SUBJECT CODE : CCS354

Strictly as per Revised Syllabus of

ANNA UNIVERSITY

Choice Based Credit System (CBCS)

Vertical - 4 (Cyber Security and Data Privacy) (CSE/IT/AI&DS)

NETWORK SECURITY

Vilas S. Bagad

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

Iresh A. Dhotre

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

® TECHNICAL ®
PUBLICATIONS
SINCE 1993
An Up-Thrust for Knowledge

TABLE OF CONTENTS

UNIT I

Chapter - 1 Introduction (1 - 1) to (1 - 60)

UNIT II

UNIT III

UNIT IV

Chapter - 4 Application Layer Security (4 - 1) to (4 - 38)

UNIT V

Chapter - 5 Security Practices (5 - 1) to (5 - 40)

UNIT I

1 Introduction

Syllabus

Basics of cryptography, conventional and public-key cryptography, hash functions, authentication, and digital signatures.

Contents

AU : Dec.-20

1.1 Basics of Cryptography

- The history of information security begins with computer security.
- Network security, to protect networking components, connections and contents.
- Information security to protect the confidentiality, integrity and availability of information assets, whether in storage, processing or transmission.
- Physical security consists of all mechanisms used to ensure that physical access to the computer systems and networks is restricted to only authorize users.
- Data security is the science and study of methods of protecting data from unauthorized disclosure and modification.
- Data and information security is about enabling collaboration while managing risk with an approach that balances availability versus the confidentiality of data.
- Security is required because the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means.
- Network security measures are needed to protect data during their transmission.
- Following are the examples of security violations.
  1. User A transmits a sensitive information file to user B. The unauthorized user C is able to monitor the transmission and capture a copy of the file during its transmission.
  2. A message is sent from a customer to a stockbroker with instructions for various transactions. Subsequently, the investments lose value and the customer denies sending the message.
  3. While transmitting the message between two users, the unauthorised user intercepts the message, alters its contents to add or delete entries and then forwards the message to destination user.

1.1.1 Basic Terminologies in Security

- Basic terminology used for security purposes are as follows :

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

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

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

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

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

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

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

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

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

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

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

1.1.2 Categories

- Various categories of computer security are :

  1. Cryptography 2. Data security

  3. Computer security 4. Network security

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

1.1.3 Techniques

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

1.1.4 Elements of Information Security

- Security goals are as follows :
  1. Confidentially 2. Integrity 3. Availability

1. Confidentiality

- Confidentiality refers to limiting information access and disclosure to authorized users and preventing access by or disclosure to unauthorized ones.
- Sensitive information should be kept secret from individuals who are not authorized to see the information.
- Underpinning the goal of confidentiality are authentication methods like user-IDs and passwords that uniquely identify a data system's users and supporting control methods that limit each identified user's access to the data system's resources.

- Confidentiality is not only applied to storage of data but also applies to the transmission of information.
- Confidentiality means that people cannot read sensitive information, either while it is on a computer or while it is traveling across a network.
- Fig. 1.1.1 Relationship between Confidentiality Integrity and Availability.

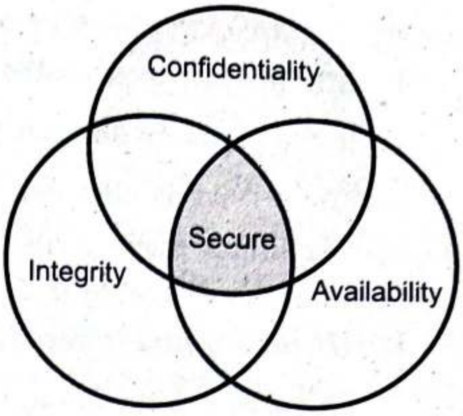


Fig. 1.1.1 Relationship between confidentiality integrity

2. Integrity

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

3. Availability

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

- For example, an object or service is thought to be available if
  - i. It is present in a usable form.
  - ii. It has capacity enough to meet the services needs.
  - iii. The service is completed an acceptable period of time.
- By combining these goals, we can construct the availability. The data item, service or system is available if
  - i. There is a timely response to our request.
  - ii. The service and system can be used easily.
  - iii. Concurrency is controlled.
  - iv. It follows the fault tolerance.
  - v. Resources are allocated fairly.

1.1.5 Threats and Vulnerability

Threat

- The term "threat" refers to the source and means of a particular type of attack.
- A threat assessment is performed to determine the best approaches to securing a system against a particular threat or class of threat.
- Penetration testing exercises are substantially focused on assessing threat profiles, to help one develop effective countermeasures against the types of attacks represented by a given threat. Where risk assessments focus more on analyzing the potential and tendency of one's resources to fall prey to various attacks, threat assessments focus more on analyzing the attacker's resources.
- Analyzing threats can help one develop specific security policies to implement in line with policy priorities and understand the specific implementation needs for securing one's resources.

Vulnerability

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

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

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

1.1.6 Cryptography

- Cryptography is the science of writing in secret code and is an ancient art. Cryptography is not only protects data from theft or alteration, but can also be used for user authentication.
- The term is derived from the Greek word kryptos, which means hidden.
- In cryptography, we start with the unencrypted data, referred to as plaintext. Plaintext is encrypted into ciphertext, which will in turn (usually) be decrypted back into usable plaintext.
- Fig 1.1.2 shows cryptography.

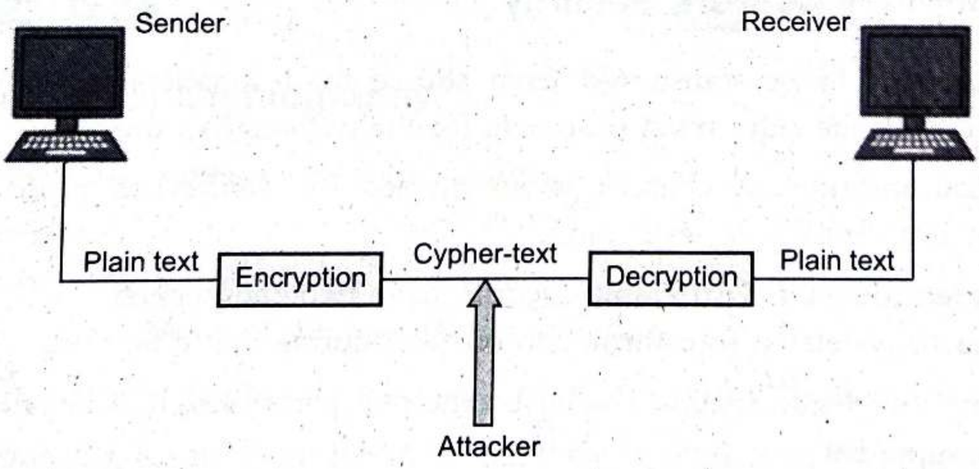


Fig. 1.1.2 Cryptography

- Cryptography provides secure communication in the presence of malicious third parties.

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

Advantages of cryptography

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

Review Question

1. *Discuss examples from real life, where the following security objectives are needed :*

   i) Confidentiality

   ii) Integrity

   iii) Non-repudiation

   Suggest suitable security mechanisms to achieve them. **AU : Dec.-20, Marks 5 + 5 + 5**

1.2 A Model for Network Security

AU : May-19, Dec.-22

- A message is to be transferred from source to destination across some sort of internet. Both the sides must cooperate for the exchange of the data.
- A logical information channel is established by defining a route through the internet from source to destination.
- All the techniques for providing security have two components :
  1. A security related transformation on the information to be sent.
  2. Some secret information shared by the two principles, it is hoped, unknown to the opponent.
- Fig. 1.2.1 shows the network security model.
- A trusted third party is needed to achieve secure transmission.

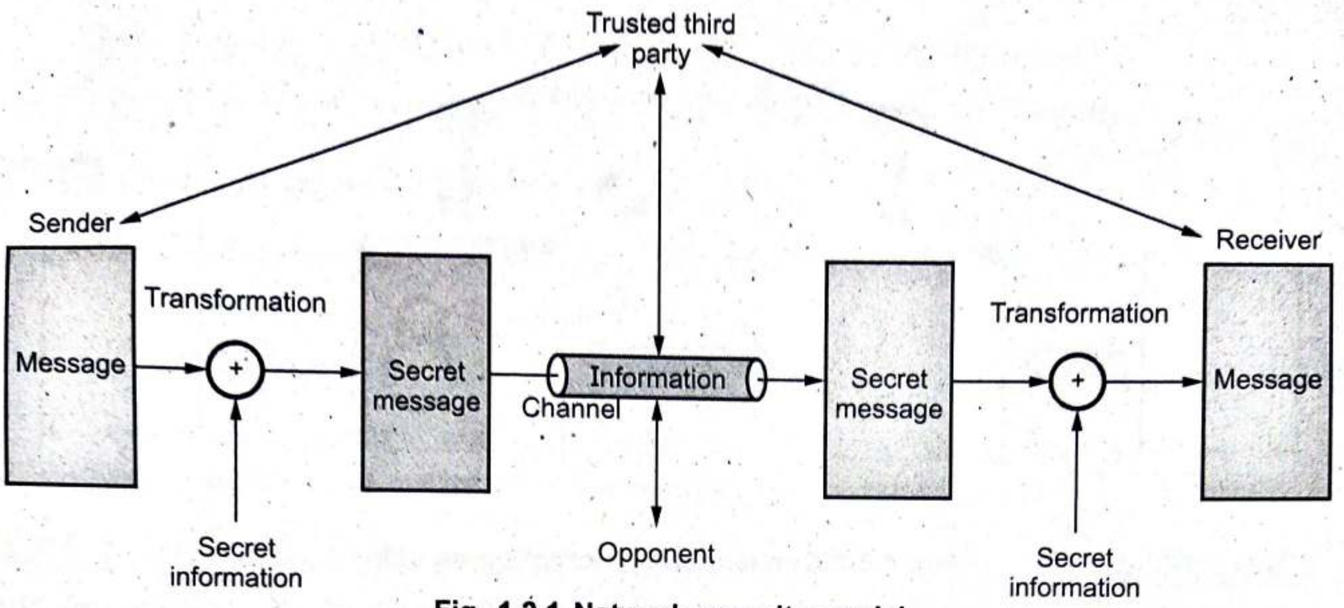


Fig. 1.2.1 Network security model

- Basic tasks in designing a particular security service.
  1. Design an algorithm for performing the security related transformation.
  2. Generate the secret information to be used with the algorithm.
  3. Develop methods for the distribution and sharing of the secret information.
  4. Specify a protocol to be used by the two principles that makes use of the security algorithm and the secret information to achieve a particular security service.

Review Question

1. *Explain the network security model and its important parameters with a neat block diagram.*

AU : May-19, Dec.-22, Marks 13

1.3 Conventional Cryptography

- A symmetric encryption model has five ingredients.

  1. Plaintext 2. Encryption algorithm 3. Secret key

  4. Ciphertext 5. Decryption algorithm
- Fig. 1.3.1 shows the conventional encryption model.
- **Plaintext** is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm** performs various substitutions and transformations on the plaintext.
- **Secret key** is a value independent of the plaintext and of the algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.

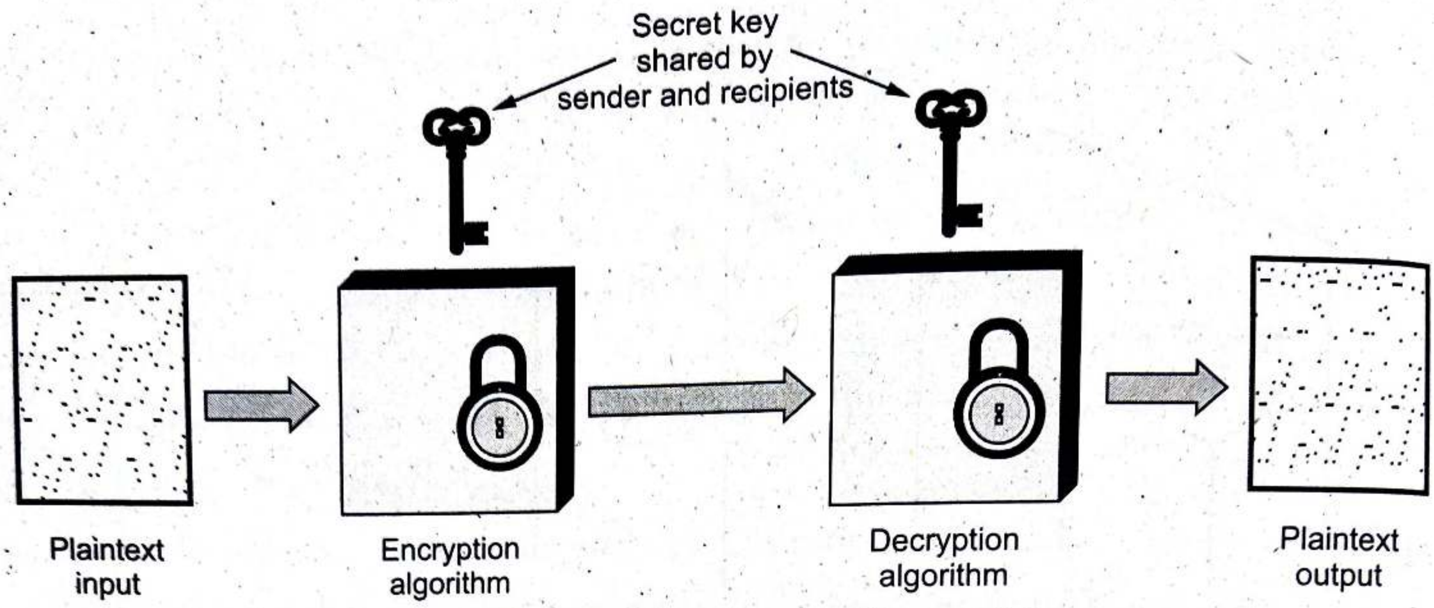


Fig. 1.3.1 Conventional encryption model

- **Ciphertext** is the scrambled message produced as output. It depends on the plaintext and the secret key.
- **Decryption algorithm** takes the ciphertext and the secret key and produces the original plaintext.
- The original intelligible message, referred to as plaintext is converted into random nonsense, referred to as ciphertext. The science and art of manipulating message to make them secure is called **cryptography.**
- An original message to be transformed is called the plaintext and the resulting message after the transformation is called the ciphertext.
- The process of converting the plaintext into ciphertext is called encryption. The reverse process is called decryption. The encryption process consists of an algorithm and a key. The key controls the algorithm.
- The objective is to design an encryption technique so that it would be very difficult or impossible for an unauthorized party to understand the contents of the ciphertext.
- A user can recover the original message only by decrypting the ciphertext using the secret key. Depending upon the secret key used, the algorithm will produce a different output. If the secret key changes, the output of the algorithm also changes.

1.3.1 Advantages of Symmetric Ciphers

1. High rates of data throughput.
2. Keys for symmetric-key ciphers are relatively short.
3. Symmetric-key ciphers can be used as primitives to construct various cryptographic mechanisms (i.e. pseudorandom number generators).

4. Symmetric-key ciphers can be composed to produce stronger ciphers.
5. Symmetric-key encryption is perceived to have an extensive history.

1.3.2 Disadvantages of Symmetric Ciphers

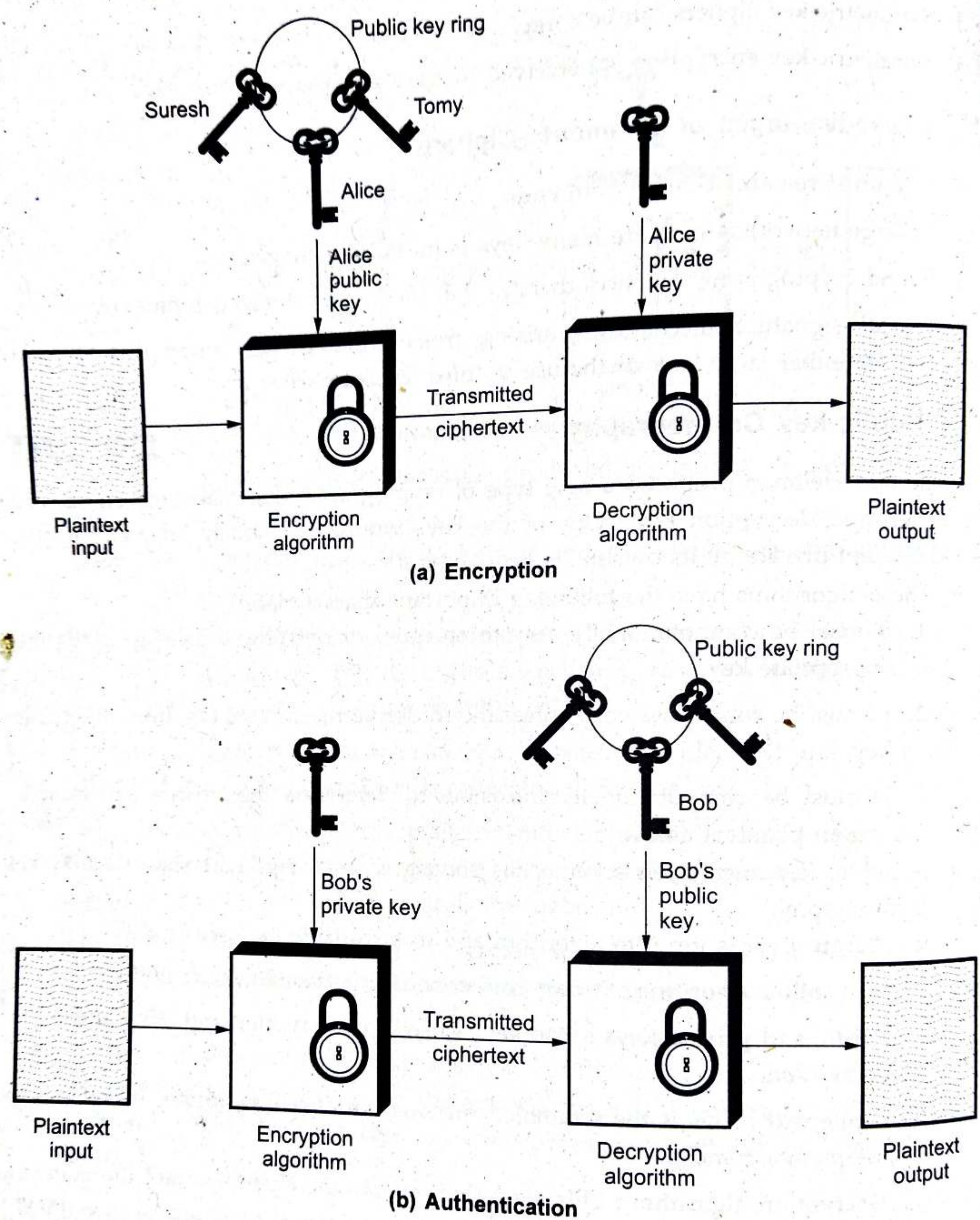
1. Key must remain secret at both ends.
2. In large networks, there are many keys pairs to be managed
3. Sound cryptographic practices dictates that the key be changed frequently
4. Digital signature mechanisms arising from symmetric-key encryption typically require either large keys or the use of third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

1.4 Public-key Cryptography

AU : May-19

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

- These algorithms have the following important **characteristic.**
  1. It must be computationally easy to encipher or decipher a message given the appropriate key.
  2. It must be computationally infeasible to derive the private key from the public key.
  3. It must be computationally infeasible to determine the private key from a chosen plaintext attack.
- A public key encryption scheme has six ingredients. Fig. 1.4.1 shows public key cryptography.
  1. **Plaintext :** It is input to algorithm and in a readable message or data.
  2. **Encryption algorithm :** It performs various transformations on the plaintext.
  3. **Public and private keys :** One key is used for encryption and other is used for decryption.
  4. **Ciphertext :** This is the scrambled message produced as output. It depends on the plaintext and the key.
  5. **Decryption algorithm :** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.
- The essential steps are the following :
  1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
  2. Each user places one of the two keys in a public register. This is the public key. The companion key is kept private.



(a) Encryption

(b) Authentication

Fig. 1.4.1 Public key cryptography

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

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

- System controls its private key. At any time, a system can change its private key. Fig. 1.4.2 shows the process of public key algorithm.

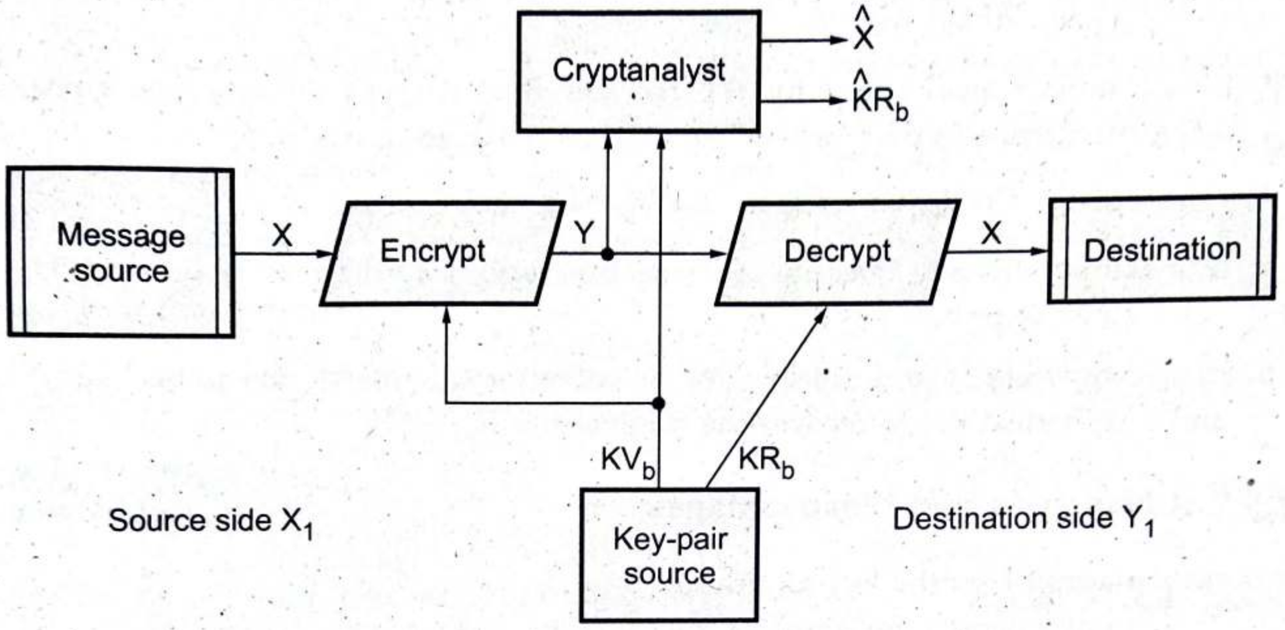


Fig. 1.4.2 Public key cryptosystem secrecy

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

$$Y = (Y_1, Y_2, Y_3 \ldots. Y_n)$$

$$Y = E_{KU_b}(X)$$

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

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

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

Requirements for public key cryptography

1. It is computationally easy for a party B to generate a pair.

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

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

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

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

4. It is computationally infeasible for an adversary, knowing the public key (PU_b) to determine the private key PR_b.

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

1.4.1 Advantages and Disadvantages

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

1.4.2 Comparison between Public Key and Private Key Algorithm

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

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

Review Question

1. *Explain public key cryptography and when it is preferred ?* **AU : May-19, Marks 5**

1.5 Security Attacks

AU : Dec.-13,19

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

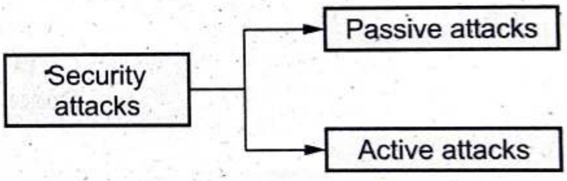


Fig. 1.5.1

1.5.1 Passive Attack

- Passive attacks are those, wherein the attacker indulges in eavesdropping on, or monitoring of data transmission. A passive attack attempts to learn or make use of information from the system but does not affect system resources.
- The attacker aims to obtain information that is in transit. The term passive indicates that the attacker does not attempt to perform any modifications to the data.
- **Passive attacks** are of two types :
  1. Release of message contents 2. Traffic analysis
- **Release of message content** is shown in Fig. 1.5.2. A telephone conversation, an electronic mail message and a transferred file may contain sensitive or confidential information we would like to prevent an opponent from learning the content of these transmissions.

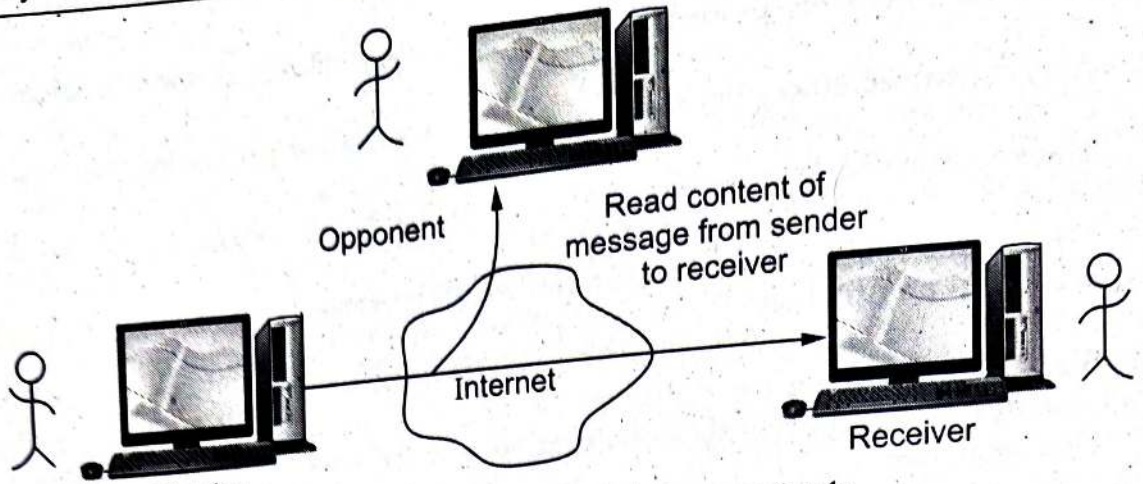


Fig. 1.5.2 Release of message contents

- **Traffic analysis :** Mask the contents of message so that opponents could not extract the information from the message. Encryption is used for masking Fig. 1.5.3 shows the traffic analysis.
- Passive attacks are very difficult to detect because they do not involve any alternation of data. It is feasible to prevent the success of attack, usually by means of encryption.

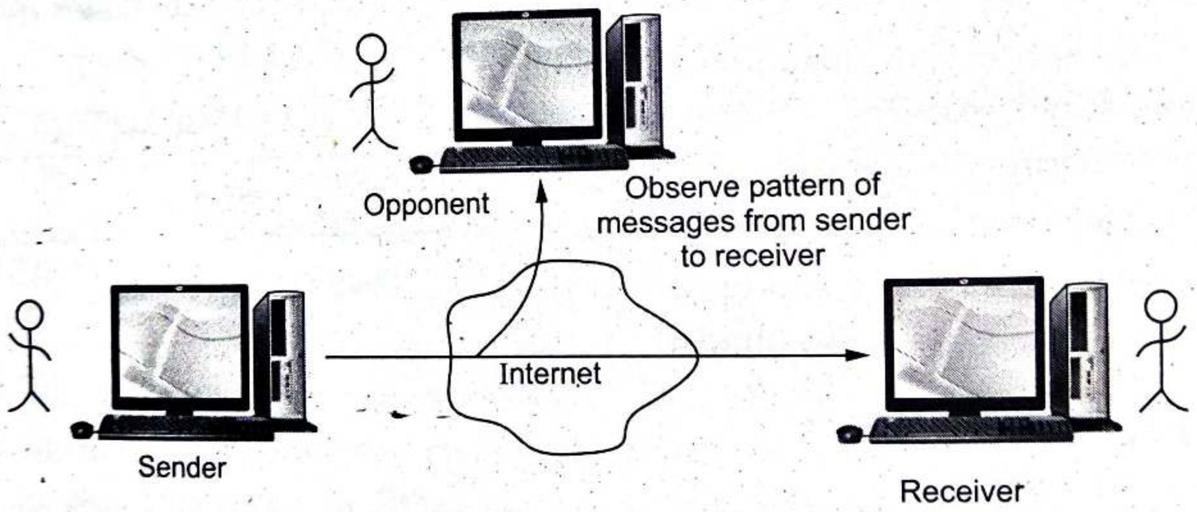


Fig. 1.5.3 Traffic analysis

1.5.2 Active Attack

- Active attacks involve some modification of the data stream or the creation of a false stream. These attacks can not be prevented easily.
- Active attacks can be subdivided into four types :
  1. Masquerade
  2. Replay
  3. Modification of message
  4. Denial of service

1. Masquerade

- It takes place when one entity pretends to be a different entity. Fig. 1.5.4 shows masquerade.

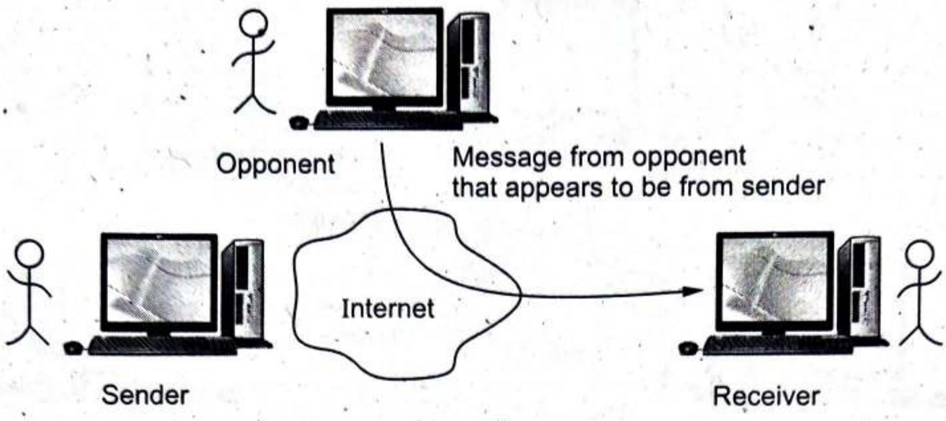


Fig. 1.5.4 Masquerade

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

2. Replay

- It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.
- Fig. 1.5.5 shows replay attack.

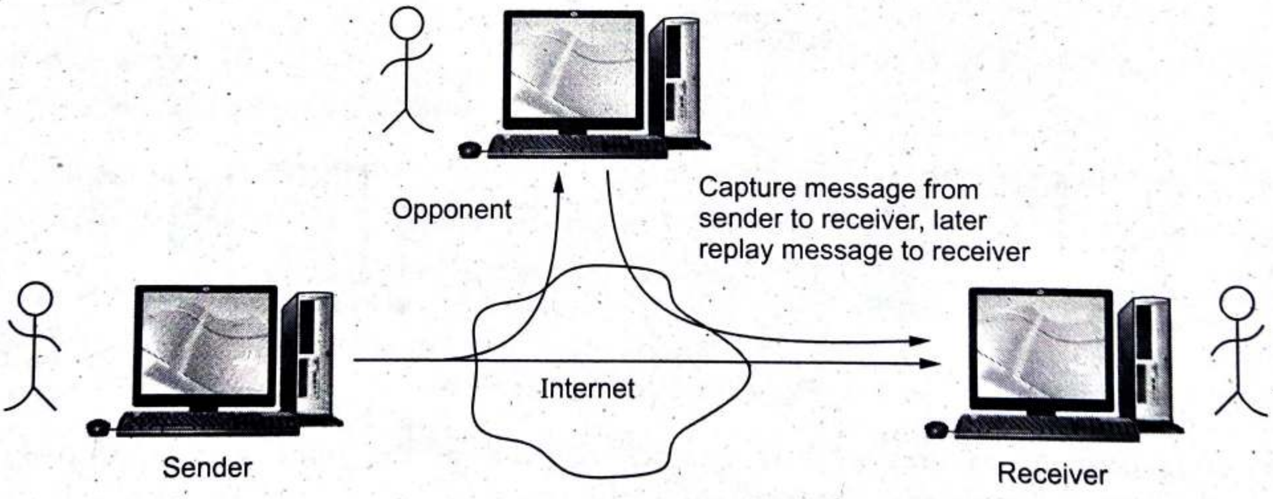

Opponent
Capture message from sender to receiver, later replay message to receiver
Internet
Sender
Receiver

Fig. 1.5.5 Replay

3. Modification of message

- It involves some change to the original message. It produces an unauthorized effect. Fig. 1.5.6 shows the modification of message.
- For example, a message meaning "Allow Rupali Dhotre to read confidential file accounts " is modified to mean "Allow Mahesh Awati to read confidential file accounts".

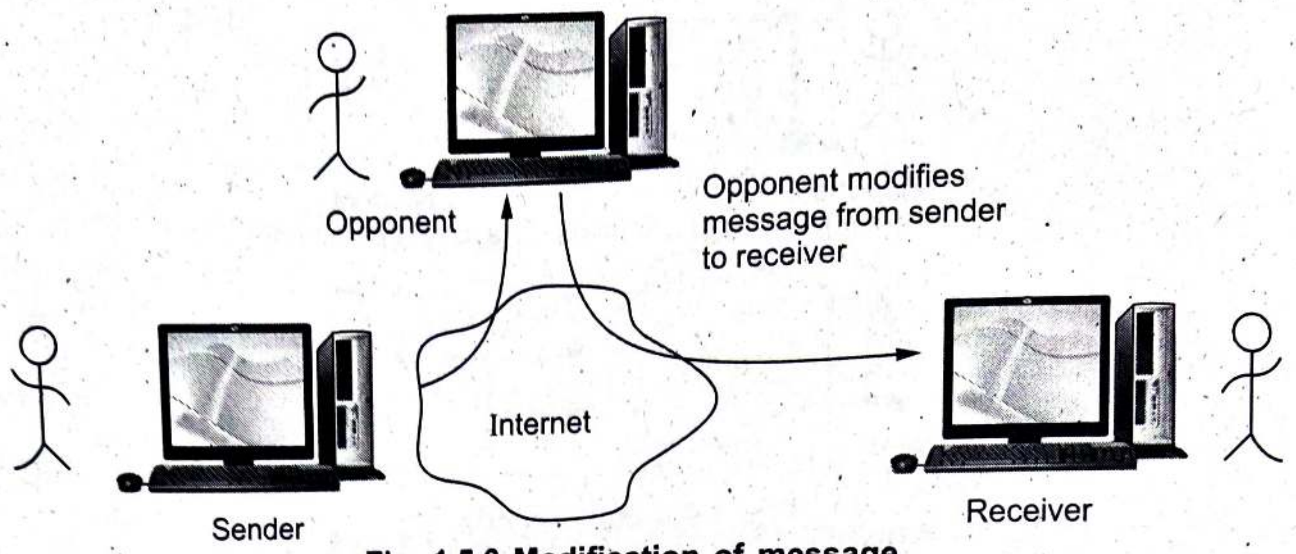


Fig. 1.5.6 Modification of message

4. Denial of service

- Fabrication causes Denial Of Service (DOS) attacks.
- DOS prevents the normal use or management of communications facilities.
- Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.
- Fig. 1.5.7 shows denial of service attack.

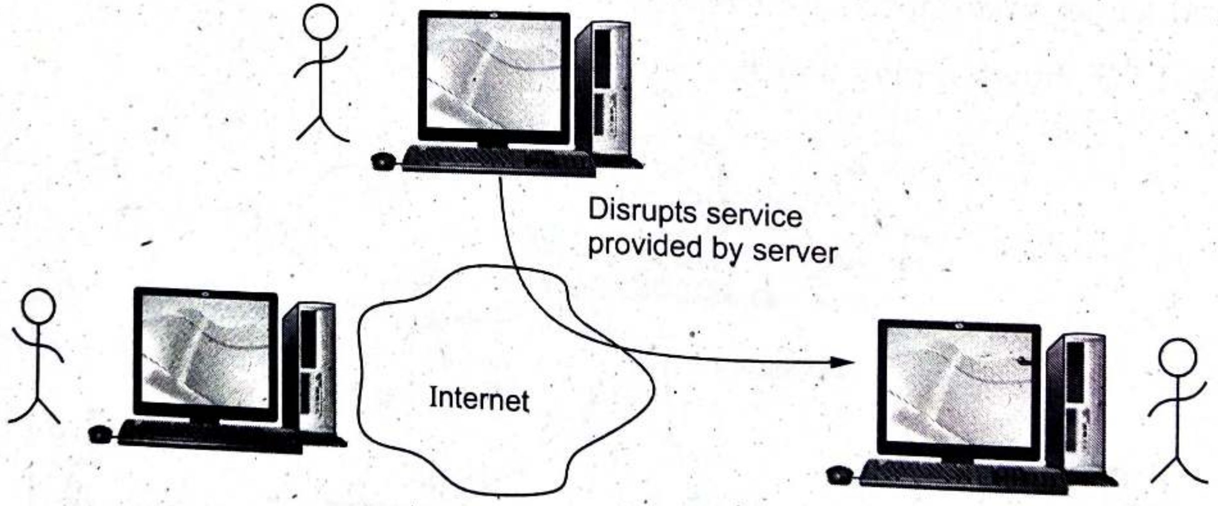


Fig. 1.5.7 Denial of service

- It is difficult to prevent active attack because of the wide variety of potential physical, software and network vulnerabilities.
- The first type of DOS attacks were single source attacks, meaning that a single system was used to attack another system and cause something on that system to fail. SYN flood is the most widely used DOS attack.
- Fig. 1.5.8 shows the SYN flood DOS attack.
- Source system sends a large number of TCP SYN packets to the target system. The SYN packets are used to begin a new TCP connection.

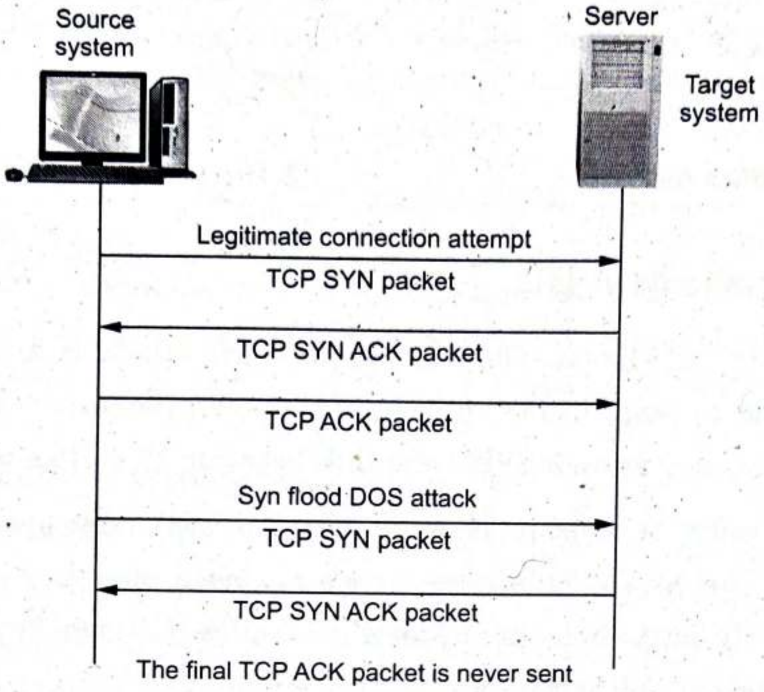


Fig. 1.5.8 SYN flood DOS attack

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

1.5.2.1 Difference between Passive and Active Attack

Sr. No.	Passive attacks	Active attacks
1.	Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions.	Active attacks involve some modification of the data stream or the creation of a false stream.
2.	**Types :** Release of message contents and traffic analysis	**Types :** Masquerade, replay, modification of message and denial of service.
3.	Very difficult to detect.	Easy to detect.

4.	The emphasis in dealing with passive attacks is on prevention rather than detection.	It is quite difficult to prevent active attacks absolutely.
5.	It does not affect the system.	It affects the system.

1.5.3 Man-in-the-Middle Attack

- In cryptography, a **Man-In-The-Middle (MITM) attack** is an attack in which an attacker is able to read, insert and modify at will, meassages between two parties without either party knowing that the link between them has been compromised.
- The attacker must be able to observe and intercept messages going between the two victims. The MITM attack can work against public-key cryptography and is also particularly applicable to the original Diffie-Hellman key exchange protocol, when used without authentication.
- The MITM attack may include one or more of
  1. Eavesdropping, including traffic analysis and possibly a known-plaintext attack.
  2. Chosen ciphertext attack, depending on what the receiver does with a message that it decrypts.
  3. Substitution attack
  4. Replay attacks
  5. Denial of service attack. The attacker may for instance jam all communications before attacking one of the parties. The defense is for both parties to periodically send authenticated status messages and to treat their disappearance with paranoia.
- MITM is typically used to refer to active manipulation of the meassages, rather than passively eavesdropping.

Example of a successful MITM attack against public-key encryption

- Suppose Alice wishes to communicate with Bob and that Mallory wishes to eavesdrop on the conversation, or possibly deliver a false message to Bob. To get started. Alice must ask Bob for his public key. If Bob sends his public key to Alice, but Mallory is able to intercept it, a man-in-the-middle attack can begin.
- Mallory can simply send Alice a public key for which she has the private, matching, key. Alice, believing this public key to be Bob's, then encrypts her message with Mallory's key and sends the enciphered message back to Bob.
- Mallory again intercepts, deciphers the message, keeps a copy, and reenciphers it using the public key Bob originally sent to Alice. When Bob receives the newly enciphered message, he will believe it came from Alice.

- This example shows the need for Alice and Bob to have some way to ensure that they are truly using the correct public keys of each other. Otherwise, such attacks are generally possible in principle, against any message sent using public-key technology.

Defenses against the attack

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

Review Questions

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

1.6 Hash Function

AU : May-17,18, Dec-19

- **Definition :** A hash function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-values.
- The data to be encoded is often called the "message", and the hash value is sometimes called the message digest or simply digest.
- The most common cryptographic uses of hash functions are with digital signatures and for data integrity.
- When hash functions are used to detect whether the message input has been altered, they are called Modification Detection Codes (MDC).
- There is another category of hash functions that involve a secret key and provide data origin authentication, as well as data integrity; these are called Message Authentication Codes (MACs).

One - way Hash Function

- A one-way hash function, also known as a message digest, fingerprint or compression function, is a mathematical function which takes a variable-length input string and converts it into a fixed-length binary sequence.
- Furthermore, a one-way hash function is designed in such a way that it is hard to reverse the process, that is, to find a string that hashes to a given value (hence the name one-way.)
- A good hash function also makes it hard to find two strings that would produce the same hash value. All modern hash algorithms produce hash values of 128 bits and higher.
- Even a slight change in an input string should cause the hash value to change drastically. Even if 1 bit is flipped in the input string, at least half of the bits in the hash value will flip as a result. This is called an **avalanche effect.**
- A common way for one-way hash functions to deal with the variable length input problem is called a **compression function.** Compression functions work by viewing the data being hashed as a sequence of n fixed-length blocks.
- To compute the hash value of a given block, the algorithm needs two things : **the data in the block and an input seed.**
- The input seed is set to some constant value, c, and the algorithm computes the hash value h_1 of the first block. Next, the hash value of the first block, h_1 is used as the seed for the second block.
- The function proceeds to compute the hash value of the second block based on the data in the second block and the hash value of the first block, h_1. So, the hash value for block n is related to the data in block n and the hash value h_{n-1} (for n>1). The hash value of the entire input stream is the hash value of the last block.

Hash Function

- A hash value h is generated by a function H of the form.

  h = H(M)

  where M = Variable - Length message

  H(M) = Fixed - Length hash value.

1.6.1 Requirements of Hash Functions

- The purpose of a hash function is to produce a fingerprint of a file, message or other block of data.

Properties

1. H can be applied to a block of data of any size.
2. H produces a fixed length output.
3. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.
4. For any given value h, it is computationally infeasible to find x such that H(x) = h. This is called **one-way property.**
5. For any given block x, it is computationally infeasible to find y ≠ x such that H(y) = H(x). This is called as **weak collision resistance.**
6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y). This is called as **strong collision resistance.**

Simple hash functions

- For a hash function, the input is viewed as a sequence of n-bit blocks. The input is processed one block at a time in an iterative fashion to produce an n-bit hash function.
- One of the simplest hash functions is the bit-by-bit exclusive-OR of every block. This can be expressed as follows

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

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

m = number of n-bit blocks in the input

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

$\oplus$ = XOR operation

- A simple way to improve matters is to perform a one bit circular shift or rotation, on the hash value after each block is processed.

The procedure is as follows

1. Initially set the n-bit hash value to zero.
2. Process each successive n-bit block of data as follows.
   a. Rotate the current hash value to the left by one bit.
   b. XOR the block into the hash value.

Fig. 1.6.1 shows two types of hash functions.

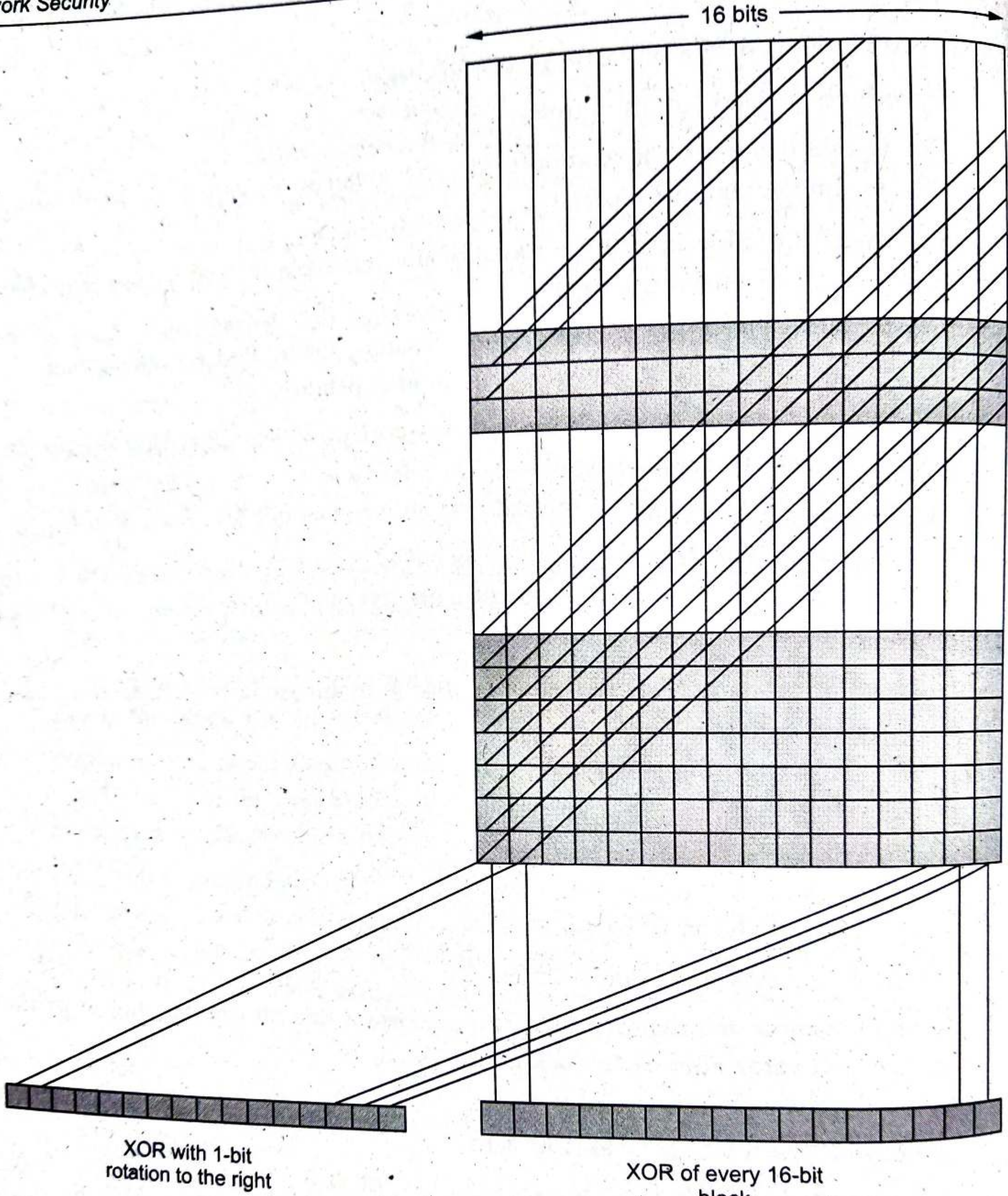

16 bits
XOR with 1-bit rotation to the right
XOR of every 16-bit block

Fig. 1.6.1 Two simple hash functions

1.6.2 Applications of Hash Function

- A typical use of a cryptographic hash would be as follows :
  1. Alice poses a tough math problem to Bob, and claims she has solved it. Bob would like to try it himself, but would yet like to be sure that Alice is not bluffing. Therefore, Alice writes down her solution, appends a random nonce, computes its hash and tells Bob the hash value. This way, when Bob comes up

with the solution himself a few days later, Alice can prove that she had the solution earlier by revealing the nonce to Bob.

2. Second application of secure hashes is verification of message integrity. Determining whether any changes have been made to a message, for example, can be accomplished by comparing message digests calculated before, and after, transmission. A message digest can also serve as a means of reliably identifying a file; several source code management systems, including Git, Mercurial and Monotone, use the sha1sum of various types of content (file content, directory trees, ancestry information, etc) to uniquely identify them.
3. A related application is password verification. Passwords are usually not stored in clear text, for obvious reasons, but instead in digest form. To authenticate a user, the password presented by the user is hashed and compared with the stored hash. This is sometimes referred to as one-way encryption.
4. Hash functions can also be used in the generation of pseudorandom bits. Hashes are used to identify files on peer-to-peer file sharing networks. For example, in an ed2k link, an MD4-variant hash is combined with the file size, providing sufficient information for locating file sources, downloading the file and verifying its contents. Magnet links are another example. Such file hashes are often the top hash of a hash list or a hash tree which allows for additional benefits.

1.6.3 Birthday Attack

- A birthday attack refers to a class of brute-force attacks.
- The attack is named after the statistical property of birthday duplication - you only need 23 people to have a larger than 50 % chance that they are born on the same day of the year.
- This is due to the fact that each time you adding one person to the set of people you are looking for duplicates in, you are looking for duplicates against all the people already in the set, not just one of them.
- The same technique can be used to look for conflicts in one-way functions. Instead of taking one output of the one-way function, you create or acquire a set of values (let us call this a) that have a some property and then create another set of other values that have different properties (let us call this b) and try to find any value that is in both a and b. This is a much smaller problem that finding a value that match a particular value in a.

- The properties in a and b might for instance be
  1. a contains secure hashes of an innocent message and b contains one of a less innocent message, so the attacker can substitute the messages at a later date.
  2. a is the password hashes of a system the attacker wants to get an account on, and b is a set of password hashes that the attacker knows the passwords for.
  3. a is the set of public keys from a Discrete Logarithms based cryptosystem where g and p are static, while b is the set of g^e mod p functions that the attacker knows e for.
- Birthday attacks are often used to find collisions of hash functions. To avoid this attack, the output length of the hash function used for a signature scheme can be chosen large enough so that the birthday attack becomes computationally infeasible.
- Resistance against this attack is why the Unix password hashes use a salt.

1.6.4 Attack on Collision Resistance

- Weak collision resistance : for any x, it is hard to find x' ≠ x such that h(x) = h(x').
- Strong collision resistance : it is hard to find any x, x' for which h(x) = h(x').
- It's easier to find collisions. Therefore strong collision resistance is a stronger assumption.
- Real world hash functions: MD5, SHA-1, SHA-256.
- The weak collision property refers guarantees that an alternative message yielding the same code cannot be found. This prevents forgery when an encrypted hash code is used.

The strong collision property refers to how resistant the hash function is to a class of attacks known as the birthday attack.

1.6.5 Secure of Hash Function and HMAC

- Attacks are of two types,
  1. Brute-force attack 2. Cryptanalysis

Brute - force attacks

1. Hash functions

- The strength of a hash function against brute-force attacks depends solely on the length of the hash code produced by the algorithm.

- **Desirable properties**
  a. **One way :** For any given code h, it is computationally infeasible to find x such that H(x) = h.

  b. **Weak collision resistance :** For any given block x, it is computationally infeasible to find $y \neq x$ with H(y) = H(x).

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

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

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

2. Message authentication codes

- Given one or more text MAC pair $[x_i, C(K, x_i)]$ it is computationally infeasible to compute any text MAC pair [x, C(K, x)] for any new input $x \neq x_i$.
- The attacker would like to come up with the valid MAC code for a given message x.
- There are two lines of attack possible. Attack the key space and attack the MAC value.
- If an attacker can determine the MAC key then it is possible to generate a valid MAC value for any input x.
- An attacker can also work on the MAC value without attempting to recover the key. Here, the objective is to generate a valid MAC value for a given message or to find a message that matches a given MAC value.
- The level of effort for brute-force attack on a MAC algorithm can be expressed as min $(2^k, 2^n)$.

Cryptanalysis

Hash functions

- The hash algorithm involves repeated use of a compression function (f), that takes two inputs and produces an n-bit output.
- Cryptanalysis of hash functions focuses on the internal structure of f and is based on attempts to find efficient techniques for producing collisions for a single execution of f.

1.6.6 HMAC

- The IPsec authentication scheme uses a scheme called Hashed Message Authentication Codes (HMAC), which is an encrypted message digest described in RFC 1024.
- HMAC uses a shared secret key between two parties rather than public key methods for message authentication.

Objectives for HMAC

1. To use, without modifications, available hash function.
2. To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required.
3. To use and handle keys in a simple way.
4. To preserve the original performance of the hash function without incurring a significant degradation.
5. To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions about the embedded hash function.

HMAC algorithm

- Fig. 1.6.2 shows HMAC structure.
- Define the following terms :

H = Embedded hash function

IV = Initial value input to hash function

M = Message input to HMAC

Y_i = i^{th} block of M, $0 \leq i \leq (L - 1)$

L = Number of blocks in M

b = Number of bits in a block

n = Length of hash code produced by embedded hash function.

K = Secret key recommended length is $\geq$ n

K^+ = K padded with zeros on the left so that the result is b bits in length.

ipad = 00110110 (36 in hexadecimal) repeated b/8 times

opad = 01011100 (5C in hexadecimal) repeated b/8 times.

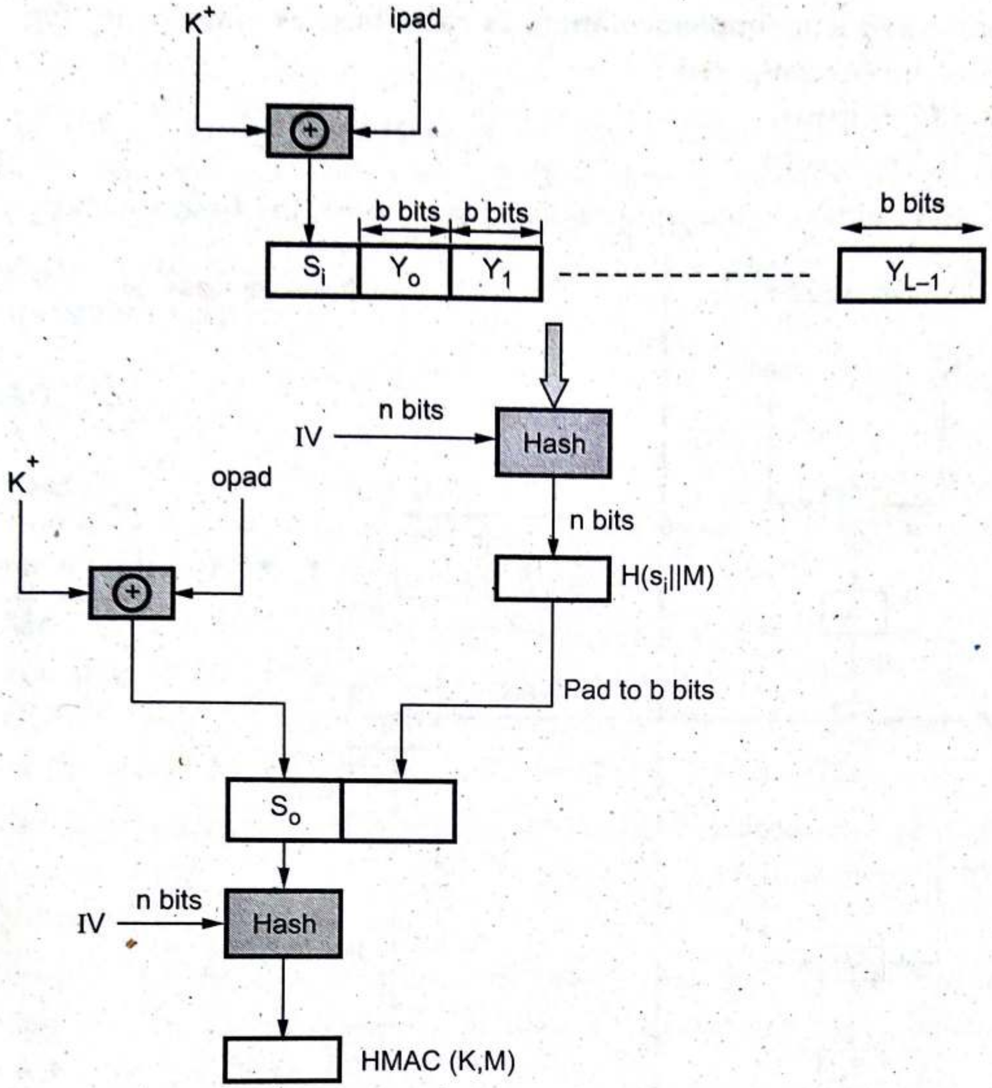


Fig. 1.6.2 HMAC structure

Then HMAC can be expressed as follows :

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

1. Append zeros to the left end of K to create a b-bit string K^+.
2. XOR K^+ with ipad to produce the b-bit block S_i.
3. Append M to S_i.
4. Apply H to the stream generated in step 3.
5. XOR K^+ with opad to produce the b-bit block S_o.
6. Append the hash result from step 4 to S_o.
7. Apply H to the stream generated in step 6 and output the result.

- A more efficient implementation is possible, as shown in Fig. 1.6.3. Two quantities are precomputed :

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

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

  Where f(CV, block) is the compression function for the hash function.

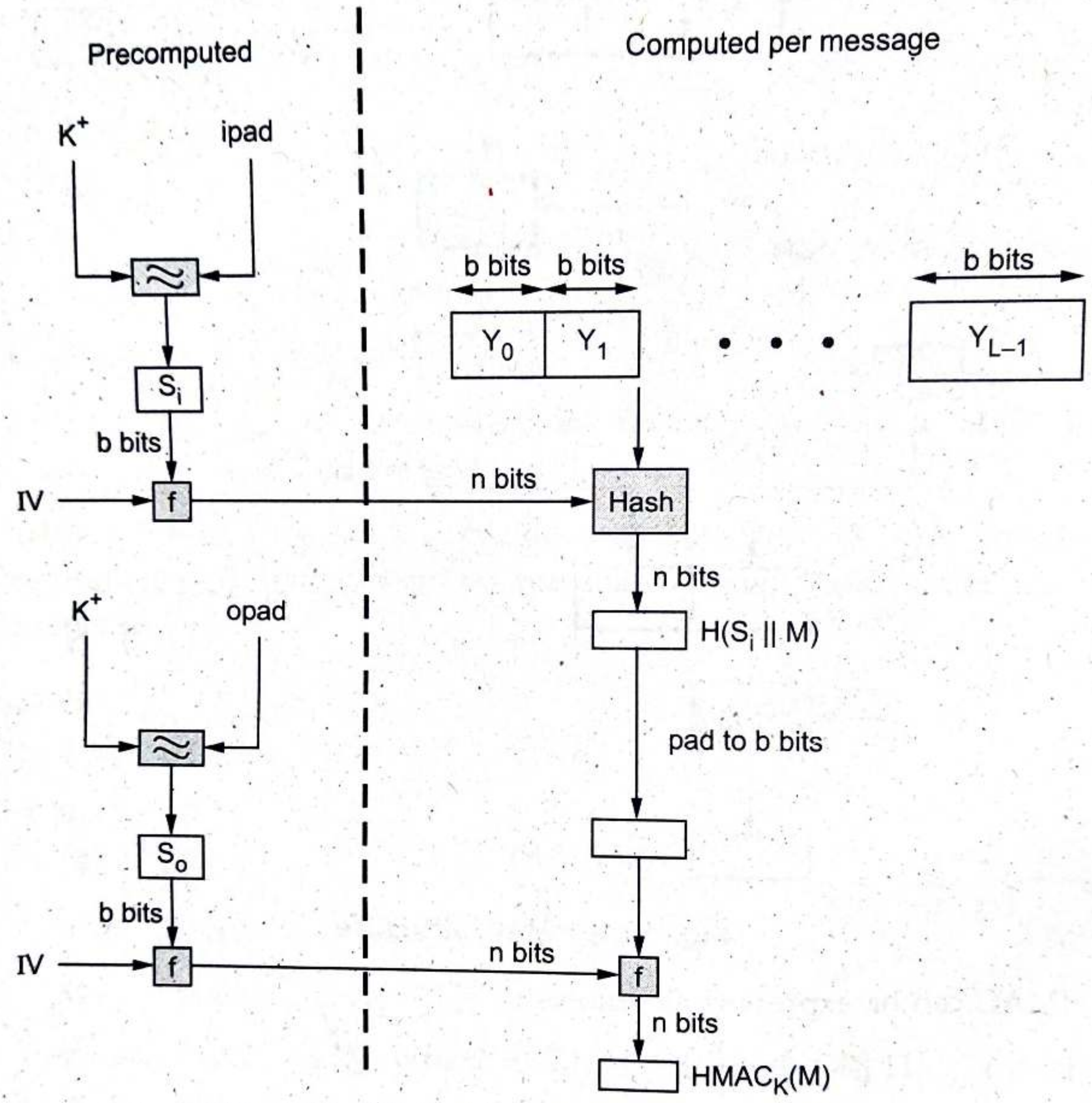


Fig. 1.6.3 Efficient implementation of HMAC

HMAC security

- Know that the security of HMAC relates to that of the underlying hash algorithm.
- Attacking HMAC requires either :
  a) Brute-force attack on key used. This in order of 2n where n is the chaining variable bit-width.

  b) Birthday attack (but since keyed would need to observe a very large number of messages). Like MD5 this is in order of 2n/2 for a hash length of n.
- Choose hash function used based on speed versus security constraints.
- Note that HMAC is more secure than MD5 for birthday attack.

a) In MD5 the attacker can choose any set of messages to find a collision (i.e. H(M) = H(M′).

b) In HMAC since the attacker does not know K, he cannot generate messages offline. For a hash code of 128 bits, this requires 264 observed blocks (i.e. 264 ∗ 29 = 273 bits) generated using the same key. On a 1 Gbps line, this requires monitoring stream of messages with no change of the key for 250,000 years (quite infeasible !!).

1.6.7 CMAC

- Cipher-based Message Authentication Code (CMAC) is a block cipher-based message authentication code algorithm. CMAC mode of operation is used with AES and triple DES.
- The CMAC on a message is constructed by splitting it into blocks of size equal to the block size of the underlying cipher, for instance, 128 bits in the case of the AES, Cipher Block Chaining (CBC)-encrypting the message and retaining the result of the last block encryption as the computed MAC value.
- To avoid certain classes of attack, the last block is subjected, before ciphering, to an exclusive disjunction (XORing) with one of two possible "subkey" values, usually denoted as K1 or K2.
- The choice of which subkey to use is determined by whether the last message block contains padding or not. The subkey values can only be computed by parties knowing the cipher key in use.
- Fig. 1.6.4 shows calculation of CMAC.

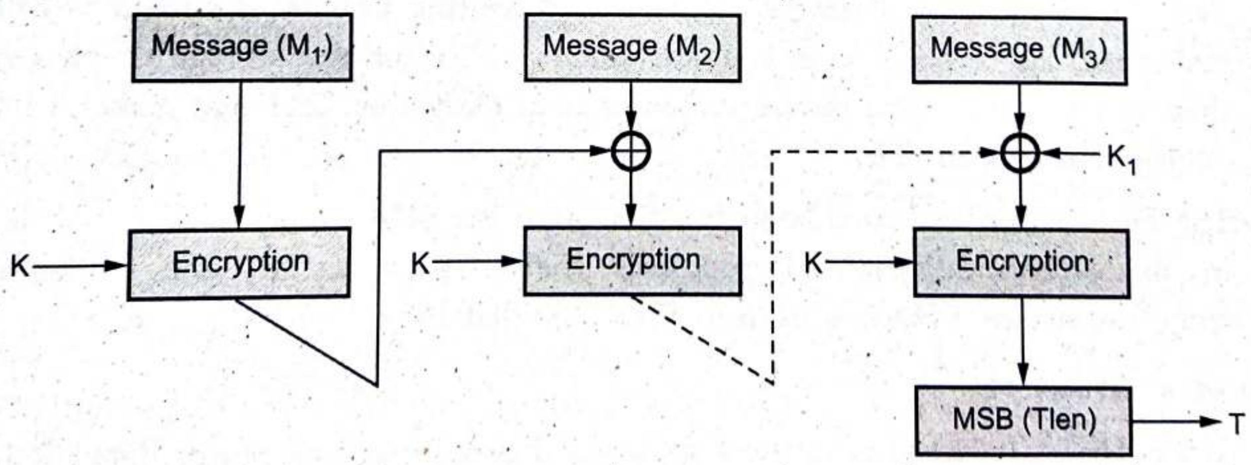


Fig. 1.6.4 Message length is integer multiple of block size

$$C_1 = E(K, M_1)$$

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

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

$$\vdots$$

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

$$T = MSB_{tlen}(C_n)$$

where

T = message authentication code

Tlen = bit length of T

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

1.6.8 Secure Hash Algorithm

- The Secure Hash Algorithm (SHA) was developed by National Institute of Standards and Technology (NIST). It is based on the MD4 algorithm. Based on different digest lengths, SHA includes algorithms such as SHA-1, SHA-256, SHA-384, and SHA-512.
- Unlike encryption, given a variable length meassge **x**, a secure hash algorithm computes a function **h(x)** which has a fixed and often smaller number of bits. When a message of any length is less than 2^{64} bits is input, the SHA-1 produces a 160-bit output called message digest.
- SHA-1 called secure bacause it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest.
- There are a number of attacks on SHA-1, all relating to what is known as collision resistance. For examples, if you are using SHA-1 for the storage of passwards, there are no passoword recovery attacks as at December 2011 that make use of the collision attacks on SHA-1.
- The most commonly used hash function from the SHA family is SHA-1. It is used in many applications and protocols that require secure and authenticated communications. SHA-1 is used in SSL/TLS, PGP, SSH, S/MIME, and IPSec.

Features of SHA-1 :

1. The SHA-1 is used to compute a message digest for a message or data file that is provided as input.
2. The message or data file should be considered to be a bit string.
3. The length of the message is the number of bits in the message (the empty message has length 0).

4. If the number of bits in a message is a multiple of 8, for compactness we can represent the message in hex.
5. The purpose of message padding is to make the total length of a padded message a multiple of 512.
6. The SHA-1 sequentially processes blocks of 512 bits when computing the message digest.
7. The 64-bit integer is 1, the length of the original message.
8. The padded message is then processed by the SHA-1 as n 512-bit block.

- SHA-1 was cracked in the year 2005 by two different research groups. In one of these two demonstrations, Xiaoyun Wang, Yigun Lisa Yin, and Hongbo Yu demonstrated that it was possible to come up with a collosion for SHA-1 within a space of size only 2^{69}, which was far fewer that the security level of 2^{80} that is associated with this hash function.
- New hash function SHA-512 is introduced to overcome problem of SHA-1.

1.6.9 Secure Hash Algorithm (SHA-512)

- The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST). SHA-1 produces a hash value of 160 bits.
- In 2002, NIST produced a revised version of the standard, FIPS 180-2, that defined three new version of SHA, with hash value lengths of 256,384 and 512 bits, known as SHA-256, SHA-384 and SHA-512.
- Comparison of SHA parameters

Sr. No.	Parameters	SHA-1	SHA-256	SHA-384	SHA-512
1.	Message digest size	160	256	384	512
2.	Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
3.	Block size	512	512	1024	1024
4.	Word size	32	32	64	64
5.	Number of steps	80	64	80	80
6.	Security	80	128	192	256

- For both SHA-1 and SHA-256, one begins by converting the message to a unique representation of the message that is a multiple of 512 bits in length, without loss of information about its exact original length in bits, as follows : Append a 1 to the message.
- Then add as many zeroes as necessary to reach the target length, which is the next possible length that is 64-bits less than a whole multiple of 512 bits. Finally, as a 64-bit binary number, append the original length of the message in bits.

Description of SHA-1

- Expand each block of 512, when it is time to use it, into a source of 80 32-bit subkeys as follows : The first 16 subkeys are the block itself. All remaining subkeys are generated as follows : Subkey N is the exclusive OR of subkeys N-3, N-8, N-14 and N-16, subjected to a circular left shift of one place. Starting from the 160-bit block value (in hexadecimal).

  67452301 EFCDAB89 98BADCFE 10325476 C3D2E1F0

  As input for the processing of the *first* 512-bit block of the modified message, for each message block, do the following
- Encipher the starting value using the 80 sub keys for the current message block. Add each of the 32-bit pieces of the cipher text result to the starting value, modulo 2^32, of course and use that result as the starting value for handling the next message block.
- The starting value created at the end of handling the last block is the hash value, which is 160 bits long.

The SHA "block cipher" component

- The main calculation in SHA enciphers a 160-bit block using 80 32-bit subkeys in 80 rounds. This calculation is somewhat similar to a series of Feistel rounds, except that instead of dividing the block into two halves, it is divided into five pieces.
- An F-function is calculated from four of the five pieces, although it is really the XOR of a function of three of the pieces and a circular left shift of a fourth, and XORed with one piece, which is also modified by being XORed with the current round's subkey and a constant.
- The same constant is used over each group of 20 rounds. One of the other blocks is also altered by undergoing a circular left shift, and then the (160-bit) blocks are rotated.
- The F-function, as well as the constant, is changed every 20 rounds. Calling the five pieces of the 160-bit block being "encrypted" a, b, c, d and e, the rounds of the SHA "block cipher" component proceed as follows

- Change a by adding the current constant to it. The constants are, in hexadecimal
  - For rounds 1 to 20 : 5A827999
  - For rounds 21 to 40 : 6ED9EBA1
  - For rounds 41 to 60 : 8F1BBCDC
  - For rounds 61 to 80 : CA62C1D6
- Change a by adding the appropriate subkey for this round to it.
- Change a by adding e, circular left-shifted 5 places to it.
- Change a by adding the main f-function of b, c and d to it, calculated as follows :
  - For rounds 1 to 20, it is (b AND c) OR (NOT b) AND (d).
  - For rounds 21 to 40, it is b XOR c XOR d.
  - For rounds 41 to 60, it is (b AND c) OR (b AND d) OR (c AND d).
  - For rounds 61 to 80, it is again b XOR c XOR d.
- Change d by giving it a circular *right* shift of 2 positions (or, for consistency, a circular left shift of 30 places.)
- Then swap pieces, by moving each piece to the next earlier one, except that the old a value is moved to e.
- There are various types in SHA such as SHA-256, SHA-384, and SHA-512.

SHA-512 logic

- Fig. 1.6.5 shows message digest generation using SHA-512.

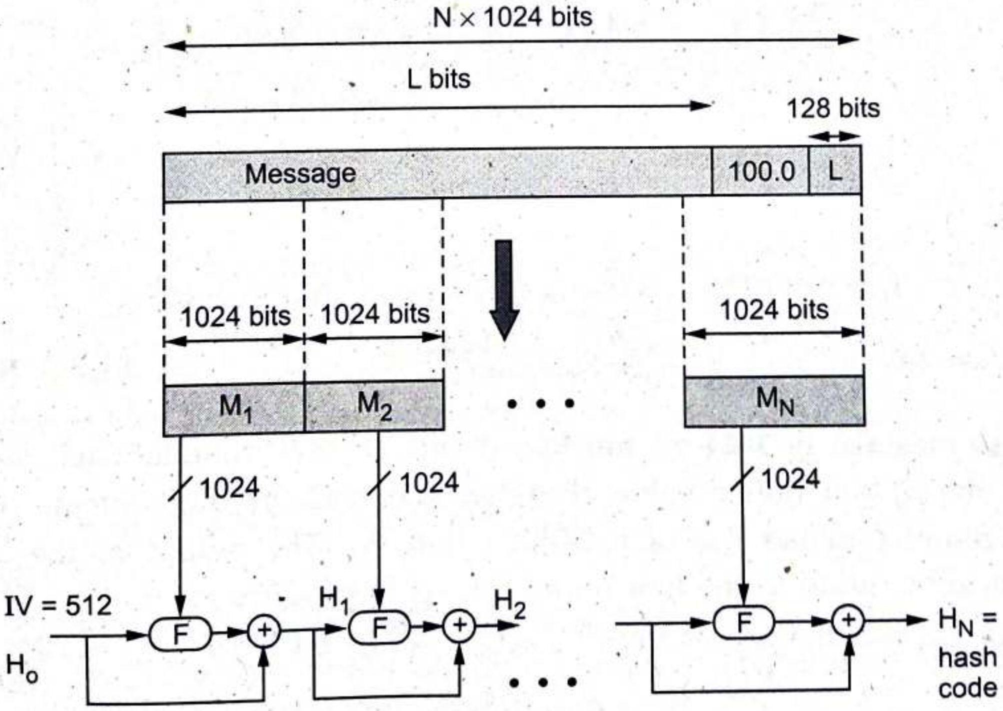


Fig. 1.6.5 Message digest using SHA-512

- The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces as output a 512-bit message digets. The input is processed in 1024-bit blocks.

Steps

1. **Append padding bits :** The message is padded so that its length is congruent to 896 modulo 1024. Padding consists of a single 1-bit followed by the necessary number of 0-bits.
2. **Append length :** A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer that contains the length of the original message (before the padding).
3. **Initialize has buffer :** A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialised to the following 64-bit integers (hexadecimal values)

Sr. No.	Register	Values
1.	a	6A09E667F3BCC908
2.	b	BB67AE8584CAA73B
3.	c	3C6EF372FE94F82B
4.	d	A54FF53A5F1D36F1
5.	e	S10E527FADE682D1
6.	f	9B05688C2B3E6C1E
7.	g	1F83D9ABFB41BD6B
8.	h	5BE0CDI9137E2179

4. **Process message in 1024-bit blocks :** It consist of 80 rounds. Each round takes as input the 512-bit buffer value abcdefgh and updates the contents of the buffer. Each round t makes use of a 64-bit value W_t. The output of the last round is added to the input to the first round (H_{i-1}) to produce H_i.

- Fig. 1.6.6 shows the processing of a single 1024 - bit block.

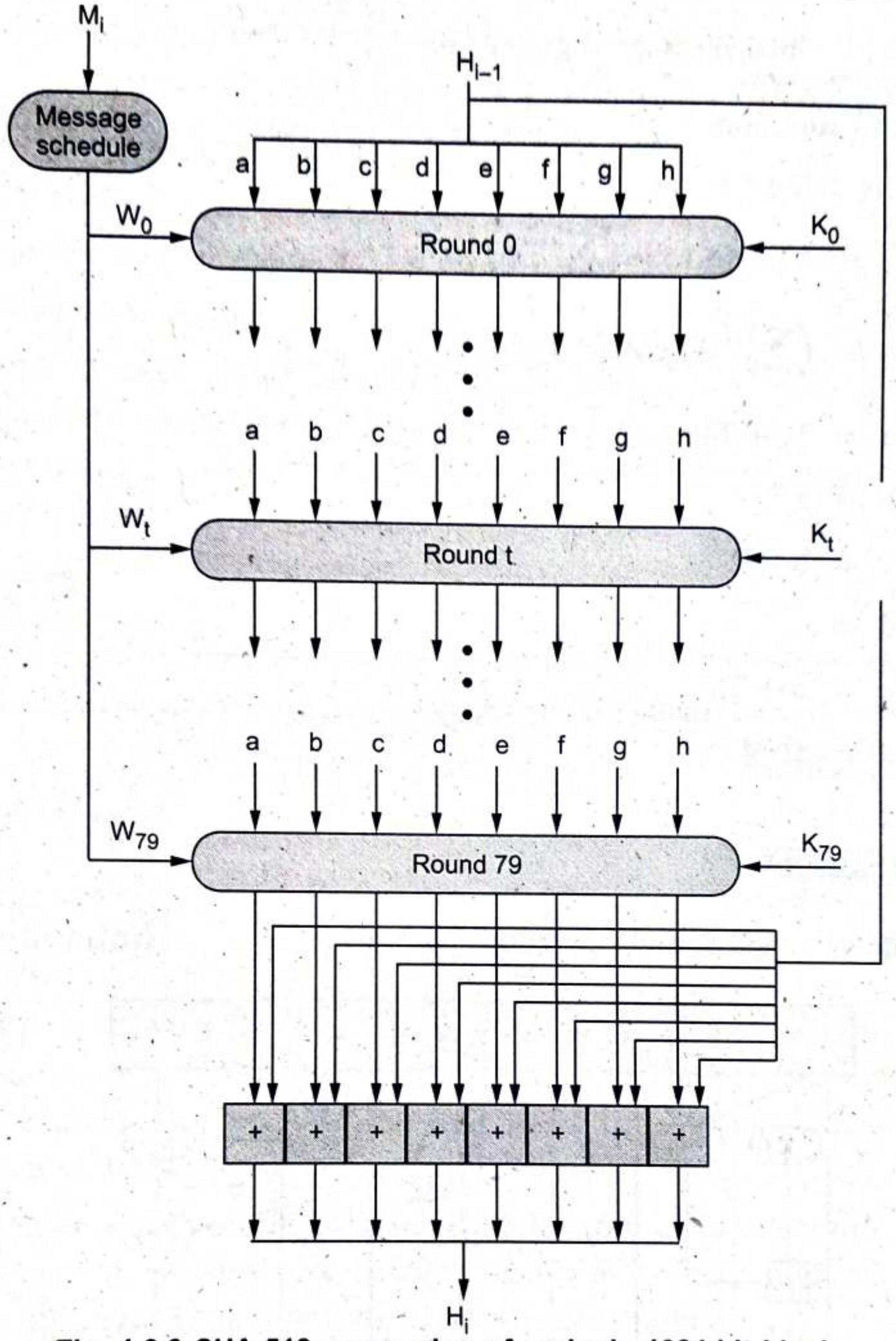


Fig. 1.6.6 SHA-512 processing of a single 1024-bit block

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

- The behaviour of SHA-512 is as-follows

$$H_0 = IV$$

$$H_i = SUM_{64}\ (H_{i-1}, \text{abcdefghj})$$

$$MD = H_N,$$

where IV = Initial value of the abcdefgh buffer.

abcdefgh_i = The output of the last round of processing of the i^{th} message block.

N = The number of blocks in the message.

SUM_{64} = Addition modulo 2^{64} performed separately on each word of the pair of inputs.

MD = Final message digest value

SHA - 512 round function

Each round is defined by the following set of equations.

$$T_1 = h+ch(e,f,g)+\left(\sum\nolimits_1^{512} e\right)+W_t+K_t$$

$$T_2 = \left(\sum\nolimits_0^{512} a\right)+Maj(a,b,c)$$

$$a = T_1 + T_2$$

$$b = a$$

$$c = b$$

$$d = c$$

$$e = d + T_1$$

$$f = e$$

$$g = f$$

$$h = g$$

Fig. 1.6.7 shows single round operation.

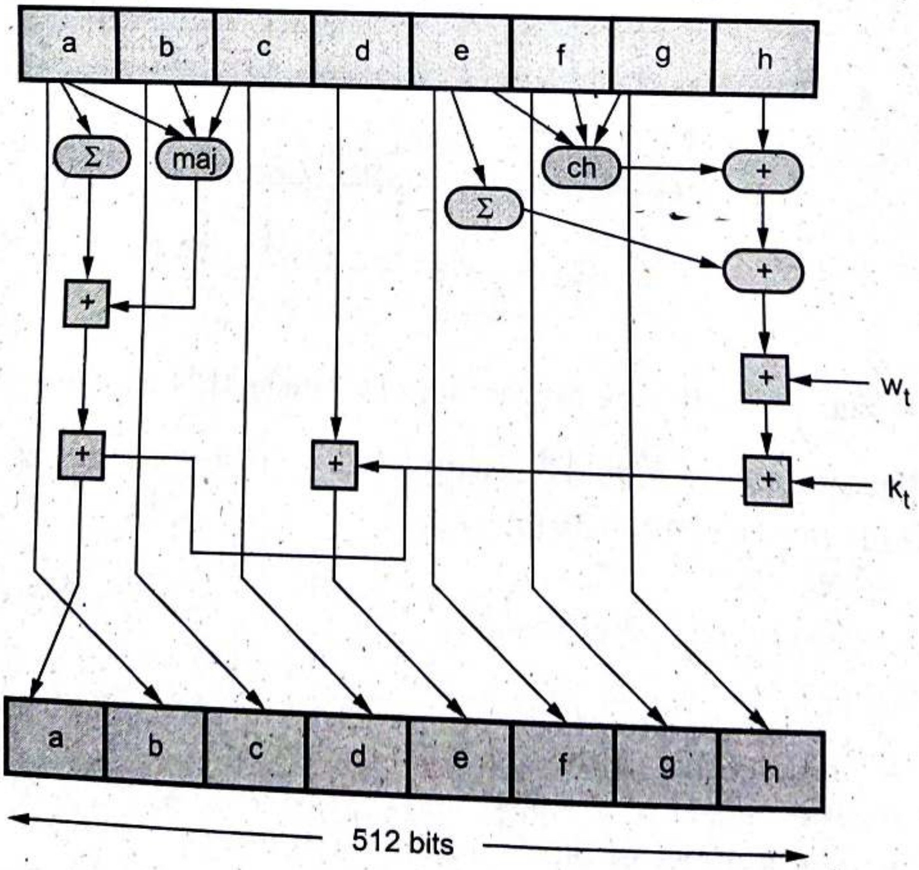


Fig. 1.6.7 Single round operation

Example 1.6.1 *Compare the performance of RIPEMD - 160 algorithm and SHA - 1 algorithm.*

AU : May-17, Marks 16

Solution : RIPEMD-160 verses SHA-1 :

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

Review Questions

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

   AU : May-18, Marks 16

2. *List the design objectives of HMAC and explain the algorithm in detail.*

   AU : Dec.-19, Marks 13

1.7 Authentication

AU : Dec.-19

Authentication

- Authentication techniques are used to verify identity. The authentication of authorized users prevents unauthorized users from gaining access to corporate information systems.
- Authentication method is of validating the identity of user, service or application. The use of authentication mechanisms can also prevent authorized users from accessing information that they are not authorized to view.
- Data authentication means providing data integrity as well as that the data have been received from the individual who claimed to supply this information.

In authentication :

a. A Brute force attack is an automated process of trial and error used to guess a person's user name, password, credit-card number of cryptographic key.

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

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

Authorization

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

In authorization :

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

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

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

1.7.1 Authentication Requirements

- Attacks can be identified as follows :
  1. **Disclosure** : Release of message contents to any person or process not possessing the appropriate cryptographic key.
  2. **Traffic analysis** : Discovery of the pattern of traffic between parties.
  3. **Masquerade** : Insertion of messages into the network from a fraudulent source.
  4. **Sequence modification** : Any modification to a sequence of messages between parties, including insertion, deletion and reordering.
  5. **Content modification** : Changes to the contents of a message, including insertion, deletion, transposition and modification.
  6. **Timing modification** : Delay or replay of messages.
  7. **Source repudiation** : Denial of transmission of message by source.
  8. **Destination repudiation** : Denial of receipt of message by destination.
- Message authentication is a procedure to verify that received messages come form the alleged source and have not been altered.
- Digital signature is an authentication technique that also includes measures to counter repudiation by the source.

1.7.2 Authentication Function

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

1) Message encryption

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

Symmetric encryption

- Fig. 1.7.1 shows the uses of message encryption in symmetric encryption.

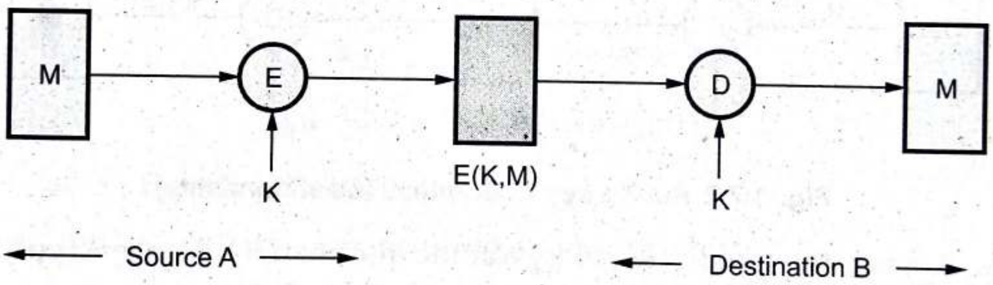


Fig. 1.7.1 Symmetric encryption (confidentiality and authentication)

- A message M transmitted from source A to destination B is encrypted using a secret key K shared by A and B. If no other party knows the key, then confidentiality is provided.
- Destination B is assured that the message was generated by A. Because of secret key used by both party, it provides authentication as well as confidentiality.
- Given a decryption function D and a secret key K, the destination will accept any input X and produce output Y = D(K, X).
- If X is the ciphertext of a legitimate message M produced by the corresponding encryption function, then Y is some plaintext message M. Otherwise, Y will likely be a meaningless sequence of bits.

- For example, suppose that we are transmitting English language message using a caesar cipher with a shift of two A sends the following legitimate ciphertext :

  nbsftfbupbutboeepftfbupbutboemjuumfmbnct

  B decrypt to produce the following plaintext :

  lzqdrdzsnzsrzmccmdrdzsnzsrzmckhsskdkzlar

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

  zuvrsoevgqxlzwigamdvnmhpmccxiuureosfbceb

  This decrypts to :

  Which does not fit the profile of ordinary English.

Public key encryption

- Public key encryption provides confidentiality but not authentication. Fig. 1.7.2 shows public key encryption with confidentiality in message encryption.

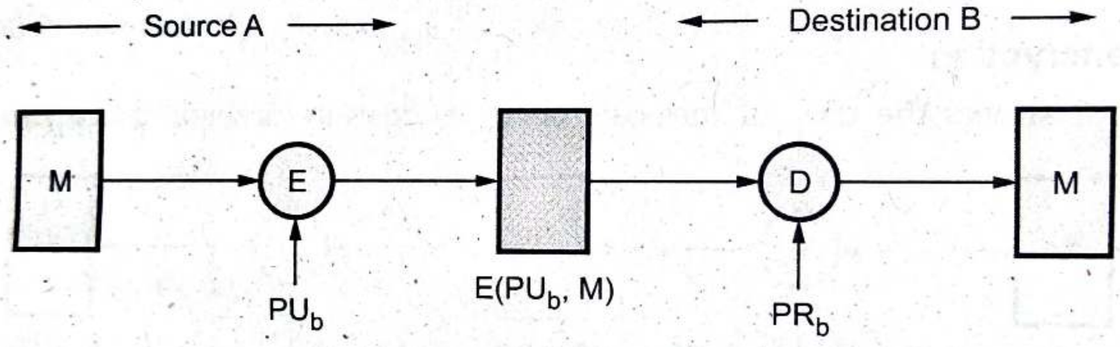


Fig. 1.7.2 Public key encryption (Confidentiality)

- Source A uses the public key PU_b of the destination B to encrypt message M. Because only B has the corresponding private key PR_b, only B can decrypt the message.
- This method provides no authentication because any opponent could also use B's public key to encrypt a message, claiming to be A.
- Fig. 1.7.3 shows the message encryption in public key encryption with authentication and signature.

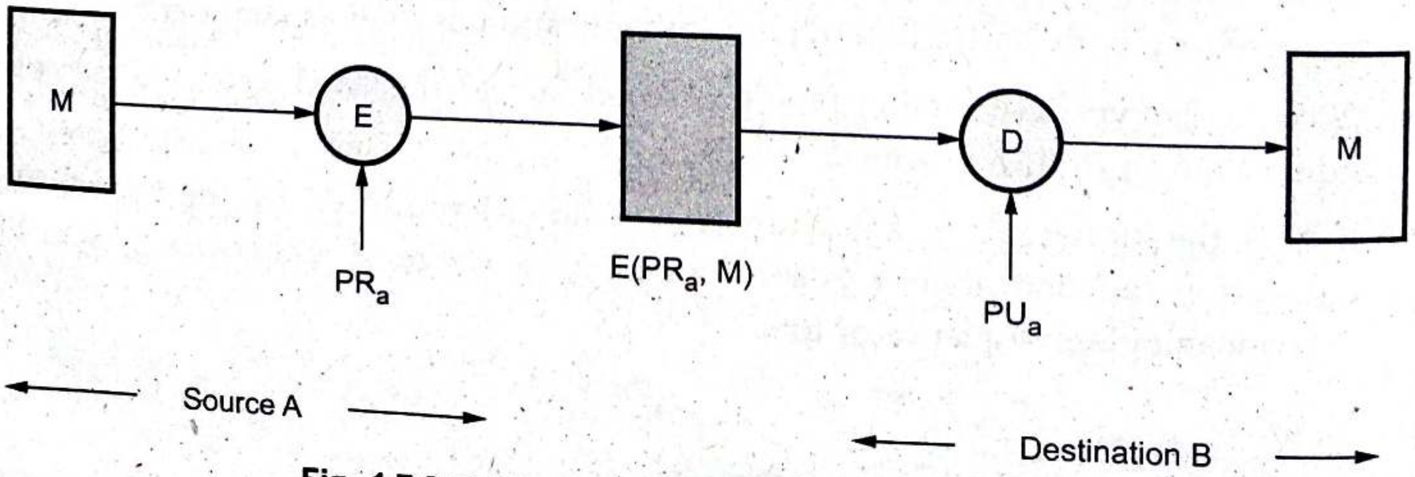


Fig. 1.7.3 Public

- A uses its private key to encrypt the message, and B uses A's public key to decrypt.
- It provides authentication. The message must have come from A because A is the only party that possesses PR_a.
- It also provides digital signature. Only A could have constructed the ciphertext because only A possesses PR_a. Not even B, the recipient could have constructed the ciphertext.
- To provide both confidentiality and authentication, A can encrypt M first using its private key, which provides the digital signature and then using B's public key, which provides confidentiality.
- Fig. 1.7.4 shows confidentiality, authentication and signature for public key encryption.

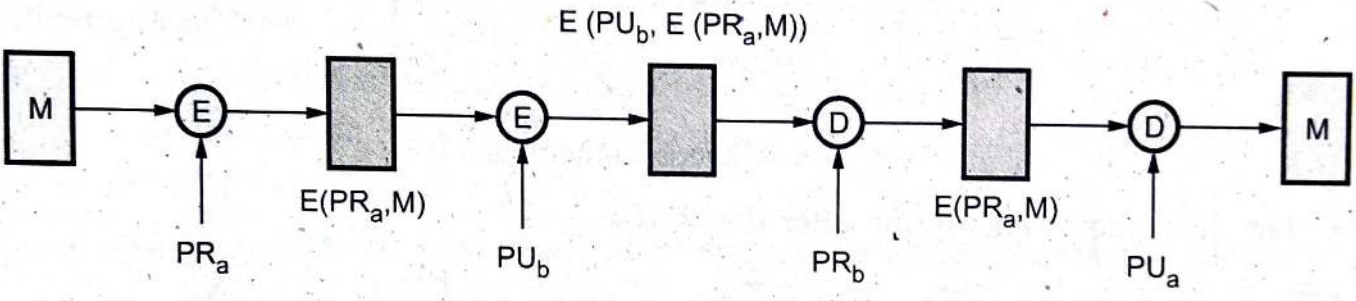


Fig. 1.7.4 Public key encryption

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

2) Message Authentication Code (MAC)

- MAC is an alternative technique which uses secret key. This technique assumes that two communicating parties, share a common secret key K.
- When A has a message to send to B, it calculates the MAC.

MAC = C(K, M)

where M = Input message

C = MAC function

K = Shared secret key

MAC = Message authentication code

- Calculated MAC and message are transmitted to the receiver. The receiver performs the same calculation on the received message.
- Received MAC is compared with the calculated MAC. If both are matches, then
  1. The receiver is assured that the message has not been altered.
  2. The receiver is assured that the message is from the alleged sender.

3. If the message includes a sequence number, then the receiver can be assumed of the proper sequence because an attacker cannot successfully alter the sequence number.

- Fig. 1.7.5 shows the message authentication.
- Fig. 1.7.5 provides authentication but not confidentiality. Confidentiality can be provided by performing message encryption either after or before the MAC algorithm.

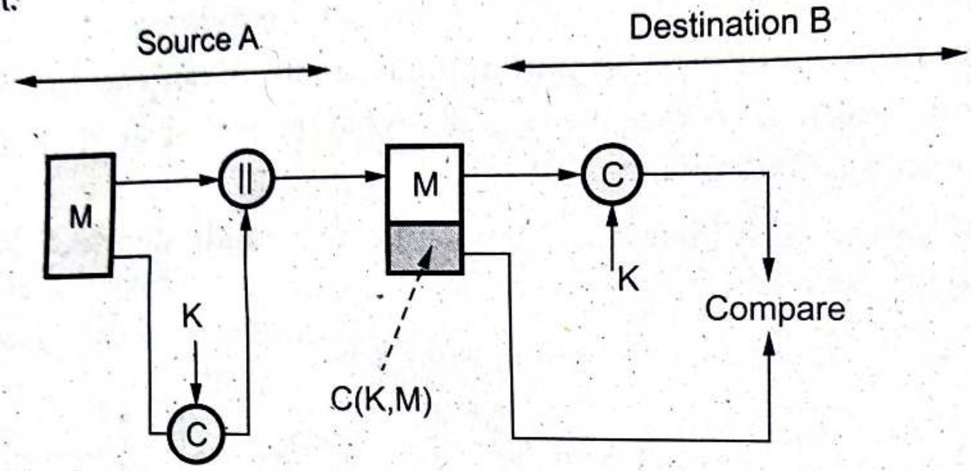


Fig. 1.7.5 Message authentication

- Fig. 1.7.6 shows encryption after the MAC.

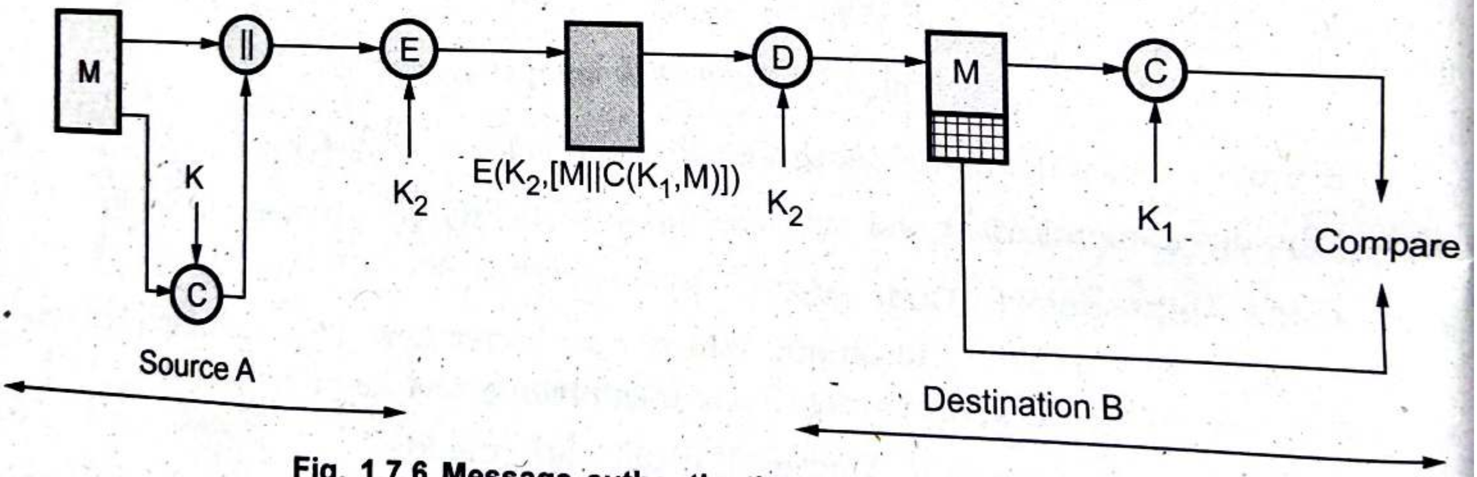


Fig. 1.7.6 Message authentication and confidentiality

- Two separate keys are needed, each of which is shared by the sender and the receiver. Here MAC is calculated with the message input and is then concatenated to the message. The entire block is then encrypted.
- Fig. 1.7.7 shows the message authentication and confidentiality with encryption.
- Here also two separate keys are needed. The message is encrypted first. Then the MAC is calculated using the resulting ciphertext and is concatenated to the ciphertext to form the transmitted block.

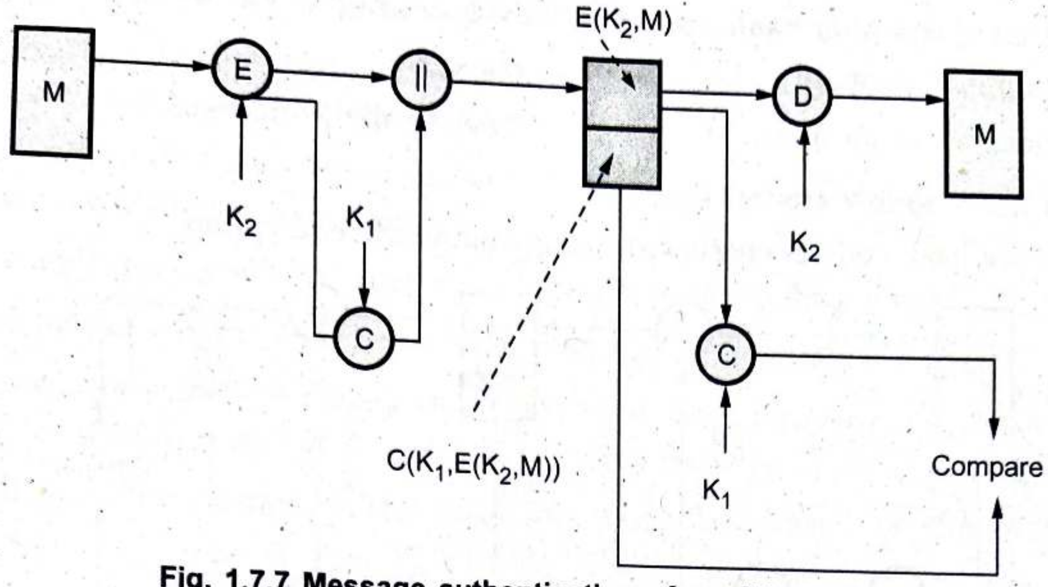


Fig. 1.7.7 Message authentication of confidentiality (authentication tied to ciphertext)

Applications of MAC

- Following are the situations in which MAC used.
  1. Application in which the same message is broadcast to a number of destinations.
  2. Authentication of a computer program in plaintext is an attractive service.
  3. Another scenario is an exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages.

3) Hash function

- A hash function takes an input m, and computes a fixed size string known as a hash.
- Unlike a MAC, a hash code does not use a key but is a function only of the input message.
- Hash code is also referred to as a **message digest** or **hash value.**
- A change to any bit or bits in the message results in a change to the hash code.
- Fig. 1.7.8 (a) shows the basic uses of hash function.

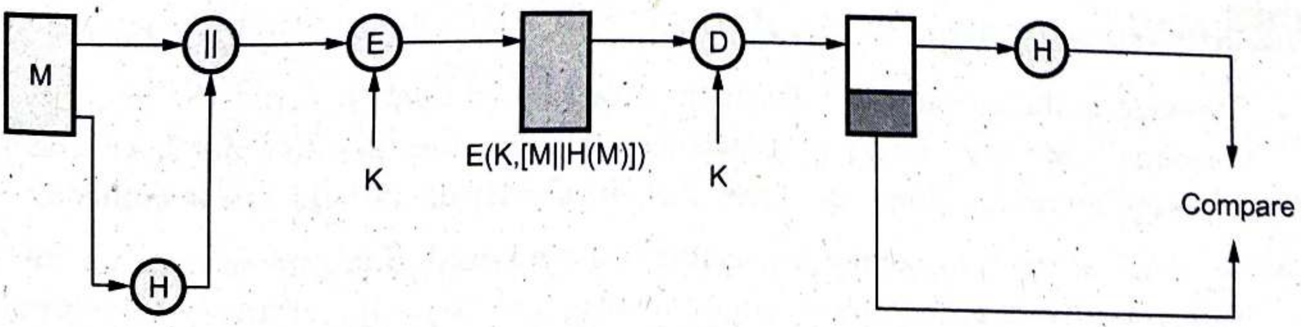


Fig. 1.7.8 (a) Encrypt message plus hash code

1. Encrypt message plus hash code

- Provide confidentiality : Only A and B share K.
- Provides authentication : H(M) is cryptographically protected.

2. Encrypt hash code - shared secret key

- Only the hash code is encrypted, using symmetric encryption.

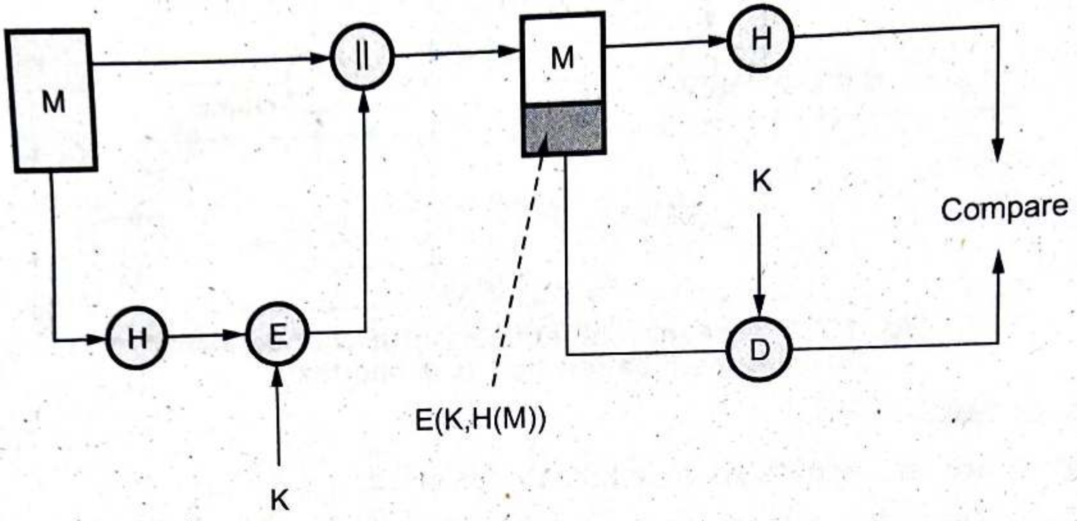


Fig. 1.7.8 (b) Encrypt hash code - shared secret key

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

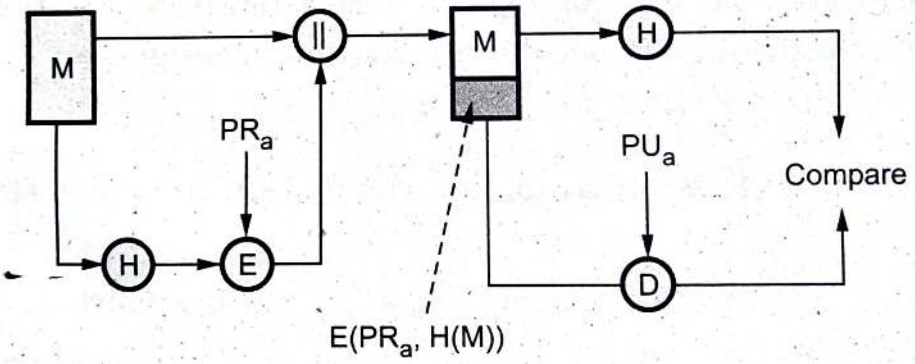


Fig. 1.7.8 (c) Encrypt hash code - sender's private key

3. Encrypt hash code - sender's private key.

- Provides authentication and digital signature.

1.7.3 MAC

- Message authentication is a mechanism or service used to verify the integrity of a message. Message integrity guarantees that the message has not been changed. Message authentication guarantees that the sender of the message is authentic.
- A MAC algorithm, sometimes called a keyed hash function accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC. The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers to detect any changes to the message content.

Properties of Message Authentication Codes

1. Cryptographic checksum : A MAC generates a cryptographically secure authentication tag for a given message.
2. Symmetric : MACs are based on secret symmetric keys. The signing and verifying parties must share a secret key.
3. Arbitrary message size : MACs accept messages of arbitrary length.
4. Fixed output length : MACs generate fixed-size authentication tags.
5. Message integrity : MACs provide message integrity: Any manipulations of a message during transit will be detected by the receiver.
6. Message authentication : The receiving party is assured of the origin of the message.
7. No non-repudiation : Since MACs are based on symmetric principles, they do not provide non-repudiation.

- MACs provide two security services, message integrity and message authentication, using symmetric ciphers. MACs are widely used in protocols. Both of these services are also provided by digital signatures, but MACs are much faster.
- MACs do not provide non-repudiation.
- In practice, MACs are either based on block ciphers or on hash functions.
- HMAC is a popular MAC used in many practical protocols such as Transport Layer Security (TLS) indicated by a small lock in the browser.

Applications of MAC

- Following are the situations in which MAC used.
  1. Application in which the same message is broadcast to a number of destinations.
  2. Authentication of a computer program in plaintext is an attractive service.
  3. Another scenario is an exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages.
- Message Authentication Codes (MAC) also known as a cryptographic check. The MAC is generated by a function C.

$$\text{MAC} = C(K, M)$$

where M = Variable length message

K = Secret key shared only by sender and receiver.

$C(K, M)$ = Fixed length authenticator

- Security of the MAC generally depends on the bit length of the key. Weakness of the algorithm is the brute force attack.
- For a ciphertext - only attack, the opponent, given ciphertext C, would perform $P_i = D(K_i, C)$ for all possible key values K_i until a P_i was produced that matched the form of acceptable plaintext.

Suppose the key size is greater than the MAC size :

- **Round 1**

Given : M_1, $MAC_1 = C(K_1\ M_1)$

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

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

- **Round 2**

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

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

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

- On average, α rounds will be needed if $K = \alpha \times n$

For example : If the key size is 80-bit and MAC is 32 bits long, then the first round will produce about 2^{48} possible keys.

Key length is less than or equal to MAC length

- First round will produce a single match.
- It is possible that more than one key will produce such a match, in which case the opponent would need to perform the same test on a new (message, MAC) pair. Consider the following MAC algorithm.
- Let $M = (X_1 \,||\, X_2 \,||\, \ldots\ldots\ldots \,||\, X_m)$ be a message that is treated as a concatenation of 64-bit blocks X_i. Then define

$$\Delta(M) = X_1 \oplus X_2 \oplus X_3 \oplus \ldots\ldots \oplus X_m.$$

$$C(K, M) = E(K, \Delta(M))$$

Where $\oplus$ is the exclusive-OR (XOR) and the encryption algorithm is DES in electronic codebook mode.

- Key length = 56 bits

  MAC length = 64 bits
- If an opponent observes $\{M \,||\, C(K, M)\}$, a brute force attempt to determine. K will require at least 2^{56} encryptions.

- Assume that an opponent knows the MAC function C but does not know K. Then the MAC function should satisfy the following requirements :
  1. If an opponent observes M and C(K, M), it should be computationally infeasible for the opponent to construct 0 message M′ such that C(K, M′) = C(K, M).
  2. C(K, M) should be uniformly distributed in the sense that for randomly chosen messages, M and M′, the probability that C(K, M) = C(K, M′) is 2^{-n}, where n is the number of bits in the MAC.
  3. Let M′ be equal to some known transformation on M. That is, M′ = f(M).

Message authentication code based on DES

- The data authentication algorithm based on DES, has been one of the most widely used MAC for a number of years. The algorithm can be defined as using the cipher block chaining mode of operation of DES with an initialization vector of zero.
- Fig. 1.7.9 shows the data authentication algorithm.

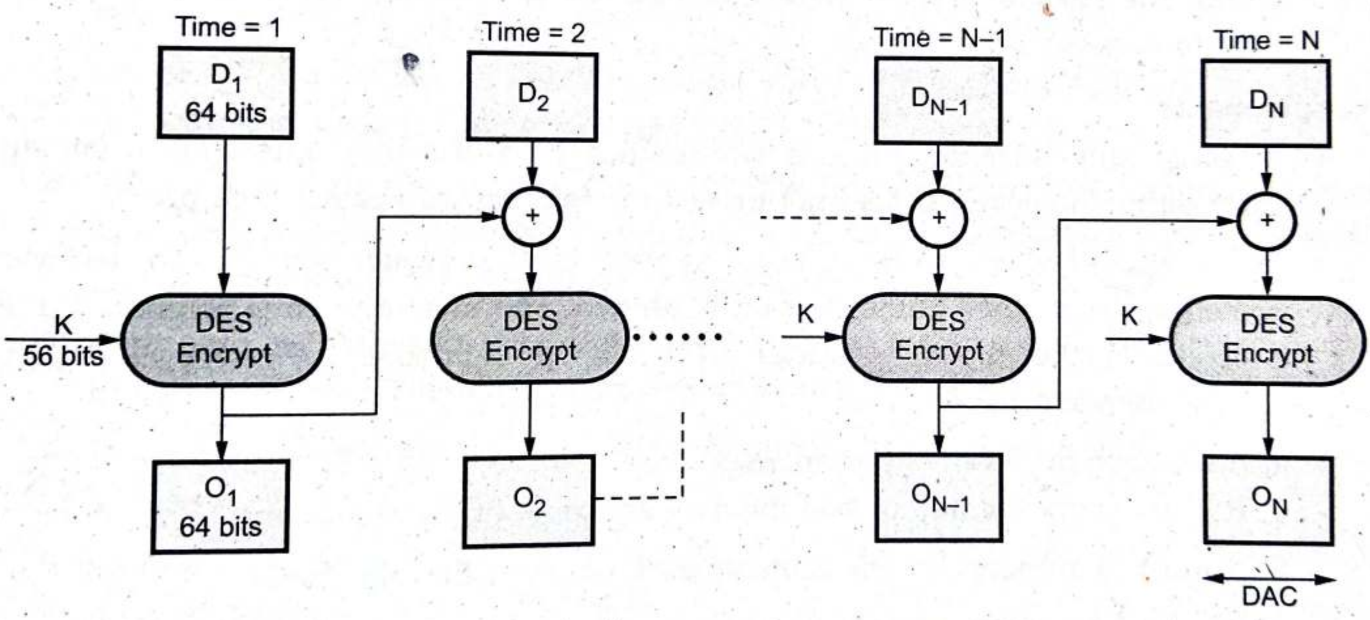


Fig. 1.7.9 Data authentication algorithm

- The algorithm can be defined as using the cipher block chaining mode of operation of DES. The data to be authenticated are grouped into contiguous 64-bit blocks : D_1, D_2, D_3, ..., D_N.
- Using the DES encryption algorithm (E) and a secret key (K), a data authentication code (DAC) is calculated as follows

$$O_1 = E(K, D_1)$$

$$O_2 = E(K, [D_2 \oplus O_1])$$

$$O_3 = E(K, [D_3 \oplus O_2])$$

$$\vdots$$

$$O_N = E(K, [D_N \oplus O_{N-1}])$$

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

Review Question

1. *Compare the uses of MAC and Hash function. Represent them using appropriate diagrams.*

AU : Dec.-19, Marks 8

1.8 Digital Signatures

AU : Dec.-19,20,22

- A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creator's private key.

Requirements

- Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other.
- In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature is analogous to the handwritten signature.
- It must have the following properties
  1. It must verify the author and the date and time of the signature.
  2. It must to authenticate the contents at the time of the signature.
  3. It must be verifiable by third parties, to resolve disputes.
- The digital signature function includes the authentication function. On the basis of these properties, we can formulate the following requirements for a digital signature.
- Must be a bit pattern depending on the message being signed.
- Signature must use some information unique to the sender to prevent forgery and denial.
- Computationally easy to produce a signature.
- Computationally easy to recognize and verify the signature.

- Computationally infeasible to forge a digital signature.
  a) either by constructing a new message for an existing digital signature.
  b). or by constructing a fraudulent digital signature for given message.
- Practical to retain a copy of the digital signature in storage .

Two general schemes for digital signatures

1) Direct 2) Arbitrated

1.8.1 Arbitrated Digital Signatures

Every signed message from A to B goes to an arbiter BB (Big Brother) that everybody trusts.

- BB checks the signature and the timestamp, origin, content, etc.
- BB dates the message and sends it to B with an indication that it has been verified and it is legitimate.

e.g. Every user shares a secret key with the arbiter

- A sends to BB in an encrypted form the plaintext P together with B's id, a timestamp and a random number RA.
- BB decrypts the message and thus makes sure it comes from A; it also checks the timestamp to protect against replays.
- BB then sends B the message P, A's id, the timestamp and the random number RA; he also sends a message encrypted with his own private key (that nobody knows) containing A's id, timestamp t and the plaintext P (or a hash).
- B cannot check the signature but trusts it because it comes from BB-he knows that because the entire communication was encrypted with KB.
- B will not accept the messages or messages containing the same RA to protect against replay.
- In case of dispute, B will show the signature he got from BB (only B may have produced it) and BB will decrypt it.

1.8.2 Direct Digital Signature

- This involves only the communicating parties and it is based on public keys.
- The sender knows the public key of the receiver.
- Digital signature : Encrypt the entire message (or just a hash code of the message) with the sender's private key.
- If confidentiality is required : Apply the receiver's public key or encrypt using a shared secret key.

- In case of a dispute the receiver B will produce the plaintext P and the signature E(KRA, P) - the judge will apply KUA and decrypt P and check the match : B does not know KRA and cannot have produced the signature himself.

Weaknesses

- The scheme only works as long as KRA remains secret : If it is disclosed (or A discloses it herself), then the argument of the judge does not hold : anybody can produce the signature.
- **Attack :** To deny the signature right after signing, simply claim that the private key has been lost-similar to claims of credit card misuse.
  i.e. If A changes her public-private keys (she can do that often) the judge will apply the wrong public key to check the signature.
- **Attack :** To deny the signature change your public-private key pair-this should not work if a PKI is used because they may keep trace of old public keys.
  i.e. A should protect her private key even after she changes the key.
- **Attack :** Eve could get hold of an old private key and sign a document with an old timestamp.

1.8.3 Digital Signature Standard

- The Digital Signature Standard (DSS) makes use of the Secure Hash Algorithm (SHA) and presents a new digital signature technique, the Digital Signature Algorithm (DSA). DSS cannot be used for encryption or key exchange. Fig. 1.8.1 shows the DSS approach.

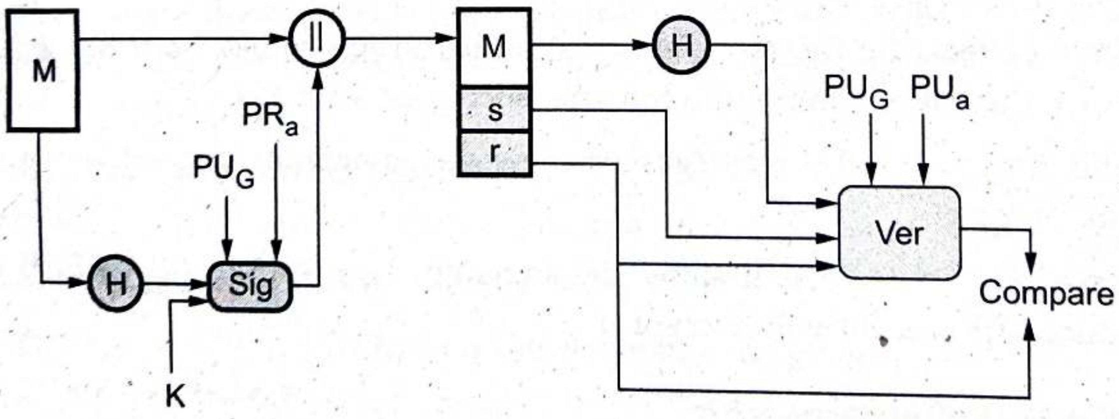


Fig. 1.8.1 DSS approach

- It uses a hash function. The hash code is provided as input to a signature function along with a random number K generated for this particular signature.
- The signature function also depends on the sender's private key (PR_a) and a set of parameters known to a group of communicating principles.
- The result is a signature consisting of two components, labeled s and r.

- At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function.
- Fig. 1.8.2 shows the RSA approach. In the RSA approach, the message to be signed is input to a hash function that producs a secure hash code of fixed length. This hash code is then encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted.

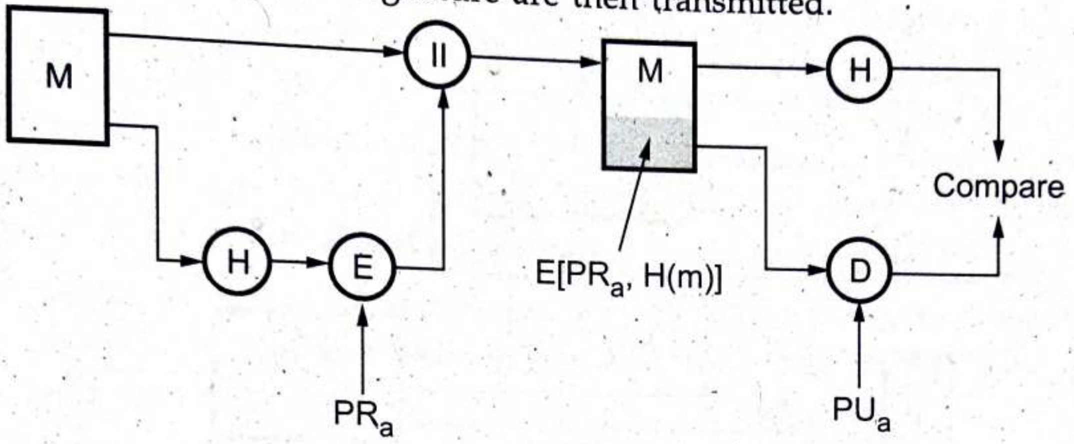


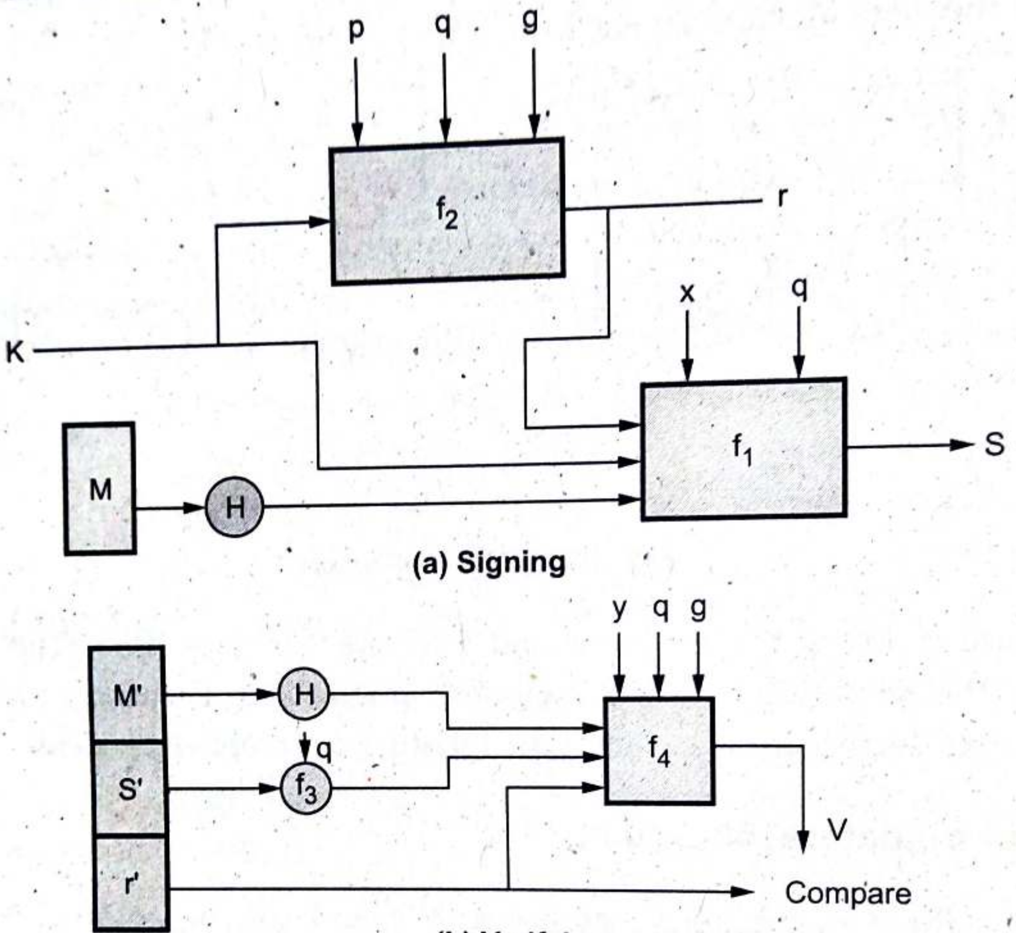
Fig. 1.8.2 RSA approach

- The recipient takes the message and produces a hash code. The recipient also decrypts the signature using the sender's public key. If the calculated hash code matches the decrypted signature, the signature is accepted as valid.

1.8.4 Digital Signature Algorithm

- There are three parameters that are public and can be common to a group of users. **Prime number q** is chosen and it is **160-bit**. A **prime number p** is selected with a length between **512** and **1024 bits** such that q divides (P – 1).
- g is chosen to be of the form $h^{(P-1)/q}$ mod p where h is an integer between 1 and (P – 1).
- With these number, user selects a private key and generate a public key. The private key x must be a number from 1 to (q – 1) and should be chosen randomly or pseudorandomly.
- The public key is calculated from the private key as y = g^x mod p.
- To create a signature, a user calculates two quantities, **rands**, that are functions of
  i) Public key components (p, q, g)
  ii) User's private key (x)
  iii) Hash code of the message H(M)
  iv) An additional integer (K)

- **At the receiving end**, verification is performed. The receiver generates a quantity V that is a function of the public key components, the sender's public key and the hash code of the incoming message. If this quantity matches the r components of the signature, then the signature is validated.
- Fig. 1.8.3 shows the functions of signing and verifying.



(a) Signing

(b) Verifying

Fig. 1.8.3 Signing and verifying

Example 1.8.1 *Consider a banking application that is expected to provide cryptographic functionalities. Assume that this application is running on top of another application wherein the end customers can perform a single task of fund transfer. The application requires cryptographic requirements based on the amount of transfer.*

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

Suggest the security scheme to be adopted in client and server side to accommodate the above requirements and justify your recommendations.

AU : Dec.-19, Marks 15

Solution :

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

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

Review Questions

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

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

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

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

1.9 Two Marks Questions with Answers

Q.1 Distinguish active and passive attack with example. AU : Dec.-22, 20, 16, May-19, 16

Ans. : Difference between passive and active attacks :

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

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

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

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

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

Q.4 Define an attack.

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

Q.5 List some examples of security attacks.

Ans. : 1) Gain unauthorized access to information.

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

3) Enlarge cheater's legitimate license.

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

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

Q.6 What is a passive attack ?

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

Q.7 What is an active attack ?

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

Q.8 Categorize passive and active attack. **AU : Dec.-17**

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

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

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

Q.9 What are the aspects of information security ?

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

Q.10 What is a threat ? List their types. **AU : May-18**

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

Q.11 What is encipherment ?

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

Q.12 Define symmetric encryption. **AU : Dec.-15**

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

Q.13 What are the essential ingradients of a symmetric cipher ?

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

Q.14 What are the two basic functions used in the encryption algorithm ?

AU : Dec.-14

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

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

Q.15 How many keys are required for two people to communicate via a cipher ?

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

Q.16 Why is asymmetric cryptography bad for huge data ? Specify the reason.

AU : May-18

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

Q.17 What are the two general approaches to attacking a cipher ?

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

1. **Cryptanalysis :** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some samples plaintext-cipher text pairs.
2. **Brute-force attack :** The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained.

Q.18 Distinguish between attack and Threat.

Ans. :

AU : Dec.-18

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

Q.19 Differentiate MAC and Hash function. AU : Dec.-22

Ans. : The major difference between hash and MAC is that MAC uses secret key during the compression. Unlike a MAC, a hash code does not use a key but is a function only of the input message.

Q.20 What is MAC ? Mention the requirement of MAC. AU : Dec.-20

Ans. : An alternative authentication technique involves the use of a small fixed size block of data, known as a cryptographic checksum or MAC that is appended to the message.

Q.21 What is a Hash in cryptography ? AU : May-18

Ans. : A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h(that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

Q.22 What is a message authentication code ?

Ans. : An alternative authentication technique involves the use of a small fixed size block of data, known as a cryptographic checksum or MAC that is appended to the message.

Q.23 What is the difference between a message authentication code and a one-way hash function ?

Ans. : The difference between a MAC and a one-way hash function is that unlike a MAC, a hash code does not use a key but is a function only of the input message.

Q.24 Is it necessary to recover the secret key in order to attack a MAC algorithm ?

Ans. : A number of keys will produce the correct MAC and the opponent has no way of knowing which the correct key is. On an average $2^{(n - k)}$ keys produce a match. Therefore attacks do not require the discovery of the key.

Q.25 What is the function of a compression function in a hash function ?

Ans. : The hash function involves repeated use of a compression function. The motivation is that if the compression function is collision resistant, then the hash function is also collision resistant function. So a secure hash function can be produced.

Q.26 What is the use of digital signature ?

Ans. : Data appended to, or a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery.

Q.27 What is a birthday attack ? AU : May-11, 14, IT

Ans. : A birthday attack is a name used to refer to class of brute-force attacks. It gets its name from the surprising result that the probability that two or more people in a group of 23 share the same birthday is greater than ½; such a result is called a birthday paradox.

Q.28 What is the utility of a detached signature ?

Ans. : A detached signature may be stored and transmitted separately from the message it signs. This is useful in several contexts. A user may wish to maintain a separate signature log of all messages sent or received. A detached signature of an executable program can detect subsequent virus infection. Finally detached signature can be used when more than one party must sign a document, such as legal contract.

Q.29 What is digital signature ? **AU : May-11,15, IT**

Ans. : Digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.

Q.30 What is one-way property ? **AU : Dec.-12, CSE/IT**

Ans. : A function that maps an arbitrary length message to a fixed length message digest is a one-way hash function if it is a one-way function.

Q.31 What are the two approaches of digital signature ? **AU : Dec.-12, CSE/IT**

Ans. : Two approaches of digital signature are RSA approach and DSS approaches.

Q.32 How is the security of MAC function expressed ? **AU : Dec-17**

Ans. : Security of MAC functions :

- The security of any HMAC function based on the cryptographic strength of the underlying hash function.
- The security of a MAC function expressed in terms of the probability of successful forgery with a given amount of time spent by the forger and a given number of message-MAC pairs created with the same key.

❑❑❑