### ANNAI TERESA COLLEGE OF ENGINEERING

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING <u>ACADEMIC YEAR-2023-24</u>

Sub Code	CB3491
Subject	Cryptography and Cyber Security
Department	CSE
Year	III
Sem www.Engo	<b>V</b> Tree.com

Submitted By: A.Ramya, HOD&AP/CSE

STAFF SIGNATURE HOD

# ANNAI TERESA COLLEGE OF ENGINEERING – THIRUNAVALUR DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING ACADEMIC YEAR-2023-2024(Odd semester)

Subject: CRYPTOGRAPHY AND CYBER SECURITY Sub Code: CB3491

Class: IV-CSE(V-SEM) Prepared By: A.Ramya, HOD&AP/CSE

### **INDEX**

S.NO	CONTENT	PAGE NO
1	SYLLABUS	
2	<b>UNIT</b> ⊻ <b>I</b> v.EnggTree.com	
3	UNIT-II	
4	UNIT-III	
5	UNIT-IV	
6	UNIT-V	

## ANNAI TERESA COLLEGE OF ENGINEERING – THIRUNAVALUR DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Subject: CRYPTOGRAPHY AND CYBER SECURITY Sub Code: CB3491

Class:CSE-III year(V-sem)

#### **SYLLABUS**

#### UNIT I INTRODUCTION TO SECURITY

9

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.

#### UNIT II SYMMETRIC CIPHERS

9

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.

#### UNIT III ASYMMETRIC CRYPTOGRAPHY

9

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.

#### UNIT IV INTEGRITY AND AUTHENTICATION ALGORITHMS 9

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.

#### UNIT V CYBER CRIMES AND CYBER SECURITY

9

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

**TOTAL:45 PERIODS** 

#### TEXT BOOKS

- 1. William Stallings, "Cryptography and Network Security Principles and Practice", Seventh Edition, Pearson Education, 2017.
- 2. Nina Godbole, Sunit Belapure, "Cyber Security: Understanding Cyber crimes, Computer Forensics and Legal Perspectives", First Edition, Wiley India, 2011.

#### REFERENCES

- 1. Behrouz A. Ferouzan, Debdeep Mukhopadhyay, "Cryptography and Network Security", 3rd Edition, Tata Mc Graw Hill, 2015.
- 2. Charles Pfleeger, Shari Pfleeger, Jonathan Margulies, "Security in Computing", Fifth Edition, Prentice Hall, New Delhi, 2015.

www.EnggTree.com

#### <u>UNIT-1 PART-A</u>

#### 1. What is meant by cryptography and cryptanalysis?

(N/D-09)

Cryptography is an art of writing hidden messages. It is a historical (or) forensic approach. Cryptanalysis is the process of analyzing hidden messages. It is a statistical (or) analytical approach.

#### 2. What are the key principles of security?

(A/M-12)

The key principle of security is the following:

- 1. Make sure you have the latest security updates & patches
- 2. Install anti-virus software
- 3. Install anti-spyware software
- 4. Use a personal firewall
- 5. Password advice

### 3. Differentiate conventional (symmetric) from public key (asymmetric) encryption. (M/J-07)

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

### 4. Distinguish between passive attack and active attack with reference to X.800.

(A/M-11)

X.800 categorize the attacks into two, namely passive and active attacks.

**Passive attacks:** Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. It includes release of message contents and Traffic analysis. Passive attacks are very difficult to detect because they do not involve any alteration of data. However, it is feasible to prevent the success of these attacks, usually by means of encryption.

Active attacks: Active attack involves some modification of the data stream or the creation of a false data stream and can be subdivided into <u>fo</u>ur categories namely as a masquerade, replay,

modification of messages and the denial of service attack.

#### 5. Define -Key and Plaintext

(M/J-09)

In cryptography, a key is defined as a piece of information that determines the functional output of a cryptographic algorithm or cipher. In encryption, a key specifies the particular transformation of plaintext into ciphertext or vice versa during decryption. Plaintext is ordinary readable text before being encrypted into ciphertext or after being decrypted.

#### 6. Find the GCD of 2740 and 1760, using Euclidean algorithm. (N/D-08)

The GCD of two numbers say a and b can be found using the following formula  $gcd(a,b) = gcd(b,a \mod b)$ GCD(2740,1760) =  $gcd(1760, 2740 \mod 1760) = gcd(1760,980) = 980$ 

#### 7. What is encipherment?

(A/M-12)

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.

#### 8. What is a passive attack?

(M/J-09)

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions.

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

www.EnggTree.com

#### 9. What is the difference between a monoal phabet cipher and a polyal phabetic cipher? (A/M-12)

Monoalphabetic cipher is a monoalphabetic cipher is a substitution cipher in which the cipher alphabet is fixed through the encryption process. All of the substitution ciphers we have seen prior to this handout are monoalphabetic; these ciphers are highly susceptible to frequency analysis. Polyalphabetic Cipher is a polyalphabetic cipher is a substitution cipher in which the cipher alphabet changes during the encryption process.

#### 10. What is the avalanche effect?

(M/J-07)

In cryptography, the avalanche effect refers to a desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions. The avalanche effect is evident if, when an input is changed slightly (for example, flipping a single

bit) the output changes significantly (e.g., half the output bits flip). In the case of high-quality block ciphers, such a small change in either the key or the plaintext should cause a drastic change in the ciphertext.

#### **UNIT I - INTRODUCTION**

Security trends - Legal, Ethical and Professional Aspects of Security, Need for Security at Multiple Levels, Security Policies - Model of Network Security - Security Attacks, Services and Mechanisms - OSI security architecture - classical encryption techniques: substitution techniques, transposition techniques, steganography - Foundations of modern cryptography: perfect security - information theory - product cryptosystem - cryptanaysis

#### PART-B

#### 1.WRITE SHORT NOTES ON SECURITY TRENDS.

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

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

- Confidentiality: This term covers two related concepts:
- Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals.
- Privacy: Assures that individuals control or influence what information related to them
  may be collected and stored and by whom and to whom that information may be
  disclosed.
- Integrity: This term covers two related concepts:
- Data integrity: Assures that information and programs are changed only in a specified and authorized manner.
- System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.
- Availability: Assures that systems work promptly and service is not denied to authorized users

These three concepts form what is often referred to as the CIA triad (Figure 1.1). The three concepts embody the fundamental security objectives for both data and for information and computing services

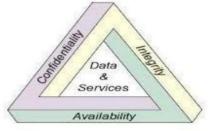


Figure 1.1 CIA triad

Although the use of the CIA triad to define security objectives is well established, some in the security field feel that additional concepts are needed to present a complete picture. Two of the most commonly mentioned are as follows:

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

from a trusted source.

- Accountability: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports non repudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.
- Computer Security Generic name for the collection of tools designed to protect data and to thwart hackers.
- Network Security Measures to protect data during their transmission.
- Internet Security Measures to protect data during their transmission over a collection of interconnected networks Our Focus is on Internet Security which consists of measures to deter, prevent, detect and correct security violations that involve the transmission and storage of information

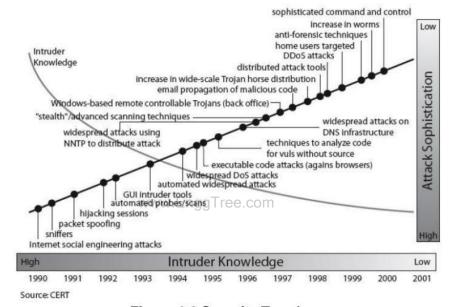


Figure 1.2 Security Trends

#### 1.1.1 THE CHALLENGES OF COMPUTER SECURITY

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

- 1. Security is not as simple as it might first appear to the novice. The requirements seem to be straightforward; indeed, most of the major requirements for security services can be given self-explanatory, one-word labels: confidentiality, authentication, non repudiation, or integrity
- 2. In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features.
- **3.** Typically, a security mechanism is complex, and it is not obvious from the statement of a particular requirement that such elaborate measures are needed.
- **4.** Having designed various security mechanisms, it is necessary to decide where to use them. This is true both in terms of physical placement and in a logical sense
- 5. Security mechanisms typically involve more than a particular algorithm or protocol

- **6.** Computer and network security is essentially a battle of wits between a perpetrator who tries to find holes and the designer or administrator who tries to close them. The great advantage that the attacker has is that he or she need only find a single weakness, while the designer must find and eliminate all weaknesses to achieve perfect security.
- **7.** There is a natural tendency on the part of users and system managers to perceive little benefit from security investment until a security failure occurs.
- **8.** Security requires regular, even constant, monitoring, and this is difficult in today"s short-term, overloaded environment.
- **9.** Security is still too often an afterthought to be incorporated into a system after the design is complete rather than being an integral part of the design process.
- **10.** Many users and even security administrators view strong security as an impediment to efficient and user-friendly operation of an information system or use of information.

#### 2.DECRIBE ABOUT LEGAL, ETHICAL AND PROFESSIONAL ASPECTS OF SECURITY.

Today millions of people perform online transactions every day. There many ways to attack computer and networks to take advantage of what has made shopping, banking, transformation of messages, investments and leisure pursuits a simple matter of dragging and clicking for many people. Thus, the laws and ethics are important aspects in data and network security. The legal system has adapted quite well to computer technology by reusing some old forms of legal protection (copyrights and patents) and creating laws where no adequate one existed (malicious access). Still the courts are not a perfect form of protection for computer, for two reasons, first court tends to be reactive instead of proactive. That is, we have to wait for regression to occur and then adjudicative it, rather than try to prevent it in first place. Second fixing a problem through the courts can be time consuming and more expensive.

The latter characteristic prevents all but the wealthy from addressing most wealthy. On other hand, 1ethics has not had to change, because ethic is more situational and personal than the law, for example the privacy of personal information becoming important part of computer network security and although technically this issue is just an aspect of confidentiality, practically it has a long history in both law and ethics.

Law and security are related in several ways. First international, national, state, city laws affect privacy, secrecy. These statutes often apply to the rights of individuals to keep personal matters private. Second law regulates the use of development, and ownership of data and programs. Patents, copy rights, and trade secrets are legal devices to protect the right of developers and owners of the information and data.

#### Cryptography and Law

**Cyber-Crime:** - Criminal activities or attacks in which computer and computer networks are tool, target, or place of criminal activity. Cybercrime categorize based on computer roles such as target, storage device and communication tool.

**Computers as targets:** To get the information from the computer system or control the computer system without the authorization or payment or alter the interfaces or data in the particular system with use of server.

Computers as storage devices: Computers can be used to further unlawful activity by using a

computer or a computer device as a assive storage medium. For example, the computer can be used to store stolen password lists, credit card details and proprietary corporate information.

Computers as communications tools: Many of the crimes falling within this category are simply traditional crimes that are committed online. Examples include the illegal sale of prescription drugs, controlled substances, alcohol, and guns; fraud; gambling; and child pornography. Other than these crimes there are more specific crimes in computer networks. There are:

**Illegal access:** The access to the whole or any part of a computer system without right. Illegal interception: The interception without right, made by technical means, of non-public transmissions of computer data to, from or within a computer system, including electromagnetic emissions from a computer system carrying such computer data.

**Data interference:** The damaging, deletion, deterioration, alteration or suppression of computer data without right.

**System interference:** The serious hindering without right of the functioning of a computer system by inputting, transmitting, damaging, deleting, deteriorating, altering or suppressing computer data.

**Computer-related forgery:** The input, alteration, deletion, or suppression of computer data, resulting in inauthentic data with the intent that it be considered or acted upon for legal purposes as if it were authentic, regardless whether or not the data is directly readable and intelligible.

**Crime related to child pornography:** Producing child pornography or distribution through a computer system and making available or distributing or transmitting child pornography through a computer system.

www.EnggTree.com

The relative lack of success in bringing cyber-criminals to justice has led to an increase in their numbers, boldness, and the global scale of their operations. It is difficult to profile cybercriminals in the way that is often done with other types of repeat offenders. The success of cybercriminals and the relative lack of success of law enforcement, influence the behaviour of cybercrime victims. As with law enforcement, many organizations that may be the target of attack have not invested sufficiently in technical, physical, and human-factor resources to prevent attacks.

The law is used regulate people for their own good and for the greater good of society. Cryptography also regulated activity.

Some Example laws which are forced on cryptography.

**Control use of cryptography:** Closely related to restrictions on content are restrictions on the use of cryptography imposed on users in certain countries. For examples, 2 In China, state council order 273 requires foreign organizations or individuals to apply permission to use encryption in China. Pakistan requires that all encryption hardware and software be inspected and approved by the Pakistan telecommunication authority.

**Cryptography and Free speech:** The Cryptography involve not just products, it involves ideas too, although governments effectively control the flow of products across borders, controlling the floe ideas either head or on the internet, is also impossible.

**Cryptography and Escrow:** Although laws enable governments to read encrypted communications. In 1996, US government offered to relax the export restriction for so called escrowed encryption, in which the government would able to obtain the encryption key for any encrypted communication.

The victory in use of law enforcement depends much more on technical skills of the people. Management needs to understand the criminal investigation process, the inputs that investigators need, and the ways in which the victim can contribute positively to the investigation.

#### 3.WRITE SHORT NOTES ON INTELLECTUAL PROPERTIES.

There are three main types of intellectual property for which legal protection is available. **Copy rights:** Copyright law protects the tangible or fixed expression of an idea, not the idea itself. Copy right properties exists when proposed work is original and creator has put original idea in concrete form and the copyright owner has these exclusive rights, protected against infringement such as reproduction right, modification right, distribution right

**Patents:** A patent for an invention is the grant of a property right to the inventor. There are 3 types in patents:-

- Utility (any new and useful process, machine, article of manufacture, or composition of matter).
- Design (new, original, and ornamental design for an article of manufacture)
- Plant (discovers and asexually reproduces any distinct and new variety of plant).

**Trade-Marks:** A trademark is a word, name, symbol or expression which used to identify the products or services in trade uniquely from others. Trade mark rights used to prevent others from using a confusingly similar mark, but not to prevent others from making the same goods or from selling the same goods or services under a clearly different mark.

- Intellectual Property Relevant to Network and Computer Security A number of forms of intellectual property are relevant in the context of network and computer security.
- Software programs: software programs are protected by using copyright, perhaps patent.
- Digital content: audio / video / media / web protected by copy right Algorithms: algorithms may be able to protect by patenting
- Privacy Law and Regulation: An issue with considerable overlap with computer security
  is that of privacy. Concerns about the extent to which personal privacy has been and
  may be compromised have led to a variety of legal and technical approaches to
  reinforcing privacy rights. A number of international organizations and national
  governments have introduced laws and regulations intended to protect individual
  privacy.
- European Union Data Protection Directive was adopted in 1998 to ensure member states protect fundamental privacy rights when processing personal info and prevent member states from restricting the free flow of personal info within EU organized around principles of notice, consent, consistency, access, security, onward transfer and enforcement. US Privacy Law have Privacy Act of 1974 which permits individuals to determine records kept, forbid records being used for other purposes, obtain access to records, ensures agencies properly collect, maintain, and use personal info and creates a private right of action for individuals. Cryptography and Ethics.

• There are many potential misuses and abuses of information and electronic communication that create privacy and security problems. Ethics refers to a system of moral principles that relates to the benefits and harms of particular actions. An ethic an objectively defined standard of right and wrong. Ethical standards are often idealistic principles because they focus on one objective. Even though religious group and professional organization promote certain standards of ethical behaviour, ultimately each person is responsible for deciding what do in a specific situation.

#### Ethical issues related to computer and info systems

Computers have become the primary repository of both personal information and negotiable assets, such as bank records, securities records, and other financial information.

Repositories and processors of information: Unauthorized use of otherwise unused computer services or of information stored in computers raises questions of appropriateness or fairness.

**Producers of new forms and types of assets:** For example, computer programs are entirely new types of assets, possibly not subject to the same concepts of ownership as other assets.

**Symbols of intimidation and deception:** The images of computers as thinking machines, absolute truth producers, infallible, subject to blame, and as anthropomorphic replacements of humans who err should be carefully considered.

#### 4.LIST OUT THE NEED FOR SECURITY AT MULTIPLE LEVELS

Multilevel security or multiple levels of security (MLS) is the application of a computer system to process information with incompatible classifications (i.e., at different security levels), permit access by users with different security clearances and needs-to-know, and prevent users from obtaining access to information for which they lack authorization.

There are two contexts for the use of multilevel security.

One is to refer to a system that is adequate to protect itself from subversion and has robust mechanisms to separate information domains, that is, trustworthy.

Another context is to refer to an application of a computer that will require the computer to be strong enough to protect itself from subversion and possess adequate mechanisms to separate information domains, that is, a system we must trust. This distinction is important because systems that need to be trusted are not necessarily trustworthy.

A threat is an object, person, or other entity that represents a constant danger to an asset.

#### **Security Policies**

The Cryptography Policy sets out when and how encryption should be used. It includes protection of sensitive information and communications, key management, and procedures to ensure encrypted information can be recovered by the organisation if necessary.

#### Role of the Security Policy in Setting up Protocols

Following are some pointers which help in setting u protocols for the security policy of an organization.

- Who should have access to the system?
- How it should be configured?
- How to communicate with third parties or systems?

Policies are divided in two categories:

- User policies
- IT policies.

User policies generally define the limit of the users towards the computer resources in a workplace. For example, what are they allowed to install in their computer, if they can use removable storages?

Whereas, IT policies are designed for IT department, to secure the procedures and functions of IT fields.

- General Policies This is the policy which defines the rights of the staff and access level
  to the systems. Generally, it is included even in the communication protocol as a
  preventive measure in case there are any disasters.
- Server Policies This defines who should have access to the specific server and with what rights. Which software's should be installed, level of access to internet, how they should be updated?
- Firewall Access and Configuration Policies It defines who should have access to the
  firewall and what type of access, like monitoring, rules change. Which ports and services
  should be allowed and if it should be inbound or outbound?
- Backup Policies It defines who is the responsible person for backup, what should be
  the backup, where it should be backed up, how long it should be kept and the frequency of
  the backup.
- VPN Policies These policies generally go with the firewall policy; it defines those users
  who should have a VPN access and with what rights. For site-to-site connections with
  partners, it defines the access level of the partner to your network, type of encryption to be
  set.

#### 5.DESCRIBE THE STRUCTURE OF A SECURITY POLICY

When you compile a security policy you should have in mind a basic structure in order to make something practical. Some of the main points which have to be taken into consideration are:

- Description of the Policy and what is the usage for?
- Where this policy should be applied?
- Functions and responsibilities of the employees that are affected by this policy.
- Procedures that are involved in this policy.
- Consequences if the policy is not compatible with company standards.

#### Types of Policies

- Permissive Policy It is a medium restriction policy where we as an administrator block
  just some well-known ports of malware regarding internet access and just some exploits
  are taken in consideration.
- Prudent Policy This is a high restriction policy where everything is blocked regarding the
  internet access, just a small list of websites is allowed, and now extra services are allowed
  in computers to be installed and logs are maintained for every user.

- Acceptance User Policy This policy regulates the behavior of the users towards a system or network or even a webpage, so it is explicitly said what a user can do and cannot in a system. Like are they allowed to share access codes, can they share resources, etc.
- User Account Policy This policy defines what a user should do in order to have or maintain another user in a specific system. For example, accessing an e-commerce webpage. To create this policy, you should answer some questions such as –
  - Should the password be complex or not?
  - o What age should the users have?
  - o Maximum allowed tries or fails to log in?
  - When the user should be deleted, activated, blocked?
- Information Protection Policy This policy is to regulate access to information, hot to process information, how to store and how it should be transferred.
- Remote Access Policy This policy is mainly for big companies where the user and their branches are outside their headquarters. It tells what should the users access, when they can work and on which software like SSH, VPN, RDP.
- Firewall Management Policy This policy has explicitly to do with its management, which
  ports should be blocked, what updates should be taken, how to make changes in the
  firewall, how long should be the logs be kept.
- Special Access Policy This policy is intended to keep people under control and monitor
  the special privileges in their systems and the purpose as to why they have it. These
  employees can be team leaders, managers, senior managers, system administrators, and
  such high designation based people.
- Network Policy This policy is to restrict the access of anyone towards the network resource and make clear who all will access the network. It will also ensure whether that person should be authenticated or not. This policy also includes other aspects like, who will authorize the new devices that will be connected with network? The documentation of network changes. Web filters and the levels of access. Who should have wireless connection and the type of authentication, validity of connection session?
- Email Usage Policy This is one of the most important policies that should be done
  because many users use the work email for personal purposes as well. As a result
  information can leak outside. Some of the key points of this policy are the employees
  should know the importance of this system that they have the privilege to use. They should
  not open any attachments that look suspicious. Private and confidential data should not be
  sent via any encrypted email.
- Software Security Policy This policy has to do with the software's installed in the user
  computer and what they should have. Some of the key points of this policy are Software of
  the company should not be given to third parties. Only the white list of software's should be
  allowed, no other software's should be installed in the computer. Warez and pirated
  software's should not be allowed.

#### 6.DESCRIBE ABOUT A MODEL FOR NETWORK SECURITY.

A model for much of what we will be discussing is captured, in very general terms, in Figure 1.3. A message is to be transferred from one party to another across some sort of Internet service.

A security-related transformation on the information to be sent, Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender

Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

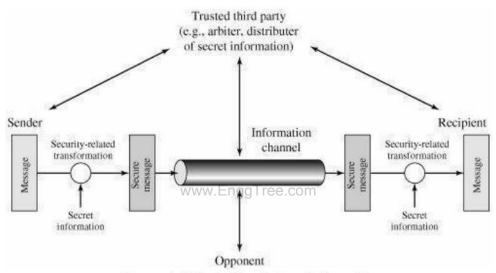


Figure 1.3 Model for Network Security

All the techniques for providing security have two components:

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

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

A general model of these other situations is illustrated by Figure 1.4, which reflects a concern for protecting an information system from unwanted access. Most readers are familiar with the concerns caused by the existence of hackers, who attempt to penetrate systems that can be accessed over a network. The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system. The intruder can be a disgruntled employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).

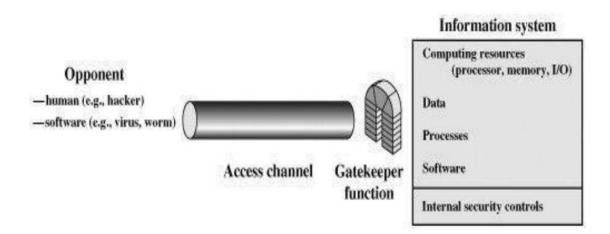


Figure 1.4 Network Access Security Model

Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and compilers. Programs can present two kinds of threats:

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

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

The security mechanisms needed to cope with unwanted access fall into two broad categories (see Figure 1.4). The first category might be termed a gatekeeper function. It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses, and other similar attacks. Once either an unwanted user or unwanted software gains access,

The second line of defense consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

#### 7. DESCRIBE ABOUT THE OSI SECURITY ARCHITECTURE

ITU-T Recommendation X.800, Security Architecture for OSI, defines such a systematic approach. The OSI security architecture is useful to managers as a way of organizing the task of providing security. This architecture was developed as an international standard, computer and communications vendors have developed security features for their products and services that relate to this structured definition of services and mechanisms.

The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as

- a. **Security attack:** Any action that compromises the security of information owned by an organization.
- b. **Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

c. **Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service. In the literature, the terms threat and attack are commonly used to mean more or less the same thing.

d.

Table 1.1 provides definitions taken from RFC 2828, InternetSecurity Glossary.

#### Threat

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

#### Attack

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

#### **ATTACKS**

The security attacks can be classified into two types' passive attacks and active attacks. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

www.EnggTree.com

#### **Passive Attacks**

Two types of passive attacks are the release of message contents and traffic analysis.

The **release of message contents** is easily understood (Figure 1.5a). A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

A second type of passive attack, **traffic analysis**, is subtler (Figure 1.5b). Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages.

Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Typically, the message traffic is not sent and received in an apparently normal fashion and the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.

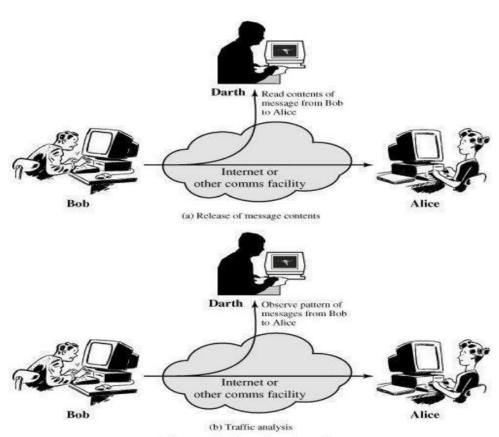


Figure 1.5 Passive Attacks

#### **Active Attacks**

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

A **masquerade** takes place when one entity pretends to be a different entity (Figure 1.6a). A masquerade attack usually includes one of the other forms of active attack. 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.

**Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (Figure 1.6b).

**Modification of messages** simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure 1.6c). For example, a message meaning "Allow John Smith to read confidential file *accounts*" is modified to mean "Allow Fred Brown to read confidential file *account*.

The **denial of service** prevents or inhibits the normal use or management of communications facilities (Figure 1.6d). This attack may have a specific target.

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success.

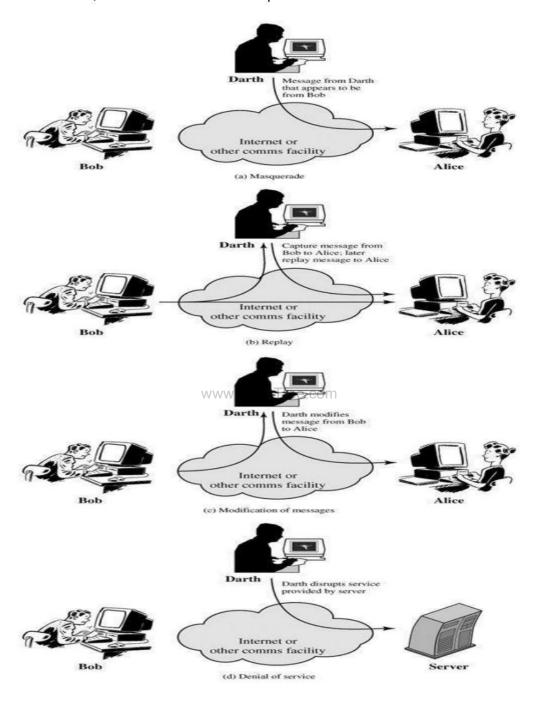


Figure 1.6 Active Attacks

#### 8.EXPLAIN ABOUT THE SERVICES AND MECHANISMS IN DETAILS.

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. Perhaps a clearer definition is found in RFC 2828, which provides the following definition: a processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms.

X.800 divides these services into five categories and fourteen specific services (Table 1.2)

#### Table 1.2 Security Services (X.800)

#### AUTHENTICATION

The assurance that the communicating entity is the one that it claims to be.

#### Peer Entity Authentication

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

#### Data-Origin Authentication

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

#### ACCESS CONTROL www.Engo

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

#### DATA CONFIDENTIALITY

The protection of data from unauthorized disclosure.

#### Connection Confidentiality

The protection of all user data on a connection.

#### Connectionless Confidentiality

The protection of all user data in a single data block

#### Selective-Field Confidentiality

The confidentiality of selected fields within the user data on a connection or in a single data block.

#### Traffic-Flow Confidentiality

The protection of the information that might be derived from observation of traffic flows.

#### DATA INTEGRITY

The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

#### Connection Integrity with Recovery

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

#### Connection Integrity without Recovery

As above, but provides only detection without recovery.

#### Selective-Field Connection Integrity

Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.

#### Connectionless Integrity

Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.

#### Selective-Field Connectionless Integrity

Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

#### NONREPUDIATION

Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

#### Nonrepudiation, Origin

Proof that the message was sent by the specified party.

#### Nonrepudiation, Destination

Proof that the message was received by the specified party.

#### **MECHANISMS**

Table 1.3 lists the security mechanisms defined in X.800. The mechanisms are divided into those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service

. Table 1.3 Security Mechanisms (X.800)

#### SPECIFIC SECURITY MECHANISMS

May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services

#### Encipherment

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

#### Digital Signature

Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).

#### Access Control

A variety of mechanisms that enforce access rights to resources.

#### Data Integrity

A variety of mechanisms used to assure the integrity of a data unit or stream of data units.

#### **Authentication Exchange**

A mechanism intended to ensure the identity of an entity by means of information exchange.

#### Traffic Padding

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

#### Routing Control

Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.

#### Notarization

The use of a trusted third party to assure certain properties of a data exchange.

#### PERVASIVE SECURITY MECHANISMS

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

#### Trusted Functionality

That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

#### Security Label

The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

#### **Event Detection**

Detection of security-relevant events

#### Security Audit Trail

Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

#### Security Recovery

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

naaTree.com

#### 9.EXPLAIN IN DETAIL ABOUT THE CLASSICAL ENCRYPTION TECHNIQUES.

Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption.

- Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the ciphertext.
- The two types of attack on an encryption algorithm are cryptanalysis, based on properties of the encryption algorithm, and brute-force, which involves trying all possible keys.
- Traditional (precomputer) symmetric ciphers use substitution and/or transposition techniques. Substitution techniques map plaintext elements (characters, bits) into ciphertext elements. Transposition techniques systematically transpose the positions of plaintext elements.

- Rotor machines are sophisticated precomputer hardware devices that use substitution techniques.
- Steganography is a technique for hiding a secret message within a larger one in such a
  way that others cannot discern the presence or contents of the hidden message.

An original message is known as the **plaintext**, while the coded message is called the **ciphertext**. The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**; restoring the plaintext from the ciphertext is **deciphering** or **decryption**. The many schemes used for encryption constitute the area of study known as **cryptography**.

Such a scheme is known as a **cryptographic system** or a **cipher**. Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**. Cryptanalysis is what the layperson calls "breaking the code" The areas of cryptography and cryptanalysis together are called **cryptology**.

#### SYMMETRIC CIPHER MODEL

A symmetric encryption scheme has five ingredients (Figure 1.7):

- Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.
- Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.
- Secret key: The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key
- Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the cipher text and the secret key and produces the original plaintext.

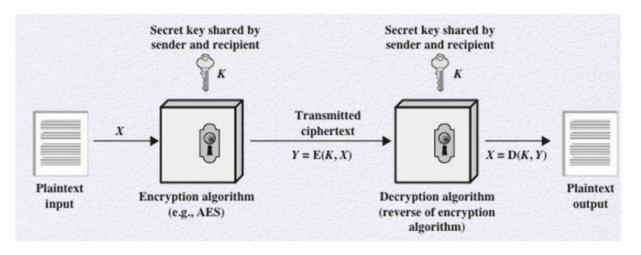


Figure 1.7 Simplified Model of Symmetric Encryption

There are two requirements for secure use of conventional encryption:

- 1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.
- 2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

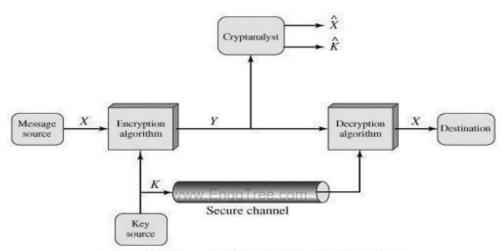


Figure 1.8 Model of Symmetric Cryptosystem

With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext  $Y=[Y1,Y2,.....Y_N]$ . We can write this as Y=E(K,X)This notation indicates that is produced by using encryption algorithm E as a function of the plaintext X , with the specific function determined by the value of the key K .

The intended receiver, in possession of the key, is able to invert the transformation:

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

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

#### Cryptography

Cryptographic systems are characterized along three independent dimensions:

#### The type of operations used for transforming plaintext to ciphertext:

All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (that is, that all operations are reversible). Most systems, referred to as *product systems*, involve multiple stages of substitutions and transpositions.

- 1. The number of keys used. If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.
- **2.** The way in which the plaintext is processed. A *block cipher* processes the input one block of elements at a time, producing an output block for each input block. A *stream cipher* processes the input elements continuously, producing output one element at a time, as it goes along.

#### 3. Cryptanalysis and Brute-Force Attack

Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintexts of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:

- **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plusperhaps some knowledge of the general characteristics of the plaintext oreven some sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
- Brute-force attack: The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

Table 1.4 summarizes the various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst. The most difficult problem is presented when all that is available is the ciphertext only.

Table 1.4 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst					
Ciphertext Only	Encryption algorithm     Ciphertext					
Known Plaintext	Encryption algorithm     Ciphertext     One or more plaintext-ciphertext pairs formed with the secret key					
Chosen Plaintext	Encryption algorithm     Ciphertext     Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key					
Chosen Ciphertext	Encryption algorithm     Ciphertext     Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key					
Chosen Text	Encryption algorithm     Ciphertext     Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key     Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key					

A **brute-force attack** involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained.

#### 11.EXPLAIN ABOUT THE SUBSTITUTION TECHNIQUES IN DETAILS.

The two basic building blocks of all encryption techniques are substitution and transposition. A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.1 If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

#### 1. Caesar Cipher

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

plain: meet me after the toga party cipher: PHHW PH DIWHU WKH WRJD SDUWB

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

plain: 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

cipher: DEFGHIJKLMNOPQRSTUVWXYZABC

Let us assign a numerical equivalent to each letter:

When letters are involved, the following conventions are used in this book. Plaintext is always in lowercase; ciphertext is in uppercase; key values are in italicized lowercase.

Let us assign a numerical equivalent to each letter:

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

n	0	p	q	r	S	t	u	V	W	X	y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Then the algorithm can be expressed as follows. For each plaintext letter, substitute the cipher text letter:

$$C = E(3, p) = (p + 3) \mod 26$$
 www.EnggTree.com

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

$$C = E(k, p) = (p + k) \mod 26$$

where takes on a value in the range 1 to 25. The decryption algorithm is simply

$$p = D(k, C) = (C - k) \mod 26$$

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. Three important characteristics of this problem enabled us to use a bruteforce cryptanalysis:

- 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.

```
PHHW PH DIWHU WKH WRJD SDUWB
KEY
        oggv og chvgt vjg vqic retva
        nffu nf bgufs uif uphb qbsuz
        meet me after the toga party
        1dds 1d zesdq sgd snfz ozqsx
        keer ke ydrep rfc rmey nyprw
        jbbq jb xcqbo qeb qldx mxoqv
        iaap ia wbpan pda pkcw lwnpu
        hzzo hz vaozm ocz ojbv kvmot
   8
        gyyn gy uznyl nby niau julns
        fxxm fx tymxk max mhzt itkmr
   10
  11
        ewwl ew sxlwj lzw lgys hsjlq
  12
        dvvk dv rwkvi kyv kfxr grikp
  13
        cuuj cu qvjuh jxu jewq fqhjo
        btti bt puitg iwt idvp epgin
        assh as other hvs houo dofhm
        zrrg zr nsgre gur gbtn cnegl
        yqqf yq mrfqd ftq fasm bmdfk
  17
  18
         xppe xp lqepc esp ezrl alcej
  19
         wood wo kpdob dro dygk zkbdi
         vnne vn joena egn expj yjach
  21
        ummb um inbmz bpm bwoi xizbg
        tlla tl hmaly aol avnh whyaf
  23
        skkz sk glzkx znk zumg vgxze
  24
        rjjy rj fkyjw ymj ytlf ufwyd
  25 qiix qi ejxiv xli xske tevxc
```

Figure 1.9 Brute-Force Cryptanalysis of Caesar Cipher

#### 2. Monoalphabetic Ciphers

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. A **permutation** of a finite set of elements is an ordered sequence of all the elements of, with each element appearing exactly once. For example, if  $S = \{a,b,c\}$ , there are six permutations of :

abc, acb, bac, bca, cab, cba

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

Recall the assignment for the Caesar cipher:

plain: a b c d e f g h l j kl m n o p q r s t u v w x y z

cipher: DEFGHIJKLMNOPQRSTUVWXYZABC

If, instead, the "cipher" line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4\*10<sup>26</sup> possible keys. This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Such an approach is referred to as a **monoalphabetic substitution cipher**, because a single cipher alphabet (mapping from plain alphabetto cipher alphabet) is used per message.

The ciphertext to be solved is

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English, such as is shown in Figure 1.9. If the message were long enough, this technique alone might be sufficient, but because this is a relatively short message, we cannot expect an exact match. In any case, the relative frequencies of the letters in the ciphertext (in percentages) are as follows:

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

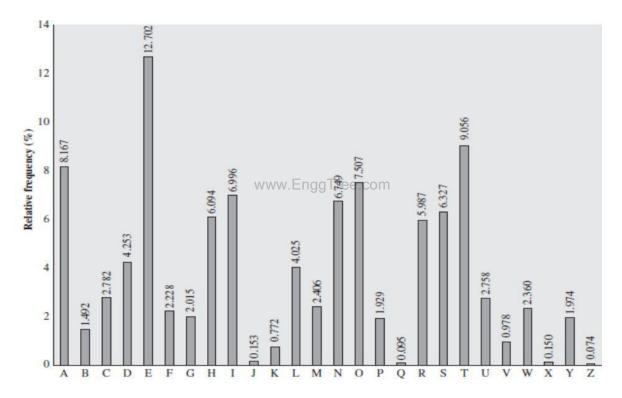


Figure 1.10 Relative Frequencies of Letters in English Text

That cipher letters P and Z are the equivalents of plain letters e and t, but it is not certain which is which. The letters S, U, O, M, and H are all of relatively high frequency and probably correspond to plain letters from the set {a, h, i, n, o, r, s}. The letters with the lowest frequencies (namely A, B, G, Y, I, J) are likely included in the set {b, j, k, q, v, x, z}.

A powerful tool is to look at the frequency of two-letter combinations, known as **digrams**. The most common such digram is th. In our ciphertext, the most common digram is ZW, which appears three times. So we make the correspondence of Z with t and W with h. Then, by our earlier hypothesis, we can equate P with e. Now notice that the sequence ZWP appears in the ciphertext, and we can translate that sequence as "the." This is the most frequent trigram (three-letter combination). Next, notice the sequence ZWSZ in the first line. We do not know that these four letters form a complete word, but if they do, it is of the form th\_t. If so, Sequates with a. So far, then, we have

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ

ta e e te a that e e a

VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX

e t ta t ha e ee a e th t a

EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

e e e tat e the t

Only four letters have been identified, but already we have quite a bit of the message. Continued analysis of frequencies plus trial and error should easily yield a solution from this point. The complete plaintext, with spaces added between words, follows:

it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow

Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet. A countermeasure is to provide multiple substitutes, known as homophones, for a single letter.

#### 3. Playfair Cipher

The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams. The Playfair algorithm is based on the use of a  $5 \times 5$  matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers's Have His Carcase

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

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

- **1.** Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.
- **2.** Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.
- **3.** Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.
- **4.** Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are  $26 \times 26 = 676$  digrams, so that identification of individual digrams is more difficult. Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult. For these reasons, the Playfair cipher was for a long time considered unbreakable. It was used as the standard field system by the British Army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

#### 4. Hill Cipher

Another interesting multiletter cipher is the Hill cipher, developed by the mathematician Lester Hill in 1929. Define the inverse M<sup>-1</sup> of a square matrix M by the equation M(M<sup>-1</sup>)= M<sup>-1</sup>M=I, where I is the identity matrix. I is a square matrix that is all zeros except for ones along the main diagonal from upper left to lower right. The inverse of a matrix does not always exist, but when

$$\mathbf{A} = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} \qquad \mathbf{A}^{-1} \mod 26 = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$$

$$\mathbf{A}\mathbf{A}^{-1} = \begin{pmatrix} (5 \times 9) + (8 \times 1) & (5 \times 2) + (8 \times 15) \\ (17 \times 9) + (3 \times 1) & (17 \times 2) + (3 \times 15) \end{pmatrix}$$

$$= \begin{pmatrix} 53 & 130 \\ 156 & 79 \end{pmatrix} \mod 26 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

it does, it satisfies the preceding equation. For example, om

To explain how the inverse of a matrix is computed, we begin by with the concept of determinant. For any square matrix  $(m \times m)$ , the **determinant** equals the sum of all the products that can be formed by taking exactly one element from each row and exactly one element from each column, with certain of the product terms preceded by a minus sign. For a 2  $\times$  2 matrix,

$$\begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$$

The determinant is  $k_{11}k_{22} - k_{12}k_{21}$ . For a 3×3 matrix, the value of the determinant is  $.k_{11}k_{22}k_{33} + k_{21}k_{32}k_{13} + k_{31}k_{12}k_{23} - k_{31}k_{22}k_{13} - k_{21}k_{12}k_{33} - k_{11}k_{32}k_{23}$ . If a square matrix **A** has a nonzero determinant, then the inverse of the matrix is computed as  $[A^{-1}]_{ij}$ =(det A)<sup>-1</sup> (-1)<sup>i+j</sup> (D<sub>ij</sub>)where (D<sub>ij</sub>) is the subdeterminant formed by deleting the jth row and the ith column of **A**, det(**A**) is the determinant of **A**, and (det A)<sup>-1</sup> is the multiplicative inverse of (det **A**) mod 26. Continuing our example,

$$\det\begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} = (5 \times 3) - (8 \times 17) = -121 \mod 26 = 9$$

We can show that  $9^{-1}$ mod26=3, because $9 \times 3 = 27$ mod26=1. Therefore, we compute the inverse of **A** as

$$\mathbf{A} = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}$$

$$\mathbf{A}^{-1} \mod 26 = 3 \begin{pmatrix} 3 & -8 \\ -17 & 5 \end{pmatrix} = 3 \begin{pmatrix} 3 & 18 \\ 9 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 54 \\ 27 & 15 \end{pmatrix} = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$$

**THE HILLALGORITHM** This encryption algorithm takes m successive plaintext letters and substitutes for them m ciphertext letters. The substitution is determined by m linear equations in

$$c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \mod 26$$
  
 $c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \mod 26$   
 $c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \mod 26$ 

which each character is assigned a numerical value (a=0,b=1,...z=25). For m=3, the system can be described as

This can be expressed in terms of row vectors and matrices:

$$(c_1 \ c_2 \ c_3) = (p \ p_2 \ p_3) \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \bmod 26$$

or

where  $\bf C$  and  $\bf P$  are row vectors of length 3 representing the plaintext and ciphertext, and  $\bf K$  is a 3 ×3 matrix representing the encryption key. Operations are performed mod 26.For example, consider the plaintext "paymoremoney" and use the encryption Key

$$\mathbf{K} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \text{ggTree.com}$$

The first three letters of the plaintext are represented by the vector (15 0 24). Then (15 0 24)K = (303 303 531) mod 26 = (17 17 11) = RRL. Continuing in this fashion, the ciphertext for the entire plaintext is RRLMWBKASPDH.

Decryption requires using the inverse of the matrix K. We can compute det K = 23, and therefore,  $(\det K)^{-1} \mod 26 = 17$ . We can then compute the inverse as

$$\mathbf{K}^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

This is demonstrated as

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \bmod{26} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

It is easily seen that if the matrix  $K^{-1}$  is applied to the ciphertext, then the plaintext is recovered.

In general terms, the Hill system can be expressed as

$$C = E(K, P) = PK \mod 26$$
  
 $P = D(K, C) = CK^{-1} \mod 26 = PKK^{-1} = P$ 

As with Playfair, the strength of the Hill cipher is that it completely hides single-letter frequencies. Indeed, with Hill, the use of a larger matrix hides more frequency information. Thus, a 3 ×3 Hill cipher hides not only single-letter but also two-letter frequency information.

Consider this example. Suppose that the plaintext "hillcipher" is encrypted using a Hill cipher to yield the ciphertext HCRZSSXNSP. Thus, we know that (78) Kmod26=(72)11 11)Kmod26=(17 25); and so on. Using the first two plaintext-ciphertext pairs, we have

$$\begin{pmatrix} 7 & 2 \\ 17 & 25 \end{pmatrix} = \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \mathbf{K} \mod 26$$

The inverse of X can be computed

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix}^{-1} = \begin{pmatrix} 25 & 22 \\ 1 & 23 \end{pmatrix}$$

SO

$$\mathbf{K} = \begin{pmatrix} 25 & 22 \\ 1 & 23 \end{pmatrix} \begin{pmatrix} 7 & 2 \\ 17 & 25 \end{pmatrix} = \begin{pmatrix} 549 & 600 \\ 398 & 577 \end{pmatrix} \mod 26 = \begin{pmatrix} 3 & 2 \\ 8 & 5 \end{pmatrix}$$

This result is verified by testing the remaining plaintext-ciphertext pairs.

#### 5. One Time Pad Cipher (or) Vernam Cipher

It is an unbreakable cryptosystem, described by Frank Miller in 1882, the one-time pad was reinvented by Gilbert Vernam in 1917 and it was later improved by the US Amry Major Joseph. It represents the message as a sequence of 0s and 1s. This can be accomplished by writing all numbers in binary, for example, or by using ASCII. The key is a random sequence of 0s and 1s of same length as the message.

Once a key is used, it is discarded and never used again. The system can be expressed as follows:

$$C_i = P_i \oplus K_i$$

$$C_i - i^{th} \text{ binary digit of cipher text} \qquad P_i - i^{th} \text{ binary digit of plaintext}$$

$$K_i - i^{th} \text{ binary digit of key} \qquad \oplus - \text{ exclusive OR opearaiton}$$

Thus the cipher text is generated by performing the bitwise XOR of the plaintext and the key. Decryption uses the same key. Because of the properties of XOR, decryption simply involves the same bitwise operation:

$$P_i = C_i \oplus K_i$$

#### Example

Alice wishes to send the message "HELLO" to Bob. If key material begins with "XMCKL" and the message is "HELLO", then use Vernam One Time Pad to Decrypt and Show the Encryption Process.

MESSAGE POSITION	H 7	E 4	L 11	L 11	O 14
KEY	X	M	С	K	L
POSITION	23	12	2	10	11

#### **OTP Encryption**

Н	Е	L	L	0	Message	
7	4	11	11	14	Message	
(H)	(E)	(L)	(L)	(O)		
23	12	2	10	11	Key	
(X)	(M)	(C)	(K)	(L)		
30	16	13	21	25	Message + Key	
4	16	13	21	25	Message + Key (mod 26)	
(E)	(Q)	(N)	(V)	(Z)		
E	Q	N	V	Z	Ciphertext	

Note: If a number is larger than 25, then the remainder afer subtraction of 26 is taken in Modular Arithmetic fashion

#### **OTP Decryption**

Е	Q	N	V	V <del>Z</del> WW E	Ciphertext
4	16	13	21	25	Ciphertext
(E)	(Q)	(N)	(V)	(Z)	200
23	12	2	10	11	Key
(X)	(M)	(C)	(K)	(L)	
-19	4	11	11	14	Ciphertext - Key
7	4	11	11	14	Ciphertext - Key (mod 26)
(H)	(E)	(L)	(L)	(O)	
Н	Е	L	L	0	Message

Note: If a number is negative then 26 is added to make the number positive

#### Example

Encryption

Plaintext is 00101001 and the key is 10101100, we obtain the ciphertext is,

Plaintext 00101001

Key 10101100

Ciphertext 10000101

Decryption

 Ciphertext
 10000101

 Key
 10101100

 Plaintext
 00101001

#### Advantages

- Encryption method is completely unbreakable for a cipher-text only known attack
- Chosen Plaintext (or) Ciphertext attacks is not possible

#### Disadvantages

- It requires a very long key which is expensive to produce and expensive to transmit.
- Once a key is used it is dangerous to reuse it for second message.

#### 6. Polyalphabetic Ciphers

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

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

**VIGEN`ERE CIPHER** The best known, and one of the simplest, polyalphabetic ciphers is the Vigenère cipher. In this scheme, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers with shifts of 0 through 25. Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter a. Thus, a Caesar cipher with a shift of 3 is denoted by the key value.

Express the Vigenère cipher in the following manner. Assume a sequence of plaintext letters and a key consisting of the sequence of letters, where typically < .The sequence of ciphertext letters is calculated as follows

$$C = C_0, C_1, C_2, \dots, C_{n-1} = E(K, P) = E[(k_0, k_1, k_2, \dots, k_{m-1}), (p_0, p_1, p_2, \dots, p_{n-1})]$$

$$= (p_0 + k_0) \mod 26, (p_1 + k_1) \mod 26, \dots, (p_{m-1} + k_{m-1}) \mod 26,$$

$$(p_m + k_0) \mod 26, (p_{m+1} + k_1) \mod 26, \dots, (p_{2m-1} + k_{m-1}) \mod 26, \dots$$

Thus, the first letter of the key is added to the first letter of the plaintext, mod 26, the second letters are added, and so on through the first letters of the plaintext. For the next letters of the plaintext, the key letters are repeated. This process continues until all of the plaintext sequence is encrypted. A general equation of the encryption process is

Ci = (pi + kimod m)mod 26
Decryption is a generalization of Equation pi= (Ci - kimod m)mod 26

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

key: deceptivedeceptive plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

#### 13.EXPLAIN ABOUT THE TRANSPOSITION TECHNIQUES IN DETAILS.

All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher. The simplest such cipher is the **rail fence** technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows. For example, to encipher the message "meet me after the toga party" with a rail fence of depth 2, we write the following:

The encrypted message is

#### **MEMATRHTGPRYETEFETEOAAT**

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

Thus, in this example, the key is 4312567.To encrypt, start with the column that is labeled 1, in this case column 3. Write down all the letters in that column. Proceed to column 4, which is labeled 2, then column 2, then column 1, then columns 5, 6, and 7. A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. For the type of columnar transposition just shown, cryptanalysis is fairly straightforward and involves laying out the ciphertext in a matrix and playing around with column positions. Digram and trigram frequency tables can be useful.

The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easily reconstructed. Thus, if the foregoing message is reencrypted using the same algorithm,

Key: 4 3 1 2 5 6 7

Input: ttnaapt

m t s u o a o d w c o i x k

nlypetz

Output: NSCYAUOPTTWLTMDNAOIEPAXTTOKZ

To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position. Thus, with 28 letters in the message, the original sequence of letters is

01 02 03 04 05 06 07 08 09 10 11 12 13 14

15 16 17 18 19 20 21 22 23 24 25 26 27 28

After the first transposition, we have

03 10 17 24 04 11 18 25 02 09 16 23 01 08

15 22 05 12 19 26 06 13 20 27 07 14 21 28

**MEMATRHTGPRYETEFETEOAAT** 

#### 14.EXPLAIN STEGANOGRAPHY IN DETAIL.

A plaintext message may be hidden in one of two ways. The methods of steganography conceal the existence of the message, whereas the methods of cryptography render the message unintelligible to outsiders by various transformations of the text.

A simple form of steganography, but one that is time-consuming to construct, is one in which an arrangement of words or letters within an apparently innocuous text spells out the real message. For example, the sequence of first letters of each word of the overall message spells out the hidden message. Figure shows an example in which a subset of the words of the overall message is used to convey the hidden message.

3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package. All Entry Forms and Fees Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told. by the 21st. Admin has improved here, though there's room for improvement still; just give us all two or three more years and we'll really show you! Please don't let these wretched 16t proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours.

Various other techniques have been used historically; some examples are the following: **Character marking:** Selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.

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

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

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

Steganography has a number of drawbacks when compared to encryption. It requires a lot of overhead to hide a relatively few bits of information, although using a scheme like that proposed in the preceding paragraph may make it more effective. Also, once the system is discovered, it becomes virtually worthless. This problem, too, can be overcome if the insertion method depends on some sort of key.

The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered. Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide.

#### 15. DESCRIBE ABOUT THE FOUNDATIONS OF MODERN CRYPTOGRAPHY

Modern encryption is the key to advanced computer and communication security. This stream of cryptography is completely based on the ideas of mathematics such as number theory and computational complexity theory as well as concepts of probability.

#### **Characteristics of Modern Cryptography**

There are four major characteristics that separate modern cryptography from the classical approach.

T 11 4 F D'W				
Table 1.5 Differences	hotwoon I radi	tional Encryption	n and Mader	h h cryntion
Table 1.5 Dillerences	Detween Haul	uonai Encivotioi	i aliu ivioueli	

Traditional Encryption	Modern Encryption
For making ciphertext, manipulation is done	For making ciphertext, operations are
in the characters of the plaintext	performed on binary bit sequence
The whole of the ecosystem is required to communicate confidentiality	Here, only the parties who want to execute secure communication possess the secret key
These are weaker as compared to modern encryption	The encryption algorithm formed by this encryption technique is stronger as compared to traditional encryption algorithms
It believes in the concept of security through obscurity	Its security depends on the publicly known mathematical algorithm

#### Context of Cryptography

Cryptology, the study of cryptosystems, can be subdivided into two branches -

- Cryptography
- Cryptanalysis

#### Cryptography

Cryptography is the art and science of making a cryptosystem that is capable of providing information security. Cryptography deals with the actual securing of digital data. It refers to the design of mechanisms based on mathematical algorithms that provide fundamental information security services.

#### Cryptanalysis

The art and science of breaking the cipher text is known as cryptanalysis. Cryptanalysis is the sister branch of cryptography and they both co-exist. The cryptographic process results in the cipher text for transmission or storage. It involves the study of cryptographic mechanism with the intention to break them. Cryptanalysis is also used during the design of the new cryptographic techniques to test their security strengths.

Note - Cryptography concerns with the design of cryptosystems, while cryptanalysis studies the breaking of cryptosystems.

#### Types of Modern Cryptography

Different algorithms have come up with powerful encryption mechanisms incorporated in them. It gave rise to two new ways of encryption mechanism for data security. These are:

- Symmetric key encryptionvww.EnggTree.com
- Asymmetric key encryption

#### Key

It can be a number, word, phrase, or any code that will be used for encrypting as well as decrypting any ciphertext information to plain text and vice versa.

Symmetric and asymmetric key cryptography is based on the number of keys and the way these keys work. Let us know about both of them in details:

#### Symmetric key encryption

Symmetric key encryption technique uses a straight forward method of encryption. Hence, this is the simpler among these two practices. In the case of symmetric key encryption, the encryption is done through only one secret key, which is known as "Symmetric Key", and this key remains to both the parties.

The same key is implemented for both encodings as well as decoding the information. So, the key is used first by the sender prior to sending the message, and on the receiver side, that key is used to decipher the encoded message.

One of the good old examples of this encryption technique is Caesar's Cipher. Modern examples and algorithms that use the concept of symmetric key encryption are RC4, QUAD, AES, DES, Blowfish, 3DES, etc.

#### Asymmetric Key Encryption

Asymmetric Encryption is another encryption method that uses two keys, which is a new and sophisticated encryption technique. This is because it integrates two cryptographic keys for implementing data security. These keys are termed as Public Key and Private Key.

The "public key", as the name implies, is accessible to all who want to send an encrypted message. The other is the "private key" that is kept secure by the owner of that public key or the one who is encrypting.

Encryption of information is done through public key first, with the help of a particular algorithm. Then the private key, which the receiver possesses, will use to decrypt that encrypted information. The same algorithm will be used in both encodings as well as decoding.

Examples of asymmetric key encryption algorithms are Diffie-Hellman and RSA algorithm.

#### Security Services of Cryptography

- Confidentiality of information.
- Data Integrity.
- Authentication.
  - o Message authentication.
  - o Entity authentication.
- Non-repudiation.

#### Cryptography Primitives

Cryptography primitives are nothing but the tools and techniques in Cryptography that can be selectively used to provide a set of desired security services –

- Encryption
- Hash functions
- Message Authentication codes (MAC)
- Digital Signatures

The following table shows the primitives that can achieve a particular security service on their own.

Primitives Service	Encryption	Hash Function	MAC	Digital Signature
Confidentiality	Yes	No	No	No
Integrity	No	Sometimes	Yes	Yes
Authentication	No	No	Yes	Yes
Non Reputation	No	No	Sometimes	Yes

Table 1.6 Primitives and Security Service

#### **Perfect Security**

Perfect Secrecy (or information-theoretic secure) means that the ciphertext conveys no information about the content of the plaintext ...... However, part of being provably secure is that you need as much key material as you have plaintext to encrypt.

#### Information Theory

Information theory studies the quantification, storage, and communication of information. It was originally proposed by Claude Shannon in 1948 to find fundamental limits on signal processing and communication operations such as data compression.

Its impact has been crucial to the success of the Voyager missions to deep space, the invention of the compact disc, the feasibility of mobile phones, the development of the Internet, the study of linguistics and of human perception, the understanding of black holes, and numerous other fields. The field is at the intersection of mathematics, statistics, computer science, physics, neurobiology, information engineering, and electrical engineering.

The theory has also found applications in other areas, including statistical inference, natural language processing, cryptography, neurobiology, human vision, the evolution and function of molecular codes (bioinformatics), model selection in statistics, thermal physics, quantum computing, linguistics, plagiarism detection, pattern recognition, and anomaly detection.

Important sub-fields of information theory include source coding, algorithmic complexity theory, algorithmic information theory, information-theoretic security, Grey system theory and measures of information.

Applications of fundamental topics of information theory include lossless data compression (e.g. ZIP files), lossy data compression (e.g. MP3s and JPEGs), and channel coding (e.g. for DSL).

A key measure in information theory is entropy. Entropy quantifies the amount of uncertainty involved in the value of a random variable or the outcome of a random process. For example, identifying the outcome of a fair coin flip (with two equally likely outcomes) provides less information (lower entropy) than specifying the outcome from a roll of a die (with six equally likely outcomes). Some other important measures in information theory are mutual information, channel capacity, error exponents, and relative entropy.

#### **Product Cryptosystems**

A product cipher combines two or more transformations in a manner intending that the resulting cipher is more secure than the individual components to make it resistant to cryptanalysis.

The product cipher combines a sequence of simple transformations such as substitution (S-box), permutation (P-box), and modular arithmetic. For transformation involving reasonable number of n message symbols, both of the foregoing cipher systems (the S-box and P-box) are by themselves wanting.

The combination could yield a cipher system more powerful than either one alone. This approach of alternatively applying substitution and permutation transformation has been used by IBM in the Lucifer cipher system, and has become the standard for national data encryption standards such as the Data Encryption Standard and the Advanced Encryption Standard. A product cipher that uses only substitutions and permutations is called a SP-network. Feistel ciphers are an important class of product ciphers.

### Downloaded from EnggTree.com

#### **CRYPTANALYSIS**

Cryptanalysis is the art of trying to decrypt the encrypted messages without the use of the key that was used to encrypt the messages. Cryptanalysis uses mathematical analysis & algorithms to decipher the ciphers.

The success of cryptanalysis attacks depends

- · Amount of time available
- Computing power available
- Storage capacity available

The following is a list of the commonly used Cryptanalysis attacks;

**Brute force attack**- this type of attack uses algorithms that try to guess all the possible logical combinations of the plaintext which are then ciphered and compared against the original cipher.

**Dictionary attack**— this type of attack uses a wordlist in order to find a match of either the plaintext or key. It is mostly used when trying to crack encrypted passwords.

Rainbow table attack— this type of attack compares the cipher text against precomputed hashes to find matches.

Other Attacks using Cryptanalysis www.EnggTree.com

**Known-Plaintext Analysis (KPA):** Attacker decrypts ciphertext with known partial plaintext.

**Chosen-Plaintext Analysis (CPA):** Attacker uses ciphertext that matches arbitrarily selected plaintext via the same algorithm technique.

Ciphertext-Only Analysis (COA): Attacker uses known ciphertext collections.

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.

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

#### **UNIT-II PART-A**

#### 1. Prove that 3 is a primitive root of 7.

(M/J-07)

That is, if is a primitive root of the prime number, then the numbers  $a \mod p$ ,  $a^2 \circ , ^{-1} \mod p$ 

, , 6.....

#### 2. Write any one technique of attacking RSA.

(A/M-11)

The most widely used public-key cryptosystem is RSA. The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.

THE FACTORING PROBLEM: We can identify three approaches to attacking RSA mathematically.

1. Factor *n* into its two prime factors .This enables calculation of

 $\mathcal{O}(n) = (p-1)x(q-1)$ , which in turn enables determination of  $\mathcal{O}(mod \mathcal{O}(n))$ .

- 2. Determine  $\mathcal{O}(n)$  directly, without first determining p and q. Again, this enables determination of  $\Theta(mod \mathcal{O}(n))$ .
- 3. Determine *d* directly, without first determining  $\mathcal{O}(n)$ .

#### 3. What is differential cryptanalysis?

(N/D-08)

Differential cryptanalysis is a technique in which chosen plaintexts with particular XOR difference patterns are encrypted. The difference patterns of the resulting ciphertext provide information that can be used to determine the encryption key.

#### 4. What is linear cryptanalysis?

(A/M-12)

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

#### 6. What are the different modes of operation in DES?

(A/M-11)

- 1. Double DES
- 2. Triple DES
- 3. Electronic Code Book
- 4. Counter mode
- 5. Cipher block chaining mode
- 6. Cipher Feedback mode

#### 7. What are the CFB and OFB modes?

(M/J-07)

The Cipher Feedback (CFB) mode and the Output Feedback (OFB) mode are two standard modes of operation 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. OFB mode is similar to the CFB mode except that the quantity exclusive-oared with each plaintext block is generated independently of both the plaintext and ciphertext.

8. What is DES? (M/J-09)

Data Encryption Standard (DES) is a widely-used method of data encryption using a private (secret) key. DES applies a 56-bit key to each 64-bit block of data. The process can run in several modes and involves 16 rounds or operations.

#### 9. Compare the symmetric and asymmetric key cryptography. (A/M-12)

Symmetric Encryption uses a single secret key that needs to be shared among the people who needs to receive the message while Asymmetric encryption uses a pair of public key, and a private key to encrypt and decrypt messages when communicating.

- 1. Symmetric Encryption is an age old technique while asymmetric Encryption is relatively new.
- 2. Asymmetric Encryption was introduced to complement the inherent problem of the need to share the key in symmetric encryption model eliminating the need to share the key by using a pair of public-private keys.

#### 10. What are the disadvantages of double DES?

(N/D-12)

The following are the disadvantages of double DES

- 1. Reduction to a single stagev. Engg Tree.com
- 2. Meet in the middle attacks.
- 3. Double DES is less secure than triple DES.
- 4. Double DES is within brute force attack.

#### UNIT II SYMMETRIC KEY CRYPTOGRAPHY

MATHEMATICS OF SYMMETRIC KEY CRYPTOGRAPHY: Algebraic Structures - Modular arithmetic - Euclid's Algorithm - Congruence and Matrices - Groups, Rings, Fields - Finite fields - SYMMETRIC KEY CIPHERS: SDES - Block cipher principles of DES - Strength of DES - Differential and Linear Cryptanalysis - Block cipher design principles - Block cipher mode of operation - Evaluation criteria of AES - Advanced Encryption Standard - RC4 - Key Distribution

# PART-B 1.WRITE SHORT NOTES ON MATHEMATICS OF SYMMETRIC KEY CRYPTOGRAPHY

#### a. ALGEBRAIC STRUCTURES

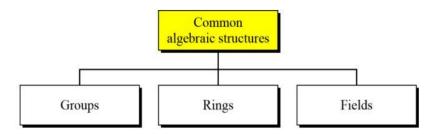


Figure 2.1 Common Algebraic Structures

www.EnggTree.com

#### i. Groups, Rings, Fields

Groups, rings, and fields are the fundamental elements of a branch of mathematics known as abstract algebra, or modern algebra.

#### Groups

A group G , sometimes denoted by  $\{G,^*\}$  ,is a set of elements with a binary operation denoted by \* that associates to each ordered pair (a,b) of elements G in an element( $a^*b$ ) in , 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, , in G.

(A3) Identity element: There is an element e in G such

that a\*e=e\*a=a for all in G.

(A4) Inverse element: For each a in G, there is an element

a' in G such that a\*a'=a'\*a=e.

If a group has a finite number of elements, it is referred to as a **finite group**, and the **order** of the group is equal to the number of elements in the group. Otherwise, the group is an **infinite group**.

A group is said to be **abelian** if it satisfies the following additional condition:

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

**CYCLIC GROUP:** A group is cyclic if every element of G is a power  $a^k$  (k is an integer) of a fixed element  $a\pounds G$ . The element is a said to generate the group G or to be a generator of

G.A cyclic group is always abelian and may be finite or infinite.

### Downloaded from EnggTree.com

#### Rings

A ring R, sometimes denoted by  $\{R, +, X\}$ , 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

(A1-A5) R is an abelian group with respect to addition; that is, R satisfies axioms A1 through A5. For the case of an additive group, we denote the identity element as 0 and the inverse of a as -a.

(M1) Closure under multiplication: If a and b belong to R, then ab is also

in R.

(M2) Associativity of multiplication: a(bc) = (ab)c for all a, b, c in R.

(M3) Distributive laws: a(b+c) = ab + ac for all a, b, c in R.

(a + b)c = ac + bc for all a, b, c in R.

A ring is said to be **commutative** if it satisfies the following additional condition:

(M4) Commutativity of multiplication: ab = ba for all a, b in R.

Next, we define an integral domain, which is a commutative ring that obeys the following axioms

(M5) Multiplicative identity: There is an element 1 in R such

that a1 = 1a = a for all a in R.

(M6) No zero divisors: If a, b in R and ab = 0, then either a = 0

or b = 0. www.EnggTree.com

#### **Fields**

A field F, sometimes denoted by  $\{F, +, X\}$ , is a set of elements with two binary operations, called addition and subtraction, such that for all a, b, c , in F the following axioms are obeyed

(A1-M6) F is an integral domain; that is, F satisfies axioms A1 through A5 and M1 through M6.

(M7) Multiplicative inverse: For each a in F, except 0, there is an element  $a^{-1}$  in F such that  $aa^{-1} = (a^{-1})a = 1$ .

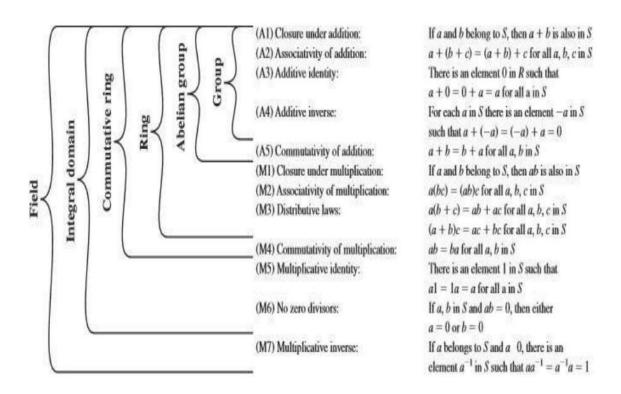


Figure 2.2 Groups, Ring and Field

www.EnggTree.com

#### b. MODULAR ARITHMETIC

If is an integer and n is a positive integer, we define a mod n to be the remainder when a is divided by n. The integer n is called the modulus. Thus, for any integer a, we can rewrite Equation as follows

$$a = qn + r \qquad 0 \le r < n; q = \lfloor a/n \rfloor$$

$$a = \lfloor a/n \rfloor \times n + (a \mod n)$$

$$11 \mod 7 = 4; \qquad -11 \mod 7 = 3$$

Two integers a and b are said to be **congruent modulo** n, if  $(a \mod n) = (b \mod n)$ . This is written as  $a = b \pmod{n}$ .

$$73 \equiv 4 \pmod{23}$$
;  $21 \equiv -9 \pmod{10}$ 

Note that if  $a = 0 \pmod{n}$ , then n|a.

#### **Modular Arithmetic Operations**

A kind of integer arithmetic that reduces all numbers to one of a fixed set [0,....,n-1] for some number n. Any integer outside this range is reduced to one in this range by taking the remainder after division by n.

Modular arithmetic exhibits the following properties

- 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.  $[(a \mod n) \times (b \mod n)] \mod n = (a \times b) \mod n$

We demonstrate the first property. Define  $(a \mod n) = r_a$  and  $(b \mod n) = r_b$ . Then we can write  $a = r_a + jn$  for some integer j and  $b = r_b + kn$  for some integer k. Then

$$(a+b) \bmod n = (r_a + jn + r_b + kn) \bmod n$$

$$= (r_a + r_b + (k+j)n) \bmod n$$

$$= (r_a + r_b) \bmod n$$

$$= [(a \bmod n) + (b \bmod n)] \bmod n$$

The remaining properties are proven as easily. Here are examples of the three properties:

**Table 2.1 Arithmetic Modulo 8** 

11 mod 8 = 
$$3$$
, 15/mod 8 = 10 mod 8 = 2  
[(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

	0	1	2	3	4	5	6	7
	0	1	2	3	4	5	6	7
Г	1	2	3	4	5	6	7	0
	2	3	4	5	6	7	0	1
Г	3	4	5	6	7	0	1	2
Г	4	5	6	7	0	1	2	3
Г	5	6	7	0	1	2	3	4
Г	6	7	0	1	2	3	4	- 5
Г	7	0	1	2	3	4	5	6

(a) Addition modulo 8

	0	1	2	3	4	5	6	7
Γ	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
ı	0	2	4	6	0	2	4	6
1	0	3	6	1	4	7	2	5
1	0	4	0	4	0	4	0	4
ı	0	5	2	7	4	1	6	3
ı	0	6	4	2	0	6	4	2
ı	0	7	6	5	4	3	2	1

(b) Multiplication modulo 8

0	
7	1
6	===
5	3
4	_
3	5
2	_
1	7
	5

(c) Additive and multiplicative inverses modulo 8

#### c. EUCLID'S ALGORITHM

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

#### **Greatest Common Divisor**

Recall that nonzero b is defined to be a divisor of a if a =mb for some m, where a,b, and m are integers. We will use the notation gcd(a, b) to mean the greatest common divisor of a and b .The greatest common divisor of a and b is the largest integer that divides both a and b

.We also define gcd(0,0) = 0.

#### **Algorithm**

The Euclid's algorithm (or Euclidean Algorithm) is a method for efficiently finding the greatest common divisor (GCD) of two numbers. The GCD of two integers X and Y is the largest number that divides both of X and Y (without leaving a remainder).

```
For every non-negative integer, a and any positive integer b
          gcd(a, b) = gcd(b, a mod b)
                Algorithm Euclids (a, b)
                        \alpha = a
                        \beta = b
                                         www.EnggTree.com
                        while (\beta > 0)
                                Rem = \alpha \mod \beta
                                \alpha = \beta
                                \beta = Rem
                        return\alpha
  Steps for Another Method
a = q1b + r1; 0 < r1 < b
b = q2r1 + r2; 0 < r2 < r1
r1 = q3r2 + r3; 0 < r3 < r2
rn-2 = qnrn-1 + rn;
                       0 < rn < rn-1
rn-1 = q1rn + 0
d = gcd(a, b) = rn
    Example 1:
        gcd (55, 22) = gcd (22, 55 mod 22)
                         = \gcd(22, 11)
                         = \gcd(11, 22 \mod 11)
                         = \gcd(11, 0)
        gcd (55, 22) is 11
```

```
Example 2:
   gcd(30, 50) = gcd(50, 30 \mod 50)
                   = \gcd(50, 30)
                   = gcd (30, 50 mod 30)
                   = \gcd(30, 20)
                   = \gcd(20, 30 \mod 20)
                   = \gcd(20, 10)
                  = \gcd(10, 20 \mod 10)
                  = gcd (10, 0)
gcd (30, 50) is 10
Another Method
              To find gcd (30,50)
                       50
                                   = 1 \times 30 + 20
                                                         gcd (30, 20)
                                   = 1 \times 20 + 10
                       30
                                                         gcd (20,10)
                       20
                                   = 1 \times 10 + 10
                                                         gcd (10,10)
                                   = 1 \times 10 + 0
                                                         gcd (10,0)
              Therefore, gcd(30,50) = 10
                                  www.EnggTree.com
Example 3:
   gcd (1970, 1066) = gcd (1066, 1970 mod 1066)
                         =gcd (1066, 904)
                         =gcd (904, 1066 mod 904)
                         =gcd (904, 162)
                         =gcd (162, 904 mod 162)
                         =gcd (162, 94)
                         =gcd (94, 162 mod 94)
                         =gcd (94, 68)
                         =gcd (68, 94 mod 68)
                         =gcd (68, 26)
                         =gcd (26, 68 mod 26)
                         =gcd (26, 16)
                         =gcd (16, 26 mod 16)
                         =gcd (16, 10)
                         =gcd (10, 16 mod 10)
                         =gcd (10, 6)
                         =gcd (6, 10 mod 6)
                         =gcd (6, 4)
```

## Downloaded from EnggTree.com

```
=gcd (4, 6 mod 4)
=gcd (4, 2)
=gcd (2, 4 mod 2)
=gcd (2, 0)
```

gcd (1970, 1066) is 2

#### Another Method

To find gcd (1970, 1066)

1970	= 1 x 1066 + 904	gcd (1066, 904)
1066	= 1 x 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	www. <b>x_6</b> ngg4rree.com	gcd (6, 4)
6	$= 1 \times 4 + 2$	gcd (4, 2)
4	$= 2 \times 2 + 0$	gcd (2, 0)

Therefore, gcd (1970, 1066) = 2

#### **Extended Euclidean Algorithm**

Extended Euclidean Algorithm is an efficient method of finding modular inverse of an integer.

Euclid's algorithm can be improved to give not just gcd (a, b), but also used to find the multiplicative inverse of a number with the modular value.

#### Example 1

Find the Multiplicative inverse of 17 mod 43

17-1 mod 43

17\*X = mod 43

X= 17-1 mod 43

43 = 17 \* 2 + 9

17 = 9 \* 1 + 8

9 = 8\*1+1

Rewrite the above equation

 $9+8(-1) = 1 \rightarrow (1)$ 

 $17+9(-1) = 8 \rightarrow (2)$ 

 $43+17(-2) = 9 \rightarrow (3)$ 

## Downloaded from EnggTree.com

```
Substitution
sub equ 2 in equ 1
(1) \rightarrow 9+8(-1) = 1 [Sub 17+9(-1) = 8]
9+(17+9(-1))(-1)=1
9+17(-1)+9(1)=1
17(-1)+9(2) = 1 \rightarrow (4)
Now sub equ (3) in equ (4)
43+17(-2) = 9 \rightarrow (3)
17(-1)+(43+17(-2))(2)=1
17(-1)+43(2)+17(-4)=1
17(-5)+43(2) = 1 \rightarrow (5)
Here -5 is the multiplicative inverse of 17. But inverse cannot be negative
17-1 \mod 43 = -5 \mod 43 = 38
So, 38 is the multiplicative inverse of 17.
Checking, 17* X ≡ 1 mod 43
                17 * 38 \equiv 1 \mod 43
                646 \equiv 1 \mod 43 (15*43 = 645)
Example 2
                                          www.EnggTree.com
Find the Multiplicative inverse of 1635 mod 26
1635-1 mod 26
1635 = 26(62) + 23
26 = 23(1) + 3
23 = 3(7) + 2
3 = 2(1) + 1
Rewriting the above equation
3+2(-1)=1 \rightarrow (1)
23+3(-7) = 2 \rightarrow (2)
26+23(-1) = 3 \rightarrow (3)
1635+26(-62) = 23 \rightarrow (4)
Substitution
sub equ (2) in equ (1)
(2) \Rightarrow 23+3(-7) = 2
3+2(-1)=1
3+(23+3(-7))(-1)=1
3+23(-1)+3(7)=1
3(8)+23(-1) = 1 \rightarrow (5)
```

```
sub equ (3) in equ (5)
26+23(-1) = 3 \rightarrow (3)
(26+23(-1))(8) + 23(-1) = 1
26(8) + 23 (-8) + 23 (-1) =1
26(8) + 23(-9) = 1 \rightarrow (6)
Sub equ (4) in equ (6)
1635+26(-62) = 23 \rightarrow (4)
26 (8) + (1635 + 26 (-62)) (-9) = 1
26 (8) + 1635 (-9) + 26 (558) =1
1635(-9) + 26(566) = 1 \rightarrow (7)
From equ (7) -9 is inverse of 1635, but negative cannot be inverse.
1635-1 \mod 26 = -9 \mod 26 = 17
So, the inverse of 1635 is 17.
Checking, 1635*X \equiv 1 \mod 26
                1635 * 17 \equiv 1 \mod 26
                27795 \equiv 1 \mod 26 (1069*26 = 27794)
```

#### d. CONGRUENCE AND MATRICES

#### **Properties of Congruences**

www.EnggTree.com

#### Congruences have the following properties:

```
1. a = b \pmod{n} if n \mid (a - b).

2. a = b \pmod{n} implies b = a \pmod{n}.

3. a = b \pmod{n} and b = c \pmod{n} imply a = c \pmod{n}.
```

To demonstrate the first point, if n|(a-b), then (a-b)=kn for some k. So we can write a=b+kn. Therefore,  $(a \bmod n)=$  (remainder when b+kn is divided by n)= (remainder when b is divided by n)= ( $b \bmod n$ ).

```
23 = 8 \pmod{5} because 23 - 8 = 15 = 5 \times 3

-11 = 5 \pmod{8} because -11 - 5 = -16 = 8 \times (-2)

81 = 0 \pmod{27} because 81 - 0 = 81 = 27 \times 3
```

The remaining points are as easily proved.

#### **Matrices**

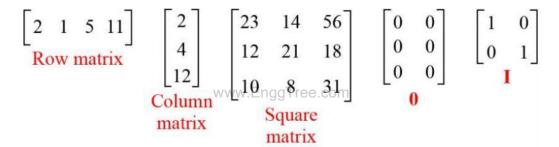
Matrix is a rectangular array in mathematics, arranged in rows and columns of numbers, symbols or expressions.

A matrix will be represented with their dimensions as I x m where I defines the row and m defines the columns

### *m* columns

#### **Examples of Matrices**

- 1. Row Matrix
- 2. Column Matrix
- 3. Square Matrix
- 4. Zero Matrixes
- 5. Identity Matrix



# e. FINITE FIELDS FINITE FIELDS OF THE FORM GF(p)

The finite field of order is generally written; GF stands for Galois field,in honor of the mathematician who first studied finite fields

#### Finite Fields of Order p

For a given prime, , we define the finite field of order , , as the set of integers together with the arithmetic operations modulo .

The simplest finite field is GF(2). Its arithmetic operations are easily summarized:

In this case, addition is equivalent to the exclusive-OR (XOR) operation, and multiplication is equivalent to the logical AND operation.

Finding the Multiplicative Inverse in It is easy to find the multiplicative inverse of an element in for small values of .You simply construct a multiplication table, such as shown in Table 2.2b,and the desired result can be read directly. However, for large values of , this approach is not practical. p p GF(p) GF(p)

## Downloaded from EnggTree.com

Table 2.2 Arithmetic in GF(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
L	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
١	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
5	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

www.EnggTree.com

#### i. Polynomial Arithmetic

We are concerned with polynomials in a single variable and we can distinguish three classes of polynomial arithmetic. • Ordinary polynomial arithmetic, using the basic rules of algebra. • Polynomial arithmetic in which the arithmetic on the coefficients is performed modulo

;that is,the coefficients are in .

Polynomial arithmetic in which the coefficients are in ,and the polynomials are defined modulo a polynomial whose highest power is some integer .

#### **Ordinary Polynomial Arithmetic**

A polynomial of degree (integer) is an expression of the form

A polynomial of degree n (integer  $n \ge 0$ ) is an expression of the form

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = \sum_{i=0}^n a_i x^i$$

where the  $a_i$  are elements of some designated set of numbers S, called the **coefficient set**, and  $a_n \neq 0$ . We say that such polynomials are defined over the coefficient set S.

A zero-degree polynomial is called a **constant polynomial** and is simply an element of the set of coefficients. An *n*th-degree polynomial is said to be a **monic polynomial** if  $a_n = 1$ .

In the context of abstract algebra, we are usually not interested in evaluating a polynomial for a particular value of x [e.g., f(7)]. To emphasize this point, the variable x is sometimes referred to as the **indeterminate**.

Addition and subtraction are performed by adding or subtracting corresponding coefficients. Thus, if

$$f(x) = \sum_{i=0}^{n} a_i x^i;$$
  $g(x) = \sum_{i=0}^{m} b_i x^i;$   $n \ge m$ 

then addition is defined as

$$f(x) + g(x) = \sum_{i=0}^{m} (a_i + b_i)x^i + \sum_{i=m+1}^{n} a_i x^i$$

and multiplication is defined as

$$f(x) \times g(x) = \sum_{i=0}^{n+m} c_i x^i$$

where

$$c_k = a_0b_k + a_1b_{k-1} + \cdots + a_{k-1}b_1 + a_kb_0$$

As an example, let  $f(x) = x^3 + x^2 + 2$  and  $g(x) = x^2 - x + 1$ , where S is the set of integers. Then

$$f(x) + g(x) = x^3 + 2x^{20} x + 3$$
  

$$f(x) - g(x) = x^3 + x + 1$$
  

$$f(x) \times g(x) = x^5 + 3x^2 - 2x + 2$$

#### Polynomial Arithmetic with Coefficients in

Let us now consider polynomials in which the coefficients are elements of some field F; we refer to this as a polynomial over the field F. In that case, it is easy to show that the set of such polynomials is a ring, referred to as a polynomial ring. That is, if we consider each distinct polynomial to be an element of the set, then that set is a ring 8 when polynomial arithmetic is performed on polynomials over a field, then division is possible. Note that this does not mean that exact division is possible. Let us clarify this distinction. Within a field, given two elements and, the quotient is also an element of the field. However, given a ring that is not a field, in Ra /b ba Zp

$$x^{3} + x^{2} + 2$$

$$+ (x^{2} - x + 1)$$

$$x^{3} + 2x^{2} - x + 3$$

(a) Addition

(b) Subtraction

$$\begin{array}{r}
x^3 + x^2 + 2 \\
\times (x^2 - x + 1) \\
\hline
x^3 + x^2 + 2 \\
-x^4 - x^3 - 2x \\
\hline
x^5 + x^4 + 2x^2 \\
\hline
x^5 + 3x^2 - 2x + 2
\end{array}$$

(c) Multiplication

$$\begin{array}{r}
 x^2 - x + 1 \overline{\smash)x^3 + x^2} + 2 \\
 \underline{x^3 - x^2 + x} \\
 \underline{2x^2 - x + 2} \\
 \underline{2x^2 - 2x + 2} \\
 x
 \end{array}$$

(d) Division

Figure 2.3 Examples of Polynomial Arithmetic

A polynomial over a field is called irreducible if and only if cannot be expressed as a product of two polynomials, both over, and both of degree lower than that of. By analogy to integers, an irreducible polynomial is also called a prime polynomial.

#### f. SYMMETRIC KEY CIPHERS

Symmetric ciphers use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. They are faster than asymmetric ciphers and allow encrypting large sets of data. However, they require sophisticated mechanisms to securely distribute the secret keys to both parties

### **Definition**A symmetric cipher defined over (K, M, C), where:

- K a set of all possible keys,
- M a set of all possible messages,
- · C a set of all possible ciphertexts

is a pair of efficient algorithms (E, D), where:

- E: K × M -> C
- D: K × C -> M

such that for every m belonging to M, k belonging to K there is an equality:

- D(k, E(k, m)) = m (the consistency rule)
- Function E is often randomized
- Function D is always deterministic

## Downloaded from EnggTree.com

#### Types of keys are used in symmetric key cryptography

Symmetric encryption (figure 2.4) uses a single key that needs to be shared among the people who need to receive the message while asymmetrical encryption uses a pair of public key and a private key to encrypt and decrypt messages when communicating.

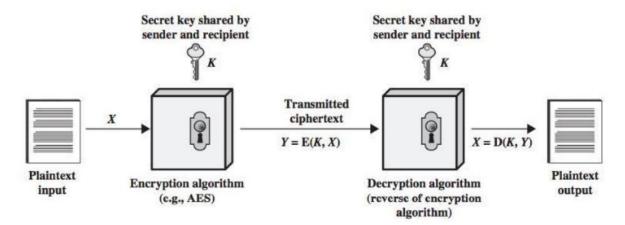


Figure 2.4 Simplified Model of Symmetric Encryption

#### 2.DESCRIBE ABOUT SIMPLIFIED DATA ENCRYPTION STANDARD (S-DES) IN DETAILS.

The overall structure of the simplified DES shown in Figure 2.5. The S-DES encryption algorithm takes an 8-bit block of plaintext (example: 10111101) and a 10-bit key as input and produces an 8-bit block of ciphertext as output.

The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext.

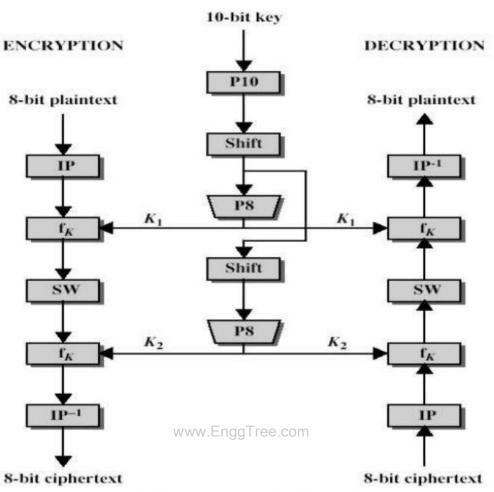


Figure 2.5 Overview of S-DES Algorithm

#### The encryption algorithm involves five functions:

- An initial permutation (IP)
- A complex function labeled fk, which involves both permutation and substitution operations and depends on a key input.
- A simple permutation function that switches (SW) the two halves of the data.
- The function fk again.

A permutation function that is the inverse of the initial permutation

The function fk takes as input not only the data passing through the encryption algorithm, but also an 8-bit key. Here a 10-bit key is used from which two 8-bit subkeys are generated.

The key is first subjected to a permutation (P10). Then a shift operation is performed. The output of the shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K1).

The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K2).

The encryption algorithm can be expressed as a composition composition1 of functions:

IP-1 o fK2 o SW o fk1 o IP, which can also be written as

Ciphertext = IP-1 (fK2 (SW (fk1 (IP (plaintext)))))

Where

K1 = P8 (Shift (P10 (Key)))

K2 = P8 (Shift (shift (P10 (Key))))

Decryption can be shown as Plaintext = IP-1 (fK1 (SW (fk2 (IP (ciphertext)))))

#### S-DES Key Generation

S-DES depends on the use of a 10-bit key shared between sender and receiver. From this key, two 8-bit subkeys are produced for use in particular stages of the encryption and decryption algorithm.(Figure 2.6)

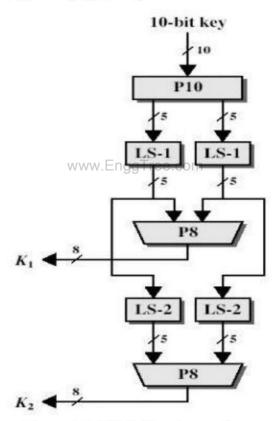


Figure 2.6 S-DES Key Generation

First, permute the key in the following fashion. Let the 10-bit key be designated as (k1, K2, k3, k4, k5, k6, k7, k8, k9, k10). Then the permutation P10 is defined as:

P10 (k1, K2, k3, k4, k5, k6, k7, k8, k9, k10) = (k3, k5, K2, k7, k4, k10 10, k1, k9, k8, k6).

P10 can be concisely defined by the display:

				P	10				
3	5	2	7	4	10	1	9	8	6

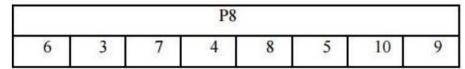
This table is read from left to right; each position in the table gives the identity of the input bit that produces the output bit in that position. So, the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on.

#### Example

The 10 bit key is (1010000010), now find the permutation from P10 for this key so it becomes (10000 01100).

Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits. In our example, the result is (00001 11000).

Next, apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:



So, The result is subkey 1 (K1). In our example, this yield (10100100).

Then go back to the pair of 5-bit strings produced by the two LS-1 functions and performs a circular left shift of 2 bit positions on each string. In our example, the value (00001 11000) becomes (00100 00011).

www.EnggTree.com

Finally, P8 is applied again to produce K2. In our example, the result is (01000011).

#### S-DES Encryption

Encryption involves the sequential application of five functions (Figure 2.7).

#### 1. Initial Permutations

The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function

			IP				
2	6	3	1	4	8	5	7

The plaintext is 10111101

Permutated output is 01111110

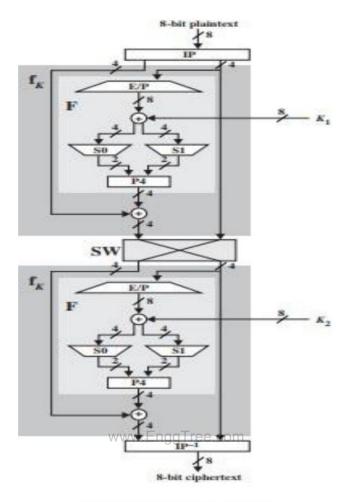


Figure 2.7 S-DES Encryption

#### 2. The Function fk

The most complex component of S-DES is the function fk, which consists of a combination of permutation and substitution functions. The functions can be expressed as follows. Let L and R be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to f K, and let F be a mapping (not necessarily one to one) from 4-bit strings to 4-bit strings. Then we let

$$Fk(L, R) = (L \oplus F(R, SK), R)$$

Where SK is a sub key and ⊕is the bit-by- bit exclusive OR function Now, describe the mapping F. The input is a 4-bit number (n1 n2 n3 n4). The first operation is an expansion/permutation operation:

			E/1	P			
4	1	2	3	2	3	4	1

Now, find the E/P from IP

IP = 01111110, it becomes

E/P = 01111101

Now, XOR with K1

The first 4 bits (first row of the preceding matrix) are fed into the S-box S0 to produce a 2- bit output, and the remaining 4 bits (second row) are fed into S1 to produce another 2-bit output.

These two boxes are defined as follows:

The S-boxes operate as follows. The first and fourth input bits are treated as a 2bit number that specify a row of the S-box, and the second and third input bits specify a column of the S-box. Each s box gets 4-bit input and produce 2 bits as output. It follows 00- 0, 01-1, 10-2, 11-3 scheme.

Here, take first 4 bits,

ake first 4 bits, Second 4 bits 
$$S_0 \Rightarrow 1101$$
  $S_1 \Rightarrow 1001$   $S_2 \Rightarrow 1001$   $S_1 \Rightarrow 1001$   $S_2 \Rightarrow 1001$   $S_1 \Rightarrow 1001$   $S_2 \Rightarrow 1001$   $S_2 \Rightarrow 1001$   $S_1 \Rightarrow 1001$   $S_2 \Rightarrow 1001$   $S_3 \Rightarrow 1001$ 

So, we get 1110

Now, find P<sub>4</sub>

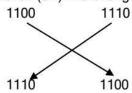
	P4		
2	4	3	\1 <sub>\\\\</sub> \.

After P<sub>4</sub>, the value is 1011

Now, XOR operation 1011⊕ 0111 => 1100

#### The Switch function 16.

The switch function (sw) interchanges the left and right 4 bits.



#### 17. Second function fk

➤ First, do E/P function and XOR with K<sub>2</sub>, the value is 01101001⊕01000011, the answer is 00101010

Now, find S<sub>0</sub> and S<sub>1</sub>

$$S_0 \Rightarrow 00 -> 0$$
  $S_1 => 10 -> 2$   $01 -> 1 => 0 = 00$   $01 -> 1 => 0 => 00$ 

Value is 0000

➤ Now, find P<sub>4</sub> and XOR operation

 $0000 \oplus 1110 = 1110$ , then concatenate last 4 bits after interchange in sw. After P<sub>4</sub>

Now value is 11101100

#### 18. Find IP-1

			IP -	-1			
4	1	3	5	7	2	8	6

So, value is 01110101

The Ciphertext is 01110101

#### 2.8.3 S-DES Decryption

- Decryption involves the sequential application of five functions.
- 1. Find IP
- After IP, value is 11101100
- 2. Function fk
  - After step 2, the answer is 11101100
- 3. Swift
  - The answer is 11001110
- 4. Second fk
  - The answer is 011111110
- 5. Find IP-1
  - 101111101 -> Plaintext

#### 3.EXPLAIN ABOUT DATA ENCRYPTION STANDARD IN DETAILS.

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

For DES, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output.

#### **DES Encryption**

The overall scheme for DES encryption is illustrated in the Figure 2.8. There are two inputs to the encryption function: the **plaintext** to be encrypted and the **key**. The plaintext must be 64 bits in length and the key is 56 bits in length.

### General Depiction of DES Encryption Algorithm

#### Phase 1

Looking at the left-hand side of the figure 2.8, we can see that the processing of the plaintext proceeds in three phases.

First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*.

#### Phase 2:

This is followed by a phase consisting of 16 rounds of the same function, which involves both permutation and substitution functions.

The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the preoutput.

#### Phase 3:

Finally, the preoutput is passed through a permutation (IP-1) that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

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

#### Operation on key:

Initially, the key is passed through a permutation function. Then, for each of the 16 rounds, a *subkey* (*Ki*) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

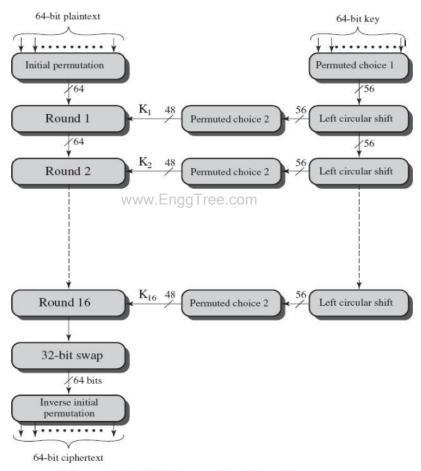


Figure 2.8 DES Encryption Algorithm

#### **Initial Permutation**

The input to a table consists 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 position of a numbered input bit in the output, which also consists of 64 bits.

# ${\bf EnggTree.com}$

				S			
(a) In	itial Per		Control Carron Carron				
	58	50	42	34	26	18	10
	60	52	44	36	28	20	12
	62	54	46	38	30	22	14
	64	56	48	40	32	24	16
	57	49	41	33	25	17	9
	59	51	43	35	27	19	11
	61	53	45	37	29	21	13
	63	55	47	39	31	23	15
Inver	se Initia	al Perm	utation	(IP <sup>-1</sup> )			
	40	8	48	16	56	24	64
	39	7	47	15	55	23	63
	38	6	46	14	54	22	62
	37	5	45	13	53	21	61
	36	4	44	12	52	20	60
	35	3	43	11	51	19	59
	34	2	42	10	50	18	58
	33	1	41	9	49	17	57
Expa	nsion P		1,00	2			
	32	1	2	3		.E <b>5</b> 1ggT	ree.co
	4	5	6	7	8	9	
	8	9	10	11	12	13	
	12	13	14	15	16	17	
	16	17	18	19	20	21	
	20	21	22	23	24	25	
	24	25	26	27	28	29	
	28	29	30	31	32	1	
Perm	utation	Functi	on (P)				
	16	7	20	21	29	12	28
	1	15	23	26	5	18	31
	2	8	24	14	32	27	3
	19	13	30	6	22	11	4
Cons	ider the	followin	g 64-bit	input Λ	<i>1</i> t.		
	<i>M</i> 1	M2	M3	<i>M</i> 4	<i>M</i> 5	<i>M</i> 6	M7
	<i>M</i> 9	<i>M</i> 10	<i>M</i> 11	<i>M</i> 12	<i>M</i> 13	<i>M</i> 14	<i>M</i> 15
	<i>M</i> 17	<i>M</i> 18	<i>M</i> 19	M20	<i>M</i> 21	M22	M23
	<i>M</i> 25	M26	M27	M28	M29	M30	<i>M</i> 31
	M33	M34	M35	M36	M37	<i>M</i> 38	<i>M</i> 39
	<i>M</i> 41	M42	<i>M</i> 43	M44	M45	M46	M47
	M49	M50	<i>M</i> 51	<i>M</i> 52	M53	M54	<i>M</i> 55
	10710	11100	,,,,,	11102	11100	1010-1	17100

where Mi is a binary digit. Then the permutation X = IP(M) is as follows:

M58	M50	<i>M</i> 42	M34	M26	<i>M</i> 18	M10	M2
<i>M</i> 60	<i>M</i> 52	M44	<i>M</i> 36	<i>M</i> 28	<i>M</i> 20	<i>M</i> 12	<i>M</i> 4
<i>M</i> 62	<i>M</i> 54	<i>M</i> 46	<i>M</i> 38	<i>M</i> 30	M22	<i>M</i> 14	<i>M</i> 6
M64	<i>M</i> 56	M48	M40	<i>M</i> 32	M24	<i>M</i> 16	M8
<i>M</i> 57	<i>M</i> 49	<i>M</i> 41	<i>M</i> 33	<i>M</i> 25	<i>M</i> 17	<i>M</i> 9	<i>M</i> 1
<i>M</i> 59	<i>M</i> 51	<i>M</i> 43	<i>M</i> 35	<i>M</i> 27	<i>M</i> 19	<i>M</i> 11	M3
<i>M</i> 61	<i>M</i> 53	<i>M</i> 45	<i>M</i> 37	<i>M</i> 29	<i>M</i> 21	<i>M</i> 13	<i>M</i> 5
M63	M55	M47	M39	<i>M</i> 31	M23	M15	M7

Inverse permutation  $Y = IP^{-1}(X) = IP^{-1}(IP(M))$ , Therefore we can see that the original ordering of the bits is restored.

#### **Details of Single Round**

The below figure 2.9 shows the internal structure of a single round. The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right). The overall processing at each round can be summarized in the following formulas:

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

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

$$= 32 \text{ bits} \longrightarrow 28 \text{ bits}$$

XOR

Figure 2.9 Single Round of DES Algorithm

The round key *Ki* is 48 bits. The *R* input is 32 bits. This *R* input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the *R* bits. The resulting 48 bits are XORed with *Ki*. This 48-bit result passes through a substitution function that produces a 32-bit output, which is then permuted.

#### **Definition of S-Boxes**

The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output. The first and last bits of the input to box Si form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for Si. The middle four bits select one of the sixteen columns as shown in figure 2.10.

The decimal value in the cell selected by the row and column is then converted to its 4-bit representation to produce the output.

For example, in S1 for input 011001, the row is 01 (row 1) and the column is 1100 (column 12). The value in row 1, column 12 is 9, so the output is 1001.

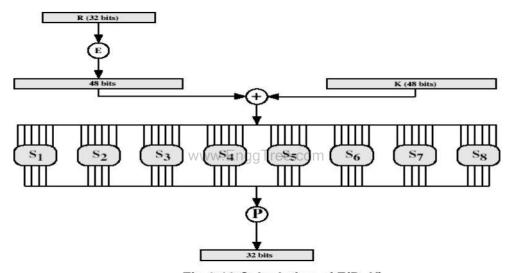


Fig 2.10 Calculation of F(R, K)

#### **Key Generation**

The 64-bit key is used as input to the algorithm. The bits of the key are numbered from 1 through 64; every eighth bit is ignored. The key is first subjected to a permutation governed by a table labeled Permuted Choice One. The resulting 56-bit key is then treated as two 28-bit quantities, labeled C0 and D0.

At each round, *Ci*-1 and *Di*-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 2, which produces a 48-bit output that serves as input to the function F(*Ri*-1, *Ki*).

<b>DES Key</b>	Schedule	Calculation
----------------	----------	-------------

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	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

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

#### (d) Schedule of Left Shifts

Roundnumber: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

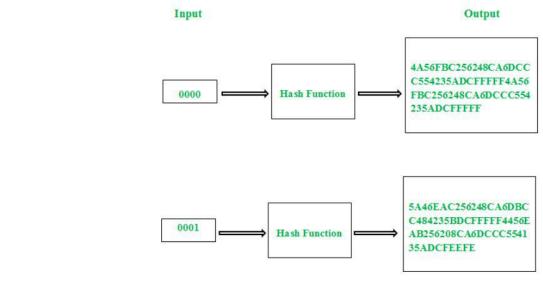
Bits rotated: 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

### **DES Decryption:** www.EnggTree.com

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

#### The Avalanche Effect:

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



## Downloaded from EnggTree.com

#### 4.EXPLAIN THE STRENGTH OF DES IN DETAILS.

The strength of DES depends on two factors: **key size** and the **nature of the algorithm**.

#### 1. The Use of 56-Bit Keys

With a key length of 56 bits, there are  $2^{56}$  possible keys, which is approximately 7.2 x  $10^{16}$ . Thus, a brute-force attack appears impractical.

#### 2. The Nature of the DES Algorithm

In DES algorithm, eight substitution boxes called S-boxes that are used in each iteration. Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes. Despite this, no one has so far succeeded in discovering the supposed fatal weaknesses in the S-boxes.

#### 3. Timing Attacks

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

#### Attacks on DES:

www.EnggTree.com

Two approaches are:

- 1. Differential crypt analysis
- 2. Linear crypt analysis

#### **Differential Cryptanalysis**

Differential cryptanalysis is the first published attack that is capable of breaking DES in less than 255 complexities. The need to strengthen DES against attacks using differential cryptanalysis played a large part in the design of the S-boxes and the permutation P.

- One of the most significant recent (public) advances in cryptanalysis
- Powerful method to analyze block ciphers
- Used to analyze most current block ciphers with varying degrees of success

#### **Differential Cryptanalysis Attack:**

The differential cryptanalysis attack is complex. The rationale behind differential cryptanalysis is to observe the behavior of pairs of text blocks evolving along each round of the cipher, instead of observing the evolution of a single text block.

Consider the original plaintext block m to consist of two halves m0, m1. Each round of DES maps the right-hand input into the left-hand output and sets the right-hand output to be a function of the left-hand input and the subkey for this round.

So, at each round, only one new 32-bit block is created. If we label each new block  $m_1(2 \le i \le 17)$ , then the intermediate message halves are related as follows:

$$m_{i+1} = m_{i+1} \bigoplus f(m_i, K_i), i = 1, 2, ..., 16$$

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

$$\begin{array}{ll} \Delta m_{i+1} &= m_{i+1} \bigoplus m^{\kappa}_{i-1} \\ &= [m_{i-1} \bigoplus f(m_i, k_i])] \bigoplus [m^{\kappa}_{i-1} \bigoplus f(m^{\kappa}_i, k_i)] \\ &= \Delta m_{i-1} \bigoplus [f(m_i, k_i) \bigoplus f(m^{\kappa}_i, k_i)] \end{array}$$

Let us suppose that there are many pairs of inputs to f with the same difference yield the same output difference if the same subkey is used. ee.com

Therefore, if we know  $\Delta m_{i-1}$  and  $\Delta m_i$  with high probability, then we know  $\Delta m_{i+1}$  with high probability. Furthermore, if a number of such differences are determined, it is feasible to determine the subkey used in the function f.

#### Linear Cryptanalysis

This attack is based on the fact that linear equation can be framed to describe the transformations.

The principle of linear crypt analysis is as follows

```
Length of CT and PT =n bits;

key=mbit

Block of cipher text is c[1]c[2]...c[n]; Block

of key is k[1]k[2]...k[m]

A[I,i,..k] = A[i] ⊕A[i] ⊕. ⊕A[k]
```

- Can attack DES with 247 known plaintexts, still in practice infeasible
- ➤ Find linear approximations with prob p!= ½
- $P[i_1,i_2,...,i_a](+)c[j_1,j_2,...,j_b] = k[k_1,k_2,...,k_c]$  Where  $i_a, j_b, k_c$  are bit locations in p, c, k

#### 5.DESCRIBE IN DETAIL ABOUT THE BLOCK CIPHER PRINCIPLES

There are three critical aspects of block cipher design:

- 1. Number of rounds,
- 2. Design of the function F
- 3. Key scheduling.

#### **Number of Rounds**

- When the greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak F.
- The number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack
- When round DES S= 16, a differential cryptanalysis attack is slightly less efficient than brute force, the differential cryptanalysis attack requires 2<sup>55</sup> operations.
- It makes it easy to judge the strength of an algorithm and to compare different algorithms.

#### Design of Function F

This is the most important function

#### Criteria needed for F,

- It must be difficult to "unscramble" the substitution performed by F.
- The function should satisfy strict avalanche criterion (SAC) which states that any
  output bit j of an S-box should change with probability 1/2 when any single input bit i is
  inverted for all i, j.
- The function should satisfy bit independence criterion(BIC), which states that output bits j and k should change independently when any single input bit i is inverted for all i, j, and k.

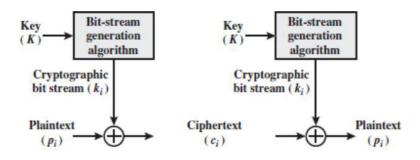
#### **Key Schedule Algorithm**

- The key is used to generate one sub key for each round.
- The sub keys to maximize the difficulty of deducing individual sub keys and the difficulty
  of working back to the main key.

#### Stream Cipher and Block Cipher

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. E.g, vigenere cipher. Figure (2.11a)

A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically, a block size of 64 or 128 bits is used. Figure (2.11b)



(a) Stream cipher using algorithmic bit-stream generator

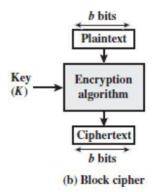
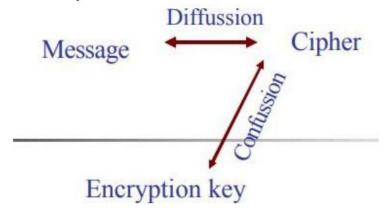


Figure 2.11 Stream Cipher and Block Cipher

- Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing.
- In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves.
- The original key is expanded so that a different key is used for each round.
- ➤ The Data Encryption Standard (DES) has been the most widely used encryption algorithm. It exhibits the classic Feistel structure.
- The DES uses a 64-bit block and a 56-bit key. Two important methods of cryptanalysis are differential cryptanalysis and linear cryptanalysis. DES has been shown to be highly resistant to these two types of attack.
- A block cipher operates on a plaintext block of n bits to produce a ciphertext block of n bits. There are possible different plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be possible), each must produce a unique ciphertext block. Such a transformation is called reversible, or non singular
- In particular, Feistel proposed the use of a cipher that alternates substitutions and permutations, where these terms are defined as follows:
  - Substitution: Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.
  - Permutation: A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

- Two methods for frustrating statistical cryptanalysis are:
  - Diffusion Each plaintext digit affects many ciphertext digits, or each ciphertext digit is affected by many plaintext digits.
  - Confusion Make the statistical relationship between a plaintext and the corresponding ciphertext as complex as possible in order to thread attempts to deduce the key.



### Feistel cipher structure

- ➤ The left-hand side of figure 2.12 depicts the structure proposed by Feistel.
- ➤ The input to the encryption algorithm is a plaintext block of length 2w bits and a key K. the plaintext block is divided into two halves L₀ and R₀.
- ➤ The two halves of the data pass through in rounds of processing and then combine to produce the ciphertext block. Each round i has inputs L<sub>i-1</sub> and R<sub>i-1</sub>, derived from the previous round, as well as the subkey K<sub>i</sub>, derived from the overall key K.
- In general, the subkeys K<sub>i</sub> are different from K and from each other. All rounds have the same structure.
- ➤ A substitution is performed on the left half of the data (as similar to S-DES). This is done by applying a round function F to the right half of the data and then taking the XOR of the output of that function and the left half of the data.
- ➤ The round function has the same general structure for each round but is parameterized by the round subkey k<sub>i</sub>. Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.
- This structure is a particular form of the substitution-permutation network.

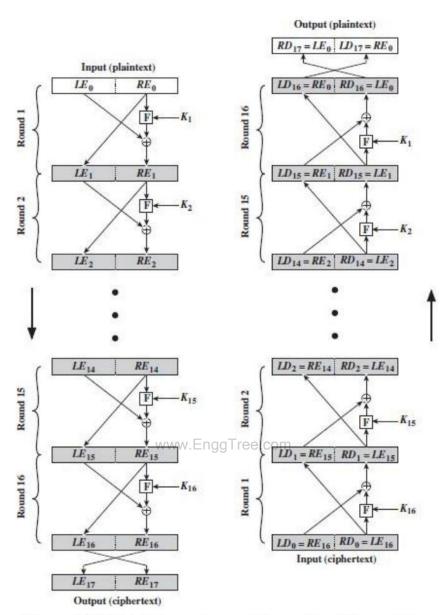


Figure 2.12 Feistel Encryption and Decryption (16 rounds)

The features of Feistel network are:

- Block size Increasing size improves security, but slows cipher
- Key size Increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- Number of rounds Increasing number improves security, but slows cipher
- Subkey generation Greater complexity can make analysis harder, but slows cipher
- Round function Greater complexity can make analysis harder, but slows cipher
- The process of decryption is essentially the same as the encryption process.
- ➤ The rule is as follows: use the cipher text as input to the algorithm, but use the subkey k<sub>i</sub> in reverse order. i.e., k<sub>n</sub> in the first round, k<sub>n-1</sub> in second round and so on.
- For clarity, we use the notation LE<sub>i</sub> and RE<sub>i</sub> for data traveling through the decryption algorithm and LD<sub>i</sub> and RD<sub>i</sub>.
- The above diagram indicates that, at each round, the intermediate value of the decryption process is same (equal) to the corresponding value of the encryption process with two halves of the value swapped.

i.e., RE<sub>i</sub> || LE<sub>i</sub> (or) equivalently RD<sub>16-i</sub> || LD<sub>16-i</sub>

- ➤ After the last iteration of the encryption process, the two halves of the output are swapped, so that the cipher text is RE<sub>16</sub> || LE<sub>16</sub>.
- The output of that round is the cipher text. Now take the cipher text and use it as input to the same algorithm.
- ➤ The input to the first round is RE<sub>16</sub> || LE<sub>16</sub>, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process.
- Now we will see how the output of the first round of the decryption process is equal to a 32-bit swap of the input to the sixteenth round of the encryption process.
- First consider the encryption process,

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

On the decryption side,

$$\begin{aligned} LD_1 &= RD_0 = LE_{16} = RE_{15} \\ RD_1 &= LD_0 \oplus F (RD_0, K_{16}) \\ &= RE_{16} \oplus F (RE_{15}, K_{16}) \\ &= [LE_{15} \oplus F (RE_{15}, K_{16})] \oplus F (RE_{15}, K_{16}) \\ &= LE_{15} \end{aligned}$$
Therefore,  $LD_1 = RE_{15}$ ,  $RD_1 = LE_{15}$ 

In general, for the ith iteration of the encryption algorithm,

$$RE_{i/\#/L}E_{i-1} \oplus F(RE_{i-1}, K_i)$$

➤ Finally, the output of the last round of the decryption process is RE<sub>0</sub> || LE<sub>0</sub>. A 32-bit swap recovers the original plaintext.

#### 7.EXPLAIN THE BLOCK CIPHER MODES OF OPERATION

- Block Cipher is the basic building block to provide data security.
- To apply the block cipher to various applications, NIST has proposed 4 modes of operation. The block cipher is used to enhance the security of the encryption algorithm

### **Multiple Encryption and Triple DES**

The vulnerability of DES to a brute-force attack has been detected by using two approaches are shown in figure 2.13

- 1. One approach is to design a completely new algorithm, of which AES is a prime example
- 2. Another alternative, which would preserve the existing investment in software and equipment, is to use multiple encryptions with DES and multiple keys.

#### **Double DES**

The simplest form of multiple encryptions has two encryption stages and two keys. Given a plaintext P and two encryption keys  $K_1$  and  $K_2$ , cipher text C is generated as

$$C = \mathrm{E}(K_2, \mathrm{E}(K_1, P))$$

Decryption requires that the keys be applied in reverse order:

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

For DES, this scheme apparently involves a key length of 56 \* 2 = 112 bits, resulting in a dramatic increase in cryptographic strength.

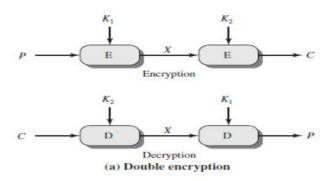


Figure 2.13 Multiple Encryption

#### Reduction to a Single Stage

Suppose it were true for DES, for all 56-bit key values, that given any two keys  $K_1$  and  $K_2$ , it would be possible to find a key  $K_3$  such that

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

#### Meet-in-the-Middle Attack

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

It is based on the observation that, if we have

$$C = \mathrm{E}(K_2, \mathrm{E}(K_1, P))$$

Then

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

Given a known pair, (P, C), the attack proceeds as follows. First, encrypt P for all 256 possible values of K<sub>1</sub>. Store these results in a table and then sort the table by the Values of X.

Next, decrypt C using all 256 possible values of K2. As each decryption is produced,

check the result against the table for a match.

If a match occurs, then test the two resulting keys against a new known plaintext-cipher text pair. If the two keys produce the correct cipher text, accept them as the correct keys.

For any given plaintext P, there are 264 possible cipher text values that could be produced by double DES. Double DES uses, in effect, a 112-bit key, so that there are 2112 possible keys.

#### **Triple DES with Two Keys**

To overcome the meet-in-the-middle attack is to use three stages of encryption with three different keys. This is called ad Triple DES or 3DES as shown in figure 2.14.

The known plain text attack in  $2^{112}$ . The key length of 56 \* 3 = 168 bits which is a drawback.

Tuchman proposed a triple encryption method that uses only two keys given plain text  $k_1,k_2$ . The final cipher text is

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

The function follows an encrypt-decrypt-encrypt (EDE)sequence

Its only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES:

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

- 3DES with two keys is a relatively popular alternative to DES
- · There are no practical cryptanalytic attacks on 3DES.
- The cost of a brute-force key search on 3DES is on the order of2<sup>112</sup>

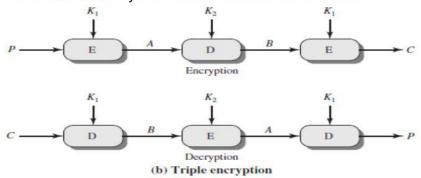


Figure 2.14 Triple DES

The first serious proposal came from Merkle and Hellman

#### 1. Merkle and Hellman

The concept is to find plaintext values that produce a first intermediate value of A = 0 and then using the meet-in-the-middle attack to determine the two keys.

- The level of effort is 2<sup>56</sup>,
- The technique requires 256 chosen plaintext-cipher text pairs, which is a number unlikely to be provided.

#### 2. known - plaintext attack:

The attack is based on the observation that if we know A and C then the problem reduces to that of an attack on double DES.

The attacker does not know A, even if P and C are known, as long as the two keys are unknown. The attacker can choose a potential value of A and then try to find a known (P, C) pair that produces A.

The attack proceeds as follows.

### Step 1:

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

#### Step 2:

- Pick an arbitrary value a for A, and create a second table with entries defined in the following fashion.
- For each of the  $2^{56}$  possible keys  $K_1 = i$ , calculate the plaintext value Pi that produces a.
- For each P<sub>i</sub> that matches an entry in Table 1, create an entry in Table 2 consisting of the K<sub>1</sub> value and the value of B that is produced.

# Downloaded from: annauniversityedu.blogspot.com Downloaded from EnggTree.com

### Step 3:

- We now have a number of candidate values of K<sub>1</sub> in Table 2 and are in a position to search for a value of K<sub>2</sub>.
- For each of the 256 possible keys K<sub>2</sub> = j, calculate the second intermediate value for our chosen value of a
- If there is a match, then the corresponding key i from Table 2 plus this value of j are candidate values for the unknown keys (K1, K2).

### Step 4:

- Test each candidate pair of keys (i, j) on a few other plaintext-cipher text pairs.
- If a pair of keys produces the desired cipher text, the task is complete. Ifno pair succeeds, repeat from step 1 with a new value of a.

#### MODE 1: Electronic Code Book

The simplest mode is the electronic codebook (ECB) mode shown in figure 2.15. Here plaintext is handled one block at a time and each block of plaintext is encrypted using the same key.

The term codebook is used because, for a given key, there is a unique cipher text for every b-bit block of plaintext.

When the message longer than b bits, to break the message into b-bit blocks. For the last block when the no of bits is less than b, padding the last block if necessary.

Decryption is performed one block at a time, always using the same key.

Uses: The ECB method is ideal for a short amount of data, such as an encryption key.

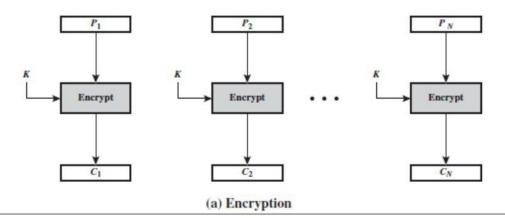
### Disadvantage:

When b" -bit block of plaintext appears more than once in the message, it always produces the same cipher text output.

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

If the message has repetitive elements with a period of repetition a multiple of b bits, then these elements can be identified by the analyst.

This may help in the analysis or may provide an opportunity for substituting or rearranging blocks.



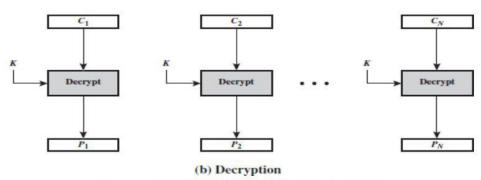


Figure 2.15 Electronic Code Book (ECB)

### Mode Properties for Evaluating and Constructing ECB

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

**Error recovery:** The property that an error in the *i*th cipher text block is inherited by only a few plaintext blocks

**Error propagation:** It is meant here is a bit error that occurs in the transmission of a cipher text block, not a computational error in the encryption of a plaintext block. **Diffusion:** Low entropy plaintext blocks should not be reflected in the cipher text blocks. Roughly, low entropy equates to predictability or lack of randomness

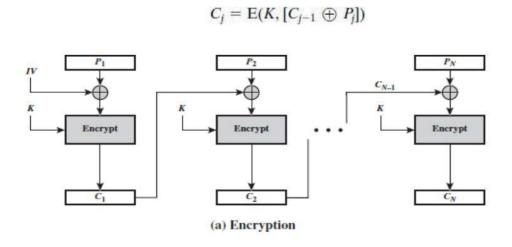
Security: Whether or not the cipher text blocks leak information about the plaintext blocks.

### MODE 2: Cipher Block Chaining Mode ngg Tree.com

This method is to overcome the disadvantage of ECB (i.e) when the PT block is repeated CBC produces different cipher text blocks

The input to the encryption function for each plaintext block bears no fixed relationship to the plaintext block. Therefore, repeating patterns of b bits are not exposed.

For decryption, each cipher block is passed through the decryption algorithm. The result is XORed with the preceding cipher text block to produce the plaintext block are shown in figure 2.16.



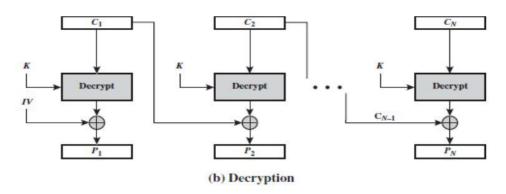


Figure 2.16 Cipher Block Chaining (CBC) Mode

Then

$$D(K, C_i) = D(K, E(K, [C_{i-1} \oplus P_i]))$$

To produce the first block of cipher text, an initialization vector (IV) is XORed with the first block of plaintext.

On decryption, the IV is XORed with the output of the decryption algorithm to recover the first block of plaintext.

Size of IV = Size of data Blocks We can define CBC mode as

www.EnggTree.com

CBC 
$$C_1 = E(K, [P_1 \oplus IV]) \qquad P_1 = D(K, C_1) \oplus IV$$
$$C_j = E(K, [P_j \oplus C_{j-1}]) j = 2, \dots, N \qquad P_j = D(K, C_j) \oplus C_{j-1} j = 2, \dots, N$$

For maximum security, the IV should be protected against unauthorized changes. This could be done by sending the IV using ECB encryption

#### Reason for protecting the IV:

If an opponent is able to fool the receiver in to using a different value for IV, then the opponent is able to invert selected bits in the first block of plaintext. To see this, consider

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

Now use the notation that X[i] denotes the ith bit of the b-bit quantity X. Then

$$P_1[i] = IV[i] \oplus D(K, C_1)[i]$$

Then, using the properties of XOR, we can state

$$P_1[i]' = IV[i]' \oplus D(K, C_1)[i]$$

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

#### MODE 3: Cipher Feedback Mode:

We know that the DES is a block cipher it is possible to convert block cipher into stream Cipher using CFB mode

The advantages of CFB is that

- Eliminates the need to pad a message
- It also can operate in real time
- The length of the CT =Length of PT

Figure 2.17 depicts the CFB scheme. In the figure 2.17, it is assumed that the unit of transmission is s bits; a common value is s = 8.

The units of plaintext are chained together; to get the cipher text is a function of all preceding plaintext. Here the plaintext is divided into segments of s bits.

### Encryption:

The input to the encryption function is a b-bit shift register that is initially set to some initialization vector (IV).

The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P1 to produce the first unit of cipher text C1.

The contents of the shift register are shifted left by s bits, and C1 is placed in the rightmost (least significant) s bits of the shift register.

This process continues until all plaintext units have been encrypted.

#### Decryption:

The same scheme is used, except that the received cipher text unit is XORed with the output of the encryption function to produce the plaintext unit.

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

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

Therefore, by rearranging terms:

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

The same reasoning holds for subsequent steps in the process.

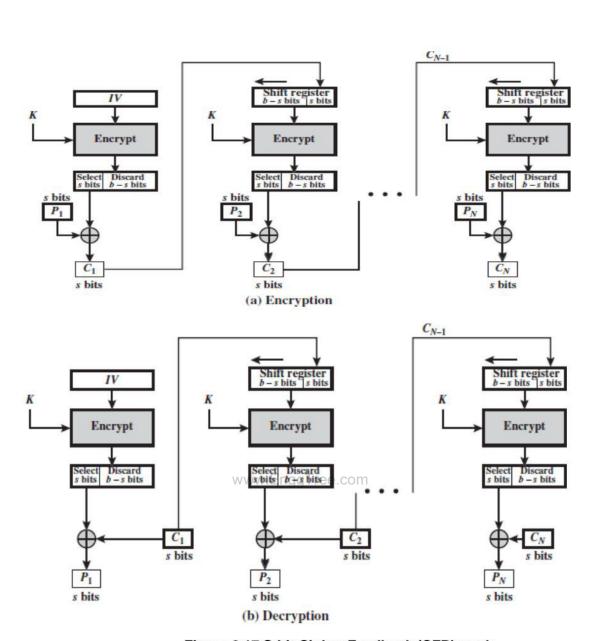


Figure 2.17 S-bit Cipher Feedback (CFB) mode

We can define CFB mode as follows

CFB 
$$\begin{cases} I_{1} = IV & I_{1} = IV \\ I_{j} = LSB_{b-s}(I_{j-1}) \| C_{j-1} & j = 2, \dots, N \\ O_{j} = E(K, I_{j}) & j = 1, \dots, N \\ C_{j} = P_{j} \oplus MSB_{s}(O_{j}) & j = 1, \dots, N \end{cases}$$
 
$$\begin{cases} I_{1} = IV \\ I_{j} = LSB_{b-s}(I_{j-1}) \| C_{j-1} & j = 2, \dots, N \\ O_{j} = E(K, I_{j}) & j = 1, \dots, N \\ P_{j} = C_{j} \oplus MSB_{s}(O_{j}) & j = 1, \dots, N \end{cases}$$

### **Output Feedback Mode**

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

The output of the encryption function is fed back to become the input for encrypting the next block of plaintext as shown in figure 2.18.

### Comparison between OFB and CFB

In CFB, the output of the XOR unit is fed back to become input for encrypting the next block.

The other difference is that the OFB mode operates on full blocks of plaintext and cipher text, whereas CFB operates on an s-bit subset. OFB encryption can be expressed as

Where

$$C_j = P_j \oplus E(K, O_{j-1})$$
$$O_{j-1} = E(K, O_{j-2})$$

we can rewrite the encryption expression as:

$$C_i = P_i \oplus E(K, [C_{i-1} \oplus P_{i-1}])$$

By rearranging terms, we can demonstrate that decryption works.

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

We can define OFB mode as follows.

$$I_{1} = \textit{Nonce}$$

$$I_{j} = O_{j-1} \qquad j = \forall 2 \forall \forall i \in \mathcal{I}_{j} \text{ for each} j = O_{j-1} \qquad j = 2, \dots, N$$

$$O_{j} = E(K, I_{j}) \qquad j = 1, \dots, N \qquad O_{j} = E(K, I_{j}) \qquad j = 1, \dots, N$$

$$C_{j} = P_{j} \oplus O_{j} \qquad j = 1, \dots, N-1$$

$$C_{N}^{*} = P_{N}^{*} \oplus \text{MSB}_{u}(O_{N}) \qquad P_{N}^{*} = C_{N}^{*} \oplus \text{MSB}_{u}(O_{N})$$

$$I_{1} = \textit{Nonce}$$

$$O_{j} = E(K, I_{j}) \qquad j = 1, \dots, N$$

$$P_{j} = C_{j} \oplus O_{j} \qquad j = 1, \dots, N-1$$

$$P_{N}^{*} = C_{N}^{*} \oplus \text{MSB}_{u}(O_{N})$$

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

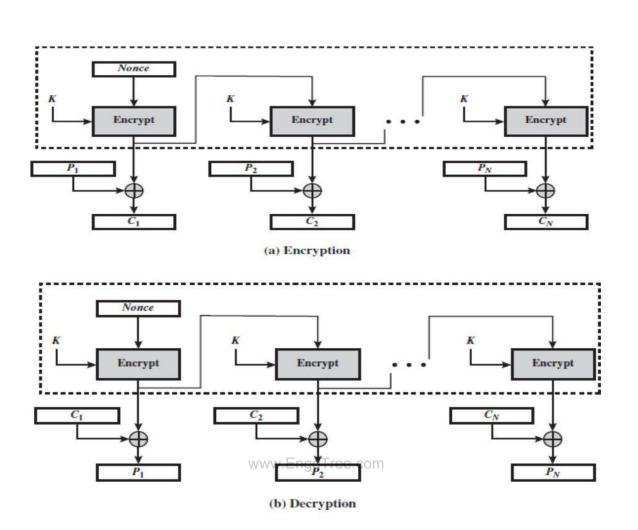


Figure 2.18 Output Feedback Mode

### Advantage:

Bit errors in transmission do not propagate (i.e.) when bit errors occurs in Ci, Pi is alone affected

### Disadvantage:

Vulnerable to message stream modification attack

#### **Counter Mode**

The counter (CTR) mode has increased recently with applications to ATM (asynchronous transfer mode) network security and IP sec (IP security).

A counter equal to the plaintext block size is used. The counter value must be different for each plaintext block as shown in figure 2.19.

The counter is initialized to some value and then incremented by 1 for each subsequent block (modulo 2b, where b is the block size). For encryption, the counter is encrypted and then XORed with the plaintext block to produce the cipher text block.

For decryption, the same sequence of counter values is used, with each encrypted counter XORed with a cipher text block to recover the corresponding plaintext block.

### Advantage:

Hardware efficiency

· CTR can be done in parallel

Software efficiency

· CTR supports parallel feature pipelining

Preprocessing

Simplicity

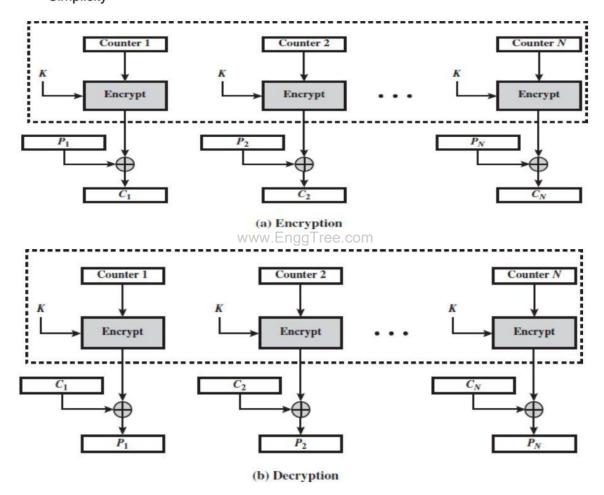


Figure 2.19 Counter Mode

### 8.DESCRIBE ABOUT THE ADVANCED ENCRYPTION STANDARD (AES).

AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications. Compared to public-key ciphers such as RSA, the structure of AES and most symmetric ciphers is quite complex and cannot be explained as easily as many other cryptographic, algorithms.

#### **Finite Field Arithmetic**

In AES, all operations are performed on 8-bit bytes. The arithmetic operations of addition, multiplication, and division are performed over the finite field GF.A field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set. Division is defined with the following rule: a/b = a(b-1).

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

The way of defining a finite field containing  $2^n$  elements; such a field is referred to as  $GF(2^n)$ . Consider the set, S, of all polynomials of degree n - 1 or less with binary coefficients. Thus, each polynomial has the form

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 = \sum_{i=0}^{n-1} a_ix^i$$

Where each a takes on the value 0 or 1. There are a total of 2<sup>n</sup> different polynomials in S. 

Appropriate definition of arithmetic operations, each such set S is a finite field.

#### The definition consists of the following elements.

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

A polynomial in GF(2n) can be uniquely represented by its n binary coefficients(an-1an-2 ca0). Therefore, every polynomial in GF(2n) can be represented by an n-bit number.

#### **AES Structure**

#### **General Structure**

 Figure 2.20 shows the overall structure of the AES encryption process. The cipher takes a plaintext block size of 128 bits, or 16 bytes. The key length can be 16, 24, or32 bytes (128, 192, or 256 bits). The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.

- The input to the encryption and decryption algorithms is a single 128-bit block. The block is depicted as a 4 \* 4 square matrix of bytes. This block is copied into the **State** array, which is modified at each stage of encryption or decryption. After the final stage, **State** is copied to an output matrix. These operations are depicted in Figure 2.21a. Similarly, the key is depicted as a square matrix of bytes. This key is then expanded into an array of key schedule words.
- Below Figure 2.20 shows the expansion for the 128-bit key. Each word is four bytes, and the total key schedule is 44 words for the 128-bit key. Note that the ordering of bytes within a matrix is by column. The first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the in matrix. The second four bytes occupy the second column, and so on. Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the w matrix. The cipher consists of N rounds, where the number of rounds depends on the key length: 10 rounds for a 16-byte key, 12 rounds for a 24-byte key, and 14 rounds for a 32-byte key (Table 2.3).
- The first *N* 1 round consist of four distinct transformation functions: Sub Bytes, Shift Rows, Mix Columns, and AddRoundKey, which are described subsequently. The final round contains only three transformations, and there is an initial single transformation (AddRoundKey) before the first vround which can be considered Round 0. Each transformation takes one or more 4 \* 4 matrices as input and produces a 4 \* 4 matrix as output Figure 5.1 shows that the output of each round is a 4 \* 4 matrix, with the output of the final round being the cipher text.

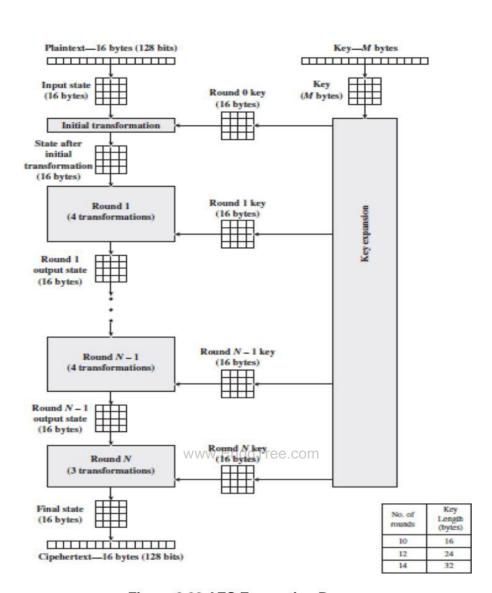


Figure 2.20 AES Encryption Process

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

**Table 2.3 AES Parameters** 

#### **Detailed Structure**

Below Figure 2.20 shows the AES cipher shows the sequence of transformations in each round and showing the corresponding decryption function.

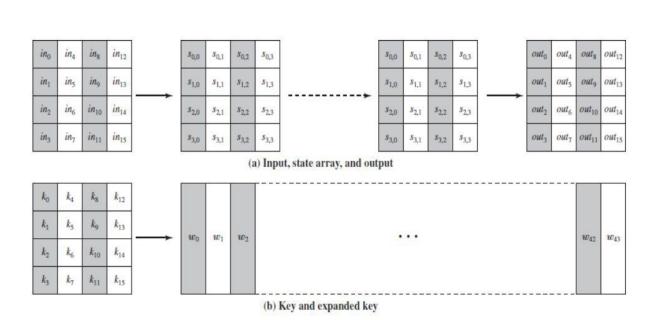


Fig: 2.21 Detail AES structure

#### Overall detail about AES structure.

- 1. It is not a Feistel structure. Recall that, in the classic Feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. AES instead processes the entire data block as a single matrix during each round using substitutions and permutation.
- The key that is provided as input is expanded into an array of forty-four 32-bitwords, w[i].
   Four distinct words (128 bits) serve as a round key for each round as shown in figure 2.22;
- 3. Four different stages are used, one of permutation and three of substitution:
  - Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block
  - ShiftRows: A simple permutation
  - MixColumns: A substitution that makes use of arithmetic over GF(28)
  - AddRoundKey: A simple bitwise XOR of the current block with a portion of the expanded key
- 4. The structure is quite simple. For both encryption and decryption as shown in figure 2.22, 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 makes use of the key. The AddRoundKey stage would provide no security because they do not use the key. We can view the cipher as alternating operations of XOR encryption (AddRoundKey) of a block, followed by scrambling of the block (the other three stages), followed by XOR encryption, and so on. This scheme is both efficient and highly secure.

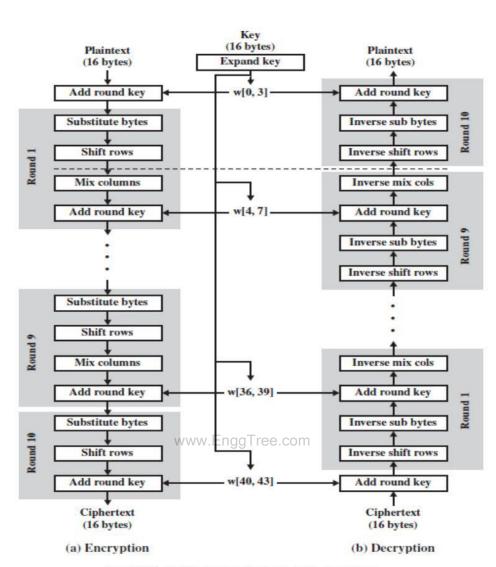


Fig 2.22 AES Encryption and Decryption

6. Each stage is easily reversible. For the Substitute Byte, ShiftRows, and MixColumns stages, an inverse function is used in the decryption algorithm. For the AddRoundKey stage, the inverse is achieved by XORing the same round key to the block, using the result that.

$$A \oplus B \oplus B = A$$

7. The decryption algorithm makes use of the expanded key in reverse order. However, the decryption algorithm is not identical to the encryption algorithm. This is a consequence of the particular structure of AES.

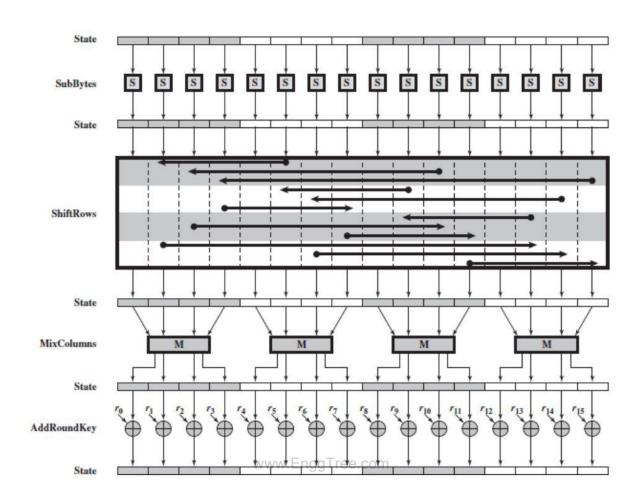


Fig 2.23 AES Encryption Round

- **8.** Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext.
- 9. The final round of both encryption and decryption consists of only three stages. Again, this is a consequence of the particular structure of AES and is required, to make the cipher reversible

#### **AES Transformation Functions**

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

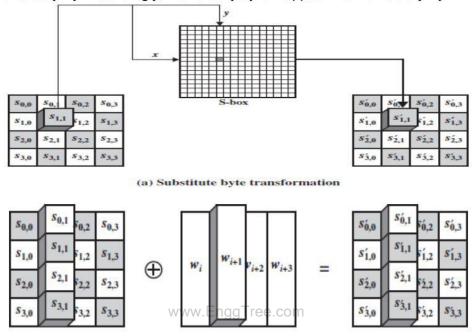
#### **Substitute Bytes Transformation**

### Type 1: Forward and Inverse Transformations:

**The forward substitute byte transformation**, called Sub Bytes, is a simple table lookup (Figure 2.24a). AES defines a 16 \* 16 matrix of byte values, called an S-box that contains a permutation of all possible 256 8-bit values.

Each individual byte of **State** is mapped into a new byte in the following way: The leftmost 4 bits of the byte are used as a row value and the rightmost 4 bits are used as a column value. These row and column values serve as indexes into the S-box to select a unique8-bit output value as shown in figure 2.25.

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



(b) Add round key transformation

Figure 2.24 AES Byte level Operations

										v							
		0	1	2	3	4	5	6	7	8	9	A	В	C	D	E	F
	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	В3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
x	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	В	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

(a) S-box

									3	0							
		0	1	2	3	4	5	6	7	8	9	A	В	C	D	E	F
	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	<b>D</b> 9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	В3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
x	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	В	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

(b) Inverse S-box

Figure 2.25 AES S-Boxes

Here is an example of the SubBytes transformation:

EA	04	65	<sub>\\\</sub> 85 <sub>\\\</sub>	EnggTree.	c.87h	F2	4D	97
83	45	5D	96		EC	6E	4C	90
5C	33	98	<b>B</b> 0	-	4A	C3	46	E7
F0	2D	AD	C5		8C	D8	95	A6

The S-box is constructed in the following fashion (Figure 2.26a).

- **1.** Initialize the S-box with the byte values in ascending sequence row by row. The first row contains  $\{00\}$ ,  $\{01\}$ ,  $\{02\}$ , c,  $\{0F\}$ ; the second row contains  $\{10\}$ ,  $\{11\}$ , etc.; and so on. Thus, the value of the byte at row y, column x is  $\{yx\}$ .
- 2. Map each byte in the S-box to its multiplicative inverse in the finite field GF(28); the value {00} is mapped to itself.
- **3.** Consider that each byte in the S-box consists of 8 bits labeled (*b*7, *b*6, *b*5, *b*4, *b*3,*b*2, *b*1, *b*0). Apply the following transformation to each bit of each byte in the S-box:

$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

Where ci is the ith bit of byte c with the value {63}; that is,  $(c_7c_6c_5c_4c_3c_2c_1c_0) = (01100011)$ . The prime ( ") indicates that the variable is to be updated by the value on the right.

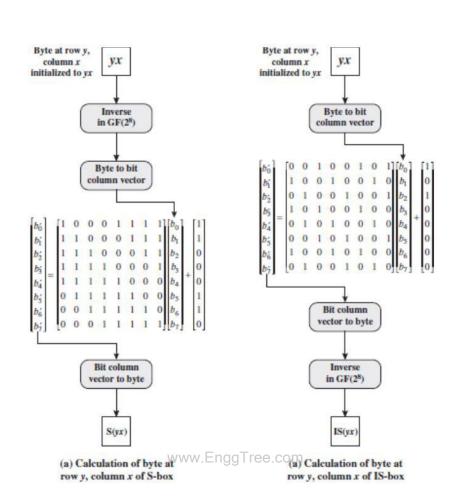


Figure 2.26 Construction of S-Box and IS-Box

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

The AES standard depicts this transformation in matrix form as follows.

- In ordinary matrix multiplication, each element in the product matrix is the sum of products of the elements of one row and one column. Each element in the product matrix is the bitwise XOR of products of elements of one row and one column.
- As an example, consider the input value {95}. The multiplicative inverse in GF(28) is {95}<sup>-1</sup> = {8A}, which is 10001010 in binary. Using above Equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The result is {2A}, which should appear in row {09} column {05} of the S-box.

### Type 2: Inverse Substitute Byte Transformation:

The **inverse substitute byte transformation**, called InvSubBytes, For example, that the input {2A}produces the output {95}, and the input {95} to the S-box produces {2A}. The inverse S-box is constructed by applying the inverse of the transformation is followed by taking the

$$b'_i = b_{(i+2) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus d_i$$

multiplicative inverse in GF(28). The inverse transformation is

where byte  $d = \{05\}$ , or 00000101. We can depict this transformation as follows.

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

InvSubBytes is the inverse of Sub Bytes, label the matrices in sub Bytes and InvSubBytes as **X** and **Y**, respectively, and the vector versions of constants c and d as **C** and **D**, respectively. For some 8-bit vector **B**, becomes  $B' = XB \oplus C$ . We need to show that  $Y(XB \oplus C) \oplus D = B$ . To multiply out, we must show  $YXB \oplus YC \oplus D = B$ . This becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}$$

We have demonstrated that **YX** equals the identity matrix, and the **YC** = **D**,so that **YC**  $\oplus$  **D** equals the null vector.

### Type 3: Shift Rows Transformation

#### Forward and Inverse Shift RowsTransformations:

The forward shift row transformation, called Shift Rows, is depicted in Figure 2.27. The first row of **State** is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-bytecircular left shift is performed. For the fourth row, a 3-byte circular left shift is performed. The following is an example of Shift Rows

87	F2	4D	97	87	7	F2	4D	97
EC	6E	4C	90	6H	3	4C	90	EC
4A	C3	46	E7~	/.Eng <del>g</del> Tree.co <b>46</b>	5	E7	4A	C3
8C	D8	95	A6	A	6	8C	D8	95

Figure 2.27 Forward Shift Row Transformation

The inverse shift row transformation, called InvShiftRows, performs the circular shifts in the opposite direction for each of the last three rows, with a 1-byte circular right shift for the second row, and as shown in figure 2.28

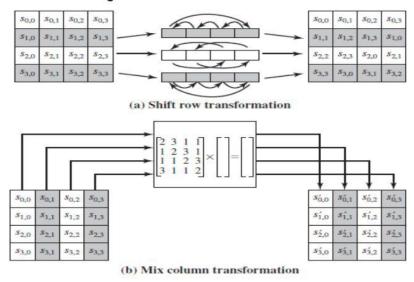


Figure 2.28 AES Row and Column Operations

#### Type 4: Mix Columns Transformation

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

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

Each element in the product matrix is the sum of products of elements of one rowand one column. In this case, the individual additions and multiplications are performed in GF(28).

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$

$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j} \oplus s_{3,j})$$

$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$

$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

The MixColumns transformation on a single column of State can be expressed as

The following is an example of MixColumns:

www.EnggTree.com

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



47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

The MixColumns transformation on the first column, we need to show that

For the first equation, we have  $\{02\}.\{87\} = (0000\ 1110) \oplus (0001\ 1011) = (0001\ 0101)$  and

{03}. {6E} = {6E} 
$$\oplus$$
 ({02}. {6E}) = (0110 1110)  $\oplus$  (1101 1100) = (1011 0010) then  
{02} \cdot {87} = 0001 0101  
{03} \cdot {6E} = 1011 0010  
{46} = 0100 0110  
{A6} =  $\frac{1010 0110}{0100 0111} = {47}$ 

The **inverse mix column transformation**, called InvMixColumns, is defined by the following matrix multiplication:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

The **inverse** of Equation need to show

$$\begin{bmatrix} 0\text{E} & 0\text{B} & 0\text{D} & 09 \\ 09 & 0\text{E} & 0\text{B} & 0\text{D} \\ 0\text{D} & 09 & 0\text{E} & 0\text{B} \\ 0\text{B} & 0\text{D} & 09 & 0\text{E} \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

That is, the inverse transformation matrix times the forward transformation matrix equals the identity matrix. To verify the first column of above Equation.

For the first equation, we have  $\{0E\}.\{02\} = 00011100$  and  $\{09\}.\{03\} = \{09\} \oplus \{09\}.\{02\} = 00001001$   $\oplus$  00010010 = 00011011then

$$\{0E\} \cdot \{02\} = 00011100$$
  
 $\{0B\} = 00001011$   
 $\{0D\}_{\text{WW.Engg}} = 00001101$   
 $\{09\} \cdot \{03\} = 00011011$   
 $00000001$ 

The encryption was deemed more important than decryption for two reasons:

- 1. For the CFB and OFB cipher modes only encryption is used.
- 2. AES can be used to construct a message authentication code and for this, only encryption is used.

### Type 5: AddRoundKey Transformation

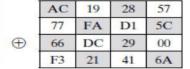
### **Forward and Inverse Transformations**

In the **forward add round key transformation**, called AddRoundKey, the 128 bits of **State** are bitwise XORed with the 128bits of the round key.

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

The following is an example ofAddRoundKey:

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC



EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

The first matrix is **State**, and the second matrix is the round key.

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

The Figure 2.29 is another view of a single round of AES, emphasizing the mechanisms and inputs of each transformation.

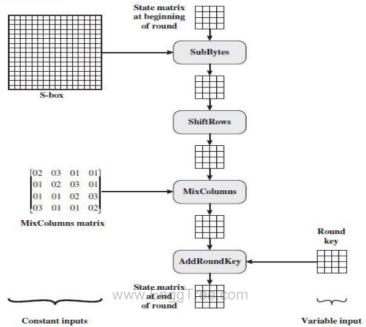


Fig 2.29 AES Key Expansion

#### Type 6: Key Expansion Algorithm

The AES key expansion algorithm takes as input a four-word (16-byte) key and produces a linear array of 44 words (176 bytes). This is sufficient to provide a four word round key for the initial AddRoundKey stage and each of the 10 rounds of the cipher.

Each added word  $\mathbf{w}[i]$ depends on the immediately preceding word,  $\mathbf{w}[i-1]$ , and the word four positions back,  $\mathbf{w}[i-4]$ . In three out of four cases, a simple XOR is used. For a word whose position in the  $\mathbf{w}$  array is a multiple of 4, a more complex function is used.

Figure 2.30 illustrates the generation of the expanded key, using the symbol g to represent that complex function. The function g consists of the following sub functions

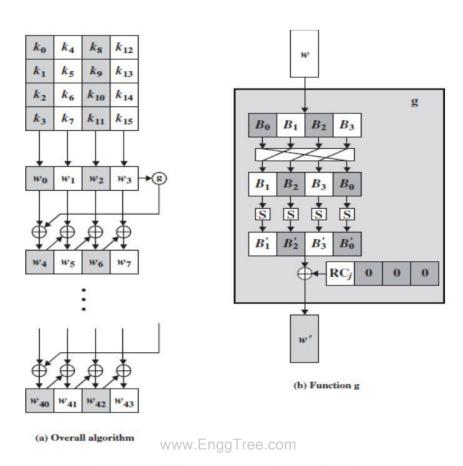


Figure 2.30 Key Expansion Algorithm

- 1. RotWord performs a one-byte circular left shift on a word. This means that a input word [B0, B1, B2, B3] is transformed into [B1, B2, B3, B0].
- 2. SubWord performs a byte substitution on each byte of its input word, using the S-box.
- 3. The result of steps 1 and 2 is XORed with a round constant, Rcon[j].

The round constant is a word in which the three rightmost bytes are always 0. Thus, the effect of an XOR of a word with Rcon is to only perform an XOR on the leftmost byte of the word. The round constant is different for each round and is defined as Rcon[j] = (RC[j], 0, 0, 0), with RC[1] = 1, RC[j] = 2 # RC[j-1] and with multiplication defined over the field GF(28). The values of RC[j] in hexadecimal are

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

For example, suppose that the round key for round 8 is

EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F

Then the first 4 bytes (first column) of the round key for round 9 are calculated asfollows:

i (decimal)	temp	After RotWord	After SubWord	Rcon (9)	After XOR with Rcon	w[i-4]	$w[i] = temp$ $\oplus w[i-4]$
36	7F8D292F	8D292F7F	5DA515D2	1B000000	46A515D2	EAD27321	AC7766F3

#### An AES Example

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

Plaintext:	0123456789abcdeffedcba9876543210
Key:	0f1571c947d9e8590cb7add6af7f6798
Ciphertext:	ff0b844a0853bf7c6934ab4364148fb9

#### Results

Table 2.4 shows the expansion of the 16-byte key into 10 round keys. The process is formed word by word, with each four-byte word occupying one column of the word round-key matrix.

Key Words	Auxiliary Function	
w0 = 0f 15 71 c9	RotWord (w3) = 7f 67 98 af = x1	
w1 = 47 d9 e8 59	subword (x1) = d2 85 46 79 = y1	
w2 = 0c b7 ad d6	Rcon (1) = 01 00 00 00	
w3 = af 7f 67 98	y1 $\oplus$ Rcon (1) = d3 85 46 79 = z1	
$w4 = w0 \oplus z1 = dc 90 37 b0$	RotWord (w7) = 81 15 a7 38 = x2	
$w5 = w4 \oplus w1 = 9b 49 df e9$	subword (x2) = 0c 59 5c 07 = y2	
$w6 = w5 \oplus w2 = 97 fe 72 3f$	Rcon (2) = 02 00 00 00	
$w7 = w6 \oplus w3 = 38 81 15 a7$	y2  Rcon (2) = 0e 59 5c 07 = z2	
$w8 = w4 \oplus z2 = d2 \ c9 \ 6b \ b7$	RotWord(w11) = ff d3 c6 e6 = x3	
$w9 = w8 \oplus w5 = 49 \ 80 \ b4 \ 5e$	SubWord(x3) = 16 66 b4 83 = y3	
$w10 = w9 \oplus w6 = de \ 7e \ c6 \ 61$	Rcon(3) = 04 00 00 00	
$w11 = w10 \oplus w7 = e6 \ ff \ d3 \ c6$	y3  Rcon(3) = 12 66 b4 8e = z3	
$w12 = w8 \oplus z3 = c0$ af df 39	RotWord (w15) = ae 7e c0 b1 = x4	
$w13 = w12 \oplus w9 = 89$ 2f 6b 67	subWord (x4) = e4 f3 ba c8 = y4	
$w14 = w13 \oplus w10 = 57$ 51 ad 06	Rcon (4) = 08 00 00 00	
$w15 = w14 \oplus w11 = b1$ ae 7e c0	y4  Rcon (4) = ec f3 ba c8 = 4	

Table 2.4 Expansion of the 16-byte key into 10 round keys

The left-hand column shows the four round-key words generated for each round. The right-hand column shows the steps used to generate the auxiliary word used in key expansion. The key itself serving as the round key for round 0.

Next, Table 2.5 shows the progression of **State** through the AES encryption process. The first column shows the value of **State** at the start of a round. For the first row, **State** is just the matrix arrangement of the plaintext. The second, third, and fourth columns show the value of **State** for that round after the SubBytes, ShiftRows,andMixColumns transformations, respectively. The fifth column shows the roundkey.

Key Words	Auxiliary Function
$w16 = w12 \oplus z4 = 2c$ 5c 65 f1	RotWord(w19) = 8c dd 50 43 = x5
$w17 = w16 \oplus w13 = a5$ 73 0e 96	SubWord(x5) = 64 cl 53 la = y5
$w18 = w17 \oplus w14 = f2$ 22 a3 90	Rcon(5) = 10 00 00 00
$w19 = w18 \oplus w15 = 43$ 8c dd 50	y5 ⊕ Rcon(5) = 74 cl 53 la = z5
$w20 = w16 \oplus z5 = 58$ 9d 36 eb	RotWord (w23) = 40 46 bd 4c = x6
$w21 = w20 \oplus w17 = fd$ ee 38 7d	SubWord (x6) = 09 5a 7a 29 = y6
$w22 = w21 \oplus w18 = 0f$ cc 9b ed	Rcon(6) = 20 00 00 00
$w23 = w22 \oplus w19 = 4c$ 40 46 bd	y6 $\oplus$ Rcon(6) = 29 5a 7a 29 = z6
$w24 = w20 \oplus z6 = 71 \text{ c7 4c c2}$	RotWord (w27) = a5 a9 ef cf = x7
$w25 = w24 \oplus w21 = 8c 29 74 \text{ bf}$	SubWord (x7) = 06 d3 bf 8a = y7
$w26 = w25 \oplus w22 = 83 \text{ e5 ef 52}$	Rcon (7) = 40 00 00 00
$w27 = w26 \oplus w23 = \text{cf a5 a9 ef}$	y7  ⊕ Rcon(7) = 46 d3 df 8a = z7
w28 = w24 ⊕ z7 = 37 14 93 48	RotWord (w31) = 7d al 4a f7 = x8
w29 = w28 ⊕ w25 = bb 3d e7 f7	SubWord (x8) = ff 32 d6 68 = y8
w30 = w29 ⊕ w26 = 38 d8 08 a5	Rcon (8) = 80 00 00 00
w31 = w30 ⊕ w27 = f7 7d a1 4a	y8 ⊕ Rcon(8) = 7f 32 d6 68 = z8
w32 = w28 ⊕ z8 = 48 26 45 20	RotWord (w35) = be 0b 38 3c = x9
w33 = w32 ⊕ w29 = f3 1b a2 d7	SubWord (x9) = ae 2b 07 eb = y9
w34 = w33 ⊕ w30 = cb c3 aa 72	Rcon (9) = 1B 00 00 00
w35 = w34 ⊕ w32 = 3c be 0b 3	y9   Rcon (9) = b5 2b 07 eb = z9
w36 = w32 $\oplus$ z9 = fd 0d 42 cb	RotWord (w39) = 6b 41 56 f9 = x10
w37 = w36 $\oplus$ w33 = 0e 16 e0 1c	SubWord (x10) = 7f 83 b1 99 = y10
w38 = w37 $\oplus$ w34 = c5 d5 4a 6e	Rcon (10) = 36 00 00 00
w39 = w38 $\oplus$ w35 = f9 6b 41 56	y10 ⊕ Rcon (10) = 49 83 b1 99 = z10
$w40 = w36 \oplus z10 = b4$ 8e f3 52 $w41 = w40 \oplus w37 = ba$ 98 13 4e $w42 = w41 \oplus w38 = 7f$ 4d 59 20 $w43 = w42 \oplus w39 = 86$ 26 18 76	

Table 2.5 progression of State through the AES encryption process

#### 10.EXPLIAN ABOUT THE RC4 ALGORITHM IN DETAIL.

RC4 is an encryption algorithm created in 1987 by Ronald Rivest of RSA Security. It is a stream cipher (figure 2.31), which means that each digit or character is encrypted one at a time. A cipher is a message that has been encoded.

A key input is pseudorandom bit generator that produces a stream 8-bit number that is unpredictable without knowledge of input key.

The output of the generator is called key-stream, is combined one byte at a time with the plaintext stream cipher using X-OR operation.

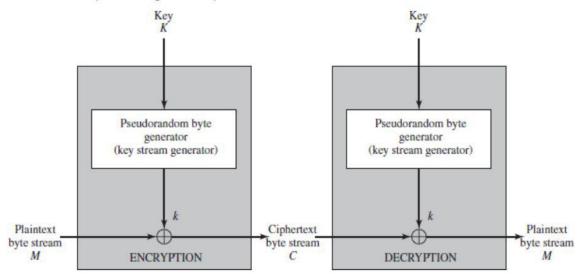


Figure 2.31 Stream Cipher Diagram

### Example

RC4 Encryption		RC4 Decryption	
10011000	Plaintext	11001000	Ciphertext
⊕ <u>01010000</u>	Key Stream	⊕ <u>01010000</u>	Key Stream
11001000	Ciphertext	10011000	Plaintext

### **Key Generation Algorithm**

A variable-length key from 1 to 256 byte is used to initialize a 256-byte state vector S, with elements S[0] to S[255]. For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion, then the entries in S are permuted again (Figure 2.32).

#### Initialization of S

The entries of S are set equal to the values from 0 to 255 in ascending orders, a temporary vector T, is created. If the length of the key k is 256 bytes, then k is assigned to T. Otherwise, for a key with length(klen) bytes, the first klen elements of T as copied from K and then K is repeated as many times as necessary to fill T.

```
// Initialization
for
i = 0 to 255 do S[i], √i; EnggTree.com
T[i] = K[i mod klen];
```

Next, use T to produce the initial permutation of S. Starting with S[0] to S[255], and for each S[i] algorithm swap it with another byte in S according to a scheme dictated by T[i], but S will still contain values from 0 to 255:

```
// Initial Permutation of S j = 0; for i = 0 to 255 do \{j = (j + S[i] + T[i]) \mod 256; Swap(S[i], S[j]); \}
```

### Pseudo random generation algorithm (Stream Generation)

Once the vector S is initialized, the input key will not be used. In this step, for each S[i] algorithm swap it with another byte in S according to a scheme dictated by the current configuration of S. After reaching S[255] the process continues, starting from S[0] again

```
//Stream Generation
i, j = 0;
while (true)
i = (i + 1) \mod 256;
j = (j + S[i]) \mod 256;
Swap(S[i], S[j]);
t = (S[i] + S[j]) \mod 256;
k = S[t];
                             S[i]+S[j]
                      2
                                                                        253 254 255
                 1
     S
                                          S[i]L
                                                             J S[j]
                                 ww$[i]ts[j].com
                             Figure 2.32 PRGA Algorithm
                                                                              253 254 255
      S 0 1 2 3 4
      T
                                    (a) Initial state of S and T
      T
                                  (b) Initial permutation of S
                                                                    S[t]
                                      t = S[i] + S[j] 4
                                   (c) Stream generation
                              Figure 2.33 RC4 Algorithm
```

### **Encrypt using XOR**

To encrypt, XOR the value k with the next byte of plaintext.

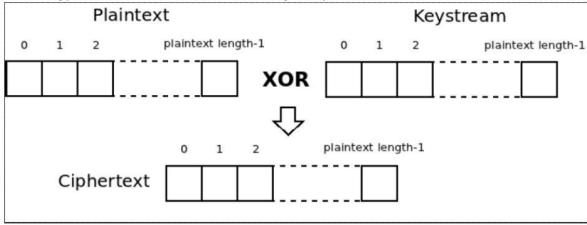


Figure 2.34 RC4 Encryption

### Decrypt using XOR

To decrypt, XOR the value k with the next byte of ciphertext.

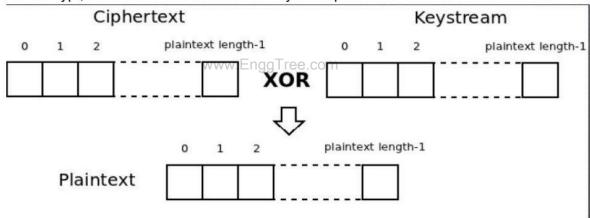


Figure 2.35 RC4 Decryption

#### Advantage

It is faster and more suitable for streaming application

#### 11.EXPLAIN ABOUT THE KEY DISTRIBUTION IN DETAILS

#### Symmetric Key Distribution Using Symmetric Encryption

- ➤ In Symmetric key encryption, the two parties to an exchange must share the same key, and that key must be protected from access by others. Therefore, the term that 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. A can select a key and physically deliver it to B.
- 2. A third party can select the key and physically deliver it to A and B.
- If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
- If A and B each has an encrypted connection to a third-party C, C can deliver a key on the encrypted links to A and B.
- Physical delivery (1 & 2) is simplest but only applicable when there is personal contact between recipient and key issuer. This is fine for link encryption where devices & keys occur in pairs, but does not scale as number of parties who wish to communicate grows. 3 are mostly based on 1 or 2 occurring first.
- A third party, whom all parties trust, can be used as a trusted intermediary to mediate the establishment of secure communications between them (4). Must trust intermediary not to abuse the knowledge of all session keys. As numbers of parties grow, some variant of 4 is only practical solution to the huge growth in number of keys potentially needed.

### **Key Distribution Centre**

- > The use of a key distribution centre is based on the use of a hierarchy of keys. At a minimum, two levels of keys are used.
- Communication between end systems is encrypted using a temporary key, often referred to as a Session key.
- > Typically, the session key is used for the duration of a logical connection and then discarded
- Master key is shared by the key distribution centre and an end system or user and used to encrypt the session key.

#### **Key Distribution Scenario**

▶ Let us assume that user A wishes to establish a logical connection with B and requires a one-time session key to protect the data transmitted over the connection. A has a master key, K<sub>a</sub>, known only to itself and the KDC; similarly, B shares the master key K<sub>b</sub> with the KDC (Figure 2.36). The following steps occur:

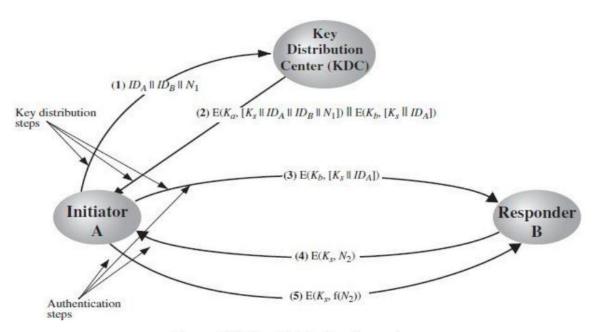


Figure 2.36 Key Distribution Scenarios

- 1. An issue 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<sub>1</sub>, for this transaction, which we refer to as a nonce. The nonce may be a timestamp, a counter, or a random number; the minimum requirement is that it differs with each request. Also, to prevent masquerade, it should be difficult for an opponent to guess the nonce. Thus, a random number is a good choice for a nonce.
- 2. The KDC responds with a message encrypted using Ka Thus, A is the only one who can successfully read the message, and A knows that it originated at the KDC. The message includes two items intended for A:
  - The one-time session key, Ks, to be used for the session
  - The original request message, including the nonce, to enable A to match this
    response with the appropriate request

Thus, A can verify that its original request was not altered before reception by the KDC and, because of the nonce, that this is not a replay of some previous request. In addition, the message includes two items intended for B:

- The one-time session key, Ks to be used for the session
- An identifier of A (e.g., its network address), IDA

These last two items are encrypted with Kb (the master key that the KDC shares with B). They are to be sent to B to establish the connection and prove A's identity.

- 3. A store the session key for use in the upcoming session and forwards to B the information that originated at the KDC for B, namely, E (K<sub>b</sub>, [K<sub>s</sub> || ID<sub>A</sub>]). Because this information is encrypted with K<sub>b</sub>, it is protected from eavesdropping. B now knows the session key (K<sub>s</sub>), knows that the other party is A (from ID<sub>A</sub>), and knows that the information originated at the KDC (because it is encrypted using K<sub>b</sub>). At this point, a session key has been securely delivered to A and B, and they may begin their protected exchange. However, two additional steps are desirable:
- 4. Using the newly minted session key for encryption, B sends a nonce, N2, to A.
- **5.** Also using  $K_s$ , A responds with  $f(N_2)$ , where f is a function that performs some transformation on  $N_2$  (e.g., adding one).

#### Session Key Lifetime

- The distribution of session keys delays the start of any exchange and places a burden on network capacity. A security manager must try to balance these competing considerations in determining the lifetime of a particular session key.
- For connection-oriented protocols, one obvious choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session.
- If a logical connection has a very long lifetime, then it would be prudent to change the session key periodically, perhaps every time the PDU (protocol data unit) sequence number cycles.
- For a connectionless protocol, such as a transaction-oriented protocol, there is no explicit connection initiation or termination.
- Thus, it is not obvious how often one needs to change the session key. The most secure approach is to use a new session key for each exchange.
- ➤ A better strategy is to use a given session key for a certain fixed period only or for a certain number of transactions.

#### **UNIT-III PART-A**

#### 1. What is meant by the Diffie-Hellman key exchange?

(A/M-12)

An element g is called a generator of a group G if every element in G can be expressed as the product of finitely many powers of g.

If p1 i n itgr hn te nmes c

multiplication as its operation. It is written as  $(Z/pZ)\times or Zp^*$ .

#### 2. How does Diffie-Hellman key exchange achieve security?

(N/D-08)

Diffie—Hellman key exchange is a specific method of exchanging cryptographic keys. It is one of the earliest practical examples of key exchange implemented within the field of cryptography. The Diffie—Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

#### 3. What is weak collision resistance? What is the use of it?

(M/J-13)

For any given block x, It is computationally infeasible to find Y X with H(Y) H(X). It guarantees than an alternative message hashing to the same value as a given message cannot found. This prevents forgery when as encrypted hash code is used.

#### 4. What is meant by EIGamal cryptosystem?

(A/M-11)

The ElGamal system is vavpublic-key-cryptosystem based on the discrete logarithm problem. It consists of both encryption and signature algorithms. The encryption algorithm is similar in nature to the Diffie-Hellman key agreement protocol.

#### 5. What is meant by one-way property in hash function?

(A/M - 11)

For any given code h, it is computationally infeasible to find X such that H(x) = h. A hash function, by itself, does not provide message authentication. A secret key must be used in some fashion with the hash function to produce authentication. A MAC, by definition, uses a secret key to calculate a code used for authentication.

#### 6. List out the requirements of kerberos.

(A/M - 11)

The requirements of Kerberos are as follows:

(1) Secure (2) Reliable (3) Transparent (4) Scalable

#### 7. What is meant by life cycle of a key?

(A/M-12)

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. Using a key repetitively allows an attacker to build up a store of ciphertext (and possibly plaintexts) which may prove sufficient for a successful cryptanalysis of the key value. If you suspect that an attacker may have obtained your key, then your key is considered compromised.

#### 8. What is a hash function?

(N/D-09)

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.

#### 9. What are the types of attacks addressed by message authentication?

There are four types of message authentication:

- 1. Masquerade: Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or no receipt by someone other than the message recipient.
- 2. Content modification: Changes to the contents of a message, including insertion, deletion, transposition, and modification.
- 3. Sequence modification: Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
- 4. Timing modification: Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.

# 10. What are two levels of functionality that comprise a message authentication or digital signature mechanism?

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 a message.

## 11. What is the difference between an unconditionally secure cipher and a computationally secure cipher?

An encryption scheme is unconditionally secure if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. An encryption scheme is said to be computationally secure if: (1) the cost of breaking the cipher exceeds the value of the encrypted information, and (2) the time required to break the cipher exceeds the useful lifetime of the information.

## 12. What is the difference between a message authentication code and a one-way hash function? (N/D-09)

A hash function, by itself, does not provide message authentication. A secret key must be used in some fashion with the hash function to produce authentication. A MAC, by definition, uses a secret key to calculate a code used for authentication.

#### **UNIT III**

#### PUBLIC KEY 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 - ElGamal cryptosystem – Elliptic curve arithmetic-Elliptic curve cryptography.

## 1.EXPLAIN ABOUT THE MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHYY Prime numbers

Prime numbers have divisors of 1 and its number itself.

#### Prime factorisation

To compute **GCD** of any two numbers in prime factorization approach we need to find **prime factors** of the two numbers.

#### Fermat Theorem or Fermat's little theorem

If **a** belongs to integer , P is a prime number that does not divide **a** then a P congruent a (mod P)
ie., a P ≡ a (mod P)

#### In special case

www.EnggTree.com

 $a^{P-1} \equiv 1 \pmod{P}$  if GCD(a,P)=1. where a and p are coprime. It is mainly used to solve modular exponentiation.

Eg. Compute the value of 2 10 mod 11

$$2^{10} \equiv 1 \pmod{11}$$

Eg. Compute the value of 2 340 mod 11

$$(2^{340})=(2^{10})^{34} \mod 11$$
  
=1 mod 11=> 1

#### //Proof.

Take division algorithm

a = p.q + r where can be  $0 \le r \le p-1$ 

let g.c.d (a,p) ie coprime

a is not divisible by p hence 1<= r <=p-1

fact says that if **a** leaves remainder r where  $1 \le r \le p-1$  on dividing by **p** then ka,  $1 \le k \le p-1$  also leaves remainders from 1 to p-1.

```
Means
If a,2a,3a,...(p-1)a surely gives remainders 1,2,3,...(p-1)
So if multiply
a*2a*3a*...*(p-1)a \equiv 1.2.3...(p-1) \pmod{p}
hence a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}
which returns
a p-1 ≡ 1 (mod p) [as mod p cannot divide (p-1)!]
hence proved. //
Eg.6 <sup>10</sup> mod 11
Sol. 6 <sup>11-1</sup> mod 11
 = 1 [as per theorem]
Eg. 5 <sup>15</sup> mod 13
=(5^2 \mod 13) * (5^{13} \mod 13)
=(25 \mod 13) * 5
=(12 * 5) \mod 13
=60 \mod 13
=8.
```

#### Euler's theorem

If n and a are coprime positive intergers then a phi(n) ≡1 (mod )n In this theorem phi(n) = n-1. n is prime number and phi(n) is Euler's phi function. Euler's phi function is also called Euler's totient function and hence named as Euler's totient theorem or Euler's theorem.

#### Euler's phi function or Euler's totient function(

Euler's phi function phi (n) returns the numbers of integers from 1 to n, that are relatively prime to n.

The phi function is computed phi(n) using various methods. They are

- 1. If n is a prime number then phi (n) = n-1
- 2. If n is a composite number then

Find the prime factors of that number and compute the phi

function value as used in step 1.otherwise 2.2.Find prime powers (p n)of the given number n. For computing the phi value of prime powers we have to use the formula

$$(p^{a} - p^{a-1}).$$

Eg. Compute Eulers's totient function for the values 3,8 1.  $phi(3)=3_51=2$ 2.  $phi(8) = 2^{3}$ 

$$= 2^{3} - 2_{3}^{3-1}$$
 (since p  $^{a}$  - p  $^{a}$  ) = 2 - 2 8-4=4

#### **Primality Testing Methods**

Primality testing method is a method to find and to prove whether the given number is prime number.

#### 1. Naive Algorithm:

Naïve Algorithm is used to divide the given input number P by all the integers starting from 2 to root of P-1

If any one of them is a divisor, then the input number P is not a prime. Otherwise, it is considered as a prime number

#### Algorithm:

- Pick any integer P that is greater than 2
- 2. Try to divide P by all integers starting from 2 to the square root of P
- If P is divisible by any one of these integer, we can conclude that P is a composite
- 4. Else P is a prime number

#### Example:

Find the primality test for the number 100 using naïve algorithm.

- 1. P=100
- 2. 2,3,4,5,6,7,8,9
- 3. Case 1: 100/2 = 50(composite)

Therefore, 100 is not a prime numbergTree.com

#### 2. Fermat's Primality Test:

```
If P is a prime and P does not divide a, which is a natural number then a^{P-1} \equiv 1 \pmod{P}
```

#### Example:

Check whether the given number 12 is prime number or not using Fermat's theorem

Given P = 12

To check whether 12 is prime number or not, we have to check

```
a_{12-1} = 1 \pmod{P}

a_{11} = 1 \pmod{12}

a_{11} = 1 \pmod{12}

a_{11} = 1 \pmod{12}

Where 1 \le a \le 12
```

We have to calculate a 11 mod 12

If it is equal to 1, then it is called prime number. Otherwise, it is called composite number.

composite number. Consider, a = 5  $5^{11} \equiv 1 \pmod{12}$ (i.e)  $5^{11} \mod{12} = 5$ It is not equal to 1 Therefore it is not a prime number

#### 3. Miller - Rabin Primality Test

```
Function Miller-Rabin (x)

x-1 = (2 )y //x is the input number for primality test

Down to aded if Con 2 is the Giser Ce. Com
```

```
select a randomly in the range [2, (x-1)] Z = a \mod x if Z congruent 1 (mod x) then return prime for i = 1 to w - 1 {
If Z congruent -1 (mod x) then return prime Z = Z \mod x }
return composite

Example:
Find the primality for 7 \times = 7
As per algorithm, x - 1 = 7 - 1 = 6 = 2 \times 3 \times = 7, w = 1, y = 3
Z = a \mod 7
a = 2 randomly), where [1<=a<=x-1]
Z = 2 \mod 7 = 1
Value of Z = 1, Z = 7 is concluded as prime number
```

#### **Chinese Remainder Theorem**

States that when the moduli of a system of linear congruencies are pairwise prime, there is a unique solution of the system modulo, the product of the moduli.

```
x \equiv a \pmod{m}.
```

Chinese mathematician Sun Tsu Suan-Ching asking the following problem:

There are certain things whose number is unknown. When divided by 3, the remainder is 2; when divided by 5, the remainder is 3; and when divided by 7, the remainder is 2. What will be the number of things?

(Otherwise) Mangos are divided into groups consisting of 3 mangos in each group remaining is 2. If the mangos are divided into groups consisting of 5 mangos in each group remaining 3.

If mangos are divided into groups consisting of 7 mangos in each group remaining 2. Totally how many mangos are available?

```
x \equiv a1 \mod(m1)

x \equiv a2 \mod(m2)

x \equiv a3 \mod(m3)

x = \Sigma(a_iM_iy_i) = (a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3) \mod M
```

Let M1, M2, ..., Mn be (pairwise) relatively prime numbers. Then the system:

```
Step1: Calculate M
M = m1*m2*m3 ... mn.
```

Step 2: Calculate M<sub>k</sub> =M/m<sub>k</sub>

Step 3: Find Inverse of MK (ie)yk

#### Find the X using CRT

```
x = 2 mod(3)

> x = 3 mod(5)

> x = 2 mod(7)

Downloaded from EnggTree.com
```

```
a_1 = 2, a_2 = 3, a_3 = 2; m_1 = 3, m_2 = 5, m_3 = 7;

i. M = m_1 \times m_2 \times m_3 = 105.

ii. For each equation calculate

M_k = M/m_k (ie) M_1 = M/m_1 = 105/3 = 35

M_2 = M/m_2 = 105/5 = 21

M_3 = M/m_3 = 105/7 = 15
```

iii. inverse of  $M_k$  (ie)  $y_k$  inverse of  $M_1$  (ie)  $y_1 = 35^{-1} \mod (3) = 35^{3-2} \mod (3) = 2$  [since Fermat's inverse theorem or easy inverse method like  $35 \times 2 \mod 3 = 1$  (ie) 2]

```
y_2 = 1; y_3 = 1

X = \Sigma(a_i M_i y_i) = (a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3) \mod M

X = [(2 \times 35 \times 2) + (21 \times 3 \times 1) + (2 \times 15 \times 1)] \mod 105

= (140 + 63 + 30) \mod 105

= 233 \mod 105

X = 23
```

#### **Exponentiation**

- Exponentiation is a type of operation where two elements are used in which one element is considered as a base element and another as an exponential element.
- For example, b is an example of exponential operation where x is a base element and y is an exponential element.w.EnggTree.com
  - When y is a positive integer, exponentiation is performed in a similar way to repeated multiplication is performed.
- Modular exponentiation is a type of exponentiation in which a modulo division operation is performed after performing an exponentiation operation.
- For example, (xy mod n), where n is an integer number.

The exponentiation is an important concept discussed in many cryptographic algorithms such as RSA, Diffie-Hellman, Elgamal, etc.,

```
Example:1

11' mod 13

11' mod 13 = 121 mod 13 = 4

11' mod 13 = (11^2 \text{ mod } 13 \times 11^2 \text{ mod } 13) \text{ mod } 13

= (4 \times 4) \text{ mod } 13

= 16 \text{ mod } 13

= 3

11' mod 13 = (11^4 \text{ mod } 13 \times 11^2 \text{ mod } 13 \times 11^1 \text{ mod } 13) \text{ mod } 13

= (3 \times 4 \times 11) \text{ mod } 13

= (132) \text{ mod } 13

= 2
```

### Find the result of 2<sup>90</sup> mod 13.

#### Solution:

```
Step 1: Split x and y into smaller parts using exponential rules as shown below: 2^{90} \mod 13 = 2^{50} \times 2^{40}
Step 2: Calculate mod n for each part 2^{50} \mod 13 = 1
```

```
2^{40} \mod 13 = 1099511627776 \mod 13 = 3

Step 3: Use \mod_5 u lar multiplication properties to combine these 2 parts, we have 2^{9} \mod 13 = (2^{5} \times 2^{9})^{10} \mod 13

= (2^{50} \mod 13 \times 2^{40} \mod 13) \mod 13

= (4 \times 3) \mod 13 = (12) \mod 13 = 12
```

#### Logarithms or Indices

- Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups. If G is a multiplicative cyclic group and g is a generator of G, then from the definition of cyclic groups, we know every element h in G can be written as g for some x. The discrete logarithm to the base g of h in the group G is defined to be x.
- For example, if the group is Z<sub>5</sub><sup>\*</sup>, and the generator is 2, then the discrete logarithm of 1 is 4 because 2 ≡ 1 mod 5.
- Input: p prime number, a- primitive root of p, b a residue mod p.
- Goal: Find k such that a<sup>k</sup> = b(mod p). (In other words, find the position of y in the large list of {a, a, ..., a}.
- 14 is a primitive root of 19.
- For example  $L_{14}(5) = 10 \mod 19$ , because  $14^{10} = 5 \pmod 19$ .
- the inverse problem to exponentiation is to find the discrete logarithm of a number modulo p
- that is to find x where a x = b mod p
- written as x=log<sub>a</sub> b mod p or x=ind<sub>a,p</sub>(b)
- if a is a primitive root then always exists

#### 2.WRITE SHORT NOTES ON ASYMMETRIC KEY CIPHERS PUBLIC

#### KEY CRYPTOGRAPHY:

Principles of public key cryptosystems

The concept of public key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption. Key distribution under symmetric key encryption requires either

- (1) Two communicantsalready share a key, which someone has been distributed to them
- (2) The use of a key distribution center.
  - Digital signatures.

#### Characteristics of Public key cryptosystems

Public key algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristics:

 It is computationally infeasible to determine the decryption key given only the knowledgeof the cryptographic algorithm and the encryption key.

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

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

#### INGREDIANTS OF PUBLIC KEY CRYPTOGRAPHY

- Plaintext: This is the readable message or data that is fed into the algorithm as input.
- 2. Encryption algorithm: The encryption algorithm performs various transformations on the plaintext.
- Public and private keys: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
- 4. Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different cipher texts.
- 5. Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

#### **Encryption:**

The essential steps are the following:

- Each user generates a pair of keys to be used for encryption and decryption of messages.
- 2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.
- If A wishes to send a confidential message to B, A encrypts the message using B"s public key.
- 4. When B receives the message, it decrypts using its private key.

With this approach(Fig), all participants have access to public keys and private keys are generated locally by each participant and therefore, need not be distributed.

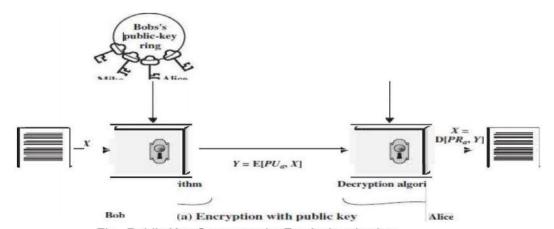


Fig . Public Key Cryptography For Authentication

Let the plaintext be X=[X1, X2, X3, ..,X] where m is the number of letters in some finite alphabets. Suppose A wishes to send a message to B.

B generates a pair of keys: a public key KUb and a private key KRb KRb is known only to B, whereas KUb is publicly available and therefore accessible by A.

With the message X and encryption key KUb as input, A forms the cipher text Y=[Y1, Y2, Y3> Yn]•

i.e., Y=EKUb(X)

The receiver can decrypt it using the private key KRb i.e., X=D KRb(Y)

The other approach (using sender s private key for encryption and sender s public key for decryption) will provide authentication which is illustrated in the following diagram(Fig 2.26).

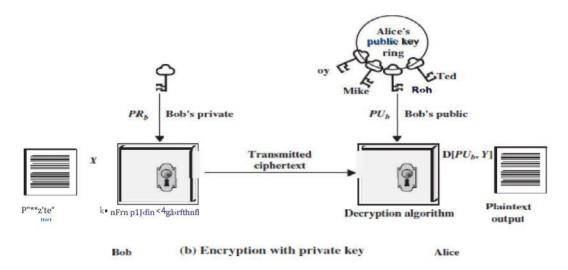


Fig . Private Key Cryptography For Authentication

www.EnggTree.com

The encrypted message serves as a digital signature. It is important to emphasize that the encryption process just described does not provide confidentiality.

Differences between public key Encryption and conventional Encryption

Conventional Encryption	Public-Key Encryption
Needed to Work:	Needed to Work:

- . The same -algorithm u'irh the same kcv is used for enczygtiun and dcc ption.
- \*. The sender and receiver must share h al gorilhrn and the key.

#### /'Vce‹Jed fnr SeciJriiy.-

- 1. The lk€'Y fTtMSt bfi kCpt 5tCF£:I.
- Tr rnusl be irnpnssihlc or at lczst impractical (+ decipher a message il the key ickept secret.
- K riov•'1cJyc 3f the algorithm pJus sample: of cipherrc xt mum be insufficient to dv\crrninc the key.
- Ono uJ urilhm ir used for «nomplion and a rcJared alyurirhm for dcc 'ption all h a pair of I:c\'.unc inr tn c ptiun and one for decryption.
- \*. The sender and ru-ceixer must cach have one of the matched peir oi kcTz (n 3t lhe same unc).
- · "\_\_\_\_\_"! -
- 1. One of the la'O keys must be kept secret.
- \*. JI must be impossible or al I«ast impractical r<1 decipher a message if one of lhc kcYs is kz'j•r sccrcl.
- Knowledge of the algori\km plus unc oi rhc kc\z
  plus sampler of ciphertext melt be irlsufficient m
  determine the othe r key'.

Public Key Cryptography for Security

www.EnggTree.com

There is some source A that produces a message in plaintext, X - [X 1, +2, ., X]. The elements of X are letters in some finite alphabet.

The message is intended for destination B. B generates a related pair of keys: a public key, PUT, and a private key, PRb PRtis known only to B, whereas PL/his publicly available and therefore accessible by A. With the message and the encryption key PL/,as input, A forms the ciphertext  $Y = [Y \ 1. \ Yo. \ . \ . \ YN]$ 

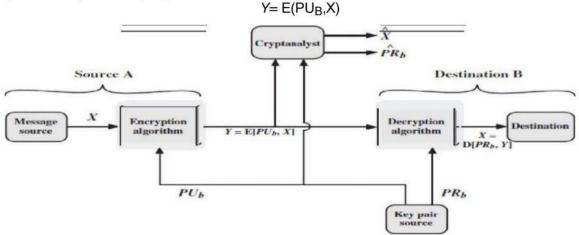


Figure.Public-Key Cryptosystem: Secrecy

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

An adversary, observing Y and having access to *PUT*, but not having access to *PRbor X*, must attempt to recover X and/or *PRb* It is assumed that the adversary does have knowledge of the encryption (E) and decryption (D) algorithms.

If the adversary is interested only in this particular message, then the focus of effort is to recover A by generating a plaintext estimate X \*. Often, however, the adversary is interested in being able to read future messages as well, in which case an attempt is made to recover PRtby generating an estimate *PRb*.

$$Y = E(PR_{a1})$$
  
 $X = D(PU_{a1} Y)$ 

In this case, A prepares a message to B and encrypts it using A's private key before transmitting it. B can decrypt the message using A's public key. Because the message was encrypted using A's private key, only A could have prepared the message.

Therefore, the entire encrypted message serves as a digital signature. In addition, it is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and in terms of data integrity.

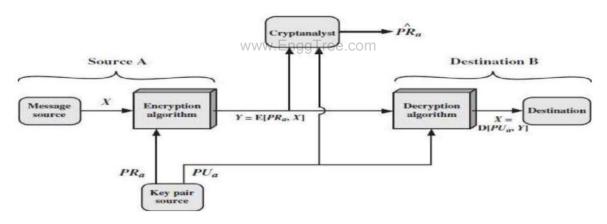


Figure .Public-Key Cryptosystem: Authentication

It is important to emphasize that the encryption process depicted in above Figures does not provide confidentiality. That is, the message being sent is safe from alteration but not from eaves dropping. This is obvious in the case of a signature based on a portion of the message, because the rest of the message is transmitted in the clear. Even in the case of complete encryption, as shown in Figure 13, there is no protection of confidentiality because any observer can decrypt the message by using the sender's public key.

#### **Authentication and Secrecy**

It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme (Figure 2.29):

Ciphertext Z = EKUb[EKRa(X)]

#### Plaintext X = EKUa[EKRb (Y)]

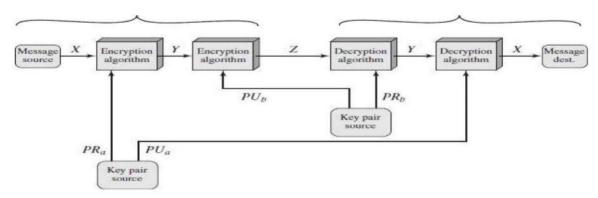


Figure .Public-Key Cryptosystem: Authentication and Secrecy

Initially, the message is encrypted using the sender s private key. This provides the digital signature. Next, we encrypt again, using the receiver s public key. The final ciphertext can be decrypted only by the intended receiver, who alone has the matching private key. Thus confidentiality is provided.

#### **Applications for Public-Key Cryptosystems**

We can classify the use of public-key cryptosystems into three categories

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

#### Requirements for public key cryptography

- It is computationally easy for a party B to generate a pair [KUb ,KRb]
- It is computationally easy for a sender A, knowing the public key and the message to be encrypted M, to generate the corresponding ciphertext: C=EKUb(M).
- It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

- It is computationally infeasible for an opponent, knowing the public key KUb, to determine the private key KRb
- It is computationally infeasible for an opponent, knowing the public key KUb, and a ciphertext C, to recover the original message M.
- The encryption and decryption functions can be applied in either order:

M = EKUb [D KRb (M) = DKUb [E KRb (M)]

#### Public-Key Cryptanalysis

#### Attack Type 1:

The public-key encryption scheme is vulnerable to a brute-force attack; therefore use large key. The tradeoff is that makes use of some sort of invertible mathematical function.

Therefore choose key size such that the brute force attack is not possible, at the same time should not be too slow for general use.

#### Attack type 2:

Attack is of other types (i.e.) given the algorithm and the public key deduce private key. This method has not been successful till date.

#### **Attack Type 3:**

A probable-message attack. When a confidential message is to be transmitted using DES,the attacker will find all  $2^{56}$  possible keys using the public key and discover the encrypted key by matching the generated cipher text and the actual cipher. This attack can be avoided by appending some random bits to the message.

#### 3.DESCRIBE THE RSA ALGORITHM IN DETAIL.

It was developed by Rivest, Shamir and Adleman. This algorithm makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n. <a href="https://www.EnggTree.com">www.EnggTree.com</a>

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

That is, the block size must be less than or equal to log 2 (n); in practice, the block size is k-bits, where 2 < n < 2 \* . Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C:

$$c = M \mod n$$
 $e \mod n = M \mod n = M \mod n$ 

Both the sender and receiver know the value of n. the sender knows the value of e and only the receiver knows the value of d. thus, this is a public key encryption algorithm with a public key of  $KU = \{e, n\}$  and a private key of  $KR = \{d, n\}$ . For this algorithm to be satisfactory for public key encryption, the following requirements must be met:

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

Select p, q p and q both prime,  $p \neq q$ 

Ca en a•• ml •> F—  $! \rightarrow i •J$ — 1)

Sc loci intege r c  $\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$ 

Calcula re d  $d \equiv e^{-1} \pmod{\phi(n)}$ 

Public key  $PU = \{e, n\}$ Private key  $PR = \{d, n\}$ 

Encryption by Bob with Alice's Public Key

Plaintext: M < n

Ciphertext:  $C = M^{\ell} \mod n$ 

Decryption by Alice with Alice's Public Key

Ciphentext: C

Plaimiext:  $M = C^d \mod n$ 

Fig. The RSA Algorithm

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

$$M = M \mod n$$

Given two prime numbers p and q and two integers, n and m, such that n=pq and 0<m<n, and arbitrary integer k, the following relationship holds

$$k6 (n) +1$$
  $k(p-1)(q-1) +1$  m mod n

where 6(n) — Euler totient function, which is the number of positive integers less than n and relatively prime to n. we can achieve the desired relationship, if

ed = k6(n)+1

This is equivalent to saying:

ed - 1, mod 6(n)

d = e - mod 6(n)

That is, e and d are multiplicative inverses mod 6(n). According to the rule of modular arithmetic, this is true only if d (and therefore e) is relatively prime to 6(n). Equivalently, gcd(6(n), d) = 1.

We are now ready to state the RSA scheme. The ingredients are the following:

p, q, two prime numbers (private, chosen)  $n \longrightarrow pq$  (public, calculated) e, with gcd(6 (n), e) = 1; 1 < e < 6 (n) (public, chosen) d-Ke-1 (mod 6 (n)) (private, calculated)

The steps involved in RSA algorithm for generating the key are

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

- Calculate n = p\*q = 17\*11 = 187
- Calculate 6(n) = (p-1)(q-1) = 16\*10 = 160.
- Select e such that e is relatively prime to 6(n) = 160 and less than 6(n); we choose

e = 7.

Determine d such that de K 1 (mod 160) and d < 160. The correct value is d — 23, because 23 • 7 = 161 = (1 • 160) + 1; d can be calculated using the extended Euclid's algorithm</li>

The resulting keys are public key PU —— {7, 187} and private key PR —— {23, 187}.

The example shows the use of these keys for a plaintext input of M—88. For encryption, we need to calculate  $C = 887 \mod 187$ .

Exploiting the properties of modular arithmetic, we can do this as follows.

```
88^{7} \mod 187 = [(88^{4} \mod 187) \cdot (88^{2} \mod 187)x (88^{1} \mod 187)] \mod 187

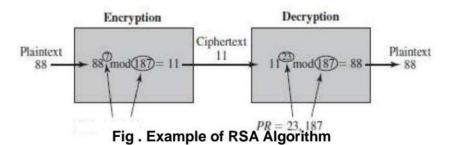
88^{1} \mod 187 = 88

88^{2} \mod 187 = 7744 \mod 187 = 77

88^{4} \mod 187 = 59,969,536 \mod 187 = 132

88^{7} \mod 187 = (88 \cdot 77 \cdot 132) \mod 187 = 894,432 \mod 187 = 11
```

For decryption, we calculate M—— 1123 mod 187:



#### Security of RSA:

There are three approaches to attack the RSA:

- 1. Brute force: This involves trying all possible private keys.
- 2. Mathematical attacks: There are several approaches, all equivalent in effort to factoring the product of two primes.
- 3. Timing attacks: These depend on the running time of the decryption algorithm.

Type 1 RSA Attack: Defense to Brute Force attack:

Use large key space (i.e) large number of bits in e and d the better secured but problems are,

- 1. Increases computing power
- 2. Factoring Problem

Type 2 RSA Attack: Mathematical Attack:

Mathematical approach takes 3 forms:

- Factor n = p\*q, hence find 6(n) and then d.
- Determine 6(n), directly without determining p and q and find

d. d=e- (mode (n))

Find d directly, without first determination 6(n).

Type 3 RSA Attack: Timing attacks:

This attack is learning for 2 reasons

- 1. Comes completely from unexpected direction
- 2. Cipher text only attack

#### Attack:

If the system does lastly the modular multiplication in majority of cases but takes longer time in few cases. The average is also longer.

The attack is done bit by bit

Start with left most bit b,,

Suppose first j bits are known.

For a given cipher text the attacker completes the j iteration.

If the bit is set then d<- (d \* a) mod n.

Methods to overcome Timing attacks:

- Constant exponentiation time: All exponentiations take the same amount of time before returning a result. This is a simple fix but does degrade performance.
- 2. Random delay: Better performance could be achieved by adding a random delay to the exponentiation algorithm to confuse the timing attack.
- 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.

#### 4.EXPLAIN ABOUT THE KEY MANAGEMENT IN DETAIL.

There are two uses of public key cryptography regarding the issues of key distribution. They are

- Distribution of public keys
- 2. Use of public key encryption to distribute secret keys

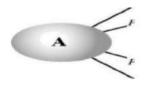
#### Distribution of Public Keys

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

- a) Public announcement
- b) Publicly available directory
- c) Public-key authority
- d) Public-key certificates

#### (a) Public Announcement of Public Keys

In public-key encryption the public key is public. Thus, if there is some broadly accepted public-key algorithm, such as RSA, any participant can send his or her public key to any other participant or broadcast the key to the community at large as shown in Figure 2.32.



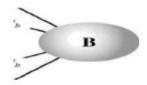


Figure. Uncontrolled Public-Key Distribution

#### Disadvantage:

Anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key.

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

#### (b) Publicly Available Directory

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization as shown in Figure 2.33. 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.
- A participant may replace the existing key with a new one at any time, due to either the key has been used for a large amount of data, or the corresponding private key has been compromised in some way.
- 4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory

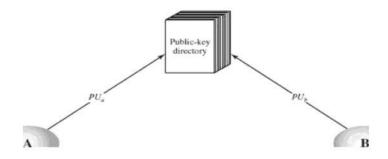


Figure Public-Key Publications

#### Vulnerabilities:

- 6 Tamper the records of public key directories.
- 6 If an adversary succeeds in obtaining or computing the private key of the directory authority, the adversary could authoritatively pass out counterfeit public keys and impersonate any participant and eavesdrop on messages sent to any participant.

#### (c) Public-Key Authority

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory. A typical scenario is illustrated in Figure 2.34.

www.EnggTree.com

As before, the scenario assumes that a central authority maintains a dynamic directory of public keys of all participants. In addition, each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key. The following steps (matched by number to Figure 2.34) occur:

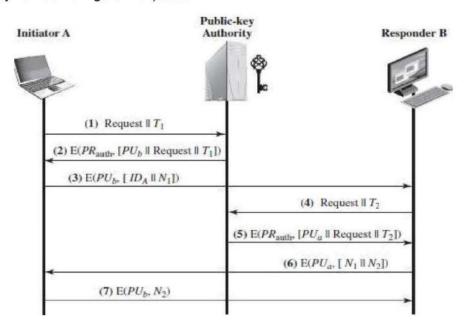


Figure . Public-Key Distribution Scenario

- 1. A sends a time stamped message to the public-key authority containing a request for the current public key of B.
- 2. The authority responds with a message that is encrypted using the authority's private key, *PRauthT* hus, A is able to decrypt the message using the authority's public key. Therefore, A is assured that the message originated with the authority. The message includes the following:
  - B's public key, PL/, which A can use to encrypt messages destined for B
  - The original request, to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority
  - The original timestamp, so A can determine that this is not an old message from the authority containing a key other than B's current public key
- 3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (*IDA*) and a nonce (/\/1), which is used to identify this transaction uniquely.
- 4,5 B retrieves A's public key from the authority in the same manner as A retrieved B's public key.
- 6.B sends a message to A encrypted with K aand containing A's nonce (/V1) as well as a new nonce generated by B (/V2) Because only B could have decrypted message (3), the presence of /V1 in message (6) assures A that the correspondent is B.
- 7. A returns 2. encrypted using B's public key, to assure B that its correspondent is A.

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

#### Disadvantages:

· Bottle neck at the authority.

#### (d) Public-Key Certificates

The scenario of Figure 2.35 is attractive, yet it has some **drawbacks**. The public-key authority could be somewhat of 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 tampering.

An alternative approach is to use **certificates** that can be used by participants to exchange keys without contacting a public-key authority.

A certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party.

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. Anyone needed this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature.

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.

These requirements are satisfied by the original proposal in. Denning added the following additional requirement:

4. Any participant can verify the currency of the certificate.

A certificate scheme is illustrated in Figure. Each participant applies to the certificate authority, supplying a public key and requesting a certificate.



#### Public-Key distribution of Secret Keys using public key cryptography:

- Use previous methods to obtain public-key
- Can use for secrecy or authentication
- Public-key algorithms are slow so usually want to use private-key encryption to protect message contents, Hence need a session key
  - a) Simple
  - b) Secret key distribution with confidentiality and authentication
  - c) Hybrid
  - d) Diffie Hell man key exchange

#### (a) Simple Secret Key Distribution:

- 1. A generates a public/private key pair (KUa,KRa) and transmits a message to B consisting of KUa and an identifier of A,IDA.
- 2. B generates a secret key , and transmits it to A, encrypted with A's public key.
- 3. A computes DKRa [EKUa [ s]] to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of Ks -
- 4. A discards KUaand KRa and B discards KUa.

#### Advantages:

- No keys exist before the start of the communication no key exist after the completion of communication
- Secure from eaves dropping

#### Disadvatages:

- Replay attack
- Meet in the middle attack
- A generates a public/private key pair {PUa, PRaj and transmits a message intended for B consisting of PL/aand an identifier of A, IDA.
- D intercepts the message, creates its own public/private key pair \(\begin{aligned} PUd, PRO \) and transmits \(PUs \) 0 0 \(IDA\) to \(B\).
- B generates a secret key, Ks, and transmits E(PL/s, Ks).
- D intercepts the message and learns Ks by computing D(PRd, E(PL/d, Ks)).
- D transmits E(PL/a, Ks) to A.

#### (b) Secret Key Distribution with Confidentiality and Authentication:

- 1. A uses B's public key to encrypt a message to B containing an identifier of A (ID A) and a nonce (N1), which is used to identify this transaction uniquely.
- 2. B sends a message to A encrypted with KUa and containing A's decrypted message (1) ,the presence of N1 in message (2) assures A that correspondent is B.
- 3. A returns Ne. encrypted using B's public key, to assurer B that its correspondent is A.
- 4. A select a secret key s find sends M = EKUb[EKRa[Ks]] TO B. Encryption of this message with B's public key ensures that only B can read it.;encryption with A's private key ensures that only A could have sent it.
- Computes DxUa[DKRb[M]]to recover thesecret key.

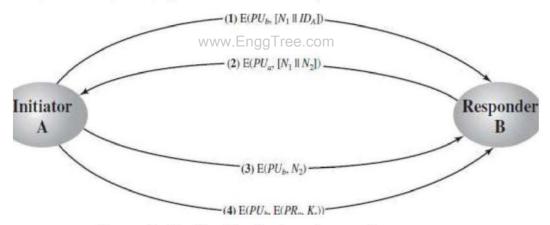


Figure . Public Key Distribution of secret Keys

#### Advantages:

Scheme ensures both confidentiality and authentication in the exchange of a secret key.

#### (c) A Hybrid Scheme

Public-key scheme is used to distribute the master keys. The following rationale is provided for using this three-level approach:

#### 1. Performance:

The public key encryption is used occasionally to update the master key between uses and KDC

When the distribution of session keys is done by public key encryption the performance degrades because of high computation needed by P.K.E.

2. **Backward compatibility:** The hybrid scheme is easily overlaid on an existing KDC scheme with minimal disruption or software changes.

The addition of a public-key layer provides a secure, efficient means of distributingmaster keys.

#### 6.EXPLIAN THE DIFFIE HELLMAN KEY EXCHANGE IN DETAILS.

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

First, we define a primitive root of a prime number p as one whose power generate all the integers from 1 to (p-1) i.e., if \_a' is a primitive root of a prime number p, then the numbers a mod p, a mod p, ... ap mod p

are distinct and consists of integers from 1 to (p-1) in some permutation.

For any integer \_b' and a primitive root \_a' of a prime number \_p', we can find a unique exponent \_i' such that

b - a' mod p where 0 i 1 (p-1)

The exponent \_i' is referred to as discrete logarithm.

#### The Algorithm

Figure 2.37 summarizes the Diffie-Hellman key exchange algorithm. There are publicly known numbers: a prime number \_q' and an integer a that is primitive root of q. suppose users A and B wish to exchange a key. User A selects a random integer XA q and computes YA = a XA mod q.

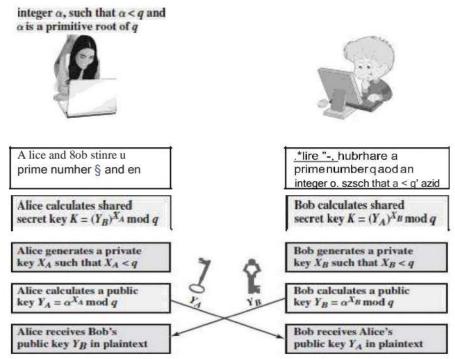


Fig.Diffie Hellman Key Exchange

Similarly, user B independently selects a random integer XB< q and computes YB a XB mod q. Each side keeps the X value private and makes the Y value available publicly to the other side.

User A computes the key as

These two calculations produce identical results.

$$K = (YB) \mod q$$

$$XB$$

$$= (a \mod q) \mod q$$

$$= (a) \mod q$$

$$= (a) \mod q$$

$$= (a) \mod q$$

$$= (a \mod q) \mod q$$

$$XB$$

$$= (YA) \mod q$$

Glob	al Public Elements
q	prime number
α	a <q a="" and="" is="" of="" primitive="" q<="" root="" td="" the=""></q>
Use	r A Key generation
Select XA Xy< q Calculate public Yr YA	mod q
User B Key generation	
Select XB XB q Calculate public YB YB=	5 150 200
W*-	of secret key by User A
$V = (VP) \mod a$	w EngaTree com
$K = (Y_A)^{XB} \mod q$	of secret key by User B

The result is that two sides have exchanged a secret key. The security of the algorithm lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms.

#### **Key Exchange Protocols**

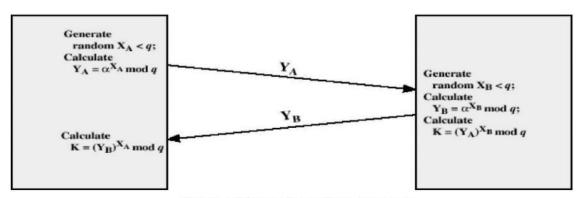


Figure . Diffe-Hellman Key Exchange

The protocol depicted in figure 2.38 is insecure against a man-in-the-middle attack. Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows:

- 1. Darth prepares for the attack by generating two random private keys XD1 and XD2 and then computing the corresponding public keys YD1 and Yo2-2. Alice transmits YA to Bob.
- = (YA) XD mod q. 3. Darth intercepts YA and transmits YD1 to Bob, Darth also calculates
- Bob receives YD1 and calculates K1 = (YD1) B mod q.
   Bob transmits XA to Alice.
- 6. Darth intercepts XA and transmits YD2 to Alice. Darth calculates 1 " (YB) XD1 mod q.
- 7. Alice receives Yo2 and calculates 2' (YD2) X mod g.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key K1 and Alice and Darth share secret key All future communication between Bob and Alice is compromised in the following way:

- Alice sends an encrypted message M: E( 2. M).
- 2. Darth intercepts the encrypted message and decrypts it, to recover M.
- 3. Darth sends Bob E(K1. M) or E(K1. M'), where M' is any message Example:

Key exchange is based on the use of the prime number q — 353 and a primitive root of 353, in this case a = 3. A and B select secret keys *EA*—— 97 and *XB*—— 233, respectively.

Each computes its public key:

A computes  $YA 3 \mod 353 = 40$ .

B computes YB— $3^{233} \mod 353 = 248$ . EnggTree.com

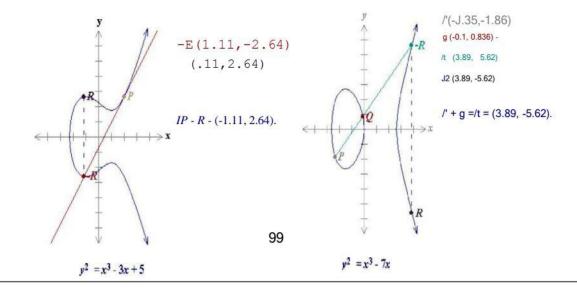
After they exchange public keys, each can compute the common secret key:

A computes K——  $(Yy) \mod 353 = 248 \mod 353 = 160$ . B computes K——  $(YA)^{XB} \mod 353 = 40^{2} \mod 353 = 160$ .

#### 7.DESCRIBE THE ELLIPTIC CURVE ARITHMETIC.

Elliptic Curves:

Where a,b,c,d and e are real numbers



A special addition operation is defined over elliptic curves and with the inclusion of a point  $\overline{\phantom{a}}$ 0" called point at infinity.

If three points are on a line intersecting an elliptic curve, then their sum is equal to this point at infinity O (which acts as the identity element for this addition operation)

Elliptic Curves over Galois field:

 $^3$ An elliptic group over the Galois Field Ed(a,b) is obtained by computing x +ax+b mod p for 0n xn p. The constants a&b are non-negative integers smaller than the prime number p must satisfy the condition.

 $4a^3 + 27b^2 \mod p \cdot 10$ 

For each value of x, one needs to determine whether or not it is a quadratic residue. If not then the point is not in the elliptic group Ed (a,b)

#### Addition and multiplication operation over elliptic groups:

Let the points P = (x1. y1) and Q = (X, Y2) be in the elliptic group Ed (a,b) and O be the point at infinity.

The rules for addition over the elliptic group Ed (a,b) are:

1.P+O=O+P=P

www.EnggTree.com

2. If 
$$x^2 = x^1$$
 and  $y^2 = -y^1$ , that is  $P = (x^1, y^1)$  and  $Q = (X^2, Y^2)' (x^1, -y^1)' - P$  Then  $P + Q = Q$ 

3. If Q I -P, then their sum P+Q =(x3.y3) is given by

$$x_3 = \lambda^2 - x_1 - x_2$$
  
 $y_3 = \lambda(x_1 - x_3) - y_1$ 

and

$$\lambda = \begin{cases} \frac{y2 - +}{x_2 - x_1} & p, & \text{,;;} \\ \frac{3x + a}{2J} & \text{if} & P = Q \end{cases}$$

Elliptic Curve Encryption:

Elliptic curve cryptography can be used to encrypt the plain text message M, into ciphertext. The plain text message M is encoded into a point PM from the finite set of points in the elliptic group, Ed(a,b).

The first step consists in choosing a generator point, G c Ed(a, b), such that the smallest value of n for which nG = o is a very large prime number.

The elliptic group Ed(a,b) and the generator point G are made public.

Each user select a private key,nA< n and compute the public key PA as PA = nAG

To encrypt the message point PM for Bob (B),

Alice (A) choses a random integer k and compute the ciphertext pair of points

Using Bob's public key PB

After receiving the ciphertext pair of points, Pc. Bob multiplies the firstpoint, (KG) with his private key nB and then adds the result to the second point in the ciphertext pair of points (PM + KPB)

$$(PM + B) - [nB(KG)] = (PM+KnBG)-[nB(KG)] = PM$$

which is the plaintext point, corresponding to the plaintext message M.

Only Bob knowing the private key nB. can remove nB(KG) from the second point of the ciphertext pair of point, i.e (PM + KGB), and hence retrieve the plaintext information PM

#### Elliptic curve cryptography

#### Security of ECC:

- 1. The cryptographic strength of elliptic curve encryption lies in the difficulty for a crypt analyst to determine the secret random number k from KP & P itself.
- 2. The fastest method to solve this problem (known as elliptic curve logarithm problem is the pollard factorization method).
- 3. The computational complexity for breaking the elliptic curve cryptosystem, using the pollard method is  $3.3 \times 10^{10}$  MIPS years for an elliptic curve key size of only 150bits.
- 4. For comparison the fastest method to break RSA, using General Number Field Sieve method to factor the composite integer n in to the two prime p & q requires  $2x10^{-11}$  MIPS years for a 768 bit RSA key &  $3x10^{-11}$  MIPS years for a RSA key length 1024 5. If the RSA key length is increased to 2048 bits, the GNES method will need  $3x10^{-20}$  MIPS years to factor n whereas increasing the elliptic curve key length to only 24 bits will impose a computational complexity of  $1.6x10^{-10}$  MIPS years.

#### Analog of Diffie-Hellman Key Exchange:

Key exchange using elliptic curves can be done in the following manner.

First pick a large integer q, which is either a prime number p or an integer of the form 2|| and elliptic curve parameters a and b. This defines the elliptic group of points Ed(a, b).

Next, pick a base point G = (x1. y1) in Ed(a, b) whose order is a very large value n. The order n of a point G on an elliptic curve is the smallest positive integer n such that nG = O. Eq(a, b) and G are parameters of the cryptosystem known to all participants.

- 1. A selects an integer nA less than n. This is A's private key. A then generates a public key PA nA x G; the public key is a point in Ed(a, b)•
- 2. B similarly selects a private key nB and computes a public key PB

#### **UNIT-IV PART-A**

#### 1. Why does PGP generate a signature before applying compression? (A/M-11)

The signature is generated before compression due to 2 reasons:

It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future.

#### 2. Write the four SSL Protocols.

(A/M-11)

- 1. SSL Handshake protocol
- 2. SSL Change cipher spec. protocol
- 3. SSL Alert Protocol
- 4. SSL Record Protocol

#### 3. What is meant by S/MIME?

(A/M-12)

S/MIME (Secure/Multipurpose Internet Mail Extensions) is a standard for public key encryption and signing of MIME data. S/MIME is on an IETF standards track and defined in a number of documents, most importantly RFCs (3369, 3370, 3850, 3851). S/MIME was originally developed by RSA Data Security Inc. The original specification used the IETF MIME specification with the de facto industry standard PKCS secure message format. Change control to S/MIME has since been vested in the IETF and the specification is now layered on cryptographic message syntax.

#### 4. What are the services provided by IPSec? ee.com

(N/D-09)

The services provided by IPSec are authentication, confidentiality and key management authentication. It ensures the identity of an entity. Confidentiality is protection of data from unauthorized disclosure. Key management is generation, exchange, storage, safeguarding, etc. of keys in a public key cryptography.

#### 5. What is meant by replay attack?

(A/M-11)

A replay attack (also known as playback attack) is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution (such as stream cipher attack).

#### 6. What is the difference between an SSL connection and SSL session? (M/J-09)

Connection is a transport that provides a suitable type of service. For SSL, such connections are peer-topeer relationships. The connections are transient. Every connection is associated with one session. Session: An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

#### 7. Why does ESP include a padding field?

(N/D-08)

The ciphertext needs to end on an eight octet boundary because the Authentication data field is properly aligned in the packet. This is what the protocol expects and if it doesn't follow the rules, it's considered to contain an error in the packet. It's like English or other languages. We expect sentences to end with a period so we know where one sentence ends and the other begins.

#### 8. What is the problem that kerberos addresses?

(A/M-12)

The problem that Kerberos addresses is this: Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service. In this environment a workstation cannot be trusted to identify its users correctly to network services.

#### 9. What is meant by the function of a compression function in a hash function?

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.

#### 10. How is signed data entity of S/MIME prepared?

Secure/Multipurpose Internet Mail Extension is a security enhancement to the MIME Internet e-mail format standard, based on technology from RSA data security. It is able to sign and/or encrypt messages.

#### 11. What are the services provided by IPSec?

- 1. Access control
- 2. Connectionless integrity
- 3. Data origin authentication
- 4. Rejection of replayed packets

#### 12. List out four requirements defined for kerberos.

(M/J-09)

The four requirements defined for Kerberos are:

- 1. Secure: A network eavesdropper should not be able to obtain the necessary information to impersonate a user. More generally Kerberos should be strong enough that a potential opponent does not find it to be the weak link.
- 2. Reliable: For all services that relay on Kerberos for access control, lack of availability of the supported services. Hence, Kerberos should be highly reliable and should employ distributed server architecture, with one system able to back up another.
- 3. Transparent: Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.
- 4. Scalable: The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

## 13. What are the entities that constitute a full-service kerberos environment? (N/D-08)

A full service environment consists of a Kerberos server, a number of clients and a number of application servers.

#### 14. What is the need of segmentation and reassembly function in PGP?

E-mail facilities often are restricted to a maximum message length. To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail. The segmentation is done after all of the other processing, including the radix-64 conversion. Thus, the session key component and signature component appear only once, at the beginning of the first segment.

www.EnggTree.com

3. A generates the secret key K = nAx PB B generates the secret key K = nB x EA.

	Global Public elements
Ed(a,b)	Elliptic curve with parameters a,b and q, where q is a prime or
an	integer of the form 2
_	
G	point on elliptic curve whose order is large value n
	Hear A Key Concretion
0.1	User A Key Generation
Select private	n A nA <n< td=""></n<>
Calculate public PA	PA nA X G
Calculate public 17	
g : g	User B Key Generation
Select private	nB nB‹n
Calculate public PA	PB = n B X G
Calculate public PA	PB=NBXG
1200 DEC 2001	Calculation of secret key by User A
K = nAX PB	www.EnggTree.com
	Calculation of secret key by User B
K = nBX PA	

Figure .ECC Diffie-Hellman Key Exchange

#### UNIT IV MESSAGE AUTHENTICATION AND INTEGRITY

**UNIT IV MESSAGE AUTHENTICATION AND INTEGRITY** Authentication requirement - Authentication function - MAC - Hash function - Security of hash function and MAC - SHA - Digital signature and authentication protocols - DSS- Entity Authentication: Biometrics, Passwords, Challenge Response protocols- Authentication applications - Kerberos, X.509

#### PART-B

#### 1.EXPLAIN THE AUTHENTICATION REQUIREMENT AND FUNCTION.

Communication across the network, the following attacks can be identified.

**Disclosure** – release of message contents to any person or process not possessing the appropriate cryptographic key.

Traffic analysis - discovery of the pattern of traffic between parties.

- In a connection oriented application, the frequency and duration of connections could be determined.
- In either a connection oriented or connectionless environment, the number and length of messages between parties could be determined.

**Masquerade** – insertion of messages into the network from fraudulent source. This can be creation of message by the attacker using the authorized port.

**Content modification** - changes to the contents of a message, including insertion, deletion, transposition, and modification.

**Sequence modification** - any modification to a sequence of messages between parties, including insertion, deletion, and reordering.

Timing modification - delay or replay of messages.

- In a connection oriented application, an entire session or sequence of messages could be replay of some previous valid session, or individual messages in the sequence could be delayed or replayed.
- In a connectionless application, an individual message could be delayed or replayed.

Source repudiation - denial of transmission of message by source.

Destination repudiation - denial of receipt of message by destination.

#### **AUTHENTICATION FUNCTION**

Any message authentication or digital signature mechanism can be viewed as having fundamentally two levels.

At the lower level, there must be some sort of function that produces an authenticator, a value to be used to authenticate a message.

At the higher-level, low-level function is then used as primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

The types of function that may be used to produce an authenticator are grouped into three classes.

Message Encryption - the ciphertext of the entire message serves as its authenticator.

#### 3. EXPLIAN MAC IN DETAIL.

**Message Authentication Code (MAC)** - a public function of the message and a secret key that produces a fixed length value that serves as the authenticator.

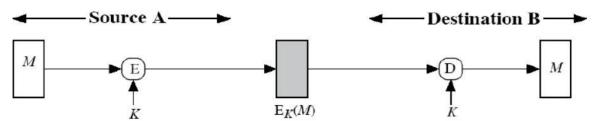
**Hash Function** - a public function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.

#### Message Encryption:

Message encryption Message encryption by itself can provide a measure of authentication.

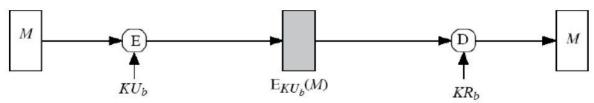
The analysis differs from symmetric and public key encryption schemes.

- (a) If symmetric encryption(fig.a) is used then:
  - A message m, transmitted from source A to destination B is encrypted using a secret key shared by A and B.
  - Since only sender and receiver knows key used
  - Receiver knows sender must have created it. Hence authentication is provided.
  - Know content cannot have been altered. Hence confidentiality is also provided.
  - If message has suitable structure, redundancy or a checksum to detect any changes
  - > Therefore Symmetric Encryption provides authentication and confidentiality.



- (a). Symmetric key encryption confidentiality, authentication and signature
- (b) If public-key encryption(Fig b) is used:

This method is the use of public key cryptography which provides confidentiality only. The sender A makes use of the public key of the receiver to encrypt the message. Here there is no authentication because any user can use B"s public key to send a message and claim that only A has sent it.



(b) Public key encryption confidentiality

In this method(Fig c) to have only authentication, the message is encrypted with the sender"s A"s private key. The receiver B uses the sender"s A"s public key to decrypt the message. Now A cannot deny that it has not transmitted since it only knows its private key. This is called as authentication or Digital Signature. Hence the problem is the,

- Receiver cannot determine whether the packet decrypted contains some useful message or random bits.
- > The problem is that anyone can decrypt the message when they know the public key of sender A.

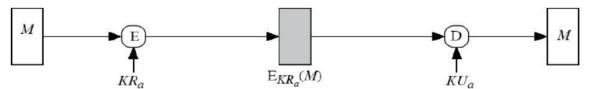


Figure (c) Public key encryption authentication and signature

This method (Fig d) provides authentication, confidentiality and digital signature. But the problem with this method is the complex public key cryptography algorithm should be applied twice during encryption and twice during decryption.

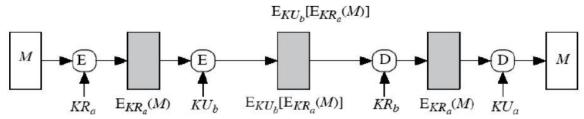


Figure (d) Public key encryption confidentiality, authentication and signature

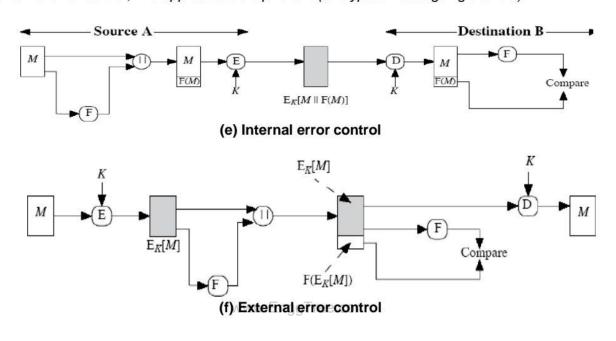
Suppose the message can be any arbitrary bit pattern, in that case, there is no way to determine automatically, at the destination whether an incoming message is the ciphertext of a legitimate message.

One solution to this problem is to force the plaintext to have some structure that is easily recognized but that cannot be replicated without recourse to the encryption function.

Append an error detecting code, also known as Frame Check Sequence (FCS) or checksum to each message before encryption "A" prepares a plaintext message M and then provides this as input to a function F that produces an FCS. The FCS is appended to M and the entire block is then encrypted.

At the destination, B decrypts the incoming block and treats the result as a message with an appended FCS. B applies the same function F to attempt to reproduce the FCS.

If the calculated FCS is equal to the incoming FCS, then the message is considered authentic. In the internal error control, the function F is applied to the plaintext, whereas in external error control, F is applied to the ciphertext (encrypted message fig e and d).



#### 4.2.MAC

An alternative authentication technique involves the use of secret key to generate a small fixed size block of data, known as cryptographic checksum or MAC that is appended to the message.

This technique assumes that two communication parties say A and B, share a common secret key "k". When A has to send a message to B, it calculates the MAC as a function of the message and the key.

$$MAC = C_K(M)$$

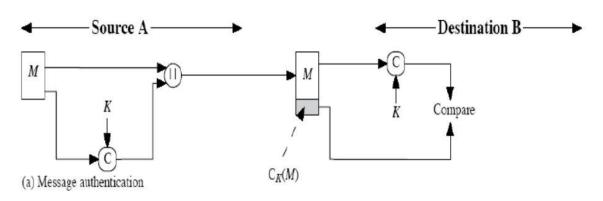
Where M – input message

C – MAC function

K – Shared secret key

The message plus MAC are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the shared secret key, to generate a new MAC.

The received MAC is compared to the calculated MAC. If it is equal, then the message is considered authentic(Fig g and h). A MAC function is similar to encryption. One difference is that MAC algorithm need not be reversible, as it must for decryption. In general, the MAC function is a many-to-one function.



#### (g) Message Authentication

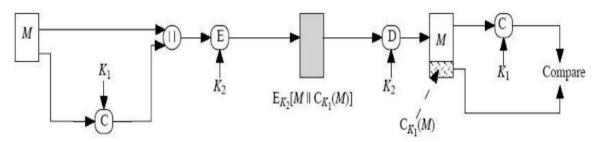


Figure (h) Message authentication and confidentiality, authentication tied to plain text

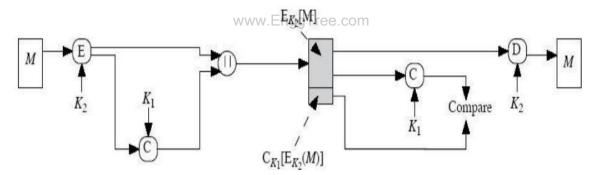


Figure (i) Message authentication and confidentiality, authentication tied to ciphertext Requirements for MAC:

When an entire message is encrypted for confidentiality, using either symmetric or asymmetric encryption, the security of the scheme generally depends on the bit length of the key.

Barring some weakness in the algorithm, the opponent must resort to a brute-force attack using all possible keys. On average, such an attack will require 2(k-1) attempts for a k-bit key.

If confidentiality is not employed, the opponent has access to plaintext messages and their associated MACs. Suppose k > n; that is, suppose that the key size is greater than the

MAC size. Then, given a known  $M_1$  and  $MAC_1$ , with  $MAC_1 = C_K$  ( $M_1$ ), the cryptanalyst can perform  $MAC_i = CK_i$  ( $M_1$ ) for all possible key values  $K_i$ .

At least one key is guaranteed to produce a match of MAC<sub>i</sub> = MAC<sub>1</sub>.

Note that a total of  $2^k$  MACs will be produced, but there are only  $2^n < 2^k$  different MAC values. Thus, a number of keys will produce the correct MAC and the opponent has no way of knowing the correct key. On average, a total of  $2^k/2^n = 2^{(k-n)}$  keys will produce a match. Thus, the opponent must iterate the attack:

#### Round 1

Given:  $M_1$ ,  $MAC_1 = C_K(M_1)$ 

Compute  $MAC_i = C_{Ki} (M_1)$  for all  $2^k$  keys

Number of matches ≈2(k-n)

#### Round 2

Given:  $M_2$ ,  $MAC_2 = C_K(M_2)$ 

Compute  $MAC_i = C_{Ki}(M_2)$  for the  $2^{(k-n)}$  keys resulting from Round 1

Number of matches ≈2(k-2xn) and so on

Consider the following MAC algorithm. Let  $M = (X_1||X_2||...||X_m)$  be a message that is treated as a concatenation of 64-bit blocks  $X_i$ . Then define

$$(M) = X_1 + X_2 ... X_m$$
  
 $C_k(M) = E_k(\Delta(M))$ 

Thus, the key length is 56 bits and the MAC length is 64 bits. If an opponent observes  $\{M||C(K,M)\}$ , a brute-force attempt to determine K will require at least  $2^{56}$  encryptions.

But the opponent can attack the system by replacing X1 through  $X_{m-1}$  with any desired values  $Y_1$  through  $Y_{m-1}$  and replacing  $X_m$  with  $Y_m$  where  $Y_m$  is calculated as follows:

$$Y_m = Y_1 + Y_2 \dots Y_{m1} + \Delta(M)$$

The opponent can now concatenate the new message, which consists of  $Y_1$  through  $Y_m$ , with the original MAC to form a message that will be accepted as authentic by the receiver. With this tactic, any message of length 64  $X_{(m-1)}$  bits can be fraudulently inserted.

Then the MAC function should satisfy the following requirements: The MAC function should have the following properties:

- If an opponent observes M and  $C_K(M)$ , it should be computationally infeasible for the opponent to construct a message M<sup>s</sup> such that  $C_K(M^s) = C_K(M)$
- $ightharpoonup C_K(M)$  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.
- Let M" be equal to some known transformation on M. i.e., M" = f(M).

# Downloaded from EnggTree.com

#### MAC based on DES

One of the most widely used MACs, referred to as Data Authentication Algorithm (DAA) is based on DES.

The algorithm(Fig 2) can be defined as using cipher block chaining (CBC) mode of operation of DES with an initialization vector of zero. The data to be authenticated are grouped into contiguous 64-bit blocks:  $D_1$ ,  $D_2$  ...  $D_n$  if necessary, the final block is padded on the right with zeros to form a full 64-bit block. Using the DES encryption algorithm and a secret key, a data authentication code (DAC) is calculated as follows:

```
\begin{split} O_1 &= E_K(D_1) \\ O_2 &= E_K(D_2 + O_1) \\ O_3 &= E_K(D_3 + O_2) \\ &\cdots \\ O_N &= E_K(D_N + O_{N-1}) \end{split}
```

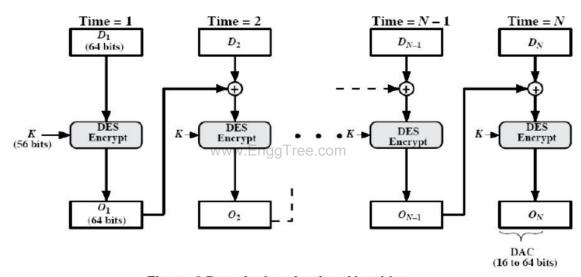


Figure.2 Data Authentication Algorithm

#### 3.EXPLAIN THE HASH FUNCTION IN DETAILS.

A variation on the message authentication code is the one way hash function. As with MAC, a hash function accepts a variable size message M as input and produces a fixed-size output, referred to as hash code H(M).

Unlike a MAC, a hash code does not use a key but is a function only of the input message. The hash code is also referred to as a message digest or hash value. There are varieties of ways in which a hash code can be used to provide message authentication, as follows:

In Fig (a) The message plus the hash code is encrypted using symmetric encryption. This is identical to that of internal error control strategy. Because encryption is applied to the entire message plus the hash code, confidentiality is also provided.

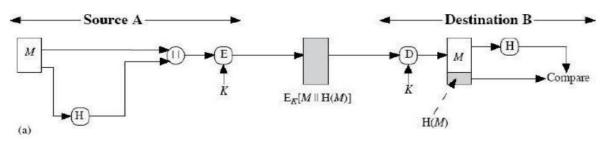
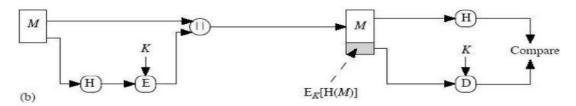


Figure (a) Hash Function

In Fig (b) Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality.

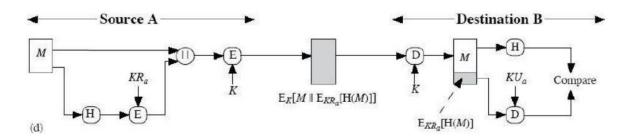


In Fig (c) Only the hash code is encrypted, using the public key encryption and using the sender"s private key. It provides authentication plus the digital signature.

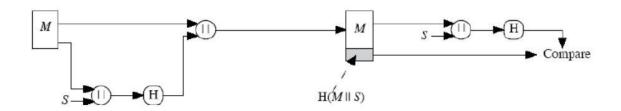


Figure (b & c) Basic use of Hash Function

In Fig (d) If confidentiality as well as digital signature is desired, then the message plus the public key encrypted hash code can be encrypted using a symmetric secret key.



In Fig (e) This technique uses a hash function, but no encryption for message authentication. This technique assumes that the two communicating parties share a common secret value "S". The source computes the hash value over the concatenation of M and S and appends the resulting hash value to M.



In Fig(f) Confidentiality can be added to the previous approach by encrypting the entire message plus the hash code.

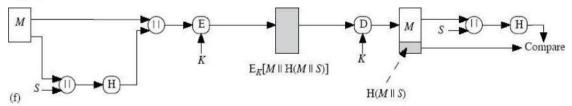


Figure (d,e & f) Basic use of Hash Function

A hash value h is generated by a function H of the form

$$h = H(M)$$

where M is a variable-length message and H(M) is the fixed-length hash value.

The hash value is appended to the message at the source at a time when the message is assumed or known to be correct. The receiver authenticates that message by recomputing the hash value.

#### Requirements for a Hash Function

- 1. H can be applied to a block of data of any size.
- 2. H produces a fixed-length output.
- 3. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.
- 4. For any given value h, it is computationally infeasible to find x such that H(x) = h. This is sometimes referred to in the literature as the one-way property.
- 5. For any given block x, it is computationally infeasible to find y x such that H(y) = H(x). This is sometimes referred to as weak collision resistance.
- 6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y). This is sometimes referred to as strong collision resistance.

The first three properties are requirements for the practical application of a hash function to message authentication.

The fourth property, the one-way property, states that it is easy to generate a code given a message but virtually impossible to generate a message given a code.

# Downloaded from EnggTree.com

The fifth property guarantees that an alternative message hashing to the same value as a given message cannot be found. This prevents forgery when an encrypted hash code is used.

The sixth property refers to how resistant the hash function is to a type of attack known as the birthday attack.

#### Simple Hash Functions

All hash functions operate using the following general principles. The input (message, file, etc.) 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 (XOR) of every block.

This can be expressed as follows:  $C_i = b_{i1} + b_{i1} \dots + b_{im}$ 

where

 $C_i = i^{th}$  bit of the hash code,  $1 \le i \le n$ m = number of n-bit blocks in the input  $b_{ij}$ = ith bit in jth block

#### Procedure:

- 1. Initially set the n-bit hash value to zero.
- 2. Process each successive n-bit block of data as follows:

www.EnggTree.com

- a. Rotate the current hash value to the left by one bit.
- b. XOR the block into the hash value.

#### **Birthday Attacks**

Suppose that a 64-bit hash code is used. One might think that this is quite secure. For example, if an encrypted hash code C is transmitted with the corresponding unencrypted message M, then an opponent would need to find an M' such that H(M') = H(M) to substitute another message and fool the receiver.

On average, the opponent would have to try about  $2^{63}$  messages to find one that matches the hash code of the intercepted message.

However, a different sort of attack is possible, based on the birthday paradox. The source, A, is prepared to "sign" a message by appending the appropriate m-bit hash code and encrypting that hash code with A's private key.

- 1. The opponent generates  $2^{m/2}$  variations on the message, all of which convey essentially the same meaning. (Fraudulent message)
- 2. The two sets of messages are compared to find a pair of messages that produces the same hash code. The probability of success, by the birthday paradox, is greater than 0.5. If no match is found, additional valid and fraudulent messages are generated until a match is made.

3. The opponent offers the valid variation to A for signature. This signature can then be attached to the fraudulent variation for transmission to the intended recipient. Because the two variations have the same hash code, they will produce the same signature; the opponent is assured of success even though the encryption key is not known.

Thus, if a 64-bit hash code is used, the level of effort required is only on the order of 232

#### MEET-IN-THE-MIDDLE ATTACK.

Divide a message M into fixed-size blocks  $M_1, M_2, ..., M_N$  and use a symmetric encryption system such as DES to compute the hash code G as follows:

$$H_o$$
 = initial value  
 $H_i$  =  $E_{Mi}$  [ $H_{i-1}$ ]  
 $G$  =  $H_N$ 

This is similar to the CBC technique, but in this case there is no secret key. As with any hash code, this scheme is subject to the birthday attack, and if the encryption algorithm is DES and only a 64-bit hash code is produced, then the system is vulnerable.

Furthermore, another version of the birthday attack can be used even if the opponent has access to only one message and its valid signature and cannot obtain multiple signings.

Here is the scenario; we assume that the opponent intercepts a message with a signature in the form of an encrypted hash code and that the unencrypted hash code is m bits long:

- 1. Calculate the unencrypted hash code G. EnggTree.com
- 2. Construct any desired message in the form Q<sub>1</sub>, Q<sub>2</sub>,..., Q<sub>N2</sub>.
- 3. Compute for  $H_i = EQ_i [H_{i-1}]$  for  $1 \le i \le (N-2)$ .
- 4. Generate  $2^{m/2}$  random blocks; for each block X, compute  $E_X[H_{N-2}]$  Generate an additional  $2^{m/2}$  random blocks; for each block Y, compute  $D_Y[G]$ , where D is the decryption function corresponding to E.
- 5. Based on the birthday paradox, with high probability there will be an X and Y such that  $E_X [H_{N-2}] = D_Y[G]$ .
- 6. Form the message  $Q_1$ ,  $Q_2$ ,...,  $Q_{N-2}$ , X, Y. This message has the hash code G and therefore can be used with the intercepted encrypted signature.

### 4.EXPLIAN THE SECURITY OF HASH FUNCTION AND MAC IN DETAILS

Just as with symmetric and public-key encryption, we can group attacks on hash functions and MACs into two categories: brute-force attacks and cryptanalysis.

#### **Brute-Force Attacks**

The nature of brute-force attacks differs somewhat for hash functions and MACs.

#### **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.

Requirements of Hash Function:

**One-way**: For any given code h, it is computationally infeasible to find x such that H(x) = h.

**Weak collision resistance**: For any given block x, it is computationally infeasible to find  $y \times with H(y) = H(x)$ .

**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 <sup>n</sup>
Weak collision resistance	2 <sup>n</sup>
Strong collision resistance	2 <sup>n/2</sup>

#### **Message Authentication Codes**

A brute-force attack on a MAC is a more difficult undertaking because it requires known message-MAC pairs. To attack a hash code, we can proceed in the following way. Given a fixed message x with n-bit hash code H/=H(x), a brute-force method of finding a collision is to pick a random bit string y and check if H(y) = H(x). The attacker can do this repeatedly off line.

To proceed, we need to state the desired security property of a MAC algorithm, which can be expressed as follows:

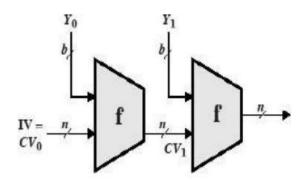
#### Cryptanalysis

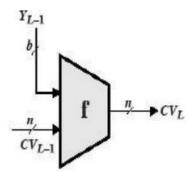
As with encryption algorithms, cryptanalytic attacks on hash functions and MAC algorithms seek to exploit some property of the algorithm to perform some attack other than an exhaustive search.

#### **Hash Functions**

The hash function takes an input message and partitions it into L fixed-sized blocks of b bits each. If necessary, the final block is padded to b bits. The final block also includes the value of the total length of the input to the hash function(Fig 3.4). The inclusion of the length makes the job of the opponent more difficult.

Either the opponent must find two messages of equal length that hash to the same value or two messages of differing lengths that, together with their length values, hash to the same value.





IV=Initial Value Y<sub>i</sub> = ith input block n=Length of Hash code CV=Changing Variable L=number of input blocks b=Length of input block

#### General structure of secure hash code

The hash algorithm involves repeated use of a compression function, f, that takes two inputs (an n-bit input from the previous step, called the chaining variable, and a b-bit block) and produces an n-bit output.

At the start of hashing, the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value. Often b > n; hence the term compression.

www.EnggTree.com

The hash function can be summarized as follows:

 $CV_0 = IV = initial n-bit value$  $CV_1 = f(CV_{i-1}, Y_{i-1}) \ 1 \le i \le L$ 

 $H(M) = CV_L$ 

Where the input to the hash function is a message M consisting of the blocks  $Y_o$ ,  $Y_1$ ,...,  $Y_{L-1}$ . The structure can be used to produce a secure hash function to operate on a message of any length.

### Message Authentication Codes:

There is much more variety in the structure of MACs than in hash functions, so it is difficult to generalize about the cryptanalysis of MACs.

#### SHA

The algorithm takes as input a message with a maximum length of less than bits and produces as output a 512-bit message digest. The input is processed in 1024-bit blocks. Figure depicts the overall processing of a message to produce a digest.

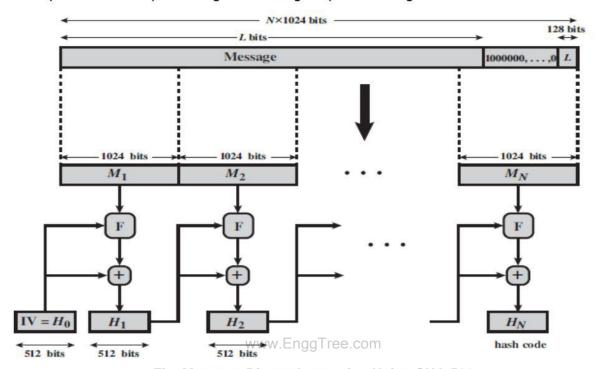


Fig .Message Digest Generation Using SHA-512

The processing consists of the following steps.

#### Step 1: Append padding bits.

The message is padded so that its length is congruent to 896 modulo 1024. Padding is always added, even if the message is already of the desired length. Thus, the number of padding bits is in the range of 1 to 1024. The padding consists of a single 1 bit followed by the necessary number of 0 bits.

#### Step 2: Append length.

A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer (most significant byte first) and contains the length of the original message (before the padding).

The outcome of the first two steps yields a message that is an integer multiple of 1024 bits in length. In Figure 3,8, the expanded message is represented as the sequence of 1024-bit blocks  $M_1, M_2, \ldots, M_N$ , so that the total length of the expanded message is NX1024 bits.

### Step 3: Initialize hash buffer.

A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialized to the following 64-bit integers (hexadecimal values):

```
a = 6A09E667F3BCC908 e = 510E527FADE682D1
b = BB67AE8584CAA73B f = 9B05688C2B3E6C1F
c = 3C6EF372FE94F82B g = 1F83D9ABFB41BD6B
d = A54FF53A5F1D36F1 h = 5BE0CD19137E2179
```

These values are stored in big-endian format, which is the most significant byte of a word in the low-address (leftmost) byte position. These words were obtained by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers.

Step 4: Process message in 1024-bit (128-word) blocks.

The heart of the algorithm (Fig 3.9) is a module that consists of 80 rounds; Each round takes as input the 512-bit buffer value, abcdefgh, and updates the contents of the buffer. At input to the first round, the buffer has the value of the intermediate hash value,  $H_{i-1}$ 

Each round makes use of a 64-bit value W<sub>t</sub>, derived from the current 1024-bit block (M<sub>i</sub>) being processed .These values are derived using a message schedule described subsequently.

Each round also makes use of an additive constant k<sub>t</sub>, where 0<=t<=79 indicates one of the 80 rounds.

The output of the eightieth round is added to the input to the first round (H<sub>i</sub>-1) to produce H<sub>i</sub>. The addition is done independently for each of the eight words in the buffer with each of the corresponding words in Hi-1, using addition modulo 264.

Step 5: Output.

After all N 1024-bit blocks have been processed, the output from the Nth stage is the 512-bit message digest.

The behavior of SHA-1 is summarized 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, defined in step 3

ABCDE<sub>q</sub> = the output of the last round of processing of the ith message block

L = the number of blocks in the message (including padding and length fields)

SUM $_{32}$  = Addition modulo  $2^{32}$  performed separately on each word of the pair of

# Downloaded from EnggTree.com

MD = final message digest value

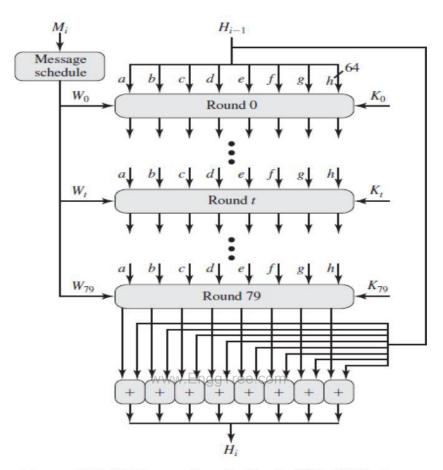


Figure. SHA-512 Processing of a Single 1024-Bit Block

### **SHA-512 Round Function**

Let us look in more detail at the logic in each of the 80 steps of the processing of one 512-bit block. Each round is defined by the following set of equations:

$$\begin{split} T_1 &= h + Ch(e,f,g) + (\sum_1^{512} e) + Wt + Kt) \\ T_2 &= (\sum_0^{512} a) + Maj(a,b,c) \\ h &= g \qquad g = f \qquad f = e \qquad e = d + T_1 \qquad d = c \\ c &= b \qquad b = a \qquad a = T_1 + T_2 \end{split}$$

Where

T = Step number; 
$$0 \le t \le 79$$
  
Ch(e,f,g) = (a AND f)  $\bigoplus$  (NOT e AND g)

# Downloaded from EnggTree.com

The conditional function: If e then f else g (Fig 3.10)

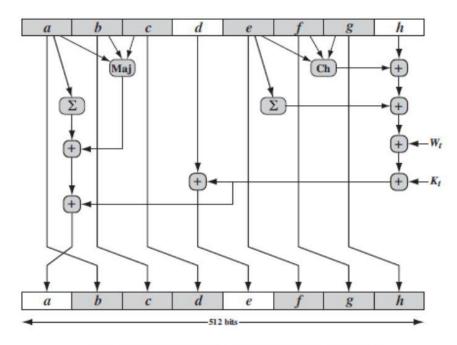


Fig .Elementary SHA Operation(single step)

The functions can be summarized as follows:

www.EnggTree.com

Steps	Function Name	Function Value
0≤t≤9	f1 = f(t,B,C,D)	$(B \land C) \lor (B! \land D)$
20≤t≤39	f2 = f(t,B,C,D)	в ФСФД
40≤t≤59	f3 = f(t,B,C,D)	$(B \land C) \lor (B \land D) \lor (C \land D)$
60≤t≤79	f4 = f(t,B,C,D)	в ФС ФD

The logical operators AND,OR,NOT,XOR, are represented by the symbols ∧ ∨ ! ⊕

Only three different functions are used.

For,  $0 \le t \le 19$  the function is the conditional function.

For  $20 \le t \le 39$  and  $60 \le t \le 79$  the function produces a parity bit.

For A0  $\leq$  t  $\leq$  59 the function is true if two or three of the argument are true.

The following diagram illustrates how the 32bit word values wt are derived from the 512 bit message.

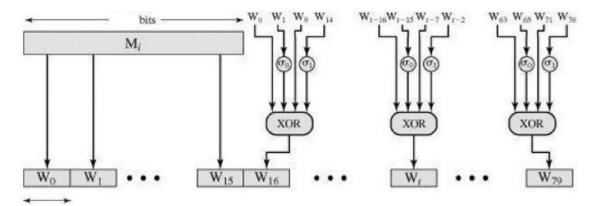


Figure.Creation of 80-word Input Sequence for SHA-512 Processing of Single Block

The first 16 values of wt are taken directly from the 16 words of the current block.the remaining values are defined as follows.

$$W_t = S''(W_{t-16} + W_{t-14} + W_{t-8} + W_{t-3})$$

Thus in the first 16 steps of processing the values of wt is equal to the corresponding word in the message block. For the remaining 64 steps the value of wt consists of the circular left shift by one bit of the XOR of four of the processing values of  $w_t$ .

Both MD5 and RIPEMD-160 uses one of the 16 words of a message block directly as input to each step function only the order of the word is permuted from round to round.

SHA-1 expands the 16 block words to 80 words for use in the compression function.

#### Comparison of SHA-1 and MD5

Because both are derived from MD4,SHA-1 and MD5 are similar to one another.

1. Security against brute - force attacks:

The most important difference is that the SHA-1 digest is 32bits longer than the MD5 digest.

Using a brute force technique the difficulty of producing any message having a given message digest is on the order of  $2^{128}$  operations for MD5 and  $2^{160}$  for SHA-1.

Using brute force technique the difficulty of producing two messages having the same message digest is on the order of  $2^{64}$  operations for MD5 and  $2^{80}$  for SHA-1. Thus SHA-1 is considerably stronger against brute force attacks.

2. Security against cryptanalysis:

MD5 is vulnerable to cryptanalytic attacks.

SHA-1 is not vulnerable to such attacks.

### 3. Speed:

Both algorithms rely on addition module 2<sup>32</sup>, so both do well on 32 bit architecture

SHA-1 involves more steps (80) and must process a 160 bit buffer compared to MD5"s 128 bit buffer.

Thus SHA-1 should execute more slowly than MD5 on the same hardware.

4. Simplicity and compactness:

Both algorithms are simple to describe and simple to implement and do not require large programs or substitution tables.

5. Little endian versus big endian architecture:

MD5 uses a little endian scheme and SHA-1 uses a big endian scheme.

#### **5.EXPLIAN THE HMAC IN DETAILS.**

### **HMAC Design Objectives:**

- To use hash functions that perform well in software and for which code is freely and widely available.
  www.EnggTree.com
- To allow for easy replacement of the embedded hash function in case faster or more secure hash functions are found or required.
- To preserve the original performance of the hash function without incurring a significant degradation.
- > To use and handle keys in a simple way.
- ➤ To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions about the embedded hash function.

The first two objectives are important to the acceptability of HMAC.

HMAC treats the hash function as a "black box." This has two benefits.

First, an existing implementation of a hash function can be used as a module in implementing HMAC. In this way, the bulk of the HMAC code is prepackaged and ready to use without modification.

Second, if it is ever desired to replace a given hash function in an HMAC implementation, remove the existing hash function module and drop in the new module.

#### **HMAC Algorithm:**

Definition of terms used in algorithm(Fig 3.12).

H = embedded hash function (e.g., MD5, SHA-1, RIPEMD-160)

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

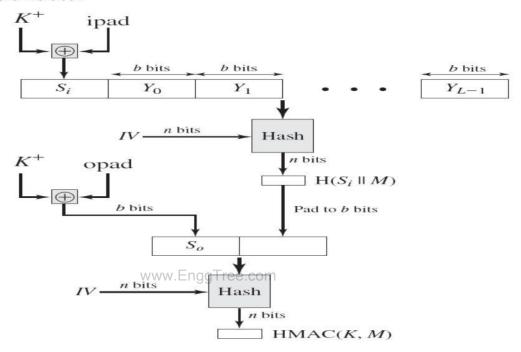


Figure .HMAC Structure

n = length of hash code produced by embedded hash function

K = secret key; recommended length is n; if key length is greater than b, the key is input to the hash function to produce an n-bit key

 $K^+$  = K padded with zeros on the left so that the result is b bits in length

ipad = 00110110 (36 in hexadecimal) repeated b/8 times

opad = 01011100 (5C in hexadecimal) repeated b/8 times

Then HMAC can be expressed as

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

We can describe the algorithm as follows:

- **1.** Append zeros to the left end of to create a -bit string (e.g., if is of length 160 bits and , then will be appended with 44 zeroes).
- 2. XOR (bitwise exclusive-OR) with ipad to produce the b-bit block  $S_i$ .
- 3. Append M to Si.
- 4. Apply H to the stream generated in step 3.
- 5. XOR K+ with opad to produce the b-bit block S<sub>0</sub>.
- Append the hash result from step 4 to S<sub>0</sub>.
- 7. Apply H to the stream generated in step 6 and output the result.

A more efficient implementation is possible. Two quantities are precomputed:

$$f(IV, (K^+ \oplus ipad))$$
  
 $f(IV, (K^+ \oplus opad))$ 

In effect, the precomputed quantities substitute for the initial value (IV) in the hash function. With this implementation (Fig 3.13), only one additional instance of the compression function is added to the processing normally produced by the hash function.

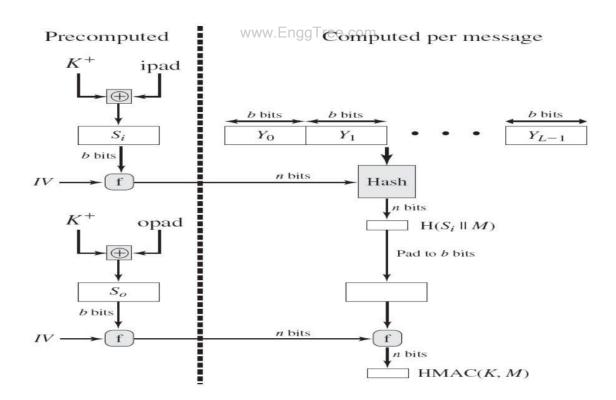


Figure .Efficient Implementation of HMAC

### Security of HMAC

The security of a MAC function is generally 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-tag pairs created with the same key.

In essence, it is proved in that for a given level of effort (time, message-tag pairs) on messages generated by a legitimate user and seen by the attacker, the probability of successful attack on HMAC is equivalent to one of the following attacks on the embedded hash function.

- **1.** The attacker is able to compute an output of the compression function even with an that is random, secret, and unknown to the attacker.
- 2. The attacker finds collisions in the hash function even when the V is random and secret.

In the first attack, we can view the compression function as equivalent to the hash function applied to a message consisting of a single b bit block. For this attack, the IV of the hash function is replaced by a secret, random value of bits. An attack on this hash function requires either a brute-force attack on the key, which is a level of effort on the order of  $2^n$ , or a birthday attack.

In the second attack, the attacker is looking for two messages M & M"and that produce the same hash:  $H(M) = H(M^s)$ .

### **8. EXPLIAN THE CMAC IN DETAIL.** www.EnggTree.com

Only messages of one fixed length of mn bits are processed, where n is the cipher block size and m is a fixed positive integer. a simple example, notice that given the CBC MAC of a one-block message X, say T = MAC(K, X), the adversary immediately knows the CBC MAC for the two block message  $X \parallel (X \oplus T)$  since this is once again T.

Black and Rogaway [BLAC00] demonstrated that this limitation could be overcome using three keys: one key K of length k to be used at each step of the cipher block chaining and two keys of length k, where k is the cipher block length.

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

First, let us define the operation of CMAC when the message is an integer multiple n of the cipher block length b. For AES, b = 128, and for triple DES, b = 64. The message is divided into n blocks (M1, M2,c, Mn). The algorithm makes use of a k-bit encryption key K and a b-bit constant, K1. For AES, the key size k is 128, 192, or 256 bits; for triple DES, the key size is 112 or 168 bits. CMAC is calculated as follows

```
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
\vdots
C_{n} = E(K, [M_{n} \oplus C_{n-1} \oplus K_{1}])
T = MSB_{Tlen}(C_{n})
```

where

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

Tlen = bit length of T

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

The CMAC operation(Fig 3.14) then proceeds as before, except that a different b-bit key K2 is used instead of K1.

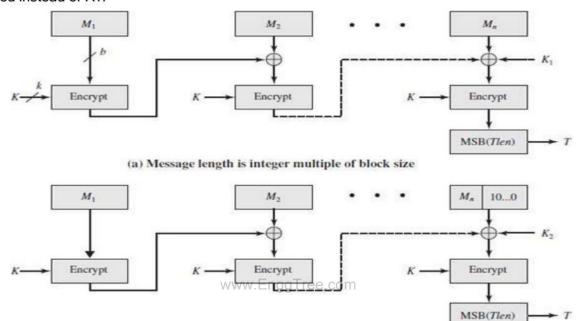


Fig .Cipher-based Message Authentication Code

The two *b*-bit keys are derived from the *k*-bit encryption key as follows.

$$L = E(K, 0^b)$$

$$K_1 = L \cdot x$$

$$K_2 = L \cdot x^2 = (L \cdot x) \cdot x$$

where multiplication (#) is done in the finite field GF(2b) and x and x2 are first and second-order polynomials that are elements of GF(2b). Thus, the binary representation of x consists of b - 2 zeros followed by 10; the binary representation of x2 consists of b - 3 zeros followed by 100.

#### 9.EXPLIAN THE DIGITAL SIGNATURE AND AUTHENTICATION PROTOCOLS

#### Digital Signature Requirements

Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other.

Disputes created by message authentication are:

Creation of fraud message.

Deny the sending of message

For example, suppose that John sends an authenticated message to Mary, the following disputes that could arise:

- 1 Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.
- 2. John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

#### Properties of digital signature:

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

#### Requirements for a digital signature:

- ➤ The signature must be a bit pattern that depends on the message being signed.
- > The signature must use some information unique to the sender, to prevent both forgery and denial.
- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature.
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be practical to retain a copy of the digital signature in storage.

#### **Direct Digital Signature**

The term **direct digital signature** refers to a digital signature scheme that involves only the communicating parties (source, destination). It is assumed that the destination knows the public key of the source.

Confidentiality can be provided by encrypting the entire message plus signature with a shared secret key (symmetric encryption). Note that it is important to perform the signature function first and then an outer confidentiality function. In case of dispute, some third party must view the message and its signature.

If the signature is calculated on an encrypted message, then the third party also needs access to the decryption key to read the original message. However, if the signature is the inner operation, then the recipient can store the plaintext message and its signature for later use in dispute resolution.

The validity of the scheme just described depends on the security of the sender"s private key.

#### Weakness of Direct Digital Signature:

- If a sender later wishes to deny sending a particular message, the sender can claim that the private key was lost or stolen and that someone else forged his or her signature.
- Another threat is that some private key might actually be stolen from X at time T. The opponent can then send a message signed with X"s signature and stamped with a time before or equal to T.

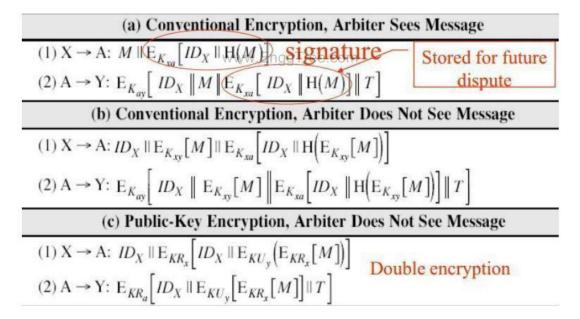
#### **Arbitrated Digital Signatures**

The problem associated with the Direct digital signature can be overcome by using arbitrated schemes.

In the arbitrated scheme, the entire signed message from the sender goes to the arbiter A. The arbiter subjects the message and signature to a number of tests to check the origin and control. The date and time is attached to the message. This indicates that the digital signature has been verified and is satisfied. The message is then transmitted to the receiver.

### Requirement of the arbiter:

As the arbiter plays a sensitive and crucial role, it should be a trusted third party.



Notation:X = Sender Y = Recipient A = Arbiter M=Message T=Timestamp

#### Scheme 1: Conventional encryption, Arbiter sees the message:

The sender X and arbiter A share the master key  $K_{ax}$  the receiver y and the arbiter A share the master key  $K_{ay}$ 

When X wants to send a message M to Y, construct a message computes the hash value H(M). This hash is encrypted using symmetric encryption with the key Kax which acts as signature. The message along with the signature is transmitted to A.

At A, it decrypts the signature and checks the hash value to validate the message. A transmit the message to Y, encrypted with  $K_{ay}$ . Y decrypt to extract the message and signature. Disadvantage:

Eaves dropper can read the message as there is no confidentiality.

#### Scheme 2: Conventional encryption, Arbiter does not see the message:

- K<sub>ax</sub> and K<sub>ay</sub> are the master keys.
- K<sub>xy</sub> is the key shared between the X and Y
- When x wants to transmit a message to Y, the packet goes to arbiter.
- ➤ The same procedure as that of I scheme is used X transmit an identifier, a copy of the message encrypted with K<sub>xy</sub> and a signature to A.
- The signature is the hash of the message encrypted with Kxa
- A decrypt the signature, and checks the hash value to validate the message.
- ➤ A cannot read the message, A attaches to it the time stamps, encrypt with K<sub>xa</sub> and transmit to Y.

Attack: The arbiter can join with an attacker and deny a message with sender's signature.

### Scheme 2: Public key encryption, Arbiter does not see the message:

This method uses the public key cryptography which gives authentication and digital signature.

The doubly encrypted message is concatenated with ID<sub>x</sub> and sent to arbiter.

- A can decrypt the outer encryption to ensure that the message has come from X.
- A then transmit the message with ID<sub>x</sub> and time stamp.

#### Advantages:

- No information is shared among parties before communication, hence fraud is avoided.
- No incorrectly dated message can be sent.

### Disadvantages:

The complex public key algorithm is to be twice for encryption and twice for decryption.

#### **Authentication Protocols**

Authentication Protocols used to convince parties of each other"s identity and to exchange session keys.

#### **Mutual Authentication**

An important application area is that of mutual authentication protocols. Such protocols enable communicating parties to satisfy themselves mutually about each other"s identity and to exchange session keys.

#### Key issues are

- confidentiality to protect session keys and prevent masqueraded and compromised
- > timeliness to prevent replay attacks

#### **Replay Attacks**

Where a valid signed message is copied and later resent

Simple replay

The opponent simply copies the message and replays it later.

Repetition that can be logged

The opponent replay a time stamped message within a valid time window.

Repetition that cannot be detected

The attacker would have suppressed the original message from the receiver. Only the replay message alone arrives.

Backward replay without modification

This replay back to the sender itself. This is possible if the sender cannot easily recognize the difference between the message sent and the message received based on the content.

#### Countermeasures include

One approach to coping with replay attacks is to attach a sequence number to each message used in an authentication exchange. A new message is accepted only if its sequence number is in the proper order.

The difficulty with this approach is that it requires each party to keep track of the last sequence number for each claimant it has dealt with. Because of this overhead, sequence numbers are generally not used for authentication and key exchange. Instead, one of the following two general approaches is used:

- Timestamps: Party A accepts a message as fresh only if the message contains a Timestamp that is close enough to A"s knowledge of current time. This approach requires that clocks among the various participants be synchronized.
- Challenge/response: Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value.

### **Using Symmetric Encryption**

- Use a two-level hierarchy of keys
- Usually with a trusted Key Distribution Center (KDC)
  - Each party shares own master key with KDC
  - KDC generates session keys used for connections between parties
  - Master keys used to distribute the session ke

#### **Needham-Schroeder Protocol**

- Original third-party key distribution protocol
- For session between A and B mediated by KDC
- Protocol overview is:
  - 1. A $\rightarrow$ KDC:  $ID_A||ID_B||N_1$
  - 2. KDC $\rightarrow$ A: EK<sub>a</sub>[K<sub>s</sub> || ID<sub>B</sub>||  $N_1$ || EK<sub>b</sub>[K<sub>s</sub>||ID<sub>A</sub>]]
  - 3.  $A \rightarrow B$ :  $EK_b[K_s||ID_A]$
  - 4. B→A: *EK*<sub>s</sub>[*N*<sub>2</sub>]
  - 5.  $A \rightarrow B$ :  $EK_s[f(N_2)]$
- Step 1: A to KDC ,transmit the id of source and destination and a nonce N<sub>1</sub> as a request.
- Step 2: A securely acquires the session key in step 2 and a packet to B encrypted with  $\mathsf{EK}_b$  is also received from KDC.
- Step 3: A transmit to B the message it got from KDC.
- Step 4: As a hand shake, B encrypts a new nonce N2 and transmit to A with Ks.
- Step 5: As a hand shake, A encrypt the function of N<sub>2</sub> with K<sub>s</sub>
- Step 4 and Step 5 as used as hand shake and prevent the reply attacks.

#### Attacks:

- Used to securely distribute a new session key for communications between A &B
- But is vulnerable to a replay attack if an old session key has been Compromised
  - Then message 3 can be resent convincing B that is communicating With A
- Modifications to address this require:
  - Timestamps
  - Using an extra nonce

#### DenningProtocol:

To overcome the above weakness by a modification to the Needham/Schroeder protocol that includes the addition of a timestamp to steps 2 and 3.

A
$$\rightarrow$$
KDC:  $ID_A||ID_B$   
KDC  $\rightarrow$  A: E( $K_a$ , [ $K_s$ ||  $ID_B$ ||  $T$ || E( $K_b$ , [ $K_s$ || $ID_A$ ||  $T$ ])])  
A $\rightarrow$ B: E( $K_b$ , [ $K_s$ ||  $ID_A$ || $T$ ])  
B $\rightarrow$ A: E( $K_s$ , N1)  
A $\rightarrow$  B: E( $K_s$ , f( $N$ 1))

T is a timestamp that assures A and B that the session key has only just been generated. Thus, both A and B know that the key distribution is a fresh exchange. A and B can verify timeliness by checking that

$$I Clock - TI < \Delta t1 + \Delta t2$$

The **Denning protocol** seems to provide an increased degree of security compared to the Needham/Schroeder protocol. However, a new concern is raised: namely, that this new scheme requires reliance on clocks that are synchronized throughout the network.

### suppress-replay attacks:

The problem occurs when a sender s clock is ahead of the intended recipient clock. In this case, an opponent can intercept a message from the sender and replay it later when the timestamp in the message becomes current at the recipient site. This replay could cause unexpected results.

#### Method to overcome:

One way to counter suppress-replay attacks is to enforce the requirement that parties regularly check their clocks against the KDC"s clock. The other alternative, which avoids the need for clock synchronization, is to rely on handshaking protocols using nonce.

This latter alternative is not vulnerable to a suppress-replay attack, because the nonce the recipient will choose in the future are unpredictable to the sender.

An attempt is made to respond to the concerns about suppress replay attacks and at the same time fix the problems in the Needham/Schroeder protocol.

### The protocol is

- 1. A:  $\rightarrow$ B:  $ID_A||N_a$  www.EnggTree.com
- 2. B:  $\rightarrow$  KDC:  $ID_B||N_b||E(K_b, [ID_A||N_a||T_b])$
- 3. KDC:  $\rightarrow$  A: E(K<sub>a</sub>, [ID<sub>B</sub> f fN<sub>a</sub>f fKs f fT<sub>b</sub>]) " E(Kb , [ID<sub>A</sub>, Ks " T<sub>b</sub>]) " N<sub>b</sub>
- 4. A:  $\rightarrow$ B: Kb, [ID<sub>A</sub>, K<sub>s</sub>, T<sub>b</sub>]), E(K<sub>s</sub>, N<sub>b</sub>)
- 1. A initiates the authentication exchange by generating a nonce,  $N_a$ , and sending that plus its identifier to B in plaintext. This nonce  $N_a$  will be returned to A in an encrypted message that includes the session key, assuring A of its timeliness.
- 2. B alerts the KDC that a session key is needed. Its message to the KDC includes its identifier and a nonce, . This nonce will be returned to B in an encrypted message that includes the session key, assuring B of its timeliness.B"s message to the KDC also includes a block encrypted with the secret key shared by B and the KDC. This block is used to instruct the KDC to issue credentials to A; the block specifies the intended recipient of the credentials, a suggested expiration time for the credentials, and the nonce received from A.
- 3. The KDC passes on to A B"s nonce and a block encrypted with the secret key that B shares with the KDC. The block serves as a "ticket" that can be used by A for subsequent authentications, as will be seen. The KDC also sends to A block encrypted with the secret key shared by A and the KDC. This block verifies that B has received A"s initial message (  $ID_B$ ) and that this is a timely message and not a replay ( $ID_B$ ), and it provides A with a session key ( $ID_B$ ) and the time limit on its use ( $ID_B$ ).
- 4. A transmits the ticket to B, together with the B"s nonce, the latter encrypted with the session key. The ticket provides B with the secret key that is used to decrypt  $E(K_S, N_b)$  to recover the

nonce. The fact that B"s nonce is encrypted with the session key authenticates that the message came from A and is not a replay.

### **Using Public-Key Encryption**

- · Have a range of approaches based on the use of public-key encryption
- Need to ensure have correct public keys for other parties
- Using a central authentication server (AS)
- Various protocols exist using timestamps or nonces

### **Denning AS Protocol**

- · Denning presented the following:
  - 1. A→AS: ID<sub>A</sub>|| ID<sub>B</sub>
  - 2. AS $\rightarrow$ A:  $E_{KRas}[ID_A||KU_a||T]||EKRas[ID_B||KU_b||T]$
  - 3.  $A \rightarrow B$ :  $E_{KRas}[ID_A||KU_a||T] || EKRas[ID_B||KU_b||T] || EKU_b[E_{KRas}[Ks||T]]$
- Note session key is chosen by A, hence AS need not be trusted to protect it
- · timestamps prevent replay but require synchronized clocks

Another approach proposed by woo and lam makes use of nonce.

- 1. A→KDC: ID<sub>A</sub>|| ID<sub>B</sub>
- 2. KDC→A: E<sub>KRauth</sub>[ID<sub>B</sub> II K<sub>Ub</sub>]
- 3. a→b: EK<sub>Ub</sub>[N<sub>a</sub> II ID<sub>A</sub>]
- www.EnggTree.com
- B→KDC: ID<sub>B</sub> II ID<sub>A</sub> II EK<sub>Uauth</sub> [N<sub>a</sub>]
- KDC→B: E<sub>KRauth</sub> [IDA II KUa] II E<sub>KUb</sub>[E<sub>KRauth</sub>[N<sub>a</sub> II K<sub>s</sub> II ID<sub>B</sub>]]
- B→A: E<sub>KUa</sub>[E<sub>KRauth</sub>[N<sub>a</sub> II K<sub>s</sub> II ID<sub>B</sub>] II N<sub>b</sub>]
- A→B: E<sub>Ks</sub>[N<sub>b</sub>]
- Step 1: A informs the KDC of its intention to establish a secure connection with B.
- Step 2: The KDC returns to A a copy of B"s public key certificate.
- Step 3: A informs B of its desire to communicate and sends a nonce Na.
- Step 4: B asks the KDC for A"s public key certificate and request a session key.;B includes A"s nonce so that the KDC can stamp the session key with that nonce.The nonce is protected using the KDC"s public key.
- Step 5: The KDC returns to B a copy of A"s public key certificate, plus the information[Na, Ks, IDB].
- Step 6: The triple  $[N_a, K_s, ID_B]$ , still encrypted with the KDC"s private key, is relayed to A, together with a nonce  $N_b$  generated by B.

All the foregoing are encrypted using A"s public key. A retrieves the session key  $K_s$  and uses it to encrypt  $N_b$  and return it to B.

Step 7: Assures B of A"s knowledge of the session key.

### **One-Way Authentication**

- Required when sender & receiver are not in communications at same time (eg. Email)
- Have header in clear so can be delivered by email system
- May want contents of body protected & sender authenticated

### **Using Symmetric Encryption**

- · can refine use of KDC but can"t have final exchange of nonce
  - 1. A→KDC: ID<sub>A</sub>|| ID<sub>B</sub>|| N<sub>1</sub>
  - 2. KDC $\rightarrow$ A: EK<sub>a</sub>[K<sub>s</sub> ||  $ID_B$ ||  $N_1$ || E $K_b$ [K<sub>s</sub>|| $ID_A$ ]]
  - 3.  $A \rightarrow B$ :  $EK_b[K_s||ID_A] || EKs[M]$
- Does not protect against replays
  - > could rely on timestamp in message, though email delays make this problematic

### **Public-Key Approaches**

If confidentiality is major concern, can use:

$$A \rightarrow B$$
:  $EKU_b[K_s] || EK_s[M]$ 

In this case, the message is encrypted with a one-time secret key. A also encrypts this one-time key with B"s public key. Only B will be able to use the corresponding private key to recover the one-time key and then use that key to decrypt the message. This scheme is more efficient than simply encrypting the entire message with B"s public key.

If authentication needed use a digital signature with a digital certificate:

$$A \rightarrow B$$
:  $M$ , II  $E_{KRa}(H(M))$ 

This method guarantees that A cannot later deny having sent the message. However, this technique is open to another kind of fraud. Bob composes a message to his boss Alice that contains an idea that will save the company money. He appends his digital signature and sends it into the e-mail system.

Eventually, the message will get delivered to Alice"s mailbox. But suppose that Max has heard of Bob"s idea and gains access to the mail queue before delivery. He finds Bob"s message, strips off his signature, appends his, and requeues the message to be delivered to Alice. Max gets credit for Bob"s idea.

To counter such a scheme, both the message and signature can be encrypted with the recipient spublic key:

$$A \rightarrow B: E_{KUb}, [M || E_{KRa}, H(M)]$$

The latter two schemes require that B know A"s public key and be convinced that it is timely. An effective way to provide this assurance is the digital certificate

$$A\rightarrow B: M \parallel EKR_a [H(M)] \parallel EKR_{as} \parallel T \parallel \parallel D_A \parallel KU_a]$$

In addition to the message,A sends B the signature encrypted with A"s private key and A"s certificate encrypted with the private key of the authentication server. The recipient of the message first uses the certificate to obtain the sender"s public key and verify that it is authentic and then uses the public key to verify the message itself.

#### 4.7.DSS

#### The DSS Approach

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

### 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. The recipient takes the message and produces a hash code. The recipient also decrypts the signature using the sender spublic key.

#### www.EnggTree.com

If the calculated hash code matches the decrypted signature, the signature is accepted as valid. Because only the sender knows the private key, only the sender could have produced a valid signature.

#### DSS approach

The DSS approach also makes use of a hash function. The hash code is provided as input to a signature function along with a random number generated for this particular signature.

The signature function also depends on the sender"s private key (PRa) and a set of parameters known to a group of communicating principals. We can consider this set to constitute a global public key ( $PU_G$ ). The result is a signature consisting of two components, labeled s and  $r(fig\ 3.15)$ .

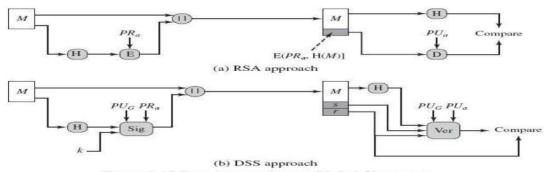


Figure 3.15 Two Approaches to Digital Signatures

At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function. The verification function also depends on the global public key as well as the sender spublic key, which is paired with the sender sprivate key.

The output of the verification function is a value that is equal to the signature component if the signature is valid. The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature.

### The Digital Signature Algorithm:

There are three parameters that are public and can be common to a group of users.

- A 160-bit prime number q is chosen.
- ➤ Next, a prime number p is selected with a length between 512 and 1024 bits such that q divides (p 1).
- Finally, g is chosen to be of the form  $h^{(p-1)/q} \mod p$ , where h is an integer between 1 and (p-1).

With these numbers in hand, each user selects a private key and generates a public key. The private key x must be a number from 1 to (q-1) and should be chosen randomly. T

The public key is calculated from the private key as  $y = g^x \mod p$ . The calculation of given(Fig 3.16) is relatively straightforward. However, given the public key y, it is believed to be computationally infeasible to determine x, which is the discrete logarithm of y to the base g, mod p.

At the receiving end, verification is performed using the formulas. The receiver generates a quantity v that is a function of the public key components, the sender spublic key, and the hash code of the incoming message. If this quantity matches the component of the signature, then the signature is validated.

### www.EnggTree.com

```
Global Public-Key Components
                                                                                              Signing
p prime number where 2^{L-1} 
                                                                          = (g^k \bmod p) \bmod q
   for 512 \le L \le 1024 and L a multiple of 64;
                                                                        s = [k^{-1} (H(M) + xr)] \bmod q
   i.e., bit length of between 512 and 1024 bits
   in increments of 64 bits
                                                                        Signature = (r, s)
  prime divisor of (p-1), where 2^{159} < q < 2^{160}; i.e., bit length of 160 bits
                                                                                             Verifying
   =h^{(p-1)/q}\bmod p,
                                                                        w = (s')^{-1} \bmod q
   where h is any integer with 1 < h < (p-1) such that h^{(p-1)/q} \mod p > 1
                                                                        u_1 = [H(M')w] \mod q
                                                                        u_2 = (r')w \mod q
                    User's Private Key
                                                                        v = [(g^{u1} y^{u2}) \bmod p] \bmod q
   random or pseudorandom integer with 0 < x < q
                                                                        TEST: v = r'
                     User's Public Key
                                                                                      = message to be signed
   = g^x \mod p
                                                                           H(M)
                                                                                     = hash of M using SHA-1
                                                                           M', r', s' = \text{received versions of } M, r, s
           User's Per-Message Secret Number
    = random or pseudorandom integer with 0 < k < q
```

Figure.The Digital Signature Algorithm (DSA)

The value r does not depend on the message at all. Instead, r is a function of k and the three global public-key components.

The multiplicative inverse of k (mod q) is passed to a function that also has as inputs the message hash code and the user"s private key.

The structure of this function is such that the receiver can recover using the incoming message and signature, the public key of the user, and the global public key. Given the difficulty of taking discrete logarithms, it is infeasible for an opponent to recover k from r to recover x from s.

The only computationally demanding task in signature generation is the exponential calculation  $g^k$ mod p. Because this value does not depend on the message to be signed, it can be computed ahead of time.

Selects a private key and generates a public key. The private key x must be a number from 1 to (q1) and should be chosen randomly. The public key is calculated from the private key as  $y = g^x \mod p$ .

To create a signature, a user calculates two quantities, r and s, that are functions of the public key components (p, q, g), the user's private key (x), the hash code of the message, H(M), and an additional integer k that should be generated randomly and be unique for each signing.

At the receiving end, verification is performed using the formulas. 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 component of the signature, then the signature is validated (Fig ).

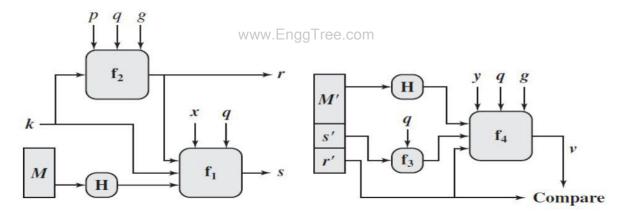


Figure DSS Signing and Verifying

#### 11.EXPLIAN THE AUTHENTICATION APPLICATIONS

One of the key aspects of cryptography and network security is authentication. It helps to establish trust by identifying a particular user or a system. There are many ways to authenticate a user. Traditionally, user ids and passwords have been used.

### 1. Authentication Requirements

During communication across networks, following attacks can be identified.

- 1. **Disclosure:** Releases 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 fraudulent source.
- Content modification: Changes to the content of the message, including insertion deletion, transposition and modification.
- **5. Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion and reordering.
- 6. Timing modification: Delay or replay of messages.
- **7. Source repudiation:** Denial of transmission of message by source.
- 8. Destination repudiation: Denial of transmission of message by destination.

The security measures for the above mentioned attacks are as follows

For 1,2 - Message Confidentiality

3,4,5,6 - Message Authentication

7 - Digital Signatures

8 - Digital signature with protocol designed to counter the attack

#### 2. Authentication Functions

Any message authentication or digital signature mechanism can be viewed as having fundamentally two levels.

 Lower level: Some function that produces an authenticator: a value to be used to authenticate a message.

Higher Level: Lower layer functions are used to create a protocol that enables a receiver to verify the authenticity of message

The different types of functions that may be used to produce an authenticator are as follows:

- Message encryption: The cipher text of the entire message serves as its authenticator.
- 2. Message Authentication Code (MAC): A public function of the message and a secret key that produces a fixed length value serves as the authenticator.
- **3. Hash function:** A public function that maps a message of any length into a fixed length hash value, which serves as the authenticator.

### 12.EXPLAIN THE KERBEROS IN DETAIL.

Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. Kerberos relies exclusively on conventional encryption, making no use of public-key encryption.

#### Motivation

A distributed architecture consists of dedicated user workstations (clients) and distributed or centralized servers. In this environment, there are three approaches to security:

- Rely on each individual client workstation to assure the identity of its user or users and rely on each server to enforce a security policy based on user identification (ID).
- Require that client systems authenticate themselves to servers, but trust the client system concerning the identity of its user.
- Require the user to prove his or her identity for each service invoked. Also require that servers prove their identity to clients.

The following are the **requirements for Kerberos**:

- **Secure:** A network eavesdropper should not be able to obtain the necessary information to impersonate a user. More generally, Kerberos should be strong enough that a potential opponent does not find it to be the weak link.
- Reliable: For all services that rely on Kerberos for access control, lack of availability
  of the Kerberos service means lack of availability of the supported services. Hence,
  Kerberos should be highly reliable and should employ distributed server architecture,
  with one system able to back up another.
- Transparent: Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.
- Scalable: The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

To support these requirements, the overall scheme of Kerberos is that of a trusted thirdparty authentication service that uses a protocol based on Needham and Schroeder.

It is trusted in the sense that clients and servers trust Kerberos to mediate their mutual authentication. Assuming the Kerberos protocol is well designed, and then the authentication service is secure if the Kerberos server itself is secure.

Two versions of Kerberos are in common use. Version 4 and Version 5

#### **Kerberos Version 4**

Version 4 of Kerberos makes use of DES, in a rather elaborate protocol, to provide the authentication service

#### Simple Authentication Dialogue

In an unprotected network environment, any client can apply to any server for service. The obvious security risk is that of impersonation. To counter this threat, servers must be able to confirm the identities of clients who request service. But in an open environment, this places a substantial burden on each server.

An alternative is to use an authentication server (AS) that knows the passwords of all users and stores these in a centralized database. In addition, the AS shares a unique secret key with each server. The simple authentication dialogue is as follows:

```
1. C >> AS: IDc||Pc||IDv
```

2. AS >> C: Ticket

3. C >> V: IDc||Ticket www.EnggTree.com

Ticket= EK<sub>v</sub>(ID<sub>c</sub>||ADc||ID<sub>v</sub>)

C : Client,

AS : Authentication Server,

V : Server, ID<sub>c</sub>: ID of the client,

Pc : Password of the client,

AD<sub>c</sub>: Address of client, ID<sub>v</sub>: ID of the server,

K<sub>v</sub>: secret key shared by AS and V,

: concatenation.

### More Secure Authentication Dialogue

There are two major problems associated with the previous approach:

- Plaintext transmission of the password.
- Each time a user has to enter the password.

To solve these problems, we introduce a scheme for avoiding plaintext passwords, and a new server, known as ticket granting server (TGS). The hypothetical scenario is as follows:

### Once per user logon session:-

```
1. C \Rightarrow AS: |D_c|||D_{tgs}|
```

2. AS >> C: Ekc (Tickettgs)

#### Once per type of service:

3. C >> TGS:  $ID_{c|}IID_{v|}|Ticket_{tgs}$ 

4. TGS >> C: ticket<sub>v</sub>

#### Once per service session:

5. C >> V: IDc|| Ticketv

Ticket<sub>tgs</sub>= Ekt<sub>gs</sub>( $ID_c||AD_c||IDt_{gs}||TS_1||Lifetime_1$ ) Ticket<sub>v</sub>= Ek<sub>v</sub>( $ID_c||AD_c||ID_v||TS2||Lifetime_2$ )

C: Client, AS: Authentication Server, V: Server,

IDc: ID of the client, Pc:Password of the client, ADc: Address of client,

IDv: ID of the server, Kv: secret key shared by AS and V,

|| : concatenation, | IDtgs: ID of the TGS server, TS1, TS2: time stamps, | lifetime:

lifetime of the ticket.

The new service, TGS, issues tickets to users who have been authenticated to AS. Thus, the user first requests a ticket-granting ticket (Ticket<sub>tgs</sub>) from the AS. The client module in the user workstation saves this ticket.

Each time the user requires access to a new service, the client applies to the TGS, using the ticket to authenticate itself. The TGS then grants a ticket for the particular service. The client saves each service-granting ticket and uses it to authenticate its user to a server each time a particular service is requested.

Let us look at the details of this scheme:  ${\sf EnggTree.com}$ 

- The client requests a ticket-granting ticket on behalf of the user by sending its user's ID and password to the AS, together with the TGS ID, indicating a request to use the TGS service
- 2. The AS responds with a ticket that is encrypted with a key that is derived from the user's password.

When this response arrives at the client, the client prompts the user for his or her password, generates the key, and attempts to decrypt the incoming message.

If the correct password is supplied, the ticket is successfully recovered.

Because only the correct user should know the password, only the correct user can recover the ticket. Thus, we have used the password to obtain credentials from Kerberos without having to transmit the password in plaintext. Now that the client has a ticket-granting ticket, access to any server can be obtained with steps 3 and 4:

- The client requests a service-granting ticket on behalf of the user. For this purpose, the client transmits a message to the TGS containing the user's ID, the ID of the desired service, and the ticket-granting ticket
- 4. The TGS decrypts the incoming ticket and verifies the success of the decryption by the presence of its ID. It checks to make sure that the lifetime has not expired. Then it compares the user ID and network address with the incoming information to authenticate the user. If the user is permitted access to the server V, the TGS issues a ticket to grant access to the requested service.

The service-granting ticket has the same structure as the ticket-granting ticket. Indeed, because the TGS is a server, we would expect that the same elements are needed to authenticate a client to the TGS and to authenticate a client to an application server.

Again, the ticket contains a timestamp and lifetime. If the user wants access to the same service at a later time, the client can simply use the previously acquired service-granting ticket and need not bother the user for a password.

Note that the ticket is encrypted with a secret key  $(K_{\nu})$  known only to the TGS and the server, preventing alteration.

Finally, with a particular service-granting ticket, the client can gain access to the corresponding service with step 5:

The client requests access to a service on behalf of the user. For this purpose, the client transmits a message to the server containing the user's ID and the service-granting ticket. The server authenticates by using the contents of the ticket.

This new scenario satisfies the two requirements of only one password query per user session and protection of the user password.

### Kerberos V4 Authentication Dialogue Message Exchange

Two additional problems remain in the more secure authentication dialogue:

- Lifetime associated with the ticket granting ticket. If the lifetime is very short, then
  the user will be repeatedly asked for a password. If the lifetime is long, then the
  opponent has the greater opportunity for replay.
- Requirement for the servers to authenticate themselves to users.

The actual Kerberos protocol version 4 is as follows

- A basic third-party authentication scheme
- Have an Authentication Server (AS)
  - o Users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- Have a Ticket Granting
  - Users subsequently request access to other services from TGS on basis of users TGT

### (a) Authentication service exchange: to obtain ticket granting ticket

(1)  $C \rightarrow AS : ID_C II ID_{tgs} II TS_1$ 

(2) AS  $\rightarrow$  C : EKc [  $K_{c,tgs}$  II  $ID_{tgs}$  II  $TS_2$  II Lifetime<sub>2</sub> II Ticket<sub>tgs</sub>]

(b) Ticket-Granting Service Exchange: to obtain service-granting ticket

# Downloaded from EnggTree.com

#### **Kerberos 4 Overview**

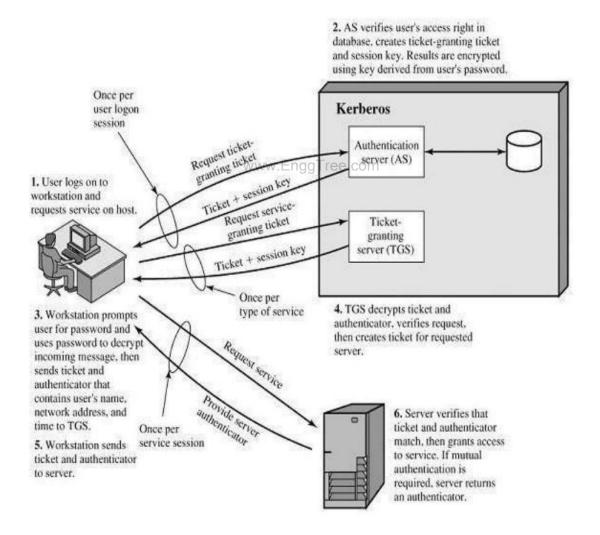


Fig 4.1 Overview of Kerberos 4

### Kerberos Realms and Multiple Kerberi

A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires the following:

- 4. The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.
- The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.

Such an environment is referred to as a Kerberos realm

The concept of realm can be explained as follows.

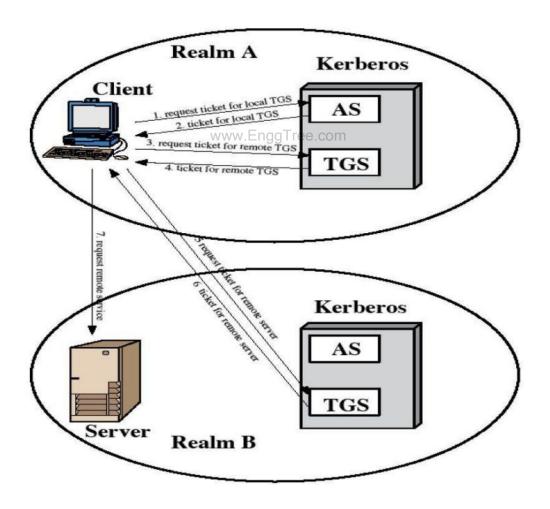


Fig .Request for service in another Realm

A Kerberos realm is a set of managed nodes that share the same Kerberos database. The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room. A read-only copy of the Kerberos database might also reside on other Kerberos computer systems.

However, all changes to the database must be made on the master computer system. Changing or accessing the contents of a Kerberos database requires the Kerberos master password. A related concept is that of a Kerberos principal, which is a service or user that is known to the Kerberos system.

Each Kerberos principal is identified by its principal name. Principal names consist of three parts: a service or user name, an instance name, and a realm name. Networks of clients and servers under different administrative organizations typically constitute different realms.

That is, it generally is not practical, or does not conform to administrative policy, to have users and servers in one administrative domain registered with a Kerberos server elsewhere.

However, users in one realm may need access to servers in other realms, and some servers may be willing to provide service to users from other realms, provided that those users are authenticated.

Kerberos provides a mechanism for supporting such inter realm authentication. For two realms to support inter realm authentication, a third requirement is added:

**6.** The Kerberos server in each interoperating realm shares a secret key with the server in the other realm. The two Kerberos servers are registered with each other.

The scheme requires that the Kerberos server in one realm trust the Kerberos server in the other realm to authenticate its users. Furthermore, the participating servers in the second realm must also be willing to trust the Kerberos server in the first realm.

#### The details of the exchanges illustrated in Fig 2are as follows:

 $C \to AS \qquad \quad :ID_C \ II \ ID_{tgs} \ II \ TS_1$ 

 $AS \rightarrow C$  :  $EK_c[K_{c,tgs} \text{ ii } ID_{tgs} \text{ II } TS_2 \text{ II } Lifetime_2 \text{ II } Ticket_{tgs}$ 

C → TGS :ID<sub>tqsrem</sub> II Ticket<sub>tqs</sub> II Authenticator<sub>c</sub>

 $TGS \rightarrow C$  :E  $K_{c.tasl}K_{c.tasrem} II ID_{tasrem} II TS_4 II Ticket_{tasrem}$ 

C→ TGS rem :IDvrem II Tickettgsrem II Authenticatorc

TGS  $_{rem} \rightarrow C : EK_{c,tgsrem} [K_{c,vrem} | II | ID_{vrem} | II | TS_6 | II | Ticket_{vrem} :$ 

C → V<sub>rem</sub> Ticket<sub>vrem II</sub>Authenticator<sub>c</sub>

### Differences between Versions 4 and 5

Version 5 is intended to address the limitations of version 4 in two areas: environmental shortcomings and technical deficiencies.

### Environmental shortcomings:

#### 7. Encryption system dependence:

Version 4 requires the use of DES. In version 5, ciphertext is tagged with an encryption type identifier so that any encryption technique may be used.

#### 8. Internet protocol dependence:

Version 4 requires the use of Internet Protocol (IP) addresses. Version 5 network addresses are tagged with type and length, allowing any network address type to be used.

### 9. Message byte ordering:

In version 4, the sender of a message employs a byte ordering of its own choosing and tags the message to indicate least significant byte in lowest address In version 5, all message structures are defined using Abstract Syntax Notation One (ASN.1) and Basic Encoding Rules (BER), which provide an unambiguous byte ordering.

#### 10. Ticket lifetime:

Lifetime values in version 4 are encoded in an 8-bit quantity in units of five minutes. In version 5, tickets include an explicit start time and end time, allowing tickets with arbitrary lifetimes.

### 11. Authentication forwarding:

Version 4 does not allow credentials issued to one client to be forwarded to some other host and used by some other client. Version 5 provides this capability.

### Technical deficiencies in the version 4 protocol:

www.EnggTree.com

- Double encryption
- PCBC encryption
- Session keys
- Password attacks

#### The Version 5 Authentication Dialogue

### (a) Authentication Service Exchange: to obtain ticket-granting ticket

- C → AS : Options II ID<sub>c</sub> II Realm<sub>c</sub> II Times II Nonce<sub>1</sub>
- (2) AS → C : Realm<sub>c</sub> II ID<sub>c</sub> II Ticket<sub>tqs</sub> II EK<sub>c</sub> [ K<sub>c,tqs</sub> II Times II Nonce<sub>1</sub> II Realm<sub>tqs</sub> II ID<sub>tqs</sub>]

Ticket<sub>tgs</sub> = EK<sub>tgs</sub> [Flags II K<sub>c,tgs</sub> II Realm<sub>c</sub> II ID<sub>c</sub> II AD<sub>c</sub> II Times]

### (b) Ticket - Granting Service Exchange:to obtain service-granting ticket

- (3) C → TGS: Optionns II ID<sub>v</sub> II Times II Nonce<sub>1</sub>
- (4) TGS → C : Realm<sub>c</sub> II ID<sub>c</sub> II Ticket<sub>v</sub> II EK<sub>c,tgs</sub>[K<sub>c,v</sub> II Times II Nonce<sub>2</sub> II Realm<sub>v</sub> II ID<sub>v</sub>]

Ticket<sub>tgs</sub> = EK<sub>tgs</sub>[Flags II K<sub>c,tgs</sub> II Realm<sub>c</sub> II ID<sub>c</sub> II AD<sub>c</sub> II Times]

Ticket<sub>v</sub> = Ek<sub>v</sub>[[Flags II K<sub>c,v</sub> II Realm<sub>c</sub> II ID<sub>c</sub> II AD<sub>c</sub> II Times]

# Downloaded from EnggTree.com

Authenticator<sub>c</sub> =  $EK_{c,tgs}[ID_c II Realm_c II TS_1]$ 

### (c) Client/Server AUTHENTICATION Exchange: to obtain service

(5) C → V : Options II Ticket<sub>v</sub> II Authenticator<sub>c</sub>

(6)  $V \rightarrow C : EK_{c,v} [TS_2 | I subkey | I Seq #]$ Ticketv =  $EK_v[Flags | I K_{c,v} | I Realm_c | I | ID_c | I AD_c | I Times]$ Authenticator<sub>c</sub> =  $E_{Kc,v}[ID_c | I Realm_c | I TS_2 | I Subkey | I Seq#]$ 

First, consider the authentication service exchange. Message (1) is a client request for a ticket-granting ticket. It includes the ID of the user and the TGS.

The following new elements are added:

- Realm: Indicates realm of user
- Options: Used to request that certain flags be set in the returned ticket
- Times: Used by the client to request the following time settings in the ticket:

o from : the desired start time for the requested ticket

o till : the requested expiration time for the requested ticket

o r<sub>time</sub> : requested renew-till time

**Nonce**: A random value to be repeated in message (2) to assure that the response is fresh and has not been replaced by an opponent .

www.EnggTree.com

Message (2) returns a ticket-granting ticket, identifying information for the client, and a block encrypted using the encryption key based on the user's password. This block includes the session key to be used between the client and the TGS, times specified in message (1), the nonce from message (1), and TGS identifying information.

The ticket itself includes the session key, identifying information for the client, the requested time values, and flags that reflect the status of this ticket and the requested options.

Let us now compare the ticket-granting service exchange for versions 4 and 5.

We see that message (3) for both versions include an authenticator, a ticket, and the name of the requested service.

In addition, version 5 includes requested times and options for the ticket and a nonce, all with functions similar to those of message (1). The authenticator itself is essentially the same as the one used in version 4.

The authenticator itself is essentially the same as the one used in version 4.

Message (4) has the same structure as message (2), returning a ticket plus information needed by the client, the latter encrypted with the session key now shared by the client and the TGS.

Finally, for the client/server authentication exchange, several new features appear in version 5. In message (5), the client may request as an option that mutual authentication is required. The authenticator includes several new fields as follows:

- Subkey: The client's choice for an encryption key to be used to protect this specific
  application session. If this field is omitted, the session key from the ticket (Kc,v) is used.
- Sequence number: An optional field that specifies the starting sequence number to be used by the server for messages sent to the client during this session. Messages may be sequence numbered to detect replays.

If mutual authentication is required, the server responds with message (6). This message includes the timestamp from the authenticator. Note that in version 4, the timestamp was incremented by one. This is not necessary in version 5 because the nature of the format of messages is such that it is not possible for an opponent to create message (6) without knowledge of the appropriate encryption keys.

#### 13.EXPLIAN THE X.509 AUTHENTICATION SERVICES

X.509 defines a framework for authentication services by the X.500 directory to its users. The directory consists of public-key certificates.

Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.

X.509 defines authentication protocols based on public key certificates. X.509 standard certificate format used in S/MIME, IP Security and SSL/TLS and SET.

#### Certificates

The certificates are created and stored in the directory by the trusted Certification Authority (CA). The directory server not having certification functions and not create public key. But the user obtains the certificate from some easily accessible location

The general format of the certificate as shown below Fig 4.3

The elements of the certificates are

- 1. **Version(V):** The default version is 1. The issuer and subject unique identifier are present in version 2. If one or more extensions are present in version 3.
- 2. Serial Number (SN): Unique integer value issued by CA
- Signature Algorithm Identifier (AI): This algorithm is used to sign the certificate with some parameters
- 4. Issuer Name (CA): The name of the CA that created and signed this certificate
- 5. Period of validity (T<sub>A</sub>): The first and last on which the certificate is valid
- 6. Subject Name (A): The name of the user to whom this certificate refers
- 7. **Subject's Public Key Information (AP):** The public key of the subject plus identifier of the algorithm for which this key is to be used, with associated parameters.

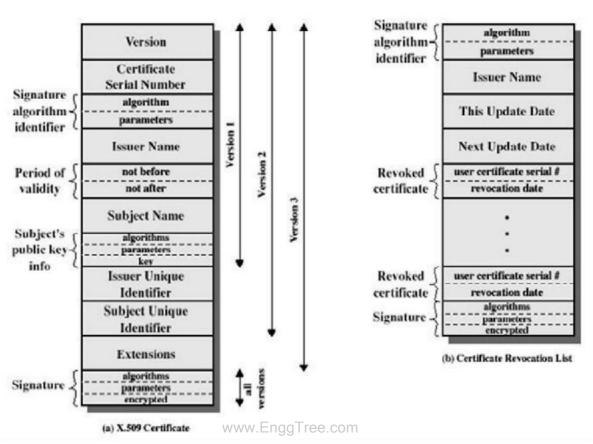


Fig X.509 AUTHENTICATION SERVICES

Issuer Unique Identifier: It is used to identify uniquely the issuing CA

- 8. Subject Unique Identifier: It is used to identify uniquely the subject
- 9. Extensions: A set of one or more extension fields
- 10. **Signature:** Covers all of the other fields of certificate; it contains hash code of other fields, encrypted with the CA's private key.

[Note: Unique identifier is used to avoid reuse of subject and issuer names over time]

#### Notation to define a certificate

 $CA << A>> = CA \{V, SN, AI, CA, TA, A, Ap\}$ 

Y<<X>> = The certificate of user X issued by certification authority Y.

Y {I} = The signing of I by Y. It consists of I with an encrypted hash code appended.

The CA signs the certificate with its private key. If the corresponding public key is known to a user, then that user can verify that a certificate signed by the CA is valid.

#### Generation and usage of certificate by a user

The user certificates generated by CA have the following characteristics:

# Downloaded from EnggTree.com

- 11. Any user with access to the public key of the CA can verify the user public key that was certified.
  - 12. No party other than the Certification Authority (CA) can modify the certificate without this being detected.
  - 13. Certificates are unforgeable,

If all users belong to the same CA, the certificates can be placed in the directory for access by all users. If the number of users increased, single CA could not be satisfied all user requirements.

For example, User A has obtained the certificate from certificate authority  $x_1$  and user B from  $x_2$ . Now the two CAs ( $x_1$  and  $x_2$ ) securities exchange their own public keys in the form of certificates. That is  $x_1$  must hold  $x_2$  s certificate and  $x_2$  holds  $x_1$  s certificate

Now A want s to access B"s public key, it follows the following chain to obtain B"s public key.

$$x_1 << x_2 >> x_2 << B >>$$

i.e., first A gets  $x_2$ "s certificate from  $x_1$ "s directory to obtain  $x_2$ "s public key. Then using  $x_2$ "s public key to obtain B"s certificate from  $x_2$ "s directory to obtain, s public key.

In the same method, B can obtain A\*s public key with the reverse chain  $x_2 << x_1 >> x_1 << A>>$ 

### **Hierarchy of CAs**

www.EnggTree.com

To obtain public key certificate of user efficiently, more than one CAs can be arranged in a hierarchy, so that navigation in easy.

The connected circles indicate the hierarchical(Fig 4.4) relationship among the CAs; the associated boxes indicate certificates maintained in the directory for each CA entry. The directory entry for each CA includes two types of certificates:

- Forward certificates: Certificates of X generated by other CAs
- Reverse certificates: Certificates generated by X that are the certificates of other CAs

User A can acquire the following certificates from the directory to establish a certification path to B:

$$X << W >> W << V >> V << Y >> < Z >> Z << B >>$$

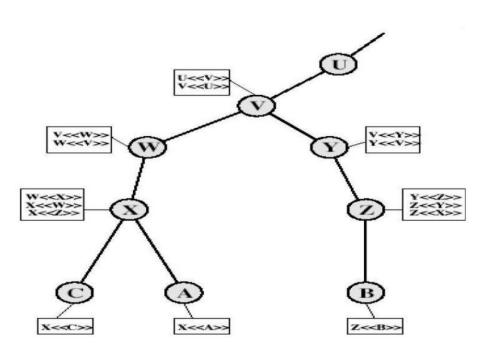


Fig4.4: Hierarchy of X.509

#### Revocation of certificates

- Certificates have a period of validity
- May need to revoke before expiry, for the following reasons eg:
  - √ User's private key is compromised
  - ✓ User is no longer certified by this CA
  - ✓ CA's certificate is compromised
- CA"s maintain list of revoked certificates
  - ✓ The Certificate Revocation List (CRL)
- Users should check Certificates with CA"s CRL

#### **Authentication Procedures**

X.509 includes three alternative authentication procedures:

- One-Way Authentication
- Two-Way Authentication
- Three-Way Authentication

### **One-Way Authentication**

One message ( A→B) used to establish

- The identity of A and that message is from A
- Message was intended for B
- Integrity & originality of message

Message must include timestamp, nonce, B's identity and is signed by A

### **Two-Way Authentication**

Two messages  $(A \rightarrow B, B \rightarrow A)$  which also establishes in addition:

- The identity of B and that reply is from B
- That reply is intended for A
- Integrity & originality of reply

Reply includes original nonce from A, also timestamp and nonce from B

### **Three-Way Authentication**

Three messages (A $\rightarrow$ B, B $\rightarrow$ A, A $\rightarrow$ B) which enables above authentication without synchronized clocks (Fig 4.5)

- Has reply from A back to B containing signed copy of nonce from B
- Means that timestamps need not be checked or relied upon

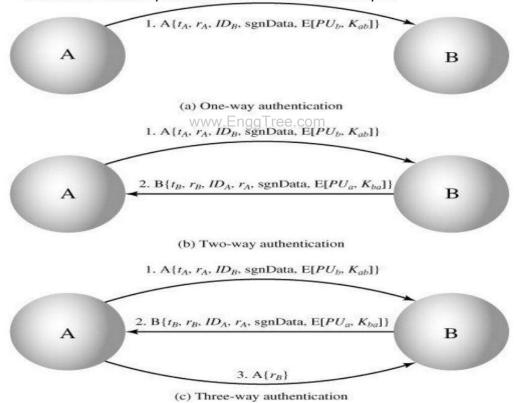


Fig: X509 Strong Authentication Procedure

### X.509 Version 3

The following requirements not satisfied by version 2:

- 14. The Subject field is inadequate to convey the identity of a key owner to a publickey user.
- 15. The Subject field is also inadequate for many applications, which typically recognize entities by an Internet e-mail address, a URL, or some other Internetrelated identification.
- 16. There is a need to indicate security policy information. There is a need to limit the damage that can result from a faulty or malicious CA by setting constraints on the applicability of a particular certificate.
- 17. It is important to be able to identify different keys used by the same owner at different times.

The certificate extensions fall into three main categories: key and policy information, subject and issuer attributes, and certification path constraints.

#### **Key and Policy Information**

These extensions convey additional information about the subject and issuer keys, plus indicators of certificate policy.. For example, a policy might be applicable to the authentication of electronic data interchange (EDI) transactions for the trading of goods within a given price range.

This area includes the following:

**Authority key identifier:** Identifies the public key to be used to verify the signature on this certificate or CRL.

Subject key identifier: Identifies the public key being certified.

**Key usage:** Indicates a restriction imposed as to the purposes for which, and the policies under which, the certified public key may be used.

**Private-key usage period:** Indicates the period of use of the private key corresponding to the public key. For example, with digital signature keys, the usage period for the signing private key is typically shorter than that for the verifying public key.

**Certificate policies:** Certificates may be used in environments where multiple policies apply.

Policy mappings: Used only in certificates for CAs issued by other CAs.

### Certificate Subject and Issuer Attributes

These extensions support alternative names, in alternative formats, for a certificate subject or certificate issuer and can convey additional information about the certificate subject, to increase a certificate user's confidence that the certificate subject is a particular person or entity. For example, information such as postal address, position within a corporation, or picture image may be required.

The extension fields in this area include the following:

- Subject alternative name: Contains one or more alternative names, using any of a variety of forms
- Subject directory attributes: Conveys any desired X.500 directory attribute values for the subject of this certificate.

#### **Certification Path Constraints**

These extensions allow constraint specifications to be included in certificates issued for CAs by other CAs. The extension fields in this area include the following:

- Basic constraints: Indicates if the subject may act as a CA. If so, a certification path length constraint may be specified.
- Name constraints: Indicates a name space within which all subject names in subsequent certificates in a certification path must be located.
- **Policy constraints**: Specifies constraints that may require explicit certificate policy identification or inhibit policy mapping for the remainder of the certification path.

www.EnggTree.com

### UNIT-V PART-A

### 1. What is Cyber crime.

• Cybercrime is defined as a crime where a computer is the object of the crime or is used as a tool to commit an offense.

### 2. List out the categories of cyber crime.

Cyber crime encloses a wide range of activities, but these can generally be divided into two categories:

- a) Crimes that aim computer networks or devices. These types of crimes involve different threats (like virus, bugs etc.) and denial-of-service attacks.
- b) Crimes that use computer networks to commit other criminal activities. These types of crimes include cyber stalking, financial fraud or identity theft.

### 3. What are 5 cyber crimes?

### **Examples of Basic Cyber Crimes**

- Stolen credit card information.
- Hacking into a Government Website EnggTree.com
- Account Takeover (ATO) Fraud.
- Compromised IoT devices
- Loss of control and access to content
- Phishing campaigns.
- Property.

### 4. List out the Classification of Cyber Crime:

Cyber Terrorism

Cyber Extortion

Cyber Warfare

Internet Fraud

Cyber Stalking

### 5. What is Password Cracking?

Password cracking is the process of using an application program to identify an unknown or forgotten password to a computer or network resource.

### 6. What is Keylogger?

Keyloggers, or keystroke loggers, are tools that record what a person types on a device. While there are legitimate and legal uses for keyloggers, many uses for keyloggers are malicious. In a keylogger attack, the keylogger software records every keystroke on the victim's device and sends it to the attacker.

### 7. What is spyware?

Spyware is malicious software that enters a user's computer, gathers data from the device and user, and sends it to third parties without their consent. A commonly accepted spyware definition is a strand of malware designed to access and damage a device without the user's consent.

### 8. What is SQL Injection?

A SQL injection (SQLi) is a technique that attackers use to gain unauthorized access to a web application database by adding a string of malicious code to a database query.

### 9. What is web security?

Web security threats are vulnerabilities within websites and applications, or attacks launched by malicious actors.

### 10. What is cyber security? List out its types.

Cyber security is the practice of protecting systems, networks, and programs from digital attacks. These cyber attacks are usually aimed at accessing, changing, or destroying sensitive information; extorting money from users via ransom ware; or interrupting normal business processes.

#### The 5 Types of Cyber security

- Critical infrastructure security.
- Application security.
- Network security.
- Cloud security.
- Internet of Things (IoT) security.

#### 1. Write Short notes on Cyber crime.

#### Cyber Crime

Meaning -

Criminal activities carried out by means of computers or the internet. Definition

\_

- Cybercrime is defined as a crime where a computer is the object of the crime or is used as a tool to commit an offense.
- A cybercriminal may use a device to access a user's personal information, confidential business information, government information, or disable a device.
- Cybercrime, also called computer crime, the use of a computer as an instrument tofurther illegal ends, such as committing fraud, trafficking in child pornography and intellectual property, stealing identities, or violating privacy.
- Cybercrime, especially through the Internet, has grown in importance as the computer has become central to commerce, entertainment, and government.
- Cyber crime or computer-oriented crime is a crime that includes a computer and anetwork. The computer may have been used in the execution of a crime or it may bethe target.

Cyber crime encloses a wide range of activities, but these can generally be divided into two categories:

- a) Crimes that aim computer networks or devices. These types of crimes involve different threats (like virus, bugs etc.) and denial-of-service attacks.
- b) Crimes that use computer networks to commit other criminalactivities. These types of crimes include cyber stalking, financial fraud or identity theft.

### Origin of the word Cyber Crime

Cyber came from cybernetics. Cybernetics influences game, system, and organizational theory. Cybernetics derived from the Greek kubernētēs which refers to a

pilot or steersman. Related is the Greek word kubernēsis which means "the gift ofgovernance" and applies to leadership.

### Who are cyber criminals?

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.

Types of Cyber Criminals:

#### 1. Hackers:

The term hacker may refer to anyone with technical skills, however, it typically refers to an individual who uses his or her skills to achieve unauthorized access to systems or networks so as to commit crimes.

### Organized Hackers:

These criminals embody organizations of cyber criminals, terrorists, and state- sponsored hackers. Cyber criminals are typically teams of skilled criminals targeted on control, power, and wealth. These criminals are extremely organized, and should even give crime as a service. These attackers are usuallyprofoundly prepared and well-funded.

### 3. Internet stalkers: www.EnggTree.com

Internet stalkers are people who maliciously monitor the web activity of their victims to acquire personal data. This type of cyber crime is conducted throughthe use of social networking platforms and malware, that are able to track an individual's PC activity with little or no detection.

### 4. Disgruntled Employees:

Disgruntled employees become hackers with a particular motive and also commit cyber crimes. It is hard to believe that dissatisfied employees can become such malicious hackers. In the previous time, they had the only option of going on strike against employers. But with the advancement of technology there is increased in work on computers and the automation of processes, it is simple for disgruntled employees to do more damage to their employers and organization by committing cyber crimes. The attacks by such employees bringthe entire system down.

### 2.Describe the Classification of Cyber Crimes in detail.

### Email spoofing

- Email spoofing is a form of cyber attack in which a hacker sends an email that has been manipulated to seem as if it originated from a trusted source.
- For example, a spoofed email may pretend to be from a well-known shopping website, asking the recipient to provide sensitive data, such as a password or credit card number.
- Alternatively, a spoofed email may include a link that installs malware on theuser's device if clicked.
- An example of spoofing is when an email is sent from a false sender address, that asks the recipient to provide sensitive data.
- This email could also contain a link to a malicious website that contains malware.

#### Spamming

- Spamming is the use of electronic messaging systems like e-mails and other digital delivery systems and broadcast media to send unwanted bulk messages indiscriminately.
- The term spamming is also applied to other media like in internet forums, instant messaging, and mobile text messaging, social networking spam, junk fax transmissions, television advertising and sharing network spam.
- Spam is any kind of unwanted, unsolicited digital communication that gets sentout in bulk.
   Often spam is sent via email, but it can also be distributed via textmessages, phone calls, or social media.

#### Cyber defamation

- The tort of cyber defamation is an act of intentionally insulting, defaming or offending another individual or a party through a virtual medium.
- It can be both written and oral.
- Defamation means giving an "injury to the reputation of a person" resulting from a statement which is false. The term defamation is used in the section 499 of Indian Penal Code, 1860.
- Cyber defamation is also known as internet defamation or online defamation in the world of internet and its users.
- Cyber defamation is also known as internet defamation or online defamation in the world of internet and its users.

Cyber defamation is a new concept but it virtually defames a person through new medium.
 The medium of defaming the individual's identity is through the help of computers via internet.

#### Internet time theft

- It refers to the theft in a manner where the unauthorized person uses internethours paid by another person.
- The authorized person gets access to another person's ISP user ID and password, either by hacking or by illegal means without that person's knowledge.
- Basically, Internet time theft comes under hacking. It is the use by anunauthorized person, of the Internet hours paid for by another person.

#### Salami Attack

- A salami attack is a small attack that can be repeated many times veryefficiently. Thus the combined output of the attack is great.
- In the example above, it refers to stealing the round-off from interest in bankaccounts.
- Even though it is less than 1 cent per account, when multiplied by millions of accounts over many months, the adversary can retrieve quite a large amount. It is also less likely to be noticeable since your average customer would assume that the amount was rounded down to the nearest cent.

#### **Data Diddling**

- Data diddling is a type of cybercrime in which data is altered as it is entered into a computer system, most often by a data entry clerk or a computer virus.
- Data diddling is an illegal or unauthorized data alteration. Changing data beforeor as it is input into a computer or output.
- Example: Account executives can change the employee time sheet information of employees before entering to the HR payroll application.

#### **Forgery**

Forger" redirects here.

- When a perpetrator alters documents stored in computerized form, the crime committed may be forgery. In this instance, computer systems are the target ofcriminal activity.
- The term forgery usually describes a message related attack against a cryptographic digital signature scheme. That is an attack trying to fabricate a digital signature for a message without having access to the respective signer'sprivate signing key.
- Among the many examples of this crime, taking another's work, whether it be written or visual, such as a artwork, and attempting to distribute it as either yourown or as an original is an example of forgery.
- Likewise, either creating fake documents or producing counterfeit items is considered to be forgery as well.

#### **Web Jacking**

- Illegally seeking control of a website by taking over a domain is know as WebJacking.
- Web jacking attack method is one kind of trap which is spread by the attacker to steal the sensitive data of any people, and those people got trapped who are not aware about cyber security.
- Web jacking attack method is another type of social engineering phishing attack where an attacker create a fake web page of victim website
- An attacker send it to the victim and when a victim click on that link, a messagedisplay on the browser "the site abc.com has move on another address, click here to go to the new location"
- If a victim does click on the link, he/she will redirect on the fake website page where an attacker can ask for any sensitive data such as credit card number, username, password etc.

### **Emanating from UseNet**

- Usenet is a kind of discussion group where people can share views on topic of their interest.
   The article posted to a newsgroup becomes available to all readers of the newsgroup.
- By its very nature, Usenet groups may carry very offensive, harmful, inaccurateor otherwise inappropriate material, or in some cases, postings that have been mislabeled or are deceptive in another way.

Therefore, it is expected that you will use caution and common sense and exercise proper judgment when using Usenet, as well as use the service at yourown risk.

### **Industrial Espionage**

- Industrial espionage describes a series of covert activities in the corporate world such as the theft of trade secrets by the removal, copying, or recording of confidential or valuable information in a company. The information obtained ismeant for use by a competitor.
- Economic or industrial espionage commonly occurs in one of two ways.
  - i) a dissatisfied employee appropriates information to advance interests or todamage the company.
  - ii) Secondly, a competitor or foreign government seeks information to advance its own technological or financial interest.
- Industrial espionage and spying can occur in any industry -- from food and beverage to fashion and entertainment.
- However, technology is one of the most targeted industries.
- Key technology industries that are often targeted include computer, semiconductor, electronics, automotive, aerospace, biotechnology, energy, pharmaceutical and high-tech manufacturing.

### **Hacking**

- Hacking refers to activities that seek to compromise digital devices, such as computers, smartphones, tablets, and even entire networks.
- Hacking is an attempt to exploit a computer system or a private network inside a computer.
   Simply put, it is the unauthorized access to or control over computernetwork security systems for some illicit purpose.
- They can destroy, steal or even prevent authorized users from accessing the system.
- Kevin Mitnick likely holds the title as the world's best hacker ever. Kevin Mitnick started hacking at an early age. He broke into the realm of public attention in the 1980s after he hacked into the North American Defense Command

 Ankit Fadia (born 1985) is an Indian author, speaker, television host, a security charlatan, and self-proclaimed white-hat computer hacker.

#### Types of Hackers

- a) White Hat Hackers These hackers utilize their programming aptitudes for a good and lawful reason. These hackers may perform network penetration tests in an attempt to compromise networks to discover network vulnerabilities. Security vulnerabilities are then reported to developers to fix them.
- b) Black Hat Hackers -

These hackers are unethical criminals who violate network security for personal gain. They misuse vulnerabilities to bargain PC frameworks.

c) Gray Hat Hackers – These hackers carry out violations and do seemingly deceptive things however not for individual addition or to cause harm. These hackers may disclose a vulnerability to the affected organization after having compromised their network.

www.EnggTree.com

### **Email bombing**

- An email bomb or "mail bomb" is a malicious act in which a large number of <a href="mail">email</a> messages are sent to a single email address in a short period of time. The purpose of an email bomb is typically to overflow a user's <a href="mailto:inbox">inbox</a>. In somecases, it will also make the mail server unresponsive.
- Email bombing is often done from a single system in which one user sends hundreds or thousands of messages to another user. In order to send the messages quickly, the email bomber may use a <u>script</u> to automate the process. By sending emails with a script, it is possible to send several thousand messagesper minute.
- Fortunately, most mail <u>servers</u> are capable of detecting email bombs before a large number of
  messages are sent. For example, if the server detects that more than ten messages are
  received from the same email address within one minute,

it may block the sender's email address or <u>IP address</u>. This simple action willstop the email bomb by rejecting additional emails from the sender.

### Intrusion

- The definition of an intrusion is an unwelcome interruption or a situation where somewhere private has an unwelcome visit or addition. When you are having a quiet nap in your backyard and your neighbor's dog comes in uninvited and jumps all over you to wake you up, this is an example of an intrusion.
- A network intrusion refers to any unauthorized activity on a digital network. Network intrusions often involve stealing valuable network resources and almost always jeopardize the security of networks and/or their data. In order to proactively detect and respond to network intrusions, organizations and their cybersecurity teams need to have a thorough understanding of how network intrusions work and implement network intrusion, detection, and response systems that are designed with attack techniques and cover-up methods in mind.

### www.EnggTree.com

#### Password sniffing

- Password Sniffing is a hacking technique that uses a special software application that allows a hacker to steal usernames and passwords simply by observing and passively recording network traffic. This often happens on public WiFi networks where it is relatively easy to spy on weak or unencrypted traffic.
- Password sniffing is an attack on the Internet that is used to steal user names and passwords from the network. Today, it is mostly of historical interest, as mostprotocols nowadays use strong encryption for passwords. However, it used to be the worst security problem on the Internet in the 1990s, when news of major password sniffing attacks were almost weekly.
- The typical implementation of a password sniffing attack involves gaining access to a computer connected to a local area network and installing a password sniffer on it. The password sniffer is a small program that listens to all traffic in the attached network(s), builds data streams out of TCP/IP packets,

# Downloaded from EnggTree.com

- and extracts user names and passwords from those streams that contain protocols that send cleartext passwords.
- The attack can also be performed in switches, routers, and printers. It is common nowadays for attackers to install presence on such devices. They don't run anti-virus and aren't easy to audit. Furthermore, traffic naturally goes through switches and routers, so no extra network packets need to be sent to fool switches into sending traffic of interest to the listening node.

### Credit card fraud

- Credit card fraud occurs when an unauthorized person gains access to your information and uses it to make purchases. ... Skimming your credit card, such as at a gas station pump. Hacking your computer. Calling about fake prizes orwire transfers.
- Here criminals make purchases or obtain cash advances using a credit card account assigned to you. This can occur through one of your existing accounts, via theft of your physical credit card or your account numbers and PINs, or by means of new credit card accounts being opened in your name without your knowledge. Once they're in, thieves then run up charges and stick you and yourcredit card company with the bill.

### **Identity Theft**

www.EnggTree.com

 Identity theft is the crime of obtaining the personal or financial information of another person to use their identity to commit fraud, such as making unauthorized transactions or purchases.

### 3. Write short notes on cyber security

### **Definition-**

Cyber security is the technique of protecting computers, networks, programs and datafrom unauthorized access or attacks that are aimed for exploitation.

OR

Cyber security refers to the measures taken to keep electronic information private and safe from damage or theft. It is also used to make sure electronic devices and data arenot misused.

OR

Cyber security is the body of technology, processes and practices designed to protect network, computers, computer programs and data from attack, damage or unauthorized access.

### **Cyber Security Threats-**

A cyber or cyber security threat is a malicious act that seeks to damage data, steal data, or disrupt digital life in general.

www.EnggTree.com

Cyber threats include computer viruses, <u>data breaches</u>, Denial of Service (<u>DoS</u>) attacks and other attack vectors.

### Viruses-

- A computer virus is a program which can harm our device and files and infectthem for no further use.
- When a virus program is executed, it replicates itself by modifying other computer programs and instead enters its own coding.
- This code infects a file or program and if it spreads massively, it may ultimately result in crashing of the device.
- Viruses affect your computer by corrupting files, interrupting Internet traffic andtaking over basic functions of your operating system.
- These behaviors can knock a system offline and cause crashes.

- Viruses can record keystrokes and screen data, and they may steal personalinformation and passwords to transmit back to the malware author.
- Particularly malicious viruses completely take over a computer and use it as aweapon against others.

### Trojan -

- A Trojan horse is malicious software that is concealed as a useful hostprogram.
- When the host program is run, the Trojan performs a harmful/unwanted
- A Trojan horse, often known as a Trojan, is malicious malware or software that appears to be legal yet has the ability to take control of yourcomputer.
- A Trojan is a computer program that is designed to disturb, steal, or otherwise harm your data or network.

### Malwares-

- Malware ("malicious software") is a type of computer program that infiltrates and damages systems without the users' knowledge.
- Malware tries to go unnoticed by either hiding or not letting the userknow about its presence on the system.
- You may notice that your system is processing at a slower rate thanusual.

### Ransom wares-

- Ransom ware is a type of <u>malware</u> that threatens to publish the victim's <u>personal</u> data or block access to it unless a <u>ransom</u> is paid.
- While some simple ransom ware may lock the system so that it is not difficult for a knowledgeable person to reverse, more advanced malware uses a technique called crypto viral extortion.
- It <u>encrypts</u> the victim's files, making them inaccessible, and demands a ransom payment to decrypt them.

 Ransom ware attacks are typically carried out using a <u>Trojan</u> as a file that the user is tricked into downloading or opening when it arrives as an email attachment.

#### Vulnerability-

- In cyber security, a vulnerability is a weakness that can be exploited by cybercriminals to gain unauthorized access to a computer system.
- After exploiting a vulnerability, a cyber attack can run malicious code, install malware and even steal sensitive data.

### Security Vulnerability Types

Computer security vulnerabilities can be divided into numerous types based on different criteria—such as where the vulnerability exists, what caused it, or how it could be used. Some broad categories of these vulnerability types include:

- Network Vulnerabilities
- Operating System Vulnerabilities
- Human Vulnerabilities
- Process Vulnerabilities www.EnggTree.com

#### Network Vulnerabilities

- These are issues with a network's hardware or software that expose it to possible intrusion by an outside party.
- Examples include insecure Wi-Fi access points and poorly-configured firewalls.

#### Operating System Vulnerabilities

- These are vulnerabilities within a particular operating system that hackers may exploit to gain access to an asset the OS is installed on—or to causedamage.
- Examples include default superuser accounts that may exist in some OS installs and hidden backdoor programs.

### Human Vulnerabilities

- The weakest link in many cyber security architectures is the <u>humanelement</u>.
- User errors can easily expose sensitive data, create exploitable accesspoints for attackers, or disrupt systems.

### CIA Triad-

- Confidentiality, integrity and availability, also known as the CIA triad, is a
  model designed to guide policies for information security within an organization.
- The model is also sometimes referred to as the AIC triad (availability, integrity and confidentiality) to avoid confusion with the Central Intelligence Agency.
- The CIA triad is a common, respected model that forms the basis for thedevelopment of security systems and policies.
- These are used for the identification of vulnerabilities and methods for addressing problems and creating effective solutions.

### Examples of CIA Triad

- The two-factor authentication (debit card with the PIN code) provides confidentiality before authorizing access to sensitive data.
- The ATM and bank software ensure data integrity by maintainingall transfer and withdrawal records made via the ATM in the user'sbank accounting.



### Confidentiality

- Confidentiality involves the efforts of an organization to make sure data is kept secret or private.
- To accomplish this, access to information must be controlled to prevent the unauthorized sharing of data—whether intentional or accidental.
- A key component of maintaining confidentiality is making sure that people without proper authorization are prevented from accessing assets important to your business. Conversely, an effective system also ensures that those who need to have access have the necessary privileges.
- For example, those who work with an organization's finances should be able to access the spreadsheets, bank accounts, and other information related to the flow of money. However, the vast majority of other employees—and perhaps even certain executives—may not be granted access. To ensure these policies are followed, stringent restrictions have to be in place to limit who can see what.

### **Integrity**

- Integrity involves making sure your data is trustworthy and free from tampering. The integrity of your data is maintained only if the data is authentic, accurate, and reliable.
- For example, if your company provides information about senior managers onyour website, this information needs to have integrity. If it is inaccurate, those visiting the website for information may feel your organization is not trustworthy. Someone with a vested interest in damaging the reputation of your organization may try to hack your website and alter the descriptions, photographs, or titles of the executives to hurt their reputation or that of thecompany as a whole.

#### Availability

- Even if data is kept confidential and its integrity maintained, it is often useless unless it is available to those in the organization and the customers they serve.
- This means that systems, networks, and applications must be functioning as they should and when they should.

# 4.Write short notes on Tools and Methods used in Cyber Crime

### **Proxy Server**

- It is a server (a computer system or an application) 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 and providing anonymity.

A proxy server may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet.

- A proxy server that passes requests and responses unmodified is usuallycalled a gateway or sometimes a tunneling proxy.
- A forward proxy is an Internet-facing proxy used to retrieve from wide www.Enggliee.com range of sources (in most cases anywhere on the Internet).
- A reverse proxy is usually an Internet-facing proxy used as a front-end tocontrol and protect access to a server on a private network. A reverse proxy commonly also performs tasks such as load-balancing, authentication, decryption or caching.

#### Anonymizer

- An anonymizer or an anonymous proxy is a tool that attempts to make activity on the Internet untraceable.
- It is a proxy server computer that acts as an intermediary and privacyshield between a client computer and the rest of the Internet.
- It accesses the Internet on the user's behalf, protecting personalinformation by hiding the client computer's identifying information.

There are many reasons for using anonymizers.

- Anonymizers help minimize risk.
- They can be used to prevent identity theft, or to protect search histories from public disclosure.
- Some countries apply heavy censorship on the internet. Anonymizers can help in allowing free access to all of the internet content, but cannot helpagainst persecution for accessing the Anonymizer website itself.
- Furthermore, as information itself about Anonymizer websites are banned in these countries, users are wary that they may be falling into agovernment-set trap.
- Anonymizers are also used by people who wish to receive objective information with the growing target marketing on the internet and targeted information.
- An anonymizer or an anonymous proxy is a tool that attempts to make activity on the Internet untraceable.
- It is a proxy server computer that acts as an intermediary and privacyshield between a client computer and the rest of the Internet.

It accesses the Internet on the user's behalf, protecting personal information by hidingthe client computer's identifying information

### Phishing

- Phishing is a cybercrime in which a target or targets are contacted by email, telephone or text message by someone posing as a genuine (legal) organization to ensnare individuals into providing sensitive data such as personally identifiable information, banking and credit card details, andpasswords.
- The information is then used to access important accounts and can resultin identity theft and financial loss.
- Phishers frequently use emotions like fear, curiosity, urgency, and greedto force recipients to open attachments or click on links.
- Phishing attacks are designed to appear to come from legitimate (legal)companies and individuals.

#### Viruses-

- A computer virus is a program which can harm our device and files and infectthem for no further use.
- When a virus program is executed, it replicates itself by modifying other computer programs and instead enters its own coding.
- This code infects a file or program and if it spreads massively, it may ultimately result in crashing of the device.
- Viruses can record keystrokes and screen data, and they may steal personalinformation and passwords to transmit back to the malware author.
- Particularly malicious viruses completely take over a computer and use it as aweapon against others.
- Viruses can record keystrokes and screen data, and they may steal personalinformation and passwords to transmit back to the malware author.
- Particularly malicious viruses completely take over a computer and use it as aweapon against others.

#### Worms-

- Computer worms are similar to viruses in that they replicate themselves and caninflict similar damage.
- Unlike viruses, which spread by infecting a host file, worms are freestandingprograms
  that do not require a host program or human assistance to propagate.
- Worms don't change programs; instead, they replicate themselves over andover.
- They just eat resources to make the system down.

#### Kevlogger -

- Keyloggers are a form of spyware where users are unaware their actions are being tracked.
- Keyloggers can be used for a variety of purposes; hackers may use them to maliciously gain access to your private information, while employersmight use them to monitor employee activities.
- A keylogger is a tool that captures and records a user's keystrokes. It can record
  instant messages, email, passwords and any other information you type at any time
  using your keyboard.

Keyloggers can be hardware or software.

### Trojan -

- A Trojan horse is malicious software that is concealed as a useful hostprogram.
- When the host program is run, the Trojan performs a harmful/unwantedaction.
- A Trojan horse, often known as a Trojan, is malicious malware or software that appears to be legal yet has the ability to take control of yourcomputer.
- A Trojan is a computer program that is designed to disturb, steal, or otherwise harm your data or network.

#### **Backdoor Attack**

- A backdoor is a malware type that negates normal authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers, giving perpetrators the ability to remotely issue system commands and updatemalware.
- A well-known backdoor example is called FinSpy. When installed on a system, it enables the attacker to download and execute files remotely on the system the moment it connects to the internet, irrespective of the system's physical location. It compromises overall system security recom

#### 6.Write short notes on Keyloggers

Keyloggers are a form of spyware where users are unaware their actions are being tracked. Keyloggers can be used for a variety of purposes; hackers may use them tomaliciously gain access to your private information, while employers might use them to monitor employee activities.

Spyware is largely invisible software that gathers information about your computeruse, including browsing. Key loggers are a form of spyware that capture every keystroke you type; they can send this information to remote servers, where log-in information--including your passwords--can be extracted and used.

A keylogger is a tool that captures and records a user's keystrokes. It can record instant messages, email, passwords and any other information you type at any timeusing your keyboard. Keyloggers can be hardware or software.

Spyware is any software that installs itself on your computer and starts covertly monitoring your online behavior without your knowledge or permission. Spyware a kind of malware that secretly gathers information about a person or organization and relays this data to other parties.

There are two common types of keyloggers. Software and Hardware keyloggers.

- Software Keyloggers.
- Hardware Keyloggers.
- Spear Phishing.
- Drive-by-Downloads.
- Trojan Horse.
- 2-Step Verification.
- Install Anti Malware Software.
- Use Key Encryption Software.

**Spyware** is mostly classified into four types: adware, system monitors, tracking including web tracking, and trojans; examples of other notorious types include digital rights management capabilities that "phone home", keyloggers, rootkits, andweb beacons.

Keystroke logging, often called keylogging, is the practice of noting (or logging) the keys struck on a keyboard, typically in a covert manner so that the person using the keyboard is unaware that such actions are being monitored.

Keystroke logger or keylogger is quicker and easier way of capturing the passwords and monitoring the victims' IT savvy behavior. It can be classified assoftware keylogger and hardware keylogger.

### 1. Software Keyloggers

Software keyloggers are software programs installed on the computer systems which usually are located between the OS and the keyboard hardware, and every keystroke is recorded. Software keyloggers are installed on a computer system by Trojans or viruses without the knowledge of the user. Cybercriminals always install such tools on the insecure computer systems available in public places and an obtain the required information about the victim very easily. A keylogger

usually consists of two files that get installed in the same directory: a dynamic linklibrary (DLL) file and an EXEcutable (EXE) file that installs the DLL file and triggers it to work. DLL does all the recording of keystrokes.

### 2. Hardware Keyloggers

To install these keylogers, physical access to the computer system is required. Hardware keyloggers are small hardware devices. These are connected to the PC and/or to the keyboard and save every keystroke into a file or in the memory of thehardware device. Cybercriminals install such devices on ATM machines to captureATM Cards'PINs. Each keypress on the keyboard of the ATM gets registered by these keyloggers These keyloggers look like an integrated part of such systems; hence, bank customers are unaware of their presence.

### 3. Antikeylogger

Antikeylogger is a tool that can detect the keylogger installed on the computersystem and also can remove the tool.

Advantages of using antikeylogger are as follows:

- Firewalls cannot detect the installations of keyloggers on the systems; hence, antikeyloggers can detect installations of keylogger.
- This software does not require regular updates of signature bases to workeffectively such as other antivirus and antispy programs if not updated, it does not serve the purpose, which makes the users at risk.
- Prevents Internet banking frauds. Passwords can be easily gained with the help of installing keyloggers.
- It prevents ID theft
- It secures E-Mail and instant messaging/chatting.

### 4. Spywares

Spyware is a type of malware, that is installed on computers which collects information about users without their knowledge. The presence of Spyware is typically hidden, from the user, it is secretly installed on the user's personal computer. Sometimes, however, Spywares such as keyloggers are installed by theowner of a shared, corporate or public computer on purpose to secretly monitor other users.

It is clearly understood from the term Spyware that it secretly monitors the user. The features and functions of such Spywares are beyond simple monitoring.

Spyware programs collect personal information about the victim, such as the Internet surfing habits/patterns and websites visited. The Spyware can also redirect

Internet surfing activities by installing another stealth utility on the users' computersystem. Spyware may also have an ability to change computer settings, which mayresult in slowing of the Internet connection speeds and slowing of response time that may result into user complaining about the Internet speed connection with Internet Service Providers (ISP).

To overcome the emergence of Spywares that proved to be troublesome for the normal user, anti-Spyware softwares are available in the market. Installation of anti-Spyware has become a common element nowadays from computer security practices perspective.

### 6. Write short notes on SOL Injection

SQL injection (SQLi) is a type of cyberattack against web applications that use SQL databases such as IBM Db2, Oracle, MySQL, and MariaDB. As the name suggests, the attack involves the injection of malicious SQL statements to interfere with the queries sent by a web application to its database.

Using SQL injection, a hacker will try to enter a specifically crafted SQL commands into a form field instead of the expected information. The intent is to secure a response from the database that will help the hacker understand the database construction, such as table names.

#### DoS and DDoS Attack

A denial-of-service (DoS) attack floods a server with traffic, making a website or resource unavailable. A distributed denial-of-service (DDoS) attack is a DoS attackthat uses multiple computers or machines to flood a targeted resource.

A DoS attack is a denial of service attack where a computer is used to flood a server with TCP and UDP packets. A DDoS attack is where multiple systems target a single system with a DoS attack. The targeted network is then bombardedwith packets from multiple locations. All DDoS = DoS but not all DoS = DDoS

What types of resources are targeted by such DoS attacks? Prevents the authorizeduse of networks, systems, or applications with the help of resources such as memory, bandwidth, CPU, system resources, network connectivity, and diskspace.

There are three main types of DoS attacks:

# Downloaded from EnggTree.com

- Application-layer Flood. In this attack type, an attacker simply floods the service with requests from a spoofed IP address in an attempt to slow or crash the service, illustrated in . ...
- Distributed Denial of Service Attacks (DDoS) ...
- Unintended Denial of Service Attacks.

#### . DOS Attack:

A DOS attack is a denial of service attack, in this attack a computer sends massive amount of traffic to a victims computer and shuts it down. Dos attack is a online attack which is used to make the website unavailable for its users when done on a website. This attack make the server of a website down which is connected to internet by sending a large number of traffic toit.

### 2. DDOS Attack:

In ddos attack means distributed denial of service in this attack dos attacks are done from many different locations using many systems.

Attention reader! Don't stop learning now. Get hold of all the important CS
Theory concepts for SDE interviews with the <u>CS Theory Course</u> at a
student-friendly price and become industry ready.

DOS	DDOS
DOS Stands for Denial of service attack.	DDOS Stands for Distributed Denial of service attack.
In Dos attack single system targets the victims system.	In DDos multiple system attacks the victims system
Victim PC is loaded from the packet of data sent from a single location.	Victim PC is loaded from the packet ofdata sent from Multiple location.
Dos attack is slower as compared to ddos.	DDos attack is faster than Dos Attack.
Can be blocked easily as only one system is used.	It is difficult to block this attack as multiple devices are sending packets and
DOS	DDOS

attacking from multiple locations.

In DOS Attack only single device is used with DOS Attack tools.

In DDos attack Bots are used to attack atthe same

time.

DOS Attcaks are Easy to

trace.

DDOS Attacks are Difficult to trace.

Volume of traffic in Dos attack is less as compared to DDos.

DDoS attacks allow the attacker to send massive volumes of traffic to the victim network.

Types of DOS Attacks are:

1. Buffer overflow attacks

2. Ping of Death or ICMP

flood

3. Teardrop Attack

Types of DDOS Attacks are:

- 1. Volumetric Attacks
- 2. Fragmentation Attacks
- 3. Application Layer Attacks

www.EnggTree.com

#### 8. Write short notes on Password Cracking

Password cracking is the process of attempting to gain Unauthorized access to restricted systems using common passwords or algorithms that guess passwords. In other words, it's an art of obtaining the correct password that gives access to a system protected by an authentication method.

Password cracking refers to various measures used to discover computer passwords. This is usually accomplished by recovering passwords from data stored in, or transported from, a computer system. Password cracking is done by either repeatedly guessing the password, usually through a computer algorithm in which the computer tries numerous combinations until the password is successfully discovered.

Password cracking can be done for several reasons, but the most malicious reason is in order to gain unauthorized access to a computer without the computer owner's awareness. This results in cybercrime such as stealing passwords for the purpose of accessing banking information.

Other, nonmalicious, reasons for password cracking occur when someone has misplaced or forgotten a password. Another example of nonmalicious password cracking may take place if a system

#### 9. Write short notes on 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

# Downloaded from EnggTree.com

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 <a href="mailto:bring-your-own-device">bring-your-own-device</a> (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.

Further, cyber criminals are well aware of this increase in endpoint usage and continue to design and launch sophisticated campaigns that exploit any vulnerabilities in corporate networks. With more endpoints, the attack surface increases, which means more opportunities for fraudsters to gain access. NAC solutions can be configured to detect any unusual or suspicious network activity and respond with immediate action, such as isolating the device from the network to prevent the potential spread of the attack.

Although <u>IoT and BYOD have changed NAC solutions</u>, NAC also serves as a perpetual inventory of users, devices, and their level of access. It serves as an active discovery tool to uncover previously unknown devices that may have gained access to all or parts of the network, requiring IT administrators to adjust security policies.

Further, organizations can choose how NAC will authenticate users who attempt to gain access to the network. IT admins can choose multi-factor authentication (MFA), which provides an additional layer of security to username and password combinations.

Restricting network access also means control of the applications and data within the network, which is normally the target of cyber criminals. The stronger the network controls, the more difficult it will be for any cyberattack to infiltrate the network.

### Network access control comes with a number of benefits for organizations:

- 1. Control the users entering the corporate network
- 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
- Segment employees into groups based on their job function and build role-based access policies
- Protect against cyberattacks by putting in place systems and controls that detect unusual or suspicious activity
- Automate incident response

# Downloaded from EnggTree.com

Generate reports and insights on attempted access across the organization

### 10. Write short notes on cloud security.

Cloud security is the set of control-based security measures and technology protection, designed to protect online stored resources from leakage, theft, and data loss. Protection includes data from cloud infrastructure, applications, and threats. Security applications uses a software the same as SaaS (Software as a Service) model.

### cloud-based security systems benefit the business by:

- Protecting the Business from Dangers
- Protect against internal threats
- o Preventing data loss
- Top threats to the system include Malware, Ransomware, and
- o Break the Malware and Ransomware attacks
- Malware poses a severe threat to the businesses.

### 11. Write short notes on web security.

www.EnggTree.com

Web security solution should provide comprehensive protection to users against web-related cyber threats. Some of the essential features of a web security solution include:

**URL Filtering:** Cybercriminals use a variety of known-bad URLs as part of phishing campaigns or to deliver malware. URL filtering makes it possible to block users from visiting these known-bad and other inappropriate sites and to enforce bandwidth limitations on certain types of sites (such as video streaming).

- Application Control: Web security solutions perform traffic inspection at the application layer,
  which means that they have insight into the application generating the traffic and the data that it
  contains. This granular visibility makes it possible for web administrators to define application-specific
  rules to ensure that access to applications and sensitive data is properly controlled both inside and
  outside of the organization.
- Data Loss Prevention: Exfiltration of sensitive and proprietary data can occur in a variety of
  ways and carries significant costs to an organization. Data loss prevention (DLP) solutions monitor
  data flows to block potential leakages of sensitive and valuable information. Antivirus: Malicious
  websites are a major delivery vector for malware such as ransomware, trojans, and information

stealers. The antivirus built into a web security solution will inspect all traffic flowing through it to determine if it contains known malware samples identified by unique signatures.

SSL Introspection: A growing percentage of web traffic uses HTTPS, which encrypts the
traffic to protect it against eavesdropping. SSL introspection allows an organization's security solutions
to inspect this encrypted web traffic, enabling them to detect and block malicious content and data
exfiltration.

The web can be a dangerous place, and it poses significant risks to an organization and its employees. A web security solution needs a wide range of features to provide effective protection against these threats.

#### **Benefits of Web Security**

A web security solution has deep visibility and granular control over Internet-bound traffic. It inspects traffic at the application layer, providing a better understanding of its function and the data that it contains. These capabilities provide a number of benefits to an organization and its employees, such as:

www.EnggTree.com

**Malicious Content Protection:** Web security blocks known-bad phishing sites and drive-by downloads, and inspects web traffic for malicious content. This helps to protect employees against malware and other threats.

- Data Security: <u>DLP solutions</u> monitor movement of an organization's sensitive data. This
  helps to ensure that sensitive and valuable data is not exposed to unauthorized users.
- Regulatory Compliance: Companies need to comply with an ever-increasing number of data protection regulations. Web security solutions help with this by providing increased visibility and control for sensitive and protected data within an organization's possession.
- Improved Network Performance: Application control enables network administrators to apply application-specific policies. This allows throttling and blocking of certain sites and traffic, improving the network performance for legitimate business traffic.
- Secure Remote Work: Web security solutions enable remote employees to work securely from anywhere. Companies can apply and enforce corporate security policies on employee devices regardless of their location.

#### 12. Write short notes on wireless security.

# Downloaded from EnggTree.com

Wireless Network provides various comfort to end users but actually they are very complex in their working. There are many protocols and technologies working behind to provide a stable connection to users. Data packets traveling through wire provide a sense of security to users as data traveling through wire probably not heard by eavesdroppers.

To secure the wireless connection, we should **focus on** the following areas –

- Identify endpoint of wireless network and end-users i.e., Authentication.
- Protecting wireless data packets from middleman i.e., Privacy.
- Keeping the wireless data packets intact i.e., Integrity.

We know that wireless clients form an association with Access Points (AP) and transmit data back and forth over the air. As long as all wireless devices follow 802.11 standards, they all coexist. But all wireless devices are not friendly and trustworthy, some rogue devices may be a threat to wireless security. Rogue devices can steal our important data or can cause the unavailability of the network.

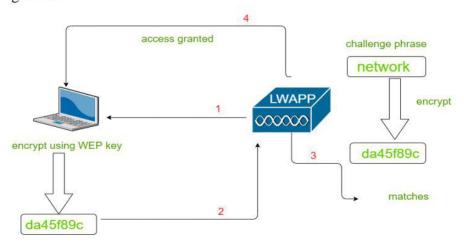
Wireless security is **ensured by** following methods-

- Authentication
- Privacy and Integrity

#### 1. Wired Equivalent Privacy (WEP):

For wireless data transmitting over the air, open authentication provides no security. WEP uses the RC4 cipher algorithm for making every frame encrypted. The RC4 cipher also encrypts data at the sender side and decrypt data at the receiving site, using a string of bits as key called WEP key.

WEP key can be used as an authentication method or encryption tool. A client can associate with AP only if it has the correct WEP key. AP tests the knowledge of the WEP key by using a challenge phrase. The client encrypts the phrase with his own key and send back to AP. AP compares the received encrypted frame with his own encrypted phrase. If both matches, access to the association is granted.



Working of WEP Authentication

### 2. Extensible Authentication Protocol (802.1x/EAP):

In WEP authentication, authentication of the wireless clients takes place locally at AP. But Scenario gets changed with 802.1x. A dedicated authentication server is added to the infrastructure. There is the participation of three devices –

1. Supplicant

Device requesting access.

Authenticator
 Device that provides access to network usually a Wlan controller (WLC).

3. Authentication Server

Device that takes client credentials and deny or grant access.



Supplicant WLC Authentication sever

EAP is further of four types with some amendments over each other –

- LEAP
- EAP-FAST
- PEAP
- EAP-TLS www.EnggTree.com

www.EnggTree.com

www.EnggTree.com