DOWNLOADED FROM STUCOR APP

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

1.1 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

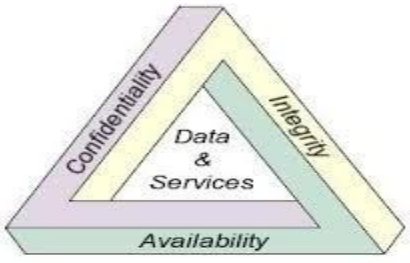


Figure 1.1 CIA triad

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

• **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

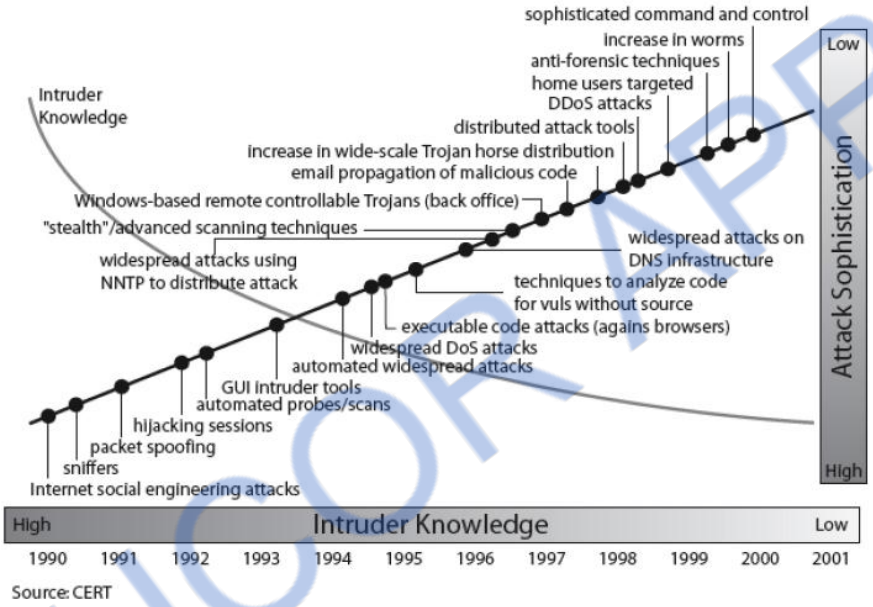


Figure 1.2 Security Trends

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

1.2 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.

1.2.1 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Computers as storage devices: Computers can be used to further unlawful activity by using a computer or a computer device as a passive 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.

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

1.2.2 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- 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.

1.2.3 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.

1.3 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.

1.3.1 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?

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

1.3.2 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.

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- **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?
    - What age should the users have?
    - 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.

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1.4 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.

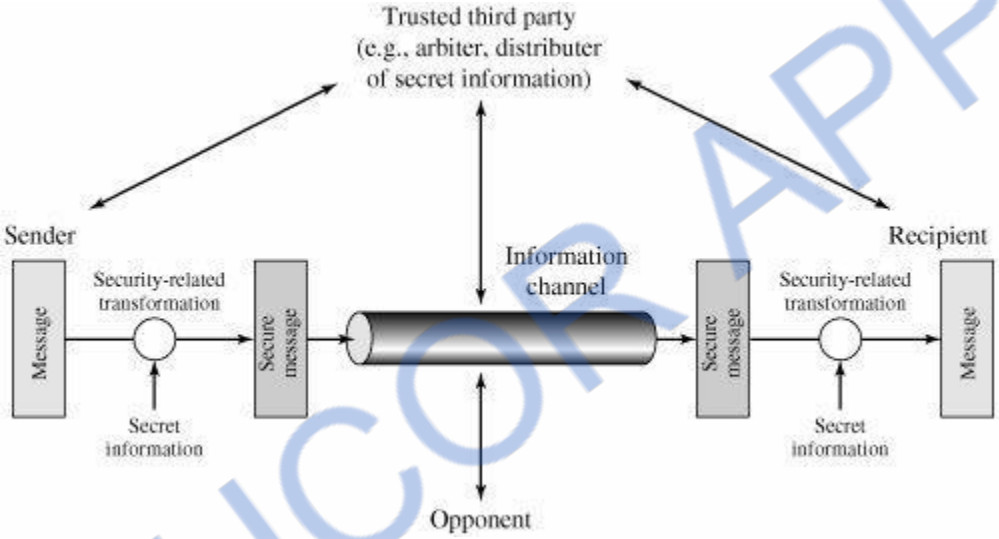


Figure 1.3 Model for Network Security

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).

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

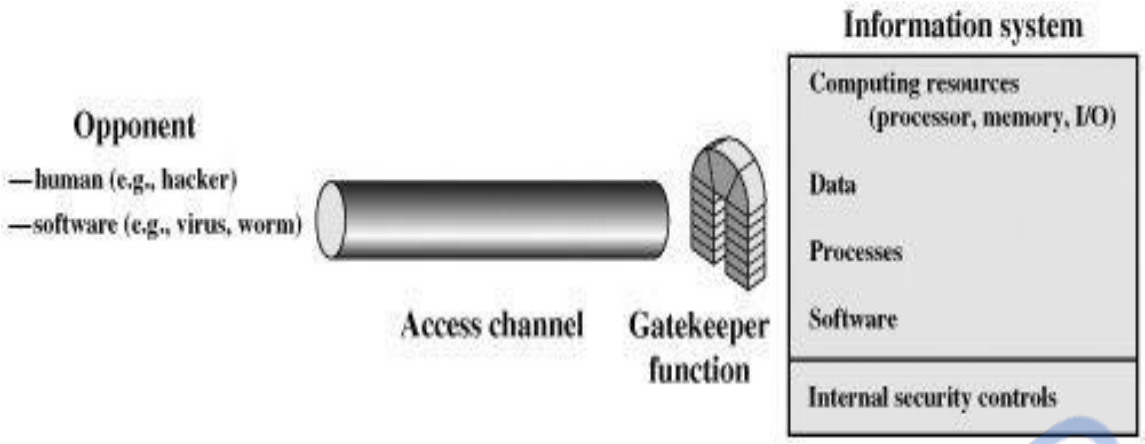


Figure 1.4 Network Access Security Model

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.

1.5 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

• **Security attack:** Any action that compromises the security of information owned by an organization.

• **Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

• **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.

•

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.

1.5.1 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.

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

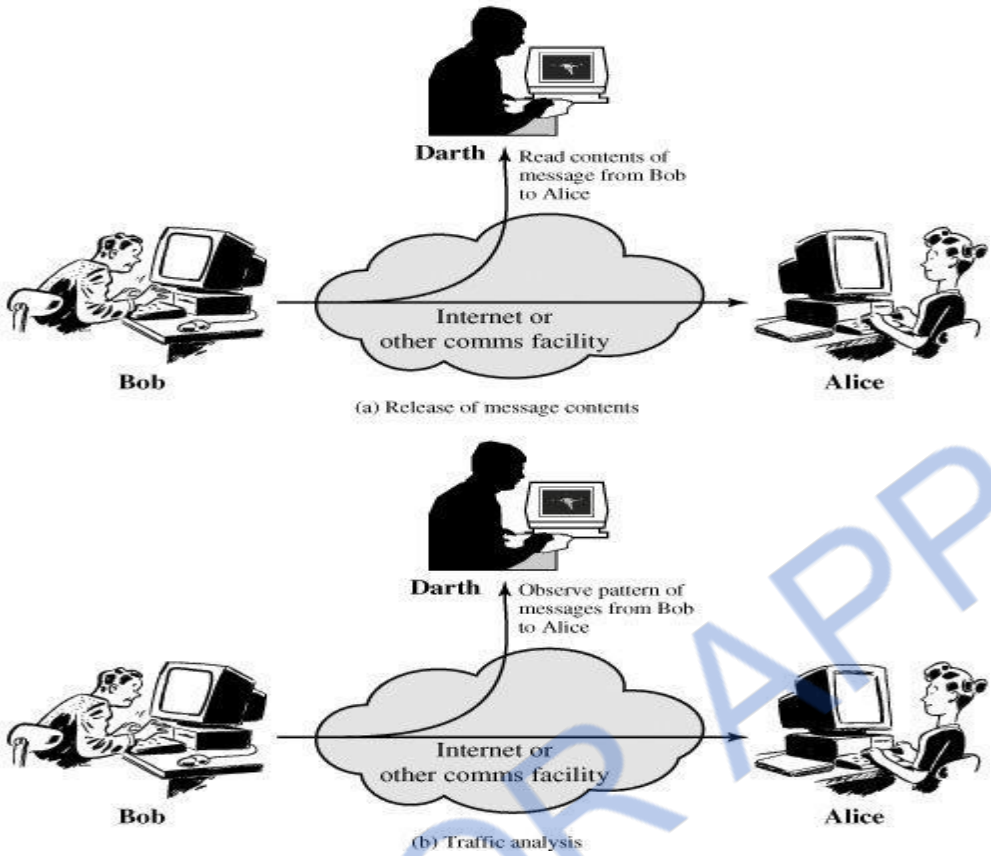


Figure 1.5 Passive Attacks

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.

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

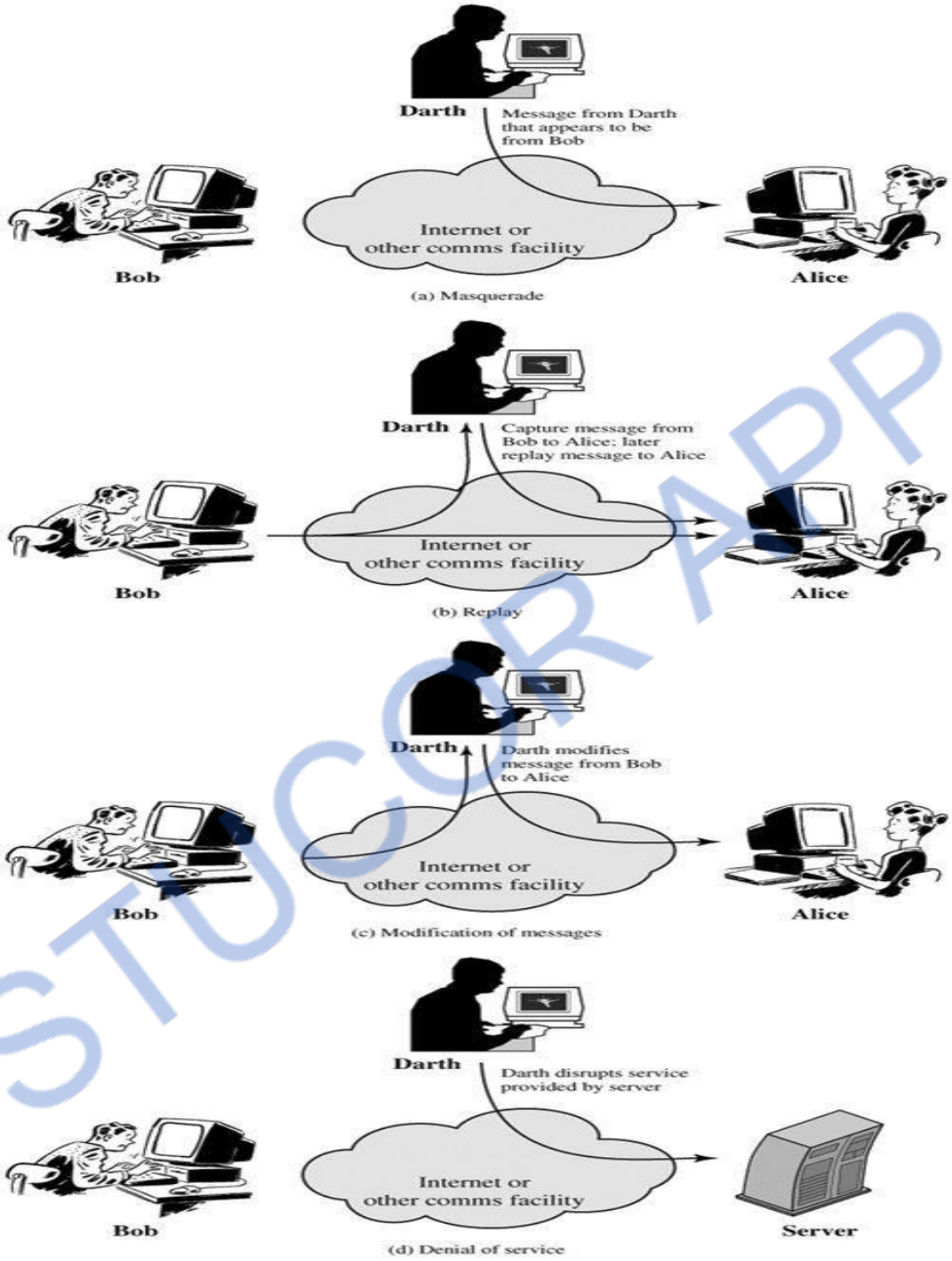


Figure 1.6 Active Attacks

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1.5.2 SERVICES

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1.5.3 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	PERVASIVE SECURITY MECHANISMS
May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.	Mechanisms that are not specific to any particular OSI security service or protocol layer.
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.	**Trusted Functionality** That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).
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).	**Security Label** The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
Access Control A variety of mechanisms that enforce access rights to resources.	**Event Detection** Detection of security-relevant events.
Data Integrity A variety of mechanisms used to assure the integrity of a data unit or stream of data units.	**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.
Authentication Exchange A mechanism intended to ensure the identity of an entity by means of information exchange.	**Security Recovery** Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.
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.	

STUCOR APP

1.6 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- 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**.

1.6.1 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.

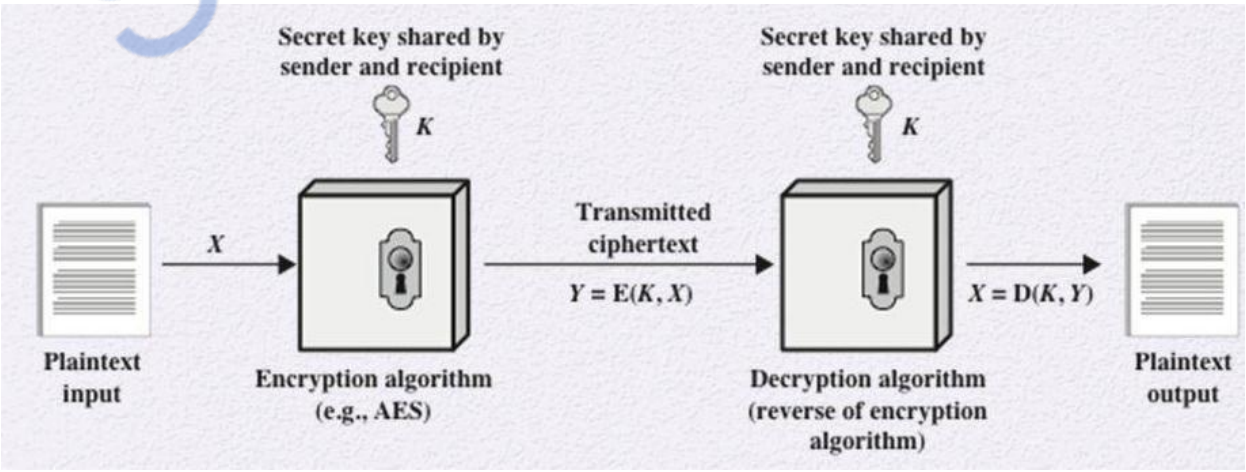


Figure 1.7 Simplified Model of Symmetric Encryption

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

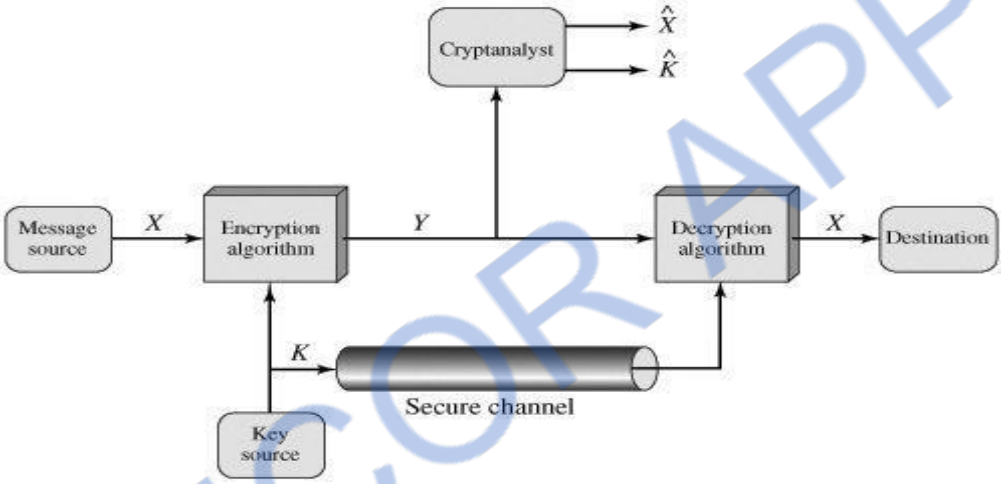


Figure 1.8 Model of Symmetric Cryptosystem

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1.6.2 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

1.6.3 SUBSTITUTION TECHNIQUES

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	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:

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

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

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

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

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

If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all the 25 possible keys. 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	eww1 ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	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 increasein 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 exactlyonce. 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 I j kl m n o p q r s t u v w x y z

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

If, instead, the “cipher” line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than $4*10^{26}$ 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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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				

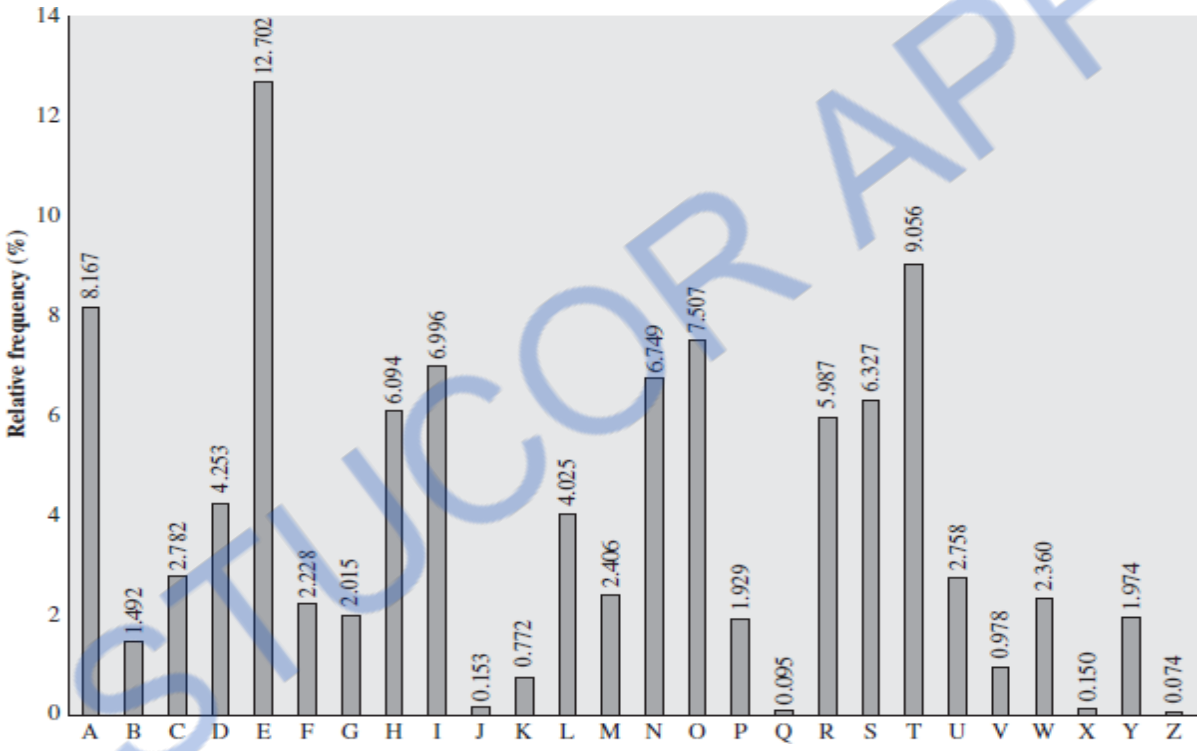


Figure 1.10 Relative Frequencies of Letters in English Text

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
 t a        e  e te  a that e e a       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 × 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey inDorothy Sayers"s *Have His Carcase*

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

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).

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 × 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 $\mathbf{M}^{-1}$of a square matrix **M** by the equation $\mathbf{M}(\mathbf{M}^{-1}) = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$, 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

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

$$\mathbf{A}\mathbf{A}^{-1} = \begin{pmatrix} (5 \times 9) + (8 \times 1) & (5 \times 2) + (8 \times 15) \\ (17 \times 9) + (3 \times 1) & (17 \times 2) + (3 \times 15) \end{pmatrix}$$
$$= \begin{pmatrix} 53 & 130 \\ 156 & 79 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

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

To explain how the inverse of a matrix is computed, we begin by with the concept of determinant. For any square matrix (*m* × 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 × 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 $[A^{-1}]_{ij}=(\det A)^{-1}\ (-1)^{i+j}\ (D_{ij})$where (D_{ij}) is the subdeterminant formed by deleting the jth row and the ith column of **A**, det(**A**) is the determinant of **A**, and $(\det A)^{-1}$ is the multiplicative inverse of (det **A**) mod 26. Continuing our example,

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

We can show that 9^{-1}mod26=3, because9×3=27mod26=1.Therefore, we compute the inverse of **A** as

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

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

$$\mathbf{C} = \mathbf{PK} \bmod 26$$

where **C** and **P** are row vectors of length 3 representing the plaintext and ciphertext, and **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

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

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, $(\det \mathbf{K})^{-1} \bmod 26 = 17$. 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 $\mathbf{K}^{-1}$ is applied to the ciphertext, then the plaintext is recovered.

In general terms, the Hill system can be expressed as

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

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

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

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 0"s and 1"s 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} binary digit of cipher text

P_i - i^{th} binary digit of plaintext

K_i - i^{th} binary digit of key

$\oplus$ – 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

OTP Encryption

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

E	Q	N	V	Z	Ciphertext
4 (E)	16 (Q)	13 (N)	21 (V)	25 (Z)	Ciphertext
23 (X)	12 (M)	2 (C)	10 (K)	11 (L)	Key
-19	4	11	11	14	Ciphertext - Key
7 (H)	4 (E)	11 (L)	11 (L)	14 (O)	Ciphertext - Key (mod 26)
H	E	L	L	O	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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

$$C = C_0, C_1, C_2, \ldots, C_{n-1} = \mathrm{E}(K, P) = \mathrm{E}[(k_0, k_1, k_2, \ldots, k_{m-1}), (p_0, p_1, p_2, \ldots, p_{n-1})]$$
$$= (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \ldots, (p_{m-1} + k_{m-1}) \bmod 26,$$
$$(p_m + k_0) \bmod 26, (p_{m+1} + k_1) \bmod 26, \ldots, (p_{2m-1} + k_{m-1}) \bmod 26, \ldots$$

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

$$Ci = (pi + ki \bmod m) \bmod 26$$

Decryption is a generalization of Equation

$$pi = (Ci - ki \bmod m) \bmod 26$$

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: deceptivedeceptivedeceptive
plaintext: wearediscoveredsaveyourself
ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1.6.4 TRANSPOSITION TECHNIQUES

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:

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

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,

```
Key:          4 3 1 2 5 6 7
Plaintext:    a t t a c k p
              o s t p o n e
              d u n t i l t
              w o a m x y z
Ciphertext:   TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

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,

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Key:	4 3 1 2 5 6 7
Input:	t t n a a p t
	m t s u o a o
	d w c o i x k
	n l y p e t z
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

1.7 STEGANOGRAPHY

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

1.8 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.

Table 1.5 Differences between Traditional Encryption and Modern Encryption

Traditional Encryption	**Modern Encryption**
For making ciphertext, manipulation is done in the characters of the plaintext	For making ciphertext, operations are 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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 encryption
- 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.
  - Message authentication.
  - 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.

Table 1.6 Primitives and Security Service

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

1.8.1 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.

1.8.2 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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).

Information theory is used in information retrieval, intelligence gathering, gambling, and even in musical composition.

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.

1.8.3 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1.9 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 pre-computed hashes to find matches.

Other Attacks using Cryptanalysis

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

2.1 ALGEBRAIC STRUCTURES

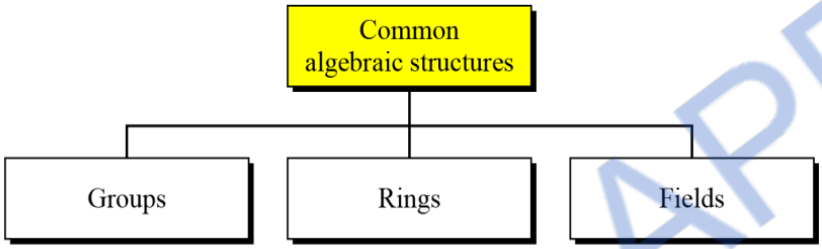


Figure 2.1 Common Algebraic Structures

2.1.1 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£ 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**.**

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 $b = 0$.

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 $aa^{-1} = (a^{-1})a = 1$.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

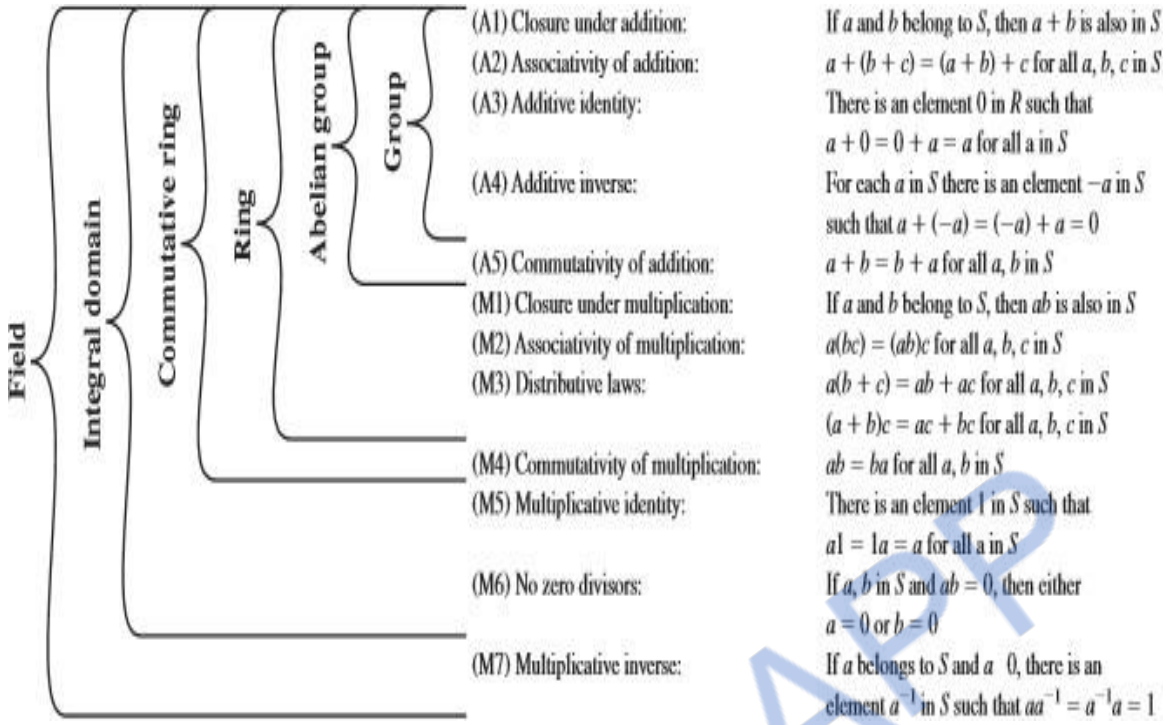


Figure 2.2 Groups, Ring and Field

2.2 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

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

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

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

Two integers a and b are said to be **congruent modulo n**, if $(a \bmod n) = (b \bmod n)$. This is written as $a \equiv b \pmod{n}$.[2]

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

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

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

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

Table 2.1 Arithmetic Modulo 8

$11 \bmod 8 = 3; 15 \bmod 8 = 7$
$[(11 \bmod 8) + (15 \bmod 8)] \bmod 8 = 10 \bmod 8 = 2$
$(11 + 15) \bmod 8 = 26 \bmod 8 = 2$
$[(11 \bmod 8) - (15 \bmod 8)] \bmod 8 = -4 \bmod 8 = 4$
$(11 - 15) \bmod 8 = -4 \bmod 8 = 4$
$[(11 \bmod 8) \times (15 \bmod 8)] \bmod 8 = 21 \bmod 8 = 5$
$(11 \times 15) \bmod 8 = 165 \bmod 8 = 5$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

(a) Addition modulo 8

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

(b) Multiplication modulo 8

w	$-w$	w^{-1}
0	0	—
1	7	1
2	6	—
3	5	3
4	4	—
5	3	5
6	2	—
7	1	7

(c) Additive and multiplicative inverses modulo 8

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2.3 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 ofa 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)
        α = a
        β = b
        while (β > 0)
                Rem = α mod β
                α = β
                β = Rem
        returnα
```

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 x 30 + 20	gcd (30, 20)
30	= 1 x 20 + 10	gcd (20,10)
20	= 1 x 10 + 10	gcd (10,10)
10	= 1 x 10 + 0	gcd (10,0)

Therefore, gcd (30,50) = 10

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)

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

=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 x 162 + 94	gcd (162, 94)
162	= 1 x 94 + 68	gcd (94, 68)
94	= 1 x 68 + 26	gcd (68, 26)
68	= 2 x 26 + 16	gcd (26, 16)
26	= 1 x 16 + 10	gcd (16, 10)
16	= 1 x 10 + 6	gcd (10, 6)
10	= 1 x 6 + 4	gcd (6, 4)
6	= 1 x 4 + 2	gcd (4, 2)
4	= 2 x 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 → (1)

17+9(-1) = 8 → (2)

43+17(-2) = 9 → (3)

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Substitution

sub equ 2 in equ 1

(1)→ 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 → (4)

Now sub equ (3) in equ (4)

43+17(-2) = 9 → (3)

17(-1)+(43+17(-2))(2)=1

17(-1)+43(2)+17(-4)=1

17(-5)+43(2) = 1→(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 ≡ 1 mod 43

646 ≡ 1 mod 43 (15*43 = 645)

Example 2

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 → (1)

23+3(-7) = 2 → (2)

26+23(-1) = 3 → (3)

1635+26(-62) = 23 → (4)

Substitution

sub equ (2) in equ (1)

(2) => 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 → (5)

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

sub equ (3) in equ (5)

26+23(-1) = 3 → (3)

(26+23(-1))(8) + 23 (-1) =1

26(8) + 23 (-8) + 23 (-1) =1

26 (8) + 23 (-9) =1 → (6)

Sub equ (4) in equ (6)

1635+26(-62) = 23 → (4)

26 (8) + (1635 + 26 (-62)) (-9) = 1

26 (8) + 1635 (-9) + 26 (558) =1

1635 (-9) + 26 (566) = 1 → (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 ≡ 1 mod 26

1635 * 17 ≡ 1 mod 26

27795 ≡ 1 mod 26 (1069*26 = 27794)

2.4 CONGRUENCE AND MATRICES

Properties of Congruences

Congruences have the following properties:

1. $a = b \pmod{n}$ if $n|(a - b)$.
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	$-11 - 5 = -16 = 8 \times (-2)$
$81 \equiv 0 \pmod{27}$	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 l x m where l defines the row and m defines the columns

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

$$\begin{array}{cc} & m \text{ columns} \\ l \text{ rows} & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{l1} & a_{l2} & \cdots & a_{lm} \end{bmatrix} \end{array}$$

Examples of Matrices
1. Row Matrix
2. Column Matrix
3. Square Matrix
4. Zero Matrixes
5. Identity Matrix

$$\underset{\text{Row matrix}}{\begin{bmatrix} 2 & 1 & 5 & 11 \end{bmatrix}} \quad \underset{\text{Column matrix}}{\begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}} \quad \underset{\text{Square matrix}}{\begin{bmatrix} 23 & 14 & 56 \\ 12 & 21 & 18 \\ 10 & 8 & 31 \end{bmatrix}} \quad \underset{\mathbf{0}}{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}} \quad \underset{\mathbf{I}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}$$

2.5 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:

+	0	1
0	0	1
1	1	0

Addition

×	0	1
0	0	0
1	0	1

Multiplication

w	$-w$	w^{-1}
0	0	—
1	1	1

Inverses

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)

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

CS8792 -
DOWNLOADED FROM STUCOR APP

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

(b) Multiplication modulo 7

w	$-w$	w^{-1}
0	0	—
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

(c) Additive and multiplicative inverses modulo 7

2.5.1 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 $n \geq 0$) is an expression of the form

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

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 nth-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**.

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

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

$$c_k = a_0 b_k + a_1 b_{k-1} + \cdots + a_{k-1} b_1 + a_k b_0$$

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

$$f(x) + g(x) = x^3 + 2x^2 - x + 3$$
$$f(x) - g(x) = x^3 + x + 1$$
$$f(x) \times g(x) = x^5 + 3x^2 - 2x + 2$$

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

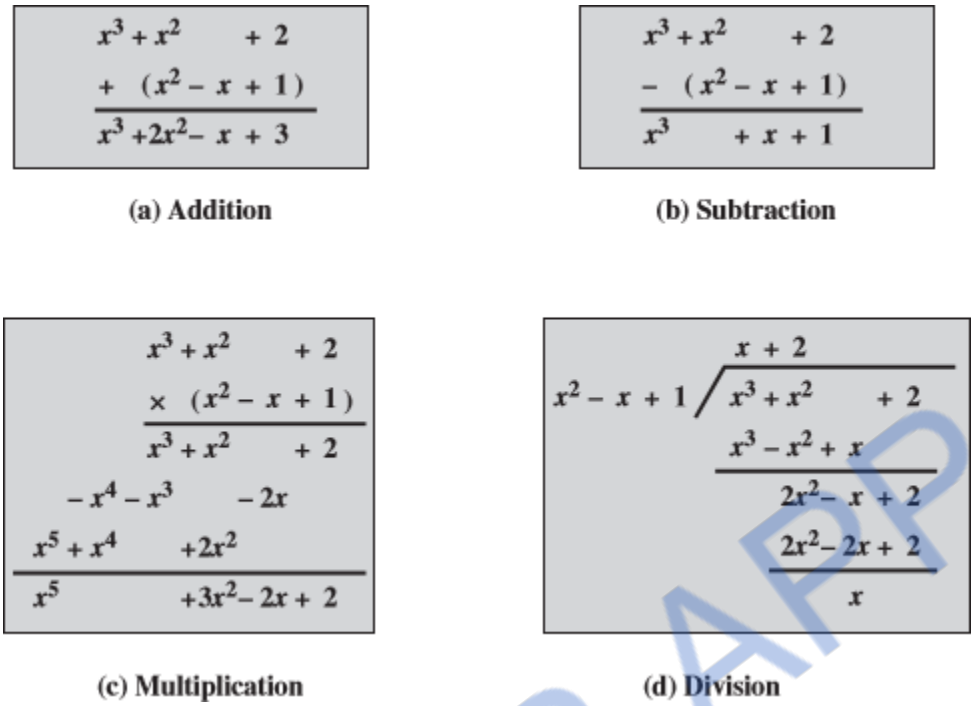


Figure 2.3 Examples of Polynomial Arithmetic

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.

2.6 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:

- $E: K \times M \rightarrow C$
- $D: K \times C \rightarrow M$

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*

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

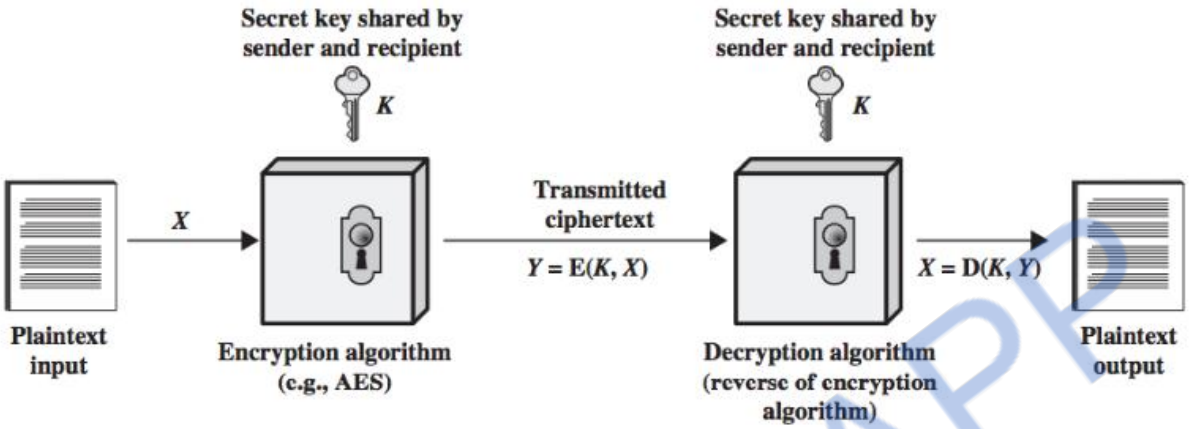


Figure 2.4 Simplified Model of Symmetric Encryption

2.7 SIMPLIFIED DATA ENCRYPTION STANDARD (S-DES)

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

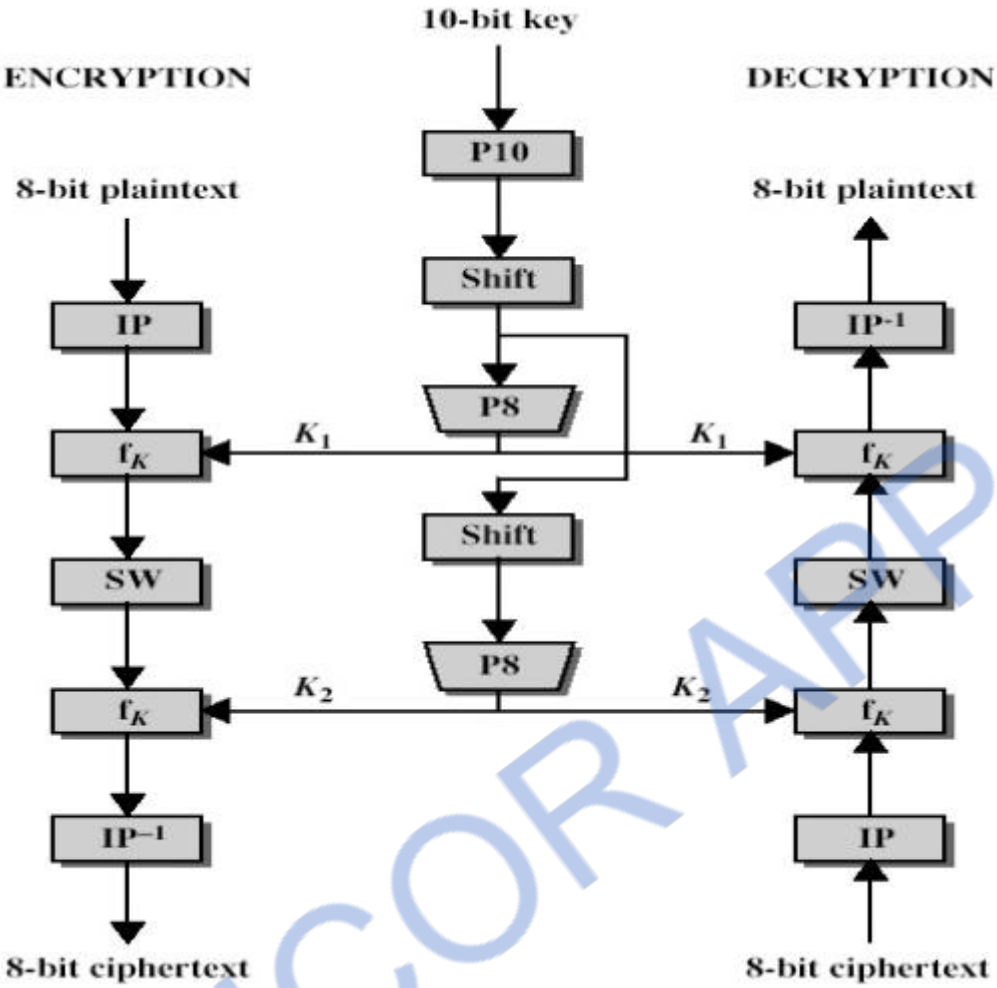


Figure 2.5 Overview of S-DES Algorithm

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).

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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)))))

2.7.1 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)

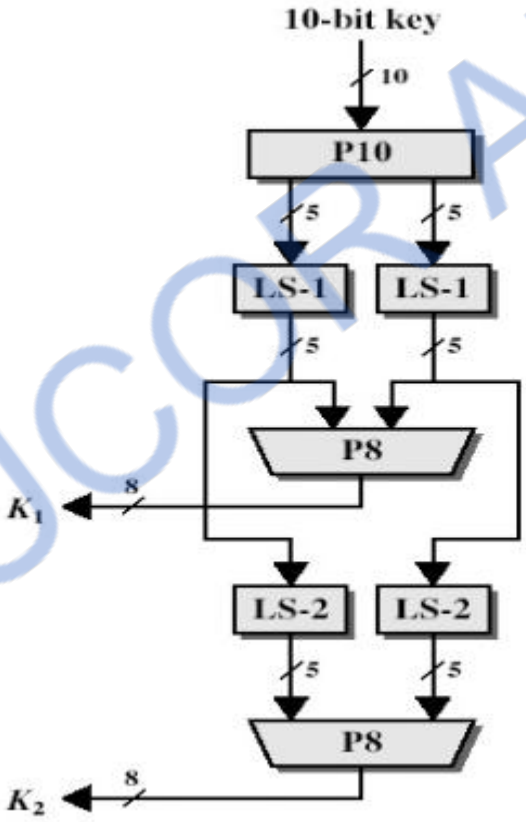


Figure 2.6 S-DES Key Generation

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:

P10									
3	5	2	7	4	10	1	9	8	6

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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:

P8							
6	3	7	4	8	5	10	9

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).

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

2.7.2 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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

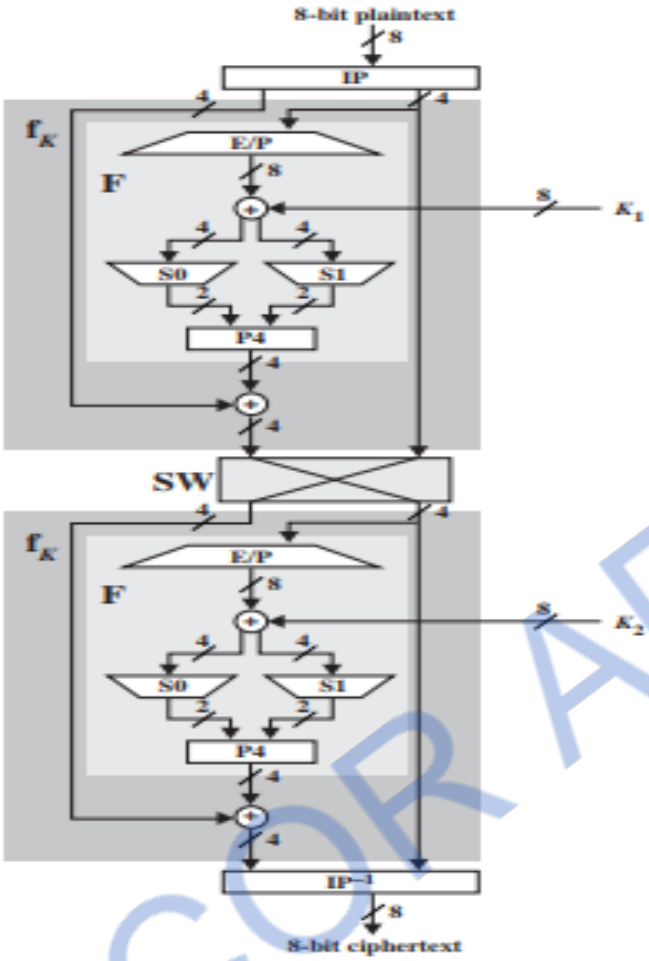


Figure 2.7 S-DES Encryption

2. The Function f_k

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/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

=> 01111101 ⊕10100100 = 11011001

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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:

$$S0 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array} \qquad S1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

The S-boxes operate as follows. The first and fourth input bits are treated as a 2-bit 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,

S_0 => 1101

11 - > 3

10 -> 2 } => 3 =>11

So, we get 1110

- Now, find P_4

Second 4 bits

S1 => 1001

11 -> 3

00 -> 0 = > 2 => 10

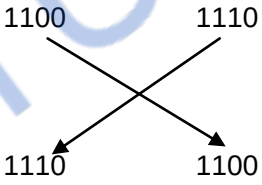
P4			
2	4	3	1

After P_4, the value is 1011

Now, XOR operation 1011⊕ 0111 => 1100

3. The Switch function

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



.

4. Second function f_k

- First, do E/P function and XOR with K_2, the value is 01101001⊕01000011, the answer is 00101010
- Now, find S_0 and S_1

S_0 => 00 - > 0

01 -> 1 } => 0 = 00

S_1 = > 10 -> 2

01 -> 1 } = > 0 => 00

Value is 0000

- Now, find P_4 and XOR operation

After P_4 => 0000 ⊕ 1110 = 1110, then concatenate last 4 bits after interchange in sw.

- Now value is 11101100

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

5. 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 f_k**
   - After step 2, the answer is 11101100
3. **Swift**
   - The answer is 11001110
4. **Second f_k**
   - The answer is 01111110
5. **Find IP-1**
   - **101111101 -> Plaintext**

2.8 DATA ENCRYPTION STANDARD

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.

2.8.1 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.

2.8.2 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

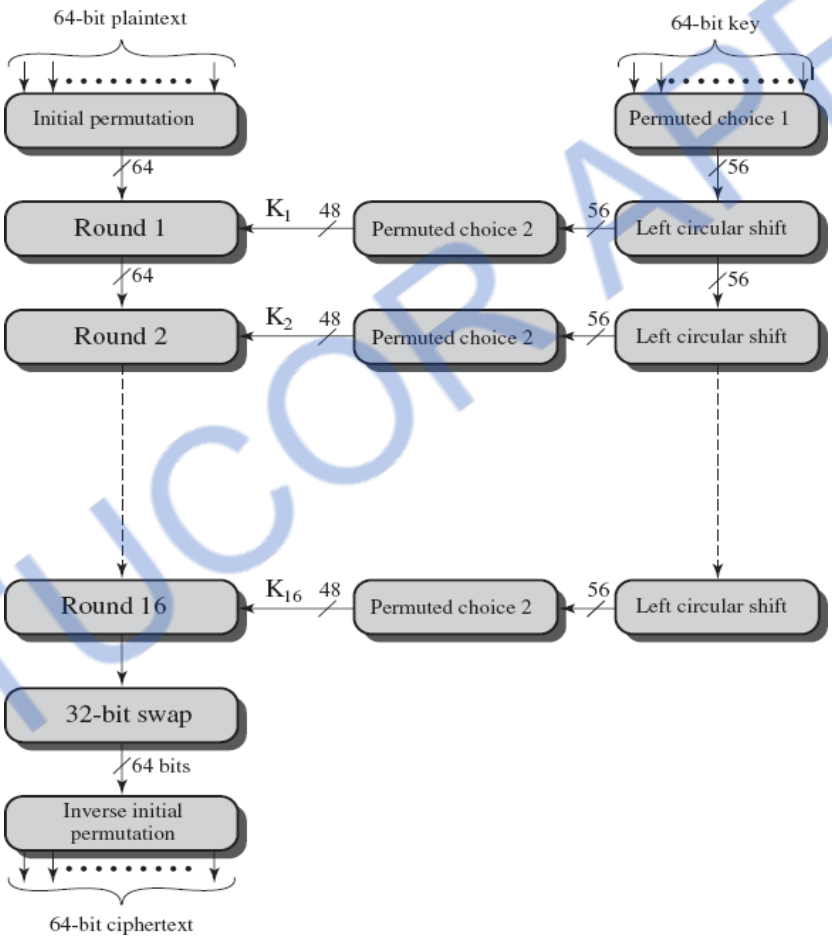


Figure 2.8 DES Encryption Algorithm

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Permutation Tables for DES

(a) Initial Permutation (IP)

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

Inverse Initial Permutation (IP^{-1})

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

Expansion Permutation (E)

32	1	2	3	4	5
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

Permutation Function (P)

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

Consider the following 64-bit input *M*:

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

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

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

2.8.3 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:

$L_i = R_{i-1}$

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

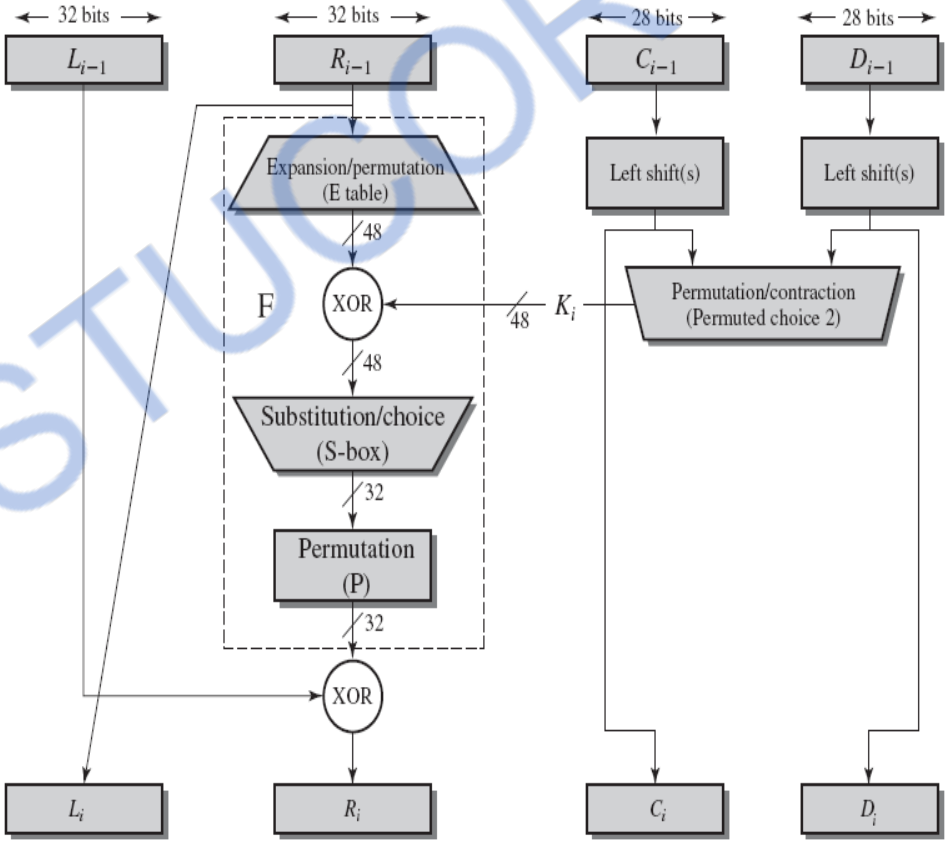


Figure 2.9 Single Round of DES Algorithm

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 S*i* form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for S*i*. 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.

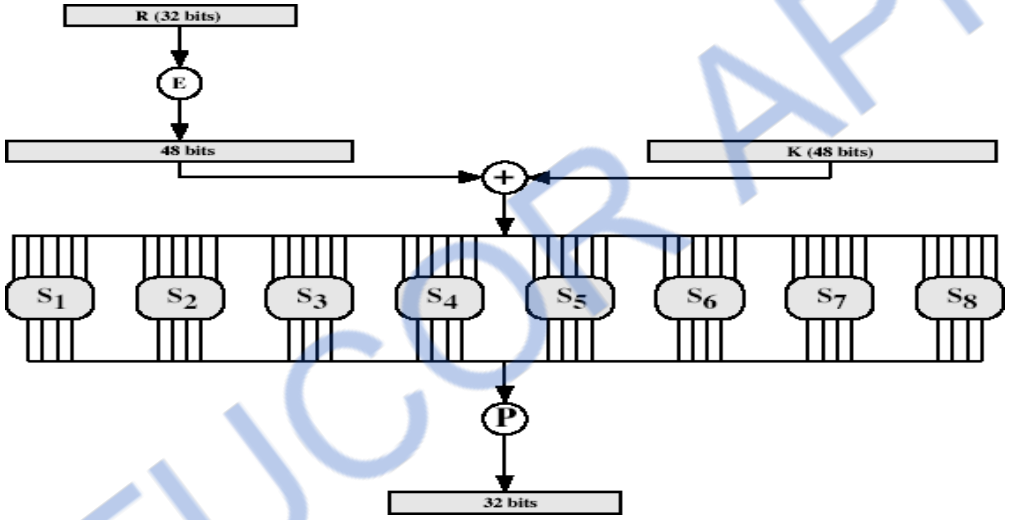


Fig 2.10 Calculation of F(R, K)

2.8.4 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 *C*0 and *D*0.

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*).

DES Key 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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

(b) Permuted Choice 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) Permuted Choice 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

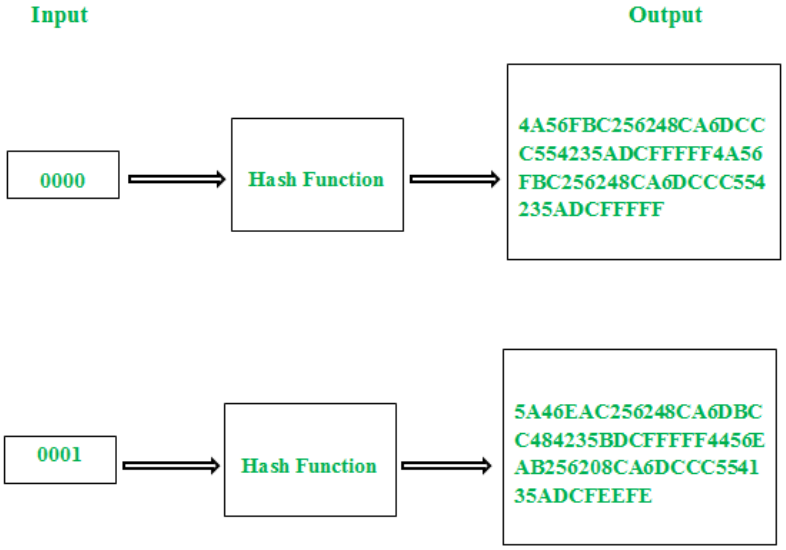
Roundnumber:1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Bits rotated : 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

2.8.5 DES Decryption:

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

2.8.6 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.



ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2.9 THE STRENGTH OF DES

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.

2.9.1 Attacks on DES:

Two approaches are:

1. Differential crypt analysis
2. Linear crypt analysis

2.9.1.1 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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 *m*0, *m*1. Each round of DES maps the right-hand input into the left-hand output and sets the right-hand output to be a function of the left-hand input and the subkey for this round.

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

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

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: $m_i = m_i \oplus m_i'$ Then we have:

$$\begin{aligned}\Delta m_{i+1} &= m_{i+1} \oplus m''_{i-1} \\ &= [m_{i-1} \oplus f(m_i,k_i)] \oplus [m''_{i-1} \oplus f(m''_i,k_i)] \\ &= \Delta m_{i-1} \oplus [f(m_i,k_i) \oplus f(m''_i,ki)]\end{aligned}$$

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.

Therefore, if we know Δm_{i-1} and Δm_i with high probability, then we know Δm_{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.

2.9.1.2 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,j,..k] = A[i] ⊕A[j] ⊕. ⊕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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2.10 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^{55} 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.

2.10.1 Stream Cipher and Block Cipher

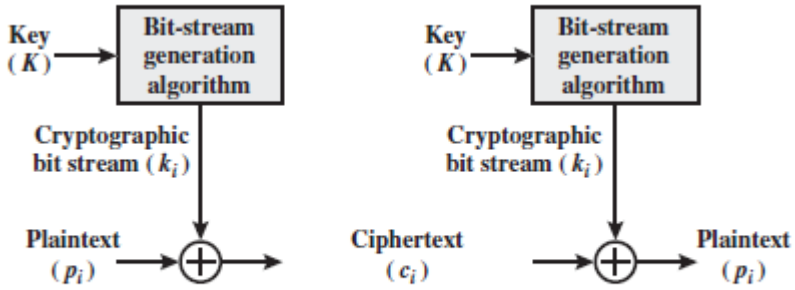
A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
E.g, vigenere cipher. Figure (2.11a)

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)

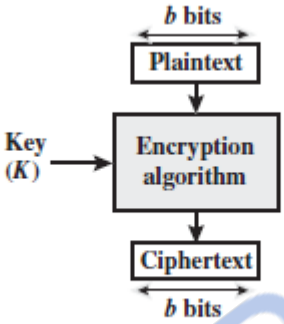
STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

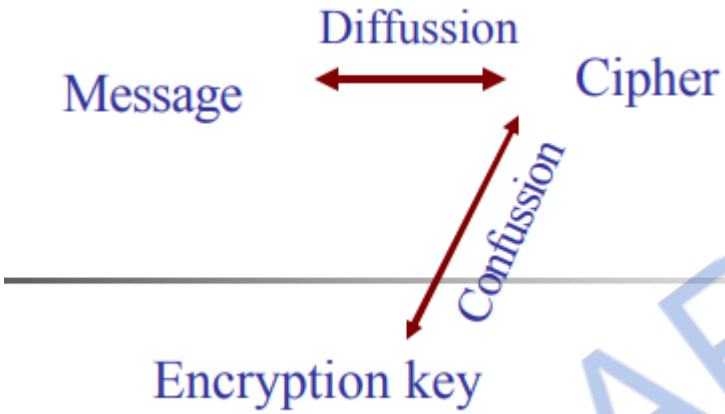
Figure 2.11 Stream Cipher and Block Cipher

- 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- 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.



2.10.2 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 L_0 and R_0.
- The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block. Each round i has inputs L_{i-1} and R_{i-1}, derived from the previous round, as well as the subkey K_i, derived from the overall key K.
- In general, the subkeys K_i 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_i. Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.
- This structure is a particular form of the substitution-permutation network.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

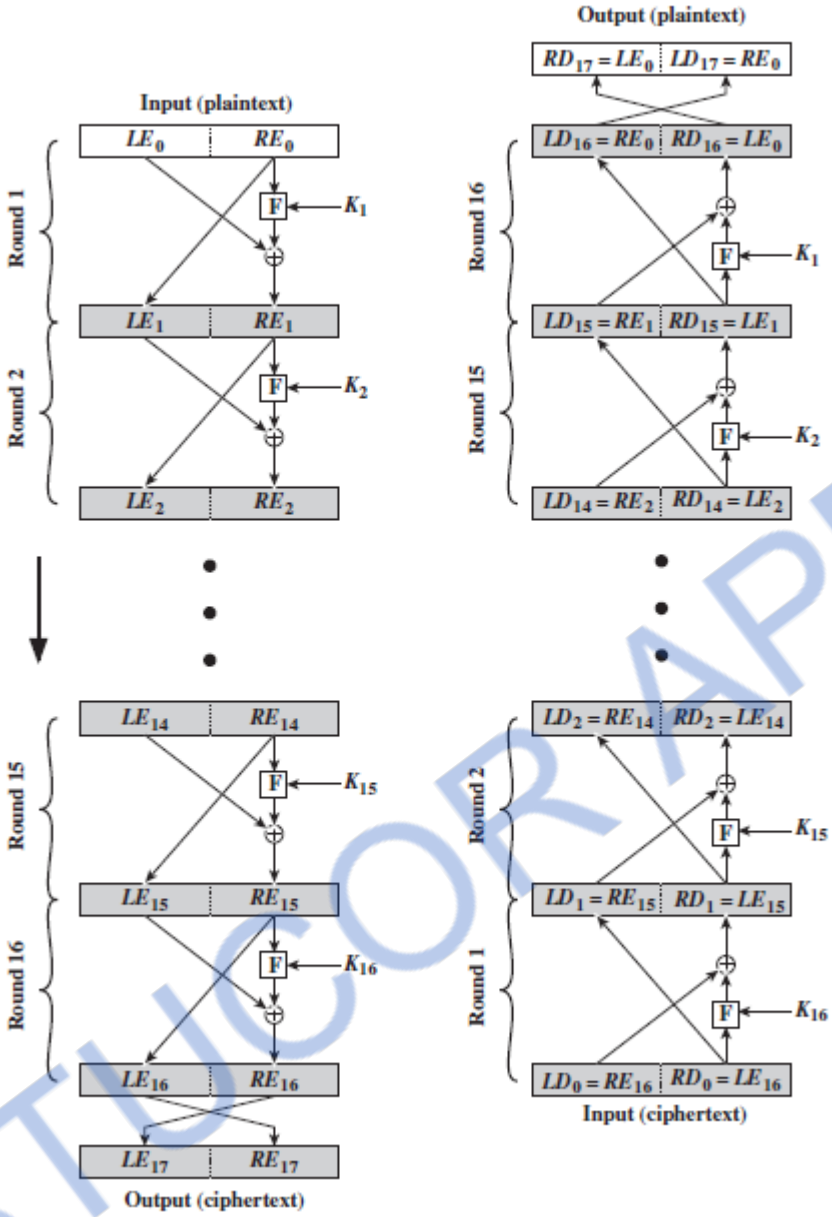


Figure 2.12 Feistel Encryption and Decryption (16 rounds)

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_i in reverse order. i.e., k_n in the first round, k_{n-1} in second round and so on.
- For clarity, we use the notation LE_i and RE_i for data traveling through the decryption algorithm and LD_i and RD_i.
- 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

i.e., $RE_i \| LE_i$ (or) equivalently $RD_{16-i} \| LD_{16-i}$

- After the last iteration of the encryption process, the two halves of the output are swapped, so that the cipher text is $RE_{16} \| LE_{16}$.
- 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 $RE_{16} \| LE_{16,}$ 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,

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

On the decryption side,

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$
$$RD_1 = LD_0 \oplus F\,(RD_O, K_{16})$$
$$= RE_{16} \oplus F\,(RE_{15}, K_{16})$$
$$= [LE_{15} \oplus F\,(RE_{15}, K_{16})] \oplus F\,(RE_{15}, K_{16})$$
$$= LE_{15}$$

Therefore, $LD_1 = RE_{15}$, $RD_1 = LE_{15}$

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

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

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

2.11 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

2.11.1 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:

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

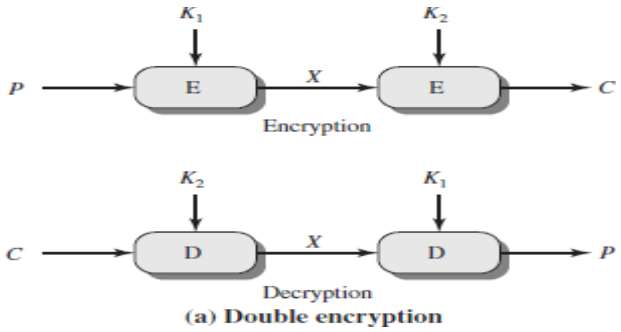


Figure 2.13 Multiple Encryption

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

$$\mathrm{E}(K_2, \mathrm{E}(K_1, P)) = \mathrm{E}(K_3, P)$$

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

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

Given a known pair, (P, C), the attack proceeds as follows. First, encrypt P for all 256 possible values of K_1. 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

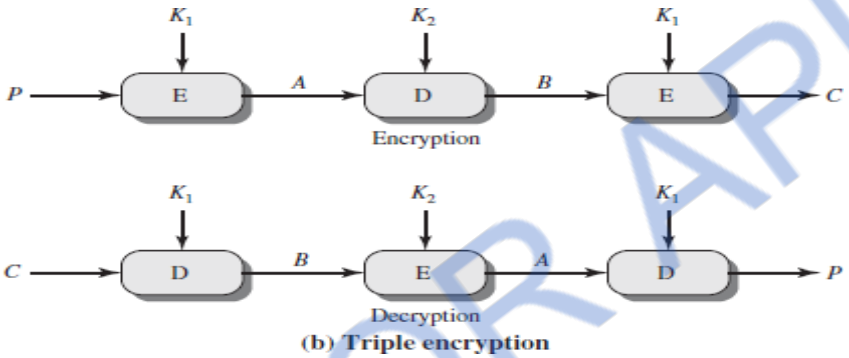
$$C = \mathrm{E}(K_1, \mathrm{D}(K_2, \mathrm{E}(K_1, P)))$$
$$P = \mathrm{D}(K_1, \mathrm{E}(K_2, \mathrm{D}(K_1, C)))$$

- 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:

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

- 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^{112}



(b) Triple encryption

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^{56},
- 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_i that matches an entry in Table 1, create an entry in Table 2 consisting of the K_1 value and the value of B that is produced.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Step 3:

- We now have a number of candidate values of K_1 in Table 2 and are in a position to search for a value of K_2.
- For each of the 256 possible keys K_2 = 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.

2.11.2 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 isa 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.

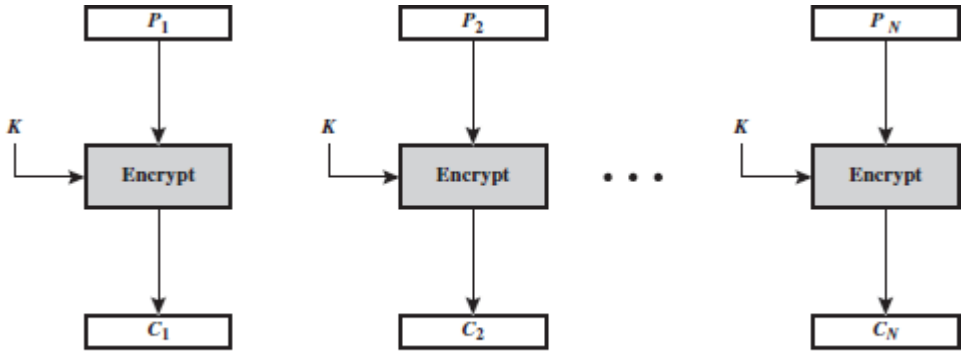
Disadvantage:

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.

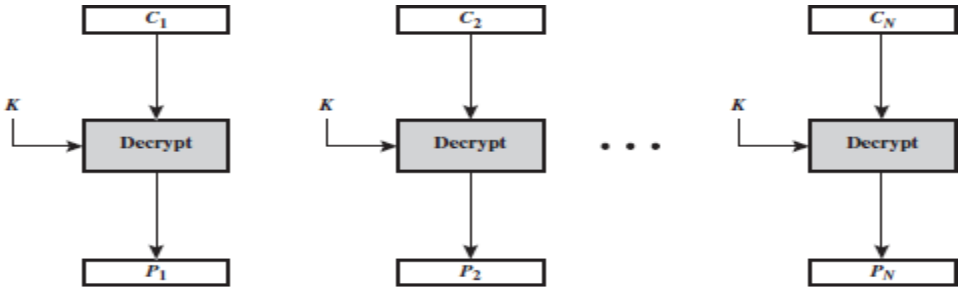
This may help in the analysis or may provide an opportunity for substituting or rearranging blocks.



(a) Encryption

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP



(b) Decryption

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.

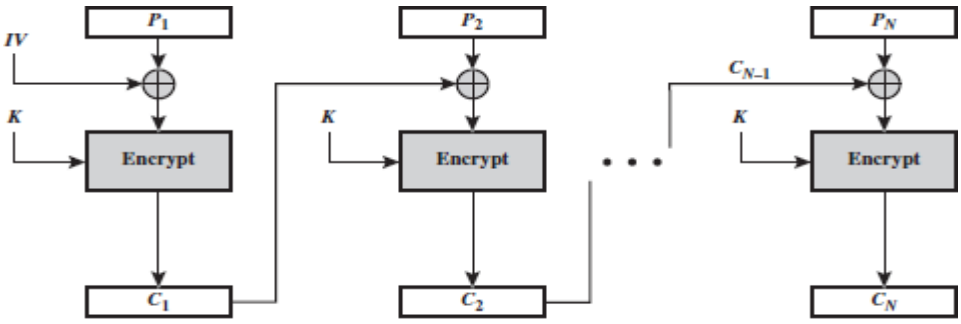
2.11.3 MODE 2: Cipher Block Chaining Mode

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.

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

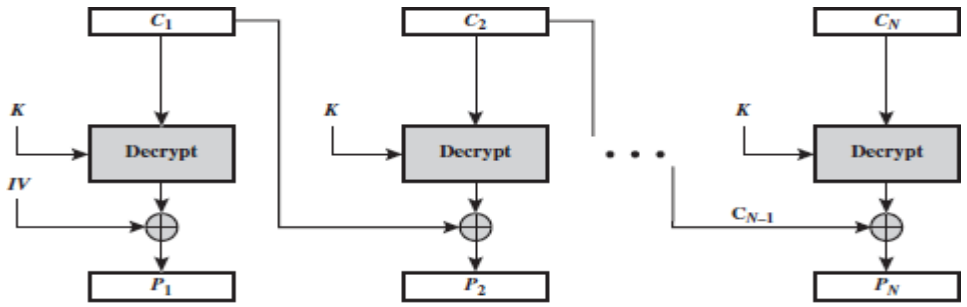


(a) Encryption

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP



(b) Decryption

Figure 2.16 Cipher Block Chaining (CBC) Mode

Then

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

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

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

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2.11.4 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

$$C_1 = P_1 \oplus \text{MSB}_s[\text{E}(K, \text{IV})]$$

Therefore, by rearranging terms:

$$P_1 = C_1 \oplus \text{MSB}_s[\text{E}(K, \text{IV})]$$

The same reasoning holds for subsequent steps in the process.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

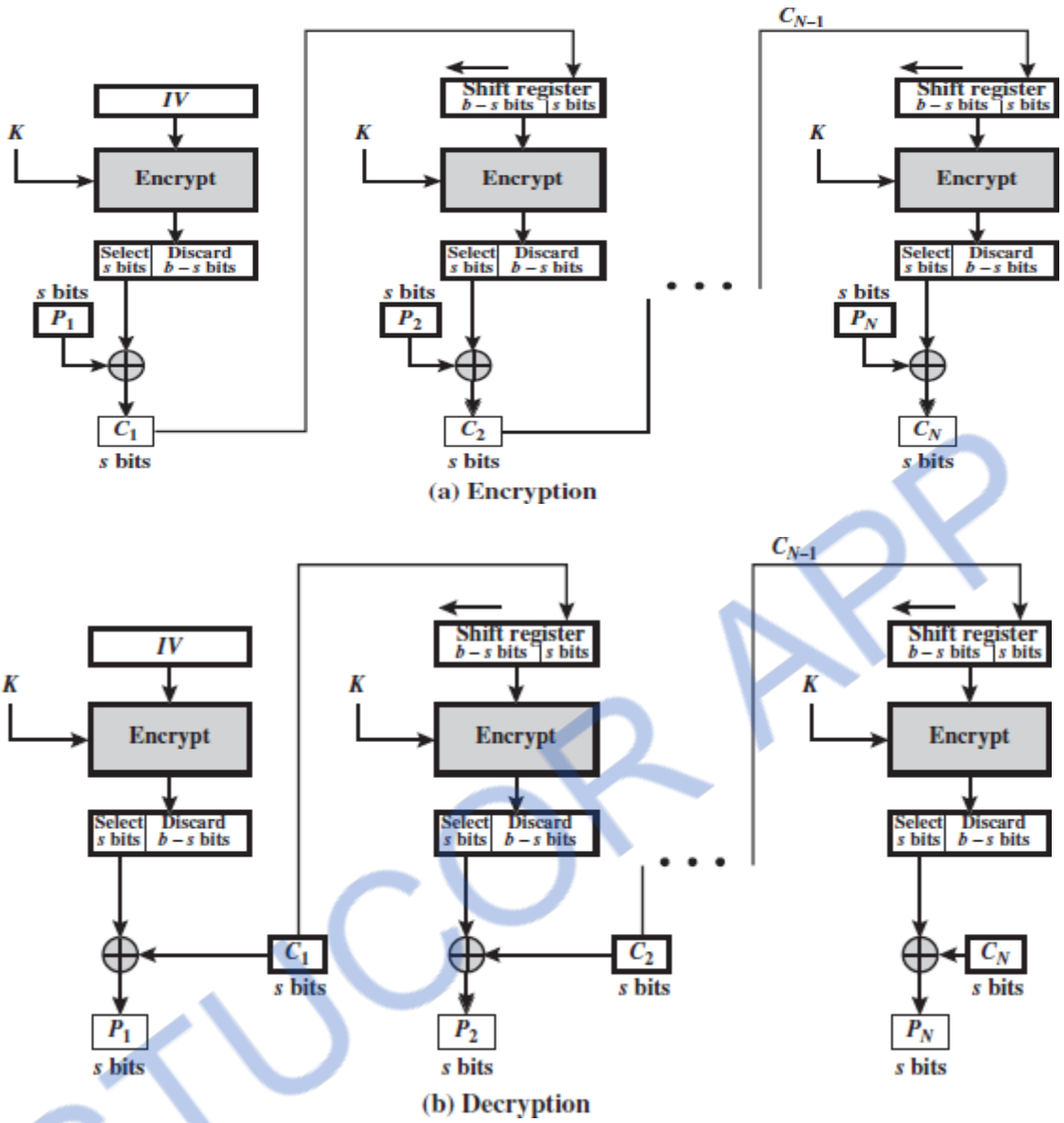


Figure 2.17 S-bit Cipher Feedback (CFB) mode

We can define CFB mode as follows

CFB	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $C_j = P_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $P_j = C_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$

2.11.5 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

$$O_{j-1} = \mathrm{E}(K, O_{j-2})$$

we can rewrite the encryption expression as:

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

By rearranging terms, we can demonstrate that decryption works.

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

We can define OFB mode as follows.

OFB	$I_1 = Nonce$ $I_j = O_{j-1}$ $j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j)$ $j = 1, \ldots, N$ $C_j = P_j \oplus O_j$ $j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$	$I_1 = Nonce$ $I_j = O_{j-1}$ $j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j)$ $j = 1, \ldots, N$ $P_j = C_j \oplus O_j$ $j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u(O_N)$

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

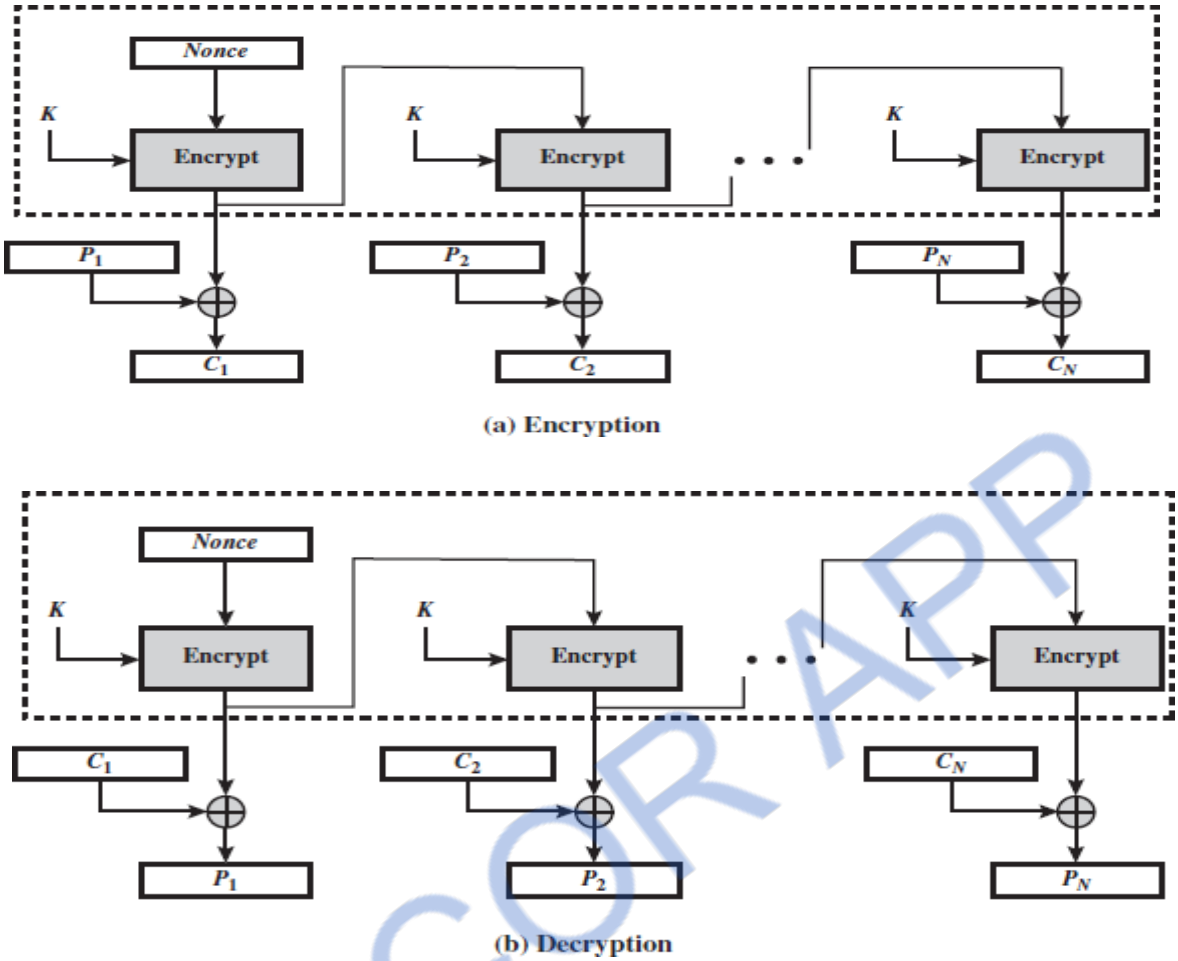


Figure 2.18 Output Feedback Mode

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

2.11.6 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

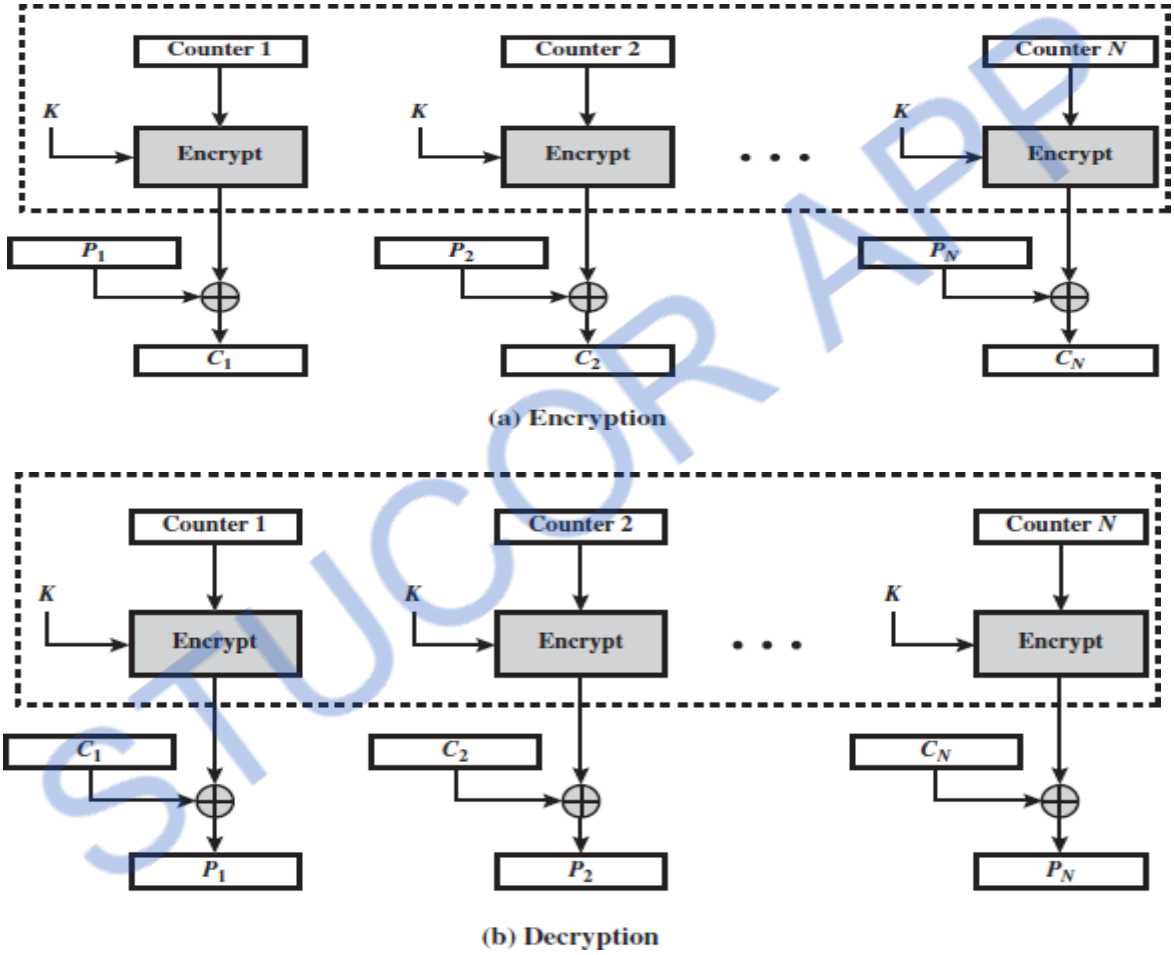


Figure 2.19 Counter Mode

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2.12 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.

2.12.1 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 Z*p* 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^nelements; 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

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

Where each a_i takes on the value 0 or 1. There are a total of 2^ndifferent polynomials in *S*.
For *n* = 3, the 2^3 = 8 polynomials in the set are

$$\begin{array}{cccc} 0 & x & x^2 & x^2 + x \\ 1 & x + 1 & x^2 + 1 & x^2 + x + 1 \end{array}$$

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(2*n*) can be uniquely represented by its *n* binary coefficients(*an*-1*an*-2 c*a*0). Therefore, every polynomial in GF(2*n*) can be represented by an *n*-bit number.

2.12.2 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, orAES-256, depending on the key length.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- 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 round, 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

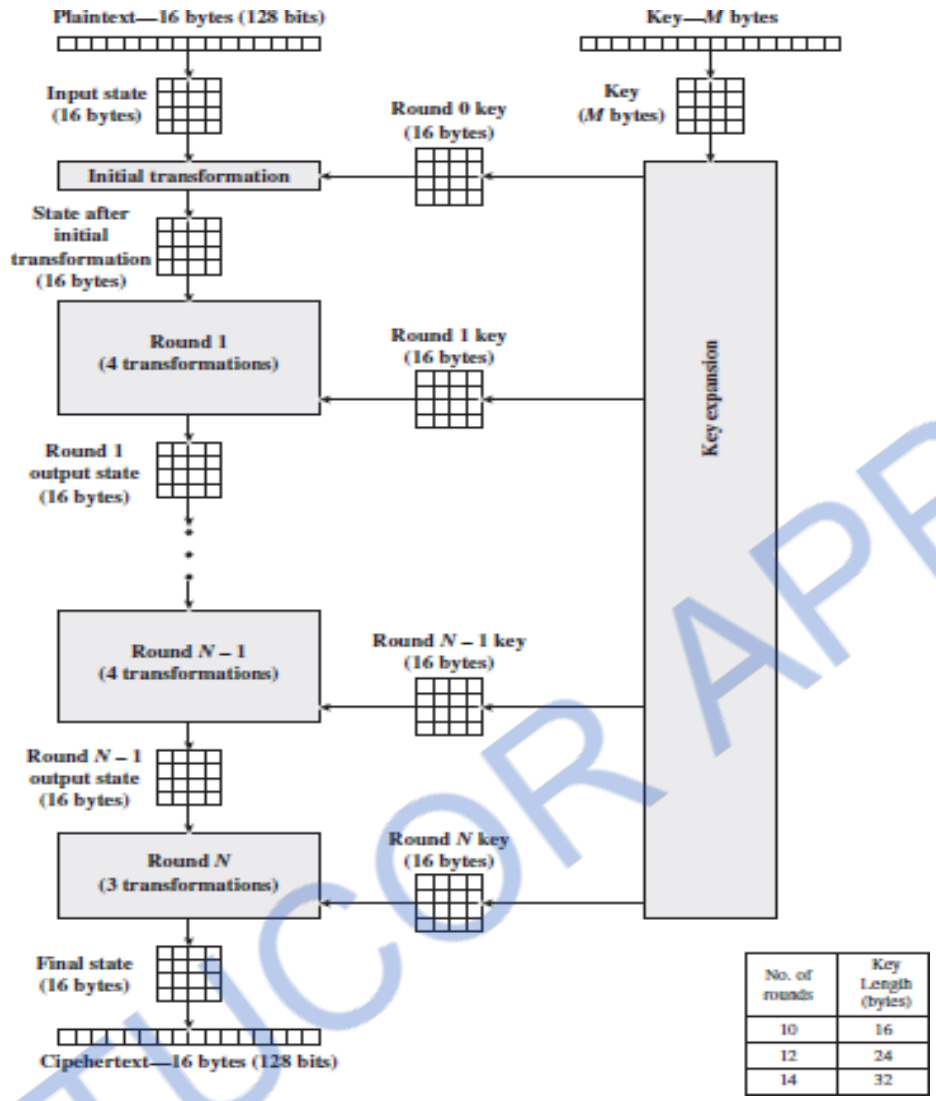


Figure 2.20 AES Encryption Process

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

2.12.3 Detailed Structure

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

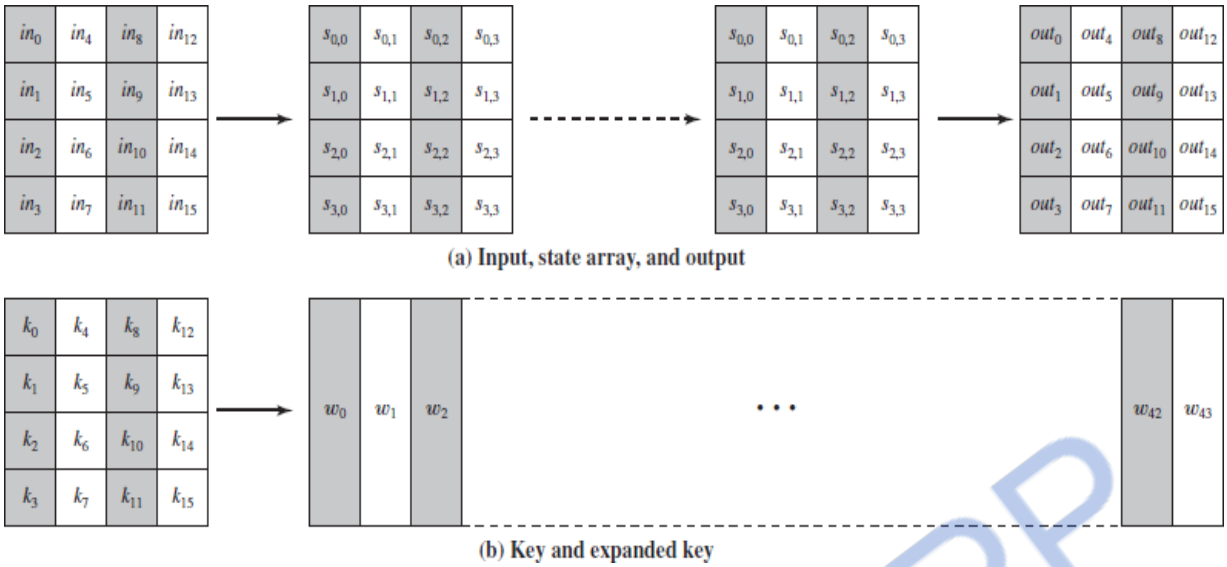


Fig: 2.21 Detail AES structure

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.
2. 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 ofthe 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

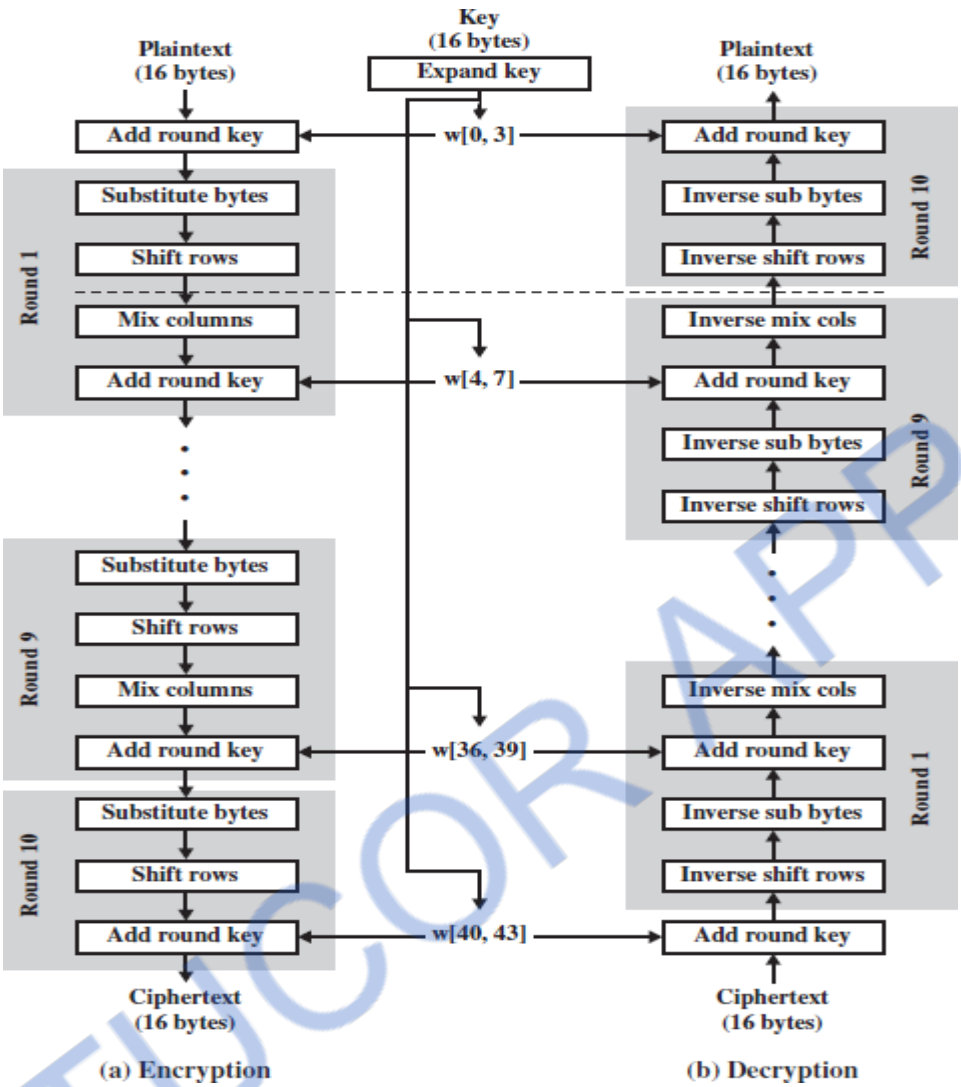


Fig 2.22 AES Encryption and Decryption

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

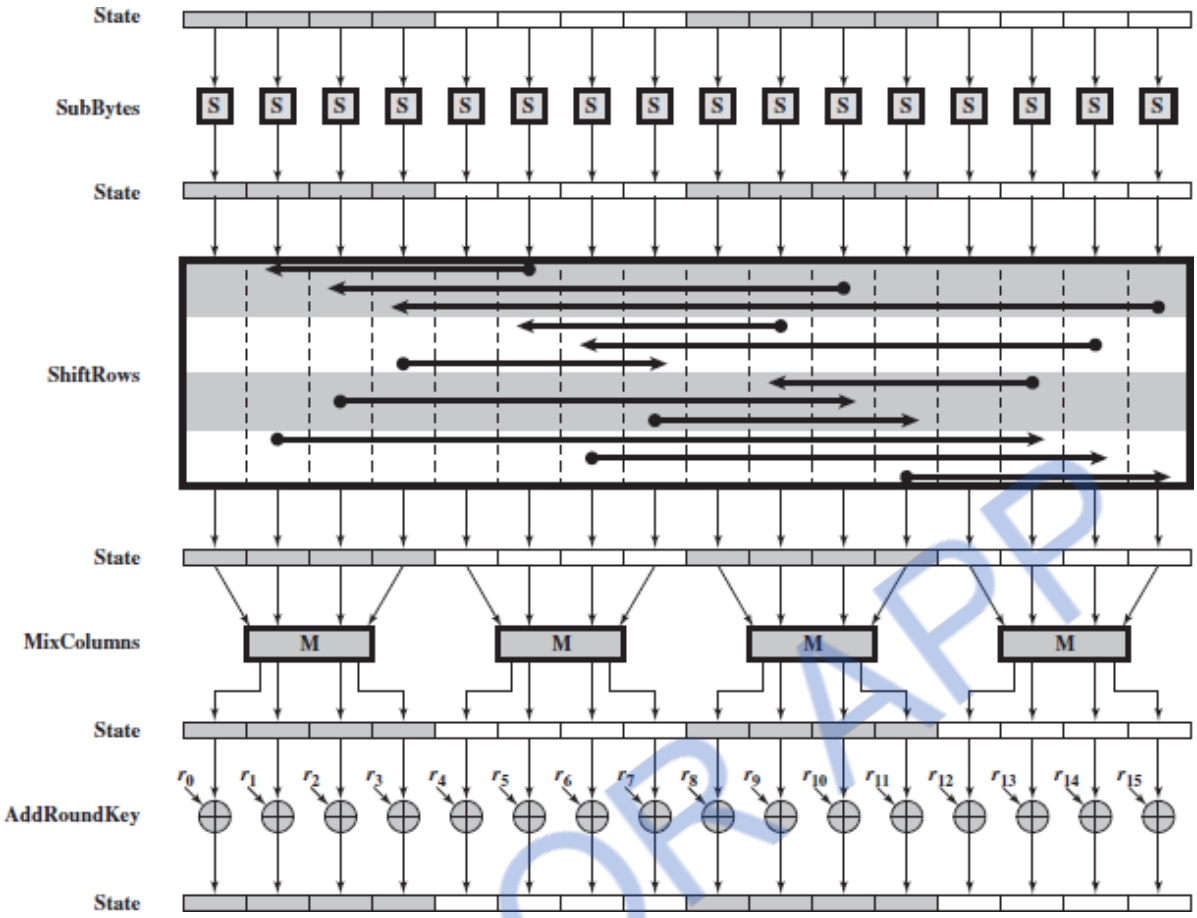


Fig 2.23 AES Encryption Round

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

2.12.4 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*:*

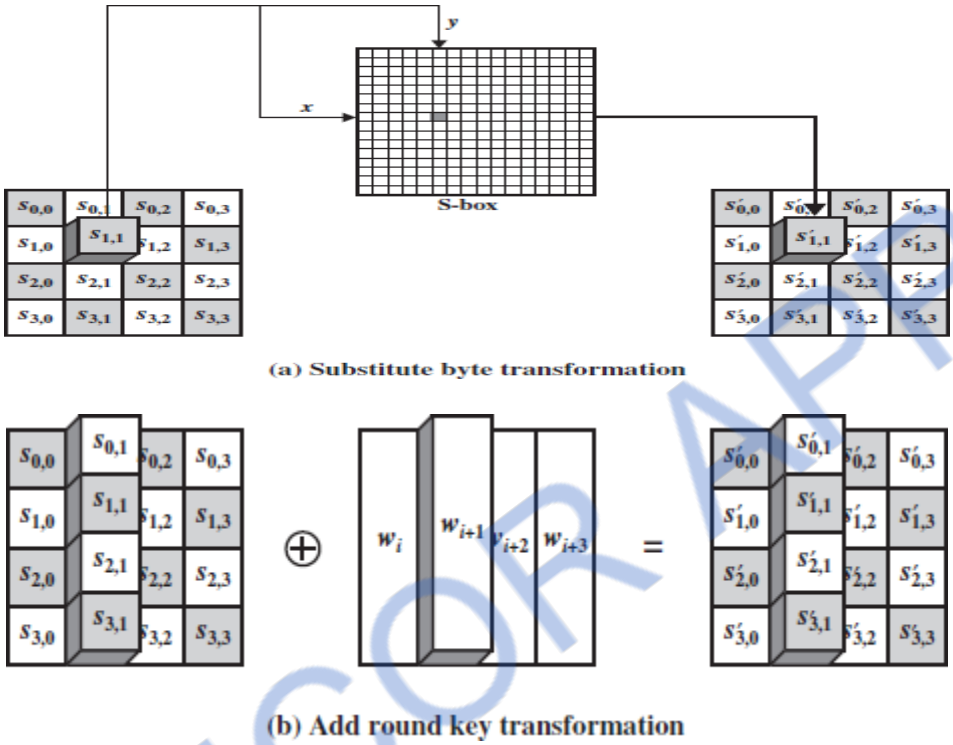
***The* forward substitute byte transformation**, called Sub Bytes, is a simple table lookup (Figure 2.24a). AES defines a 16 * 16 matrix of byte values, called an S-box that contains a permutation of all possible 256 8-bit values.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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}.



(a) Substitute byte transformation

(b) Add round key transformation

Figure 2.24 AES Byte level Operations

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	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	B3	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
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
B	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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	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	D9	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	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
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
B	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	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
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 *i*th 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

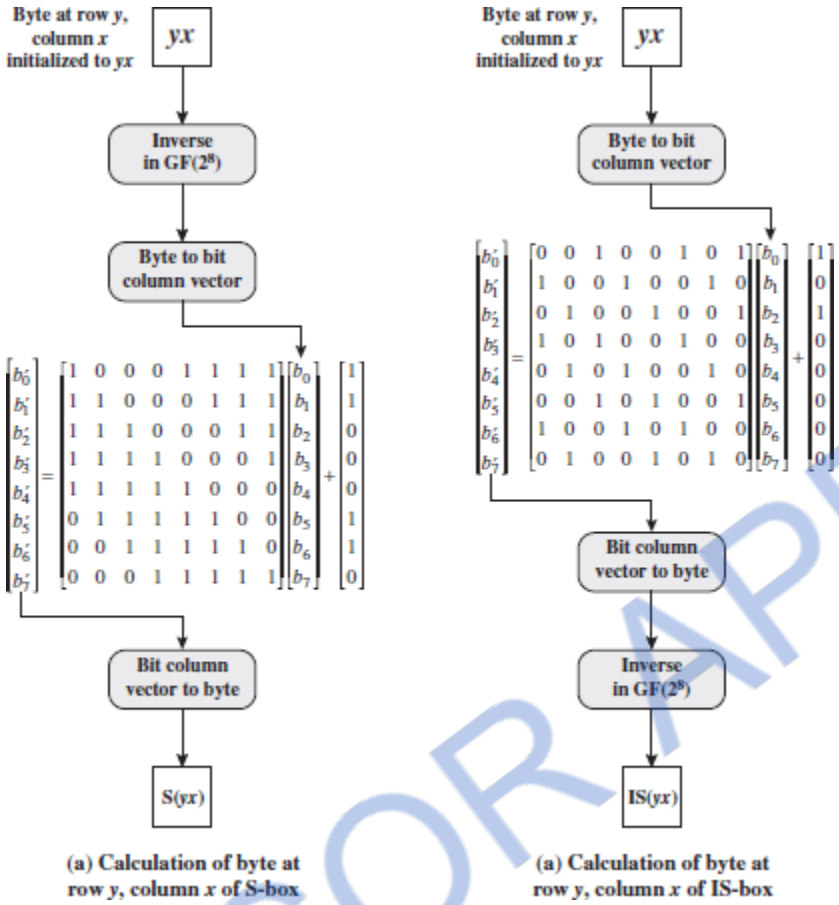


Figure 2.26 Construction of S-Box and IS-Box

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

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 $\{95\}^{-1} = \{8A\}$, which is 10001010 in binary. Using above Equation

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

CS8792 -
DOWNLOADED FROM STUCOR APP

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

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.

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

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 $\mathbf{B'} = \mathbf{XB} \oplus \mathbf{C}$. We need to show that $\mathbf{Y(XB} \oplus \mathbf{C)} \oplus \mathbf{D} = \mathbf{B}$**.** To multiply out, we must show$\mathbf{YXB} \oplus \mathbf{YC} \oplus \mathbf{D} = \mathbf{B}$**.** This becomes

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

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

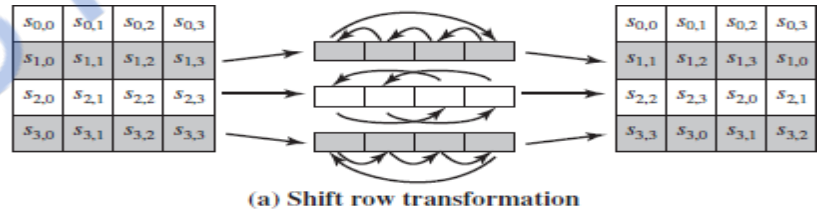
87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

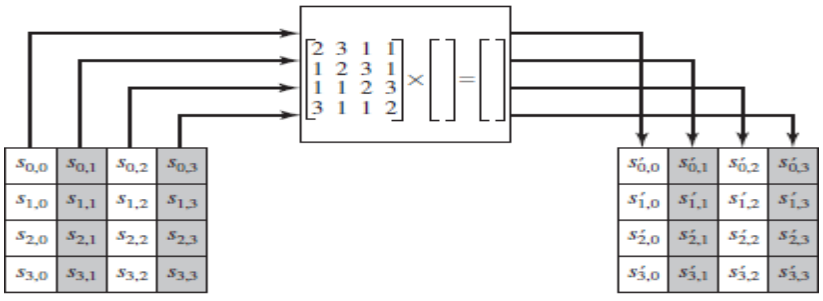
87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

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



(a) Shift row transformation



(b) Mix column transformation

Figure 2.28 AES Row and Column Operations

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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(2^8).

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

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

The following is an example of MixColumns:

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

$$\begin{aligned} (\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} &= \{47\} \\ \{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} &= \{37\} \\ \{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) &= \{94\} \\ (\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) &= \{ED\} \end{aligned}$$

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

$$\begin{aligned} \{02\} \cdot \{87\} &= 0001\ 0101 \\ \{03\} \cdot \{6E\} &= 1011\ 0010 \\ \{46\} &= 0100\ 0110 \\ \{A6\} &= \underline{1010\ 0110} \\ & \quad 0100\ 0111 = \{47\} \end{aligned}$$

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

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

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}⊕{09}.{02} = 00001001 ⊕ 00010010 =00011011then

$$\begin{aligned} \{0E\} \cdot \{02\} &= 00011100 \\ \{0B\} &= 00001011 \\ \{0D\} &= 00001101 \\ \{09\} \cdot \{03\} &= \underline{00011011} \\ & \quad\; 00000001 \end{aligned}$$

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

⊕

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

=

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

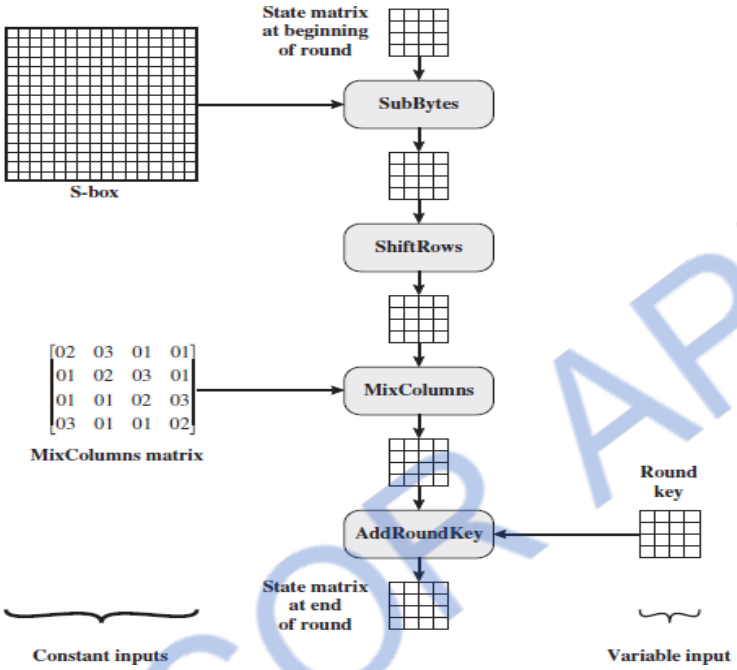


Fig 2.29 AES Key Expansion

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 **w**[i]depends on the immediately preceding word, **w**[i - 1], and the word four positions back, **w**[i - 4]. In three out of four cases, a simple XOR is used. For a word whose position in the **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

```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++)    w[i] = (key[4*i], key[4*i+1],
                                       key[4*i+2],
                                       key[4*i+3]);

     for (i = 4; i < 44; i++)
     {
      temp = w[i - 1];
      if (i mod 4 = 0)    temp = SubWord (RotWord (temp))
                                 ⊕ Rcon[i/4];
      w[i] = w[i-4] ⊕ temp
     }
}
```

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

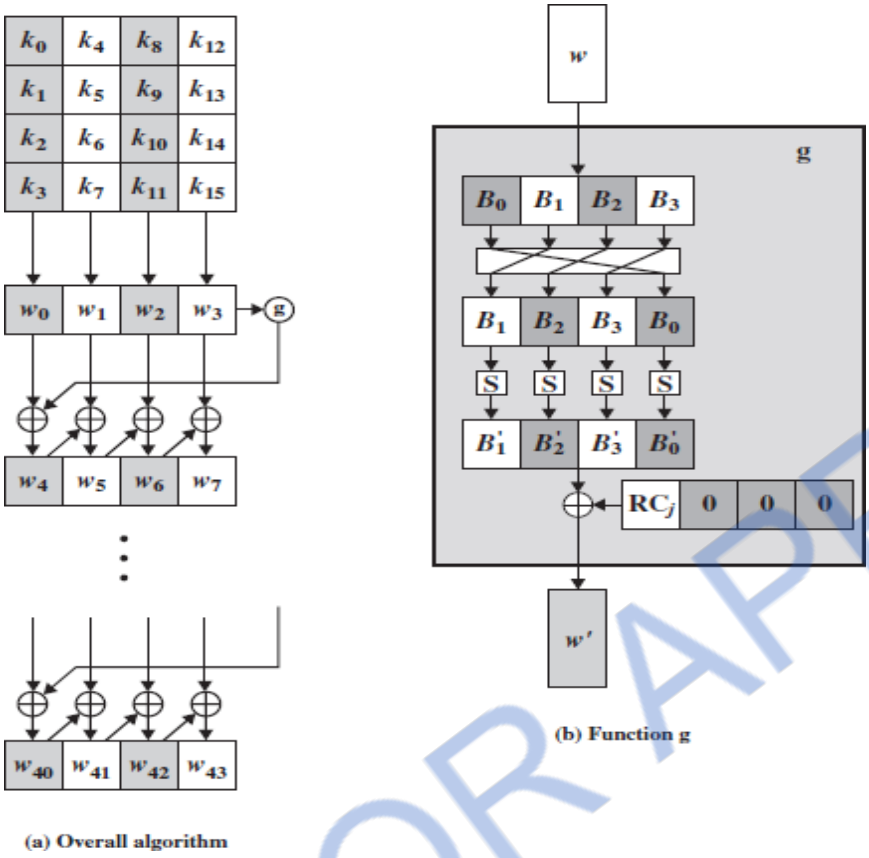


Figure 2.30 Key Expansion Algorithm

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 ⊕ w[i−4]
36	7F8D292F	8D292F7F	5DA515D2	1B000000	46A515D2	EAD27321	AC7766F3

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Key Words	Auxiliary Function
w16 = w12 ⊕ z4 = 2c 5c 65 f1 w17 = w16 ⊕ w13 = a5 73 0e 96 w18 = w17 ⊕ w14 = f2 22 a3 90 w19 = w18 ⊕ w15 = 43 8c dd 50	RotWord(w19) = 8c dd 50 43 = x5 SubWord(x5) = 64 c1 53 1a = y5 Rcon(5) = 10 00 00 00 y5 ⊕ Rcon(5) = 74 c1 53 1a = z5
w20 = w16 ⊕ z5 = 58 9d 36 eb w21 = w20 ⊕ w17 = fd ee 38 7d w22 = w21 ⊕ w18 = 0f cc 9b ed w23 = w22 ⊕ w19 = 4c 40 46 bd	RotWord (w23) = 40 46 bd 4c = x6 SubWord (x6) = 09 5a 7a 29 = y6 Rcon(6) = 20 00 00 00 y6 ⊕ Rcon(6) = 29 5a 7a 29 = z6
w24 = w20 ⊕ z6 = 71 c7 4c c2 w25 = w24 ⊕ w21 = 8c 29 74 bf w26 = w25 ⊕ w22 = 83 e5 ef 52 w27 = w26 ⊕ w23 = cf a5 a9 ef	RotWord (w27) = a5 a9 ef cf = x7 SubWord (x7) = 06 d3 bf 8a = y7 Rcon (7) = 40 00 00 00 y7 ⊕ Rcon(7) = 46 d3 df 8a = z7
w28 = w24 ⊕ z7 = 37 14 93 48 w29 = w28 ⊕ w25 = bb 3d e7 f7 w30 = w29 ⊕ w26 = 38 d8 08 a5 w31 = w30 ⊕ w27 = f7 7d a1 4a	RotWord (w31) = 7d a1 4a f7 = x8 SubWord (x8) = ff 32 d6 68 = y8 Rcon (8) = 80 00 00 00 y8 ⊕ Rcon(8) = 7f 32 d6 68 = z8
w32 = w28 ⊕ z8 = 48 26 45 20 w33 = w32 ⊕ w29 = f3 1b a2 d7 w34 = w33 ⊕ w30 = cb c3 aa 72 w35 = w34 ⊕ w32 = 3c be 0b 3	RotWord (w35) = be 0b 38 3c = x9 SubWord (x9) = ae 2b 07 eb = y9 Rcon (9) = 1B 00 00 00 y9 ⊕ Rcon (9) = b5 2b 07 eb = z9
w36 = w32 ⊕ z9 = fd 0d 42 cb w37 = w36 ⊕ w33 = 0e 16 e0 1c w38 = w37 ⊕ w34 = c5 d5 4a 6e w39 = w38 ⊕ w35 = f9 6b 41 56	RotWord (w39) = 6b 41 56 f9 = x10 SubWord (x10) = 7f 83 b1 99 = y10 Rcon (10) = 36 00 00 00 y10 ⊕ Rcon (10) = 49 83 b1 99 = z10
w40 = w36 ⊕ z10 = b4 8e f3 52 w41 = w40 ⊕ w37 = ba 98 13 4e w42 = w41 ⊕ w38 = 7f 4d 59 20 w43 = w42 ⊕ w39 = 86 26 18 76	

Table 2.5 progression of State through the AES encryption process

2.13 RC4 ALGORITHM

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.

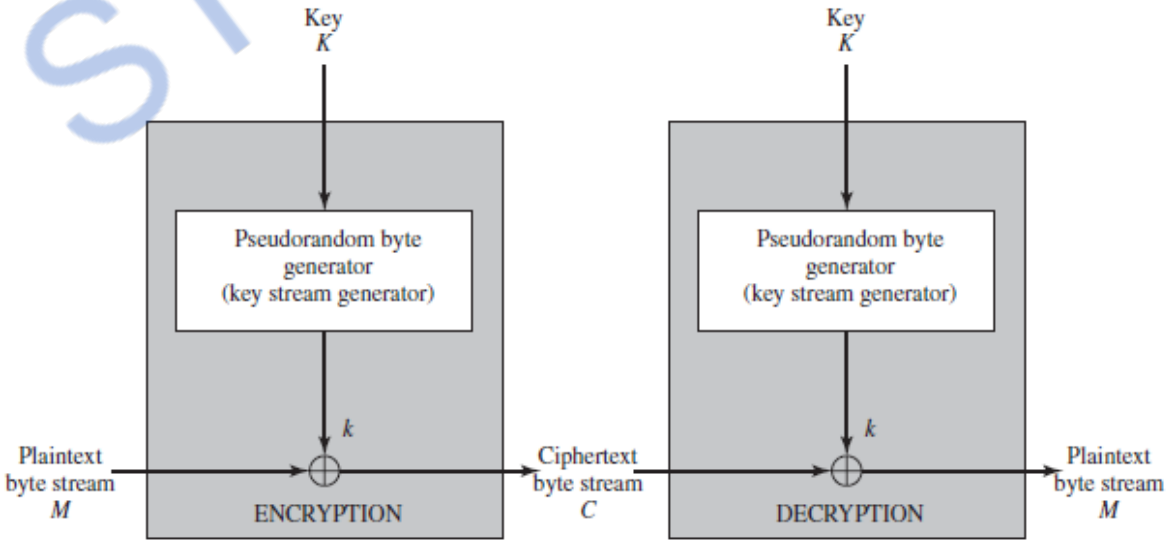


Figure 2.31 Stream Cipher Diagram

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Example

RC4 Encryption		RC4 Decryption	
10011000	Plaintext	11001000	Ciphertext
⊕ 01010000	Key Stream	⊕ 01010000	Key Stream
11001000	Ciphertext	10011000	Plaintext

2.13.1 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;
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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

```
//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];
```

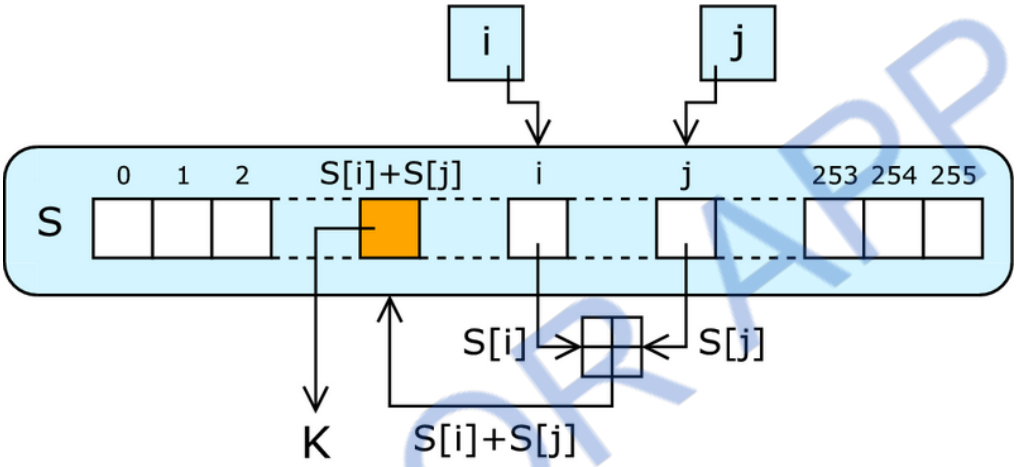


Figure 2.32 PRGA Algorithm

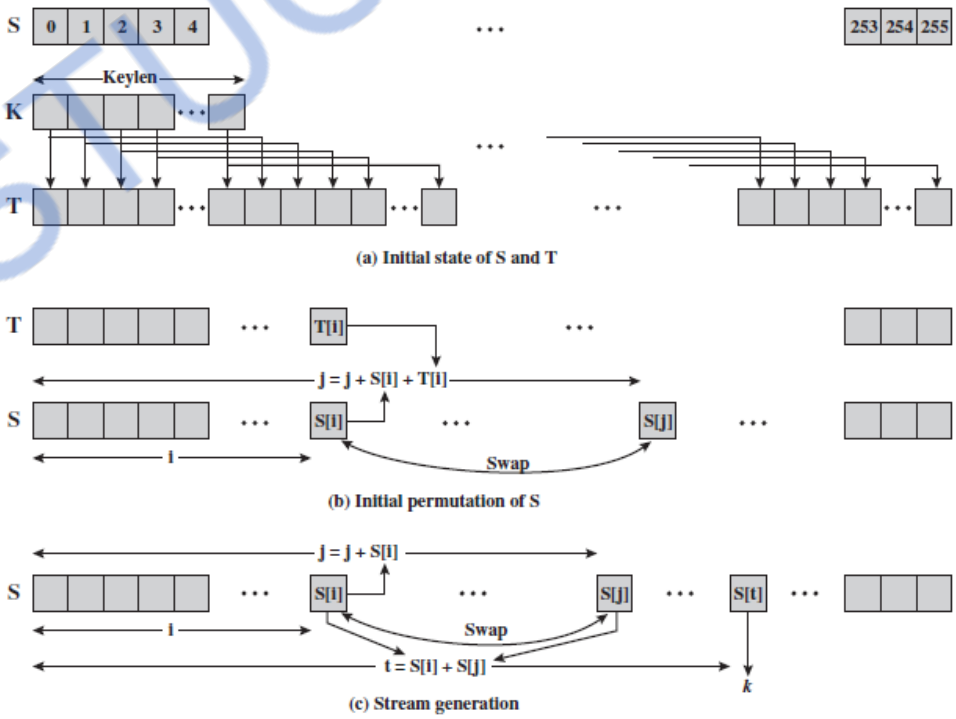


Figure 2.33 RC4 Algorithm

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Encrypt using XOR

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

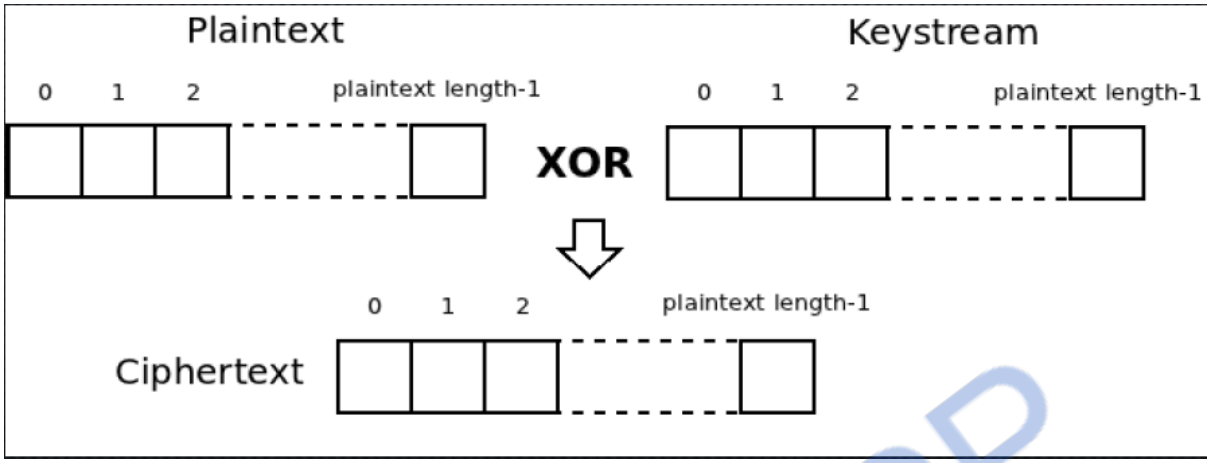


Figure 2.34 RC4 Encryption

Decrypt using XOR

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

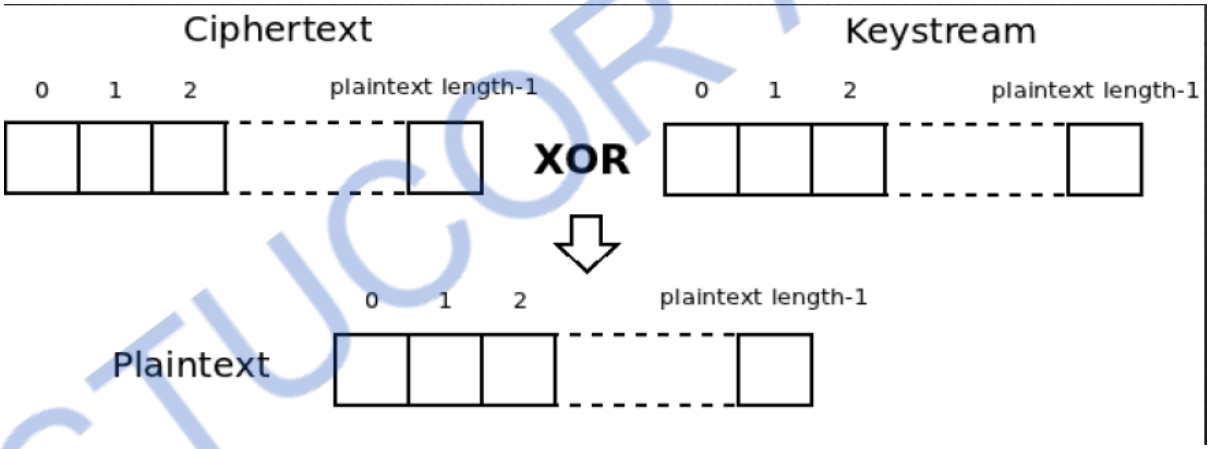


Figure 2.35 RC4 Decryption

Advantage

- It is faster and more suitable for streaming application

2.14 Key Distribution

2.14.1 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:

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.
3. 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.
4. 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.

2.14.2 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.

2.14.3 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_a, known only to itself and the KDC; similarly, B shares the master key K_b with the KDC (Figure 2.36). The following steps occur:

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

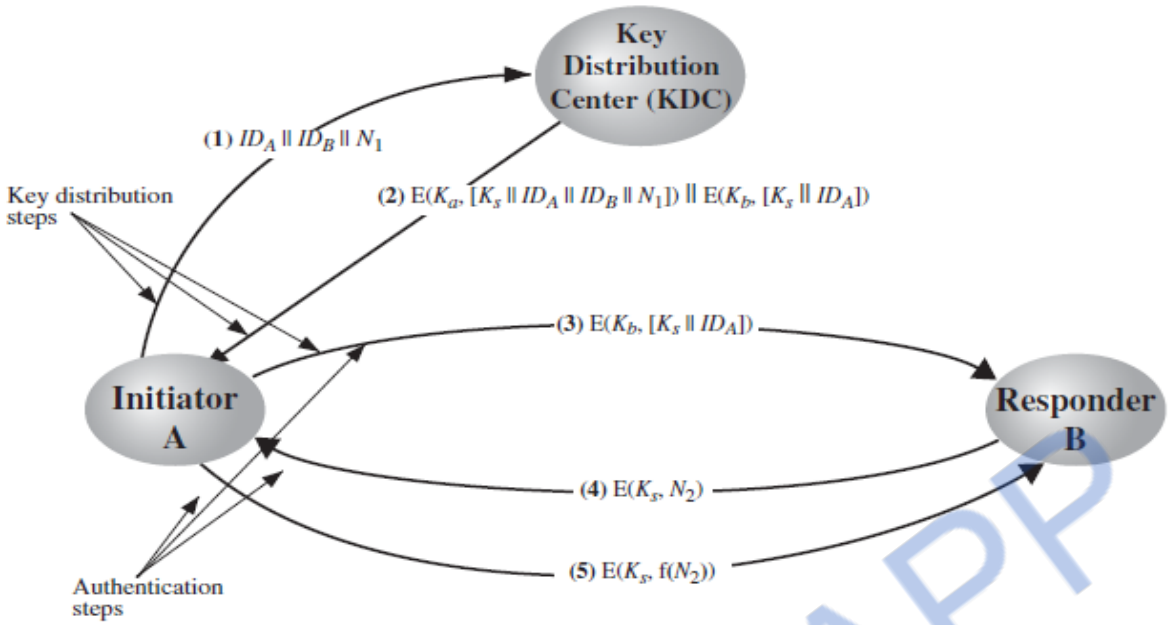


Figure 2.36 Key Distribution Scenarios

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_1, 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.

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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_b, [K_s || ID_A]). Because this information is encrypted with K_b, it is protected from eavesdropping. B now knows the session key (K_s), knows that the other party is A (from ID_A), and knows that the information originated at the KDC (because it is encrypted using K_b). 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, N_2, 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).

2.14.4 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.

STUCOR APP
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

UNIT III PUBLIC KEY CRYPTOGRAPHY 9 MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY

TOPICS: 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.

PRIME NUMBERS

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., $a^p \equiv a \pmod P$

In special case

$a^{P-1} \equiv 1 \pmod P$ if GCD(a,P)=1 . where a and p are coprime.

It is mainly used to solve modular exponentiation.

Eg. Compute the value of $2^{10} \bmod 11$

$2^{10} \equiv 1 \pmod{11}$

Eg. Compute the value of $2^{340} \bmod 11$

$(2^{340})=(2^{10})^{34} \bmod 11$

$=1^{34} \bmod 11 \Rightarrow 1$

//Proof.

Take division algorithm

a = p.q + r where can be 0<= r <= 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<= r<= p-1 on dividing by **p** then ka,1<= k<= p-1 also leaves remainders from 1 to p-1.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 ≡ 1.2.3…(p-1) (mod p)
hence a^{p-1}.(p-1)! ≡ (p-1)! (mod p)
which returns
a^{p-1} ≡ 1 (mod p) [as mod p cannot divide (p-1)!]
hence proved. //

Eg.6^{10} mod 11
Sol. 6^{11-1} mod 11
= 1 [as per theorem]

Eg. 5^{15} 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
2.1 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)=3-1=2
2. phi(8)= 2^3
= $2^3 - 2^{3-1}$ (since $p^a - p^{a-1}$) =$2^3 - 2^2$ =8–4=4

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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:

1. 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
3. 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 number.

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
$a^{P-1} \equiv 1 \pmod{P}$
$a^{12-1} \equiv 1 \pmod{12}$
$a^{11} \equiv 1 \pmod{12}$
Where 1 <= a < 12
We have to calculate $a^{11} \bmod 12$
If it is equal to 1, then it is called prime number. Otherwise, it is called composite number.
Consider, a = 5
$5^{11} \equiv 1 \pmod{12}$
(i.e) $5^{11} \bmod 12 = 5$
It is not equal to 1
Therefore it is not a prime number

3.Miller – Rabin Primality Test

Function Miller-Rabin (x)
$x-1 = (2^{w})y$ //x is the input number for primality test
// y is an odd number and 2 is the base

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

select **a** randomly in the range [2, (x-1)]
$Z = a^y \bmod x$
if Z congruent 1 (mod x) then return prime
for i = 1 to w – 1
{
If Z congruent -1 (mod x) then return prime
$Z = Z^2 \bmod x$
}
return composite

Example:
Find the primality for 7
x = 7
As per algorithm, $x - 1 = 7 - 1 = 6 = 2^1 \times 3$
x = 7, w = 1, y = 3
$Z = a^y \bmod 7$
a = 2 (randomly), where [1<=a<=x-1]
$Z = 2^3 \bmod 7 = 1$
Value of Z = 1, 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?

- $x \equiv a1 \pmod{m1}$
- $x \equiv a2 \pmod{m2}$
- $x \equiv a3 \pmod{m3}$
- $x = \Sigma(a_i M_i y_i) = (a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3) \bmod M$

Let M1, M2, …, Mn be (pairwise) relatively prime numbers. Then the system:
Step1: Calculate M

- M = m1*m2*m3 … mn.

Step 2: Calculate $M_k = M/m_k$
Step 3: Find Inverse of M_K (ie)y_k

Find the X using CRT

- $x \equiv 2 \pmod 3$
- $x \equiv 3 \pmod 5$
- $x \equiv 2 \pmod 7$

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

$a_1 = 2$, $a_2 = 3$, $a_3 = 2$; $m_1 = 3$, $m_2 = 5$, $m_3 = 7$;

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)** $y_1 = 35^{-1} \bmod (3) = 35^{3-2} \bmod (3) = 2$ [since Fermat's inverse theorem or easy inverse method like 35 x ? mod 3=1 (ie) 2]

$y_2 = 1$; $y_3 = 1$

$X = \Sigma(a_i M_i y_i) = (a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3) \bmod M$

$X = [(2 \times 35 \times 2) + (21 \times 3 \times 1) + (2 \times 15 \times 1)] \bmod 105$

$= (140 + 63 + 30) \bmod 105$

$= 233 \bmod 105$

$X = 23$

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.
- 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, **(x^y 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

$11^7 \bmod 13$

$11^2 \bmod 13 = 121 \bmod 13 = 4$

$11^4 \bmod 13 = (11^2 \bmod 13 \times 11^2 \bmod 13) \bmod 13$

$= (4 \times 4) \bmod 13$

$= 16 \bmod 13$

$= 3$

$11^7 \bmod 13 = (11^4 \bmod 13 \times 11^2 \bmod 13 \times 11^1 \bmod 13) \bmod 13$

$= (3 \times 4 \times 11) \bmod 13$

$= (132) \bmod 13$

$= 2$

Find the result of 2^{90} mod 13.

Solution:

Step 1: Split x and y into smaller parts using exponential rules as shown below:

$2^{90} \bmod 13 = 2^{50} \times 2^{40}$

Step 2: Calculate mod n for each part

$2^{50} \bmod 13 = 1125899906842624 \bmod 13 = 4$

ALSO REFER SUBJECT TEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2^{40} mod 13 = 1099511627776 mod 13 = 3
Step 3: Use modular multiplication properties to combine these 2 parts, we have
2^{90} mod 13 = (2^{50} x 2^{40}) mod 13
= (2^{50} mod 13 x 2^{40} mod 13) mod 13
= (4 x 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^x 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_5^*, and the generator is 2, then the discrete logarithm of 1 is 4 because $2^4 \equiv 1 \bmod 5$.
- Input: p - prime number, a- primitive root of p, b - a residue mod p.
- Goal: Find k such that $a^k = b(\bmod p)$. (In other words, find the position of y in the large list of {a, a^2, . . . , a^{q-1}}.
- 14 is a primitive root of 19.
- For example $L_{14}(5) = 10 \bmod 19$, because $14^{10} = 5(\bmod 19)$.
- the inverse problem to exponentiation is to find the **discrete logarithm** of a number modulo p
- that is to find x where $a^x = b \bmod p$
- written as $x=\log_a b \bmod p$ or $x=ind_{a,p}(b)$
- if a is a primitive root then always exists

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 mostdifficult 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

1. 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.
3. **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:

1. 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.
3. 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.

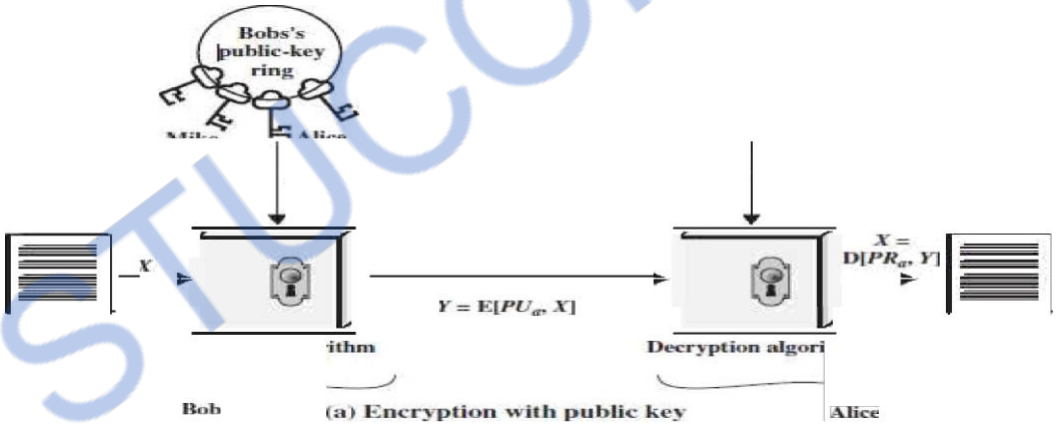


Fig . Public Key Cryptography For Authentication

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› . Y n] •

i.e., Y=EKUb(X)

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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).

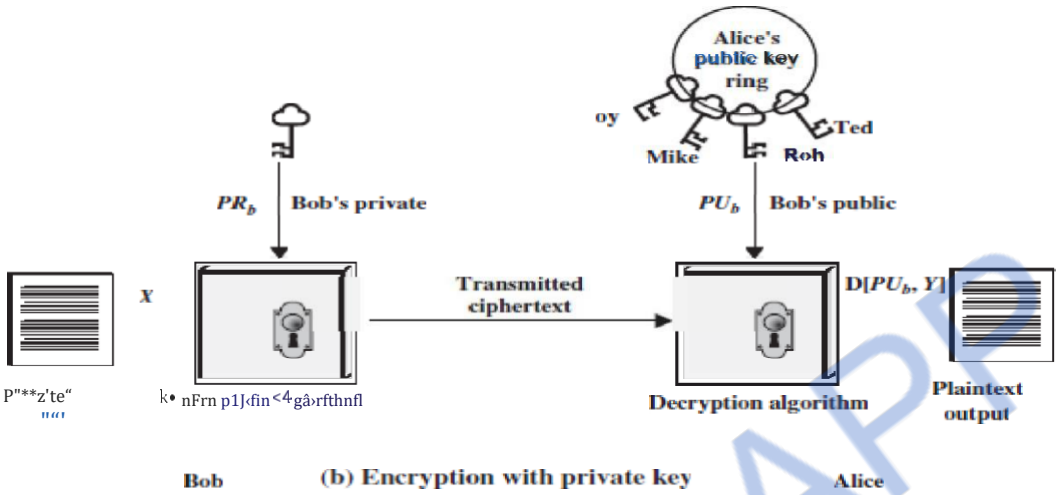


Fig .Private Key Cryptography For Authentication

The encrypted message serves as a digital signature. It is important to emphasize that the encryption process just described does not provide confidentiality.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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. 2. Tr rnusl be irnpnssihlc or at lczst impractical ‹+ decipher a message il the key i‹ kept secret. 3. K riov•'1cJyc 3f the algorithm pJus sample: of cipherrc xt mum be insufficient to dv\crrninc the key.	1 . Ono uJ **urilhm ir** used for «ncmplion anJ 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-cei›cr must cach have one of the matched pcir oi kcTz (n 3t lhc same unc). ' " —_‖'"! -' 1. One of the la'O keys must be kept secret. *. Jl must be impossible or al l«ast impractical r‹1 decipher a message if ‹one of lhc kcYs is kz'j•r sccrcl. 3. Knowledge ‹›f the algori\km plus unc oi rhc kc}z plus sampler of ciphertext melt be irlsufficicnl m determine the othe r key'.

Public Key Cryptography for Security

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]*

$Y = E(PU_B, X)$

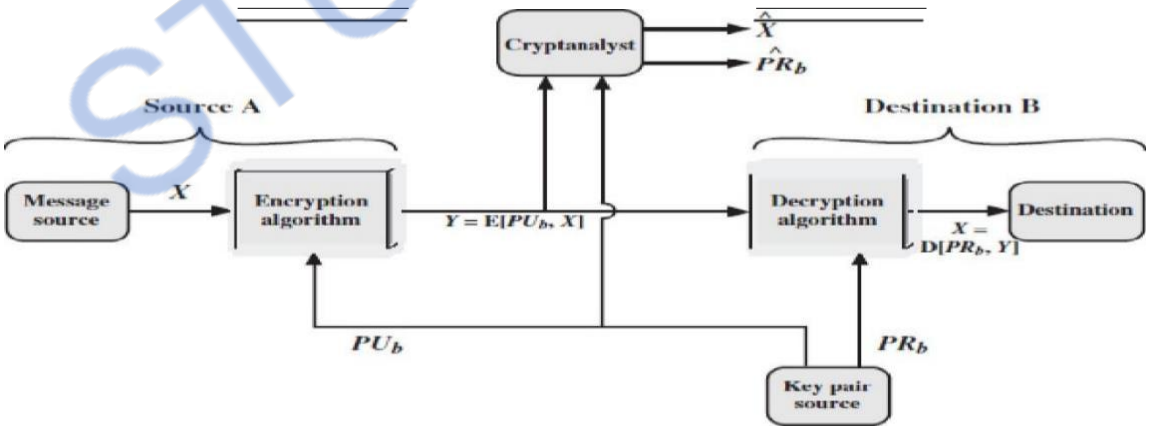


Figure.Publlc-Key Cryptosystem: Secrecy

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

$$X \text{—— D (PRd›}$$

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.

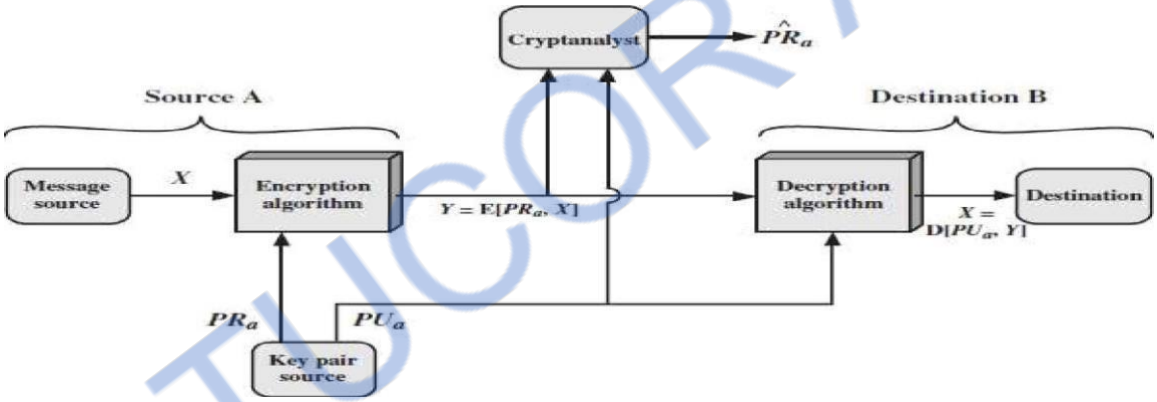


Figure .Public-Key Cryptosystem: Authentication

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 Figure13, 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)]

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Plaintext X = EKUa[EKRb (Y)]

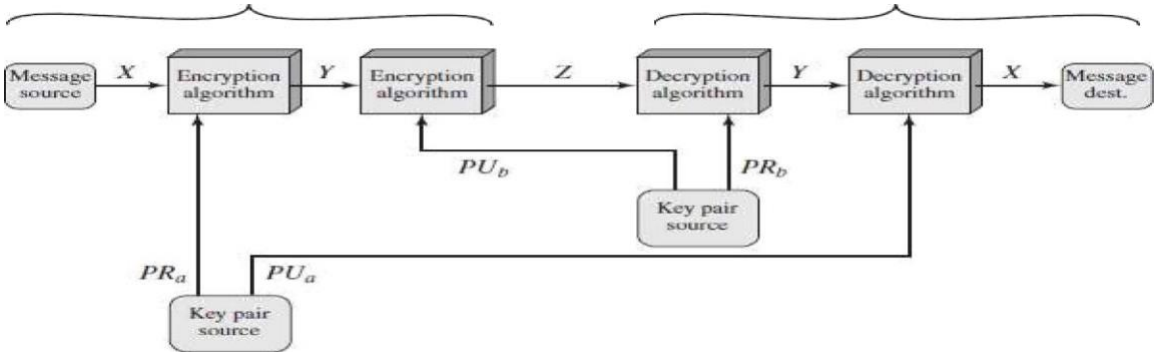


Figure .Public-Key Cryptosystem: Authentication and **Secrecy**

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:

M = DKRb (C) = DKRb [E KUb (M)]

- 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)]

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

RSA ALGORITHM

It was developed by Rivest, Shamir and Adleman. This algorithm makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n.

The RSA scheme is a cipher in which the plaintext and 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 $2^k < n < 2^{k*1}$. Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C:

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

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

Both the sender and receiver know the value of n. the sender knows the value of e and only the receiver knows the value of d. thus, this is a public key encryption algorithm with a public key of KU = {e, n) and a private key of KR = {d, n}. For this algorithm to be satisfactory for public key encryption, the following requirements must be met:

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

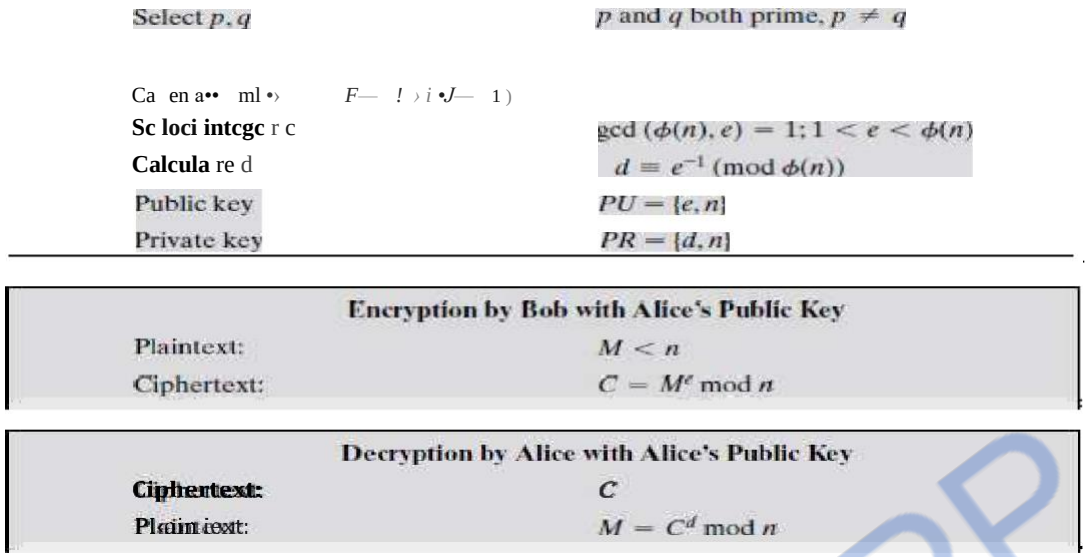


Fig . The RSA Algorithm

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

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

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-1 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 —— 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- 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

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.

$88^7 \bmod 187 = [(88^4 \bmod 187) \cdot (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$
$88^1 \bmod 187 = 88$
$88^2 \bmod 187 = 7744 \bmod 187 = 77$
$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$
$88^7 \bmod 187 = (88 \cdot 77 \cdot 132) \bmod 187 = 894{,}432 \bmod 187 = 11$

For decryption, we calculate *M* —— 1123 mod 187:

$11^{23} \bmod 187 = [(11^1 \bmod 187) \cdot (11^2 \bmod 187) \cdot (11^4 \bmod 187) \cdot (11^{\text{‘}} \bmod 187) \cdot (11 \bmod 187)] \bmod 187$
$11 \bmod 187 = 11$
$^1 \bmod 187 = 121$
$11^2 \bmod 187 = 14{,}641 \bmod 187 = 55$
$11^8 \bmod 187 = 214{,}358{,}881 \bmod 187 = 33$
$11^{23} \bmod 187 = (11 \cdot 121 \cdot 55 \cdot 33 \cdot 33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$

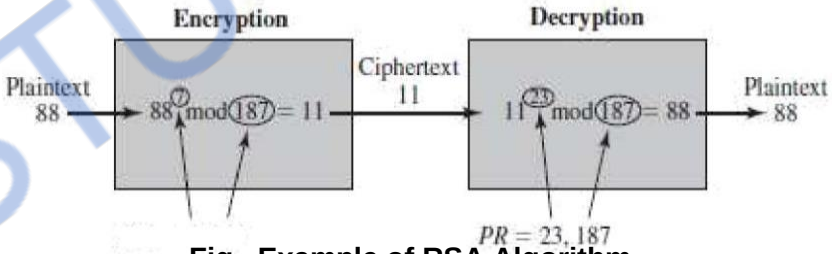


Fig . Example of RSA Algorithm

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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. $d=e^{-1}(\text{mode } (n))$
- 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:

1. 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.
3. Blinding: Multiply the cipher text by a random number before performing exponentiation. This process prevents the attacker from knowing what cipher text bits are being processed inside the computer and therefore prevents the bit-by-bit analysis essential to the timing attack.

KEY MANAGEMENT

There are two uses of public key cryptography regarding the issues of key distribution. They are

1. Distribution of public keys
2. Use of public key encryption to distribute secret keys

Distribution of Public Keys

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.
3. 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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

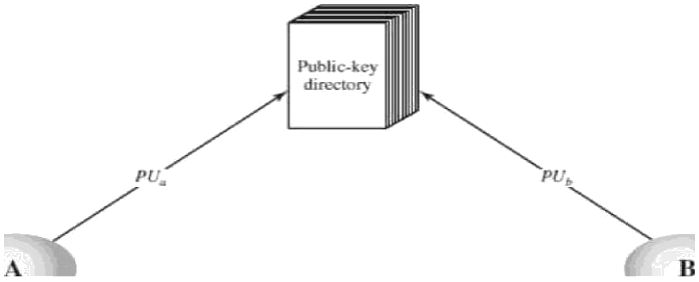


Figure Public-Key Publications

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.

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:

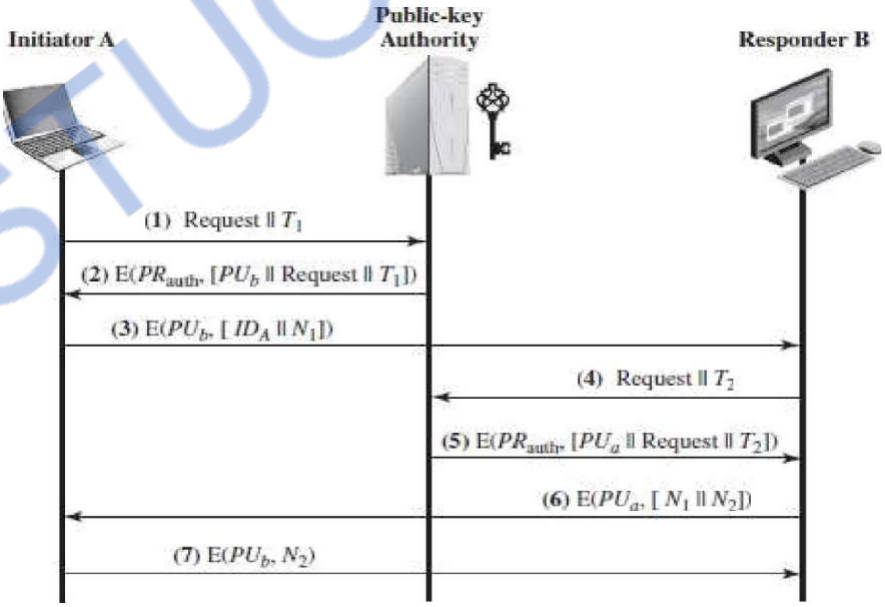


Figure .Public-Key Distribution Scenario

r

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 (/V1), 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

(a) Obtaining certificates from CA

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 generatesa 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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 *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.
5. Computes DxUa[DKRb[M]]to recover thesecret key.

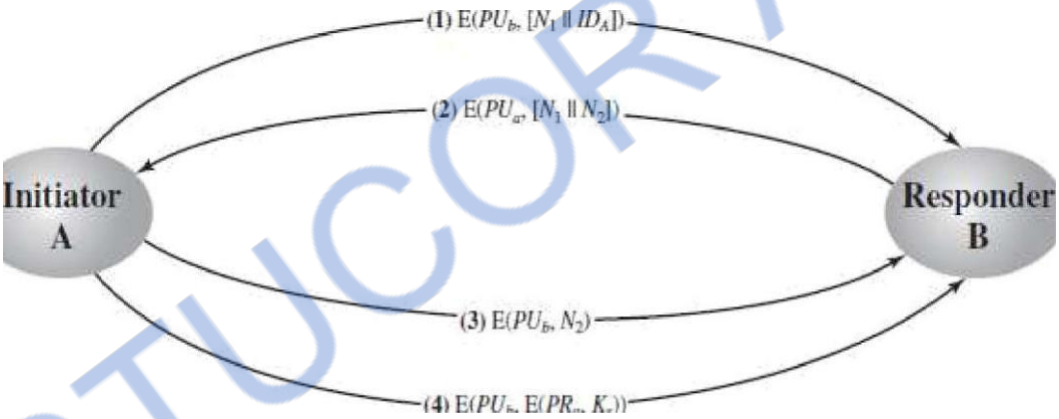


Figure . Public Key Distribution of secret Keys

Advantages:

Scheme ensures both confidentiality and authentication in the exchange of a secret key.

(c) A Hybrid Scheme

Public-key scheme isused to distribute the master keys. The following rationale is provided for using thisthree-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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2. **Backward compatibility:** The hybrid scheme is easily overlaid on an existingKDC scheme with minimal disruption or software changes.

The addition of a public-key layer provides a secure, efficient means of distributingmaster keys.

DIFFIE HELLMAN KEY EXCHANGE

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^2 mod p, ... ap^{-1} 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.

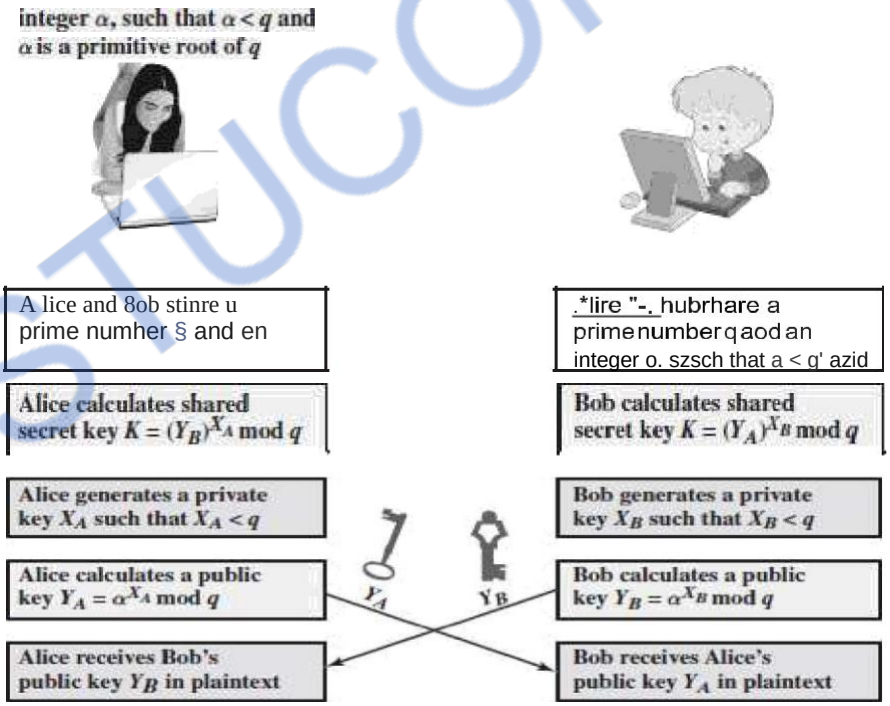


Fig.Diffie Hellman Key Exchange

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

User A computes the key as

$$K = (YB) \mod q \text{ and}$$

User B computes the key as

$$K = (YA)^{XB} \mod q$$

These two calculations produce identical results.

$$K = (YB) \mod q$$
$$= (a^{XB} \mod q) \mod q$$
$$= (a^{XB}) \mod q$$
$$= (a^{XB}) \mod q$$
$$= (a \mod q)^{XB} \mod q$$
$$= (Y_A)^{XB} \mod q$$

Global Public Elements	
q	prime number
α	a<q and a is the primitive root of q
User 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= ^{XB} mod q	
Generation of secret key by User A	
K = (YB) mod q	
Generation of secret key by User B	
K = $(Y_A)^{XB}$ mod q	

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

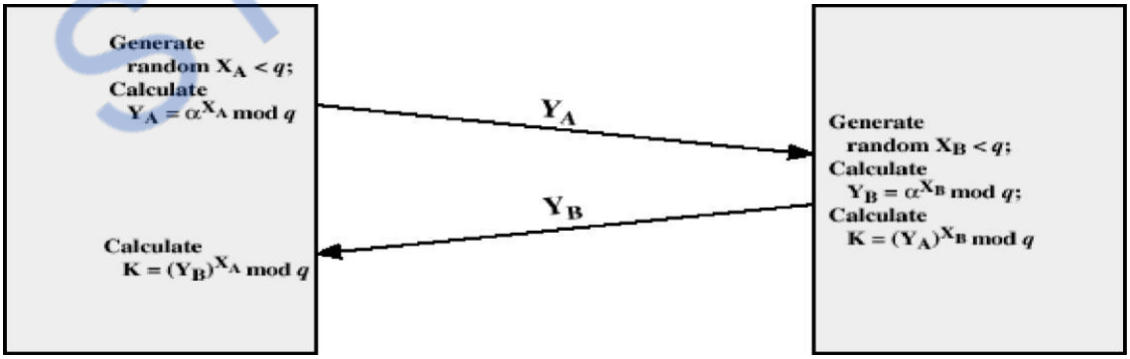


Figure .Diffe-Hellman Key Exchange

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.
3. Darth intercepts YA and transmits YD1 to Bob. Darth also calculates = (YA) ^{X}D mod q.
4. Bob receives YD1 and calculates K1 = (YD1) ^{X}B mod q.
5. Bob transmits XA to Alice.
6. Darth intercepts XA and transmits YD2 to Alice. Darth calculates 1 " (YB) ^{X}D1 mod q.
7. Alice receives Yo2 and calculates 2' (YD2) ^{X} mod q.

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:

1. 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 ^{7} mod 353 = 40.
B computes *YB*—— 3^{233} mod 353 = 248.
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* —— (Y A)^{XB}mod 353 = 40 ^{2}*^{3} mod 353 = 160.

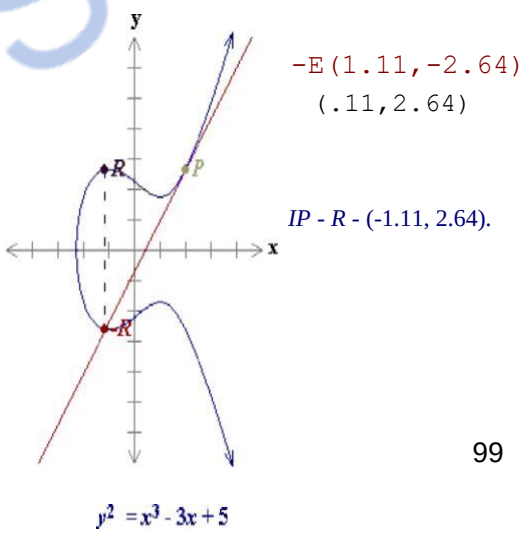
ELLIPTIC CURVE ARITHMETIC

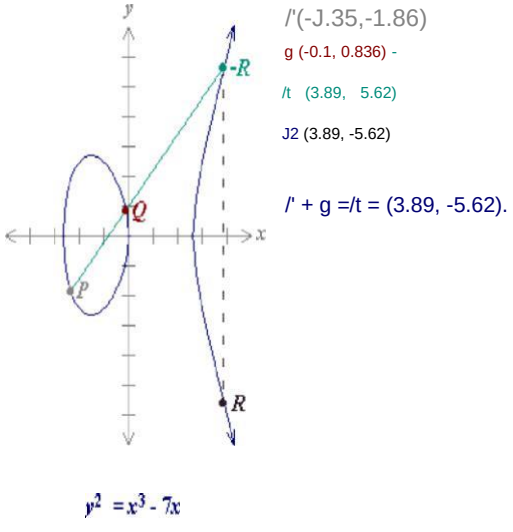
Elliptic Curves:

An elliptic Curve is a Cubic equation of the form

$$Y^2+axy+by = x^3+cx^2+dx+e$$

Where a,b,c,d and e are real numbers





ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

A special addition operation is defined over elliptic curves and with the inclusion of a point —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:

An elliptic group over the Galois Field Ed(a,b) is obtained by computing x^3+ax+b mod p for 0ñ xñ p. The constants a&b are non-negative integers smaller than the prime number p must satisfy the condition.

$$4a^3 + 27b^2 \bmod p \; 1 \; 0$$

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

2. If x2 = x1 and y2 = -y1, that is P = (x1,y1) and Q = (X2,Y2)' (x1.-y1)' -P Then P+Q = O

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

$$\lambda = \begin{cases} \text{y2 —+ } j \quad p, \quad ,;; \\ x_2 - x_1 \\ \dfrac{3x + a}{2J,} \quad if \quad P = Q \end{cases}$$

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 c

Using Bob's public key PB

Pc' [(KG),(PM B)]

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 3.3×10^{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 2×10^{11} MIPS years for a 768 bit RSA key & 3×10^{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 3×10^{20} MIPS years to factor n whereas increasing the elliptic curve key length to only 24 bits will impose a computational complexity of 1.6×10^{2} 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 2ll 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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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
User A Key Generation	
Select private	**n A nA<n**
Calculate public PA	PA nA X G
User B Key Generation	
Select private	nB nB‹n
Calculate public PA	PB = n B X G
Calculation of secret key by User A	
K = nAX PB	
Calculation of secret key by User B	
K = nBX PA	

Figure .ECC Diffie-Hellman Key Exchange

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

4. AUTHENTICATION REQUIREMENT

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.

4.1.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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

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.

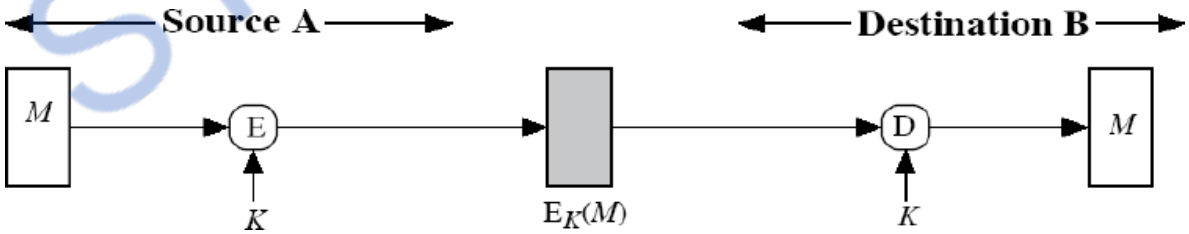
Message Encryption:

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.



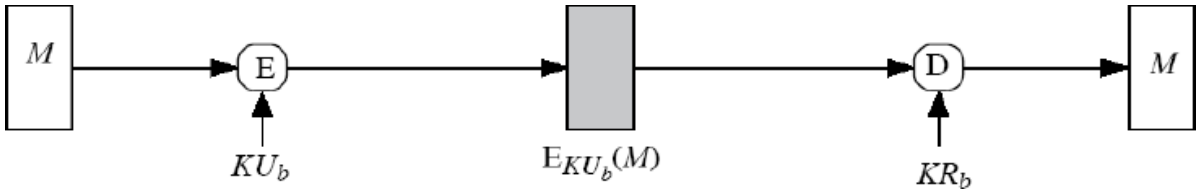
(a).Symmetric key encryption confidentiality, authentication and signature

(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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP



(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.

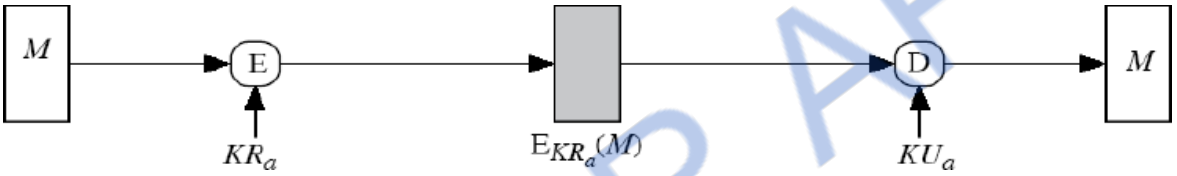


Figure (c) Public key encryption authentication and signature

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.

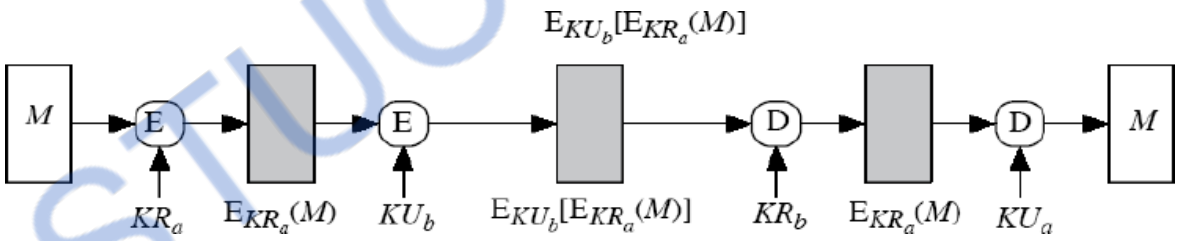


Figure (d) Public key encryption confidentiality, authentication and signature

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.

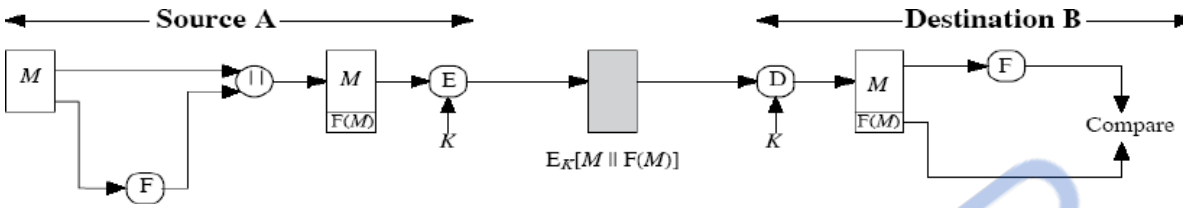
Append an error detecting code, also known as Frame Check Sequence (FCS) or checksum to each message before encryption „A‟ prepares a plaintext message M and then provides this as input to a function F that produces an FCS. The FCS is appended to M and the entire block is then encrypted.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

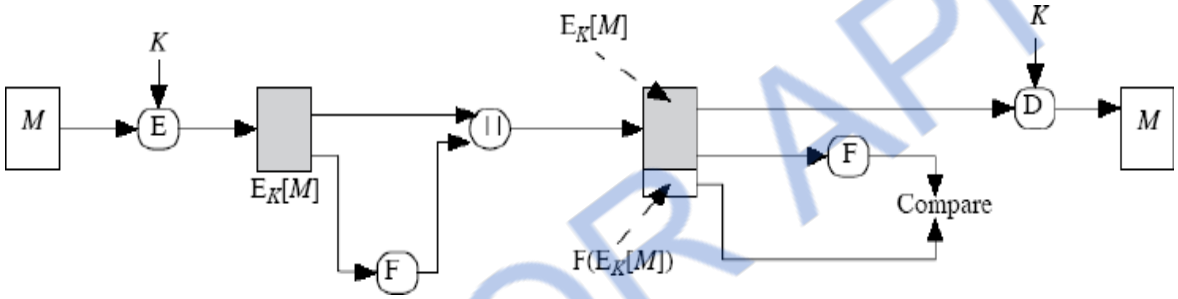
DOWNLOADED FROM STUCOR APP

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).*



(e) Internal error control



(f) External error control

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

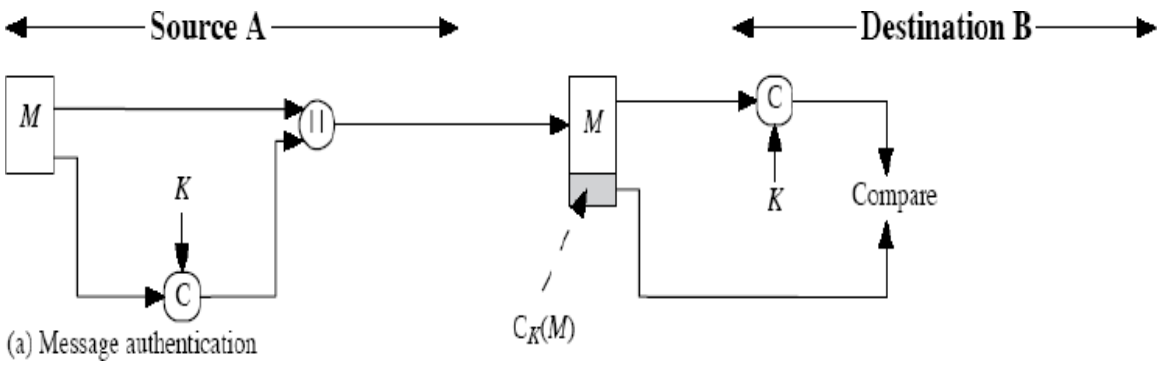
The message plus MAC are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the shared secret key, to generate a new MAC.

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP



(g) Message Authentication

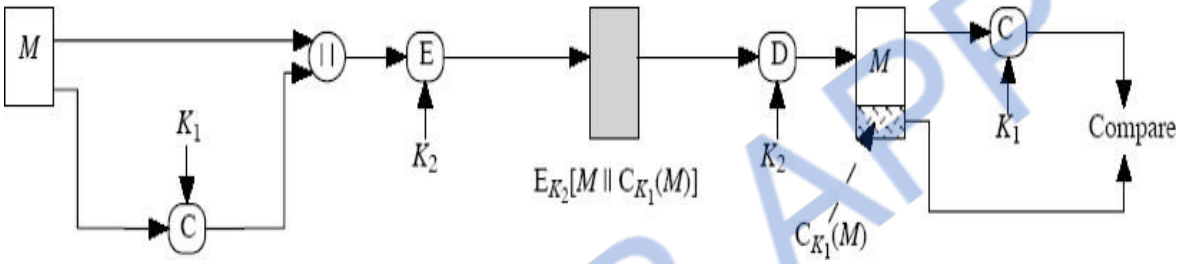


Figure (h) Message authentication and confidentiality, authentication tied to plain text

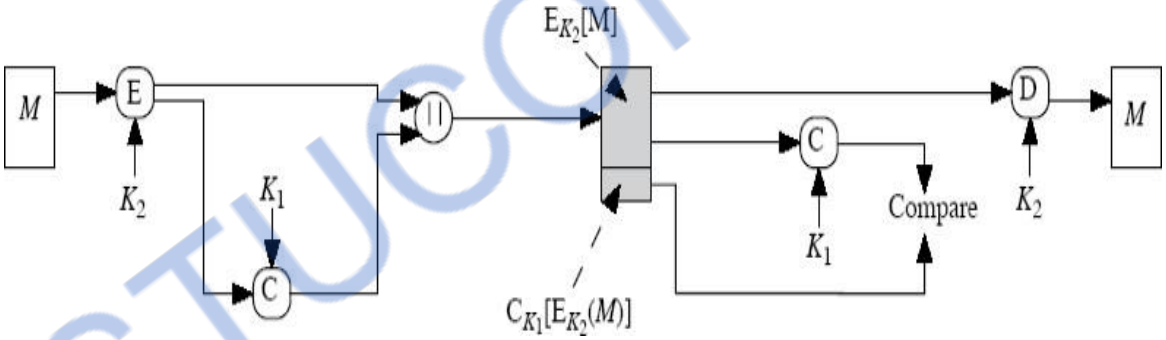


Figure (i) Message authentication and confidentiality, authentication tied to ciphertext

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

MAC size. Then, given a known M_1 and MAC_1, with $MAC_1 = C_K(M_1)$, the cryptanalyst can perform $MAC_i = CK_i(M1)$ for all possible key values Ki.

At least one key is guaranteed to produce a match of $MAC_i = MAC_1$.

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 $\approx 2^{(k-n)}$

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 $\approx 2^{(k-2xn)}$ 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

$$\Delta(M) = X_1 + X_2 \ldots X_m$$
$$C_k(M) = E_k(\Delta(M))$$

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:

$$Y_m = Y_1 + Y_2 \ldots\ldots Y_{m1} + \Delta(M)$$

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‟ such that $C_K(M') = C_K(M)$
- $C_K(M)$ 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).

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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:

$O_1 = E_K(D_1)$
$O_2 = E_K(D_2 + O_1)$
$O_3 = E_K(D_3 + O_2)$
……
$O_N = E_K(D_N + O_{N-1})$

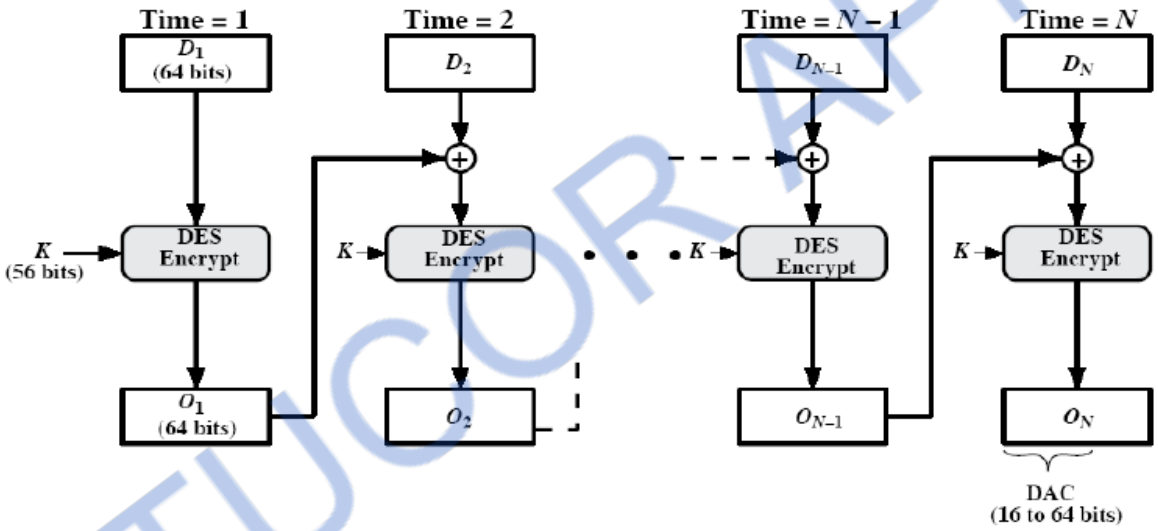


Figure.2 Data Authentication Algorithm

4.3.HASH FUNCTION

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

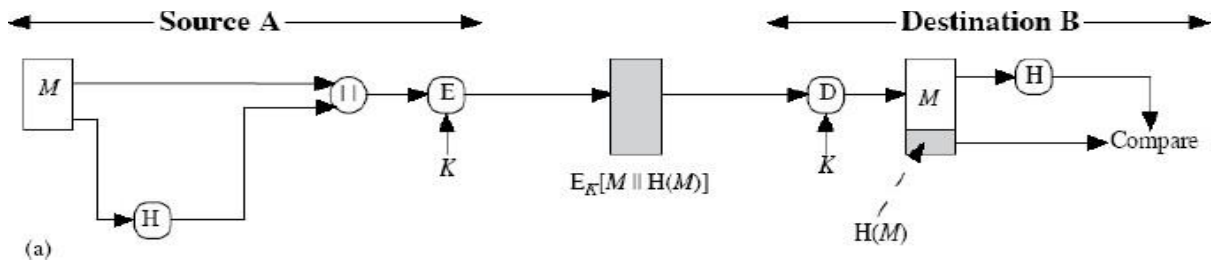
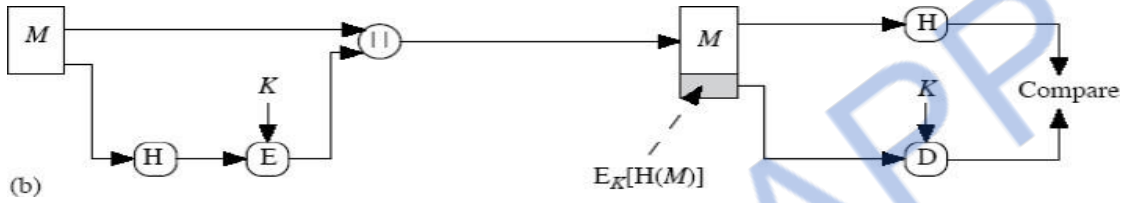


Figure (a) Hash Function

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.

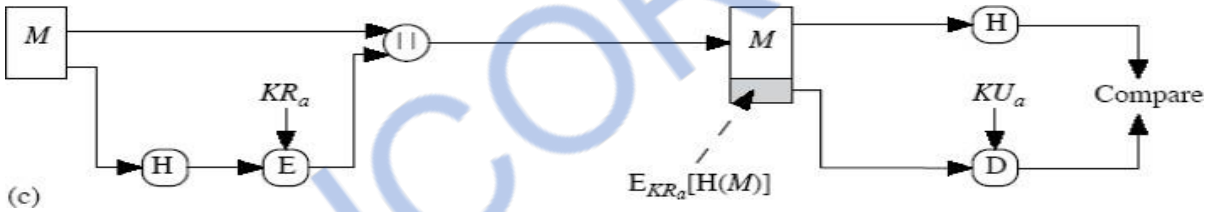
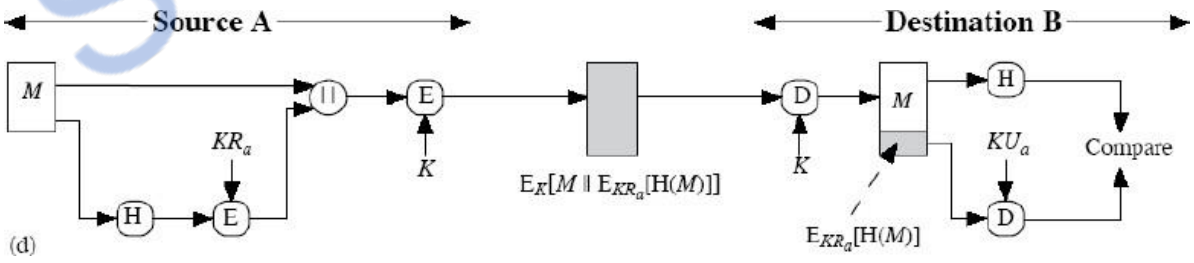


Figure (b & c) Basic use of Hash Function

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.

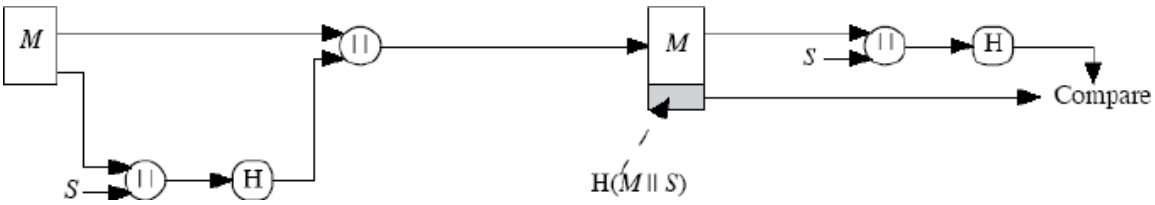


In Fig (e) This technique uses a hash function, but no encryption for message authentication. This technique assumes that the two communicating parties share a common secret value „S". The source computes the hash value over the concatenation of M and S and appends the resulting hash value to M.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP



In Fig(f) Confidentiality can be added to the previous approach by encrypting the entire message plus the hash code.

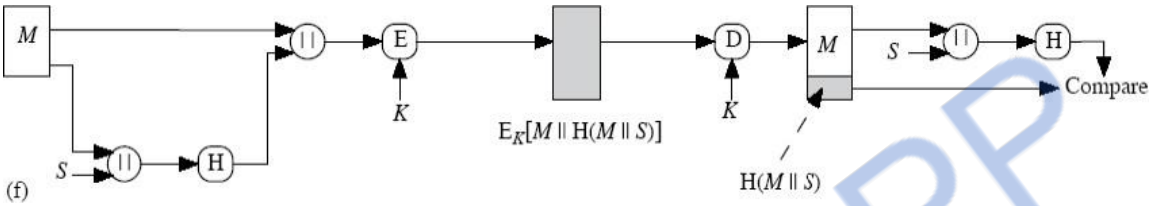


Figure (d,e & f) Basic use of Hash Function

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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: $C_i = b_{i1} + b_{i1} \ldots + b_{im}$

where

C_i = i^{th} bit of the hash code, $1 \leq i \leq n$
m = number of n-bit blocks in the input
b_{ij}= 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:

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 2^{32}

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:

$$H_o = \text{initial value}$$
$$H_i = E_{Mi}[H_{i-1}]$$
$$G = H_N$$

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.

2. Construct any desired message in the form $Q_1, Q_2, ..., Q_{N2}$.

3. Compute for $H_i = EQ_i [H_{i-1}]$ for $1 \leq i \leq (N-2)$.

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.

4.4.SECURITY OF HASH FUNCTION AND MAC

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 x with H(y) = H(x).

Strong collision resistance: It is computationally infeasible to find any pair (x, y) such that H(x) = H(y).

For a hash code of length n, the level of effort required, as we have seen is proportional to the following:

One way	2^n
Weak collision resistance	2^n
Strong collision resistance	$2^{n/2}$

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

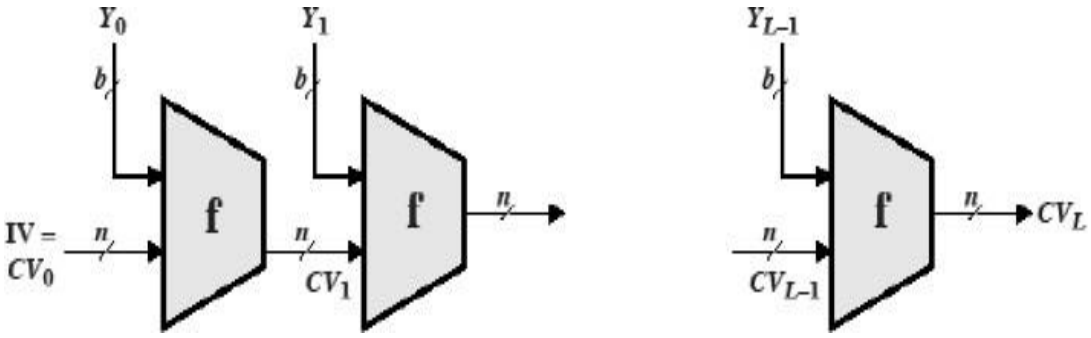
The hash function takes an input message and partitions it into L fixed-sized blocks of b bits each. If necessary, the final block is padded to b bits. The final block also includes the value of the total length of the input to the hash function(Fig 3.4). The inclusion of the length makes the job of 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP



IV=Initial Value
Y_i = 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.

The hash function can be summarized as follows:

CV_0 = IV = initial n-bit value
$CV_I = f(CV_{i-1}, Yi_{-1})$ $1 \leq i \leq L$
$H(M) = CV_L$
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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

4.5.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 **3.1** depicts the overall processing of a message to produce a digest.

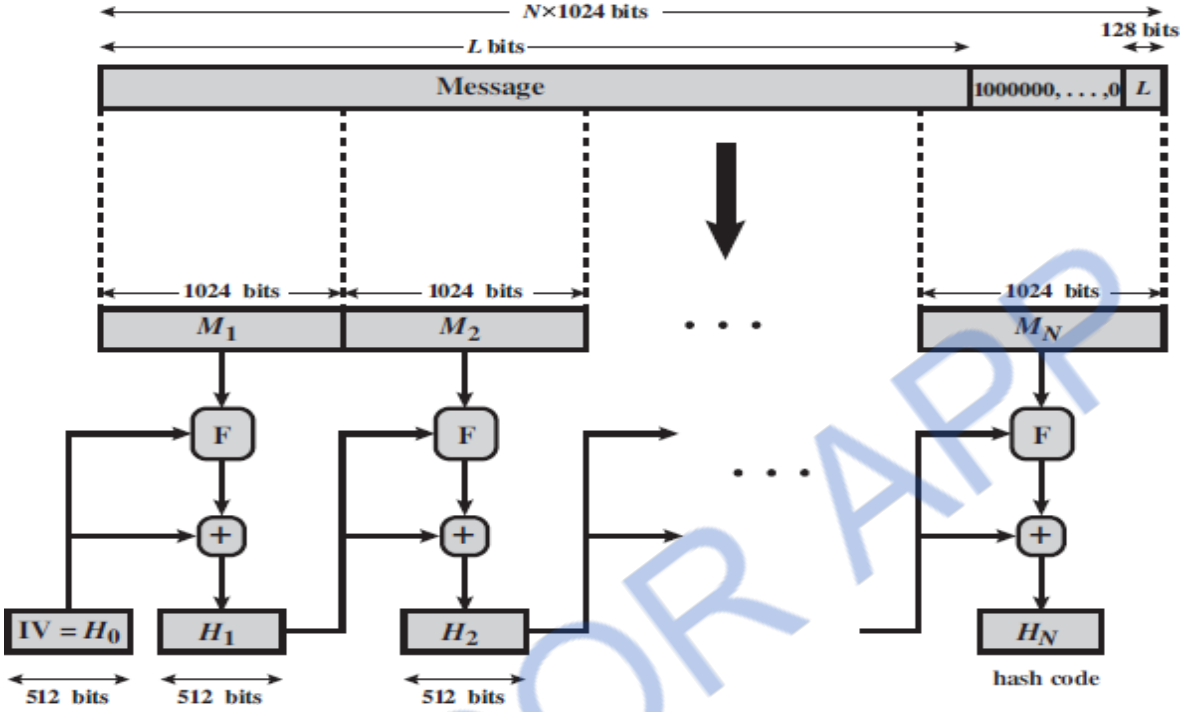


Fig .Message Digest Generation Using SHA-512

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, ... M_N , so that the total length of the expanded message is NX1024 bits.

Step 3: Initialize hash buffer.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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_t, derived from the current 1024-bit block (M_i) being processed .These values are derived using a message schedule described subsequently.

Each round also makes use of an additive constant k_t, 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_i-1) to produce H_i.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

$ABCDE_q$ = the output of the last round of processing of the *i*th 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 inputs

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

MD = final message digest value

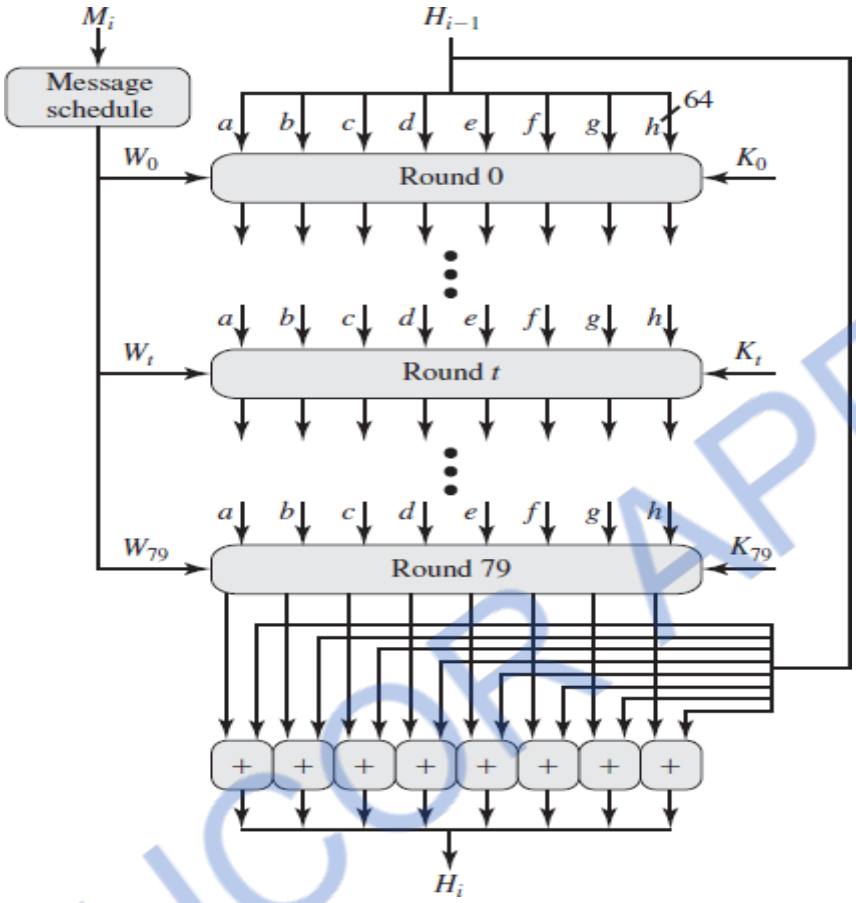


Figure. SHA-512 Processing of a Single 1024-Bit Block

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:

$T_1 = h + Ch(e,f,g) + (\Sigma_1^{512} e) + Wt + Kt)$

$T_2 = (\Sigma_0^{512} a) + Maj(a,b,c)$

$h = g$ $g = f$ $f = e$ $e = d + T_1$ $d = c$
$c = b$ $b = a$ $a = T_1 + T_2$

Where

T = Step number; $0 \le t \le 79$

Ch(e,f,g) = (a AND f) $\oplus$ (NOT e AND g)

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

The conditional function: If e then f else g (Fig 3.10)

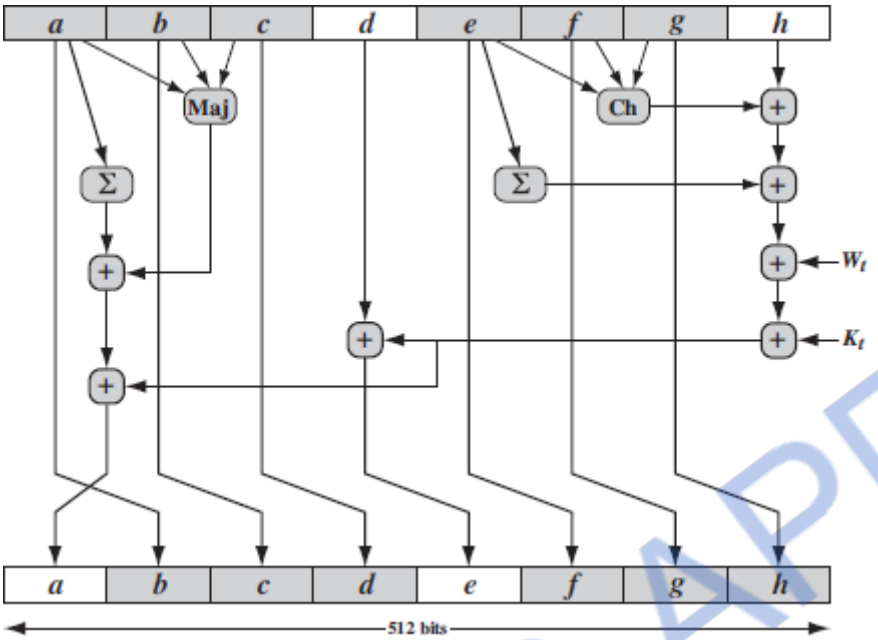


Fig .Elementary SHA Operation(single step)

The functions can be summarized as follows:

Steps	Function Name	Function Value
0≤t≤9	f1 = f(t,B,C,D)	(B Λ C) V (B! Λ D)
20≤t≤39	f2 = f(t,B,C,D)	B ⊕C ⊕D
40≤t≤59	f3 = f(t,B,C,D)	(B Λ C) V (B Λ D) V (C Λ D)
60≤t≤79	f4 = f(t,B,C,D)	B ⊕C ⊕D

The logical operators AND,OR,NOT,XOR, are represented by the symbols Λ V ! ⊕

Only three different functions are used.

For, 0≤ t ≤ 19 the function is the conditional function.

For 20 ≤ t ≤ 39 and 60 ≤ t ≤ 79 the function produces a parity bit.

For A0 ≤ t ≤ 59 the function is true if two or three of the argument are true.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

The following diagram illustrates how the 32bit word values wt are derived from the 512 bit message.

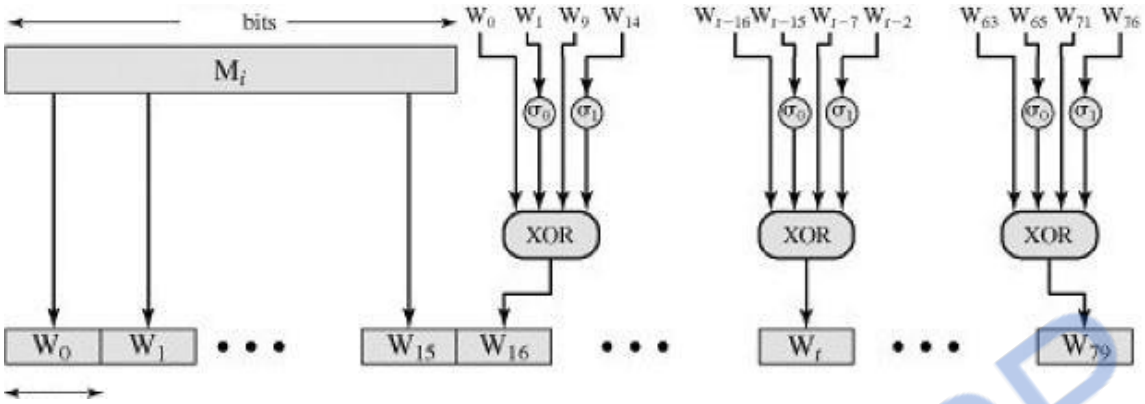


Figure.Creation of 80-word Input Sequence for SHA-512 Processing of Single Block

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

SHA-1 is not vulnerable to such attacks.

3. Speed:

Both algorithms rely on addition module 2^{32}, 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.

HMAC

HMAC Design Objectives:

- To use hash functions that perform well in software and for which code is freely and widely available.
- 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)

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

IV = initial value input to hash function

M = message input to HMAC

Y_i =*i* th block of *M*, 0 ≤ *i* ≤ (*L* – 1)

L =number of blocks in *M*

b = number of bits in a block

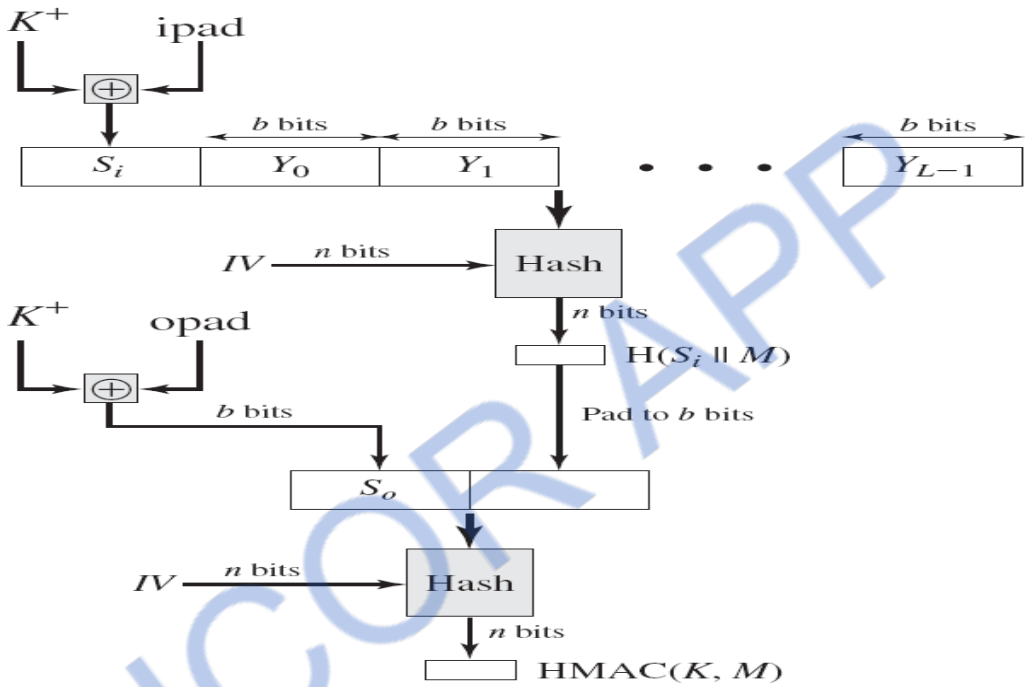


Figure .HMAC Structure

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

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

We can describe the algorithm as follows:

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 S_i.

4. Apply H to the stream generated in step 3.

5. XOR K^+ with opad to produce the b-bit block S_0.

6. Append the hash result from step 4 to S_0.

7. Apply H to the stream generated in step 6 and output the result.

A more efficient implementation is possible. Two quantities are precomputed:

$$f(IV, (K^+ \oplus \text{ipad}))$$

$$f(IV, (K^+ \oplus \text{opad}))$$

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.

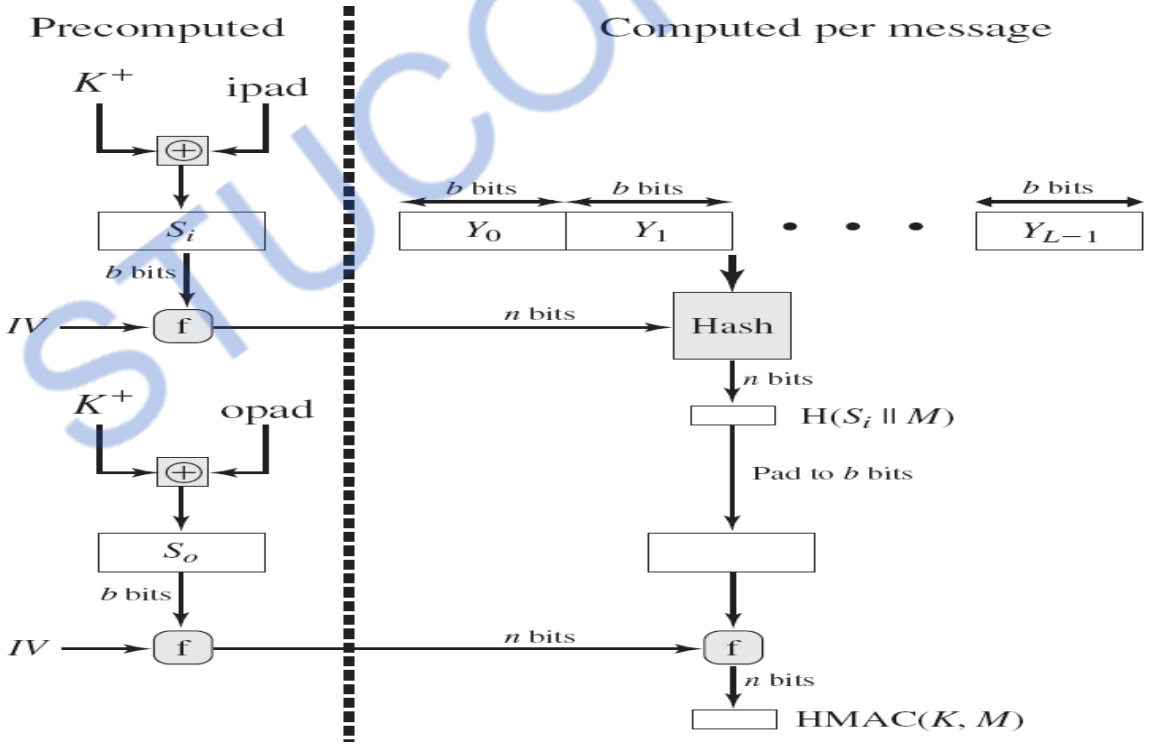


Figure .Efficient Implementation of HMAC

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 theIV 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*‟) .

CMAC

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 $X \| (X \oplus T)$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 *b*, where *b* 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 (*M*1, *M*2,c, *Mn*). The algorithm makes use of a *k*-bit encryption key *K* and a *b*-bit constant, *K*1. For AES, the key size *k* is 128, 192, or 256 bits; for triple DES, the key size is 112 or 168 bits. CMAC is calculated as follows

$$
\begin{aligned}
C_1 &= \mathrm{E}(K, M_1) \\
C_2 &= \mathrm{E}(K, [M_2 \oplus C_1]) \\
C_3 &= \mathrm{E}(K, [M_3 \oplus C_2]) \\
&\cdot \\
&\cdot \\
&\cdot \\
C_n &= \mathrm{E}(K, [M_n \oplus C_{n-1} \oplus K_1]) \\
T &= \mathrm{MSB}_{Tlen}(C_n)
\end{aligned}
$$

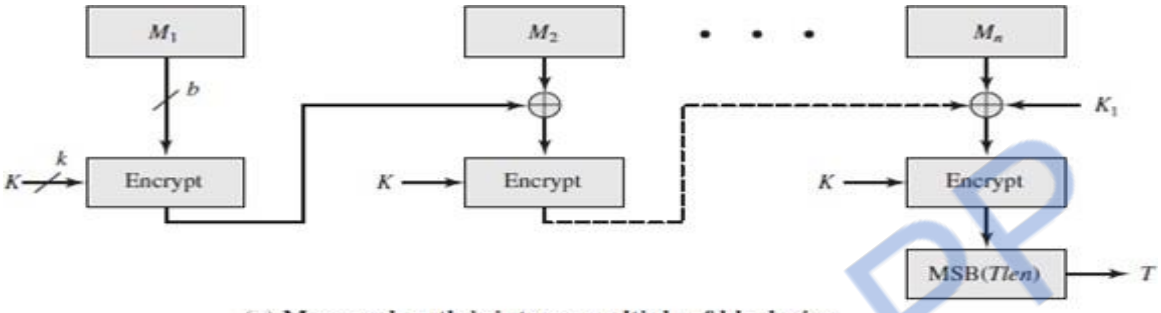
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

where

T = message authentication code, also referred to as the tag
Tlen = bit length of T
MSB*s*(*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 *K*2 is used instead of *K*1.



(a) Message length is integer multiple of block size

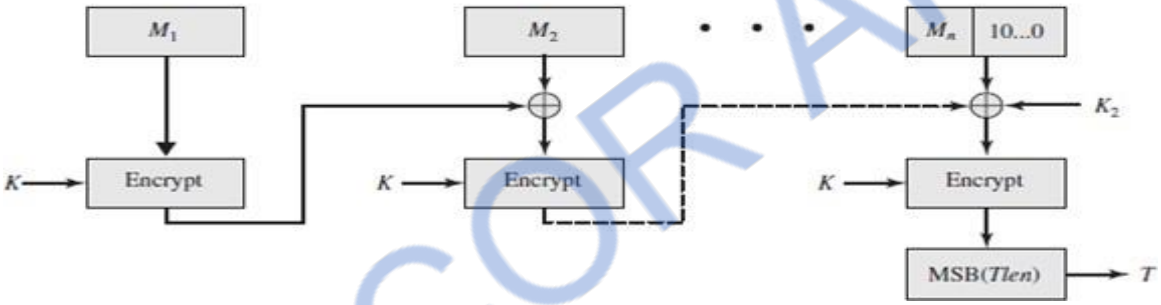


Fig .Cipher-based Message Authentication Code

The two *b*-bit keys are derived from the *k*-bit encryption key as follows.

$$L = \mathrm{E}(K, 0^b)$$
$$K_1 = L \cdot x$$
$$K_2 = L \cdot x^2 = (L \cdot x) \cdot x$$

where multiplication (#) is done in the finite field GF(2*b*) and *x* and *x*2 are first and second-order polynomials that are elements of GF(2*b*). Thus, the binary representation of *x* consists of *b* - 2 zeros followed by 10; the binary representation of *x*2 consists of *b* - 3 zeros followed by 100.

4.6. 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- 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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

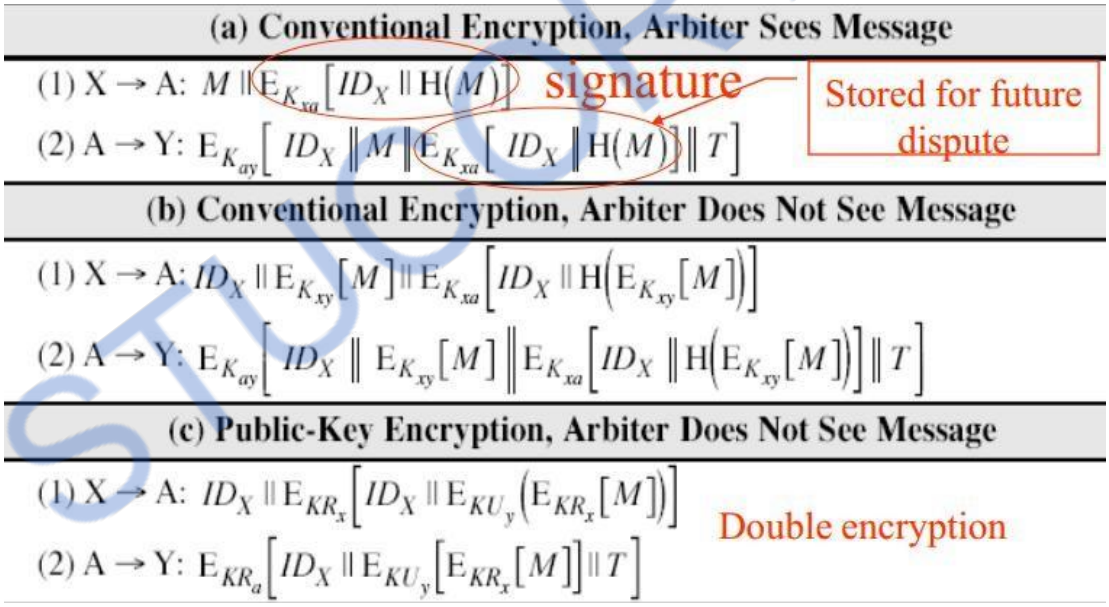
Arbitrated Digital Signatures

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}

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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_{ax} and K_{ay} are the master keys.
- K_{xy} 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_{xy} and a signature to A.
- The signature is the hash of the message encrypted with K_{xa}
- 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_{xa} 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_x 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_x 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Needham-Schroeder Protocol

- Original third-party key distribution protocol
- For session between A and B mediated by KDC
- Protocol overview is:

1. A→KDC: $ID_A || ID_B || N_1$
2. KDC→A: $EK_a[K_s || ID_B || N_1 || EK_b[K_s || ID_A]]$
3. A→B: $EK_b[K_s || ID_A]$
4. B→A: $EK_s[N_2]$
5. A→B: $EK_s[f(N_2)]$

Step 1: A to KDC ,transmit the id of source and destination and a nonce N_1 as a request.
Step 2: A securely acquires the session key in step 2 and a packet to B encrypted with 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 N_2 and transmit to A with K_s.
Step 5: As a hand shake, A encrypt the function of N_2 with K_s
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→KDC: $ID_A || ID_B$

KDC → A: $E(K_a, [K_s II ID_B \; II \; T \; II \; E(K_b, [K_s \; III D_A \; II \; T])])$

A→B: $E(K_b, [K_s \; II \; ID_A \; IIT])$

B→A: $E(K_s, N1)$

A→ B: $E(K_s, f(N1))$

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

$$I \text{ Clock} - T \; I < \Delta t1 + \Delta t2$$

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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"s 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"s 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: →B: $ID_A||N_a$
2. B: → KDC: $ID_B||N_b||E(K_b, [ID_A||N_a||T_b])$
3. KDC: → A: $E(K_a, [ID_B ƒ ƒ N_a ƒ ƒ Ks ƒ ƒ T_b]) „ E(Kb , [ID_A „ Ks „ T_b]) „ N_b$
4. A: →B: $Kb, [ID_A „ K_s „ T_b]) „ E(K_s, N_b)$

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 (N_a), and it provides A with a session key (K_S) and the time limit on its use (T_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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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_A|| ID_B
2. AS→A: $E_{KRas}[ID_A||KU_a||T]$ || $EKRas[ID_B||KU_b||T]$
3. A→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_A|| ID_B
2. *KDC*→A: $E_{KRauth}[ID_B$ II $K_{Ub}]$
3. a→b: $EK_{Ub}[N_a$ II $ID_A]$
4. B→KDC: ID_B II ID_A II EK_{Uauth} $[N_a]$
5. KDC→B: E_{KRauth} [IDA II KUa] II $E_{KUb}[E_{KRauth}[N_a$ II K_s II $ID_B]]$
6. B→A: $E_{KUa}[E_{KRauth}[N_a$ II K_s II $ID_B]$ II $N_b]$
7. A→B: $E_{Ks}[N_b]$

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 N_a.

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$[N_a,K_s,ID_B]$.

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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_A|| ID_B|| N_1
2. KDC→A: $EK_a[K_s$ || ID_B|| N_1|| $EK_b[K_s||ID_A]$]
3. A→B: $EK_b[K_s||ID_A]$ || $EK_s[M]$

- 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:

$$A \rightarrow B: M \text{„II } E_{KRa}(H(M))$$

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"s public key:

$$A \rightarrow B: E_{KUb}, [M \,||\, E_{KRa}, H(M)]$$

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

$$A \rightarrow B: M \,||\, EKR_a [H(M)] \,||\, EKR_{as} \,II\, T|| \,ID_A \,||\, KU_a]$$

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‟s public key.

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)*.

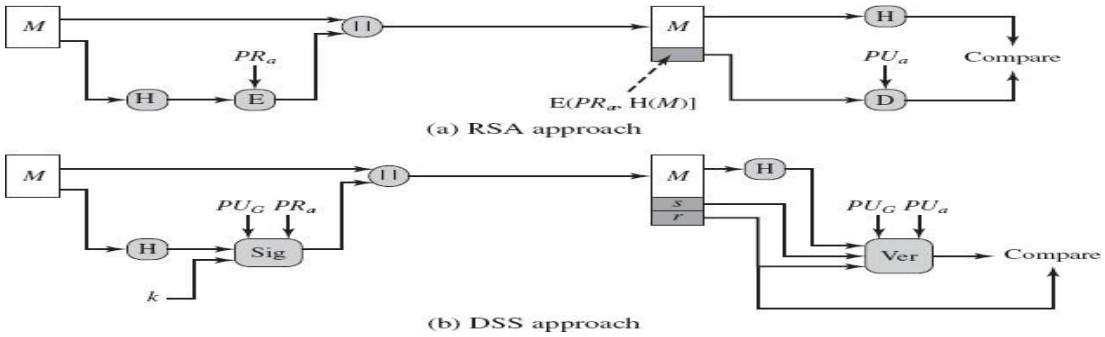


Figure 3.15 Two Approaches to Digital Signatures

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function. The verification function also depends on the global public key as well as the sender"s public key, which is paired with the sender"s private key.

The output of the verification function is a value that is equal to the signature component if the signature is valid. The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature.

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 $h^{(p-1)/q} \bmod p$, 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 $y = g^x \bmod p$ .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"s public key, and the hash code of the incoming message. If this quantity matches the component of the signature, then the signature is validated.

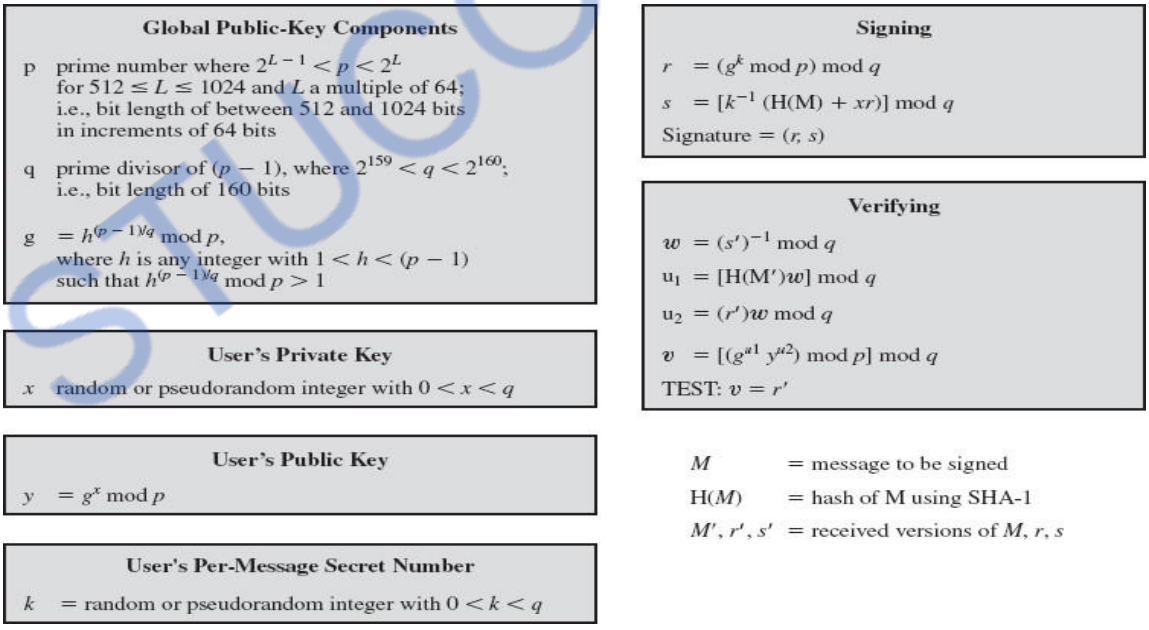


Figure.The Digital Signature Algorithm (DSA)

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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 $g^k \bmod p$. 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 (*q*1) and should be chosen randomly. The public key is calculated from the private key as $y = g^x \bmod p$.

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).

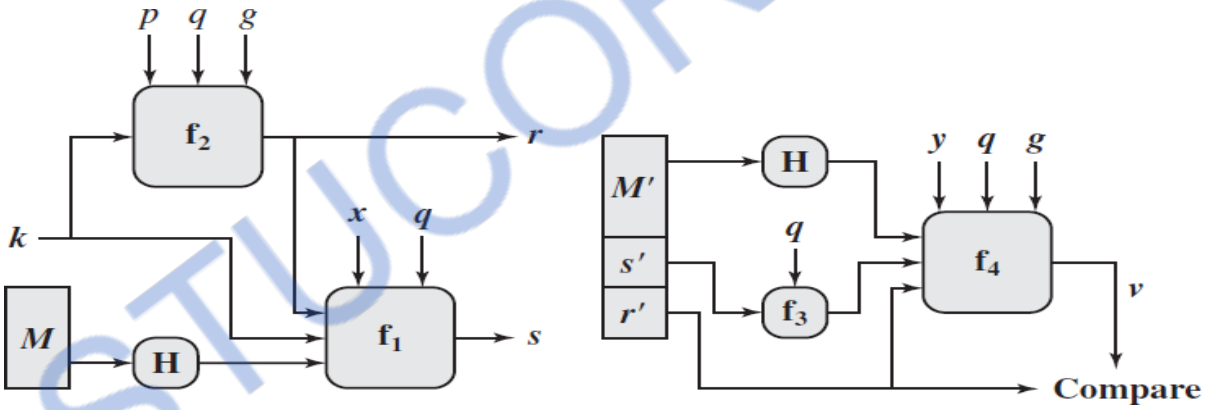


Figure DSS Signing and Verifying

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

4.8.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.

4. **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.

1. **Lower level:** Some function that produces an authenticator: a value to be used to authenticate a message.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

2. **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:

1. **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.

4.9.KERBEROS

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 third-party authentication service that uses a protocol based on Needham and Schroeder.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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

1.A Simple Authentication Dialogue

In an unprotected network environment, any client can apply to any server for service. The obvious security risk is that of impersonation. 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: $ID_c||P_c||ID_v$

2. AS >> C: Ticket

3. C >> V: $ID_c||$Ticket

Ticket= $EK_v(ID_c||ADc||ID_v)$

C : Client,
AS : Authentication Server,
V : Server, ID_c : ID of the client,
P_c : Password of the client,
AD_c : Address of client, ID_v : ID of the server,
K_v : secret key shared by AS and V,
|| : concatenation.

2.A 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 >> AS: $ID_c||ID_{tgs}$
2. AS >> C: Ek_c ($Ticket_{tgs}$)

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Once per type of service:

3. C >> TGS: $ID_c||ID_v||Ticket_{tgs}$
4. TGS >> C: $ticket_v$

Once per service session:

5. C >> V: $ID_c||\ Ticket_v$

$Ticket_{tgs} = Ekt_{gs}(ID_c||AD_c||IDt_{gs}||TS_1||Lifetime_1)$

$Ticket_v = Ek_v(ID_c||AD_c||ID_v||TS2||Lifetime_2)$

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 ($Ticket_{tgs}$) 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:

1. 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:

3. 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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_v) 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:

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)
    - 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) $C \rightarrow AS : ID_C \text{ II } ID_{tgs} \text{ II } TS_1$ (2) $AS \rightarrow C : EKc [K_{c,tgs} \text{ II } ID_{tgs} \text{ II } TS_2 \text{ II } Lifetime_2 \text{ II } Ticket_{tgs}]$
(b) Ticket-Granting Service Exchange: to obtain service-granting ticket

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

(3) C → TGS: ID_v II $Ticket_{tgs}$ II $Authenticator_c$
(4) TGS → C: $EK_{c,tgs}[K_{c,y}$ II ID_v II TS_4 II $Ticket_v]$
$Ticket_{tgs} = E_{K,tgs}[K_{c,tgs}$ II ID_C II AD_C II ID_{tgs} IITS_2 II $Lifetime_2]$
$Ticket_v = E_{Kv}[K_{c,v}$ II ID_C II AD_C II ID_v IITS_4 II $Lifetime_4]$
$Authenticator_C = E_{Ktgs}$ [ID_C II AD_C II $TS_3]$

(c) Client/Server Authentication Exchange: to obtain service

(5) C → V : $Ticket_v$ II $Authenticator_c$
(6) V → C: $E_{kc,v}[TS_5 + 1]$
$Ticket_v = EK_v[K_{c,v}$ II ID_C II AD_C II Id_v II TS_4 II $Lifetime_4]$

$Authenticator_c = EK_{tgs}$ $[ID_C$ II AD_C II $TS_3]$

Kerberos 4 Overview

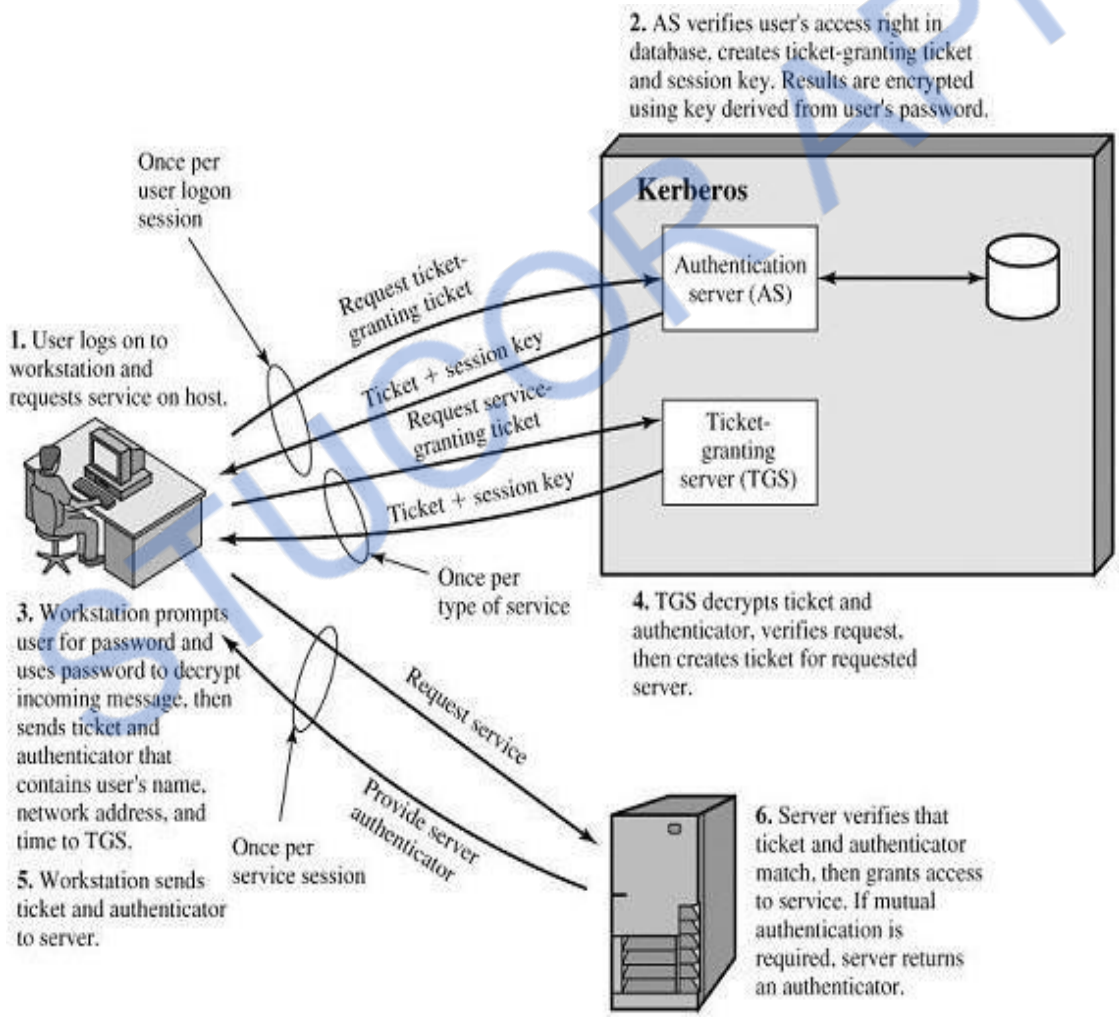


Fig 4.1 Overview of Kerberos 4

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.
5. 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.

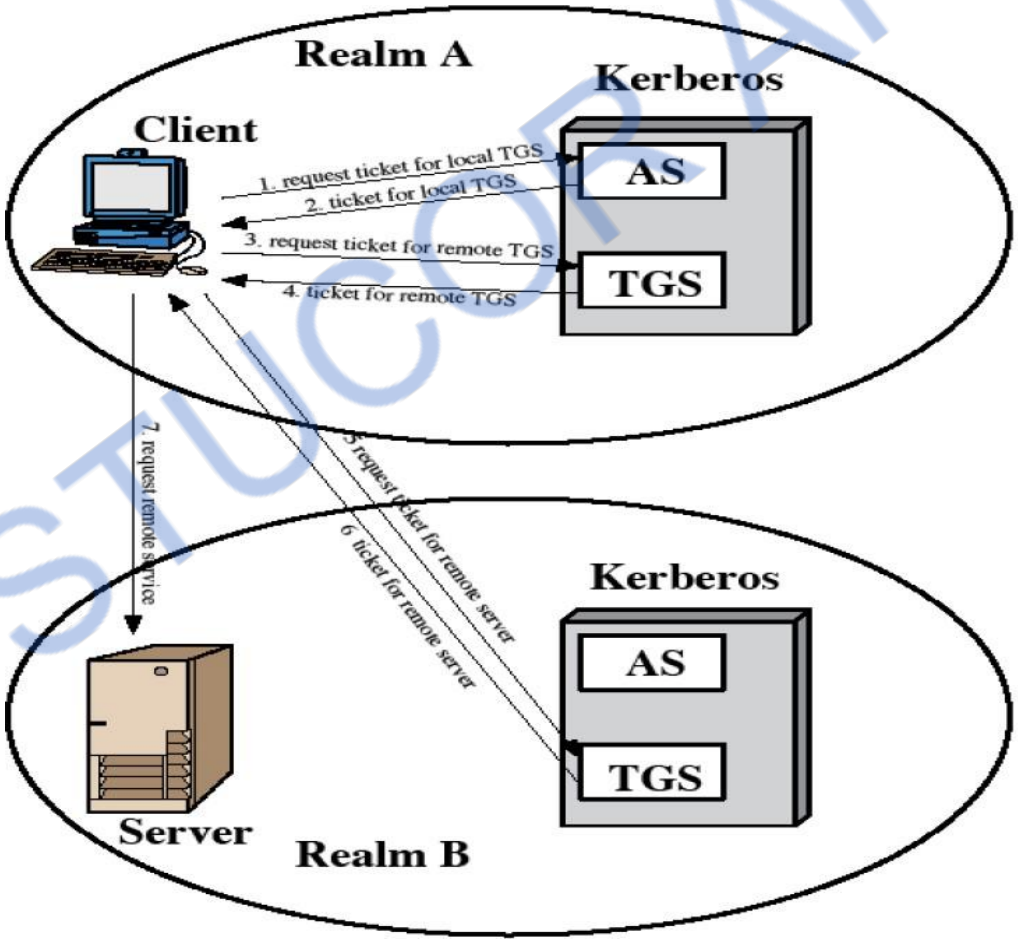


Fig .Request for service in another Realm

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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:

$C \rightarrow AS$: $ID_C \text{ II } ID_{tgs} \text{ II } TS_1$

$AS \rightarrow C$: $EK_c[K_{c,tgs} \text{ ii } ID_{tgs} \text{ II } TS_2 \text{ II Lifetime}_2 \text{ II Ticket}_{tgs}$

$C \rightarrow TGS$: $ID_{tgsrem} \text{ II Ticket}_{tgs} \text{ II Authenticator}_c$

$TGS \rightarrow C$: $E\ K_{c,tgs}[K_{c,tgsrem} \text{ II } ID_{tgsrem} \text{ II } TS_4 \text{ II Ticket}_{tgsrem}$

$C \rightarrow TGS_{rem}$: $ID_{vrem} \text{ II Ticket}_{tgsrem} \text{ II Authenticator}_c$

$TGS_{rem} \rightarrow C$: $EK_{c,tgsrem}[\ K_{c,vrem} \text{ II } ID_{vrem} \text{ II } TS_6 \text{ II Ticket}_{vrem}$:

$C \rightarrow V_{rem}$: $\text{Ticket}_{vrem \text{ II}} \text{Authenticator}_c$

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:

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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:

- Double encryption
- PCBC encryption
- Session keys
- Password attacks

The Version 5 Authentication Dialogue

(a) Authentication Service Exchange: to obtain ticket-granting ticket
(1) $C \rightarrow AS$: Options II ID_c II $Realm_c$ II Times II $Nonce_1$
(2) $AS \rightarrow C$: $Realm_c$ II ID_c II $Ticket_{tgs}$ II EK_c [$K_{c,tgs}$ II Times II $Nonce_1$ II $Realm_{tgs}$ II ID_{tgs}]
$Ticket_{tgs} = EK_{tgs}$ [Flags II $K_{c,tgs}$ II $Realm_c$ II ID_c II AD_c II Times]
(b) Ticket – Granting Service Exchange:to obtain service-granting ticket
(3) $C \rightarrow TGS$: Optionns II ID_v II Times II $Nonce_1$
(4) $TGS \rightarrow C$: $Realm_c$ II ID_c II $Ticket_v$ II $EK_{c,tgs}$[$K_{c,v}$ II Times II $Nonce_2$ II $Realm_v$ II ID_v]
$Ticket_{tgs} = EK_{tgs}$[Flags II $K_{c,tgs}$ II $Realm_c$ II ID_c II AD_c II Times]
$Ticket_v = Ek_v$[[Flags II $K_{c,v}$ II $Realm_c$ II ID_c II AD_c II Times]

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

$Authenticator_c = EK_{c,tgs}[ID_c \text{ II } Realm_c \text{ II } TS_1]$
(c) Client/Server AUTHENTICATION Exchange: to obtain service
(5) C → V : Options II $Ticket_v$ II $Authenticator_c$ (6) V → C : $EK_{c,v}$ [TS_2 II subkey II Seq #] $Ticketv = EK_v[Flags \text{ II } K_{c,v} \text{ II } Realm_c \text{ II } ID_c \text{ II } AD_c \text{ II Times}]$ $Authenticator_c = E_{Kc,v}[ID_c \text{ II } Realm_c \text{ II } TS_2 \text{ II Subkey II Seq\#}]$

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:
  - from : the desired start time for the requested ticket
  - till : the requested expiration time for the requested ticket
  - r_{time} : 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 .

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:

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- **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.

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
3. **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_A):** 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

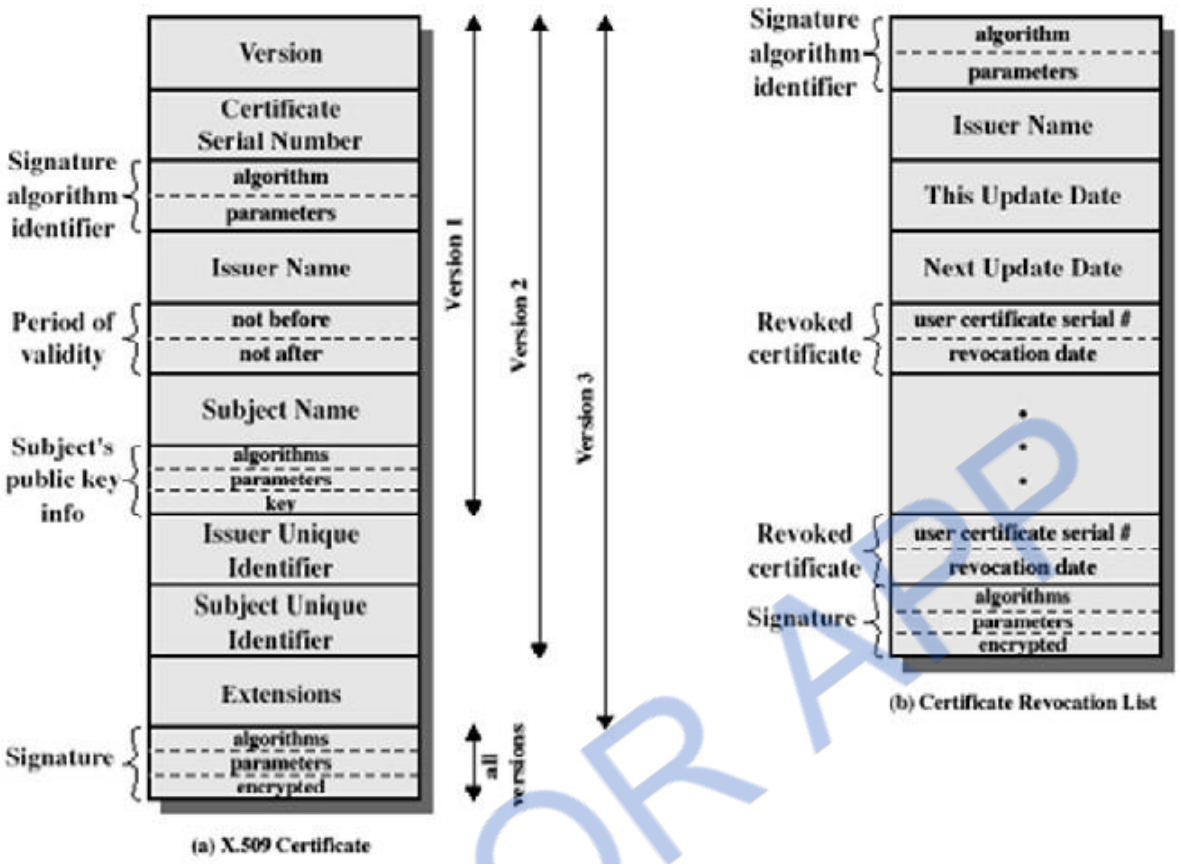


Fig X.509 AUTHENTICATION SERVICES

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}
where
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:

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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 x2) 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

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>>

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

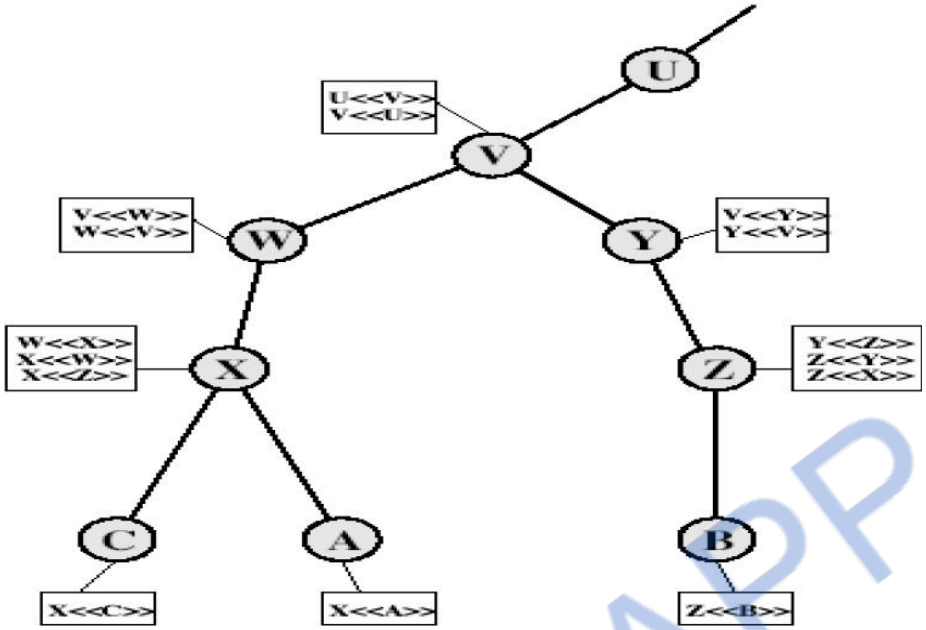


Fig4.4 : Hierarchy of X.509

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

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Message must include timestamp, nonce, B's identity and is signed by A

Two-Way Authentication

Two messages (A→B, B→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→B, B→A, A→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

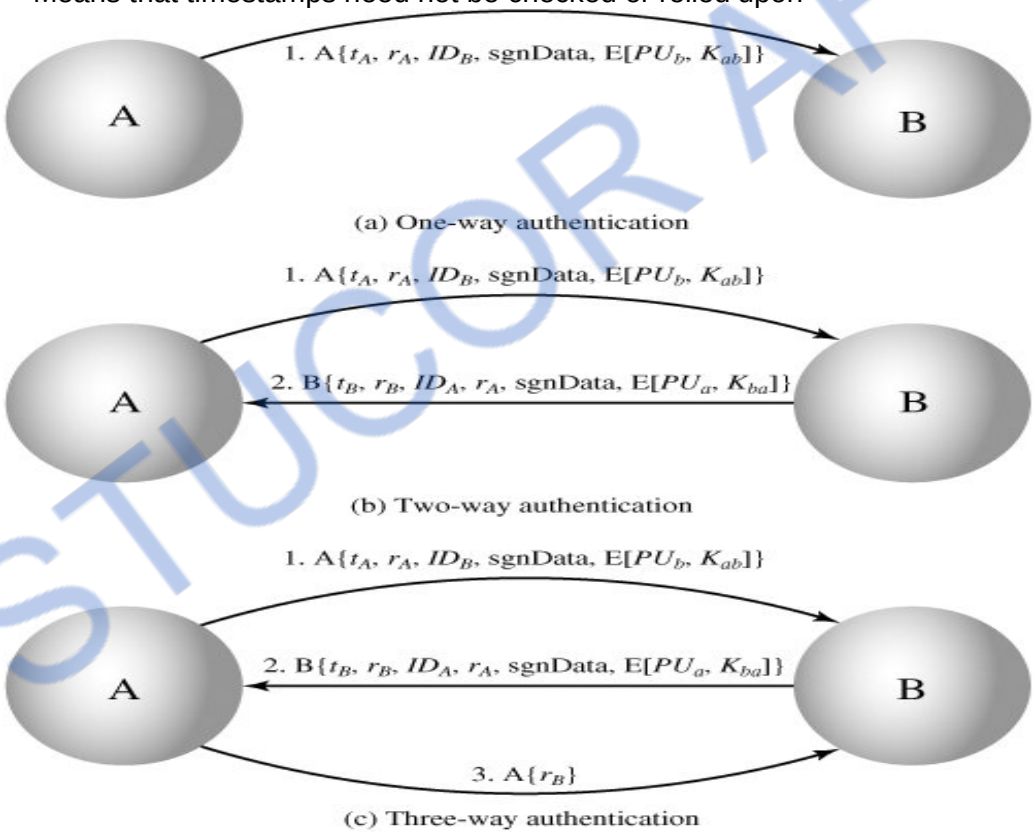


Fig: X509 Strong Authentication Procedure

X.509 Version 3

The following requirements not satisfied by version 2:

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

14. The Subject field is inadequate to convey the identity of a key owner to a public-key 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 Internet-related 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.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

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.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

UNIT V - SECURITY PRACTICE AND SYSTEM SECURITY

Electronic Mail security – PGP, S/MIME – IP security – Web Security - SYSTEM SECURITY: Intruders – Malicious software – viruses – Firewalls.

Electronic Mail security

5.1.1 PRETTY GOOD PRIVACY (PGP)

PGP provides the confidentiality and authentication service that can be used for electronic mail and file storage applications.

The steps involved in PGP are

- Select the best available cryptographic algorithms as building blocks.
- Integrate these algorithms into a general purpose application that is independent of operating system and processor and that is based on a small set of easy-touse commands.
- Make the package and its documentation, including the source code, freely available via the internet, bulletin boards and commercial networks.
- Enter into an agreement with a company to provide a fully compatible, low cost commercial version of PGP.

PGP has grown explosively and is now widely used.

A number of reasons can be cited for this growth.

- It is available free worldwide in versions that run on a variety of platform.
- It is based on algorithms that have survived extensive public review and are considered extremely secure.e.g., RSA, DSS and Diffie Hellman for public key encryption
- It has a wide range of applicability.
- It was not developed by, nor it is controlled by, any governmental or standards organization.

5.1.1.1. Operational description

The actual operation of PGP consists of five services:

1. Authentication
2. Confidentiality
3. Compression
4. E-mail compatibility
5. Segmentation.

1. Authentication: The sequence for authentication is as follows:

- The sender creates the message
- SHA-1 is used to generate a 160-bit hash code of the message
- The hash code is encrypted with RSA using the sender‟s private key and the result is pretended to the message.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
- The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

2. Confidentiality

Confidentiality is provided by encrypting messages to be transmitted or to be stored locally as files. In both cases, the conventional encryption algorithm CAST-128 may be used.

The 64-bit cipher feedback (CFB) mode is used. In PGP, each conventional key is used only once. That is, a new key is generated as a random 128-bit number for each message. Thus although this is referred to as **a session key**, it is in reality a **onetime key**. To protect the key, it is encrypted with the receiver's public key.

The sequence for confidentiality is as follows:

- The sender generates a message and a random 128-bit number to be used as a session key for this message only.
- The message is encrypted using CAST-128 with the session key.
- The session key is encrypted with RSA, using the receiver's public key and is prepended to the message.
- The receiver uses RSA with its private key to decrypt and recover the session key.
- The session key is used to decrypt the message.

Confidentiality and authentication

Here both services may be used for the same message. First, a signature is generated for the plaintext message and prepended to the message. Then the plaintext plus the signature is encrypted using CAST-128 and the session key is encrypted using RSA.

3. Compression

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space for both e-mail transmission and for file storage.

The signature is generated before compression for two reasons:

- It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification. If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required.
- Even if one were willing to generate dynamically a recompressed message from verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and as a result, produce different compression forms.

Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult. The compression algorithm used is ZIP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

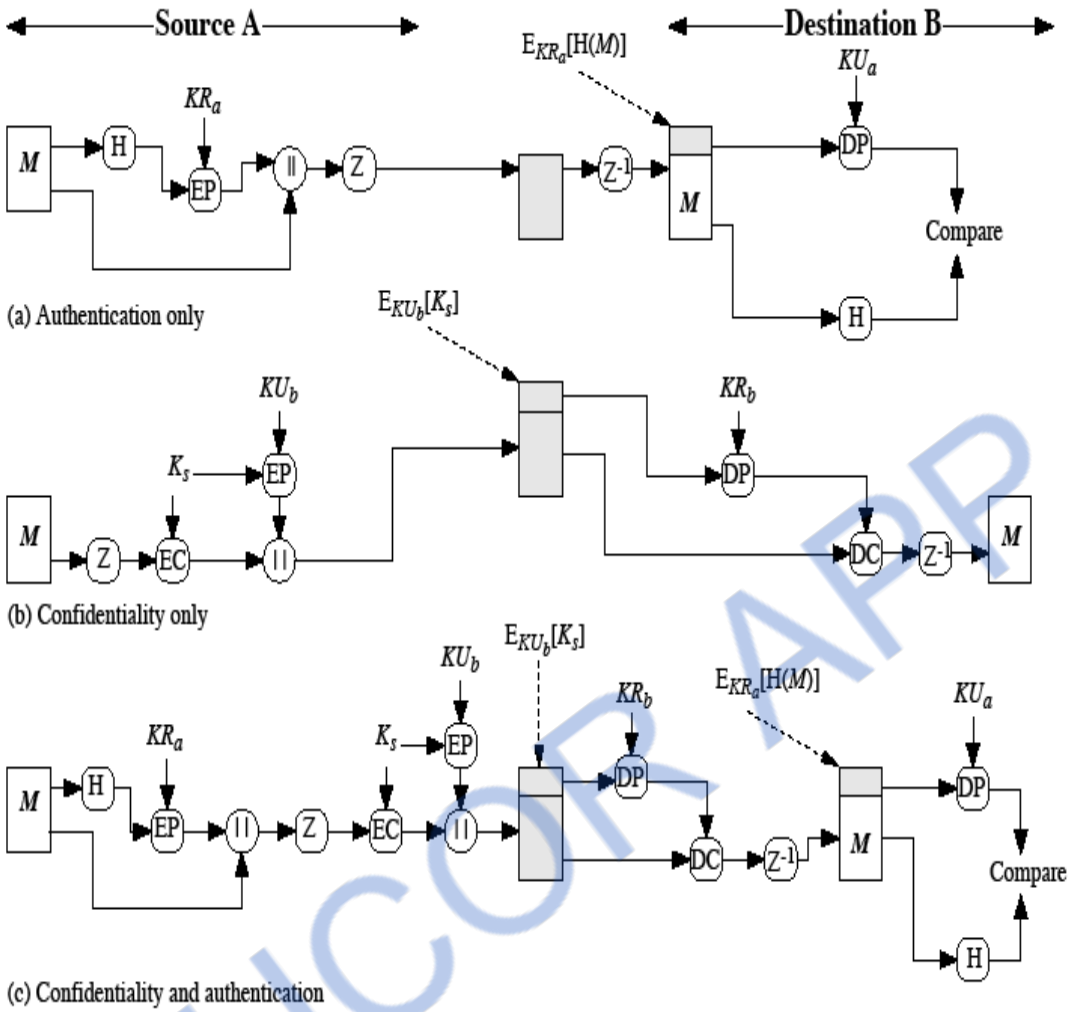


Fig 5.1: PGP Cryptographic Functions

4. E-mail compatibility

Many electronic mail systems only permit the use of blocks consisting of ASCII texts. To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters. The scheme used for this purpose is **radix-64 conversion**. Each group of three octets of binary data is mapped into four ASCII characters.

5. Segmentation and reassembly

E-mail facilities often are restricted to a maximum length. E.g., many of the facilities accessible through the internet impose a maximum length of 50,000 octets. Any message longer than that must be broken up into smaller segments, each of which is mailed separately.

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 the other processing, including the radix-64 conversion.

At the receiving end, PGP must strip off all e-mail headers and reassemble the entire original block before performing the other steps.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

5.1.1.2. Cryptographic keys and key rings

Three separate requirements can be identified with respect to these keys:

- A means of generating unpredictable session keys is needed.
- It must allow a user to have multiple public key/private key pairs.
- Each PGP entity must maintain a file of its own public/private key pairs as well as a file of public keys of correspondents.

a. Session key generation

Each session key is associated with a single message and is used only for the purpose of encryption and decryption of that message. Random 128-bit numbers are generated using CAST-128 itself.

The input to the random number generator consists of a 128-bit key and two 64-bit blocks that are treated as plaintext to be encrypted. Using cipher feedback mode, the CAST-128 produces two 64-bit cipher text blocks, which are concatenated to form the 128-bit session key. The plaintext input to CAST-128 is itself derived from a stream of 128-bit randomized numbers. These numbers are based on the keystroke input from the user.

b. Key identifiers

If multiple public/private key pair are used, then how does the recipient know which of the public keys was used to encrypt the session key?

One simple solution would be to transmit the public key with the message but, it is unnecessary wasteful of space. Another solution would be to associate an identifier with each public key that is unique at least within each user.

The solution adopted by PGP is to assign a key ID to each public key that is, with very high probability, unique within a user ID. The key ID associated with each public key consists of its least significant 64 bits. i.e., the key ID of public key KU_a is ($KU_a \bmod 2^{64}$).

A message consists of three components.

- **Message component** – includes actual data to be transmitted, as well as the filename and a timestamp that specifies the time of creation
- **Session key component** – includes session key and the identifier of the recipient public key.
- **Signature component** – includes the following
  - **Timestamp** – time at which the signature was made.
  - **Message digest** – hash code.
  - **Two octets of message digest** – to enable the recipient to determine if the correct public key was used to decrypt the message.
  - **Key ID of sender's public key** – identifies the public key

Notation:

- **EkU_b=** encryption with user B's Public key
- **EKR_a=** encryption with user A's private key
- **EK_s** = encryption with session key
- **ZIP** = Zip compression function
- **R64** = Radix- 64 conversion function

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

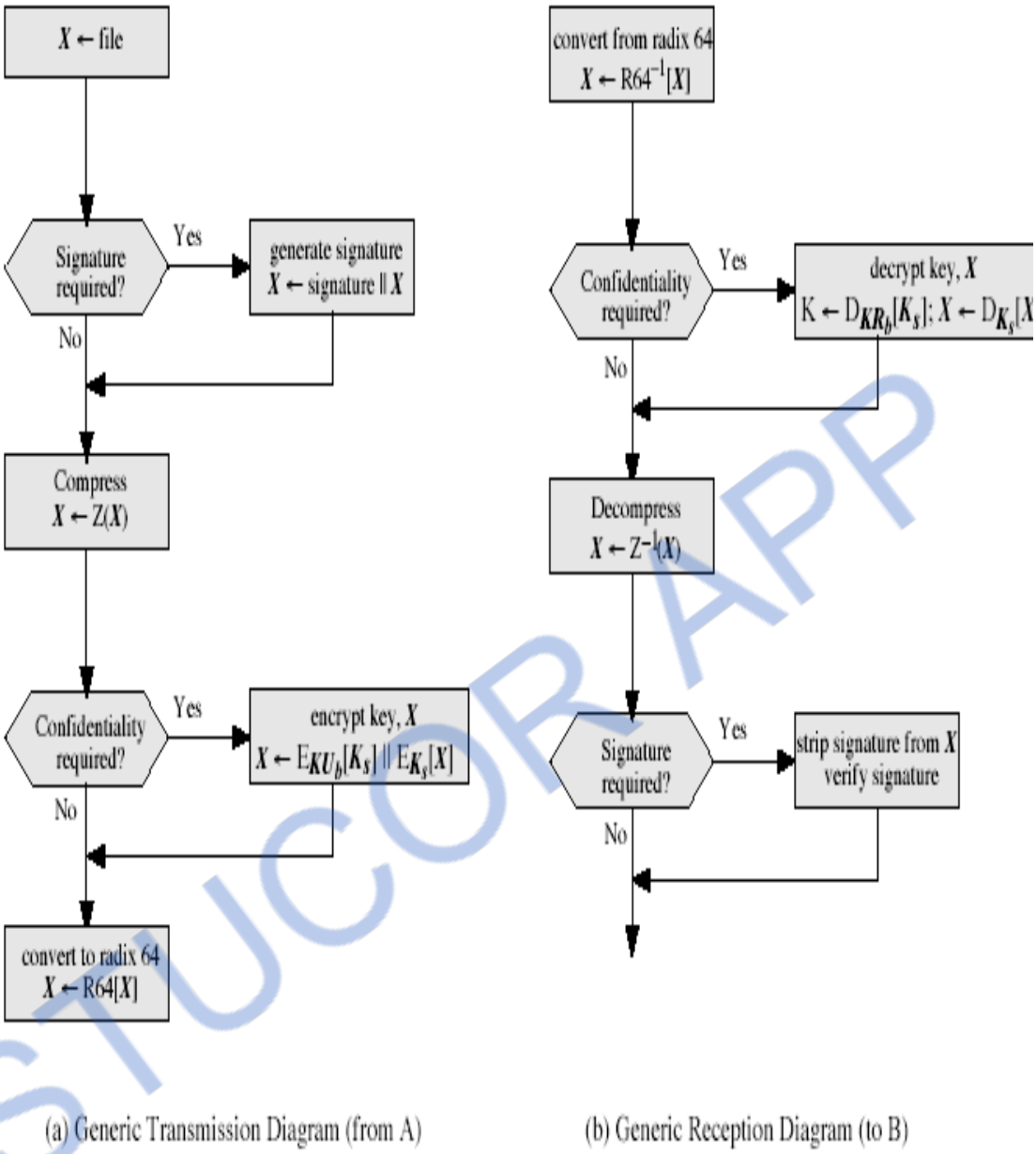


Fig 5.2: Transmission and Reception of PGP message

PGP provides a pair of data structures at each node, one to store the public/private key pair owned by that node and one to store the public keys of the other users known at that node. These data structures are referred to as private key ring and public key ring.

The general structures of the private and public key rings are shown below:

Timestamp - the date/time when this entry was made.
Key ID - the least significant bits of the public key.
Public key - public key portion of the pair.
Private Key - private key portion of the pair.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

User ID - the owner of the key
Key legitimacy field – indicates the extent to which PGP will trust that this is a valid public key for this user.

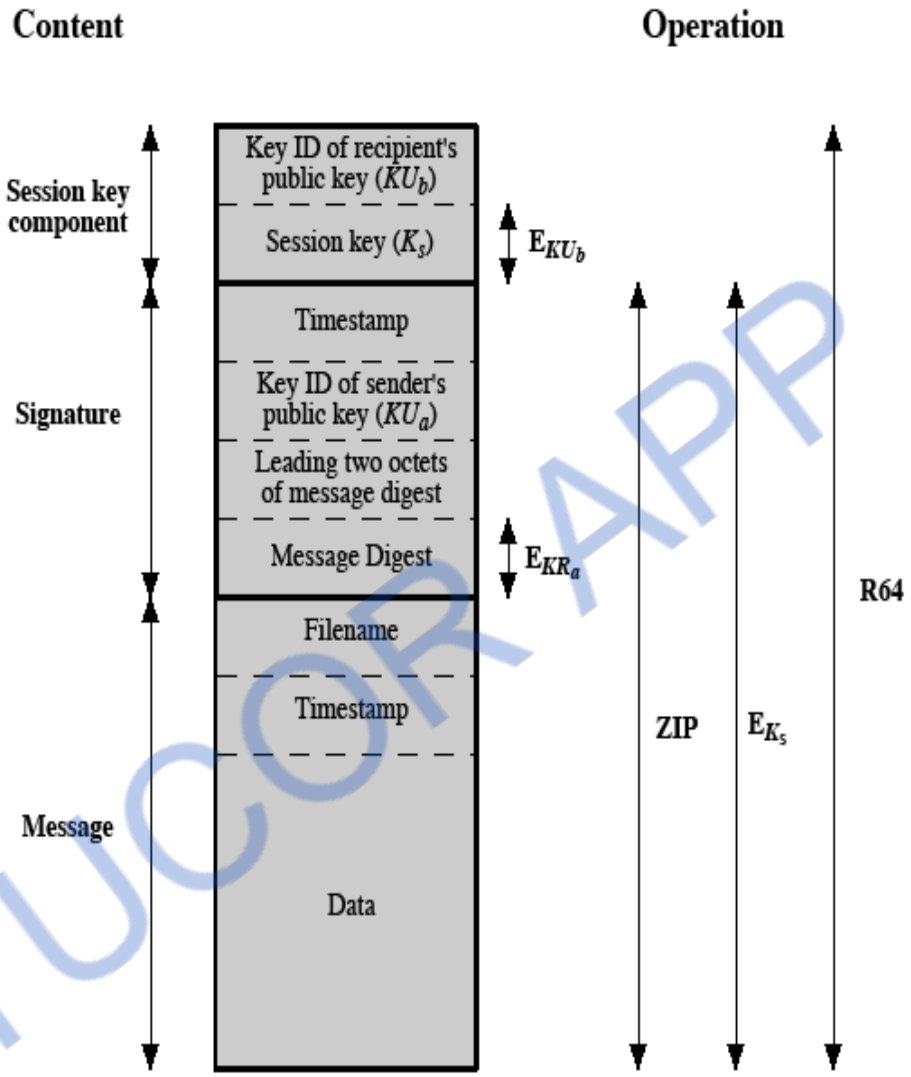


Fig 5.3: General Format of PGP message (From A to B)

Signature trust field – indicates the degree to which this PGP user trusts the signer to certify public key.

Owner trust field - indicates the degree to which this public key is trusted to sign other public key certificates.

PGP message generation First consider message transmission and assume that the message is to be both signed and encrypted. The sending PGP entity performs the following steps:

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1. Signing the message

- PGP retrieves the sender's private key from the private key ring using user ID as an index. If user ID was not provided, the first private key from the ring is retrieved.
- PGP prompts the user for the passphrase (password) to recover the unencrypted private key.
- The signature component of the message is constructed.

Private Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

* = field used to index table

Fig 5.4: General structure of private and public key Rings

2. Encrypting the message

- PGP generates a session key and encrypts the message.
- PGP retrieves the recipient's public key from the public key ring using user ID as index.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

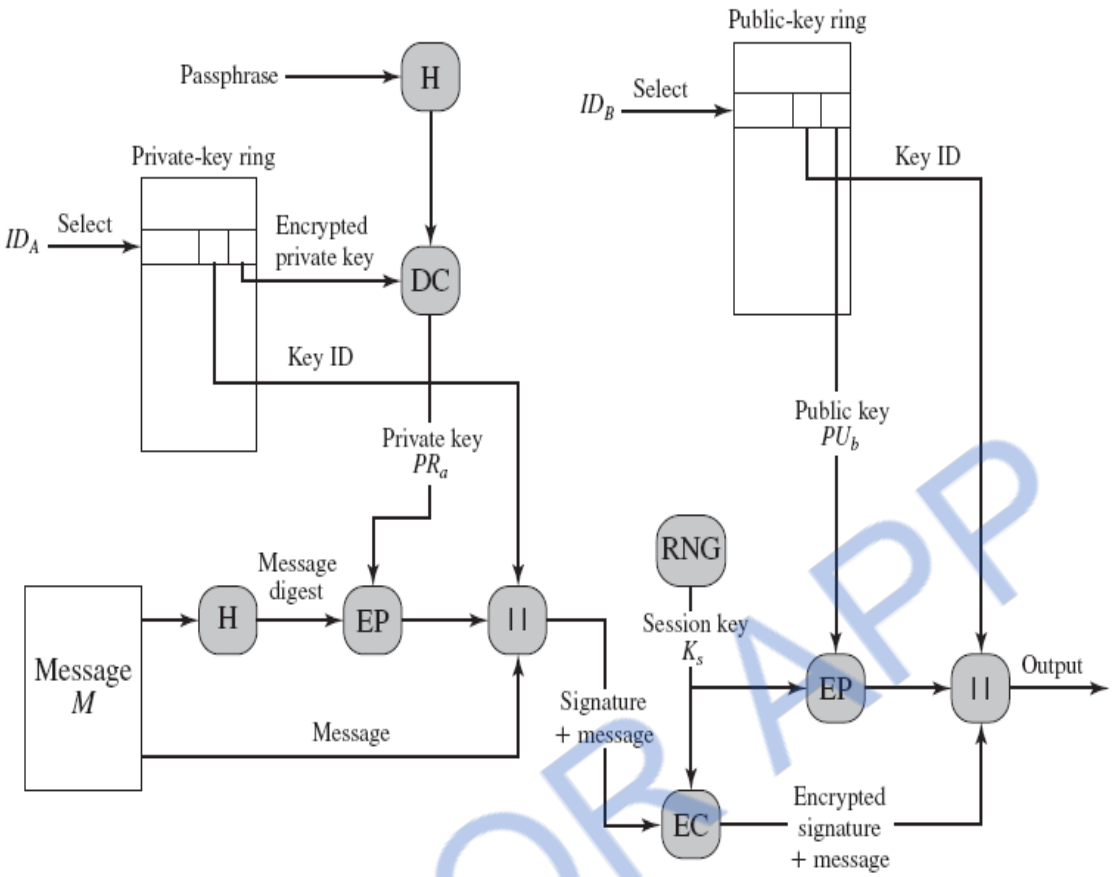


Fig 5.5: PGP message generation

The receiving PGP entity performs the following steps:

1. Decrypting the message

- PGP retrieves the receiver's private key from the private key ring, using the key ID field in the session key component of the message as an index.
- PGP prompts the user for the passphrase (password) to recover the unencrypted private key.
- PGP then recovers the session key and decrypts the message.

2. Authenticating the message

- PGP retrieves the sender's public key from the public key ring, using the key ID field in the signature key component of the message as an index.
- PGP recovers the transmitted message digest.
- PGP computes the message digest for the received message and compares it to the transmitted message digest to authenticate.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

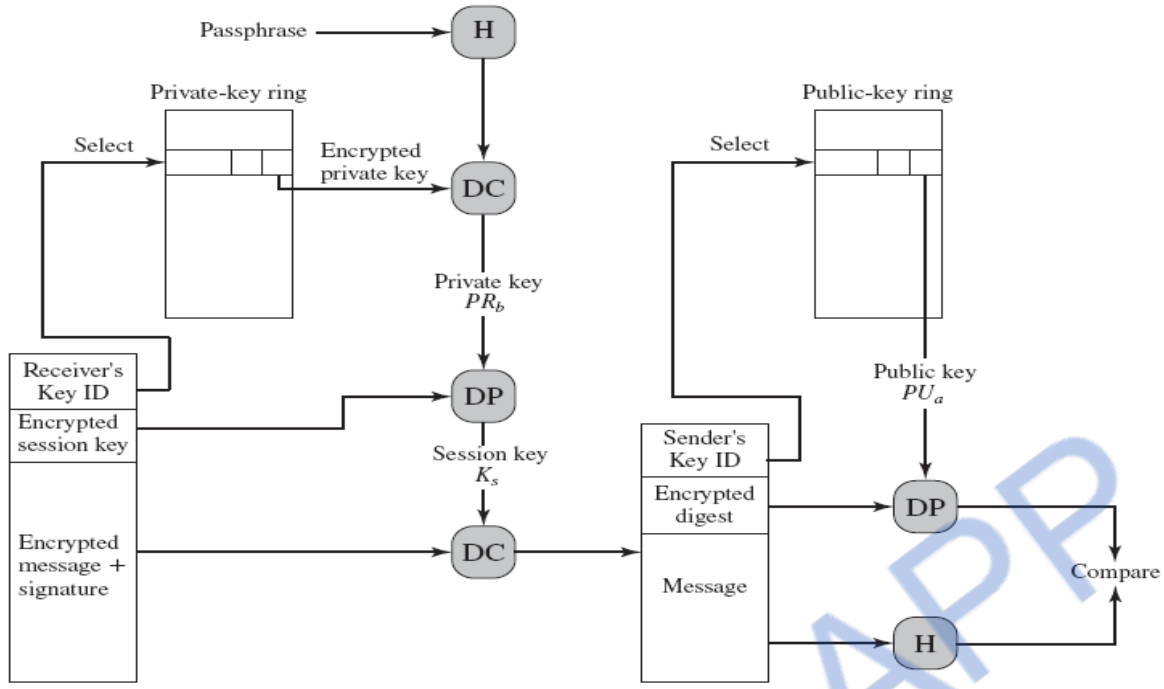


Fig 5.6: PGP message reception

5.1.2. S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extension) is a security enhancement to the MIME Internet e-mail format standard, based on technology from RSA Data Security.

5.1.2.1 Multipurpose Internet Mail Extensions

MIME is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of the use of SMTP (Simple Mail Transfer Protocol) or some other mail transfer protocol and RFC 822 for electronic mail.

Following are the limitations of SMTP/822 scheme:

1. SMTP cannot transmit executable files or other binary objects.
2. SMTP cannot transmit text data that includes national language characters because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail message over a certain size.
4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
5. SMTP gateways to X.400 electronic mail networks cannot handle non textual data included in X.400 messages.
6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821. Common problems include:

- Deletion, addition, or reordering of carriage return and linefeed
- Truncating or wrapping lines longer than 76 characters
- Removal of trailing white space (tab and space characters)
- Padding of lines in a message to the same length

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- Conversion of tab characters into multiple space characters

MIME is intended to resolve these problems in a manner that is compatible with existing RFC 822 implementations. The specification is provided in RFCs 2045 through 2049.

5.1.3 OVERVIEW

The MIME specification includes the following elements:

1. **Five new message header** fields are defined, which may be included in an RFC 822 header. These fields provide information about the body of the message.
2. **A number of content formats** are defined, thus standardizing representations that support multimedia electronic mail.
3. **Transfer encodings** are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

In this subsection, we introduce the five message header fields. The next two subsections deal with content formats and transfer encodings.

The five header fields defined in MIME are as follows:

- **MIME-Version**: Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.
- **Content-Type**: Describes the data contained in the body with sufficient detail.
- **Content-Transfer-Encoding**: Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.
- **Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
- **Content-Description**: A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

5.1.3.1 MIME Content Types

There are seven different major types of content and a total of 15 subtypes

MIME Content Types		
Type	**Subtype**	**Description**
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	Gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript.
	octet-stream	General binary data consisting of 8-bit bytes.

For the **text type** of body, no special software is required to get the full meaning of the text, aside from support of the indicated character set. The primary subtype is plain text, which is simply a string of ASCII characters or ISO 8859 characters. The enriched subtype allows greater formatting flexibility.

The **multipart type** indicates that the body contains multiple, independent parts. The Content-Type header field includes a parameter, called boundary, that defines the delimiter between body parts.

The **multipart/digest subtype** is used when each of the body parts is interpreted as an RFC 822 message with headers. This subtype enables the construction of a message whose parts are individual messages. For example, the moderator of a group might collect e-mail messages from participants, bundle these messages, and send them out in one encapsulating MIME message.

The **message type** provides a number of important capabilities in MIME. The message/rfc822 subtype indicates that the body is an entire message, including header and body. Despite the name of this subtype, the encapsulated message may be not only a simple RFC 822 message, but also any MIME message.

The **message/partial subtype** enables fragmentation of a large message into a number of parts, which must be reassembled at the destination. For this subtype, three parameters are specified in the Content-Type: Message/Partial field: an id common to all fragments of the same message, a sequence number unique to each fragment, and the total number of fragments.

The **message/external-body subtype** indicates that the actual data to be conveyed in this message are not contained in the body. Instead, the body contains the information needed to access the data.

5.1.3.2. MIME Transfer Encodings

MIME Transfer Encodings

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present but the lines are not necessarily short enough for SMTP transport.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

The quoted-printable transfer encoding is useful when the data consists largely of octets that correspond to printable ASCII characters. In essence, it represents non safe characters by the hexadecimal representation of their code and introduces reversible (soft) line breaks to limit message lines to 76 characters.

The base64 transfer encoding, also known as radix-64 encoding, is a common one for encoding arbitrary binary data in such a way as to be invulnerable to the processing by mail transport programs.

5.1.3.3. Canonical Form

An important concept in MIME and S/MIME is that of canonical form. Canonical form is a format, appropriate to the content type that is standardized for use between systems. This is in contrast to native form, which is a format that may be peculiar to a particular system.

5.1.3.4. S/MIME Functionality

In terms of general functionality, S/MIME is very similar to PGP. Both offer the ability to sign and/or encrypt messages.

Functions:

S/MIME provides the following functions:

- **Enveloped data:** This consists of encrypted content of any type and encrypted-content encryption keys for one or more recipients.
- **Signed data**: A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base 64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.
- **Clear-signed data:** As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base 64. As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.
- **Signed and enveloped data:** Encrypted data may be signed and signed data or clear-signed data may be encrypted.

5.1.3.5. Cryptographic Algorithms

Table 1 summarizes the cryptographic algorithms used in S/MIME. S/MIME uses the following terminology, taken from RFC 2119 to specify the requirement level:

- **Must:** The definition is an absolute requirement of the specification. An implementation must include this feature or function to be in conformance with the specification.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- **Should:** There may exist valid reasons in particular circumstances to ignore this feature or function, but it is recommended that an implementation include the feature or function.

Table1: Cryptographic Algorithms Used in S/MIME	
Function	**Requirement**
Create a message digest to be used in forming a digital signature. Encrypt message digest to form digital signature.	MUST support SHA-1. Receiver SHOULD support MD5 for backward compatibility. Sending and receiving agents MUST support DSS. Sending agents SHOULD support RSA encryption. Receiving agents SHOULD support verification of RSA signatures with key sizes 512 bits to 1024 bits.
Encrypt session key for transmission with message.	Sending and receiving agents SHOULD support Diffie-Hellman. Sending and receiving agents MUST support RSA encryption with key sizes 512 bits to 1024 bits.
Encrypt message for transmission with one-time session key.	Sending and receiving agents MUST support encryption with triple DES Sending agents SHOULD support encryption with AES. Sending agents SHOULD support encryption with RC2/40.
Create a message authentication code	Receiving agents MUST support HMAC with SHA-1. Receiving agents SHOULD support HMAC with SHA-1.

5.1.3.6 S/MIME MESSAGES

S/MIME makes use of a number of new MIME content types, which are shown in Table 2. All of the new application types use the designation PKCS. This refers to a set of public-key cryptography specifications issued by RSA Laboratories and made available for the S/MIME effort.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Table 2: S/MIMEContent Types			
Type	**Subtype**	**SMIME Parameter**	**Description**
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	PKCS 7-MIME	Signed Data	A signed S/MIME entity.
	PKCS 7-MIME	Enveloped Data	An encrypted S/MIME entity.
	PKCS 7-MIME	degenerate signed Data	An entity containing only public-key certificates.
	PKCS 7-MIME	Compressed Data	A compressed S/MIME entity
	PKCS 7-SIGNATURE	signed Data	The content type of the signature subpart of a multipart/signed message.

5.1.4 SECURING A MIME ENTITY

S/MIME secures a MIME entity with a signature, encryption, or both. A MIME entity may be an entire message (except for the RFC 822 headers), or if the MIME content type is multipart, then a MIME entity is one or more of the subparts of the message. In all cases, the message to be sent is converted to canonical form.

In particular, for a given type and subtype, the appropriate canonical form is used for the message content. For a multipart message, the appropriate canonical form is used for each subpart.

The use of transfer encoding requires special attention.

1) Enveloped Data

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm (RC2/40 or triple DES).
2. For each recipient, encrypt the session key with the recipient's public RSA key.
3. For each recipient, prepare a block known as Recipient Info that contains an identifier of the recipient's public-key certificate, an identifier of the algorithm used to encrypt the session key, and the encrypted session key.
4. Encrypt the message content with the session key.

The Recipient Info blocks followed by the encrypted content constitute the enveloped Data. This information is then encoded into base 64. To recover the encrypted message, the recipient first strips off the base64 encoding. Then the recipient's private key is used to recover the session key. Finally, the message content is decrypted with the session key.

2) Signed Data

The steps for preparing a signed Data MIME entity are as follows:

- Select a message digest algorithm (SHA or MD5).

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- Compute the message digest, or hash function, of the content to be signed
- Encrypt the message digest with the signer's private key.
-
- 4. Prepare a block known as Signer Info that contains the signer's public-key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest

To recover the signed message and verify the signature, the recipient first strips off the base64 encoding. Then the signer's public key is used to decrypt the message digest.

The recipient independently computes the message digest and compares it to the decrypted message digest to verify the signature.

3) Clear Signing

- Clear signing is achieved using the multipart content type with a signed subtype.
- As was mentioned, this signing process does not involve transforming the message to be signed, so that the message is sent "in the clear."
- Thus, recipients with MIME capability but not S/MIME capability are able to read the incoming message.

A multipart/signed message has two parts.

The first part can be any MIME type but must be prepared so that it will not be altered during transfer from source to destination. This means that if the first part is not 7bit, then it needs to be encoded using base64 or quoted-printable.

This second part has a MIME content type of application and a subtype of PKCS7-signature The protocol parameter indicates that this is a two-part clear-signed entity. The receiver can verify the signature by taking the message digest of the first part and comparing this to the message digest recovered from the signature in the second part.

5.1.4.1 Registration Request

The user will apply to a certification authority for a public-key certificate. The S/MIME entity is used to transfer a certification request.

- The certification request includes certification Request Info block, followed by an identifier of the public-key encryption algorithm, followed by the signature of the certification Request Info block, made using the sender's private key.
- The certification Request Info block includes a name of the certificate subject (the entity whose public key is to be certified) and a bit-string representation of the user's public key.

Certificates-Only Message

A message containing only certificates or a certificate revocation list (CRL) can be sent in response to a registration request. The message is an application/PKCS7-MIME type/subtype with an SMIME-type parameter of degenerate. The steps involved are the same as those for creating a signed Data message, except that there is no message content and the signer Info field is empty.

S/MIME Certificate Processing

S/MIME uses public-key certificates that conform to version 3 of X.509 The key-management scheme used by S/MIME is in some ways a hybrid between a strict X.509 certification hierarchy and PGP's web of trust.

User Agent Role

An S/MIME user has **several key-management functions** to perform:

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Key generation:

The user of some related administrative utility (e.g., one associated with LAN management) MUST be capable of generating a key pair from a good source of nondeterministic random input and be protected in a secure fashion.

Registration:

A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate.

Certificate storage and retrieval:

A user requires access to a local list of certificates in order to verify incoming signatures and to encrypt outgoing messages. Such a list could be maintained by the user or by some local administrative entity on behalf of a number of users.

VeriSign Certificates

There are several companies that provide certification authority (CA) services. There are a number of Internet-based CAs, including VeriSign, GTE, and the U.S. Postal Service.

VeriSign provides a CA service that is intended to be compatible with S/MIME and a variety of other applications. VeriSign issues X.509 certificates with the product name VeriSign Digital ID.

The information contained in a Digital ID depends on the type of Digital ID and its use. At a minimum, each Digital ID contains

- Owner's public key
- Owner's name or alias
- Expiration date of the Digital ID
- Serial number of the Digital ID
- Name of the certification authority that issued the Digital ID
- Digital signature of the certification authority that issued the Digital ID

Digital IDs can also contain other user-supplied information, including

- Address
- E-mail address
- Basic registration information (country, zip code, age, and gender)

VeriSign provides three levels, or classes, of security for public-key certificates. A user requests a certificate online at VeriSign's Web site or other participating Web sites. Class 1 and Class 2 requests are processed on line, and in most cases take only a few seconds to approve. Briefly, the following procedures are used:

- For Class 1 Digital IDs, VeriSign confirms the user's e-mail address by sending a PIN and Digital ID pick-up information to the e-mail address provided in the application.

- For Class 2 Digital IDs, VeriSign verifies the information in the application through an automated comparison with a consumer database in addition to performing all of the checking associated with a Class 1 Digital ID.

    - Finally, confirmation is sent to the specified postal address alerting the user that a Digital ID has been issued in his or her name.

- For Class 3 Digital IDs, VeriSign requires a higher level of identity assurance. An individual must prove his or her identity by providing notarized credentials or applying in person.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

5.1.4.2 Enhanced Security Services

Three enhanced security services have been proposed in an Internet draft.

Signed receipts:

A signed receipt may be requested in a Signed Data object. Returning a signed receipt provides proof of delivery to the originator of a message and allows the originator to demonstrate to a third party that the recipient received the message.

Security labels:

A security label may be included in the authenticated attributes of a SignedData object. A security label is a set of security information regarding the sensitivity of the content that is protected by S/MIME encapsulation. The labels may be used for access control, by indicating which users are permitted access to an object.

Secure mailing lists:

When a user sends a message to multiple recipients, a certain amount of per-recipient processing is required, including the use of each recipient's public key.

The user can be relieved of this work by employing the services of an S/MIME Mail List Agent (MLA). An MLA can take a single incoming message, perform the recipient-specific encryption for each recipient, and forward the message.

The originator of a message need only send the message to the MLA, with encryption performed using the MLA's public key.

5.1.5 NON REPUDIATION

Non-repudiation is the assurance that someone cannot deny something. Typically, non-repudiation refers to the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated.

To repudiate means to deny. On the Internet, a digital signature is used not only to ensure that a message or document has been electronically signed by the person that purported to sign the document, but also, since a digital signature can only be created by one person, to ensure that a person cannot later deny that they furnished the signature.

Since no security technology is absolutely fool-proof, some experts warn that a digital signature alone may not always guarantee non-repudiation. It is suggested that multiple approaches be used, such as capturing unique biometric information and other data about the sender or signer that collectively would be difficult to repudiate.

Email non-repudiation involves methods such as email tracking that is designed to ensure that the sender cannot deny having sent a message and/or that the recipient cannot deny having received it

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

5.2 IP SECURITY

5.2.1 OVERVIEW OF IPSEC

5.2.1.1 Applications of IPSec

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include the following:

- Secure branch office connectivity over the Internet
- 2 Secure remote access over the Internet
- Establishing extranet and intranet connectivity with partners
- Enhancing electronic commerce security

5.2.1.2 Benefits of IPSec:

- When IPSec is implemented in a firewall or router, it provides strong security
- IPSec in a firewall is resistant to bypass if all traffic from the outside must use IP, and the firewall is the only means of entrance from the Internet into the organization.
- IPSec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPSec is implemented in the firewall or router.
- IPSec can be transparent to end users. There is no need to train users on security mechanisms
- IPSec can provide security for individual users if needed.

5.2.1.3 Routing Applications

IPSec can play a vital role in the routing architecture required for internet working.

The following are examples of the use of IPSec. IPSec can assure that

- A router advertisement (a new router advertises its presence) comes from an authorized router
- A neighbor advertisement (a router seeks to establish or maintain a neighbor relationship with a router in another routing domain) comes from an authorized router.
- A redirect message comes from the router to which the initial packet was sent.
- A routing update is not forged.

5.2.2 IP SECURITY ARCHITECTURE

5.2.2.1 IPSec Documents

The IPSec specification consists of numerous documents. The most important of these, issued in November of 1998, are RFCs 2401, 2402, 2406, and 2408:

- **RFC 2401:** An overview of a security architecture
- **RFC 2402:** Description of a packet authentication extension to IPv4 and IPv6
- **RFC 2406:** Description of a packet encryption extension to IPv4 and IPv6
- **RFC 2408:** Specification of key management capabilities

The documents are divided into seven groups:

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

5.2.2.2 Architecture

Covers the general concepts, security requirements, definitions, and mechanisms defining IPSec technology.

Encapsulating Security Payload (ESP):

Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.

Authentication Header (AH):

Covers the packet format and general issues related to the use of AH for packet authentication.

Encryption Algorithm:

A set of documents that describe how various encryption algorithms are used for ESP.

Authentication Algorithm:

A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.

Key Management:

Documents that describe key management schemes.

Domain of Interpretation (DOI):

Contains values needed for the other documents to relate to each other. These include identifiers for approved encryption and authentication algorithms, as well as operational parameters such as key lifetime.

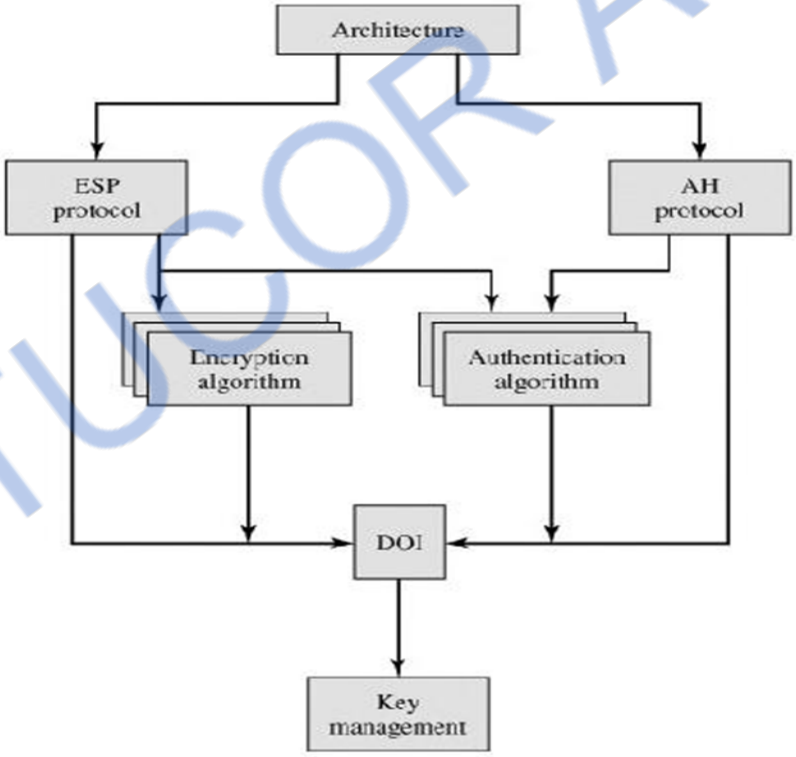


Fig 5.7: IP security Document overview

IPSec Services

IPSec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Two protocols are used to provide security:

- An authentication protocol: Designated by the header of the protocol, Authentication Header (AH);
- Encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP).

The services are

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Security Associations

A key concept that appears in both the authentication and confidentiality mechanisms for IP is the security association (SA). An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it.

A security association is uniquely identified by three parameters:

- Security Parameters Index (SPI)
- IP Destination Address
- Security Protocol Identifier

SA Parameters

A security association is normally defined by the following parameters:

a) Sequence Number Counter:

A 32-bit value used to generate the Sequence Number field in AH or ESP headers.

b) Sequence Counter Overflow:

A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA.

c) Anti-Replay Window:

Used to determine whether an inbound AH or ESP packet is a replay.

d) AH Information:

Authentication algorithm, keys, key lifetimes, and related parameters being used with AH (required for AH implementations).

e) ESP Information:

Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP (required for ESP implementations).

f) Lifetime of This Security Association:

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

A time interval or byte count after which an SA must be replaced with a new SA or terminated, plus an indication of which of these actions should occur .

g) IPSec Protocol Mode: Tunnel, transport.
h) Path MTU: Any observed path maximum transmission unit and aging variables.

5.2.2.3. Modes of Transfer

Both AH and ESP support two modes of use: transport and tunnel mode.

Transport Mode:

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet.

Tunnel Mode:

Tunnel mode provides protection to the entire IP packet. To achieve this, after the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new "outer" IP packet with a new outer IP header.

The entire original, or inner, packet travels through a "tunnel" from one point of an IP network to another; no routers along the way are able to examine the inner IP header. Because the original packet is encapsulated, the new, larger packet may have totally different source and destination addresses, adding to the security.

5.2.2.4. Authentication Header

The Authentication Header provides support for data integrity and authentication of IP packets. The Authentication Header consists of the following fields:

- **Next Header (8 bits):** Identifies the type of header immediately following this header.
- **Payload Length (8 bits):** Length of Authentication Header in 32-bit words, minus 2.
- **Reserved (16 bits):** For future use.
- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value.
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value (ICV), or MAC

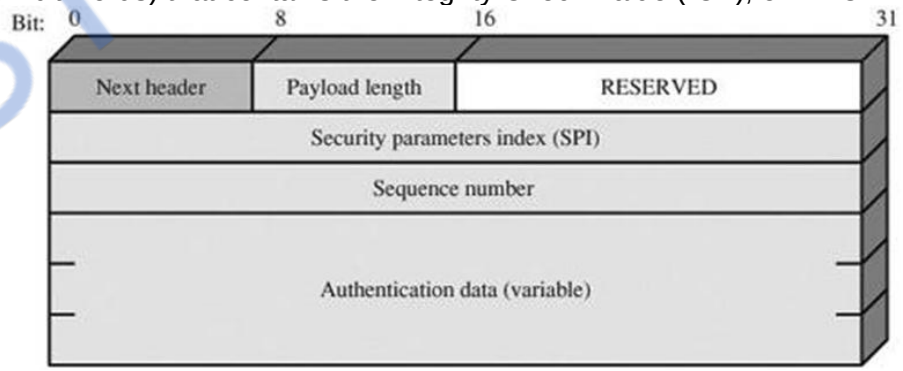


Fig 5.8: IPSec Authentication Header

Anti-Replay Service

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

When a new SA is established, the sender initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1.

If anti-replay is enabled (the default), the sender must not allow the sequence number to cycle past 2^{32}-1 back to zero. Otherwise, there would be multiple valid packets with the same sequence number. If the limit of 2^{32}-1 is reached, the sender should terminate this SA and negotiate a new SA with a new key.

Integrity Check Value

The Authentication Data field holds a value referred to as the Integrity Check Value. The ICV is a message authentication code or a truncated version of a code produced by a MAC algorithm.

Transport and Tunnel Modes

For transport mode AH using IPv4, the AH is inserted after the original IP header and before the IP payload

For tunnel mode AH, the entire original IP packet is authenticated, and the AH is inserted between the original IP header and a new outer IP header

5.2.2.5. Encapsulating Security Payload

The Encapsulating Security Payload provides confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality.

The diagram shows the format of an ESP packet. It contains the following fields:

- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.
- **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
- **Padding (0255 bytes):**The purpose of this field is discussed later.
- **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.
- **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP).
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

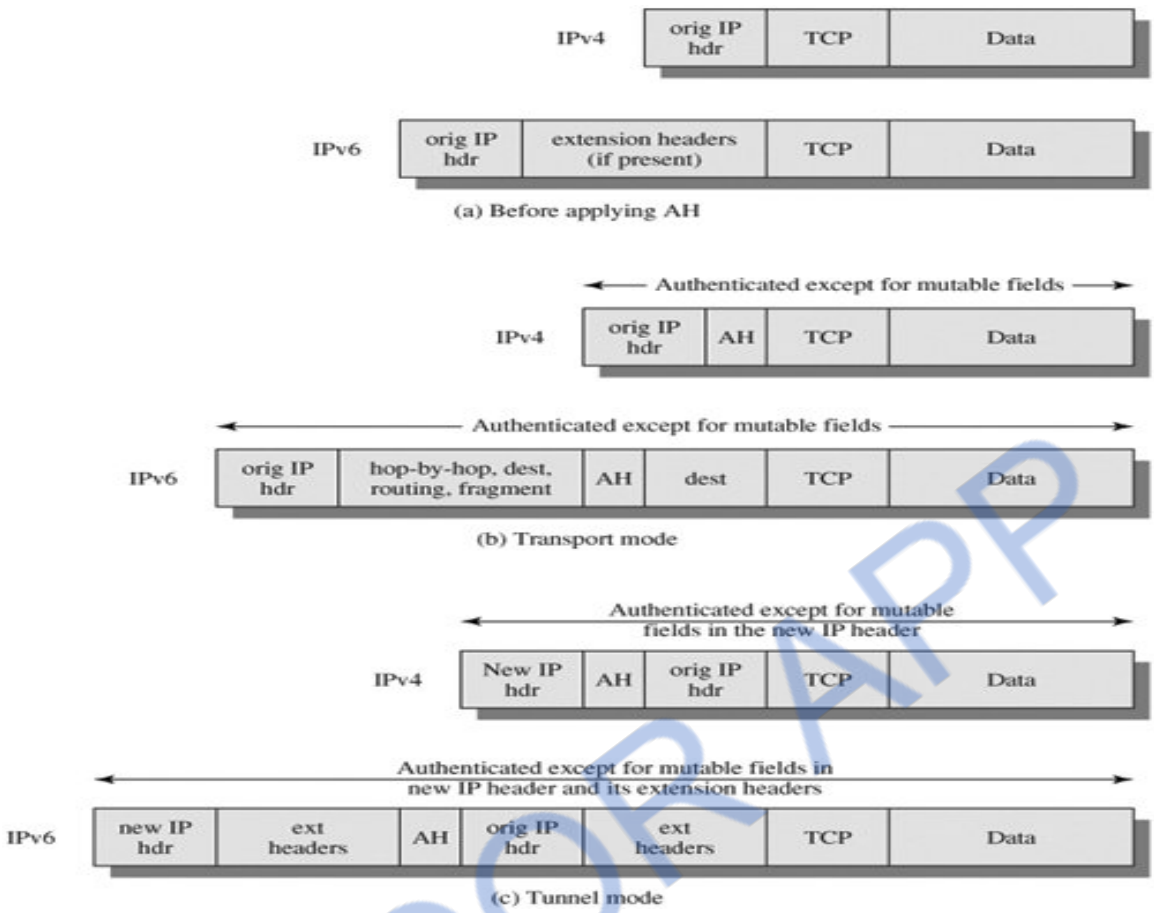


Fig 5.9: Scope of AH Authentication

Padding:

The Padding field serves several purposes:

- If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.
- The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment.
- Additional padding may be added to provide partial traffic flow confidentiality by concealing the actual length of the payload.

Transport and Tunnel Modes

ESP service can be used. In the upper part of the figure, encryption (and optionally authentication) is provided directly between two hosts.

The diagram shows how tunnel mode operation can be used to set up a virtual private network.

In this example, an organization has four private networks interconnected across the Internet.

Hosts on the internal networks use the Internet for transport of data but do not interact with other Internet-based hosts.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

By terminating the tunnels at the security gateway to each internal network, the configuration allows the hosts to avoid implementing the security capability. The former technique is support by a transport mode SA, while the latter technique uses a tunnel mode SA.

Transport Mode ESP:

For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (e.g., TCP, UDP, ICMP) and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet; if authentication is selected, the ESP Authentication Data field is added after the ESP trailer.

The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the cipher text plus the ESP header.

For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (e.g., TCP, UDP, ICMP) and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet; if authentication is selected, the ESP Authentication Data field is added after the ESP trailer.

The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the cipher text plus the ESP header.

Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application. This mode of operation is also reasonably efficient, adding little to the total length of the IP packet. One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets.

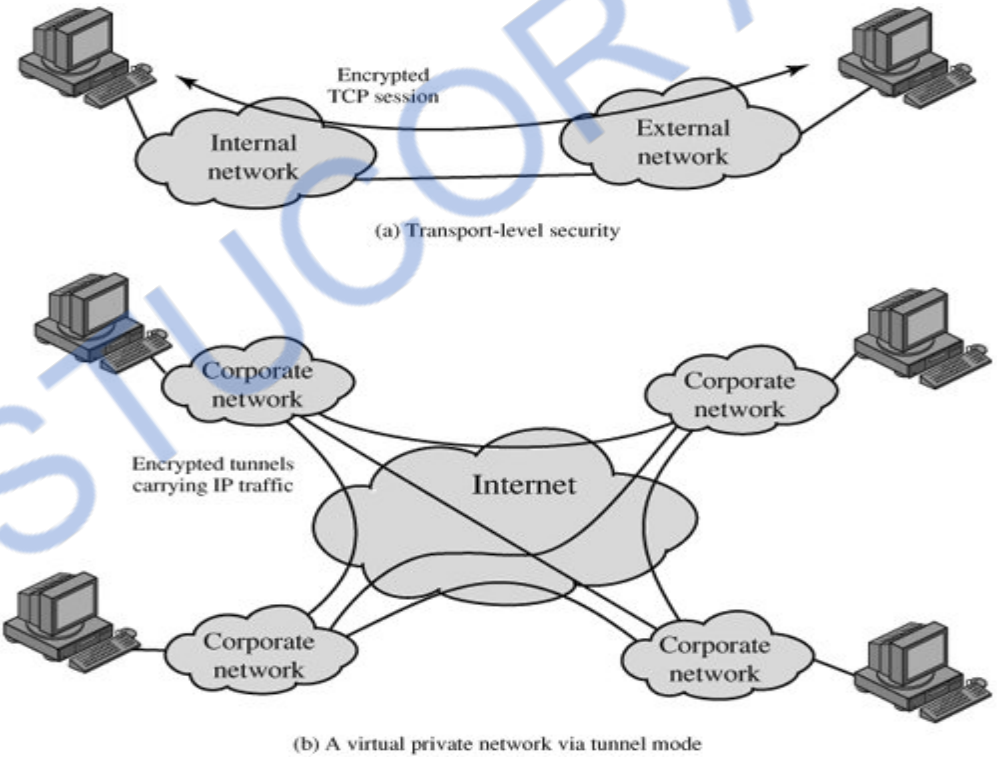


Fig 5.10: Transport Mode

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Tunnel Mode ESP

Tunnel mode ESP is used to encrypt an entire IP packet. For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted. This method can be used to counter traffic analysis.

Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header. Intermediate routers would be unable to process such a packet.

Therefore, it is necessary to encapsulate the entire block (ESP header plus cipher text plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis.

5.2.3 COMBINING SECURITY ASSOCIATIONS

An individual SA can implement either the AH or ESP protocol but not both. Sometimes a particular traffic flow will call for the services provided by both AH and ESP. Further, a particular traffic flow may require IPsec services between hosts and, for that same flow, separate services between security gateways, such as firewalls.

Security associations may be combined into bundles in two ways:

Transport adjacency:

Refers to applying more than one security protocol to the same IP packet without invoking tunneling. This approach to combining AH and ESP allows for only one level of combination; further nesting yields no added benefit since the processing is performed at one IPsec instance: the (ultimate) destination.

Iterated tunneling:

Refers to the application of multiple layers of security protocols effected through IP tunneling. This approach allows for multiple levels of nesting, since each tunnel can originate or terminate at a different IPsec site along the path.

The two approaches can be combined, for example, by having a transport SA between hosts travel part of the way through a tunnel SA between security gateways.

One interesting issue that arises when considering SA bundles is the order in which authentication and encryption may be applied between a given pair of endpoints and the ways of doing so. We examine that issue next. Then we look at combinations of SAs that involve at least one tunnel.

Authentication plus Confidentiality

Encryption and authentication can be combined in order to transmit an IP packet that has both confidentiality and authentication between hosts. We look at several approaches.

5.2.3.1 ESP with Authentication Option

This approach is illustrated in diagram. In this approach, the user first applies ESP to the data to be protected and then appends the authentication data field. There are actually two subcases:

Transport mode ESP: Authentication and encryption apply to the IP payload delivered to the host, but the IP header is not protected.

Tunnel mode ESP: Authentication applies to the entire IP packet delivered to the outer IP destination address (e.g., a firewall), and authentication is performed at that destination. The entire inner IP packet is protected by the privacy mechanism for delivery to the inner IP destination.

For both cases, authentication applies to the cipher text rather than the plaintext.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Transport Adjacency

Another way to apply authentication after encryption is to use two bundled transport SAs, with the inner being an ESP SA and the outer being an AH SA. In this case, ESP is used without its authentication option. Because the inner SA is a transport SA, encryption is applied to the IP payload. The resulting packet consists of an IP header (and possibly IPv6 header extensions) followed by an ESP.AH is then applied in transport mode,

so that authentication covers the ESP plus the original IP header (and extensions) except for mutable fields. The advantage of this approach over simply using a single ESP SA with the ESP authentication option is that the authentication covers more fields, including the source and destination IP addresses.The disadvantage is the overhead of two SAs versus one SA.

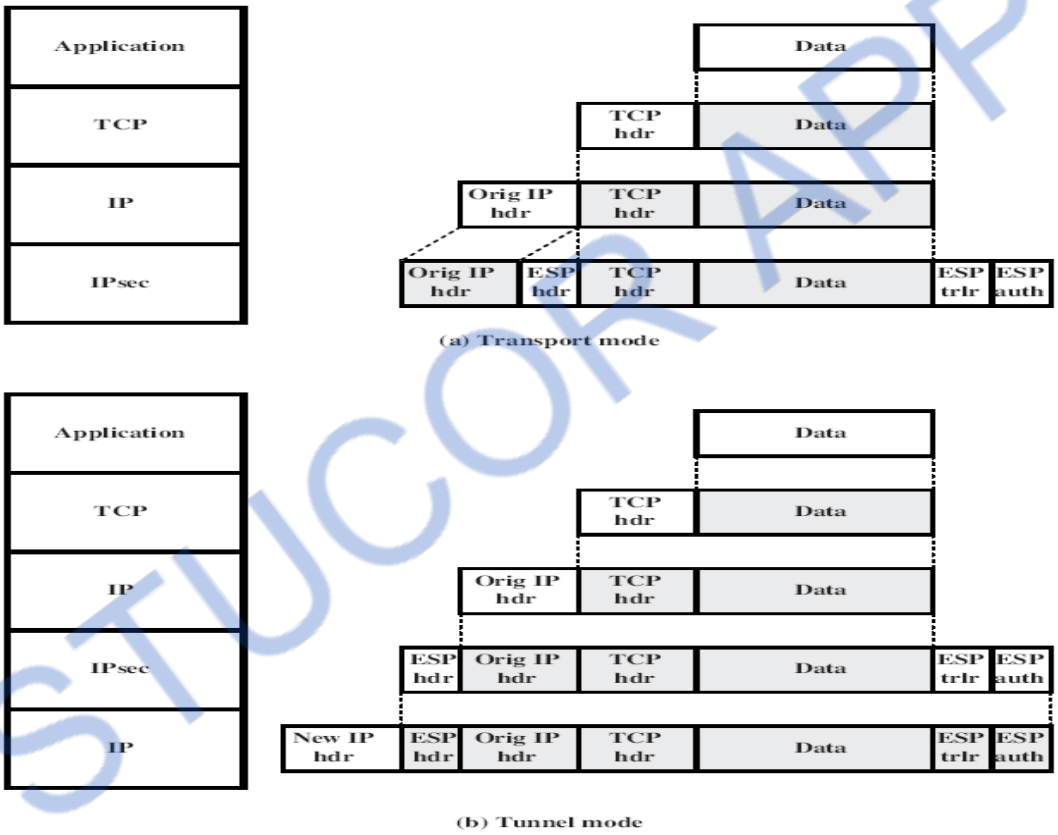


Fig 5.11: Protocol Operation for ESP

Basic Combinations of Security Associations

The IPsec Architecture document lists four examples of combinations of SAs that must be supported by compliant IPsec hosts (e.g., workstation, server) or security gateways (e.g. firewall, router).These are illustrated in Figure .

The lower part of each case in the figure represents the physical connectivity of the elements; the upper part represents logical connectivity via one or more nested SAs. Each SA

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

can be either AH or ESP. For host-to-host SAs, the mode may be either transport or tunnel; otherwise it must be tunnel mode.

Case 1: All security is provided between end systems that implement IPsec.

For any two end systems to communicate via an SA, they must share the appropriate secret keys. Among the possible combinations are

- AH in transport mode
- ESP in transport mode
- ESP followed by AH in transport mode (an ESP SA inside an AH SA)
- Any one of a, b, or c inside an AH or ESP in tunnel mode

We have already discussed how these various combinations can be used to support authentication, encryption, authentication before encryption, and authentication after encryption.

Case 2: Security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec. This case illustrates simple virtual private network support. The security architecture document specifies that only a single tunnel SA is needed for this case. The tunnel could support AH, ESP, or ESP with the authentication option. Nested tunnels are not required, because the IPsec services apply to the entire inner packet.

Case 3: This builds on case 2 by adding end-to-end security. The same combinations discussed for cases 1 and 2 are allowed here. The gateway-to-gateway tunnel provides either authentication, confidentiality, or both for all traffic between end systems.When the gateway-to-gateway tunnel is ESP, it also provides a limited form of traffic confidentiality. Individual hosts can implement any additional IPsec services required for given applications or given users by means of end-to-end SAs.

Case 4: This provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall. Only tunnel mode is required between the remote host and the firewall. As in case 1, one or two SAs may be used between the remote host and the local host.

5.2.4 KEY MANAGEMENT

The key management portion of IPSec involves the determination and distribution of secret keys. Two types of key management:

Manual: A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.

Automated: An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration.

The default automated key management protocol for IPSec is referred to as ISAKMP/Oakley and consists of the following elements:

Oakley Key Determination Protocol: Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but providing added security. Oakley is generic in that it does not dictate specific formats.

Internet Security Association and Key Management Protocol (ISAKMP):ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

(firewall, router).These are illustrated in Figure .The lower part

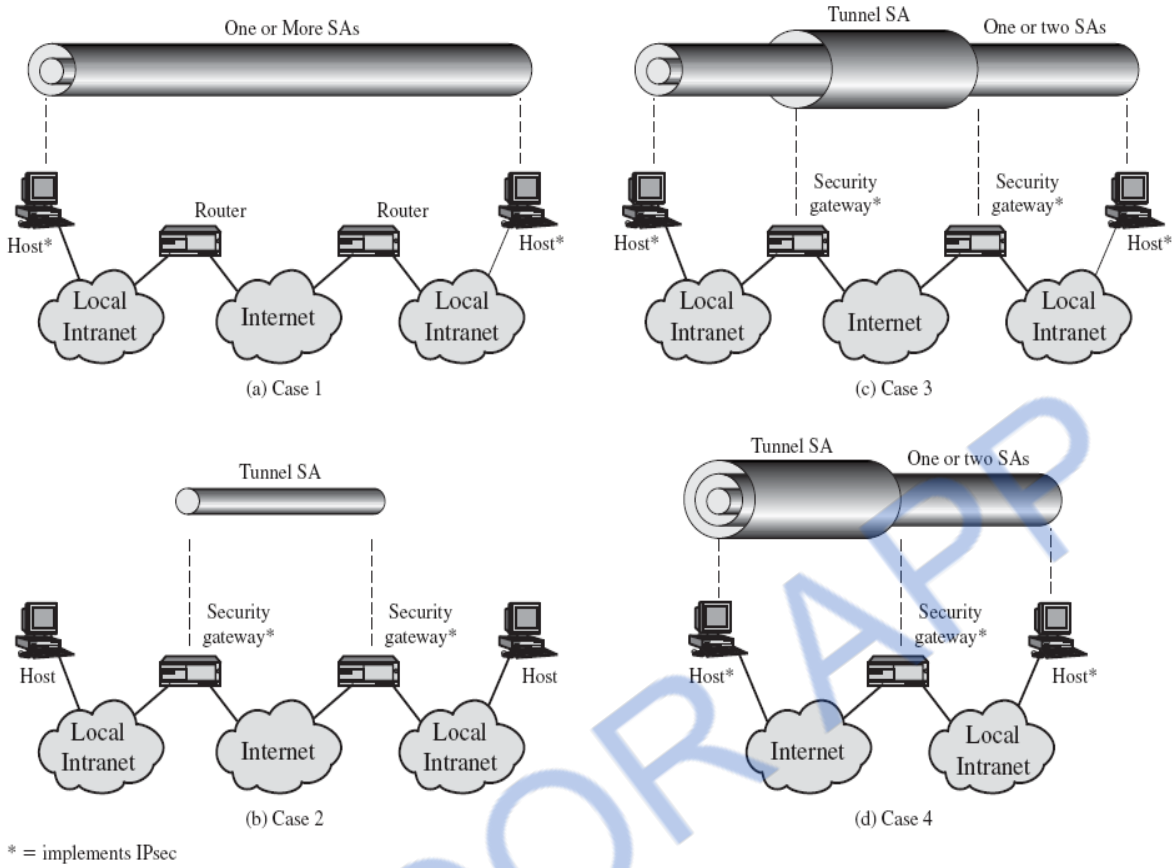


Fig 5.12: Basic Combinations of Security Associations

5.2.4.1. Oakley Key Determination Protocol

Oakley is a refinement of the Diffie-Hellman key exchange algorithm. There is prior agreement on two global parameters: q, a large prime number; and a primitive root of q. A selects a random integer X_A as its private key, and transmits to B its public key $Y_A = \square^{XA}$ mod q. Similarly, B selects a random integer X_B as its private key and transmits to A its public key $Y_B = \square^{XB}$ mod q. Each side can now compute the secret session key:

$K = (Y_B)^{XA} \text{mod } q = (Y_A)^{XB} \text{ mod } q = \quad ^{XAXB} \text{ mod } q$

The Diffie-Hellman algorithm has two attractive features:

- Secret keys are created only when needed. There is no need to store secret keys for a long period of time, exposing them to increased vulnerability.
- The exchange requires no preexisting infrastructure other than an agreement on the global parameters.

5.2.4.2. Features of Oakley

The Oakley algorithm is characterized by five important features:

- It employs a mechanism known as cookies to thwart clogging attacks.
- It enables the two parties to negotiate a group; this, in essence, specifies the global parameters of the Diffie-Hellman key exchange.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- It uses nonces to ensure against replay attacks.
- It enables the exchange of Diffie-Hellman public key values.
- It authenticates the Diffie-Hellman exchange to thwart man-in-the-middle attacks.

Three different authentication methods can be used with Oakley:

- Digital signatures
- Public-key encryption
- Symmetric-key encryption

5.2.4.3. ISAKMP

ISAKMP defines procedures and packet formats to establish, negotiate, modify, and delete security associations.

ISAKMP Header Format

An ISAKMP message consists of an ISAKMP header followed by one or more payloads. It consists of the following fields:

- **Initiator Cookie (64 bits):** Cookie of entity that initiated SA establishment, SA notification, or SA deletion.
- **Responder Cookie (64 bits)**: Cookie of responding entity; null in first message from initiator.
- **Next Payload (8 bits):** Indicates the type of the first payload in the message; payloads are discussed in the next subsection.
- **Major Version (4 bits)**: Indicates major version of ISAKMP in use.
- **Minor Version (4 bits):** Indicates minor version in use.
- **Exchange Type (8 bits):** Indicates the type of exchange.
- **Flags (8 bits):** Indicates specific options set for this ISAKMP exchange.
- **Message ID (32 bits):** Unique ID for this message.
- **Length (32 bits):** Length of total message (header plus all payloads) in octets.

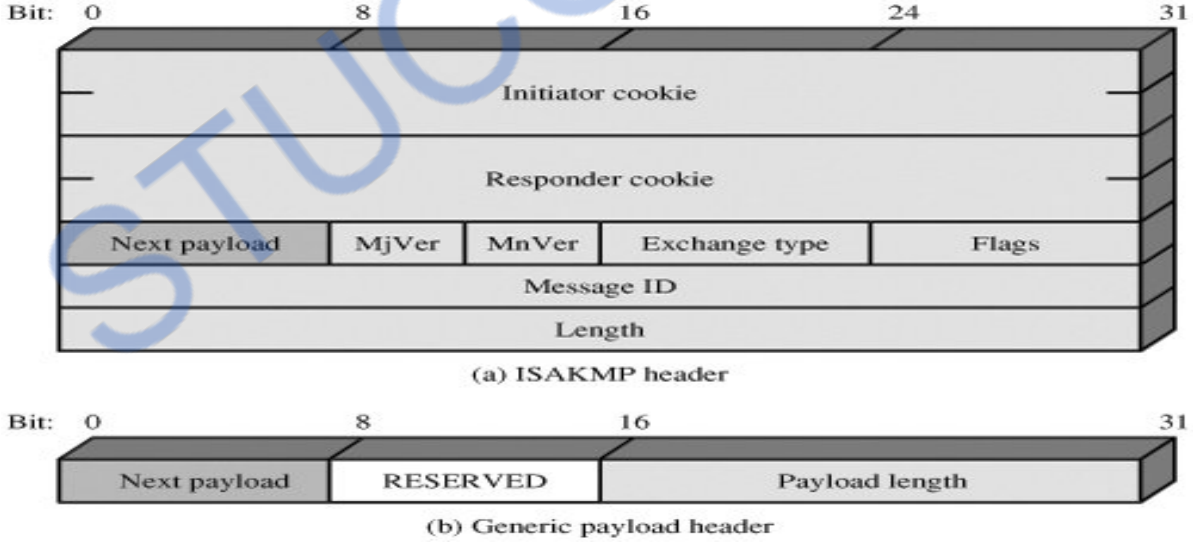


Fig 5.13: ISAKMP Header Format

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

5.3 WEB SECURITY

5.3.1 WEB SECURITY CONSIDERATIONS

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.

Web Security Threats

A Comparison of Threats on the Web

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

Two types of attacks are:

Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.

Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.

5.3.2 WEB TRAFFIC SECURITY APPROACHES

One way to provide Web security is to use IP Security. The advantage of using IPSec is that it is transparent to end users and applications and provides a general-purpose solution.

HTTP	FTP	SMTP
SSL/TLS		
TCP		

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

IP

HTTP	FTP	SMTP
TCP		
IP/IPSec		

(a) Network level

(b) Transport Level

	S/MIME	PGP	SET
Kerberos	SMTP		HTTP
UDP	TCP		
IP			

(c) Application Level

Fig5.14: Relative Facilities in the TCP/IP Protocal Stack

Location of Security

5.3.3 SECURE SOCKET LAYER AND TRANSPORT LAYER SECURITY

5.3.3.1 SSL Architecture

SSL is designed to make use of TCP to provide a reliable end-to-end secure service.

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

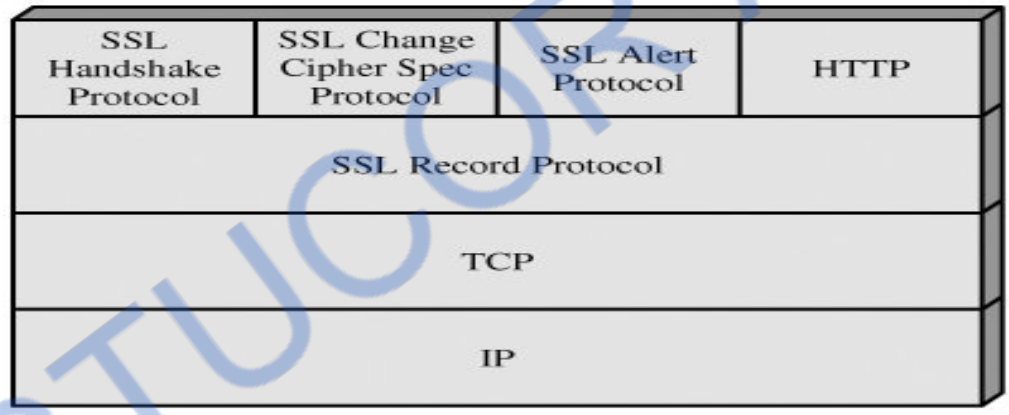


Fig 5.15: SSL Protocol Stack

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

Connection:

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

Session:

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

A session state is defined by the following parameters

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
- Master secret
- Is resumable

A connection state is defined by the following parameters:

- Server and client random
- Server write MAC secret
- Client write MAC secret
- Server write key
- Client write key.
- Initialization vectors
- Sequence numbers

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections:

Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.

Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

The diagram indicates the overall operation of the SSL Record Protocol. The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled and then delivered to higher-level users.

The first step is fragmentation. Each upper-layer message is fragmented into blocks of 2^{14} bytes (16384 bytes) or less. Next, compression is optionally applied. Compression must be lossless and may not increase the content length by more than 1024 bytes. In SSLv3 (as well as the current version of TLS), no compression algorithm is specified, so the default compression algorithm is null.

The next step in processing is to compute a **message authentication code** over the compressed data.

The final step of SSL Record Protocol processing is to prepend a header, consisting of the following fields:

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

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

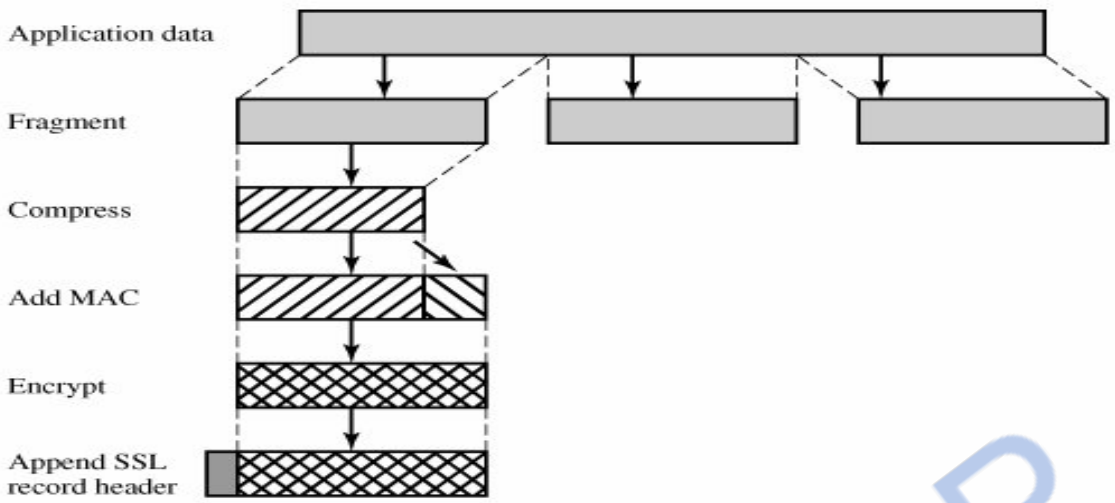


Fig 5.16: SSL Record Protocol Operation

Change Cipher Spec Protocol

This protocol consists of a single message which consists of a single byte with the value 1.

Alert Protocol

The Alert Protocol is used to convey SSL-related alerts to the peer entity.

Each message in this protocol consists of two bytes The first byte takes the value warning(1) or fatal(2) to convey the severity of the message. The second byte contains a code that indicates the specific alert.

- **unexpected_message:** An inappropriate message was received.
- **bad_record_mac:** An incorrect MAC was received.
- **decompression_failure:** The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).
- **handshake_failure:** Sender was unable to negotiate an acceptable set of security parameters given the options available.
- **illegal_parameter:** A field in a handshake message was out of range or inconsistent with other fields.

  The remainder of the alerts is the following:
  - **Close notify:** Notifies the recipient that the sender will not send any more messages on this connection. Each party is required to send a close_notify alert before closing the right side of a connection.
  - **No certificate:** May be sent in response to a certificate request if no appropriate certificate is available.
  - **bad_certificate:** A received certificate was corrupt (e.g., contained a signature that did not verify).
  - **unsupported_certificate:** The type of the received certificate is not supported.
  - **certificate_revoked:** A certificate has been revoked by its signer.
  - **certificate_expired:** A certificate has expired.
  - **certificate_unknown:** Some other unspecified issue arose in processing the certificate, rendering it unacceptable.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Handshake Protocol

This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted.

The Handshake Protocol consists of a series of messages exchanged by client and server.

Establish Security Capabilities

This phase is used to initiate a logical connection and to establish the security capabilities that will be associated with it. The exchange is initiated by the client, which sends a client_hello message with the following parameters:

1. Version: The highest SSL version understood by the client.

2. Random: A client-generated random structure, consisting of a 32-bit timestamp and 28 bytes generated by a secure random number generator. These values serve as nonces and are used during key exchange to prevent replay attacks.

3. Session ID: A variable-length session identifier. A nonzero value indicates that the client wishes to update the parameters of an existing connection or create a new connection on this session.

4. CipherSuite: This is a list that contains the combinations of cryptographic algorithms supported by the client, in decreasing order of preference.

5. **Compression Method:** This is a list of the compression methods the client supports. After sending the client_hello message, the client waits for the server_hello message, which contains the same parameters as the client_hello message.

Server Authentication and Key Exchange

The server begins this phase by sending its certificate; The certificate message is required for any agreed-on key exchange method except anonymous Diffie-Hellman.

Next, a server_key_exchange message may be sent if it is required. The certificate request message includes two parameters: certificate_type and certificate_authorities.

Client Authentication and Key Exchange

If the server has requested a certificate, the client begins this phase by sending a certificate message.Next is the client_key_exchange message, which must be sent in this phase.

Finally, in this phase, the client may send a certificate_verify message to provide explicit verification of a client certificate.

SSL Handshake Protocol Message Types

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

SSL Handshake Protocol Message Types	
Message Type	**Parameters**
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

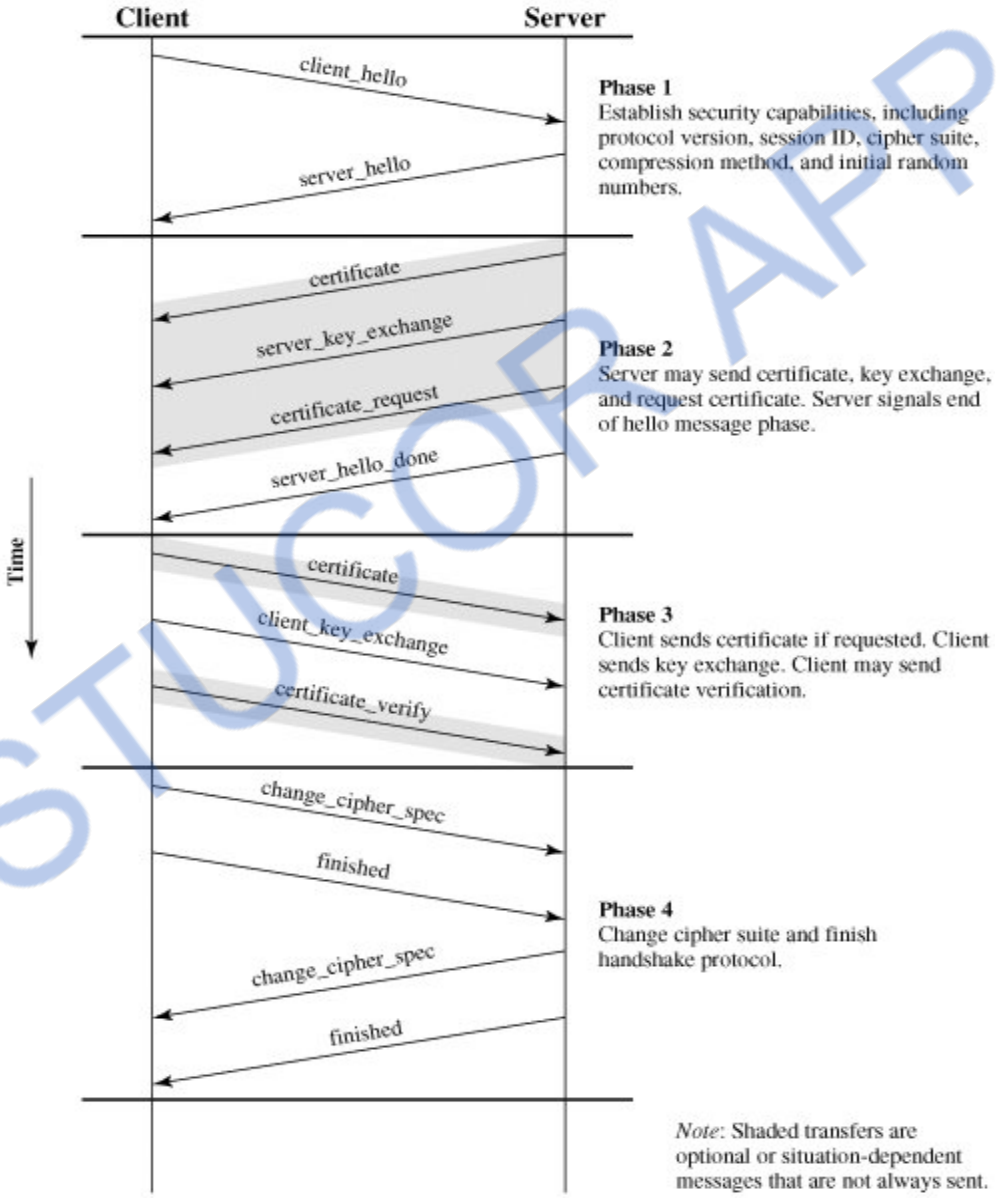


Fig 5.17: Handshake Protocol Action

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Finish

This phase completes the setting up of a secure connection. The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec.

The client then immediately sends the finished message under the new algorithms, keys, and secrets.

Cryptographic Computations

Master Secret Creation

The shared master secret is a one-time 48-byte value (384 bits) generated for this session by means of secure key exchange. The creation is in two stages.

First, a pre_master_secret is exchanged. Second, the master_secret is calculated by both parties. For pre_master_secret exchange, there are two possibilities:

- **RSA:** A 48-byte pre_master_secret is generated by the client, encrypted with the server's public RSA key, and sent to the server. The server decrypts the ciphertext using its private key to recover the pre_master_secret.

- **Diffie-Hellman:** Both client and server generate a Diffie-Hellman public key. After these are exchanged, each side performs the Diffie-Hellman calculation to create the shared pre_master_secret.

5.3.4 SECURE ELECTRONIC TRANSACTION

SET is an open encryption and security specification designed to protect credit card transactions on the Internet.

SET is not itself a payment system. Rather it is a set of security protocols and formats that enables users to employ the existing credit card payment infrastructure on an open network, such as the Internet, in a secure fashion.

SET provides three services:

- Provides a secure communications channel among all parties involved in a transaction
- Provides trust by the use of X.509v3 digital certificates
- Ensures privacy because the information is only available to parties in a transaction when and where necessary

5.3.4.1 Key Features of SET

- Confidentiality of information
- Integrity of data
- Cardholder account authentication
- Merchant authentication

5.3.4.2 SET Participants

- **Cardholder:** A cardholder is an authorized holder of a payment card (e.g., MasterCard, Visa) that has been issued by an issuer.
- **Merchant:** A merchant is a person or organization that has goods or services to sell to the cardholder.
- **Issuer**: This is a financial institution, such as a bank, that provides the cardholder with the payment card.
- **Acquirer**: This is a financial institution that establishes an account with a merchant and processes payment card authorizations and payments.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- **Payment gateway**: This is a function operated by the acquirer or a designated third party that processes merchant payment messages.
- **Certification authority (CA):** This is an entity that is trusted to issue X.509v3 public-key certificates for cardholders, merchants, and payment gateways.

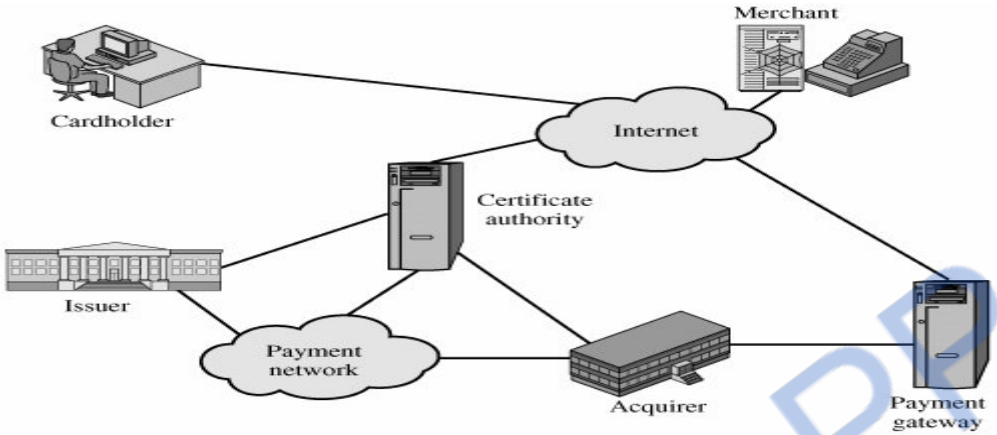


Fig 5.18: Secure Electronic Commerce Components

5.3.4.3 SET Transaction

- **Customer opens account:** The customer obtains a credit card account, such as MasterCard or Visa, with a bank that supports electronic payment and SET.
- **Customer receives a certificate:** After verification the customer receives X.509V3, digital certificate which is signed by the bank. This certificate verifies the customer's RSA public key and expiration date.
- **Merchants have their own certificates:**
  - ✓ Merchants who accepts card need to have 2 certificates for 2 public keys owned by them.
  - ✓ One certificate is used for signing of message and the other is used for key exchange.
  - ✓ The merchants also need the copy of payment gateway's public key certificate.
- **Customer places an order:**
  - ✓ The customer places the order containing the list of items to be purchased to the merchant.
  - ✓ The merchant returns the order form having the items, price, total price and order number.
- **Merchant is verified:** The merchant along with the order form sends its certificate copy. The customer can verify the same.
- **Order and payment are sent:**
  - ✓ The customer sends order and payment information into the merchant along with customer's certificate.
  - ✓ This is order conformation of the order form.
  - ✓ The payment contains the card details. This is encrypted, so it cannot be read by the merchant.
  - ✓ The certificate sent can be verified by the merchant.

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- **Merchant requests payment authorization:** The merchant sends the payment information to the payment gateway. The merchant requests for authentication of the customer, credit limit, validity.
- **Merchant confirms order:** The merchant sends conformation of the order to the customer.
- **Merchant provides goods or service**
- **Merchant requests payment**

5.3.4.4 Dual Signature

The purpose of the dual signature is to link two messages that are intended for two different recipients. In this case, the customer wants to send the order information (OI) to the merchant and the payment information (PI) to the bank. The merchant does not need to know the customer's credit card number, and the bank does not need to know the details of the customer's order.

The customer takes the hash (using SHA-1) of the PI and the hash of the OI. These two hashes are then concatenated and the hash of the result is taken. Finally, the customer encrypts the final hash with his or her private signature key, creating the dual signature. The operation can be summarized as

$$DS = E(PR_c, [H (H (PI) || H (O I)])$$

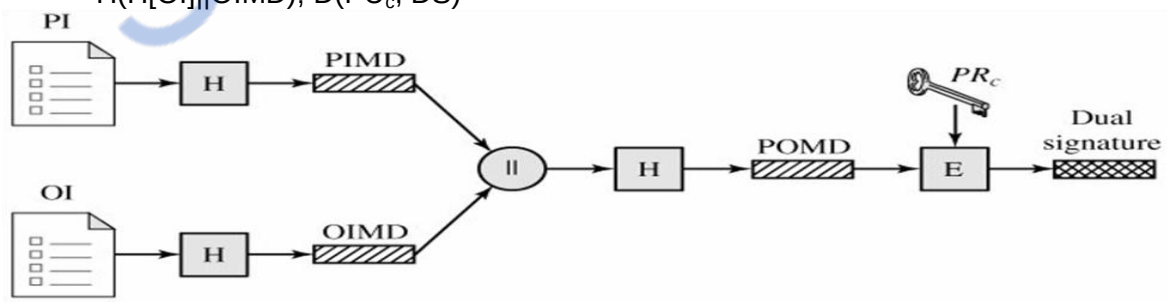
Where PR_c is the customer's private signature key. Now suppose that the merchant is in possession of the dual signature (DS), the OI, and the message digest for the PI (PIMD). The merchant also has the public key of the customer, taken from the customer's certificate. Then the merchant can compute the quantities

$$H (PIMS || H[OI]); D(PU_c, DS)$$

Where PU_c is the customer's public signature key. If these two quantities are equal, then the merchant has verified the signature.

Similarly, if the bank is in possession of DS, PI, the message digest for OI (OIMD), and the customer's public key, then the bank can compute

$$H(H[OI]||OIMD); D(PU_c, DS)$$



PI = Payment information
OI = Order information
H = Hash function(SHA-1)
II = Concatenation
PIMD = PI message digest
OIMD = OI message digest
POMD = Payment order message digest
E = Encryption(RSA)

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

PR_C = Customer's private signature key

Payment Processing

- Purchase request
- Payment authorization
- Payment capture

Purchase Request

Before the Purchase Request exchange begins, the cardholder has completed browsing, selecting, and ordering. The end of this preliminary phase occurs when the merchant sends a completed order form to the customer.

The purchase request exchange consists of four messages: Initiate Request, Initiate Response, Purchase Request, and Purchase Response.

- verifies cardholder certificates using CA sigs
- verifies dual signature using customer's public signature key to ensure order has not been tampered with in transit & that it was signed using cardholder's private signature key
- processes order and forwards the payment information to the payment gateway for authorization (described later)
- sends a purchase response to cardholder

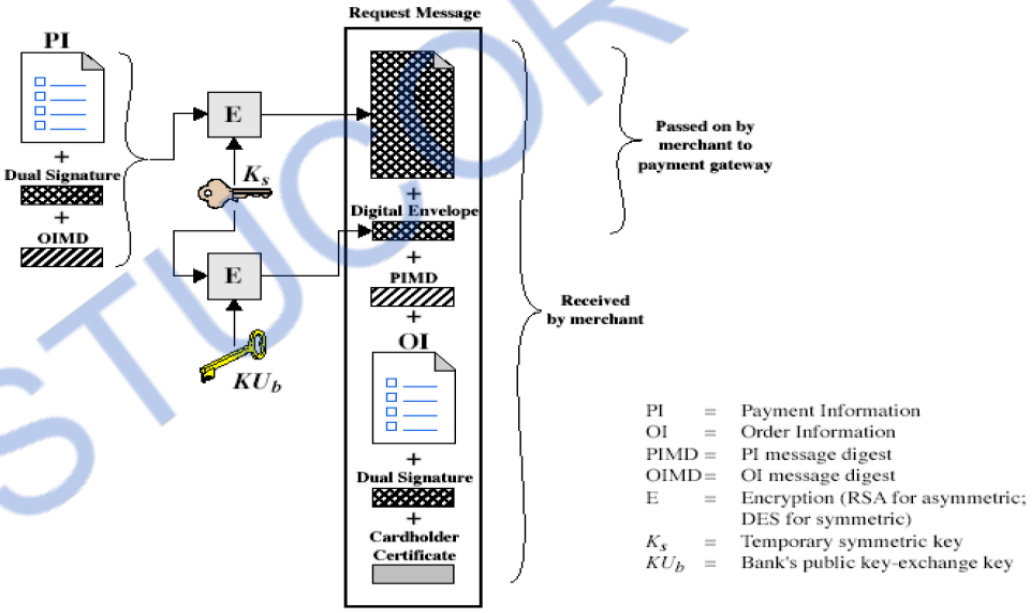


Fig 5.19: Purchase Request – Customer

Payment Authorization

The payment authorization ensures that the transaction was approved by the issuer. This authorization guarantees that the merchant will receive payment; the merchant can therefore provide the services or goods to the customer. The payment authorization exchange consists of two messages: Authorization Request and Authorization response.

- Verifies all certificates

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

- Decrypts digital envelope of authorization block to obtain symmetric key & then decrypts authorization block
- Verifies merchant's signature on authorization block
- Decrypts digital envelope of payment block to obtain symmetric key & then decrypts payment block
- Verifies dual signature on payment block
- Verifies that transaction ID received from merchant matches that in PI received (indirectly) from customer
- Requests & receives an authorization from issuer
- Sends authorization response back to merchant

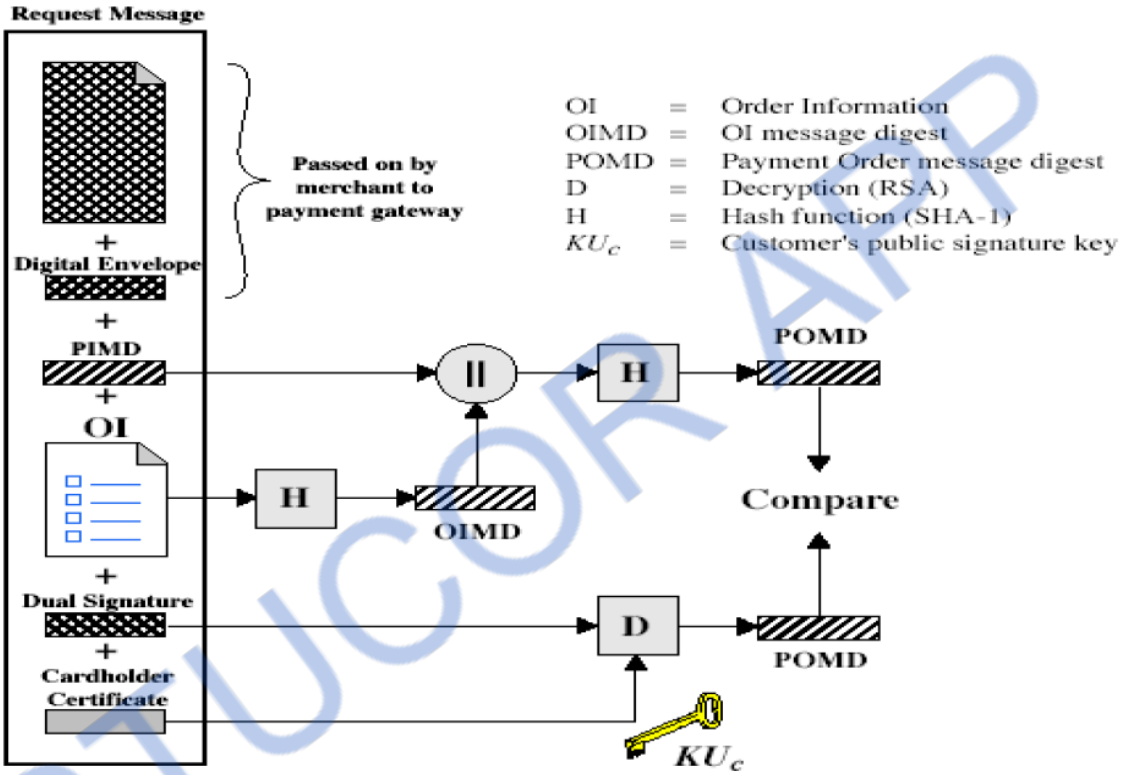


Fig 5.20: Purchase Request – Merchant

Payment Capture

To obtain payment, the merchant engages the payment gateway in a payment capture transaction, consisting of a capture request and a capture response message.

- Merchant sends payment gateway a payment capture request
- Gateway checks request
- Then causes funds to be transferred to merchants account
- Notifies merchant using capture response

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

COLLEGE

SYSTEM SECURITY:

Topic 1: Firewalls

Topic 2: Intruders

Topic 3: Malicious software

Topic 4: Viruses

Refer this Written notes for better Understanding

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Firewalls

Firewall are frequently used to prevent unauthorised Internet users from accessing private networks connected to the Internet,

All messages entering or leaving the intranet pass through the firewall, which examines each message & blocks those do not meet the specified security criteria.

Firewall is a term used for a "barrier" between a network of machines & users that operate under a common security policy & generally trust each other & the outside world.

Related terminology

1- Rules ⟶ AXS GUARD
2- Policies
3. Security level
4. Firewall Rights
5. Firewall GUI : ⟶ a) Create, Edit
   b) Assign Firewall policies
   c) Consult Firewall logs & Firewall status.

Characteristics

1. Traffic from inside to outside & vice versa must pass through the firewall.
2. Only authorized traffic will be allowed to pass
3. Firewall itself is immune to penetration

STUCOR APP

Scanned by CamScanner
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

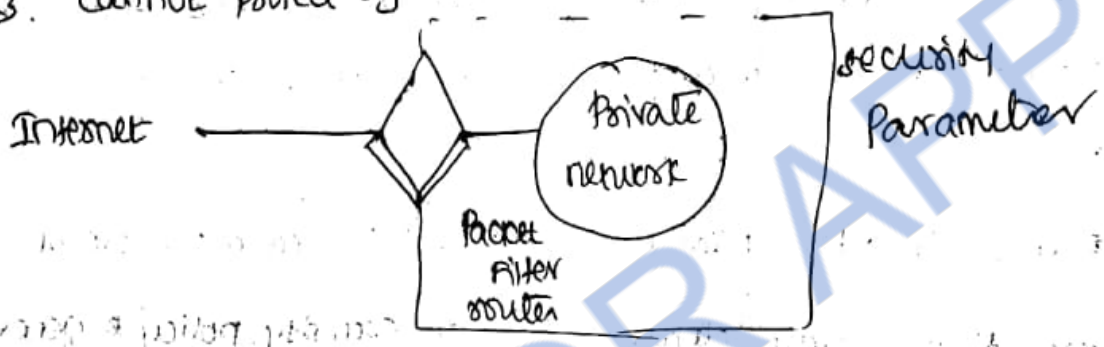
DOWNLOADED FROM STUCOR APP

4 Techniques to control access

a) Service control – Type of Internet service that can be accessed.
b) Direction control – Particular service gets initialized & allowed to flow
c) User control – All the users who have access to the other side of firewall
d) Behaviour control – Particular services are used for an application

Limitations

1. Firewall can be bypassed
2. Does not protect against internet threats
3. Cannot protect against transfer of virus infected programs



Types

1. Packet Filter.
2. Application-level gateway.
3. circuit-level.

Packet [static filtering. (works on network layer].
A packet filter router applies a set of rules of each incoming & outgoing IP packet & then forwards or discards the packet.

Based on
a) Source IP address
b) Destination IP "
c) Source & destination
d) IP protocol field.
e) Interface.

E: A Firewall technique used to control n/w access by monitoring outgoing & incoming packets & allowing them to pass or halt based on the transport level address (source & destination IP addresses, protocols & ports.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

Advantages

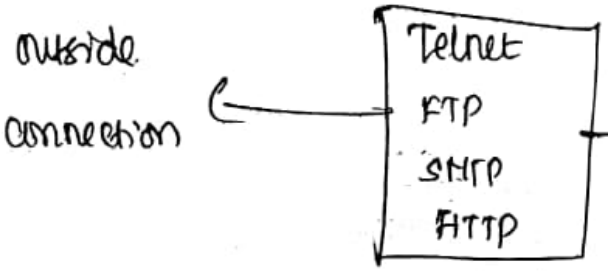
1. simplicity 2. Fast operations 3. Transparent to users

Limitations

1. Firewalls do not examine upper layer data. since limited information is available to firewall its functional is limited.

2. It does not support advanced user authentication schemes.

Application-level



Eg: SMTP application proxies can be configured to allow only certain commands like mailfrom: rcpt to: etc to pass through firewall & block other commands.

* It is also called a proxy server & it acts as a relay of Application level traffic.
* The user contacts the gateway using a TCP/IP application such as Telnet or FTP & gateway asks the user for the name of the remote host to be accessed.
* When userid is provided, the gateway contacts the application on the remote host & relays that tcp segment between two end points.
* gateway not implement proxy code, the service is not supported.

Adv
1. More secure than packet
2. Easy to log & audit all

Disadvantage
1. Additional processing overhead on each Application

3) Circuit-level [works on session layer]

Stand alone system.

NO# End-End TCP connection

Two connection
1) one between itself & an Inner host
2) one between itself & outer host

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

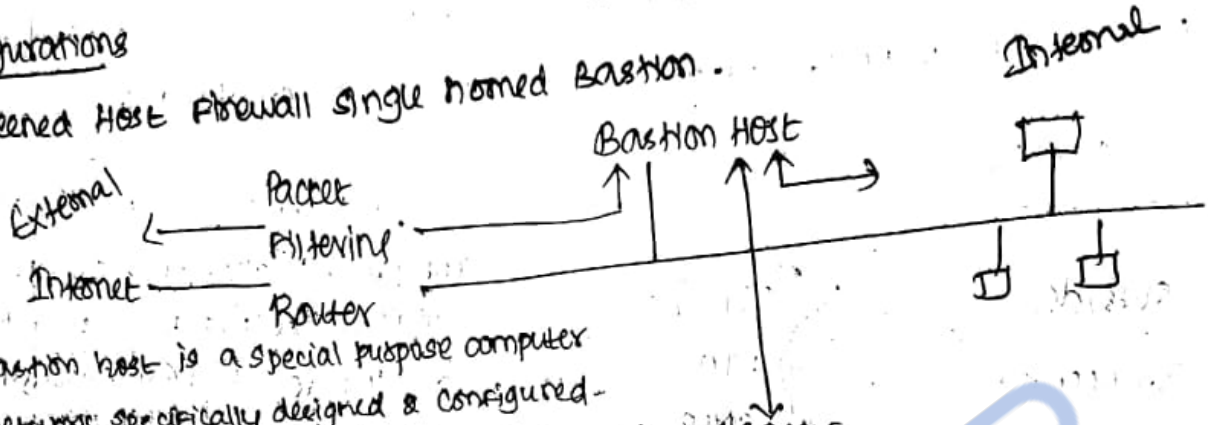
Firewall Designs

Bastion Host

System identified by the firewall administration as a critical strong point in n/w security.

Configurations

1. Screened Host Firewall single homed Bastion.

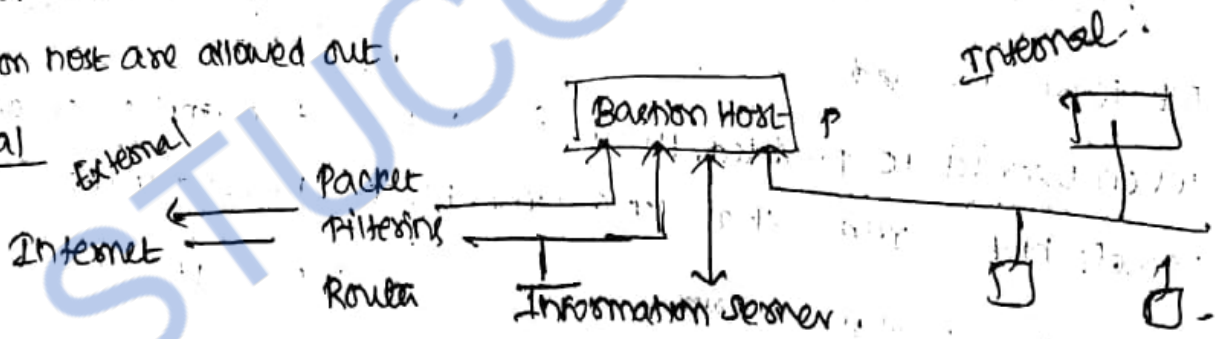


A bastion host is a special purpose computer on a network specifically designed & configured to withstand attacks. The computer generally hosts a single app for eg. a proxy server, and all other services are removed or limited to reduce the threat to computer. The Router is configured so that (External n/w)

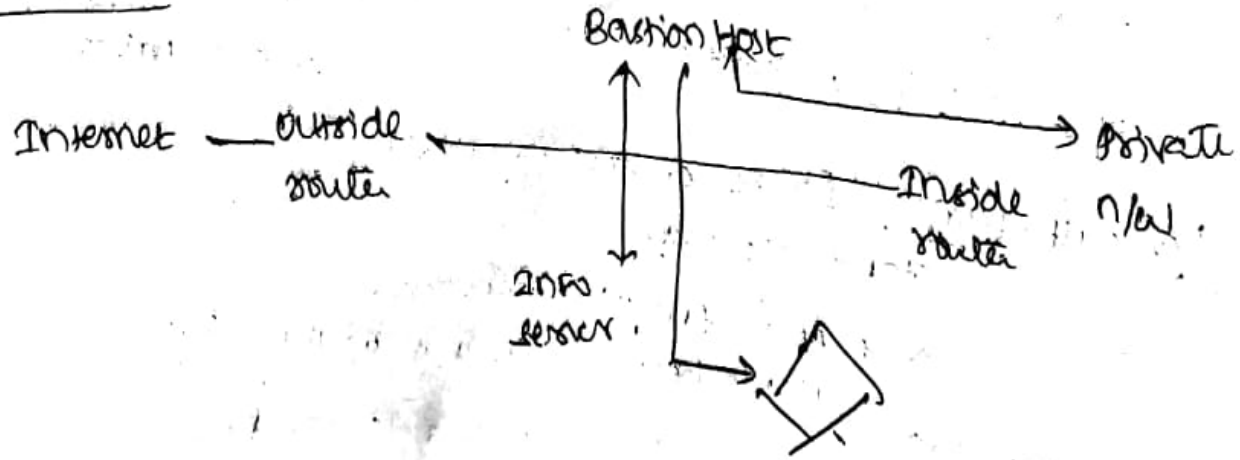
a) for traffic from the Internet, only IP packets destined for the Bastion host are allowed in Internal n/w

b) for traffic from the internal network, only IP packets from the Bastion host are allowed out.

Dual



Screened subnet



STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

1) Two filtering routers

1. one between bastion host & Internet
2. one between bastion host & Internal n/w

2) Both Internet & the internal network have access to the host on the screened subnet but traffic across the screened subnet is blocked.

Ans router → screened subnet to Internet.

Inside → screened subnet to Internal n/w

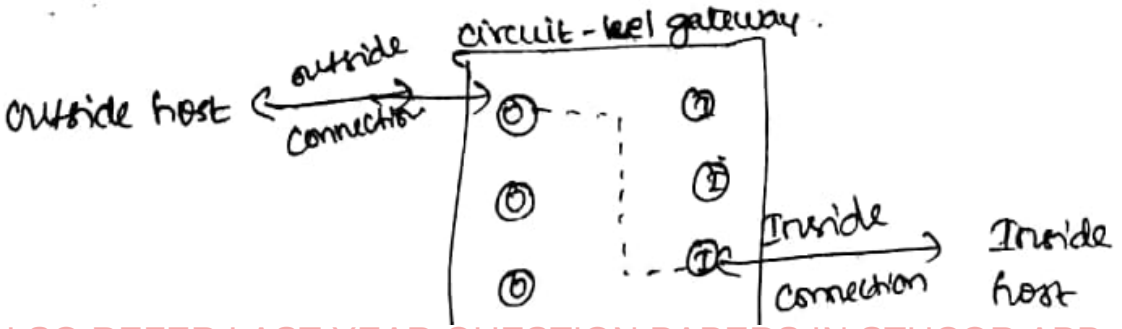
Packet Filters	Application-level gateway	circuit-level gateways.
1. simplest & least secure Firewall mech.	1. Most secure approach	More secure than packet, but not as secure as application
2. Many routers Provide this function.	2. Unique pgm for each application	Relay TCP connection
3. Passes or rejects packets based on rules	3. Good for authentication, logging:	Permission granted to Post address.
4. Hard to manage.	4. Not always transparent to Users.	No application-level checking
	Used for e-mail, FTP, Telnet, WWW	

Application gateways → Application, Presentation, Session

circuit gateways → Transport

Packet Filtering → Network

MAC layer Firewall → Data link, Physical.



STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

Routers

Routers route packets of data from one network to another. Some routers even control the internet's infrastructure.

Gateways

Gateways are the portals that computers use to connect to the internet. One computer can be a gateway for others which is made possible through ICS (Internet connection sharing)

Differences

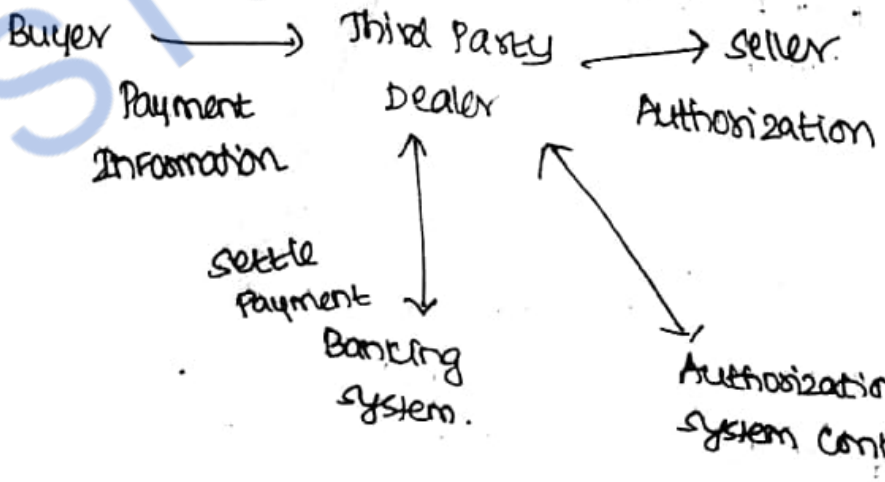
Router coordinates data transfer from one computer to another within a n/w & to other n/w.

Gateway is any device specifically designed to provide all the computer in a n/w with access to WWW.

SET For E-Commerce Transactions

The Secure Electronic Transaction (SET) is the credit card payment protocol that securely transfers the money from customer to the merchant with high integrity.

Structure

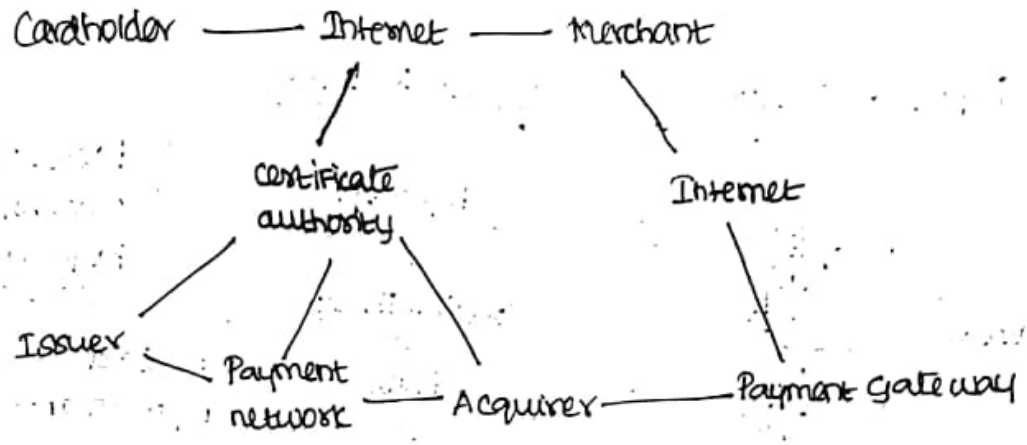


STUCOR APP

Scanned by CamScanner
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

SET Components



Requirements

* Payment Confidentiality
* Ordering Confidentiality
* Transmitted Data Integrity
* Transmitted data privacy
* Card holder Buyer Authentication
* Seller Authentication

Dual signature

Authorizing that particular payment is intended for particular order.

1. Payment Information (PI) for bank
2. Order Information (OI) for merchant

" The purpose of dual signature is to link two messages that are intended for two different recipients.

Customer wants to send the Order Information (OI) to the merchant & the payment Information (PI) to the bank. Merchant does not need to know the customer's credit-card number, and the bank does not need to know the details of the customer's order.

The Dual signature is the encrypted MD (with the customer's secret key) of the concatenated MD's of PI and OI. The dual signature is sent to both the merchant and the bank. The protocol arranges for the merchant to see the MD of the PI without seeing the PI itself, and the bank sees the MD of the OI but not the OI itself. The dual signature can

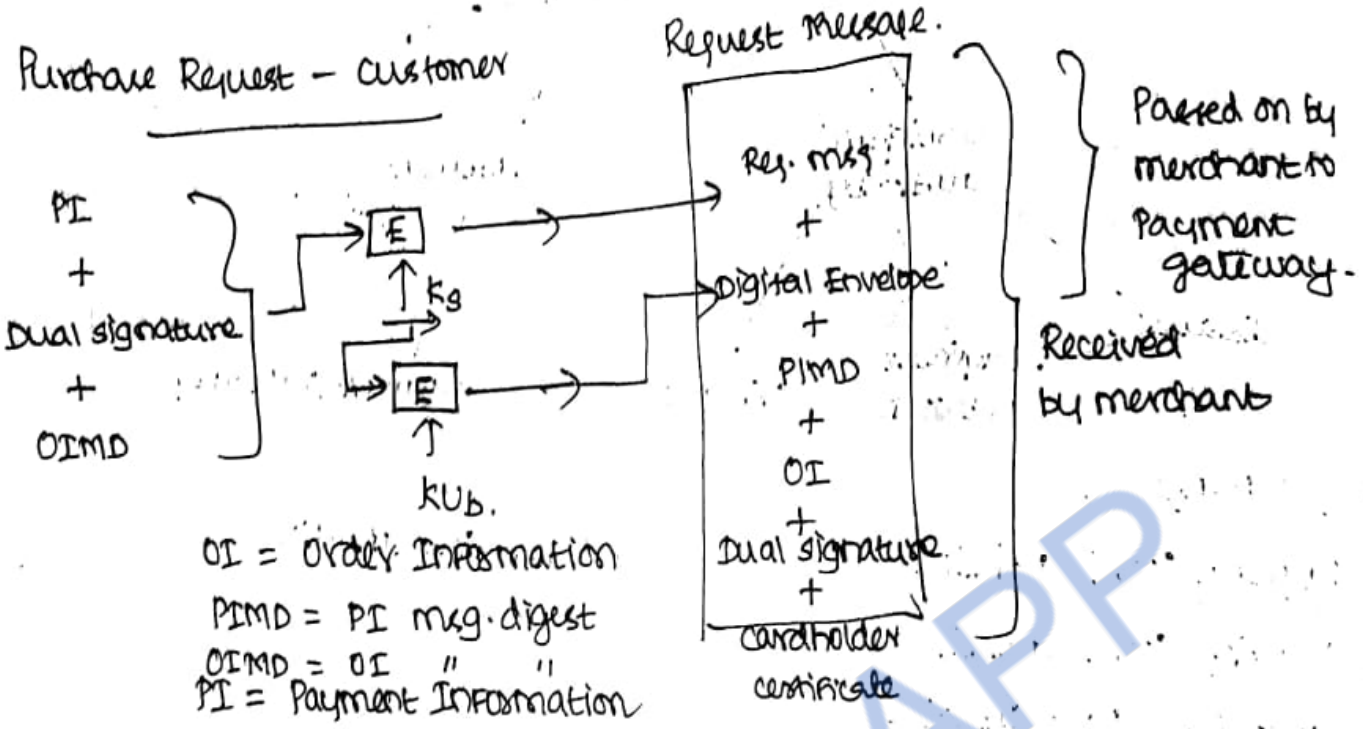
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

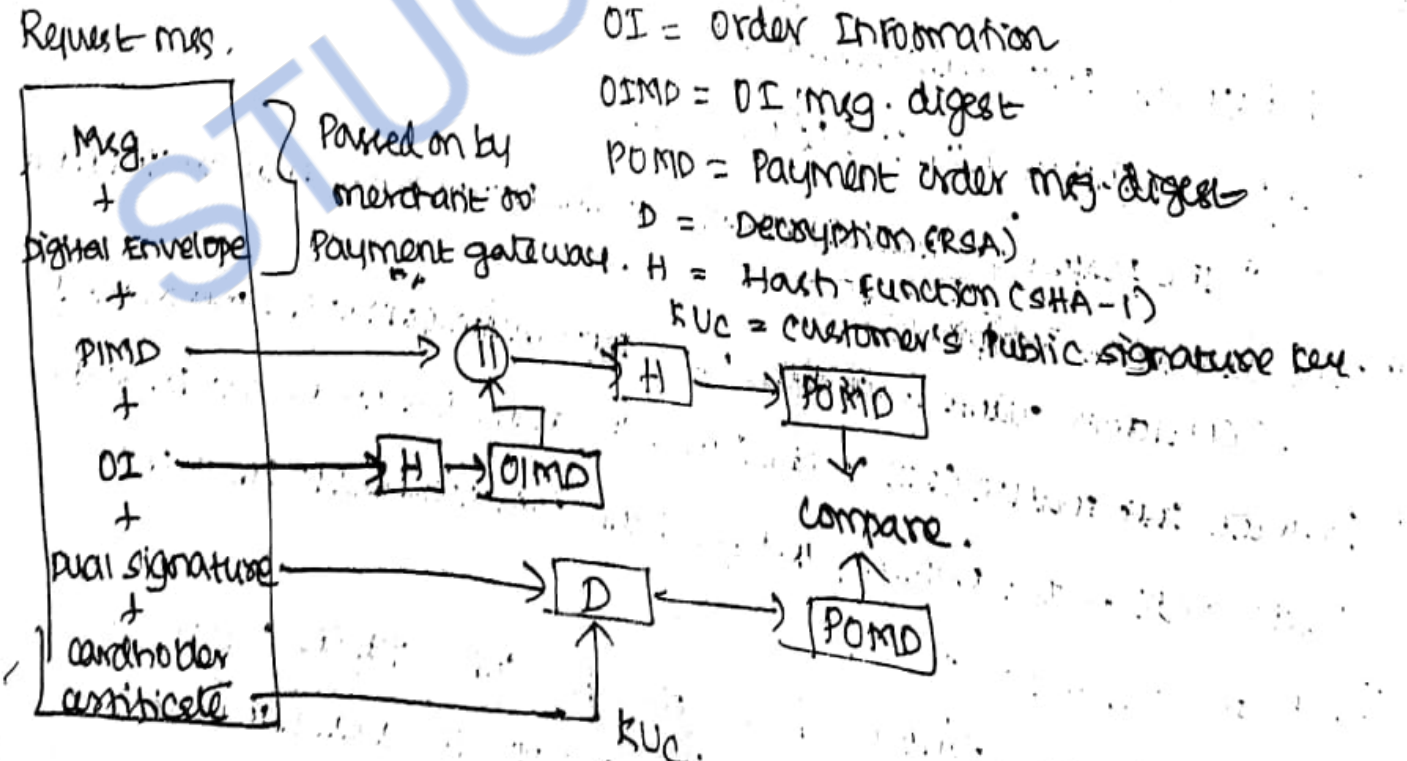
be verified using the MD of the OI or PI. Its MD does not reveal the content of the OI or PI, thus privacy is preserved.

Purchase Request – customer



OI = Order Information
PIMD = PI msg. digest
OIMD = OI " "
PI = Payment Information
E = Encryption (RSA for asymmetric; DES for symmetric).
Ks = Temporary symmetric key.
KUb = Bank's Public key-exchange key.

Purchase Request – Merchant



OI = Order Information
OIMD = OI msg. digest
POMD = Payment order msg. digest
D = Decryption (RSA)
H = Hash function (SHA-1)
KUc = Customer's Public signature key.

STUCOR APP

Scanned by CamScanner

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

1. Verifies cardholder certificates using CA
2. Verifies dual Signature using Customer's Public Signature Key to ensure order has not been tampered with in transit & that it was signed using cardholder's Private Signature key.
3. Processes order & forward the Payment information to the Payment gateway for authorization
4. Sends a Purchase response to Cardholder.

Intrusion Detection

Intruders:

1. Masquerader: Person or a system that is not authorized to use the computer but penetrates a system and access controls to exploit an authorized users account.
2. Misfeasor: An authorized user who misuses the privileges by accessing data, programs or resources for which such access is not authorized.
3. Clandestine User: An individual who seizes supervisory control of the system to evade auditing & access controls or to suppress audit collection.

Techniques

The main aim of any Intruder is to gain access to a system or to increase the range of privileges accessible on a system.

The Intruder needs to know certain details that are protected in the system. This can be done by gaining access to the password file for each authorized user.

Techniques (intrusion)

→ Target acquisition & Information gathering
→ Initial access
→ Privilege escalation
→ Covering tracks

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

Password guessing

→ one of the most common attacks

→ attackers knows a login (from email / web page etc.)

→ try default pswds shipped with systems

→ try all short passwords

→ then try by searching dictionaries of common words

→ Intelligent searches try passwords associated with the user (variations on names, birthday, phone, common words / interfaces)

Password capture

→ watching over shoulder as password is entered

→ using a trojan horse program to collect

→ extracting recorded info after successful login (web history / cache, last number dialed etc)

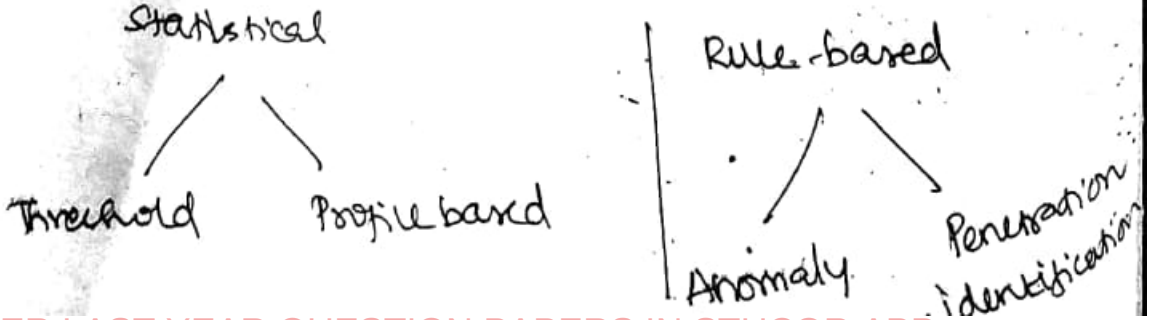
Approaches to Intrusion Detection

1. Statistical Anomaly Detection:

Involves collection of data relating to the behaviour of authorized users over a period of time.

2. Rule based detection

Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.



Statistical
- Threshold
- Profile based

Rule-based
- Anomaly
- Penetration identification

STUCOR APP

Scanned by CamScanner

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

DOWNLOADED FROM STUCOR APP

Statistical

a) Threshold detection

Involves counting the number of occurence of a specific event type over an interval of time.

b) Profile based

Profile of the activity of each user is developed and used to detect the changes in the past behaviour of individual accounts.

Metrics used

1. Counter – keeps a count of certain event types is kept over a particular period of time.
2. Gauge – Used to measure the current value of some entity
3. Interval timer – Notes the length of time between two related events
4. Resource utilization – Quantity of resource that is consumed during a period of time.

Rule Based

a) Rule based Anomaly Detection

→ Historical Audit records are analyzed to identify usage patterns & to generate automatically rules that described patterns.

→ Based on observing past behaviour & assumes that the future will be like the past.

b) Penetration Identific

Rules for identifying known penetration that would exploit known weaknesses.

Rules specific to machine & O.S

Audit Records → Fundamental tool

1. Native Audit record. : Any multiuser os includes accounting s/w that collects the information on user activity
2. Detection specific audit. Facility that can be implemented by generated audit records.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

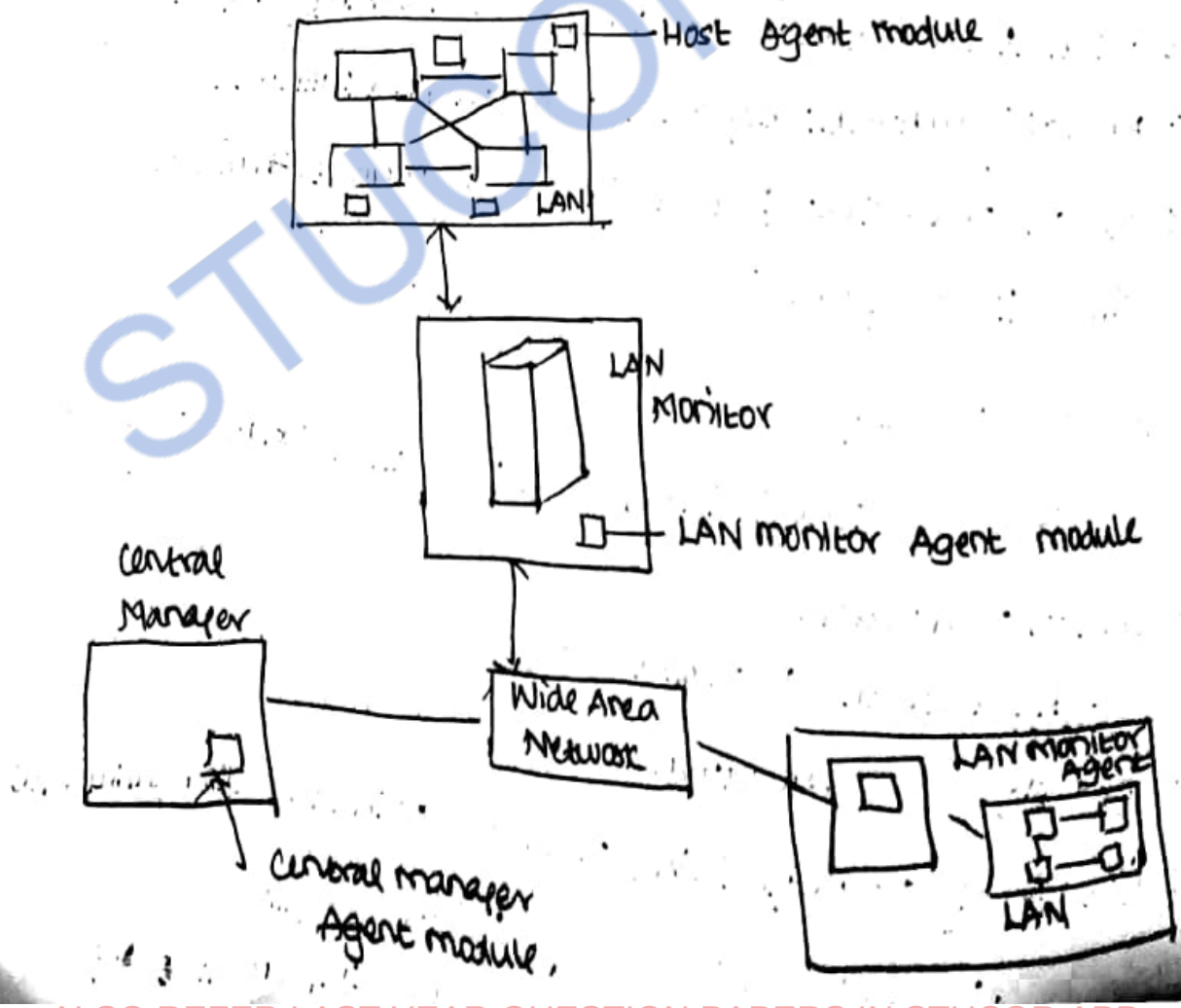
Fields

1. subject → Initiate an action Eg: terminal user, group of users
2. Action → Performed by subject Eg: login, read, execute
3. Object → Entities on which Action is performed Eg: files, Programs, rec
4. Exception condition → Raised
5. Usage → quantitative elements amount list of resources used
6. Time stamp → Action took place.

Shyam	Read	<shyam> Hello.Exe	0	CPU = 0005	1106872165
subject	Action	Object	Exception	Resource	TS

Distributed Intrusion Detection method

Architecture of the Intrusion Detection system for Distributed Network is defined by



STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

Three components

1. Host Agent module 2. LAN monitor Agent module 3. Central manager Agent module.

1. Each host system is configured with the agent module to monitor and collect the audit information.

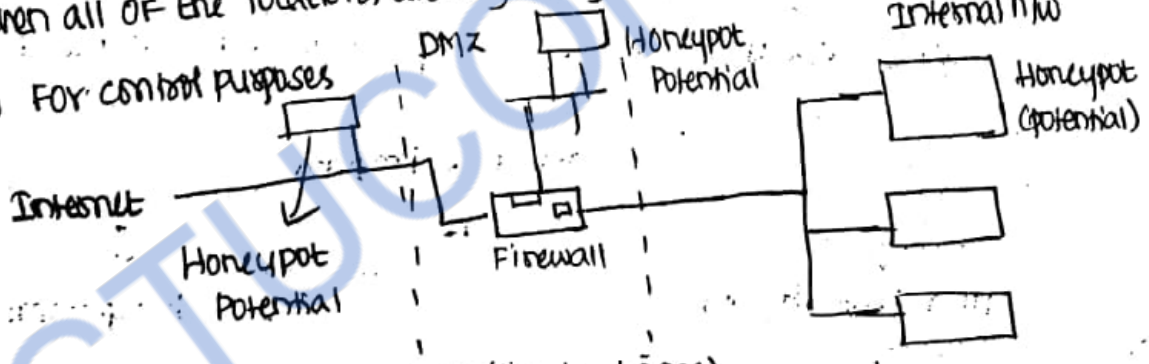
2. Each LAN network has LAN monitor Agent module that analyzes the user agent module & reposts the same to central Manager. ~~Agent module~~

3. The central manager Agent module collects the repost from LAN monitor agent module & compares the reposts with its predefined reports to detect the intrusion.

Honeypots

Another Intrusion Detection system. Honeypot systems are decoy servers or systems setup to gather information regarding an attacker or intruder into your system.

Honeypots can be setup inside, outside or in the DMZ of a Firewall design or even all of the locations although they are most often deployed inside of a Firewall for control purposes



DMZ-(Demilitarized zone)

↓

Add an additional layer of security to an LAN

To Designed, to divert an attacker from accessing critical systems. It collects information about the attackers activity, then it encourages the attackers to say stay on the system long enough for administrator to respond.

STUCOR APP

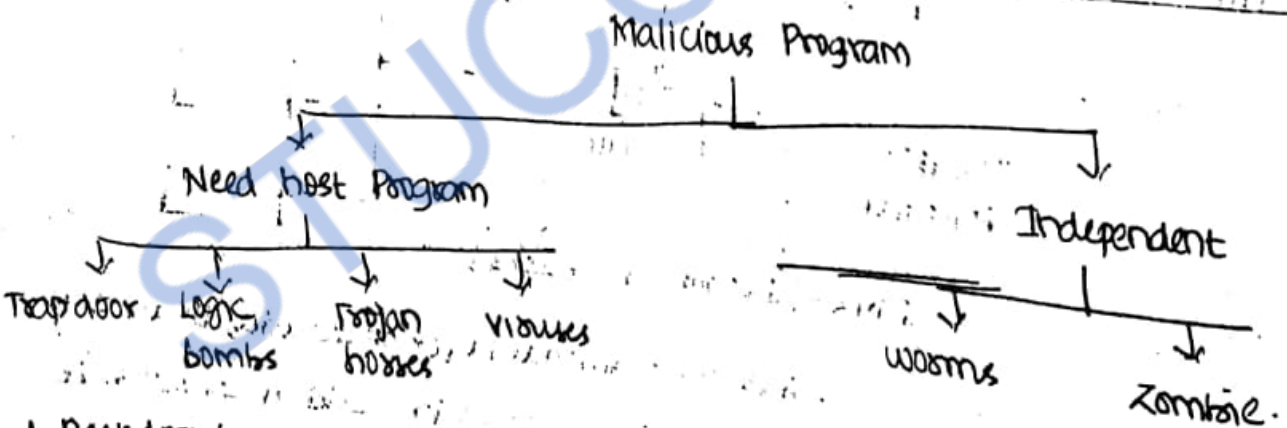
ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

Viruses and Related threats

	Virus	Worms
How does it infect a computer system?	It inserts itself into a file or executable program	It exploits a weakness in an application or OS by replicating itself
How can it spread?	It has to rely on users transferring infected files/programs to other computer systems	It can use a n/w to replicate itself to other computer system without user intervention
Does it infect files?	Yes, it deletes or modifies files. sometimes a virus also change the location of files.	Usually not. Worms usually monopolize the CPU & memory.
Whose speed is more?	Virus is slower than worm	Worm is faster than virus. Eg: code red worm affected 3 lak PC in 14 hrs.
Definition	The virus is the program code that attaches itself to application program & when application pgm run it runs along with it.	The worm is code that replicate itself in order to consume resources to bring it down.



1. **Backdoor:**

Backdoor is also called a trapdoor and it is a secret entry point in the program to gain access with program output. It completely damages the system with unimaginable number of intruders use backdoor to give access to the programs.

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

2. Logic bomb:
Piece of software code that is embedded in the program. Events

i) Destroy some activities in the system
2) Embedded code sends the specified information to attacker.
3) Inform the attacker, that specific user have logged into machine for attack to begin.

3) Trojan horse
Useful Application program to user view contains hidden piece of code that causes damage to system activity.

a) Deletes the user file.
b) can indirectly perform many damages that are not possible even by an unauthorized users.

4) Virus: Piece of software code or package that searches other programs or files & destroys them by inserting a copy of itself into the system.

5) Worms: Copies itself into the new machine & makes the replication.

a) Self carried worms actively transmit itself in the infection process.

b) Second channel worms: Secondary communication channel to complete the infection

c) Embedded worm: Makes itself as a copy in the normal communication channel either by appending or replacing the original message.

6) Zombie: Secretly captures another internet attached system and then uses that machine to embed the attacks that are very difficult to identify. DOS

7) Bacteria: Replicates by affecting the processor, memory & CPU disk.

Characteristics of virus / Nature of virus

Dormant → Propagation → Triggering → Execution

Dormant: Idle state & activated by some event date, capacity of disk

Propagation: Keeps identical copy of itself into another pgm. Every infected pgm will have the virus which is now in the propagation phase.

Triggering: Makes the virus to be activated to perform its desired function for which this virus is designed.

Execution: Virus to be executed in the target pgm. Outcome of virus execution may be data destruction

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

Types of virus / Classification

1. Parasitic virus [Attaches to Executable files & replicates]
2. Memory Resident virus : Lodges in main memory as a part of resident system Pgm.
3. Boot sector virus : Infects a master boot record or spreads a system is booted from dsk
4. Stealth virus : [To Hide from detection by antivirus s/w]
5. Polymorphic virus : mutates with every infection ["signature" different antivirus cannot detect]
6. Email virus : Melissa virus" spreading Email virus made use of a ms word attachment
7. Metamorphic virus : [similar to Polymorphic, gets more power after each infection w. rewrites its own code.
8. Macro virus : Infect Ms word document.

Worms : Remote Execution capability : [worms executes a copy of itself on another system.

Remote login capability : A worm logs onto a remote system as a user & then uses commands to copy itself from one system to another.

Virus Countermeasures

1. Antivirus methods — Virus detection, Virus Identification, Virus Removal or Deletion
2. Techniques for Antivirus.

Generation of Antivirus s/w

1. First generation - Basic virus scanner [Uses virus signature & detect only specific virus
2. Second - Heuristic : Not using signature & uses heuristic rules to find.
3. Third - Activity : Identify not by structure by the action performed.
4. Fourth - Full Featured : Antivirus s/w techniques that are Parallely executed [common capability to restrict reproduce in the system]

Techniques for Antivirus method [To develop antivirus]

1) Generic Decryption method
2) Digital Immune method
3) Behaviour Blocking s/w

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

1) General Decryption Method:

Detect & Fastly scan the Polymorphic virus from the system. Generally the virus must decrypt itself to be activated in the system.

1. CPU Emulator: (Interprets the instruction in executable file rather than on Processor)
2. Virus signature scanner [Scans the code for known virus signature]
3. Emulation control module [Controls execution of virus infected code]

2) Digital Immune method. [Designed for Internet based virus]

Information from virus Infected client m/c

↓

Administrative m/c.

↓

Received by virus Analysis m/c Engine

↓

Analyze the virus Behaviour

↓

Analyze the virus structure.

↓

Extract the virus signature.

↓

Desire virus description to Individual user.

3) Behaviour Blocking method: [Integrates with OS of a computer & monitors for malicious actions.

1) Modifying critical system settings such as startup
2) Operating, view, delete or modify files.
3) Initiating N/w communications.
4. Formatting disk.
5. Modify Exec files.
6. scripting of e mail & msg to client.

Practical Implementation of cryptography

1. Code book: Code replaces word or phrase with character.

∧	%	@	∞	⌫	≠
and	mrg.	Night	Tanks	Bombrun	Tmw.

pt. T => Tanks and bomb run tmw. Night.

cod => ∞∧⌫≠@.

2. Nomenclators: Use elements of substitution cipher.

a	b	c
0	≠	∧

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP

Scanned by CamScanner

DOWNLOADED FROM STUCOR APP

3. Smart cards

Ex: For building entry system, ATM's etc.

4. Biometrics :: Authenticating an individual by personal characteristics.

Token is unique [Replace password-based authentication]

STUCOR APP

ALSO REFER LAST YEAR QUESTION PAPERS IN STUCOR APP