example 4.7.1 *Consider a banking application that is expected to provide cryptographic functionalities. Assume that this application is running on top of another application wherein the end customers can perform a single task of fund transfer. The application requires cryptographic requirements based on the amount of transfer.*

Transfer amount	Cryptography functions required
1 - 2000	Message digest
2001 - 5000	Digital signature
5000 and above	Digital signature and encryption

Suggest the security scheme to be adopted in client and server side to accommodate the above requirements and justify your recommendations. **AU : Dec.-19, Marks 13**

Solution :

Transfer Amount	Cryptography function required
1 - 2000	Message digest - To verify the finger print of the transactions
2001 - 5000	Digital signature - To ensure the message integrity and non-repudiation
5000 and service	Digital signature and encryption - To ensure the message integrity and non-repudiation and confidential

- If fund transfer amount is upto 2000, we simply require a message digest to obtain and verify the finger print or integrity of the message. Here we use SSL to avoid attacks. This is the example of cryptography services.
- If the transaction amount is in between 2000 to 5000, we require a digital signature to ensure not only message integrity but also non-repudiation. This is an example of authorization services.
- At last, if the transaction amount is more than 5000, we must not only sign a message but also encrypt it. This is a combination of authorization services and cryptography services.

Review Questions

1. *Consider a banking application that is expected to provide cryptographic functionalities. Assume that this application is running on top of another application wherein the end customers can perform a single task of fund transfer. The application requires cryptographic requirements based on the amount of transfer.*

Transfer amount	Cryptography functions required
1-2000	Message digest
2001-5000	Digital signature
5000 and above	Digital signature and encryption

Suggest the security scheme to be adopted in client and server side to accommodate the above requirements and justify your recommendations. **AU : Dec.-20, Marks 15**

2. *What is a digital signature ? Explain the key generation, signing and signature verification algorithm. Bring out the steps followed to create a digital signature.* **AU : Dec.-22, Marks 13**

4.8 Authentication Protocols **AU : Dec.-16,19**

- Authentication protocols are used to convince parties of each others identity and to exchange session keys. They may be one-way or mutual.
- 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.

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.

4.8.1 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.

4.8.1.1 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 password is 8 random character, it is impossible to crack the password. In order to eliminate guessable passwords four basic techniques are suggested.

1. User education
2. Computer generated password
3. Reactive password checking
4. Proactive password checking

4.8.1.2 Certificate based Authentication

- Client have a public key certificate. Fig. 4.8.1 shows the 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.

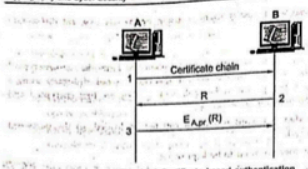


Fig. 4.8.1 Certificate based authentication

- 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.
- If they match, he concludes that A has used the private key corresponding to the public key in his certificate.

4.8.2 Mutual Authentication

It is often necessary for both communicating themselves to each other.

4.8.2.1 Based on a 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 spedal way and then returns a result. This type of protocols are called challenge-response protocols.
- The working of this protocol is as follows.
- 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.

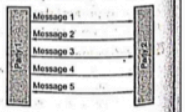


Fig. 4.8.2 Two way authentication using a challenge-response protocol

- 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.

4.8.2 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.

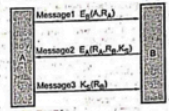


Fig. 4.8.3 Authentication using public key cryptography

4.8.3 Needham Schroeder Protocol

- 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's 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.

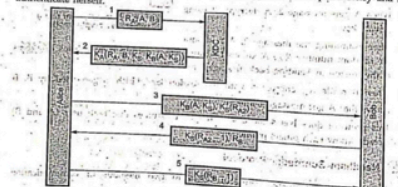


Fig. 4.8.4 Needham Schroeder authentication protocol

Review Questions

1. *Discuss client server mutual authentication, with example flow diagram.*
2. *Suggest and explain about an authentication scheme for mutual authentication between the user and the server which relies on symmetric encryption.* AU : Dec.-19, Marks 15

AU : Dec.-16, Marks 16

4.9 Schnorr Signature

AU : Dec.-17

- 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 $a^q = 1 \bmod p$. 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: $e = H(M \mid\mid x)$
  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 $e = H(M \mid\mid x')$.

Example 4.9.1 *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 $h(40067) = 81$

$S_1 = h(M \mid\mid e_1^r \bmod p) = h(400 \mid\mid 9^{11} \bmod p)$

$= h(400 \mid\mid 67) = h(40067) = 81$

$S^2 = r + ds_1 \bmod q$

$= 11 + 23 \times 81 \bmod 83 = 48$

Review Question

1. *Write down the steps involved in Schnorr digital signature scheme used for authenticating a person.* AU : Dec.-17, Marks 8

4.10 Digital Signature Scheme

AU : May-15, Dec.-17,18

- Signing messages digitally which is functionally equivalent to a physical signature, but which is at least as resistant to forgery as its physical counterpart. Schemes which provide this functionality are called **digital signature schemes.**
- A Digital Signature Scheme will have two components, a private signing algorithm which permits a user to securely sign a message and a public verification algorithm which permits anyone to verify that the signature is authentic.
- The signing algorithm needs to "bind" a signature to a message in such a way that the signature cannot be pulled out and used to sign another document, or have the original message modified and the signature remain valid.

4.10.1 ElGamal Cryptosystem

- 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	$7^3 \bmod 11 = 2$	$0 \times 5^3 \bmod 11 = 0$
1	$7^3 \bmod 11 = 2$	$1 \times 5^3 \bmod 11 = 4$

- 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 :

$(2, 4) : m = 4(2)^{-2} = 4(4)^{-1} = 12 \bmod 11 = 1$ (4 and 3 are inverse mod 11)

$(2, 8) : m = 8(2)^{-2} = 8(4)^{-1} = 24 \bmod 11 = 2$

$(2, 0) : m = 0(2)^{-2} = 0(4)^{-1} = 0 \bmod 11 = 0$

$(2, 4) : m = 1$

Alice reassembles the message into the letter "a".

Review Questions

1. *Explain ElGamal public key cryptosystems with an example.* **AU : May-15, Marks 16**
2. *Write down the steps involved in Elgamal digital signature scheme used for authenticating a person.* **AU : Dec.-17, Marks 8**
3. *Explain Elgamal digital signature scheme.* **AU : Dec.-18, Marks 13**

4.11 Entity Authentication

- Entity authentication is the process by which one entity (the verifier) is assured of the identity of a second entity (the claimant) that is participating in a protocol.

- This assurance is usually obtained by requiring the claimant to provide corroborating evidence of the claimed identity to the verifier. The claimed identity can either be presented to the verifier as part of the protocol or can be presumed by context.
- The term identification is sometimes used as a synonym for entity authentication, however it is also sometimes used to simply refer to the process of claiming or stating an identity without providing the corroborating evidence required for entity authentication.
- **Properties of entity protocol :**
  1. **Reciprocity of identification :** Both claimant and verifier may prove their identities to other providing unilateral or mutual identification.
  2. **Computational efficiency :** The number of operations required to execute a protocol.
  3. **Communication efficiency :** Bandwidth required or number of passes.
  4. **Third-part involvement :** E.g. Online trusted third party to distribute common symmetric keys.
  5. **Security guarantees :** E.g. Provable security and zero knowledge security.
  6. **Storage of secrets :** Location and method used.

4.11.1 Biometrics Authentication

- Biometric authentication is simply the process of verifying your identity using your measurements or other unique characteristics of your body, then logging you in a service, an app, a device and so on.
- Biometric identification verifies you are you based on your body measurements.
- Biometric identification systems can be grouped based on the main physical characteristic that lends itself to biometric identification.
- **Fingerprint identification :** Fingerprint ridges are formed in the womb; you have fingerprints by the fourth month of fetal development. Once formed, fingerprint ridges are like a picture on the surface of a balloon. As the person ages, the fingers do get larger. However, the relationship between the ridges stays the same, just like the picture on a balloon is still recognizable as the balloon is inflated.
- **Hand geometry :** Hand geometry is the measurement and comparison of the different physical characteristic of the hand. Although hand geometry does not have the same degree of permanence or individuality as some other characteristics, it is still a popular means of biometric authentication.

- **Retina scan :** A retina scan provides an analysis of the capillary blood vessels located in the back of the eye; the pattern remains the same throughout life. A scan uses a low intensity light to take an image of the pattern formed by the blood vessels. Retina scans were first suggested in the 1930's.
- **Iris scan :** An iris scan provides an analysis of the rings, furrows and freckles in the colored ring that surrounds the pupil of the eye. More than 200 points are used for comparison.
- **Face recognition :** Facial characteristics are depends on the size and shape of facial characteristics and their relationship to each other. Although this method is the one that human beings have always used with each other, it is not easy to automate it. Typically, this method uses relative distances between common landmarks on the face to generate a unique "faceprint".
- **Signature :** Although the way you sign your name does change over time and can be consciously changed to some extent, it provides a basic means of identification.
- **Voice analysis :** The analysis of the pitch, tone, cadence and frequency of a person's voice.

4.11.2 Password

- Password authentication is also called weak authentication.
- This is amongst the most conventional schemes where in a user has an "user id" and a "password". User id acts like a claim and password as evidence supporting the claim.
- The system checks to see if it matches or not. Here demonstration of knowledge of the secret which is password in this case; corroborates that the person is verified.

Advantages :

1. It has better entropy than a short password.
2. It is easier to remember than the usual passwords.

Disadvantages :

1. This is really weak against attacks as intruder can hear over communication channel and impersonate it later.
2. It is also very easy to replay the same message and use it later.
3. Exhaustive search or password guessing i.e. dictionary attacks can also be used. One has to type extra.

4.11.3 Challenge-Response Identification

- It is also called **strong authentication.**
- The central idea of challenge response is that claimant proves its identity to verifier by demonstrating knowledge of a secret known to be associated with entity without revealing the secret itself to the verifier during the protocol.
- The challenge is usually time variant and is random number.
- As every time the challenge is different, even if the adversary is monitoring the network it won't help as challenge changes every time.
- Challenge-response authentication uses a cryptographic protocol that allows to prove that the user knows the password without revealing the password itself. Using this method, the application first obtains a random challenge from the server.
- It then computes the response by applying a cryptographic hash function to the server challenge combined with the user's password.
- Finally, the application sends the response along with the original challenge back to the server. Because of the "one-way" properties of the hash function, it is impossible to recover the password from the response sent by the application.
- Upon receiving the response, the server applies the same hash function to the challenge combined with its own copy of the user's password. If the resulting value matches the response sent by the application, this indicates with a very high degree of probability that the user has submitted the correct password.
- Challenge-response by symmetric-key techniques. Here, A is the claimant and B is the verifier. The communication takes place as,

$$A \leftarrow B : r_B$$

$$A \rightarrow B : E_k(r_B, B^*)$$

- B sends A a random number. To prove its claim, A then encrypts the random number send by B using the symmetric encryption key k.
- It also sends the optional field of the verifier as B. This prevents reflection attack as the key used is bi-directional key k.
- B then decrypts the message sent by A to see the random number is the same as it had sent. It also sees if the identifier matches. If either of them is not true it stops any further communication.

4.12 Authentication Applications - Kerberos AU : May-14,15,18,19, Dec.-21

- 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.

4.12.1 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.

4.12.2 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.

4.12.2.1 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 password 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
Password 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 password and sends message to AS including user ID + server ID + user password. 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 authenticity of data then grants the requested service. In above hypothetical dialogue, symbol || represents concatenation.

4.12.2.2 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.	C → AS
	2.	AS → C
ii	3.	C → TGS (Ticket-granting server)
	4.	TGS → C
iii	5.	C → V
		V → C

- Fig. 4.12.1 shows how the steps are executed in Kerberos version 4.

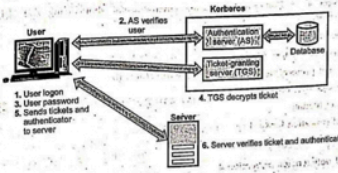


Fig. 4.12.1 Overview of kerberos

4.12.2.3 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.

4.12.3 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

4.12.3.1 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 4.10.1 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 4.10.1

- 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	

4.12.4 Comparison between Kerberos Versions 4 and 5

Parameters	Kerberos Versions 4	Kerberos Versions 5
Encryption algorithms used	DES only	DES and other encryptions
Ticket lifetime	5 min units, Maximum = 1280 minutes	Start and end time is arbitrary
Message byte ordering	Tagged message with ordering	Abstract syntax notation on basis encoding rules.
Password attack	Initial request in clear and use it for offline attack.	Need to send pre-authentication data
Two times encryption	Supported	Not supported
Session Keys	Replay risk using repeated ticket	Sub session key once only
Hierarchy of Realms	Limits to pairs	Transition allowed

4.12.5 Strengths of Kerberos

1. Passwords are never sent across the network unencrypted. This prevents those unscrupulous people from being able to read the most important data sent over the network.
2. Clients and applications services mutually authenticate. Mutual authentication allows for both ends to know that they truly know whom they are communicating with.
3. Tickets have a limited lifetime, so if they are stolen, unauthorized use is limited to the time frame that the ticket is valid.
4. Authentication through the AS only has to happen once. This makes the security of Kerberos more convenient.

5. Shared secret keys between clients and services are more efficient than public-keys.
6. Many implementations of Kerberos have a large support base and have been put through serious testing.
7. Authenticators, created by clients, can only be used once. This feature prevents the use of stolen authenticators.

4.12.6 Weakness of Kerberos

1. Kerberos only provides authentication for clients and services.
2. Kerberos 4 uses DES, which has been shown to be vulnerable to brute-force-attacks with little computing power.
3. The principal-key database on the KDC has to be hardened or else bad things can happen.
4. Like any security tool, it is also vulnerable to users making poor password choices.
5. Kerberos doesn't work well in a time-sharing environment.
6. Kerberos requires a continuously available Kerberos Server. If the Kerberos Server goes down, the Kerberos network is unusable.
7. Kerberos does not protect against modifications to system software like Trojan horses.

4.12.7 Difference between Kerberos and SSL

No.	Kerberos	SSL
1)	Uses private key encryption.	Uses public key encryption.
2)	Based on the trusted third party.	Based on certificates.
3)	Ideal for network environment.	Ideal for the WWW.
4)	Key revocation can be accomplished by disabling a user at the authentication server	Key revocation requires revocation server to keep track of bad certificate.
5)	Password resides in user's minds where they are usually not subject to secret attack.	Certificates sit on user's hard drive where they are subject to being cracked.
6)	Kerberos open source and free available.	Uses patented material, so the service is not free.

Review Questions

1. *Elaborately explain Kerberos authentication mechanism with suitable diagrams.* AU : May-14, Marks 16
2. *Explain Kerberos Version 4 in detail.* AU : May-15, Marks 16
3. *Explain briefly about the architecture and certification mechanisms in Kerberos and X.509.* AU : May-18, Marks 16
4. *What is Kerberos ? Explain how it provides authenticated service.* AU : May-19, Marks 7
5. *Discuss the four requirements of Kerberos.* AU : Dec.-21, Marks 4

4.13 Mutual Trust : Key Management and Distribution

- The purpose of public key cryptography is,
  1. The distribution of public keys.
  2. The use of public key encryption to distribute secret keys.

4.13.1 Distribution of Public Keys

- Different methods have been proposed for the distribution of public keys. These 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. 4.13.1 shows the public key distribution.

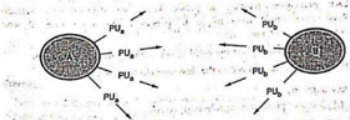


Fig. 4.13.1 Public key distribution

- 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 newsgroups 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. 4.13.2 shows public key publication.

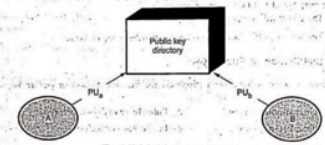


Fig. 4.13.2 Public key publication

- 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. 4.13.3 shows public key distribution scenario.
- 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.

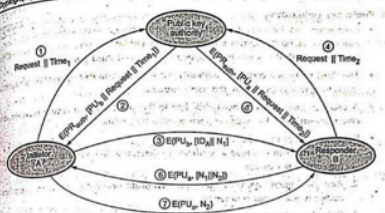


Fig. 4.13.3 Public key distribution scenario

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.

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. 4.13.4. Each participant applies to the certificate authority, supplying a public key and requesting a certificate.

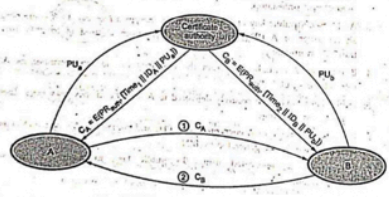


Fig. 4.13.4 Exchange of public key certificates

- 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.

4.13.2 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. 4.13.5 shows use of public key encryption.

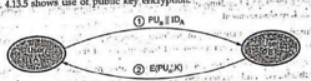


Fig. 4.13.5 Use of public key encryption

- 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. 4.13.6 shows the public key distribution of secret keys. (Refer Fig. 4.13.6 on next page)
- 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.

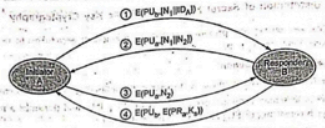


Fig. 4.13.6 Public key distribution of secret keys

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.

4.13.3 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.
- 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. 4.13.7 shows the hierarchy of certificate authorities. (Refer Fig. 4.13.7 on next page)
- In the Fig. 4.13.7, 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.

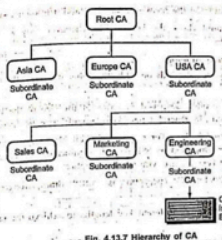


Fig. 4.13.7 Hierarchy of CA

- 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.

4.13.4 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. 4.13.8 shows the use of a key hierarchy.

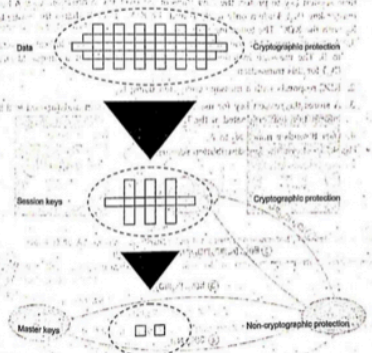


Fig. 4.13.8 Use of a key hierarchy

- 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.

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. 4.13.9 shows the key distribution scenario.

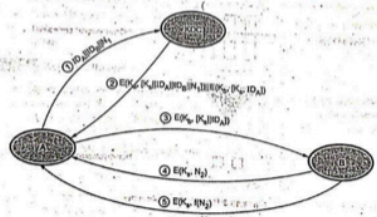


Fig. 4.13.9 Key distribution scenario

- 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. 4.13.10 shows automatic key distribution for connection - oriented protocol.

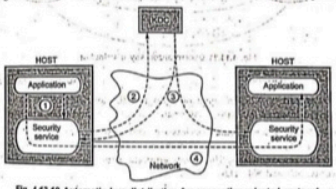


Fig. 4.13.10 Automatic key distribution for connection-oriented protocol

- 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

- 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.

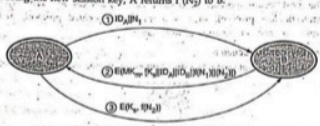


Fig. 4.13.11 Decentralized key distribution

4.14 X.509 Certificates

AU : May-18, 19, Dec.-21

- 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.

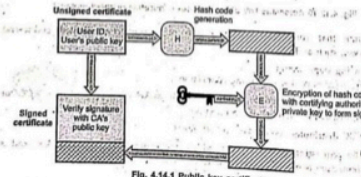


Fig. 4.14.1 Public key certificate

- X.509 standard uses RSA algorithm and hash function for digital signature. Fig. 4.14.1 shows generation of public key certificate. (Refer Fig. 4.14.1 on previous page)

4.14.1 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. 4.14.2.

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. 4.14.2 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

$CA\langle\langle A\rangle\rangle = CA\{V, SN, AI, CA, T_A A, A_P\}$

where,

$CA\langle\langle A\rangle\rangle$ indicates the certificate of user A issued by certification authority CA.

$CA\{V \ldots\ldots A_P\}$ indicates signing of $V \ldots\ldots A_P$ by CA.

4.14.2 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.

4.14.3 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. 4.14.3. Certificate revocation list. Each certificate has unique serial number of identify the certificate.

Signature algorithm identifier
Issuer name
Latest update
Next update
User certificate serial
Revoked certificate
Revocation date
Signature

Fig. 4.14.3 Certificate revocation list

4.14.4 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. 4.14.4.

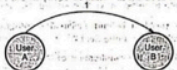


Fig. 4.14.4 One way authentication

2. Two-way authentication

- Two-way authentication allows both parties to communicate and verify the identity of the user.

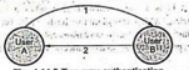


Fig. 4.14.5 Two-way authentication

3. Three-way authentication

- Three-way authentication is used where synchronized clockes are not available Fig. 4.14.6 shows three-way authentication.

Fig. 4.14.6 Three-way authentication

Review Questions

1. *Explain briefly about the architecture and certification mechanisms in Kerberos and X.509.* **AU : May-18, Marks 16**
2. *Explain the format of the X.509 certificate.* **AU : May-19, Marks 6**
3. *Shortly describe about the elements of X509 certificate.* **AU : Dec.-21, Marks 9**

4.15 Two Marks Questions with Answers

Q.1 Differentiate MAC and Hash function. **AU : Dec.-22**

Ans. : The major difference between hash and MAC is that MAC uses secret key during the compression. Unlike a MAC, a hash code does not use a key but is a function only of the input message.

Q.2 Name the four requirements defined by Kerberos. **AU : Dec.-22**

Ans. : Kerberos requirements are secure, reliable, transparents and scalable.

Q.3 What mathematical problem is behind security of the ElGamal cryptosystem ? **AU : Dec.-21**

Ans. : The mathematical problem behind ElGamal cryptography is the difficulty of computing discrete logarithms in a finite field.

Q.4 What is MAC ? Mention the requirement of MAC. **AU : Dec.-20**

Ans. : An alternative authentication technique involves the use of a small fixed size block of data, known as a cryptographic checksum or MAC that is appended to the message.

Q.5 What is a Hash in cryptography ? **AU : May-18**

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.

Q.6 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.7 What two levels of functionality comprise a message authentication or digital signature mechanism ?

Ans. : Two levels of functionality comprise a message authentication or digital signature mechanisms are Low-level authentication and Higher-level authentication. At the lower level there must be some sort of function that produces an authenticator : a value to be used to authenticate a message. This lower level function is then used as primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of message.

Q.8 What is a message authentication code ?

Ans. : An alternative authentication technique involves the use of a small fixed size block of data, known as a cryptographic checksum or MAC that is appended to the message.

Q.9 What is the difference between a message authentication code and a one-way hash function ?

Ans. : The difference between a MAC and a one-way hash function is that unlike a MAC, a hash code does not use a key but is a function only of the input message.

Q.10 Is it necessary to recover the secret key in order to attack a MAC algorithm ?

Ans. : A number of keys will produce the correct MAC and the opponent has no way of knowing which the correct key is. On an average $2^n - 1$ keys produce a match. Therefore attacks do not require the discovery of the key.

Q.11 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.12 What is public-key certificate ?

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

Q.13 What are the requirements for the use of a public-key certificate scheme ?

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

Q.14 What is the life cycle of a key ?

Ans. : Keys have limited lifetimes for a number of reasons. The most important reason is protection against cryptanalysis. Each time the key is used, it generates a number of ciphertexts. Ford describes the life cycle of a key as follows :
- key generation and possibly registration for a public key.
- key distribution
- key activation/deactivation
- key replacement or key update

- key revocation
- key termination, involving destruction and possibly archival.

Q.15 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.16 What is a birthday attack ? AU : May-11, 14, IT

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 ½; such a result is called a birthday paradox.

Q.17 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.18 What is digital signature ? AU : May-11,15, IT

Ans. : Digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.

Q.19 What is one-way property ? AU : Dec.-12, CSE/IT

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.20 What are the two approaches of digital signature ? AU : Dec.-12, CSE/IT

Ans. : Two approaches of digital signature are RSA approach and DSS approaches.

Q.21 Write any two differences between MD4 and secure hash algorithm. AU : Dec.-13, CSE/IT

Ans. :

Sr. No.	MD4	SHA
1.	Pad message so its length is 448 mod 512.	Pad message so its length is a multiple of 512 bits.
2.	Initialise the 4-word (128-bit) buffer (A, B, C, D)	Initialise 5-bit (160 bit) buffer (A, B, C, D, E)
3.	Process the message in 16-word chunks using 3 rounds of 16-bit operations each on chunk and buffer	Process the message in 16-word chunks using 4 rounds of 20-bit operations.

Q.22 Define password protection.

Ans. : Password protection is the front line protection against intruder to the system. A password authenticate the ID and provides security to the system.

Q.23 List the authentication requirements. AU : May-14

Ans. : 1. Disclosure 2. Traffic analysis 3. Masquerade 4. Sequence modification
5. Content modification 6. Timing modification
7. Source repudiation 8. Destination repudiation

Q.24 What are the security services provided by digital signature ? AU : Dec.-14

Ans. : Security services provided by Digital Signature are message authentication, message integrity and Non-repudiation. Message authentication is a mechanism or service used to verify the integrity of a message. Integrity ensures that information is not changed or altered in transit. Non-repudiation 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.

Q.25 Name the authentication protocols. AU : Dec.-15

Ans. : Kerberos is an authentication protocol. It provides a way to authenticate clients to services to each other through a trusted third party.

Q.26 List four requirements that were defined for kerberos. AU : Dec.-15

Ans. : Requirement of Kerberos: Security, Reliability, Transparency and Scalability.

Q.27 List any four password selection strategies. AU : Dec.-15

Ans. : • In order to eliminate guessable passwords four basic techniques are suggested.

1. User education
2. Computer generated password
3. Reactive password checking
4. Proactive password checking

Q.28 State any three requirement for authentication. AU : Dec.-16, CSE/IT

Ans. : a) **Sequence modification :** Any modification to a sequence of message between parties, including insertion, deletion and reordering.
b) **Content modification :** Changes to the contents of a message, including insertion, deletion, transposition and modification.
d) **Timing modification :** Delay or replay of messages.

Q.29 Differentiate MAC and hash function. AU : Dec.-16, CSE/IT

Ans. : The major difference between hash and MAC is that MAC uses secret key during the compression. Unlike a MAC, a hash code does not use a key but is a function only of the input message.

Q.30 What is the role of compression function in hash function ? AU : May-17

Ans. : A compression function takes a fixed length input and returns a shorter, fixed-length output.

Q.31 Specify the various types of authentication protocol. AU : May-17

Ans. : Authentication protocols are Mutual vs one-way authentications, symmetric vs public-key approaches, Needham Schroeder protocol.

Q.32 How is the security of MAC function expressed ? AU : Dec.-17

Ans. : Security of MAC functions :

- The security of any HMAC function based on the cryptographic strength of the underlying hash function.
- The security of a MAC function expressed in terms of the probability of successful forgery with a given amount of time spent by the forger and a given number of message-MAC pairs created with the same key.

Q.33 Mention the significance of signature function in Digital Signature Standard (DSS) approach. AU : Dec.-17

Ans. : The DSS uses an algorithm that is designed to provide only the digital signature function. A digital signature is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. An algorithm provides the capability to generate and verify signatures. Signature generation makes use of a private key to generate a digital signature. Signature verification makes use of a public key which corresponds to, but is not the same as, the private key. Each user possesses a private and public key pair. Public keys are assumed to be known to the public in general. Private keys are never shared. Anyone can verify the signature of a user by employing that user's public key. Signature generation can be performed only by the possessor of the user's private key.

Q.34 How digital signatures differ from authentication protocols ? AU : May-18

Ans. : Digital signatures provide the ability to verify author, date and time of signature, authenticate message contents and verified by third parties to resolve disputes. Authentication Protocols used to convince parties of each others and identity and to establish session keys.

Q.35 What entities constitute a full-service Kerberos environment ?

Ans. : A full service environment consists of a Kerberos server, a number of clients and a number of application servers.

Q.36 What are the principle differences between Kerberos version 4 and version 5 ? AU : May-11, IT

Ans. :

i) Kerberos V.4 requires DES and V.5 allows many encryption techniques.
ii) V.4 requires use of IP and V.5 allows other network protocols.
iii) Version 5 has a longer ticket lifetime.
iv) Version 5 allows tickets to be renewed.
v) Version 5 can accept any symmetric-key algorithm.
vi) Version 5 uses a different protocol for describing data types.
vii) Version 5 has more overhead than version 4.

Q.37 When are the certificates revoked in X.509 ? AU : May-15

Ans. : 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.

Q.38 What are the requirement for message authentication ? AU : April/May-19

Ans. : Requirement for message authentication is disclosure, Masquerade, Content modification, sequence modification, timing modification, Source repudiation and destination repudiation.

Q.39 Show how SHA is more secure than MD5 ? AU : April/May-19

Ans. : SHA is more secure than MD5 due to a variety of reasons. First, it produces a larger digest, 160-bit compared to 128-bit, so a brute force attack would be much more difficult to carry out. Also, no known collisions have been found for SHA.

Q.40 What is realm in Kerberos ? AU : Nov./Dec.-19

Ans. : A Kerberos realm is the domain over which a Kerberos authentication server has the authority to authenticate a user, host or service. A realm name is often, but not always the upper case version of the name of the DNS domain over which it presides. The Kerberos server shares a secret key with other Kerberos servers. Therefore, A Kerberos realm is a set of these managed "nodes" that share the same Kerberos database.

Q.41 What entities constitute a full service in Kerberos environment ? AU : Nov./Dec.-19

Ans. : A full-service environment consists of a Kerberos server, a number of clients, and a number of application servers.

Q.42 What is key distribution center ? AU : Dec.-15

Ans. : A key distribution center is responsible for distributing keys to pairs of users such as hosts, processes, applications. Each user must share a unique key with the key distribution center for purposes of key distribution.

Q.43 What are the advantages of key distribution ?

Ans. :

- It is easy to add and remove entities from the network.
- Each entity needs to store only one long-term secret key.
- The public file could reside with each entity.
- Prevent an active adversary from impersonation.

UNIT V

5 Cyber Crimes and Cyber Security

Syllabus

Cyber Crime and Information Security – classifications of Cyber Crimes – Tools and Methods – Password Cracking, Keyloggers, Spywares, SQL Injection – Network Access Control – Cloud Security – Web Security – Wireless Security

Contents

5.1 Cyber Crime and Information Security

- Cybercriminals may use computer technology to access personal information, business trade secrets, or use the Internet for exploitive or malicious purposes.
- Criminals can also use computers for communication and document or data storage. Criminals who perform these illegal activities are often referred to as hackers.
- Cybercrime may also be referred to as computer crime. A cybercriminal is an individual who commits cybercrimes, where he/she makes use of the computer either as a tool or as a target or as both.
- The Department of Justice categorizes computer crime in three ways :
  1. The computer as a target : Attacking of other computers. For example, spreading viruses in the computer.
  2. The computer is used like a weapon : Using a computer to commit "traditional crime" that like in the physical world. For example, it is like fraud or illegal gambling.
  3. The computer as an accessory : using a computer as a "fancy filing cabinet" to store illegal or stolen information.
- Cybercrime requires no physical contact with victims. They can be located anywhere in the world. This both reduces the chances of being caught and makes it very difficult for law enforcement to fingerprint a cybercriminal.
- It also greatly increases the potential number of victims of an attack and the return on investment.

Reasons for success of cyber criminals

- Today's cyber security paradigm is a reactive cycle : when a threat is exposed, it is analyzed and a counter-solution is designed with response times varying from weeks to years.
- The trouble is that attackers can easily reuse pieces of previous malware, modify them, and create a brand new threat, bypassing the newly updated security measures.
- Attackers can simply copy pieces of code from previous malware, such as exploits, decryptors or modules (keyloggers, backdoors etc.), and incorporate them into the new malware they are developing.
- Alternatively, attackers can imitate the operational methods performed by other malware, needed for the success of the operation.

- Cybercriminals often work in organized groups. They are as follows :
  1. **Programmers :** Write code or programs used by cybercriminal organization
  2. **Distributors :** Distribute and sell stolen data and goods from associated cybercriminals
  3. **IT experts :** Maintain a cybercriminal organization's IT infrastructure, such as servers, encryption technologies and databases
  4. **Hackers :** Exploit systems, applications and network vulnerabilities
  5. **Fraudsters :** Create and deploy schemes like spam and phishing
  6. **System hosts and providers :** Host sites and servers that possess illegal contents
  7. **Cashiers :** Provide account names to cybercriminals and control drop accounts
- There are many reasons why cyber-criminals are doing cyber-crime. Some of the reasons are given below :
  1. Difficulty in personal identification
  2. For the sake of recognition
  3. For earning quick money.
  4. Low marginal cost of online activity due to global reach.
  5. Start as hobby and then any reason.
  6. Catching by law and enforcement agency is less effective and more expensive.
  7. New opportunity to do legal acts using technical architecture.
  8. Official investigation and criminal prosecution is rare.

5.1.1 Types of Cyber Crimes

- There are many types of cyber crimes and the most common ones are explained below :
  1. **Hacking :** This is a type of crime wherein a person's computer is broken into so that his personal or sensitive information can be accessed.
  2. **Theft :** This crime occurs when a person violates copyrights and downloads music, movies, games and software.
  3. **Cyberstalking :** This is a kind of online harassment wherein the victim is subjected to a barrage of online messages and emails.
  4. **Identity theft :** This has become a major problem with people using the Internet for cash transactions and banking services. In this cyber crime, a criminal accesses data about a person's bank account, credit cards, debit card

and other sensitive information to siphon money or to buy things online in the victim's name.

5. **Malicious software :** These are Internet-based software or programs that are used to disrupt a network. The software is used to gain access to a system to steal sensitive information or data or causing damage to software present in the system.

6. **Child soliciting and abuse :** This is also a type of cyber crime wherein criminals solicit minors via chat rooms for the purpose of child pornography.

Example of cyber crime :

a. Online banking fraud
b. Fake antivirus
c. 'Stranded traveler' scams
d. 'Fake escrow' scams
e. Advanced fee fraud
f. Infringing pharmaceuticals
g. Copyright-infringing software
h. Copyright-infringing music and video
i. Online payment card fraud
j. In-person payment card fraud
k. Industrial cyber-espionage and extortion
l. Welfare fraud.

- The trafficking, distribution, posting, and dissemination of obscene material including pornography, indecent exposure, and child pornography, constitutes one of the most important Cybercrimes known today.
- Stealing the significant information, data, account number, credit card number transmit the data from one place to another. Hacking and cracking are amongst the gravest Cybercrimes known till date.

5.1.2 Information Security Life Cycles

- Fig. 5.1.1 shows information security life cycle.
- Security in development and support processes is an essential part of a comprehensive quality assurance and production control process and usually involves training and continuous oversight by the most experienced staff.
- Rules for system and software development should be developed.

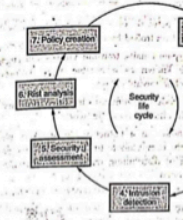


Fig. 5.1.1 Security life cycle

- These rules should incorporate secure software development techniques such as user authentication, session control, logging, and data validation and sanitization.
- Security life cycle involves following phases :
  1. Planning
  2. Policy implementation
  3. Monitoring
  4. Intrusion detection
  5. Security assessment
  6. Risk analysis
  7. Security policy creation.

Security categorization standards help organizations make the appropriate selection of security controls for their information systems.

Security planning ensures that user fully document any agreed upon security controls, whether they are just planned or in place.

The security plan also provides a complete characterization or description of the information system and attachments of or references to key documents that support the information security program of the agency.

5.1.3 Botnets

- A botnet is an interconnected network of computers infected with malware without the user's knowledge and controlled by cybercriminals.
- They're typically used to send spam emails, transmit viruses and engage in other acts of cybercrime. Sometimes known as a zombie army, botnets are often considered one of the biggest online threats today.
- Computers in a botnet, called nodes or zombies, are often ordinary computers sitting on desktops in homes and offices around the world.
- Typically, computers become nodes in a botnet when attackers illicitly install malware that secretly connects the computers to the botnet and they perform tasks such as sending spam, hosting or distributing malware or other illegal files, or attacking other computers.
- Fig. 5.1.2 shows botnet.
- Attackers usually install bots by exploiting vulnerabilities in software or by using social engineering tactics to trick users into installing the malware. Users are often unaware that their computers are being used for malicious purposes.
- The word Botnet is formed from the words 'robot' and 'network'. Cybercriminals use special Trojan viruses to breach the security of several users' computers, take control of each computer, and organize all of the infected machines into a network of 'bots' that the criminal can remotely manage.

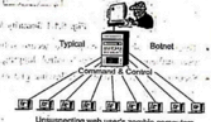


Fig. 5.1.2 Botnet

- A zombie or bot is often created through an Internet port that has been left open and through which a small Trojan horse program can be left for future activation. At a certain time, the zombie army "controller" can unleash the effects of the army by sending a single command, possibly from an Internet Relay Channel (IRC) site.
- Botnets can be used to :
  1. Send out spam emails
  2. Launch a Distributed Denial of Service Attack
  3. Commit advertising fraud
  4. Distribute malware, or spyware

- Keep phishing websites active and frequently change their domains to remain anonymous and undetected by law enforcement.

5.1.4 Zombie

- Zombie computer is a computer connected to the Internet that has been compromised and controlled by an attacker without user's consent.
- Zombie network (Botnet) refers to a network of zombie computers under the remote control by an attacker. Attackers control their botnets through some command and control centers to perform illegal activities.
- If your computer is infected by malicious code such as Trojan Horse, your computer may be controlled by an attacker and may become a zombie.
- Types of attacks perpetrated by a zombie network include denial of service attacks, adware, spyware, spam and click fraud.
- The following steps are used to create zombie networks :
  1. A zombie network operator uses a bot to infect thousands of computers with worms or viruses that carry a deadly payload.
  2. The bot inside an infected computer logs on to an online server - usually IRC but sometimes Web.
  3. The zombie network operator leases zombie network services to a customer.
  4. The customer provides the zombie network operator with spam or any other material, which is run through the zombie network.
- Another botnet called, Gameover Zeus Botnet, allows cyber criminals to retrieve banking passwords from infected machines, or use the botnet to infect more computers.

How and Why Do Cyber Criminals Use Botnets ?

- The value of bots and botnets to criminals comes from aggregating massive numbers of computers they can control simultaneously to perform malicious activities.
- Cyber criminals may use the botnets to send spam, phishing emails, or other scams to trick consumers into giving up their financial information.
- Cyber criminals may also collect information from the bot-infected machines and use it to steal identities, incurring loans, and purchase charges under the user's name.
- Cyber criminals may use botnets to create Denial - of - Service (DoS) attacks that flood a legitimate service or network with a crushing volume of traffic. The volume may severely slow down, or even shut down, the organization's business operations.

- Revenue from DoS attacks come through extortion and leasing botnets. The criminals will rent botnets to groups interested in inflicting damage to another entity.
- The "renters" will use the botnet for sending spam and phishing emails or attacking legitimate websites and networks.

5.2 Classifications of Cyber Crimes

1. Cyber pornography

- Pornography on the internet may take various forms. It may include hosting of website containing some obscene or prohibited material or use of computer for producing obscene materials. Such material tends to pervert the thinking of adolescents and corrupt their mind set.
- A person who publishes or transmits or causes to be published in the electronic form any material which is lascivious, or if its effects in such as to tend to deprave or corrupt the persons who are likely to see, wad or hear the matter contained or embodied in it, is liable to punishment.
- The important ingredients of such an offence are publication and transmission through any electronic medium, of pornographic material in any electronic form.
- Cyber pornography is in simple words defined as the act of using cyberspace to create, display, distribute, import, or publish pornography or obscene materials. With the advent of cyberspace, traditional pornographic content has now been largely replaced by online/digital pornographic content.
- Pornography has no legal or consistent definition. The definition of pornography depends how the society, norms and their values are reacting to the pornographic content.

2. Email spoofing

- A hacker logging in to a computer of under was to his victim often will login under a different identity. This is called spoofing. The hacker able to the by, having previously actual password or having created a new identity by fooling the computer into thinking he is the system's operator.
- A spoofed a may email may be said to be one which the be miss represent its origin. That is, it shows its online to be different from which it actually originates.
- For example, where A sends a threatening a email to the president of the students a union threatening to detente a nuclear sent from the college compos and this email was sent from the account of some other student "A" would a be quality of email spoofing.

3. Identity theft

- Identity theft and fraud is one of the most common types of cybercrime. The term Identity Theft is used, when a person purports to be some other person, with a view to creating a fraud for financial gains.
- When this is done online on the Internet, its is called online identity theft.
- The most common source to steal identity information of others, are data breaches affecting government or federal websites.
- It can be data breaches of private websites too, that contain important information such as, credit card information, address, email ID's, etc.

4. Data diddling

- This offence involves changing or reusing of data in sub till of ways which makes of it different to put the data subtitle ways which data back of or be curtain of its accuracy.
- This is resorted to for the purpose of illegal monetary gains or for community of fraud of financial scam. In case of scan the criminal are change of data which is related on the scan.
- In this data are changed of computer system, record are destroy of and alterations of information of and other type of frauds.

5. Email bombing

- This is another form of internet misuse where individuals directs amass numbers of mail to the victim or an address in attempt to overflow the mailbox, which may be an individual or a company or even mail servers there by ultimately resulting into crashing. There are two methods of perpetrating an email bomb which include mass mailing and list linking.

6. Internet time thefts

- This form is kinds of embezzlements where the fraudulent uses the Internet surfing hours of the victim as their own which can be complete by obtaining access to the login ID and the password, an example is Colonel Bajwa's case in this incident the Internet hours were used up by a unauthorized person.

7. Salami attacks

- This kind of crime is normally consisting of a number of smaller data security attacks together end resulting in one major attack.
- This method normally takes place in the financial institutions or for the purpose of committing financial crimes. An important feature of this type of offence is that the alteration is so small that it would normally go unnoticed.

- This form of cybercrime is very common in banks where employees can steal small amount and it's very difficult to detect or trace.

8. Web jacking

- This is where the hacker obtains access and can control web site of another person, where he or she can destroy or alter the information on the site as they see fit to them. This type of method of cybercrime is done for satisfying political agendas or for purely monetary means.

9. Hacking

- In other words can be referred to as the unauthorized access to any computer systems or network. This method can occur if computer hardware and software has any weaknesses which can be infiltrated if such hardware or software has a lack in patching, security control, configuration or poor password choice.

10. Software piracy

Software piracy is the illegal copying, distribution, or use of software. It is such a profitable "business" that it has caught the attention of organized crime groups in a number of countries.

- Piracy includes casual copying of particular software by an individual or business.
- Using pirated software is also risky for users. Aside from the legal consequences of using pirated software, users of pirated software forfeit some practical benefits as well. Those who use pirate software :
  a) Increase the chances that the software will not function correctly or will fail completely;
  b) Forfeit access to customer support, upgrades, technical documentation, training, and bug fixes;
  c) Have no warranty to protect themselves;
  d) Increase their risk of exposure to a debilitating virus that can destroy valuable data;
  e) May find that the software is actually an outdated version, a beta (test) version, or a nonfunctioning copy;
  f) Are subject to significant fines for copyright infringement; and
  g) Risk potential negative publicity and public and private embarrassment.
- The software licensure agreement is a contract between the software user and the software developer. Usually, this agreement has certain terms and conditions the software user must follow.

- When the user doesn't follow the rules and regulations, they are guilty of software piracy. Some of these terms and conditions prohibit :
  1. Using multiple copies of a single software package on several computers
  2. Passing out copies of software to others without the proper documentation
  3. Downloading or uploading pieces of software via bulletin boards for others to copy
  4. Downloading and installing shareware without paying for it.
- Examples of documents that support the information security program include a configuration management plan, a contingency plan, an incident response plan, a security awareness and training plan, rules of behavior, a risk assessment, a security test and evaluation results, system interconnection agreements, security authorizations and accreditations, and a plan of action and milestones.
- This step provides the necessary security authorization of an information system to process, store, or transmit information that is required.
- This authorization is granted by a senior organization official and is based on the verified effectiveness of security controls to some agreed upon level of assurance and an identified residual risk to agency assets or operations.
- Monitoring ensures that controls continue to be effective in their application through periodic testing and evaluation.
- Security control monitoring, such as verifying the continued effectiveness of those controls over time, and reporting the security status of the information system to appropriate agency officials are essential activities of a comprehensive information security program.
- Assessment may be internal or external. The internal assessment is a controlled network attack simulation that is used to gauge the exposure present on internal systems, applications, and network devices.
- The assessment provides a more structured approach to identifying vulnerabilities that may go undetected.
- The goal of an external assessment is to quantify the security risk that is associated with Internet - connected systems.
- Preliminary risk assessment : This step results in an initial description of the security needs of the system. A preliminary risk assessment should define the threat environment in which the system will operate.

5.3 Tools and Methods

- Tools and Methods used in Cybercrime are Proxy Servers, Anonymizers, Password Cracking, Key loggers and Spyware.

1. Proxy Server

- Proxy Server works with the web/content filtering function to centralize the access to the internet and foreign networks. All internet traffic will be processed within the Proxy Server in order to be able to control and log access.
- A proxy server is software that acts on behalf of an application that is trying to communicate from one network to another. Proxy server software can run on a machine by itself or along with other software such as packet filtering.
- Proxy Servers is a server that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity.
- Proxies were invented to add structure and encapsulation to distributed systems. Today, most proxies are web proxies, facilitating access to content on the World Wide Web, providing anonymity and may be used to bypass IP address blocking.

2. Anonymizers

- An Anonymizers or an anonymous proxy is a tool that attempts to make activity on the Internet untraceable. Anonymization is a data processing technique that removes or modifies personally identifiable information; it results in anonymized data that cannot be associated with any one individual. It's also a critical component of Google's commitment to privacy.
- It is a proxy server computer that acts as an intermediary and privacy shield between a client computer and the rest of the Internet. It accesses the Internet on the user's behalf, protecting personal information by hiding the client computer's identifying information.

5.3.1 Password Cracking

- When your log in to a computer and enter password, the computer checks that password belongs to you and then grants access.
- The password is the secret that is known only to the user and server. But it would be quite dangerous to store the passwords in the file in the computer.
- If an internal attacker obtains access to that file, all passwords stored on that computer could get compromised.
- Password cracking is one of the oldest hacking arts. Every system must store passwords somewhere in order to authenticate users.

- However, in order to protect these passwords from being stolen, they are encrypted. Password cracking is the art of decrypting the passwords in order to recover them.
- A password cracking program if used ethically can be used by the system administrator to detect weak passwords amongst the system so they can be changed. A password cracking program is most likely used to check the security of your own system.
- Crack is a type of password cracking utility that runs through combinations of passwords until it finds one that it matches. It also scans the content of a password file looking for weak login passwords.
- Passwords are not stored in clear text format. As a rule, passwords are stored as hashes. Hashes are one-way encryption that is unique for a given input. In the Windows operating system, passwords on the local system are stored in the SAM file, while Linux stores them in the /etc/shadow file.
- Reasons behind password cracking :
  1. To gain unauthorized access to a computer/server.
  2. Some time we forget the password so to recover a password
  3. To check the security of your system
  4. To do the crime with other name.
- Manual password cracking is easy. Attacker uses following method for password cracking.
  1. Select administrator account or guest account
  2. Make a list of possible password. Here date of birth, pet name, company name, any particular event happens to that person are consider.
  3. Prepared the password list with higher priority to lower priority
  4. Try one by one password until you found the proper password.
  5. Password is stored in database with encrypted format. Manual cracking of password is time consuming process. Encrypted password is used to ensure confidentiality.

5.4 Keyloggers

- A keylogger is a type of surveillance software that has the capability to record every keystroke you make to a log file.
- A keylogger recorder can record instant messages, e-mail, and any information you type at any time using your keyboard. The log file created by the keylogger can then be sent to a specified receiver.

- A keylogger is a program that runs in the background or hardware, recording all the keystrokes. Once keystrokes are logged, they are hidden in the machine for later retrieval, or shipped raw to the attacker.
- Security using Keyloggers will monitor email, internet, chats or anything that requires a keystroke. This will help capture all information in image and/or text form.
- Keyloggers are a type of malicious malware that track the users' keystrokes and captures the characters that are pressed in and writes the information to a file.
- There are two types of keylogger : Hardware keylogger and software keylogger

5.4.1 Hardware Keyloggers

- Hardware Keyloggers are small electronic devices used for capturing the data in between a keyboard device and I/O port. These devices have built in memory where they store the keystrokes. They must be retrieved by the person who installed it in order to obtain the information.
- Hardwar keyloggers is not detected by anti-viral software or scanners.
- Hardwar keyloggers are of three types :
  1. Inline devices that are attached to the keyboard cable.
  2. Devices which can be installed inside standard keyboards.
  3. Replacement keyboards that contain the key logger already built-in.

Advantages :

1. Antivirus techniques cannot catch these.
2. Work on all computing platforms.

Disadvantages :

1. It can be spotted by a suspicious user.

5.4.2 Software Keyloggers

- Software Keyloggers track systems, collect keystroke data within the target operating system, store them on disk or in remote locations, and send them to the attacker who installed the Keyloggers.
- They are implemented as traditional applications or kernel-based. In almost all malicious instances of this type of keylogger, users participated in some way in the software's installation.

- Anti-malware, personal firewall, and host-based intrusion prevention (HIPS) solutions detect and remove application keyloggers.
- Software keylogger detection methods include :
  1. Scan local drives for log.txt or other log file names associated with known keyloggers.
  2. Implement solutions that detect unauthorized file transfers via FTP or other protocols.
  3. Scan content sent via email or other authorized means looking for sensitive information.
  4. Detect encrypted files transmitted to questionable destinations.
- Software keyloggers can be detected using software tools. For this reason, users of keyloggers often prefer hardware solutions.

Advantages :

1. Are hard to detect
2. Can be deployed remotely via a software-vulnerability attack.
3. Are fairly easy to write

Disadvantages :

1. A good Antivirus scheme could sniff these out.
2. Far fewer cons with the software, so these are much more common than hardware-type keyloggers.

5.5 Spyware

- It is a type of malware that is secretly or surreptitiously installed into an information system to gather information on individuals or organizations without their knowledge; a type of malicious code.
- Spyware is the term given to a category of software which aims to steal personal or organizational information. It is done by performing a set of operations without appropriate user permissions, sometimes even covertly.
- General actions a spyware performs include advertising, collection of personal information and changing user configuration settings of the computer.
- A Spyware is generally classified into adware, tracking cookies, system monitors and Trojans. The most common way for a spyware to get into the computer is through freeware and shareware as a bundled hidden component.
- Once a spyware gets successfully installed, it starts sending the data from that computer in the background to some other place. These days' spywares are

usually used to give popup advertisements based on user habits and search history. But when a spyware is used maliciously, it is hidden in the system files of the computer and difficult to differentiate.

5.6 SQL Injection

AU : Dec.-20

- A SQL injection attack involves placing SQL statements in the user input.
- **SQL injection** is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.
- SQL Injection is subset of the an unverified/unsanitized user input vulnerability and the idea is to convince the application to run SQL code that was not intended.It is an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another.
- SQL injection attacks are also known as SQL insertion attacks
- **Forms of the vulnerability**
  1. Incorrectly filtered escape characters
  2. Incorrect type handling
  3. Blind SQL injection
  4. Parameterized statements
  5. Escaping
- SQL injection attack occurs when :
  1. An unintended data enters a program from an untrusted source.
  2. The data is used to dynamically construct a SQL query
- The main consequences are:
  1. **Confidentiality :** Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL Injection vulnerabilities.
  2. **Authentication :** If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.
  3. **Authorization :** If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL Injection vulnerability.

4. **Integrity :** Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL Injection attack.

SQL Injection Remedies

- There are two complementary and successful methods of mitigating SQL Injection attacks :
  1. Parameterized queries using bound, typed parameters
  2. Careful use of parameterized stored procedures.

Review Question

1. *Assume when an attacker tries to modify the database content by inserting an UPDATE statement. Identify this SQL injection attack method and justify. Detail the methods used to prevent SQL injection attack.* AU : Dec.-20, Marks 13

5.7 Network Access Control

- Network Access Control (NAC), also known as network admission control, is the process of restricting unauthorized users and devices from gaining access to a corporate or private network.
- NAC ensures that only users who are authenticated and devices that are authorized and compliant with security policies can enter the network.
- As endpoints proliferate across an organization typically driven by bring-your-own-device (BYOD) policies and an expansion in the use of Internet-of-Things (IoT) devices more control is needed. Even the largest IT organizations do not have the resources to manually configure all the devices in use. The automated features of a NAC solution are a sizable benefit, reducing the time and associated costs with authenticating and authorizing users and determining that their devices are compliant.
- NAC solutions help organizations control access to their networks through the following capabilities :
  1. **Policy lifecycle management :** Enforces policies for all operating scenarios without requiring separate products or additional modules.
  2. **Profiling and visibility :** Recognizes and profiles users and their devices before malicious code can cause damage.
  3. **Guest networking access :** Manage guests through a customizable, self-service portal that includes guest registration, guest authentication, guest sponsoring, and a guest management portal.

4. **Security posture check :** Evaluates security-policy compliance by user type, device type, and operating system.
5. **Incidence response :** Mitigates network threats by enforcing security policies that block, isolate, and repair noncompliant machines without administrator attention.
6. **Bidirectional integration :** Integrate with other security and network solutions through the open/RESTful API.

- Network access control comes with a number of benefits for organizations :
  1. Control the users entering the corporate network.
  2. Control access to the applications and resources users aim to access.
  3. Allow contractors, partners, and guests to enter the network as needed but restrict their access.
  4. Segment employees into groups based on their job function and build role-based access policies.
  5. Protect against cyberattacks by putting in place systems and controls that detect unusual or suspicious activity.
  6. Automate incident response.
  7. Generate reports and insights on attempted access across the organization.

5.8 Cloud Security

- The NIST define cloud computing as: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."
- Cloud provider is responsible for the physical infrastructure and the cloud consumer is responsible for application configuration, personalization and data.
- Broad network access refers to resources hosted in a cloud network that are available for access from a wide range of devices. Rapid elasticity is used to describe the capability to provide scalable cloud computing services.
- Cloud security is the protection of data stored online via cloud computing platforms from theft, leakage, and deletion. Methods of providing cloud security include firewalls, penetration testing, tokenization, virtual private networks (VPN), and avoiding public internet connections.

- Cloud security refers to an array of policies, technological procedures, services, and solutions designed to support safe functionality when building, deploying, and managing cloud-based applications and associated data.
- Cloud security is designed to protect the following, regardless of your responsibilities :
  a) Physical networks - Routers, electrical power, cabling, climate controls, etc.
  b) Data storage - Hard drives, etc.
  c) Data servers - Core network computing hardware and software
  d) Computer virtualization frameworks - Virtual machine software, host machines, and guest machines
  e) Operating systems (OS) - Software that houses
  f) Middleware - Application programming interface (API) management,
  g) Runtime environments - Execution and upkeep of a running program
  h) Data - All the information stored, modified, and accessed
  i) Applications - Traditional software services (email, tax software, productivity suites, etc.)
  j) End-user hardware - Computers, mobile devices, Internet of Things (IoT) devices, etc.
- Cloud computing security addresses both physical and logical security issues across all the different service models of software, platform and infrastructure. It also addresses how these services are delivered in the Public, Private, Hybrid and Community delivery models.

5.8.1 Cloud Security Challenges and Risks

- Cloud computing security challenges fall into three broad categories :
  1. **Data protection :** Securing your data both at rest and in transit.
  2. **User authentication :** Limiting access to data and monitoring who accesses the data.
  3. **Disaster and data breach :** Contingency Planning.
- **Data protection :** Data needs to be encrypted at all times, with clearly defined roles when it comes to who will be managing the encryption keys.
- **User authentication :** Data resting in the cloud needs to be accessible only by those authorized to do so, making it critical to both restrict and monitor who will be accessing the company's data through the cloud. In order to ensure the integrity of user authentication, companies need to be able to view data access logs and audit trails to verify that only authorized users are accessing the data.

- **Contingency planning :** With the cloud serving as a single centralized repository for a company's mission-critical data, the risks of having that data compromised due to a data breach or temporarily made unavailable due to a natural disaster are real concerns.
- If information is encrypted while passing through the cloud, who controls the encryption/decryption keys? Is it the customer or the cloud vendor? Most customers probably want their data encrypted both ways across the Internet using Secure Sockets Layer protocol.
- They also most likely want their data encrypted while it is at rest in the cloud vendor's storage pool. Be sure that you, the customer, control the encryption/decryption keys, just as if the data were still resident on your own servers.
- Data integrity means ensuring that data is identically maintained during any operation.
- Cloud-based services will result in many mobile IT users accessing business data and services without traversing the corporate network. This will increase the need for enterprises to place security controls between mobile users and cloud-based services.
- Placing large amounts of sensitive data in a globally accessible cloud leaves organizations open to large distributed threats, attackers no longer have to come onto the premises to steal data, and they can find it all in the one "virtual" location.
- Virtualization efficiencies in the cloud require virtual machines from multiple organizations to be co-located on the same physical resources. Although traditional data center security still applies in the cloud environment, physical segregation and hardware-based security cannot protect against attacks between virtual machines on the same server.
- Operating system and application files are on a shared physical infrastructure in a virtualized cloud environment and require system, file, and activity monitoring to provide confidence and auditable proof to enterprise customers that their resources have not been compromised or tampered with.
- In the cloud computing environment, the enterprise subscribes to cloud computing resources, and the responsibility for patching is the subscriber's rather than the cloud computing vendor's.
- The need for patch maintenance vigilance is imperative. Lack of due diligence in this regard could rapidly make the task unmanageable or impossible, leaving you with "virtual patching" as the only alternative.

- **Confidentiality :** Confidentiality refers to limiting information access. Sensitive information should be kept secret from individuals who are not authorized to see the information.
- In cloud environments, confidentiality primarily pertains to restricting access to data in transit and storage.
- Integrity can extend to how data is stored, processed, and retrieved by cloud services and cloud-based IT resources.
- Some common cloud security threats include :
  a) Risks of cloud-based infrastructure including incompatible legacy IT frameworks, and third-party data storage service disruptions.
  b) Internal threats due to human error such as misconfiguration of user access controls.
- External threats caused almost exclusively by malicious actors, such as malware, phishing, and DDoS attacks

5.8.2 General Issues Securing the Cloud

- The common security issues around cloud computing divided into four main categories :
  a) Cloud infrastructure, platform and hosted code : This comprises concerns related to possible virtualization, storage and networking vulnerabilities.
  b) Data : This category comprises the concerns around data integrity, data lock in, data remanence, provenance, and data confidentiality and user privacy specific concerns.
  c) Access : This comprises the concern around cloud access (authentication, authorization and access control or AAA), encrypted data communication, and user identity management.
  d) Compliance : Because of its size and disruptive influence, the cloud is attracting attention from regulatory agencies, especially around security audit, data location; operation trace- ability and compliance concerns.

5.9 Web Security

- Web security means protecting a website or web application by detecting, preventing and responding to cyber threats.
- Web security is a set of procedures, practices, and technologies for assuring the reliable, predictable operation of web servers, web browsers, other programs that communicate with web servers, and the surrounding Internet infrastructure.

- Web server is a program that stores files and makes them accessible via the network or the internet. A web server requires both hardware and software
- Primary facets are as follows :
  a. Securing the web server and the data that is on it
  b. Securing information that transmit between the web server and the user
  c. Securing the end user's computer and other devices that people use to access the Internet
- Policy to protect itself against web server attacks :
  1. Patch management: this involves installing patches to help secure the server. A patch is an update that fixes a bug in the software. The patches can be applied to the operating system and web server system.
  2. Vulnerability scanning system: these include tools such as Snort, NMap, Scanner Access Now Easy.
  3. Firewalls can be used to stop simple DoS attacks by blocking all traffic coming the identify source IP addresses of the attacker.
  4. Antivirus software can be used to remove malicious software on the server
  5. Disabling Remote Administration
  6. Default accounts and unused accounts must be removed from the system
  7. Default ports and settings (like FTP at port 21) should be changed to custom port and settings
  8. Secure installation and configuration of the operating system
  9. Secure installation and configuration of the web server software
- Web server is secure in following ways :
  1. The computer itself must be secured using traditional computer security techniques.
  2. Special programs that provide web service must be secured.
  3. User need to examine the operating system and the web service to see if there are any unexpected interactions between the two that might compromise the system's overall security

5.9.1 Web Security Issue

- The Web is very visible. The WWW is widely used by businesses, government agencies, and many individuals. But the Internet and the Web are extremely vulnerable to compromises of various sorts, with a range of threats.
- Complex software hides many security flaws. Web servers are easy to configure and manage. Users are not aware of the risks.

- These can be described as passive attacks including eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.
- Active attacks including impersonating another user, altering messages in transit between client and server, and altering information on a Web site. The Web needs added security mechanisms to address these threats.

Web Traffic Security Approaches

- Various approaches are used for providing security to the Web. One of the examples is IP security.
- Following table shows the comparison of threats on the web.

Parameters	Threats	Consequences	Countermeasures
Integrity	1. Modification of user data 2. Trojan horse browser 3. Modification of memory 4. Modification of message traffic in transit	1. Loss of information 2. Compromise of machine 3. Vulnerability to all other threats	Cryptographic checksums
Confidentiality	1. Eavesdropping on the Net 2. Theft of information from server 3. Theft of data from client 4. Information about network configuration 5. Information about which client talks to server	1. Loss of information 2. Loss of privacy	Encryption, Web proxies
Denial of Service	1. Killing of user threads 2. Flooding machine with bogus requests 3. Filling up disk or memory 4. Isolating machine by DNS attacks	1. Disruptive 2. Annoying 3. Prevent user from getting work done	Difficult to prevent

Authentication	1. Impersonation of legitimate users	1. Misrepresentation of user	Cryptographic techniques
	2. Data forgery	2. Belief that false information is valid	

- Fig 5.9.1 shows the relative location of security facilities in the TCP/IP protocol stack.

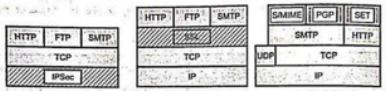


Fig. 5.9.1 Relative locations of security facilities in TCP/IP

5.9.2 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. 5.9.2 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.

Fig. 5.9.2 TLS

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. 5.9.3 shows the TLS handshake protocol.

Fig. 5.9.3 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.

5.10 Wireless Security

Need of security in Wireless System

- In wireless LAN, data transmission and receiving medium is an air using radio frequency. It minimizes the wiring connection. Wireless LANs combine data connectivity with user mobility.
- Wireless LANs is used by types of organization and users. So the security issue in wireless networks is much more critical than in wired networks. Packet sent on a wireless system is quite broadcast so it is possible to other user to collect the data.
- Serious countermeasures must be taken for some of the critical applications. Following is the key factors contributing to higher security risk of wireless networks.
  1. **Communication Channel :** Wireless networking typically involves broadcast communications, which is far more susceptible to eavesdropping and jamming than wired networks. Wireless networks are also more vulnerable to active attacks that exploit vulnerabilities in communications protocols.
  2. **Mobility :** Wireless devices are far more portable and mobile, thus resulting in a number of risks.
  3. **Accessibility :** Some wireless devices, such as sensors and robots, may be left unattended in remote and/or hostile locations, thus greatly increasing their vulnerability to physical attacks.

5.10.1 Attacks of Wireless Network

- Because of common Network Layers, most of the attacks in the wired network will also work against wireless client.
- On a wireless network, an adversary is able to eavesdrop on all messages within the emission area, by operating in promiscuous mode and using a packet sniffer.
- Most of the organization store sensitive and confidential information on marketing, credit records, income tax, trade secrets, national security data, and classified military data, among others. The access of such data by unauthorized users may entail loss of money or release of confidential information to competitors or enemies.

- Wireless networks are more vulnerable to security threats, due to the computation and power limitations.
- Attacks are of two types : **Passive attack and Active attack.**

5.10.2 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 :** 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.
  2. **Traffic analysis :** Mask the contents of message so that opponents could not extract the information from the message.
- Passive attacks are very difficult to detect because they do not involve any alteration of data. It is feasible to prevent the success of attack, usually by means of encryption.

5.10.3 Active Attack

- Active attacks involve some modification of the data stream or the creation of a false stream. These attacks cannot be prevented easily. Active attacks can be subdivided into four types :
  1. Masquerade
  2. Replay
  3. Modification of message
  4. Denial of service
- **Masquerade** takes place when one entity pretends to be a different entity. **Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.
- **Modification** of message involves some change to the original message. It produces an unauthorized effect, DOS prevents the normal use or management of communications facilities. Fabrication causes Denial Of Service (DOS) attacks.

5.10.4 Type of Wireless Attack

- The main categories of *attack on wireless computer networks* are as follows :
  1. **Interruption of service :** Resource becomes unavailable because it is destroyed.
  2. **Modification :** Attacker gain access of the resources and modify the database values, alters the program etc.
  3. **Fabrication :** the attacker send fake message to the neighboring nodes without receiving any related message.
  4. **Jamming :** Jamming is a special class of DoS attacks which are initiated by malicious node after determining the frequency of communication. In this type of attack, the jammer transmits signals along with security threats. Jamming attacks also prevents the reception of legitimate packets.
  5. **Attacks against encryption :** Wired equivalent privacy encryption method is used 802.11b wireless LAN but there is some weakness in this algorithm. Sophisticated attacker can break the WEP method.
  6. **Brute force attacks against passwords of access points.** A 'brute force' login attack is a type of attack against a access point to gain access by guessing the username and password, over and over again.
  7. **Mis-configuration :** Because of heavy load on the network admin, most of the access points are not configure properly. These access points remain at high risk of being accessed by unauthorized parties or hackers.
  8. **Interception :** As the communication takes place on wireless medium can easily be intercepted with receiver tuned to the proper frequency. The main aim of such attacks is to obtain the confidential information that should be kept secret during the communication. The information may include private key, public key, location or passwords of the nodes.

5.10.5 Wireless Equivalent Privacy Protocol

- Wired Equivalent Privacy (WEP) is a security protocol, specified in the IEEE Wireless Fidelity (Wi-Fi) standard, 802.11b, that is designed to provide a Wireless Local Area Network (WLAN). WEP is designed to provide the same level of security as that of a wired LAN.
- The WEP algorithm was designed to be used to protect wireless communication from unauthorized eavesdropping and restricting access to a wireless network.
- A wired local area network (LAN) is generally protected by physical security mechanism that are effective for a controlled physical environment, because a wireless network broadcasts messages using radio, it is particularly susceptible to eavesdropping.

- WEP aims to provide security by encrypting data over radio waves so that it is protected as it is transmitted from one end point to another. However, WEP is not as secure as believed. WEP is used at the two lowest layers of the OSI model - the data link and physical layers; it therefore does not offers end-to-end security.
- WEP is part of the IEEE 802.11 standard. It uses the stream cipher RC4 for confidentiality and the CRC-32 checksum for integrity. Fig. 5.10.1 shows basic WEP Encryption where RC4 Keystream XORed with Plaintext.
- Standard 64-bit WEP uses a 40 bit key, which is concatenated to a 24-bit Initialization Vector (IV) to form the RC4 traffic key. But restrictions on cryptographic technology limit the key size. Once the restrictions were lifted, all of the major manufacturers eventually implemented an extended 128-bit WEP protocol using a 104-bit key size.
- Key size is not the only major security limitation in WEP. Cracking a longer key requires interception of more packets, but there are active attacks that stimulate the necessary traffic. There are other weakness in WEP, including the possibility of IV collisions and altered packets, that are not helped at all by a longer key.

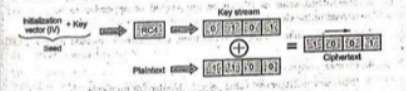


Fig. 5.10.1

- Because RC4 is a stream cipher, the same traffic key must never be used twice. The purpose of an IV, which is transmitted as plain text, is to prevent any repetition, but a 24-bit IV is not long enough to ensure this on a busy network. The way the IV was used also opened WEP to a related key attack. For a 24-bit IV, there is a 50% probability the same IV will repeat after 5000 packet.
- WEP security involves two parts : **Authentication and Encryption.**
- When device initially join the LAN, then authentication starts. It prevents the device or station to join the network unless they know the WEP key. Fig. 5.10.2 shows WEP authentication.
- Wireless device sends authentication request to the wireless access point, then wireless access point sends 128 bit random challenge in a clear text to the requesting client. The wireless device uses the shared secret key to sign the challenge and sends it to the wireless access point.

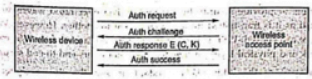


Fig. 5.10.2 WEP authentication

- Wireless access point decrypts the signed message using the shared secret key and verifies the challenge that it has sent before. If the challenge matches, then authentication succeeds otherwise not.
- In WEP, same key is used for authentication and encryption. So it is difficult to tell whether the subsequent message come from the trusted device or from an impostor. There is possibility of man in the middle attack.

Strengthening WEP

- Following are the solution to overcome the weakness of WEP:
  1. Initialization Vector size should be increases.
  2. The hashed value of IV can be pre-pended or appended to the cipher-text instead of the clear-text.
  3. For the data integrity verification, use different method instead of CRC checksum.
  4. Change secret key regularly.
  5. Better key management using handshake protocols.
  6. New authentication mechanisms using the Extensible Authentication Protocol(EAP).

5.11 Two Marks Questions with Answers

Q.1 Define cyber-crime.

Ans. : **Cybercrime** is defined as a crime in which a computer is the object of the crime (hacking, phishing, spamming) or is used as a tool to commit an offense (child pornography, hate crimes). Internet connected activities are as vulnerable to crime. Computer crime is any illegal activity that is perpetrated through the use of a computer.

Q.2 Which are the elements of cyber crime ?

Ans. :

- Location/Place : Where offender is in relation to crime.
- Victim : Target of offense - Government, corporation, organization, individual

- Offender : Who the offender is in terms of demographics, motivation, level of sophistication ?
- Action : What is necessary to eliminate threat ?

Q.3 What is cyber security ?

Ans. :

- Cyber security is a potential activity by which information and other communication systems are protected from and/or defended against the unauthorized use or modification or exploitation or even theft.
- Cyber security is a well-designed technique to protect computers, networks, different programs, personal data, etc., from unauthorized access

Q.4 What are the classifications of cybercrimes ?

Ans. : Classifications of Cybercrimes are email spoofing, Cyber-stalking, Unauthorized access or control over the computer system and Indecent exposure

Q.5 What is password sniffing ?

Ans. : A password sniffer is a software application that scans and records passwords that are used or broadcasted on a computer or network interface. It listens to all incoming and outgoing network traffic and records any instance of a data packet that contains a password

Q.6 What is virtual crime ?

Ans. : Virtual crime refers to a virtual criminal act that takes place in a massively multiplayer online game (MMOG). The huge time and effort invested into such games can lead online "crime" to spill over into real world crime, and even blur the distinctions between the two.

Q.7 Explain spyware.

Ans. :

- It is a type of malware that is secretly or surreptitiously installed into an information system to gather information on individuals or organizations without their knowledge; a type of malicious code.
- Spyware is the term given to a category of software which aims to steal personal or organizational information. It is done by performing a set of operations without appropriate user permissions, sometimes even covertly

Q.8 What is SQL injection ?

Ans. : A SQL injection attack involves placing SQL statements in the user input.SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability is present when user input is

SOLVED MODEL QUESTION PAPER

[As Per New Syllabus]

Cryptography and Cyber Security

Semester - V (CSE)

Time : Three Hours] [Maximum Marks : 100

Answer ALL Questions

PART A - (10 × 2 = 20 Marks)

Q.1 What is product cipher ? **(Refer Two Marks Q.26 of Chapter - 1)**

Q.2 What is a passive attack ?**(Refer Two Marks Q.9 of Chapter - 1)**

Q.3 Write down the purpose of the S-boxes in DES ? **(Refer Two Marks Q.14 of Chapter - 2)**

Q.4 What is an avalanche effect ? **(Refer Two Marks Q.6 of Chapter - 2)**

Q.5 What is discrete logarithm ? **(Refer Two Marks Q.11 of Chapter - 3)**

Q.6 What is key distribution center ?**(Refer Two Marks Q.17 of Chapter - 3)**

Q.7 What is realm in Kerberos ? **(Refer Two Marks Q.40 of Chapter - 4)**

Q.8 What is a Hash in cryptography ? **(Refer Two Marks Q.5 of Chapter - 4)**

Q.9 Define anonymization.**(Refer Two Marks Q.18 of Chapter - 5)**

Q.10 Define cloud computing.**(Refer Two Marks Q.10 of Chapter - 5)**

PART B - (5 × 13 = 65 Marks)

Q.11 a) i) What is transposition techniques ? Explain difference between substitution techniques and transposition techniques. **(Refer section 1.9)** [6]

ii) Describe in detail security mechanism. **(Refer section 1.5)** [7]

OR

b) i) What are elements of information security ? explain **(Refer section 1.1.4)** [6]

ii) Define passive attack. Explain types of passive attacks. **(Refer section 1.3.1)** [7]

Q.12 a) i) To find 11^{23} mod 53. **(Refer example 2.2.1)** [6]

ii) What is block cipher ? Explain advantages and disadvantages of block cipher. **(Refer section 2.7)** [7]

OR

b) i) What is finite field ? List the properties of finite field. Explain construction of finite field. **(Refer section 2.4)** [6]

OR

b) i) Perform encryption and decryption using RSA algorithm for $p = 17$, $q = 11$, $e = 7$ and $M = 88$. **(Refer example 3.7.6)** [7]

ii) For each of the following elements of DES, indicate the comparable element in AES or explain why it is not needed in AES.

a. XOR of subkey material with the input to the function f

b. XOR of the f function output with left side of the block

c. The f function d. Permutation P

e. Swapping of halves of the block. **(Refer example 2.11.1)** [8]

□□□