Index

- CRYPTOGRAPHY
  - 01_Origin of Cryptography
    - Origin of Cryptography
      - History of Cryptography
      - Hieroglyph – The Oldest Cryptographic Technique
      - Steganography
      - Evolution of Cryptography
  - 02_Modern cryptography
    - What is cryptology?
    - Cryptology is the study of cryptosystems
    - What are the two branches of cryptology?
      - What is Cryptography?
      - What is Cryptanalysis?
    - Security Services of Cryptography
      - Explain Confidentiality
      - Explain Data Integrity
      - Explain Authentication
      - Explain Non-repudiation
    - What are cryptography primitives?
    - Characteristics of Modern Cryptography
  - 03_Cryptosystems
    - Cryptographic Services
      - Explain Confidentiality
      - Explain Data Integrity
      - Explain Authentication
      - Explain Non-repudiation
    - Cryptanalyst
    - Cryptographic Primitives
    - Cryptosystems
      - Symmetric Key Encryption
      - Asymmetric Key Encryption
    - Kerckhoff's Principle for Cryptosystem
  - 04_Attacks on Cryptosystems
    - Passive Attacks
    - Active Attacks
    - Assumptions of Attacker
    - Environment around Cryptosystem

- Details of the Encryption Scheme
- Availability of Ciphertext
- Availability of Plaintext and Ciphertext
- Cryptographic Attacks
- Practicality of Attacks

▸ 05_Traditiaonal_ciphers

▸ 06_Modern Symmetric Key Encryption

▸ 07_Block Cipher

▸ 08_Feistel Block Cipher

- What is the encryption process of Fiestel Cipher?
  - Encryption Process
  - Decryption Process
  - Number of Rounds

▸ 09_DES

- Data Encryption Standard
  - Initial and Final Permutation
  - Round Function
  - Key Generation
  - DES Analysis
- What are the two properties of DES ?

▸ 10_Triple DES

- Triple DES
- Explain 3-KEY Triple DES
  - 3-KEY Triple DES

▸ 11_Advanced Encryption Standard

- Operation of AES
- Encryption Process
- Byte Substitution (SubBytes)
- Shiftrows
- MixColumns
- Addroundkey
- Explain Decryption Process of AES
  - Decryption Process
  - AES Analysis

▸ 12_Block Cipher Modes of Operation

- Electronic Code Book (ECB) Mode
- Operation
- Analysis of ECB Mode
- Cipher Block Chaining (CBC) Mode
- Operation
- Analysis of CBC Mode
- Cipher Feedback (CFB) Mode

- Operation
- Analysis of CFB Mode
- Output Feedback (OFB) Mode
- Counter (CTR) Mode
- Operation
- Analysis of Counter Mode

▸ 14_Public Key Encryption
- Public Key Cryptography
- ElGamal Cryptosystem
- Elliptic Curve Cryptography (ECC)
- RSA Cryptosystem
- Generation of RSA Key Pair
- Example
- Encryption and Decryption
- RSA Encryption
- RSA Decryption
- RSA Analysis
- ElGamal Cryptosystem
- Generation of ElGamal Key Pair
- Encryption and Decryption
- ElGamal Encryption
- ElGamal Decryption
- ElGamal Analysis
- Elliptic Curve Cryptography (ECC)
- RSA and ElGamal Schemes – A Comparison

▸ 18_Cryptography Digital signatures
- Model of Digital Signature
- Importance of Digital Signature
- Encryption with Digital Signature

▸ 19_Public Key Infrastructure
- Key Management
- Public Key Infrastructure (PKI)
- Digital Certificate
- Certifying Authority (CA)
- Key Functions of CA
- Classes of Certificates
- Registration Authority (RA)
- Certificate Management System (CMS)
- Private Key Tokens
- Hierarchy of CA

▸ 20_Cryptography Benefits & Drawbacks
- Cryptography – Benefits

- Cryptography – Drawbacks
- Future of Cryptography

▸ 15_Data Integrity in Cryptography
- Threats to Data Integrity
- Explain Passive threats
- Passive Threats
- Explain Active Threats

▸ 16_Cryptography Hash functions
- Features of Hash Functions
-
- What are the properties of Hash functions?
  - Design of Hashing Algorithms
- What are the popular hash functions?
  - Popular Hash Functions
  - Message Digest (MD)
  - Secure Hash Function (SHA)
  - RIPEMD
  - Whirlpool

▸ applications of hash functions
- Password Storage
- Data Integrity Check

▸ Authentication

▸ 17_Message Authentication
- Message Authentication Code (MAC)
- What are the Limitations of MAC?

▸ ADDITIONAL_READINGS

▸ KECCSAC

CRYPTOGRAPHY

1. Origin of Cryptography
2. Modern Cryptography
3. Cryptosystems
   1) Cryptographic services
   2) Cryptanalyst
   3) Cryptographic Primitives
   4) Cryptosystems

****

01_Origin of Cryptography

Origin of Cryptography ⚓

Human beings from ages had two inherent needs – (a) to communicate and share information and (b) to communicate selectively. These two needs gave rise to the art of coding the messages in such a way that only the intended people could have access to the information. Unauthorized people could not extract any information, even if the scrambled messages fell in their hands.

What is cryptography?

• Cryptography is the art and science of making a cryptosystem that is capable of providing information security.

• *The art and science of concealing the messages to introduce secrecy in information security is recognized as cryptography.*

• Cryptography deals with the actual securing of digital data. It refers to the design of mechanisms based on mathematical algorithms that provide fundamental information security services.

The word 'cryptography' was coined by combining two Greek words, 'Krypto' meaning hidden and 'graphene' meaning writing.

History of Cryptography ⚓

The art of cryptography is considered to be born along with the art of writing. As civilizations evolved, human beings got organized in tribes, groups, and kingdoms. This led to the emergence of ideas such as power, battles, supremacy, and politics. These ideas further fueled the natural need of people to communicate secretly with selective recipients which in turn ensured the continuous evolution of cryptography as well.
The roots of cryptography are found in Roman and Egyptian civilizations.

Hieroglyph – The Oldest Cryptographic Technique ⚓

The first known evidence of cryptography can be traced to the use of 'hieroglyph'. Some 4000 years ago, the Egyptians used to communicate by messages written in hieroglyphs. This code was the secret known only to the scribes who used to transmit messages on behalf of the kings. One such hieroglyph is shown below.

What are simple mono-alphabetic substitution ciphers?

Later, the scholars moved on to using simple mono-alphabetic substitution ciphers during 500 to 600 BC.

◇ Replacing alphabets of messages with other alphabets with some secret rule are known as mono-alphabetic substitution ciphers.

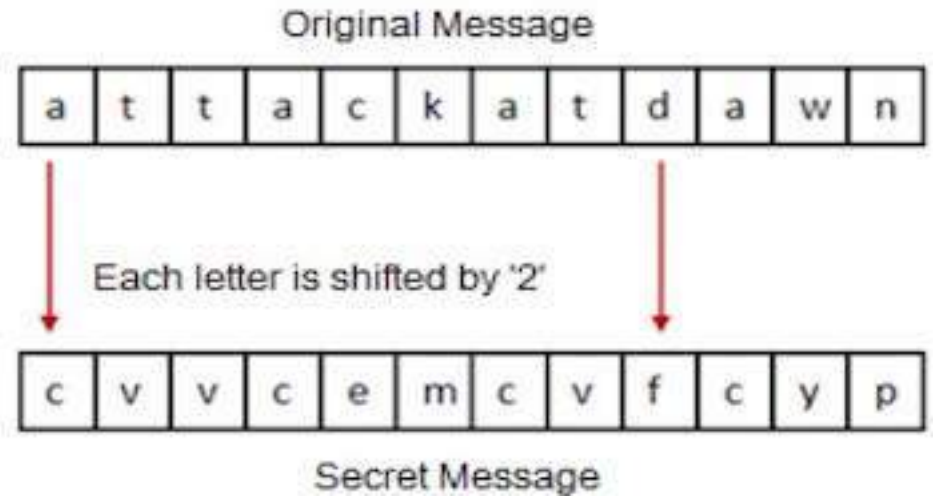
◇ This rule becomes a key to retrieve the original message back from the garbled/cipher message.

How does the Caesar Shift cipher operate?

The earlier Roman method of cryptography, popularly known as the Caesar Shift Cipher,

◇ Caesar Shift Ciphers relies on shifting the letters of a message by an agreed number (three was a common choice),

◇ the recipient of this message would then shift the letters back by the same number and obtain the original message.



Steganography ⚓

What is Steganography?

◇ In the Steganography method, people not only want to protect the secrecy of an information by concealing it, but they also want to make sure any unauthorized person gets no evidence that the information even exists.

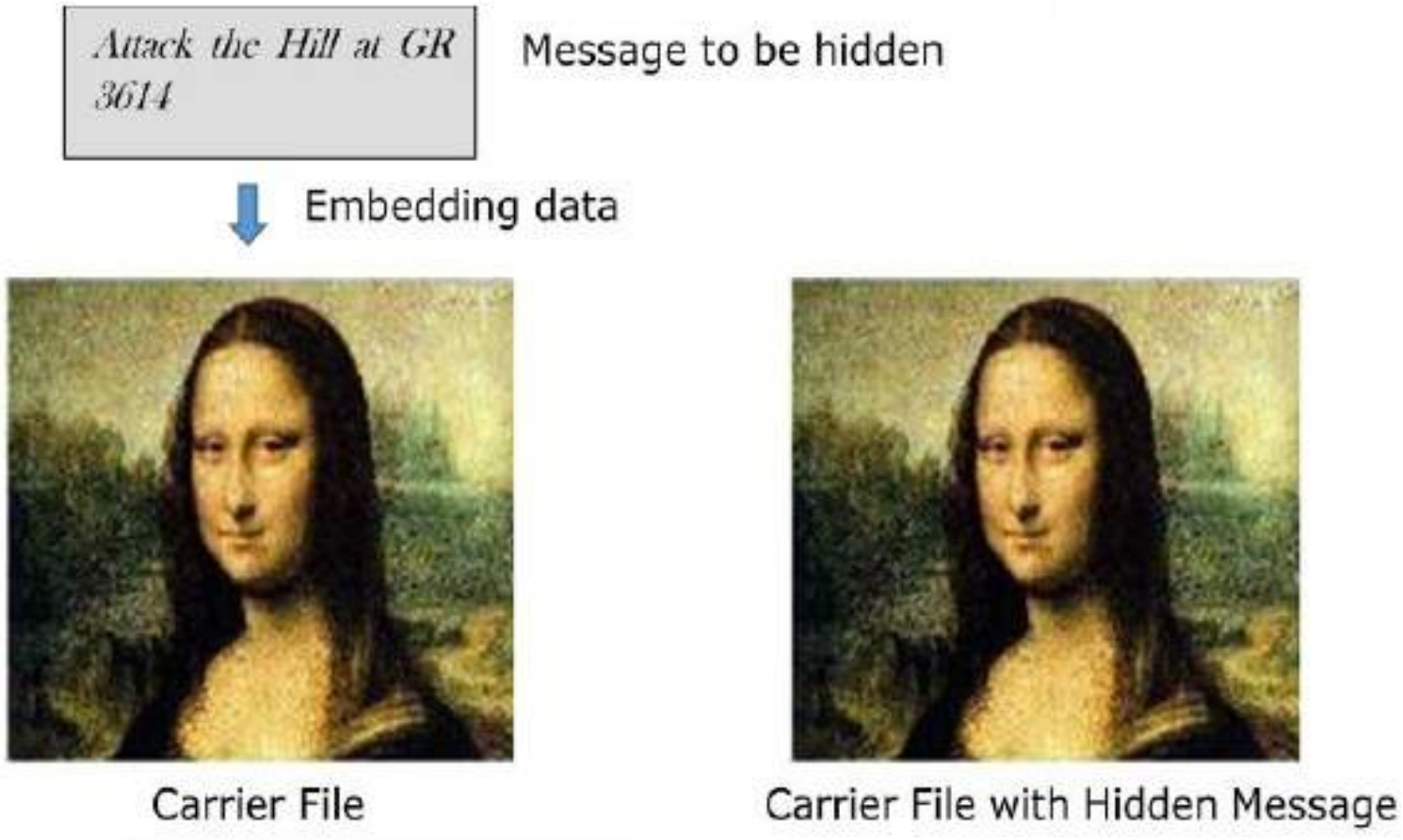
◇ For example, invisible watermarking.

◇ Steganography is similar to cryptography but adds another dimension to Cryptography.

What is the advantage of Steganography compared to cryptography?

◇ In steganography, an unintended recipient or an intruder is unaware of the fact that observed data contains hidden information.

◇ In cryptography, an intruder is normally aware that data is being communicated, because they can see the coded/scrambled message.



Evolution of Cryptography ⚓

During and after the European Renaissance, various Italian and Papal states led the rapid

proliferation of cryptographic techniques. Various analysis and attack techniques were researched in this era to break the secret codes.
◇ Improved coding techniques such as Vigenere Coding came into existence in the 15th century, which offered moving letters in the message with a number of variable places instead of moving them the same number of places.

◇ Only after the 19th century, cryptography evolved from the ad hoc approaches to encryption to the more sophisticated art and science of information security.
◇ In the early 20th century, the invention of mechanical and electromechanical machines, such as the Enigma rotor machine, provided more advanced and efficient means of coding the information.
◇ During the period of World War II, both cryptography and cryptanalysis became excessively mathematical.

With the advances taking place in this field, government organizations, military units, and some corporate houses started adopting the applications of cryptography. They used cryptography to guard their secrets from others. Now, the arrival of computers and the Internet has brought effective cryptography within the reach of common people.

02_Modern cryptography

What is cryptology?

Cryptology is the study of cryptosystems

What are the two branches of cryptology?

1. Cryptography

2. Cryptanalysis

What is Cryptography?

• Cryptography is the art and science of making a cryptosystem that is capable of providing information security.

• Cryptography deals with the actual securing of digital data. It refers to the design of mechanisms based on mathematical algorithms that provide fundamental information security services.

• You can think of cryptography as the establishment of a large toolkit containing different techniques in security applications.

What is Cryptanalysis?

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

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

Security Services of Cryptography

What are the 4 fundamental information security services?

1. Confidentiality

2. Data Integrity
3. Authentication
4. Non-repudiation

Explain Confidentiality ⚓

◇ Confidentiality is a security service that keeps the information secure from an unauthorized person.
◇ Confidentiality can be achieved through numerous means starting from physical securing to the use of mathematical algorithms for data encryption.
◇ Confidentiality is the fundamental security service provided by cryptography.
◇ It is sometimes referred to as privacy or secrecy.

Explain Data Integrity ⚓

◇ Data integrity is a security service that deals with identifying any alteration to the data.
◇ The data may get modified by an unauthorized entity intentionally or accidently.
◇ Data integrity cannot prevent the alteration of data, but provides a means for detecting whether data has been manipulated in an unauthorized manner.
◇ Data Integrity service confirms whether data is intact or not since it was last created, transmitted, or stored by an authorized user.

Explain Authentication ⚓

◇ Authentication service provides the identification of the originator.
◇ It confirms to the receiver that the data received has been sent only by an identified and verified sender.

What are the two variants of authentication?

1. Message authentication
2. Entity authentication.

◇ Message authentication identifies the originator of the message without any regard to the router or system that has sent the message.
◇ Entity authentication is assurance that data has been received from a specific entity, say a particular website.
◇ Apart from the originator, authentication may also provide assurance about other

parameters related to data such as the date and time of creation/transmission.

Explain Non-repudiation

◇ Non-repudiation is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action.
◇ It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party.

Non-repudiation is a property that is most desirable in situations where there are chances of a dispute over the exchange of data. For example, once an order is placed electronically, a purchaser cannot deny the purchase order, if non-repudiation service was enabled in this transaction.

What are cryptography primitives?

Cryptography primitives are tools and techniques that can be selectively used to provide a set of desired security services.

What are the 4 cryptographic primitives?

1. Encryption
2. Hash functions
3. Message Authentication codes (MAC)
4. Digital Signatures

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

Which cryptographic primitive provides which security service?

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

Note – Cryptographic primitives are intricately related and they are often combined to achieve a set of desired security services from a cryptosystem.

Characteristics of Modern Cryptography

Modern cryptography is the cornerstone of computer and communications security. Its foundation is based on various concepts of mathematics such as number theory, computational-complexity theory, and probability theory.
There are three major characteristics that separate modern cryptography from the classical approach.

Classic Cryptography
It manipulates traditional characters, i.e., letters and digits directly.
It is mainly based on 'security through obscurity'. The techniques employed for coding were ke secret and only the parties involved in communication knew about them.
It requires the entire cryptosystem for communicating confidentially.

03_Cryptosystems

01_Cryptography - Basics

What is Cryptography?

• Cryptography is the art and science of making a cryptosystem that is capable of providing information security.

Where is cryptography used?

• Securing data on storage and during communication from adversaries.

What is a Cryptosystem or Cipher System?

• A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services.

A cryptosystem is also referred to as a cipher system.

What are the Components of a Cryptosystem?

1. Plaintext
2. Ciphertext
3. Encryption Key
4. Decryption Key
5. Encryption program
6. Decryption program
7. key Space.

What is a Plaintext?

• It is the data to be protected during storage or transmission.

What is a Ciphertext data?

• It is the scrambled version of the plaintext produced by the encryption algorithm using a specific encryption key.

◇ The ciphertext is not guarded. (guarded)

◇ It flows on public channels. It can be intercepted or compromised by anyone who has access to the communication channel.

What is an Encryption Algorithm.?

◇ It is a mathematical process that produces a ciphertext for any given plaintext using an encryption key.

How to create a cipher text?

◇ With the help of a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.

What is a Decryption Algorithm?

◇ It is a mathematical process that produces a unique plaintext for any given ciphertext and decryption key.
◇ It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext.
◇ The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.

What is an Encryption Key?

◇ It is a value that is known to the sender/owner.
◇ The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.

What is a Decryption Key?

◇ It is a value that is known to the receiver.
◇ The decryption key is related to the encryption key, but is not always identical to it.
◇ The receiver inputs the decryption key into the decryption algorithm along with the ciphertext in order to compute the plaintext.

What is a Key Space?

◇ For a given cryptosystem, a collection of all possible decryption keys is called a key space.

Who is an Interceptor or Attacker?

◇ An interceptor (an attacker) is an unauthorized entity who attempts to determine the plaintext.

◇ He can see the ciphertext and may know the decryption algorithm. He, however, must never know the decryption key.

Cryptographic Services

Cryptographic Services

What are the security services provided by Cryptography?

1. Confidentiality
2. Data Integrity
3. Authentication
4. Non-repudiation

Explain Confidentiality ⚓

◇ Confidentiality is a security service that keeps the information secure from an unauthorized person.

◇ Confidentiality can be achieved through numerous means starting from physical securing to the use of mathematical algorithms for data encryption.

◇ Confidentiality is the fundamental security service provided by cryptography.

◇ It is sometimes referred to as privacy or secrecy.

Explain Data Integrity ⚓

◇ Data integrity is a security service that deals with identifying any alteration to the data.

◇ The data may get modified by an unauthorized entity intentionally or accidently

◇ Data Integrity service confirms whether data is intact or not since it was last created, transmitted, or stored by an authorized user.

◇ Data integrity cannot prevent the alteration of data, but provides a means for detecting whether data has been manipulated in an unauthorized manner.

Explain Authentication ⚓

◇ Authentication service provides the identification of the originator.

◇ It confirms to the receiver that the data received has been sent only by an identified and verified sender.

What are the two variants of authentication?

Authentication service has two variants −

1. Message authentication
2. Entity authentication.

◇ Message authentication identifies the originator of the message without any regard to the router or system that has sent the message.

◇ Entity authentication is assurance that data has been received from a specific entity, say a particular website.

◇ Apart from the originator, authentication may also provide assurance about other

parameters related to data such as the date and time of creation/transmission.

Explain Non-repudiation ⚓

◇ Non-repudiation is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action.
◇ It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party.

Non-repudiation is a property that is most desirable in situations where there are chances of a dispute over the exchange of data. For example, once an order is placed electronically, a purchaser cannot deny the purchase order, if non-repudiation service was enabled in this transaction.

Cryptanalyst

Cryptanalyst

What is Cryptanalysis?

◇ The art and science of breaking the ciphertext is known as cryptanalysis.

Cryptanalysis is the sister branch of cryptography and they both co-exist.

◇ The cryptographic process results in the ciphertext for transmission or storage.
◇ It involves the study of cryptographic mechanisms with the intention to break them.
◇ Cryptography concerns the design of cryptosystems, while Cryptanalysis studies the breaking of cryptosystems.

Cryptographic Primitives

What are Cryptographic Primitives?

Cryptographic primitives are the tools and techniques in Cryptography that can be selectively used to provide a set of desired security services

List the 4 cryptographic primitives?

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

The following table shows the primitives that can achieve a particular security service on their own.
https://www.tutorialspoint.com/cryptography/modern_cryptography.htm

Which primitive can achieve which service?

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

https://www.tutorialspoint.com/cryptography/cryptosystems.htm

Let us discuss a simple model of a cryptosystem that provides confidentiality to the information being transmitted
https://www.tutorialspoint.com/cryptography/images/cryptosystem.jpg

The illustration shows a sender who wants to transfer some sensitive data to a receiver in such a way that any party intercepting or eavesdropping on the communication channel cannot extract the data.

The objective of this simple cryptosystem is that at the end of the process, only the sender and the receiver will know the plaintext.

Cryptosystems

What are the two types of Cryptosystems?

1. Symmetric Key Encryption
2. Asymmetric Key Encryption

How are cryptosystems classified?

◇ Based on the manner in which encryption-decryption is carried out, the cryptosystems are classified as

1. Symmetric Cryptosystems
2. Asymmetric Key Cryptosystems

What are the differences between Symmetric Key Encryption and Asymmetric Key Encryption?

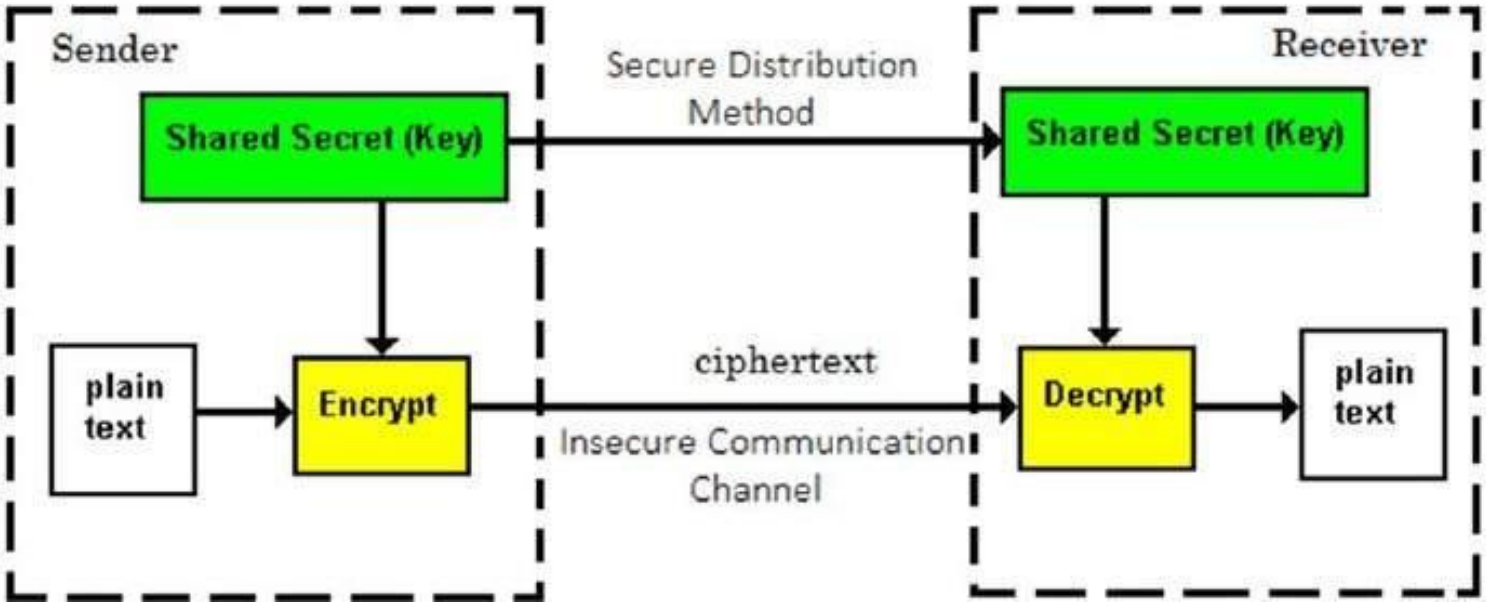
◇ The main difference between these cryptosystems is the relationship between the encryption and the decryption key.
◇ Logically, in any cryptosystem, both the keys are closely associated.
◇ It is practically impossible to decrypt the ciphertext with the key that is unrelated to the encryption key.

Symmetric Key Encryption

Explain Symmetric Key Encryption

◇ The encryption process where the same keys are used for encrypting and decrypting the information is known as Symmetric Key Encryption.
◇ The study of symmetric cryptosystems is referred to as symmetric cryptography.
◇ Symmetric cryptosystems are also sometimes referred to as secret key cryptosystems.
◇ A few well-known examples of symmetric key encryption methods are
◇ Digital Encryption Standard (DES),
◇ Triple-DES (3DES),
◇ IDEA, and
◇ BLOWFISH

https://www.tutorialspoint.com/cryptography/images/symmetric_key_encryption.jpg



Prior to 1970, all cryptosystems employed symmetric key encryption. Even today, its relevance is very high and it is being used extensively in many cryptosystems. It is very unlikely that this encryption will fade away, as it has certain advantages over asymmetric key encryption.

What are the features of symmetric Key encryption?

◇ Persons using symmetric key encryption must share a common key prior to exchange of information.
◇ Keys are recommended to be changed regularly to prevent any attack on the system.
◇ A robust mechanism needs to exist to exchange the key between the communicating

parties. As keys are required to be changed regularly, this mechanism becomes expensive and cumbersome
◇ In a group of n people, to enable two-party communication between any two persons, the number of keys required for the group is n × (n – 1)/2.
◇ Length of Key (number of bits) in this encryption is smaller and hence, process of encryption-decryption is faster than asymmetric key encryption
◇ Processing power of a computer system required to run a symmetric algorithm is less.

What is the Challenge of Symmetric Key Cryptosystem?

1. Key establishment
2. Trust Issues.

◇ Key establishment -

• Before any communication, both the sender and the receiver need to agree on a secret symmetric key. It requires a secure key establishment mechanism in place

◇ Trust Issue -

• Since the sender and the receiver use the same symmetric key, there is an implicit requirement that the sender and the receiver 'trust' each other.
• For example, it may happen that the receiver has lost the key to an attacker and the sender is not informed

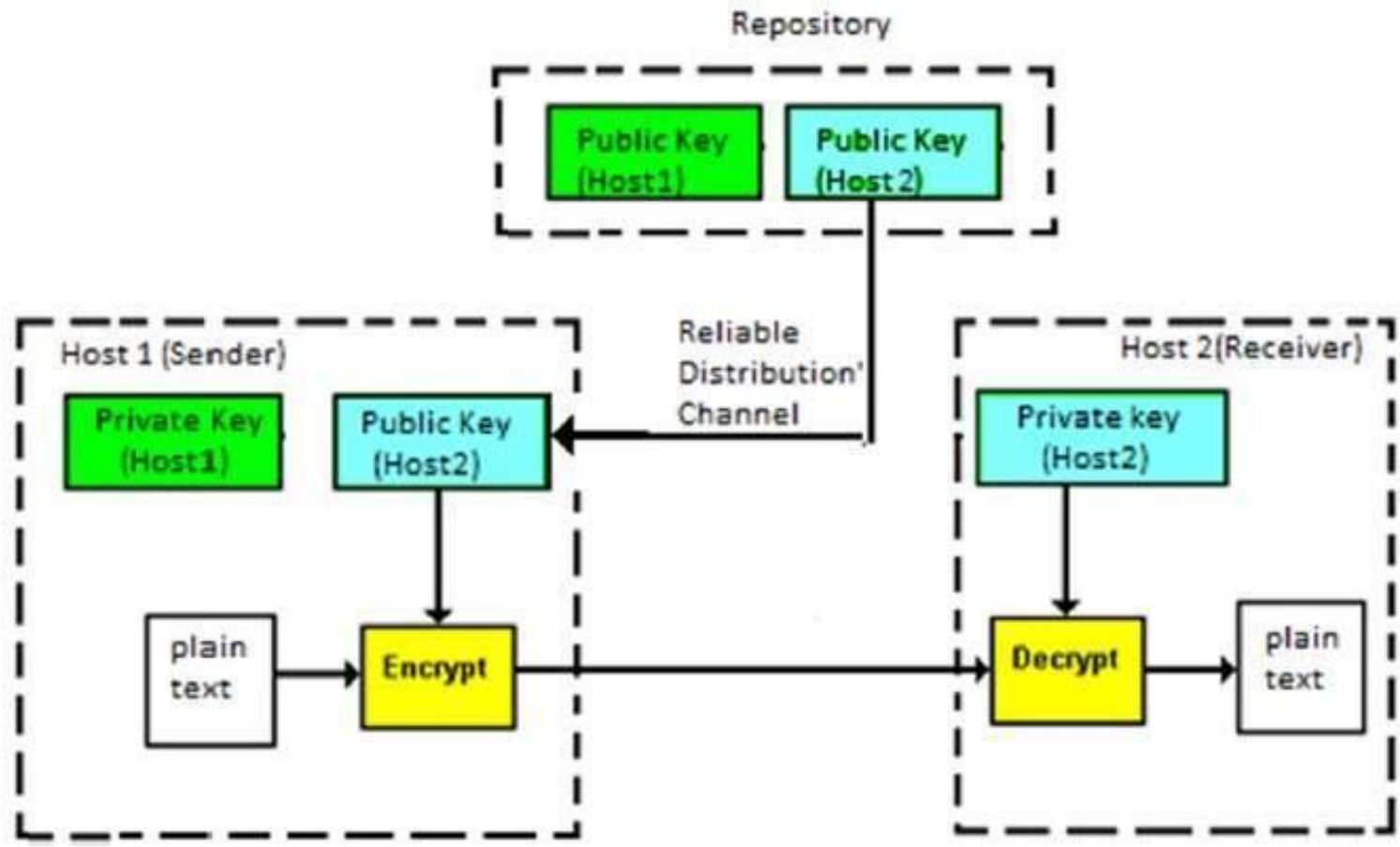
These two challenges are highly restraining for modern day communication.
◇ Today, people need to exchange information with non-familiar and non-trusted parties. For example, communication between online sellers and customers.
◇ These limitations of symmetric key encryption gave rise to asymmetric key encryption schemes.

Asymmetric Key Encryption

What is Asymmetric Key Encryption

◇ The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption.
◇ Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible.

The process is depicted in the following illustration
https://www.tutorialspoint.com/cryptography/images/asymmetric_key_encryption.jpg



Asymmetric Key Encryption was invented in the 20th century to overcome the necessity of pre-shared secret keys between communicating persons.

What are the features of asymmetric encryption schemes?

The salient features of this encryption scheme are as follows -
◇ Every user in this system needs to have a pair of dissimilar keys, private key and public key.
◇ These keys are mathematically related - when one key is used for encryption, the other can decrypt the ciphertext back to the original plaintext.
◇ It requires putting the public key in a public repository and the private key as a well-

guarded secret. Hence, this scheme of encryption is also called Public Key Encryption.
◇ Though public and private keys of the user are related, it is computationally not feasible to find one from another. This is a strength of this scheme.
◇ When Host1 needs to send data to Host2, he obtains the public key of Host2 from repository, encrypts the data, and transmits
◇ Host2 uses his private key to extract the plaintext.
◇ Length of Keys (number of bits) in this encryption is large and hence, the process of encryption-decryption is slower than symmetric key encryption.
◇ Processing power of a computer system required to run an asymmetric algorithm is higher.

Symmetric cryptosystems are a natural concept. In contrast, public-key cryptosystems are quite difficult to comprehend.

You may think, how can the encryption key and the decryption key be 'related', and yet it is impossible to determine the decryption key from the encryption key? The answer lies in the mathematical concepts

It is possible to design a cryptosystem whose keys have this property. The concept of public-key cryptography is relatively new. There are fewer public-key algorithms known than symmetric algorithms.

What are the challenges of Public Key Cryptosystem (asymmetric key systesm)
◇ The user needs to trust that the public key that he is using in communications with a person really is the public key of that person and has not been spoofed by a malicious third party.

◇ This is usually accomplished through a Public Key Infrastructure (PKI) consisting of a trusted third party.
◇ The third party securely manages and attests to the authenticity of public keys.
◇ When the third party is requested to provide the public key for any communicating person X, they are trusted to provide the correct public key.

1. The third party satisfies itself about user identity by the process of attestation, notarization, or some other process - that X is the one and only, or globally unique, X.
2. The most common method of making the verified public keys available is to embed them in a certificate which is digitally signed by the trusted third party.

Relation between Encryption Schemes . table .
Symmetric Cryptosystems Public Key Cryptosystems
Relation between Keys Same Different, but mathematically related
Encryption Key Symmetric Public
Decryption Key Symmetric Private

◇ Due to the advantages and disadvantages of both the systems, symmetric key and public-key cryptosystems are often used together in the practical information security systems.

Kerckhoff's Principle for Cryptosystem

Kerckhoff's Principle for Cryptosystem

In the 19th century, a Dutch cryptographer A. Kerckhoff furnished the requirements of a good cryptosystem. Kerckhoff stated that a cryptographic system should be secure even if everything about the system, except the key, is public knowledge. The six design principles defined by Kerckhoff for cryptosystem are -

What are the 6 principles of cryptosystem by kerckhoff?

1. The cryptosystem should be unbreakable practically, if not mathematically.

2. Falling of the cryptosystem in the hands of an intruder should not lead to any compromise of the system, preventing any inconvenience to the user.

3. The key should be easily communicable, memorable, and changeable

4. The ciphertext should be transmissible by telegraph, an unsecure channel.

5. The encryption apparatus and documents should be portable and operable by a single person.

6. it is necessary that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

The second rule is currently known as the Kerckhoff principle. It is applied in virtually all the contemporary encryption algorithms such as DES, AES, etc. These public algorithms are considered to be thoroughly secure. The security of the encrypted message depends solely on the security of the secret encryption key.

Keeping the algorithms secret may act as a significant barrier to cryptanalysis. However, keeping the algorithms secret is possible only when they are used in a strictly limited circle.

In the modern era, cryptography needs to cater to users who are connected to the Internet. In such cases, using a secret algorithm is not feasible, hence Kerckhoff principles became essential guidelines for designing algorithms in modern cryptography.

04_Attacks on Cryptosystems

Attacks On Cryptosystems
https://www.tutorialspoint.com/cryptography/attacks_on_cryptosystems.htm

Jumping up.

In the present era, not only business but almost all the aspects of human life are driven by information. Hence, it has become imperative to protect useful information from malicious activities such as attacks. Let us consider the types of attacks to which information is typically subjected to.

How are attacks categorized?
• Attacks are typically categorized based on the action performed by the attacker. An attack, thus, can be

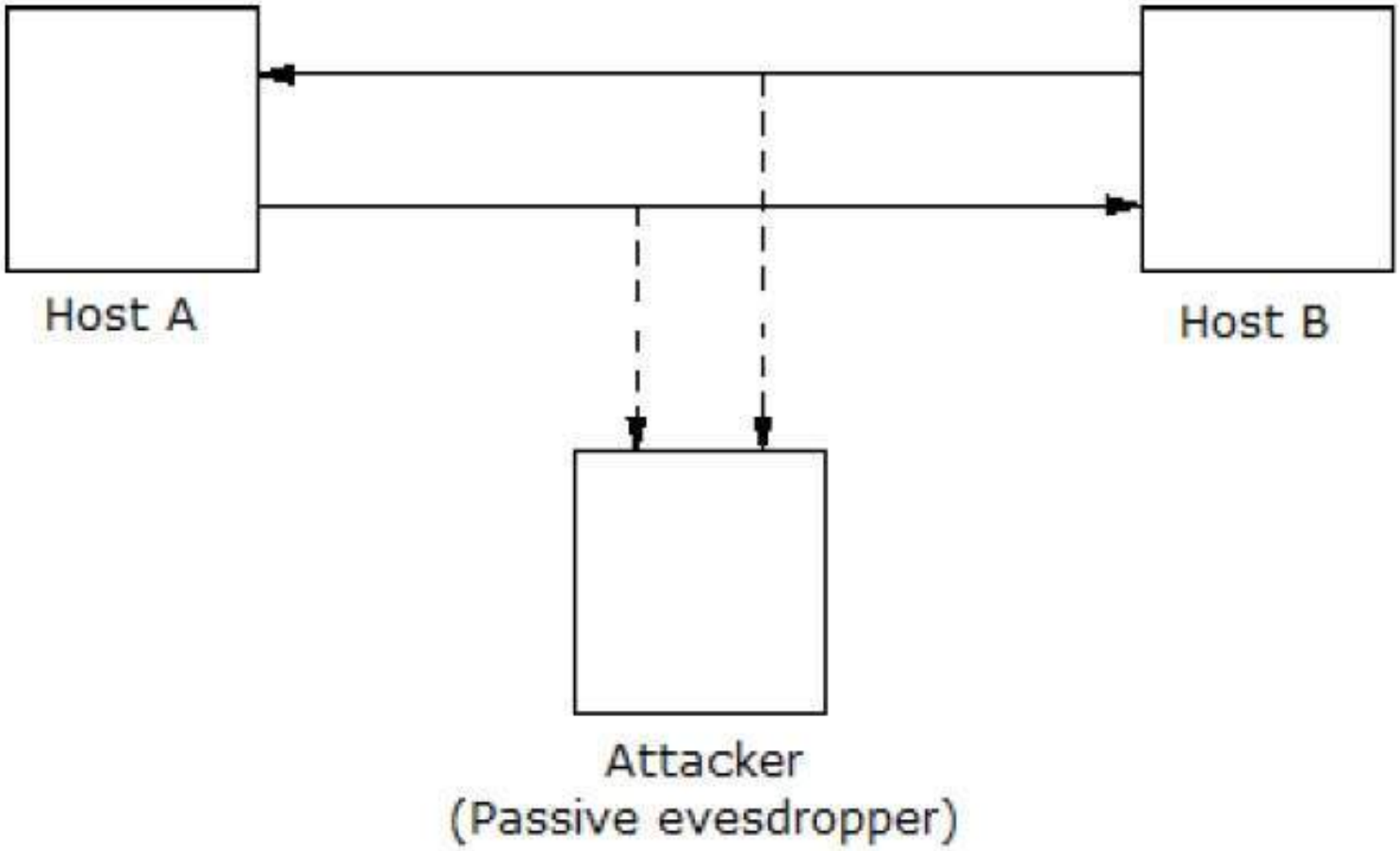
1. passive.
2. active.

Passive Attacks ⚓

Explain Passive Attacks:
◇ The main goal of a passive attack is to obtain unauthorized access to the information.
◇ A passive attack is often seen as *stealing* information.
◇ For example, actions such as intercepting and eavesdropping on the communication channel can be regarded as passive attack.

◇

◇ These actions are passive in nature, as they neither affect information nor disrupt the communication channel.
◇ The only difference in stealing physical goods and stealing information is that theft of data still leaves the owner in possession of that data. Passive information attack is thus more dangerous than stealing of goods, as information theft may go unnoticed by the owner.



Active Attacks ⚓

Explain Active Attacks:

1. An active attack involves changing the information in some way by conducting some process on the information.

2. For example, Modifying the information in an unauthorized manner.

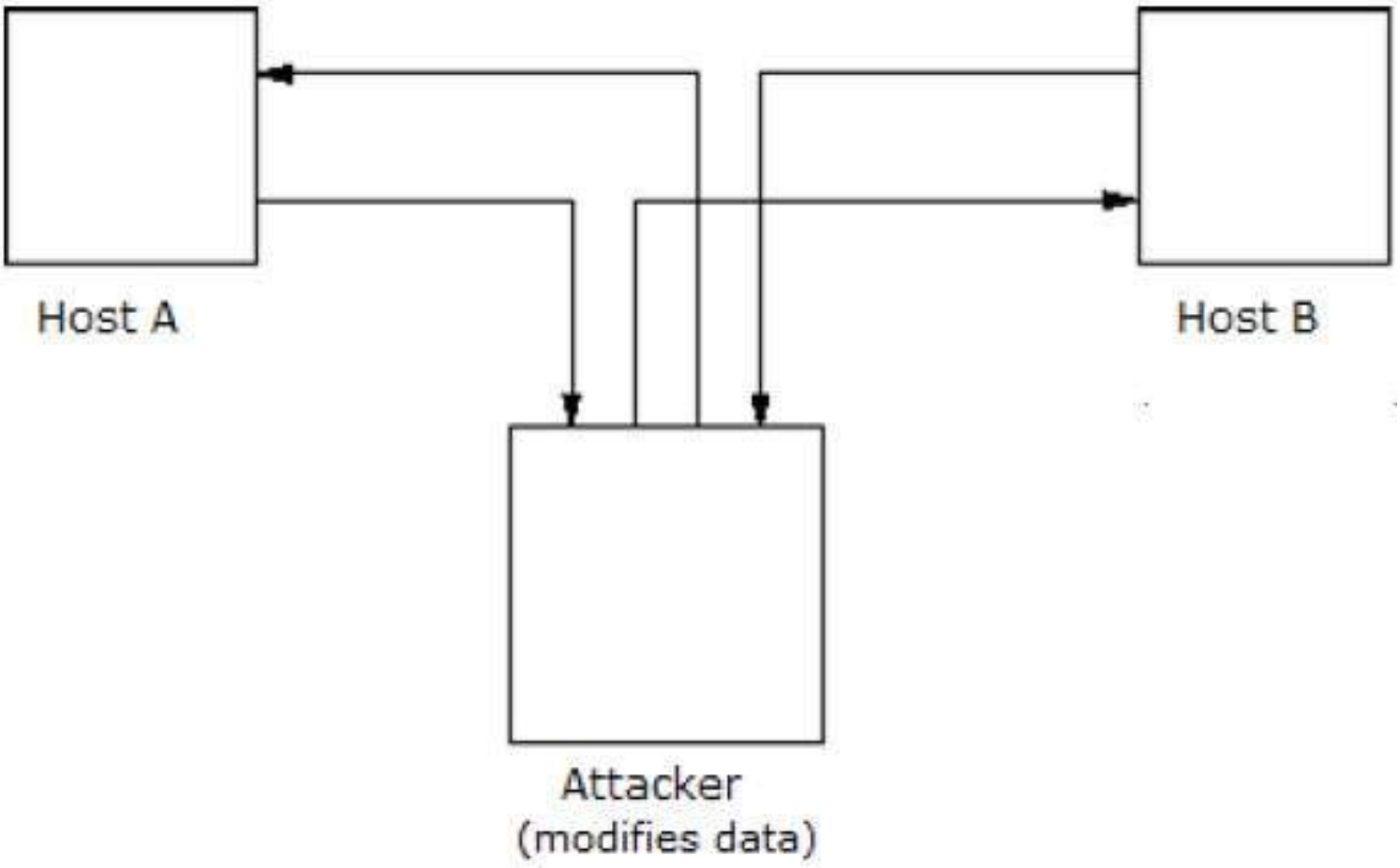
3.

1. Initiating unintended or unauthorized transmission of information.

2. Alteration of authentication data such as originator name or timestamp associated with information

3. Unauthorized deletion of data.

4. Denial of access to information for legitimate users (denial of service).



Cryptography provides many tools and techniques for implementing cryptosystems capable of preventing most of the attacks described above.

Assumptions of Attacker ⚓

Let us see the prevailing environment around cryptosystems followed by the types of attacks employed to break these systems −

Environment around Cryptosystem ⚓

While considering possible attacks on the cryptosystem, it is necessary to know the cryptosystems environment. The attacker's assumptions and knowledge about the environment decides his capabilities.
In cryptography, the following three assumptions are made about the security environment and attacker's capabilities.

Details of the Encryption Scheme ⚓

The design of a cryptosystem is based on the following two cryptography algorithms −
◇ Public Algorithms − With this option, all the details of the algorithm are in the public

domain, known to everyone.

◇ Proprietary algorithms − The details of the algorithm are only known by the system designers and users.

In case of proprietary algorithms, security is ensured through obscurity. Private algorithms may not be the strongest algorithms as they are developed in-house and may not be extensively investigated for weakness.
Secondly, they allow communication among closed groups only. Hence they are not suitable for modern communication where people communicate with a large number of known or unknown entities. Also, according to Kerckhoff's principle, the algorithm is preferred to be public with the strength of encryption lying in the *key*.
Thus, the first assumption about the security environment is that the encryption algorithm is known to the attacker.

Availability of Ciphertext⚓

We know that once the plaintext is encrypted into ciphertext, it is put on unsecure public channel (say email) for transmission. Thus, the attacker can obviously assume that it has access to the ciphertext generated by the cryptosystem.

Availability of Plaintext and Ciphertext⚓

This assumption is not as obvious as others. However, there may be situations where an attacker can have access to plaintext and corresponding ciphertext. Some such possible circumstances are −
◇ The attacker influences the sender to convert the plaintext of his choice and obtains the ciphertext.

◇ The receiver may divulge the plaintext to the attacker inadvertently. The attacker has access to corresponding ciphertext gathered from open channel.

◇ In a public-key cryptosystem, the encryption key is in open domain and is known to any potential attacker. Using this key, he can generate pairs of corresponding plaintexts and ciphertexts.

Cryptographic Attacks⚓

What is the basic intention of a cryptographic attack?

◇ The basic intention of an attacker is to break a cryptosystem and to find the plaintext from the ciphertext. To obtain the plaintext, the attacker only needs to find out the secret

decryption key, as the algorithm is already in public domain.

Hence, he applies maximum effort towards finding out the secret key used in the cryptosystem. Once the attacker is able to determine the key, the attacked system is considered as *broken* or *compromised*.
Based on the methodology used, attacks on cryptosystems are categorized as follows –
List the different types of cryptographic attacks:
1. Ciphertext Only Attacks (COA)

2. Known Plaintext Attack (KPA)

3. Chosen Plaintext Attack (CPA)

4. Dictionary Attack

5. Brute Force Attack (BFA)

6. Birthday Attack

7. Man in Middle Attack (MIM)

8. Side Channel Attack (SCA)

9. Timing Attacks

10. Power Analysis Attacks

11. Fault analysis Attacks

Explain each one of the cryptographic attacks:
◇ Ciphertext Only Attacks (COA) – In this method, the attacker has access to a set of ciphertext(s). He does not have access to corresponding plaintext. COA is said to be successful when the corresponding plaintext can be determined from a given set of ciphertext. Occasionally, the encryption key can be determined from this attack. Modern cryptosystems are guarded against ciphertext-only attacks.

◇ Known Plaintext Attack (KPA) – In this method, the attacker knows the plaintext for some parts of the ciphertext. The task is to decrypt the rest of the ciphertext using this information. This may be done by determining the key or via some other method. The best example of this attack is *linear cryptanalysis* against block ciphers.

◇ Chosen Plaintext Attack (CPA) − In this method, the attacker has the text of his choice encrypted. So he has the ciphertext-plaintext pair of his choice. This simplifies his task of determining the encryption key. An example of this attack is *differential cryptanalysis* applied against block ciphers as well as hash functions. A popular public key cryptosystem, RSA is also vulnerable to chosen-plaintext attacks.

◇ Dictionary Attack − This attack has many variants, all of which involve compiling a 'dictionary'. In the simplest method of this attack, the attacker builds a dictionary of ciphertexts and corresponding plaintexts that he has learnt over a period of time. In the future, when an attacker gets the ciphertext, he refers the dictionary to find the corresponding plaintext.

◇ Brute Force Attack (BFA) − In this method, the attacker tries to determine the key by attempting all possible keys. If the key is 8 bits long, then the number of possible keys is $2^8 = 256$. The attacker knows the ciphertext and the algorithm, now he attempts all the 256 keys one by one for decryption. The time to complete the attack would be very high if the key is long.

◇ Birthday Attack − This attack is a variant of brute-force technique. It is used against the cryptographic hash function. When students in a class are asked about their birthdays, the answer is one of the possible 365 dates. Let us assume the first student's birthdate is 3rd Aug. Then to find the next student whose birthdate is 3rd Aug, we need to enquire $1.25*\sqrt{365} \approx 25$ students.
Similarly, if the hash function produces 64 bit hash values, the possible hash values are 1.8×10^{19}. By repeatedly evaluating the function for different inputs, the same output is expected to be obtained after about 5.1×10^9 random inputs.
If the attacker is able to find two different inputs that give the same hash value, it is a collision and that hash function is said to be broken.

◇ Man in Middle Attack (MIM) − The targets of this attack are mostly public key cryptosystems where key exchange is involved before communication takes place.

■ Host *A* wants to communicate to host *B*, hence requests public key of *B*.

■ An attacker intercepts this request and sends his public key instead.

■ Thus, whatever host *A* sends to host *B*, the attacker is able to read.

■ In order to maintain communication, the attacker re-encrypts the data after reading with his public key and sends it to B.

■ The attacker sends his public key as *A*'s public key so that *B* takes it as if it is taking it from *A*.

◇ Side Channel Attack (SCA) − This type of attack is not against any particular type of cryptosystem or algorithm. Instead, it is launched to exploit the weakness in physical implementation of the cryptosystem.

◇ Timing Attacks − They exploit the fact that different computations take different times to compute on the processor. By measuring such timings, it is be possible to know about a particular computation the processor is carrying out. For example, if the encryption takes a longer time, it indicates that the secret key is long.

◇ Power Analysis Attacks − These attacks are similar to timing attacks except that the amount of power consumption is used to obtain information about the nature of the underlying computations.

◇ Fault analysis Attacks − In these attacks, errors are induced in the cryptosystem and the attacker studies the resulting output for useful information.

Practicality of Attacks ⚓

The attacks on cryptosystems described here are highly academic, as the majority of them come from the academic community. In fact, many academic attacks involve quite unrealistic assumptions about the environment as well as the capabilities of the attacker. For example, in chosen-ciphertext attack, the attacker requires an impractical number of deliberately chosen plaintext-ciphertext pairs. It may not be practical altogether. Nonetheless, the fact that any attack exists should be a cause of concern, particularly if the attack technique has the potential for improvement.

05_Traditiaonal_ciphers

Traditional Ciphers
In the second chapter, we discussed the fundamentals of modern cryptography. We equated cryptography with a toolkit where various cryptographic techniques are considered as the basic tools.

What is symmetric Key Encryption?
• In Symmetric Key Encryption the key used for encryption and decryption is the same.

In this chapter, we discuss this technique further and its applications to develop various cryptosystems.

Earlier Cryptographic Systems

Before proceeding further, you need to know some facts about historical cryptosystems All of these systems are based on symmetric key encryption schemes.

The only security service these systems provide is confidentiality of information

How Traditional Ciphers treat data compared to Modern Ciphers?
◇ Unlike modern systems which are digital and treat data as binary numbers, the earlier traditional systems worked on alphabets as basic elements.

These earlier cryptographic systems are also referred to as Ciphers.

What is a cipher?
◇ In general, a cipher is simply just a set of steps (an algorithm) for performing both an encryption, and the corresponding decryption.

What are the traditional symmetric ciphers?
1. Caesar Cipher

2. Simple Substitution Cipher

3. Playfair Cipher

4. Vigenere Cipher

5. Transposition Cipher

Caesar Cipher

How does a Caesar Cipher operate?

◇ It is a mono-alphabetic cipher wherein each letter of the plaintext is substituted by another letter to form the ciphertext. It is the simplest form of substitution cipher scheme.

◇ This cryptosystem is generally referred to as the Shift Cipher. The concept is to replace each alphabet by another alphabet which is 'shifted' by some fixed number between 0 and 25.

For this type of scheme, both sender and receiver agree on a 'secret shift number' for shifting the alphabet. This number which is between 0 and 25 becomes the key of encryption.

The name 'Caesar Cipher' is occasionally used to describe the Shift Cipher when the 'shift of three' is used.

Process of Shift Cipher

In order to encrypt a plaintext letter, the sender positions the sliding ruler underneath the first set of plaintext letters and slides it to LEFT by the number of positions of the secret shift.
The plaintext letter is then encrypted to the ciphertext letter on the sliding ruler underneath.
The result of this process is depicted in the following illustration for an agreed shift of three positions. In this case, the plaintext 'tutorial' is encrypted to the ciphertext 'WXWRULDO'.
Here is the ciphertext alphabet for a Shift of 3 -

https://www.tutorialspoint.com/cryptography/images/process_of_shift_cipher.jpg

On receiving the ciphertext, the receiver who also knows the secret shift, positions his sliding ruler underneath the ciphertext alphabet and slides it to RIGHT by the agreed shift number, 3 in this case.
He then replaces the ciphertext letter by the plaintext letter on the sliding ruler underneath.

Hence the ciphertext ‘WXWRULDO’ is decrypted to ‘tutorial’. To decrypt a message encoded with a Shift of 3, generate the plaintext alphabet using a shift of ‘-3’ as shown below -

Security Value

Caesar Cipher is not a secure cryptosystem because there are only 26 possible keys to try out. An attacker can carry out an exhaustive key search with available limited computing resources.

Simple Substitution Cipher

◇ It is an improvement to the Caesar Cipher.

◇ Instead of shifting the alphabets by some number, this scheme uses some permutation of the letters in the alphabet.

For example, A.B…..Y.Z and Z.Y……B.A are two obvious permutation of all the letters in alphabet. Permutation is nothing but a jumbled up set of alphabets.
With 26 letters in alphabet, the possible permutations are 26! (Factorial of 26) which is equal to 4x1026. The sender and the receiver may choose any one of these possible permutation as a ciphertext alphabet. This permutation is the secret key of the scheme.

Process of Simple Substitution Cipher

Write the alphabets A, B, C,...,Z in the natural order.
The sender and the receiver decide on a randomly selected permutation of the letters of the alphabet.
Underneath the natural order alphabets, write out the chosen permutation of the letters of the alphabet. For encryption, the sender replaces each plaintext letter by substituting the permutation letter that is directly beneath it in the table. This process is shown in the following illustration. In this example, the chosen permutation is K,D, G, ..., O. The plaintext ‘point’ is encrypted to ‘MJBXZ’.
Here is a jumbled Ciphertext alphabet, where the order of the ciphertext letters is a key.

https://www.tutorialspoint.com/cryptography/images/simple_substitution_cipher.jpg

On receiving the ciphertext, the receiver, who also knows the randomly chosen permutation, replaces each ciphertext letter on the bottom row with the corresponding plaintext letter in the top row. The ciphertext ‘MJBXZ’ is decrypted to ‘point’.

Security Value

◇ The Simple Substitution Cipher is a considerable improvement over the Caesar Cipher.

The possible number of keys is large (26!) and even the modern computing systems are not yet powerful enough to comfortably launch a brute force attack to break the system. However, the Simple Substitution Cipher has a simple design and it is prone to design flaws, say choosing obvious permutations, this cryptosystem can be easily broken.

Monoalphabetic and Polyalphabetic Cipher

What is a monoalphabetic cipher?

◇ Monoalphabetic cipher is a substitution cipher in which for a given key, the cipher alphabet for each plain alphabet is fixed throughout the encryption process.

◇ For example, if ‘A’ is encrypted as ‘D’, for any number of occurrences in that plaintext, ‘A’ will always get encrypted to ‘D’.

All of the substitution ciphers we have discussed earlier in this chapter are monoalphabetic; these ciphers are highly susceptible to cryptanalysis.

What is a Polyalphabetic Cipher?

◇ Polyalphabetic Cipher is a substitution cipher in which the cipher alphabet for the plain alphabet may be different at different places during the encryption process.

◇ The next two examples, playfair and Vigenere Cipher are polyalphabetic ciphers.

What are the examples for Polyalphabetic ciphers?

List the two polyalphabetic ciphers

1. Playfai

2. Vigenere

Explain Playfair Cipher

◇ In this scheme, pairs of letters are encrypted, instead of single letters as in the case of simple substitution cipher.

In playfair cipher, initially a key table is created. The key table is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table as we need only 25 alphabets instead of 26. If the plaintext contains J, then it is replaced by I.

The sender and the receiver decide on a particular key, say ‘tutorials’. In a key table, the first characters (going left to right) in the table is the phrase, excluding the duplicate letters. The rest of the table will be filled with the remaining letters of the alphabet, in natural order. The key table works out to be -

https://www.tutorialspoint.com/cryptography/traditional_ciphers.htm

---

Vigenere Cipher

Process of Vigenere Cipher
Security Value

Variants of Vigenere Cipher
Vernam Cipher , One-time pad.

One-time pad
Security Value
Shift Cipher - Easy to Break
One-time Pad - Impossible to Break

Transposition Cipher

06_Modern Symmetric Key Encryption

What are the features of Modern Symmetric Key Encryption compared to traditional ciphers?

• Digital data is represented in strings of binary digits (bits) unlike alphabets.

• Modern cryptosystems need to process these binary strings to convert into another

• binary string.

How are symmetric encryption schemes classified?

◇ Based on how these binary strings are processed, a symmetric encryption schemes can be classified in to -

1. Block Ciphers

2. Stream Ciphers

What are the two classification of symmetric encryption?

1. Block-Ciphers

2. Stream-Ciphers

Explain Block Ciphers

Block Ciphers

◇ In this Block Cipher scheme, the plain binary text is processed in blocks (groups) of bits at a time

◇ i.e. a block of plaintext bits is selected, a series of operations is performed on this block to generate a block of ciphertext bits

◇ The number of bits in a block is fixed

◇ For example, the schemes DES and AES have block sizes of 64 and 128, respectively.

Explain Stream Ciphers

◇ In this scheme, the plaintext is processed one bit at a time

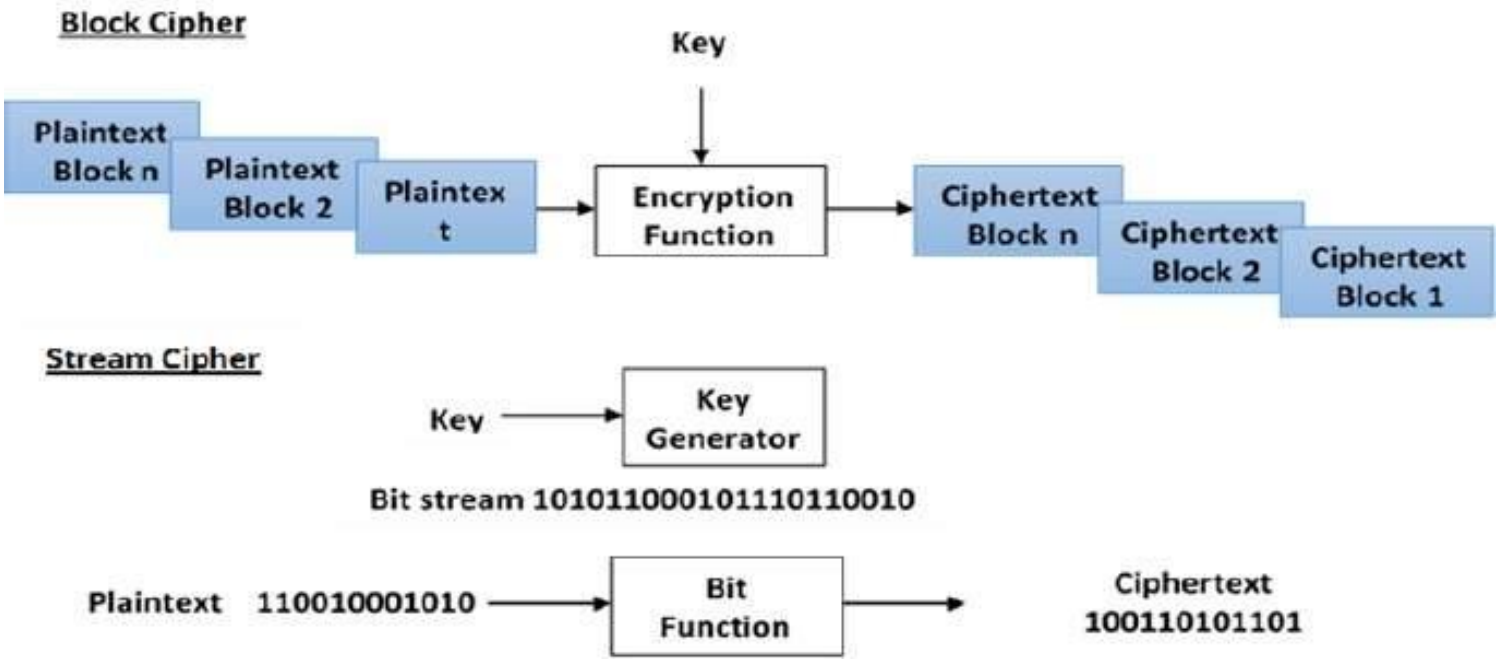
◇ i.e. one bit of plaintext is taken, and a series of operations is performed on it to generate one bit of ciphertext.

◇ Technically, stream ciphers are block ciphers with a block size of one bit. **XX**

Technically, stream ciphers are _____ with a block size of _____.

((Block Ciphers, one bit))

https://www.tutorialspoint.com/cryptography/images/block_and_stream_ciphers.jpg

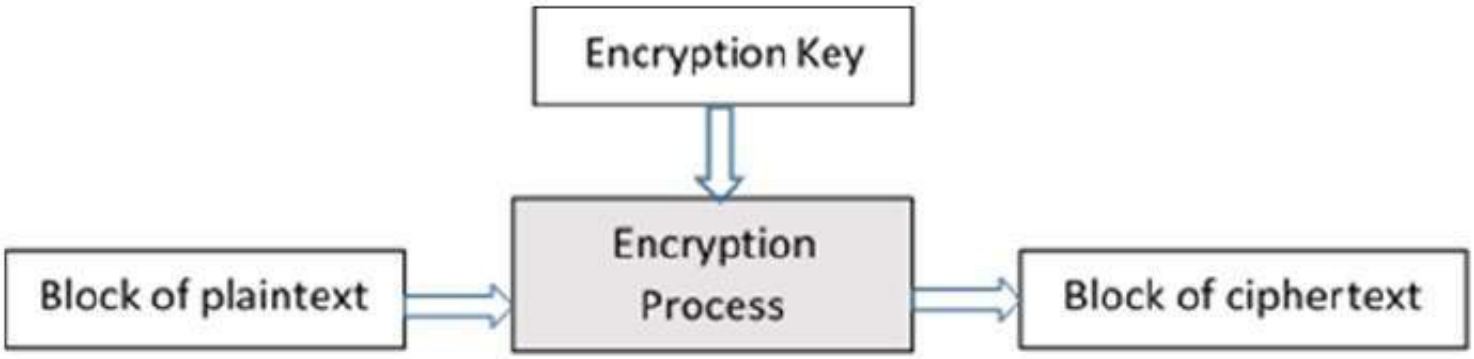


07_Block Cipher

Explain Block Cipher

The basic scheme of a block cipher is depicted as follows -

https://www.tutorialspoint.com/cryptography/images/block_cipher.jpg



• A block cipher takes a block of plaintext bits and generates a block of ciphertext bits, generally of the same size.

• The size of the block is fixed in the given scheme.

• The choice of block size does not directly affect to the strength of encryption scheme

• The strength of cipher depends up on the key length

Block Size

What aspects to keep in mind while selecting block size?

1. Avoid very small block size

2. Do not have very large block size

3. Multiples of 8 bit

◇ Though any size of block is acceptable, following aspects are borne in mind while selecting a size of a block

◇ Avoid very small block size

Say a block size is m bits. Then the possible plaintext bits combinations are then 2m. If the attacker discovers the plain text blocks corresponding to some previously sent ciphertext blocks, then the attacker can launch a type of ‘dictionary attack’ by building up a dictionary of plaintext/ciphertext pairs sent using that encryption key. A larger block size makes attack harder as the dictionary needs to be larger.

◇ Do not have very large block size

With very large block sizes, the cipher becomes inefficient to operate. Such plaintexts will need to be padded before being encrypted.

◇ Multiples of 8 bit

A preferred block size is a multiple of 8 as it is easy for implementation as most computer processors handle data in multiple of 8 bits.

What is padding?

◇ The process of adding bits to the last block is referred to as padding.

Explain Padding in Block Cipher

◇ Block ciphers process blocks of fixed sizes (say 64 bits).

◇ The length of plaintexts is mostly not a multiple of the block size.

◇ For example, a 150-bit plaintext provides two blocks of 64 bits each with third block of balance 22 bits

◇ The last block of bits needs to be padded up with redundant information so that the length of the final block equal to block size of the scheme

In our example, the remaining 22 bits need to have additional 42 redundant bits added to provide a complete block

The process of adding bits to the last block is referred to as padding.

What are the disadvantages of using padding?

◇ Too much padding makes the system inefficient.

◇ Also, padding may render the system insecure at times, if the padding is done with the same bits always.

Block Cipher Schemes

List the popular Block Cipher schemes

1. Digital Encryption Standard (DES)

2. Triple DES (3DES)

3. Advanced Encryption Standard (AES)

4. IDEA

5. Twofish

6. Serpent

There are a vast number of block ciphers schemes that are in use. Many of them are publically known. Most popular and prominent block ciphers are listed below.

◇ Digital Encryption Standard (DES) - The popular block cipher of the 1990s. It is now considered as a ‘broken’ block cipher, due primarily to its small key size.

◇ Triple DES - It is a variant scheme based on repeated DES applications. It is still a respected block ciphers but inefficient compared to the new faster block ciphers available.

◇ Advanced Encryption Standard (AES) - It is a relatively new block cipher based on the encryption algorithm Rijndael that won the AES design competition.

◇ IDEA - It is a sufficiently strong block cipher with a block size of 64 and a key size of 128 bits. A number of applications use IDEA encryption, including early versions of Pretty Good Privacy (PGP) protocol. The use of the IDEA scheme has a restricted adoption due to patent issues.

◇ Twofish - This scheme of block cipher uses block size of 128 bits and a key of variable length. It was one of the AES finalists. It is based on the earlier block cipher Blowfish with a block size of 64 bits.

◇ Serpent - A block cipher with a block size of 128 bits and key lengths of 128, 192, or 256 bits, which was also an AES competition finalist. It is slower but has a more secure design than other block ciphers.

In the next sections, we will first discuss the model of block cipher followed by DES and AES, two of the most influential modern block ciphers.

Feistel Block Cipher

08_Feistel Block Cipher

What is Feistel Cipher?

• Feistel Cipher is not a specific scheme of block cipher.

• It is a design model from which many different block ciphers are derived.
A cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption.

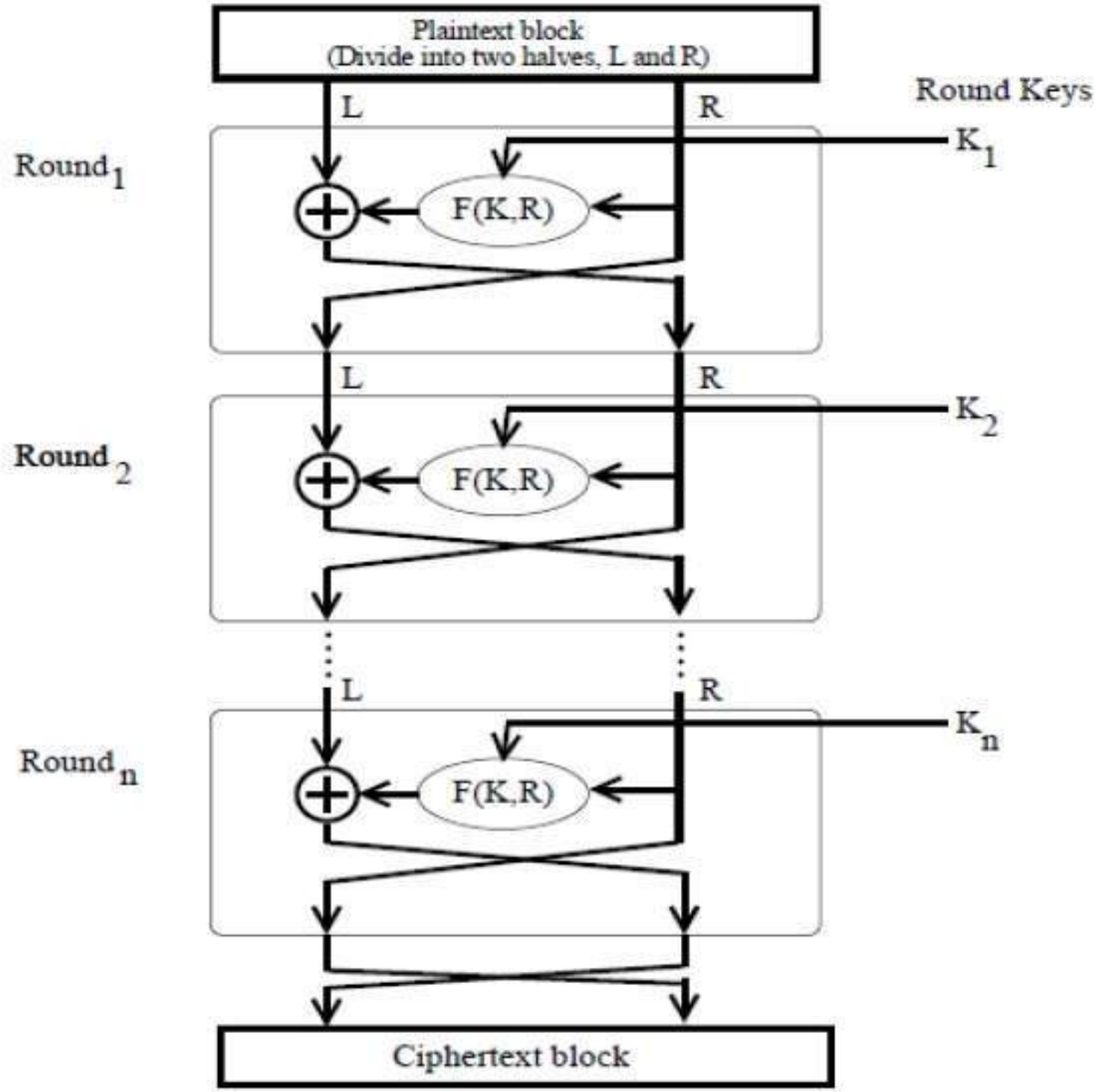
• DES is just one example of a Feistel Cipher.

What is the encryption process of Fiestel Cipher?

Encryption Process

◇ The encryption process uses the Feistel structure consisting of multiple rounds of processing of the plaintext, each round consisting of a “substitution” step followed by a permutation step.

Feistel Structure is shown in the following illustration –



◇ The input block to each round is divided into two halves that can be denoted as L and R for the left half and the right half.

◇ In each round, the right half of the block, R, goes through unchanged. But the left half, L, goes through an operation that depends on R and the encryption key. First, we apply an encrypting function 'f' that takes two input – the key K and R. The function produces the output f(R,K). Then, we XOR the output of the mathematical function with L.

◇ In real implementation of the Feistel Cipher, such as DES, instead of using the whole encryption key during each round, a round-dependent key (a subkey) is derived from the encryption key. This means that each round uses a different key, although all these subkeys are related to the original key.

◇ The permutation step at the end of each round swaps the modified L and unmodified R. Therefore, the L for the next round would be R of the current round. And R for the next round will be the output L of the current round.

◇ Above substitution and permutation steps form a 'round'. The number of rounds are specified by the algorithm design.

◇ Once the last round is completed then the two sub blocks, 'R' and 'L' are concatenated in this order to form the ciphertext block.

The difficult part of designing a Feistel Cipher is the selection of round function 'f'. In order to be an unbreakable scheme, this function needs to have several important properties that are beyond the scope of our discussion.

Decryption Process ⚓

What is the decryption process in Fiestel Cipher?

◇ The process of decryption in Feistel cipher is almost similar. Instead of starting with a block of plaintext, the ciphertext block is fed into the start of the Feistel structure and then the process thereafter is exactly the same as described in the given illustration.

The process is said to be almost similar and not exactly the same. In the case of decryption, the only difference is that the subkeys used in encryption are used in the reverse order.
The final swapping of 'L' and 'R' in the last step of the Feistel Cipher is essential. If these are not swapped then the resulting ciphertext could not be decrypted using the same algorithm.

Number of Rounds ⚓

Wha is number of rounds in Fiestel Cipher?

◇ The number of rounds used in a Feistel Cipher depends on desired security from the system.

◇ More rounds provide a more secure system. But at the same time, more rounds mean the inefficient slow encryption and decryption processes.

■ Number of rounds in the systems thus depends upon efficiency–security tradeoff.

09_DES

Data Encryption Standard⚓

What does DES stand for ? -- Data Encryption standard.

What does NIST stand for?-- National Institute of Standards and Technology.

What is DES?

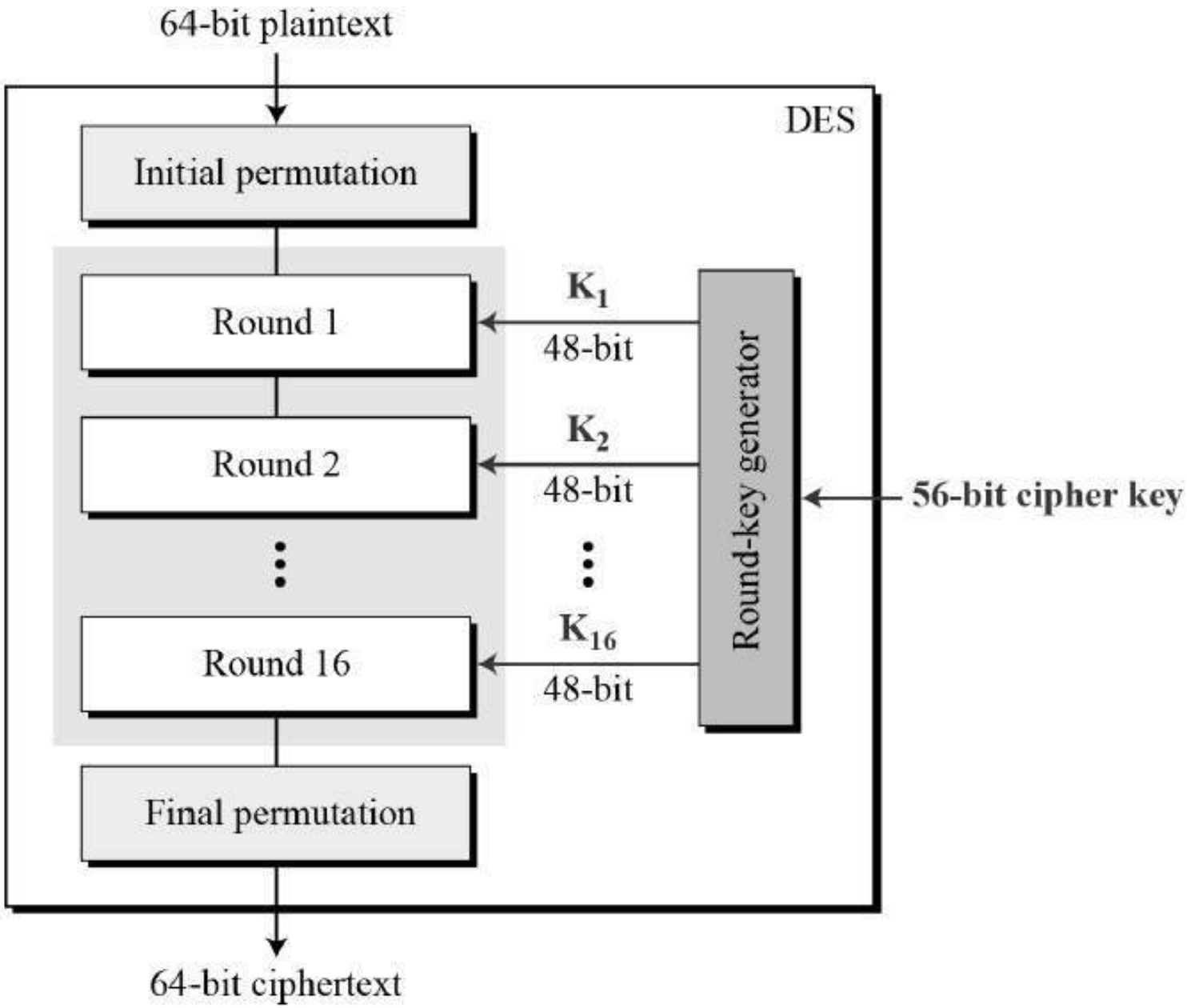
• The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

• DES is an implementation of a Feistel Cipher structure.

◇ It uses a 16 round Feistel structure.

◇ The block size is 64-bit.

◇ Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −



What symmetric cipher structure DES is based on?

◇ Fiestel Cipher Structure

What are the requirements of DES Cipher?

Since DES is based on the Feistel Cipher, all that is required to specify DES is –

◇ Round function

◇ Key schedule

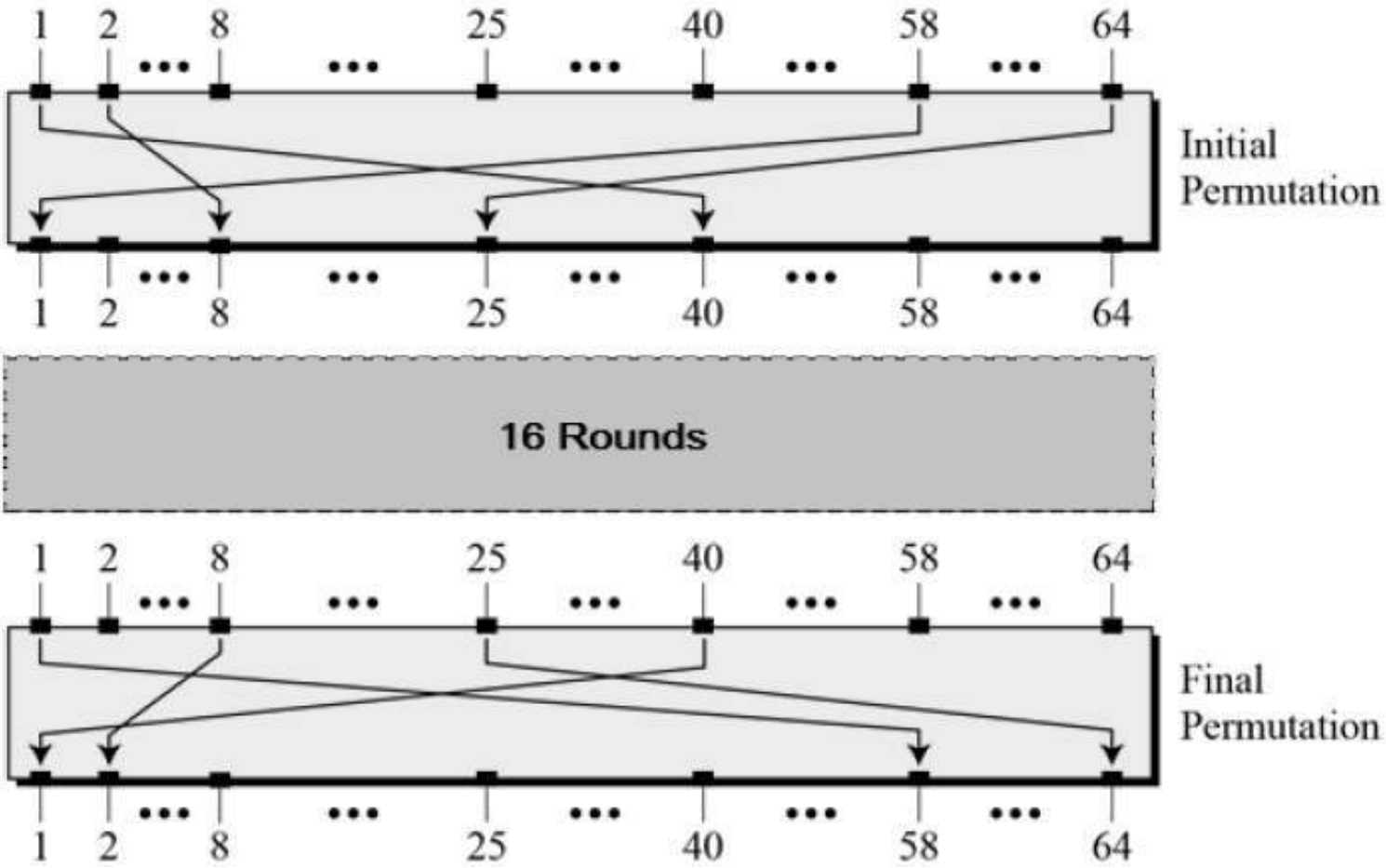
◇ Any additional processing – Initial and final permutation

Initial and Final Permutation

Explain Initial and Final Permutation?

◇ The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES. The initial and final permutations are shown as follows −

◇

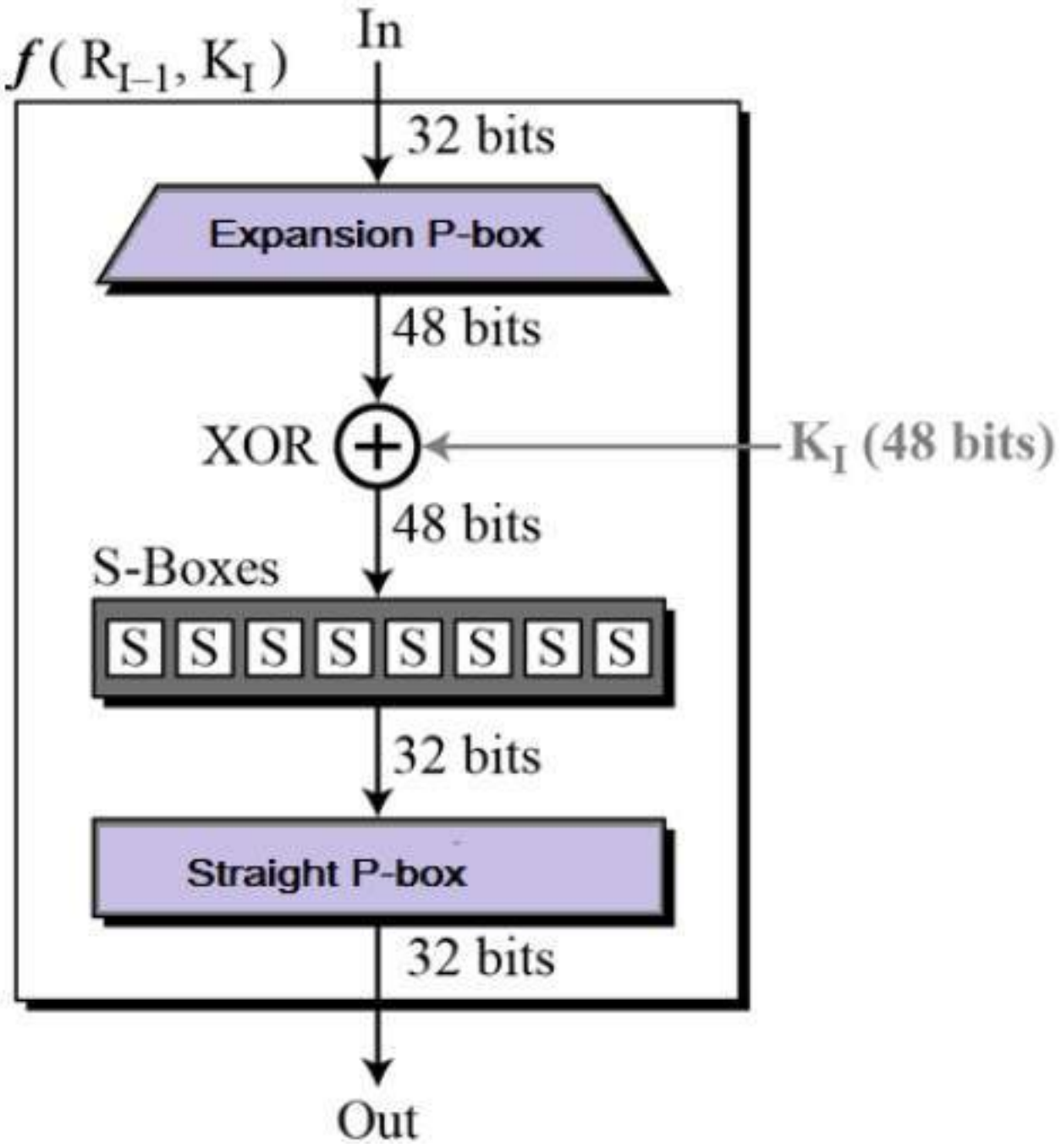


Round Function

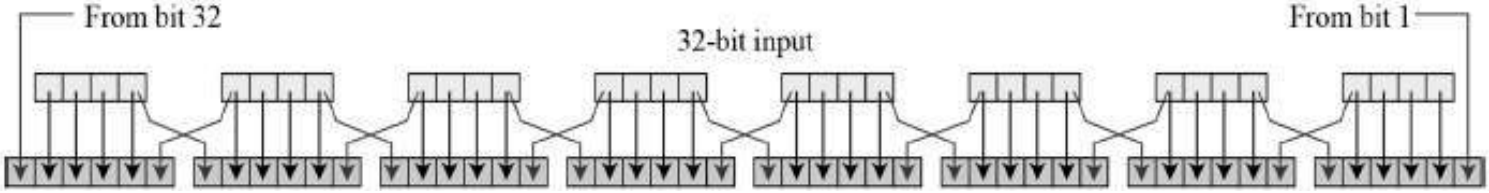
Explain Round Function?

◇ The heart of this cipher is the DES function, *f*.

◇ The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



◇ Expansion Permutation Box − Since the right input is 32-bit and the round key is 48-bit, we first need to expand the right input to 48 bits. Permutation logic is graphically depicted in the following illustration −

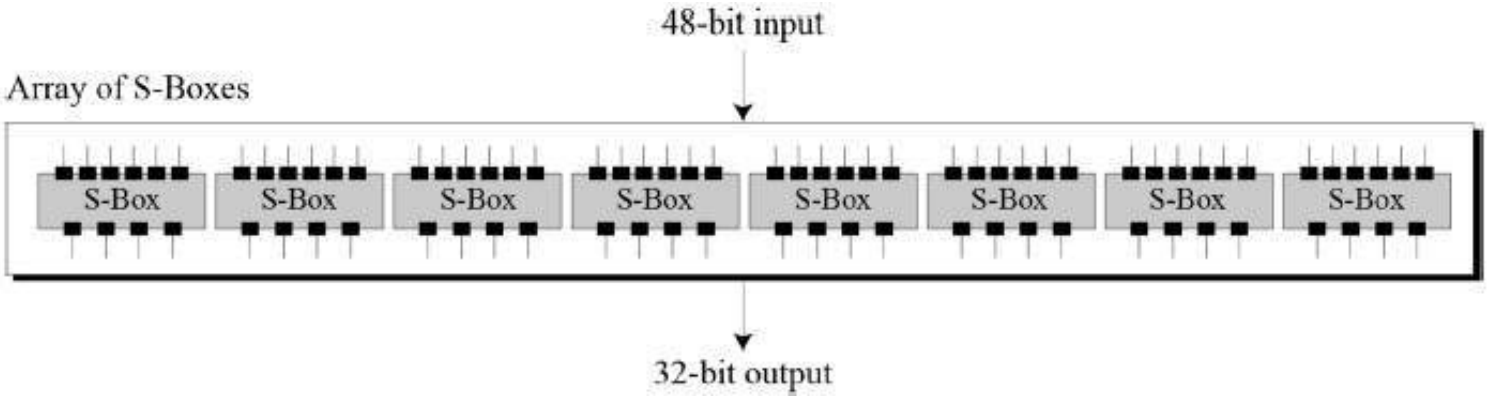


◇ The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown −

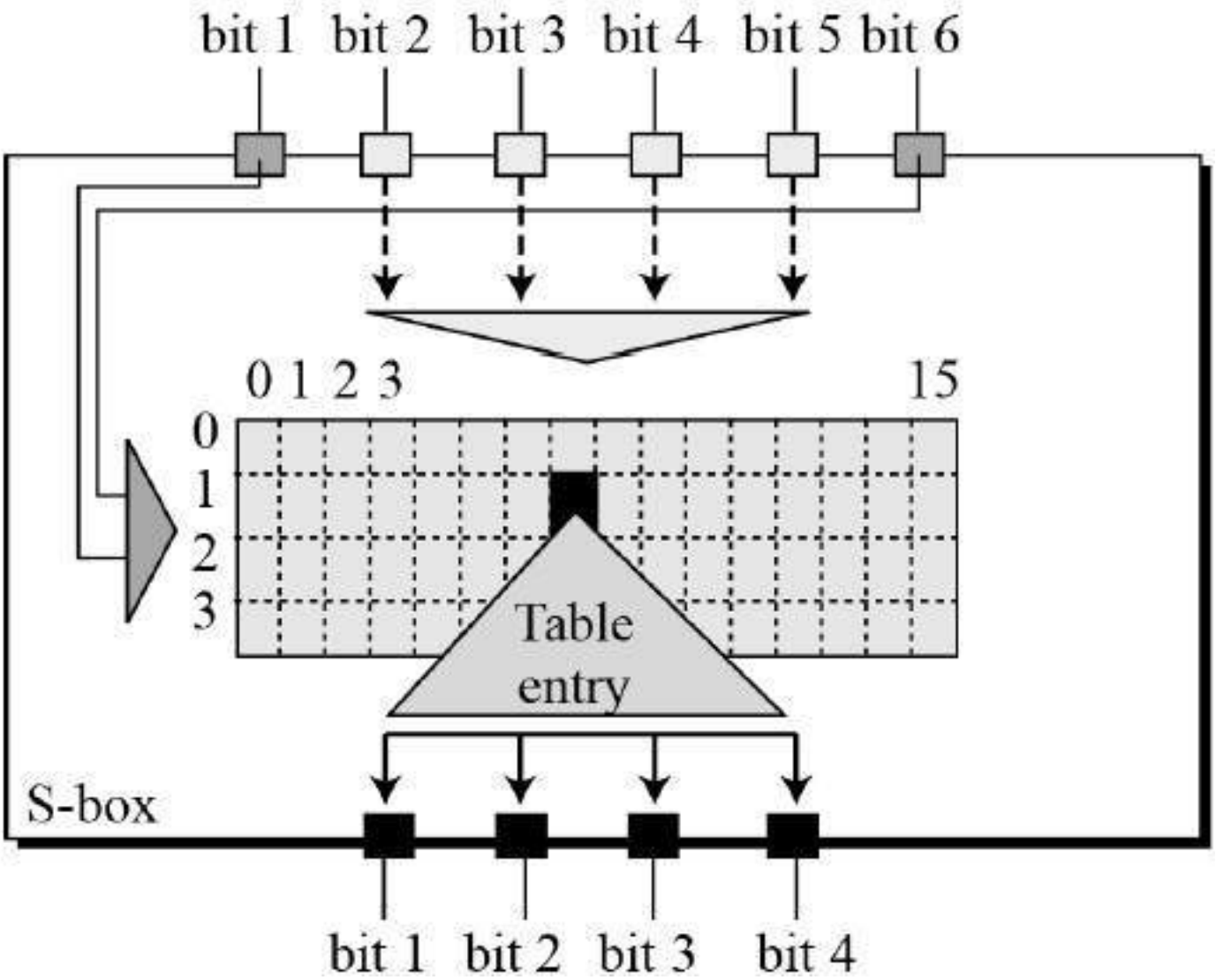
32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

◇ XOR (Whitener). − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.

◇ Substitution Boxes. − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration −



◇ The S-box rule is illustrated below −



◇ There are a total of eight S-box tables. The output of all eight s-boxes is then combined into a 32 bit section.

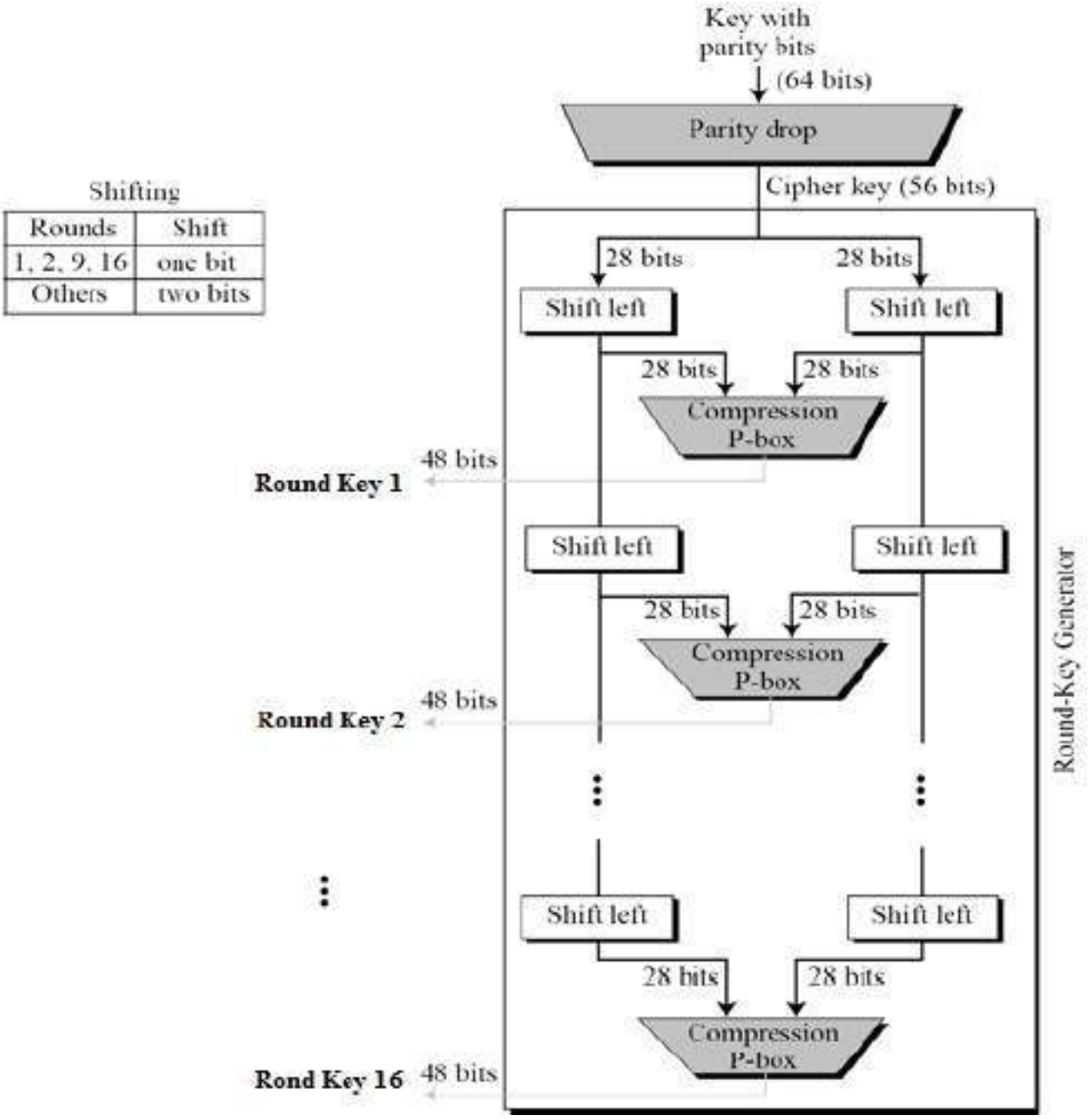
◇ Straight Permutation – The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Key Generation⚓

Explain Key generation

◇ The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration −



The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

DES Analysis ⚓

What are the two properties of DES ? ⚓

◇ Avalanche Effect

◇ Completeness

Explain Avalanche effect, Completeness?

The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.

◇ Avalanche effect − A small change in plaintext results in a very great change in the ciphertext.

◇ Completeness − Each bit of ciphertext depends on many bits of plaintext.

During the last few years, cryptanalysis have found some weaknesses in DES when keys selected are weak keys. These keys shall be avoided.

DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

10_Triple DES

Triple DES⚓

The speed of exhaustive key searches against DES after 1990 began to cause discomfort amongst users of DES. However, users did not want to replace DES as it takes an enormous amount of time and money to change encryption algorithms that are widely adopted and embedded in large security architectures.

The pragmatic approach was not to abandon the DES completely, but to change the manner in which DES is used. This led to the modified schemes of Triple DES (sometimes known as 3DES).

Incidentally, there are two variants of Triple DES known as 3-key Triple DES (3TDES) and 2-key Triple DES (2TDES).

What are the two variants of Triple DES (3DES) ?

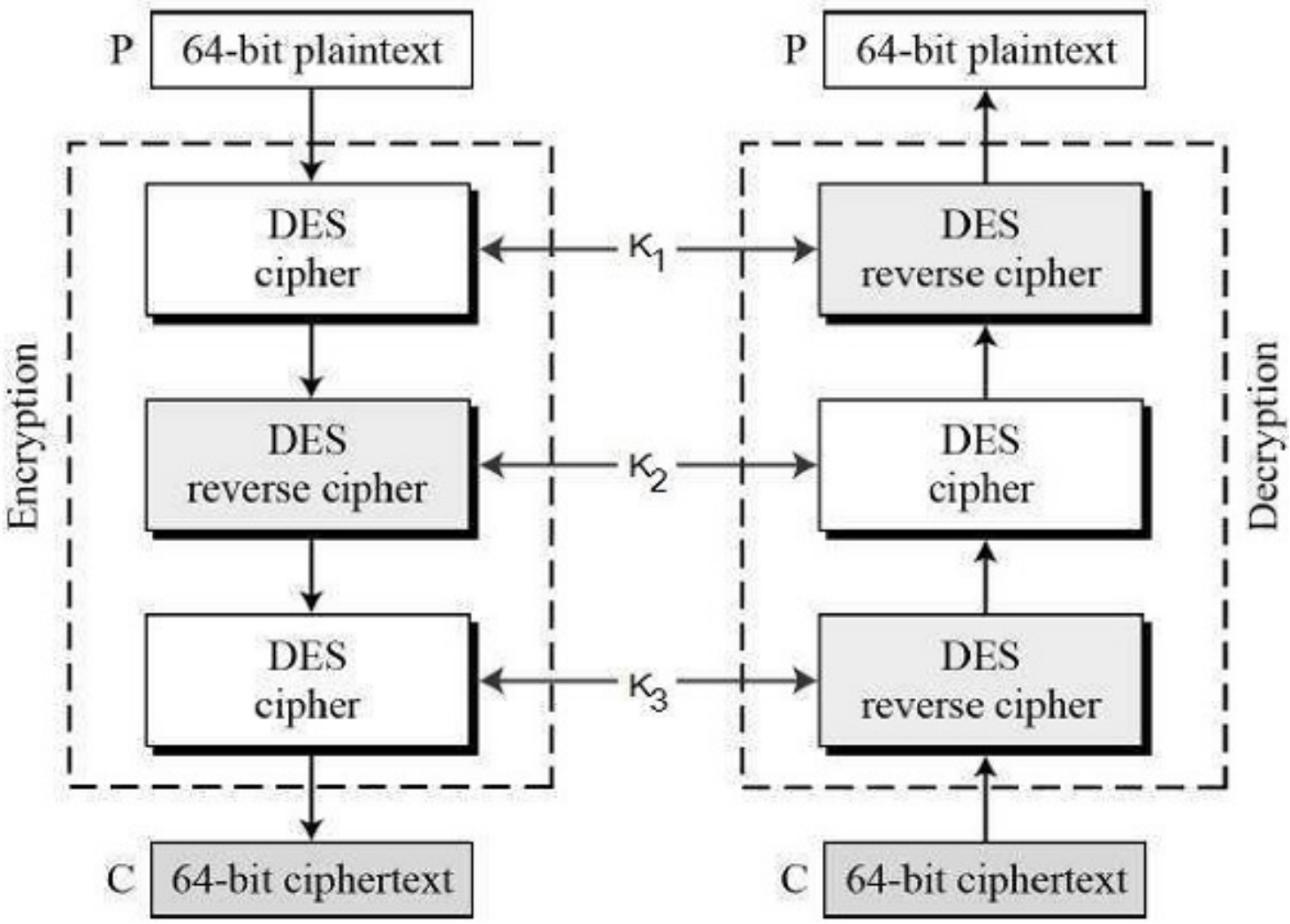
1. 3-key Triple DES (3TDES)

2. Triple DES (2TDES).

Explain 3-KEY Triple DES⚓

3-KEY Triple DES⚓

Before using 3TDES, users first generate and distribute a 3TDES key K, which consists of three different DES keys K_1, K_2 and K_3. This means that the actual 3TDES key has length $3 \times 56 = 168$ bits. The encryption scheme is illustrated as follows −



Explain the encryption-decryption process of 3-KEY Triple DES

The encryption-decryption process is as follows –

• Encrypt the plaintext blocks using single DES with key K1.

• Now decrypt the output of step 1 using single DES with key K2.

• Finally, encrypt the output of step 2 using single DES with key K3.

• The output of step 3 is the ciphertext.

• Decryption of a ciphertext is a reverse process. User first decrypt using K3, then encrypt with K2, and finally decrypt with K1.

Due to this design of Triple DES as an encrypt–decrypt–encrypt process, it is possible to use a 3TDES (hardware) implementation for single DES by setting K1, K2, and K3 to be the same value. This provides backwards compatibility with DES.

Explain 2TDES

Second variant of Triple DES (2TDES) is identical to 3TDES except that K3is replaced by K1.

In other words, users encrypt plaintext blocks with key K_1, then decrypt with key K_2, and finally encrypt with K_1 again. Therefore, 2TDES has a key length of 112 bits.

What is the disadvantage of Triple DES compared to DES?

Triple DES systems are significantly more secure than single DES, but these are clearly a much slower process than encryption using single DES.

11_Advanced Encryption Standard

What does AES stand for? -- Advanced Encryption Standard

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six times faster than triple DES.

How fast is AES compared to DES?
It is found at least six time faster than triple DES

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attacks. Triple DES was designed to overcome this drawback but it was found slow.

What are the features of AES?
1. Symmetric key symmetric block cipher

2. 128-bit data, 128/192/256-bit keys

3. Stronger and faster than Triple-DES

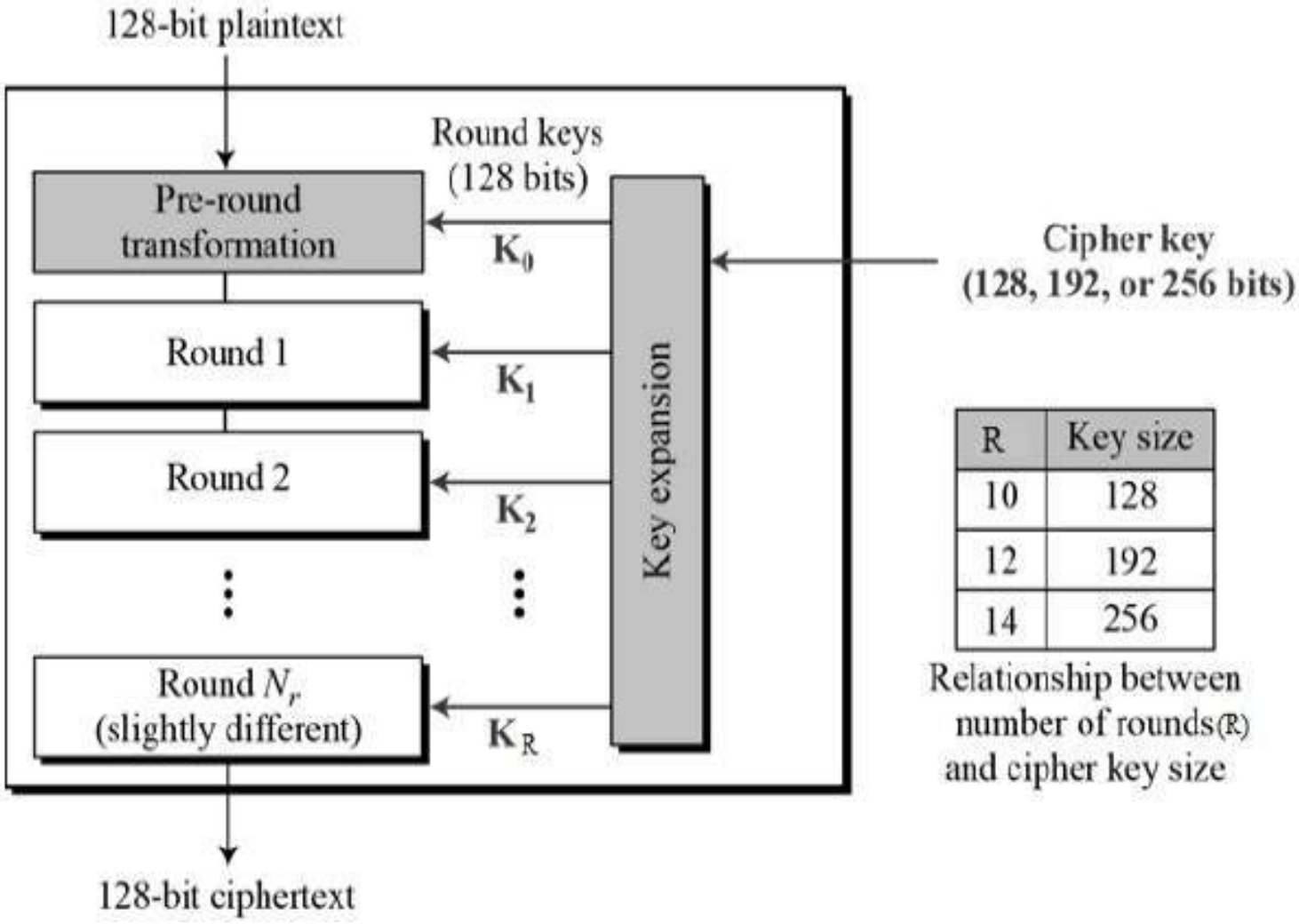
4. Provides full specification and design details

5. Software implementable in C and Java

Operation of AES⚓

Explain the Operation of AES:
AES is an iterative rather than Feistel cipher. It is based on the 'substitution–permutation network'. It comprises a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).
AES is based on " 'substitution–permutation network' "
AES performs all its computations on bytes rather than bits
Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix –
Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the

original AES key.
The schematic of AES structure is given in the following illustration −



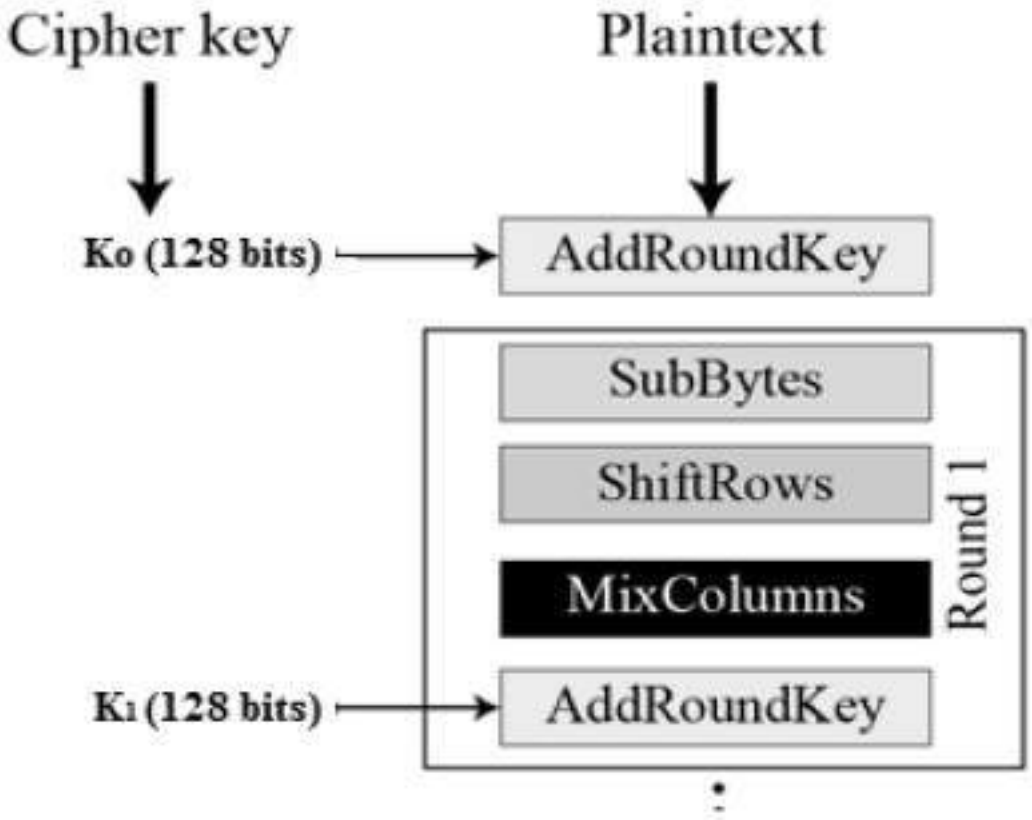
R	Key size
10	128
12	192
14	256

Relationship between number of rounds (R) and cipher key size

Encryption Process ⚓

Explain Encryption process of AES

Here, we restrict to description of a typical round of AES encryption. Each round comprises four sub-processes. The first round process is depicted below −



Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of the row. Shift is carried out as follows –

• First row is not shifted.

• Second row is shifted one (byte) position to the left.

• Third row is shifted two positions to the left.

• Fourth row is shifted three positions to the left.

• The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new

bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Explain Decryption Process of AES

Decryption Process

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order –

◇ Add round key

◇ Mix columns

◇ Shift rows

◇ Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related.

AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES have been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.
However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

12_Block Cipher Modes of Operation

In this chapter, we will discuss the different modes of operation of a block cipher. These are procedural rules for a generic block cipher. Interestingly, the different modes result in different properties being achieved which add to the security of the underlying block cipher.

What is the mode of operation of a block cipher?

• The Modes are procedural rules for a generic block cipher

What is the benefit of using different Modes?

◇ The different modes result in different properties being achieved which add to the security of the underlying block cipher.

How does a block cipher process the data?

◇ A block cipher processes the data blocks of fixed size.

◇ Usually, the size of a message is larger than the block size.

◇ Hence, the long message is divided into a series of sequential message blocks, and the cipher operates on these blocks one at a time.

List the different types of Modes available in block cipher processing?

1. ECB- Electronic Code Book

2. CBC- Cipher Block Chaining

3. CFB- Cipher FeedBack mode

4. OFB- Output FeedBack mode

5. CTR- Counter mode

Electronic Code Book (ECB) Mode⚓

What is ECB

This mode is the most straightforward way of processing a series of sequentially listed message blocks.

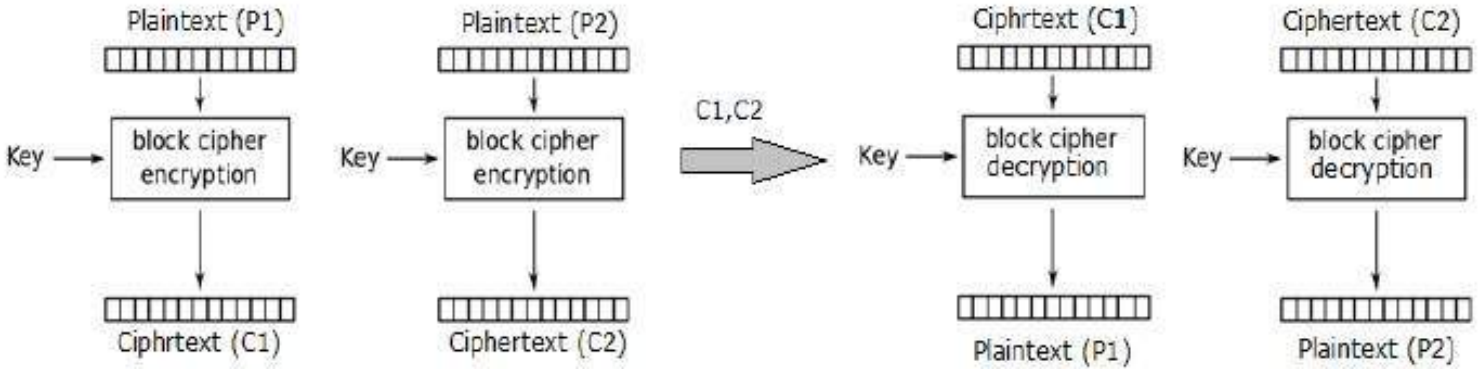
Operation

Explain the Operation of the ECB?

◇ The user takes the first block of plaintext and encrypts it with the key to produce the first block of ciphertext.

◇ He then takes the second block of plaintext and follows the same process with the same key and so on and so forth.

The ECB mode is deterministic, that is, if plaintext blocks P1, P2,..., Pm are encrypted twice under the same key, the output ciphertext blocks will be the same.
In fact, for a given key technically we can create a codebook of ciphertexts for all possible plaintext blocks. Encryption would then entail only looking up for the required plaintext and selecting the corresponding ciphertext. Thus, the operation is analogous to the assignment of code words in a codebook, and hence gets an official name − Electronic Codebook mode of operation (ECB). It is illustrated as follows −



Analysis of ECB Mode

In reality, any application data usually has partial information which can be guessed. For example, the range of salary can be guessed. A ciphertext from ECB can allow an attacker to guess the plaintext by trial-and-error if the plaintext message is predictable.
For example, if a ciphertext from the ECB mode is known to encrypt a salary figure, then a small number of trials will allow an attacker to recover the figure. In general, we do not wish to use a deterministic cipher, and hence the ECB mode should not be used in most applications.

Cipher Block Chaining (CBC) Mode

What is CBC Mode?

CBC mode of operation provides message dependence for generating ciphertext and makes the system non-deterministic.

Operation

Explain the Operation of CBC?

The operation of CBC mode is depicted in the following illustration. The steps are as follows –

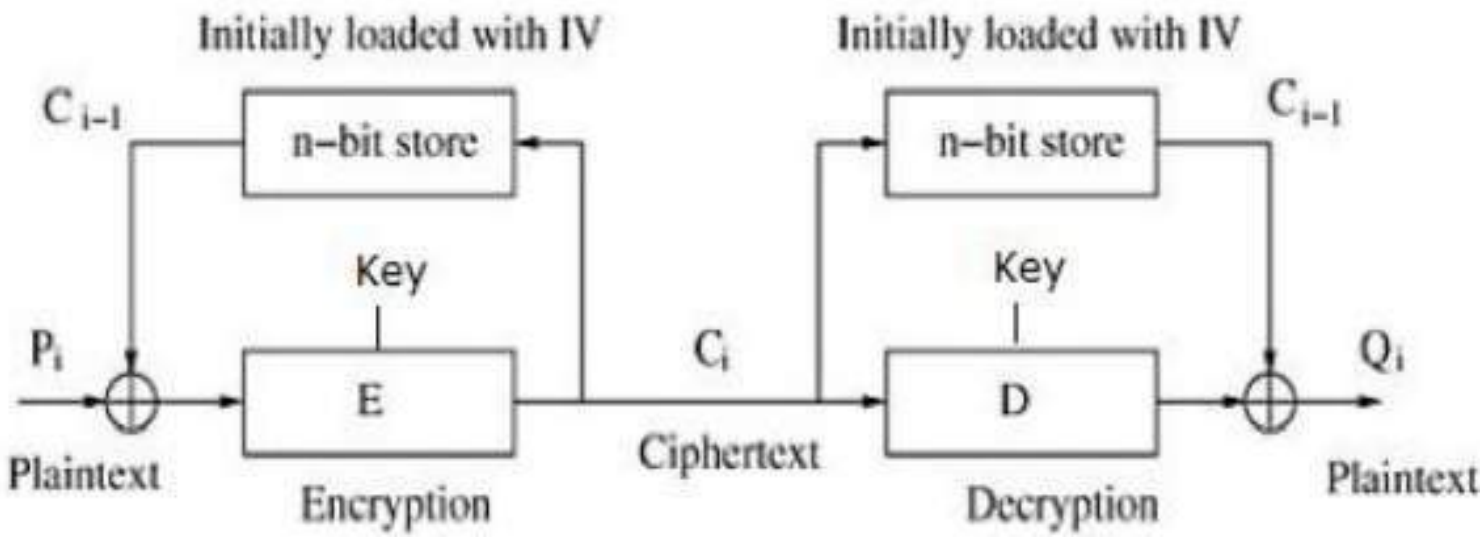
◇ Load the n-bit Initialization Vector (IV) in the top register.

◇ XOR the n-bit plaintext block with data value in the top register.

◇ Encrypt the result of XOR operation with the underlying block cipher with key K.

◇ Feed the ciphertext block into the top register and continue the operation till all plaintext blocks are processed.

◇ For decryption, IV data is XORed with the first ciphertext block decrypted. The first ciphertext block is also fed into to register replacing IV for decrypting the next ciphertext block.



Analysis of CBC Mode⚓

In CBC mode, the current plaintext block is added to the previous ciphertext block, and then the result is encrypted with the key. Decryption is thus the reverse process, which involves decrypting the current ciphertext and then adding the previous ciphertext block to the result.

Advantage of CBC over ECB is that changing IV results in different ciphertext for identical messages. On the drawback side, the error in transmission gets propagated to a few further blocks during decryption due to the chaining effect.

It is worth mentioning that CBC mode forms the basis for a well-known data origin authentication mechanism. Thus, it has an advantage for those applications that require both symmetric encryption and data origin authentication.

Cipher Feedback (CFB) Mode⚓

What is CFB?

Explain CFB Mode?

In this mode, each ciphertext block gets ‘fed back’ into the encryption process in order to encrypt the next plaintext block.

Operation⚓

Explain the Operation of CFB?

The operation of CFB mode is depicted in the following illustration. For example, in the present system, a message block has a size ‘s’ bits where 1 < s < n. The CFB mode requires an initialization vector (IV) as the initial random n-bit input block. The IV need not be secret.
Steps of operation are –

◇ Load the IV in the top register.

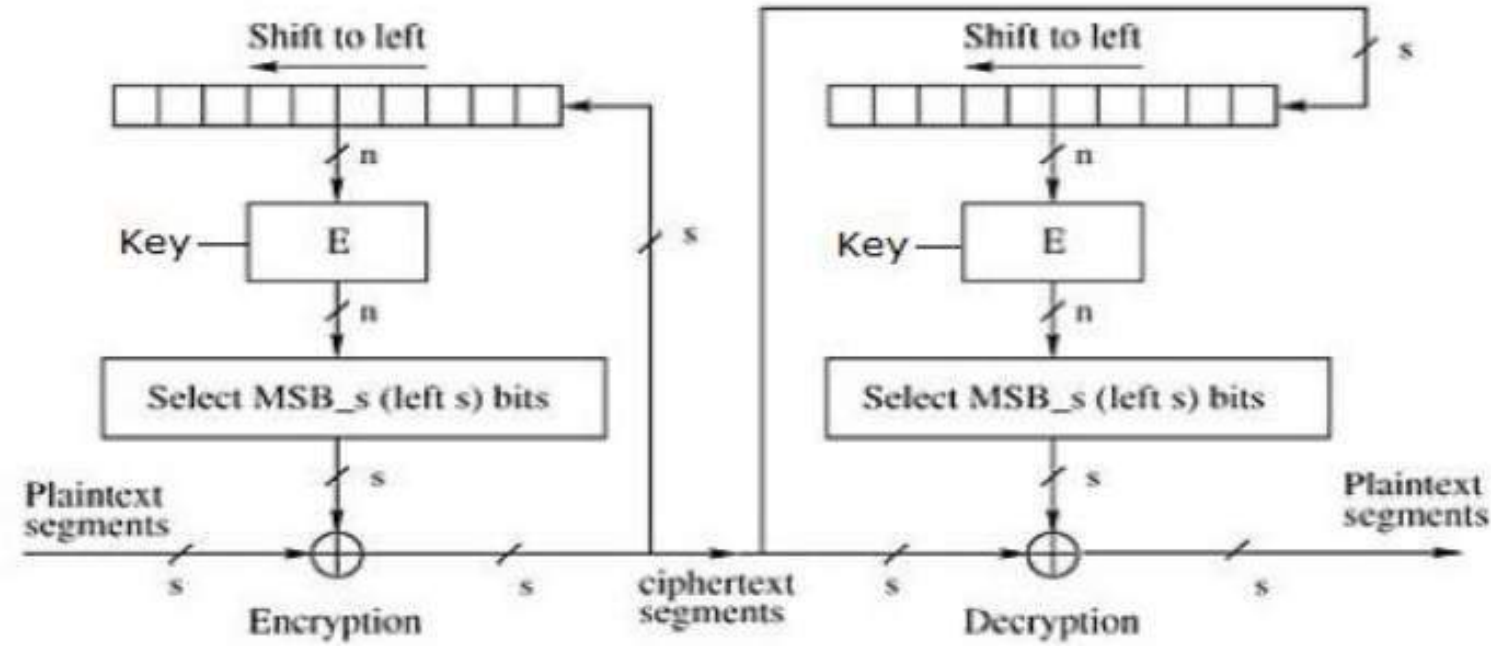
◇ Encrypt the data value in the top register with the underlying block cipher with key K.

◇ Take only ‘s’ number of most significant bits (left bits) of output of encryption process and XOR them with ‘s’ bit plaintext message block to generate ciphertext block.

◇ Feed the ciphertext block into the top register by shifting already present data to the left and continue the operation till all plaintext blocks are processed.

◇ Essentially, the previous ciphertext block is encrypted with the key, and then the result is XORed to the current plaintext block.

◇ Similar steps are followed for decryption. Pre-decided IV is initially loaded at the start of decryption.



Analysis of CFB Mode

CFB mode differs significantly from ECB mode, the ciphertext corresponding to a given plaintext block depends not just on that plaintext block and the key, but also on the previous ciphertext block. In other words, the ciphertext block is dependent on the message.

CFB has a very strange feature. In this mode, the user decrypts the ciphertext using only the encryption process of the block cipher. The decryption algorithm of the underlying block cipher is never used.

Apparently, CFB mode is converting a block cipher into a type of stream cipher. The encryption algorithm is used as a key-stream generator to produce a key-stream that is placed in the bottom register. This key stream is then XORed with the plaintext as in case of stream cipher.

By converting a block cipher into a stream cipher, CFB mode provides some of the advantageous properties of a stream cipher while retaining the advantageous properties of a block cipher.

On the flip side, the error of transmission gets propagated due to changing of blocks.

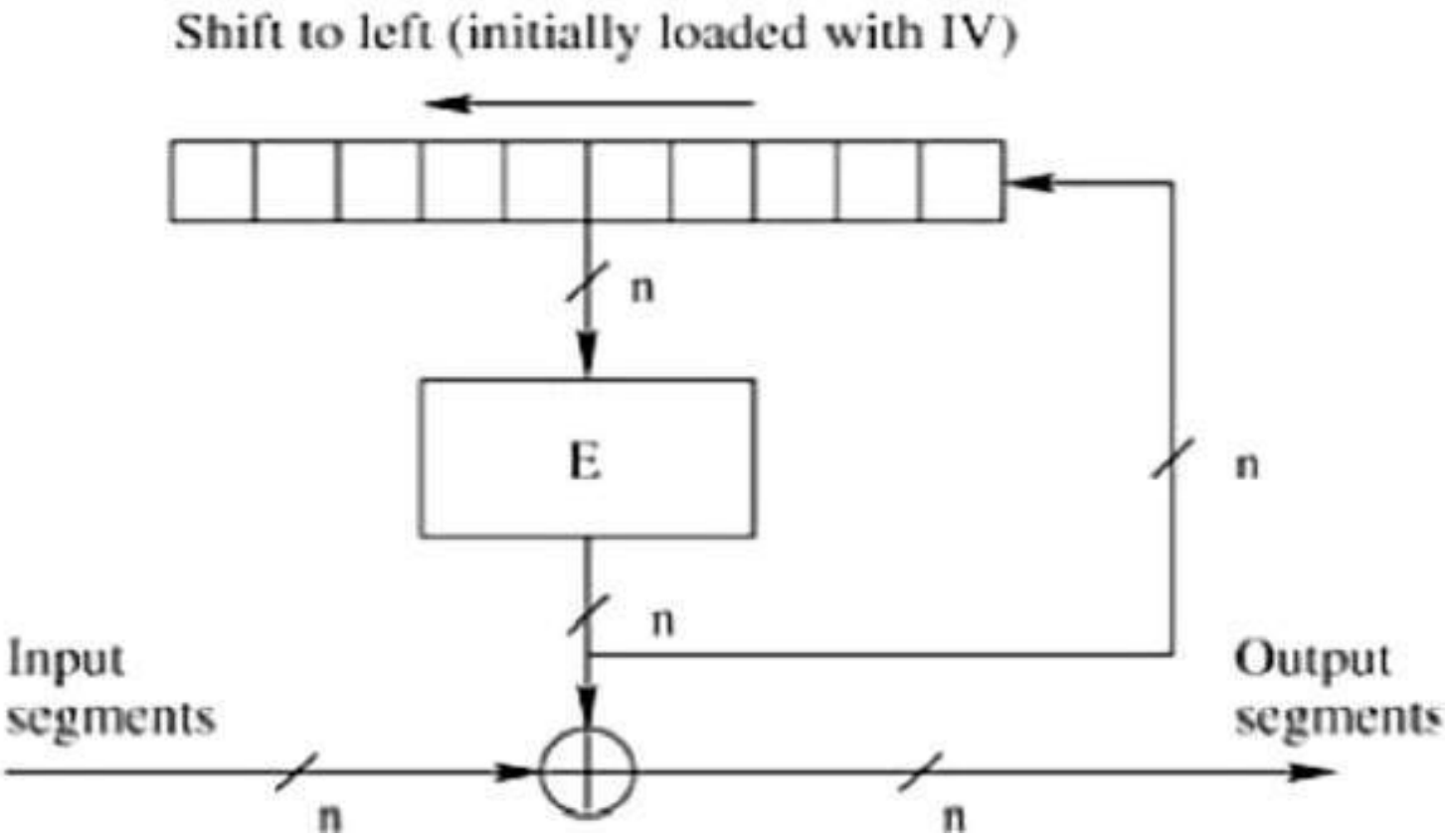
Output Feedback (OFB) Mode

What is OFB?

It involves feeding the successive output blocks from the underlying block cipher back to it. These feedback blocks provide a string of bits to feed the encryption algorithm which act as the key-stream generator as in case of CFB mode.

The key stream generated is XOR-ed with the plaintext blocks. The OFB mode requires an IV as the initial random n-bit input block. The IV need not be secret.

The operation is depicted in the following illustration –



Counter (CTR) Mode

What is CTR Mode?

Explain CTR Mode?

It can be considered as a counter-based version of CFB mode without the feedback. In this mode, both the sender and receiver need access to a reliable counter, which computes a new shared value each time a ciphertext block is exchanged. This shared counter is not necessarily a secret value, but the challenge is that both sides must keep the counter synchronized.

Operation

Both encryption and decryption in CTR mode are depicted in the following illustration.
Steps in operation are –

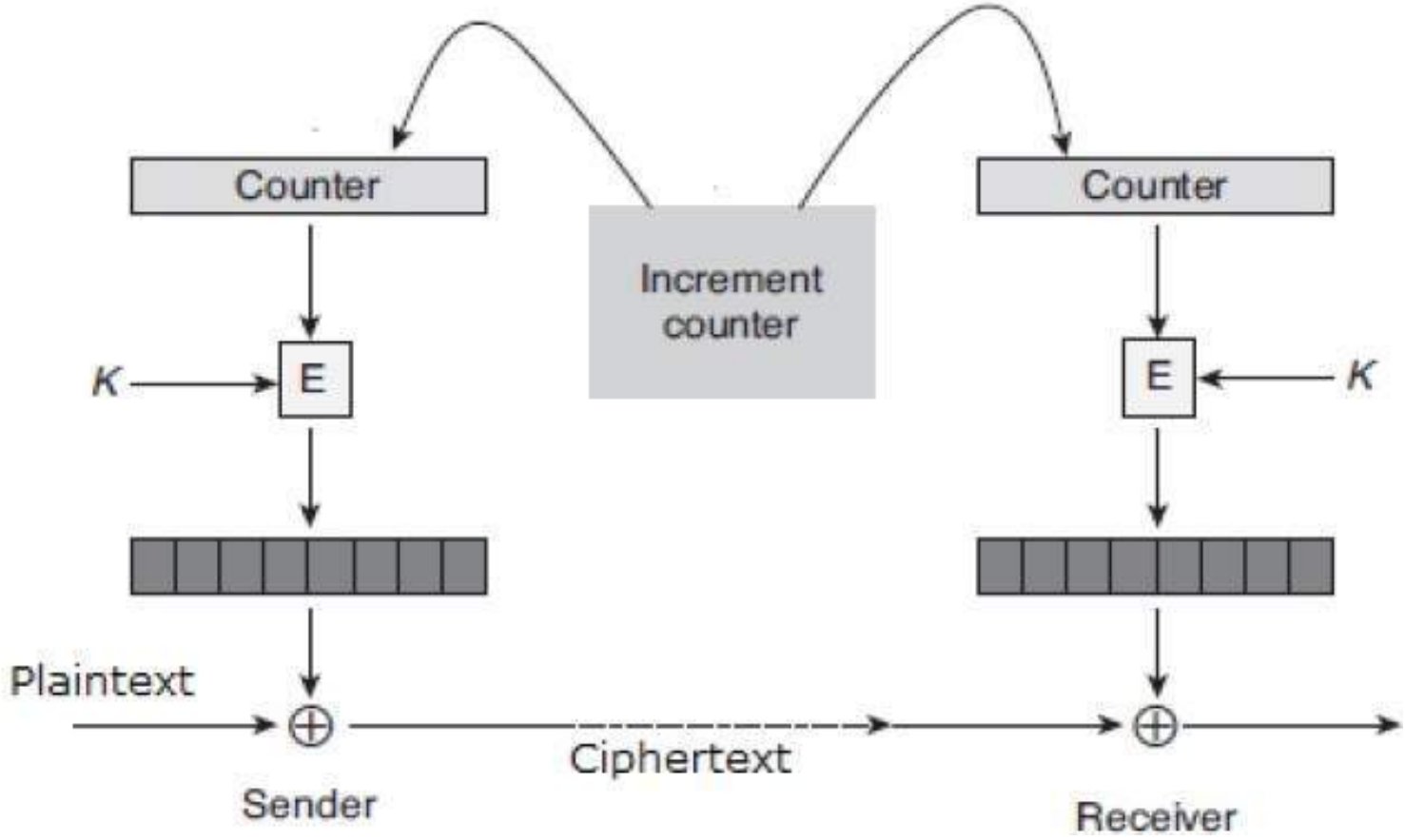
◇ Loading the initial counter value in the top register is the same for both the sender and the receiver. It plays the same role as the IV in CFB (and CBC) mode.

◇ Encrypt the contents of the counter with the key and place the result in the bottom register.

◇ Take the first plaintext block P1 and XOR this to the contents of the bottom register. The result of this is C1. Send C1 to the receiver and update the counter. The counter update replaces the ciphertext feedback in CFB mode.

◇ Continue in this manner until the last plaintext block has been encrypted.

◇ The decryption is the reverse process. The ciphertext block is XORed with the output of encrypted contents of counter value. After decryption of each ciphertext block counter is updated as in case of encryption.



Analysis of Counter Mode ⚓

It does not have message dependency and hence a ciphertext block does not depend on the previous plaintext blocks.

Like CFB mode, CTR mode does not involve the decryption process of the block cipher. This is because the CTR mode is really using the block cipher to generate a key-stream, which is encrypted using the XOR function. In other words, CTR mode also converts a block cipher to a stream cipher.

The serious disadvantage of CTR mode is that it requires a synchronous counter at sender and receiver. Loss of synchronization leads to incorrect recovery of plaintext.

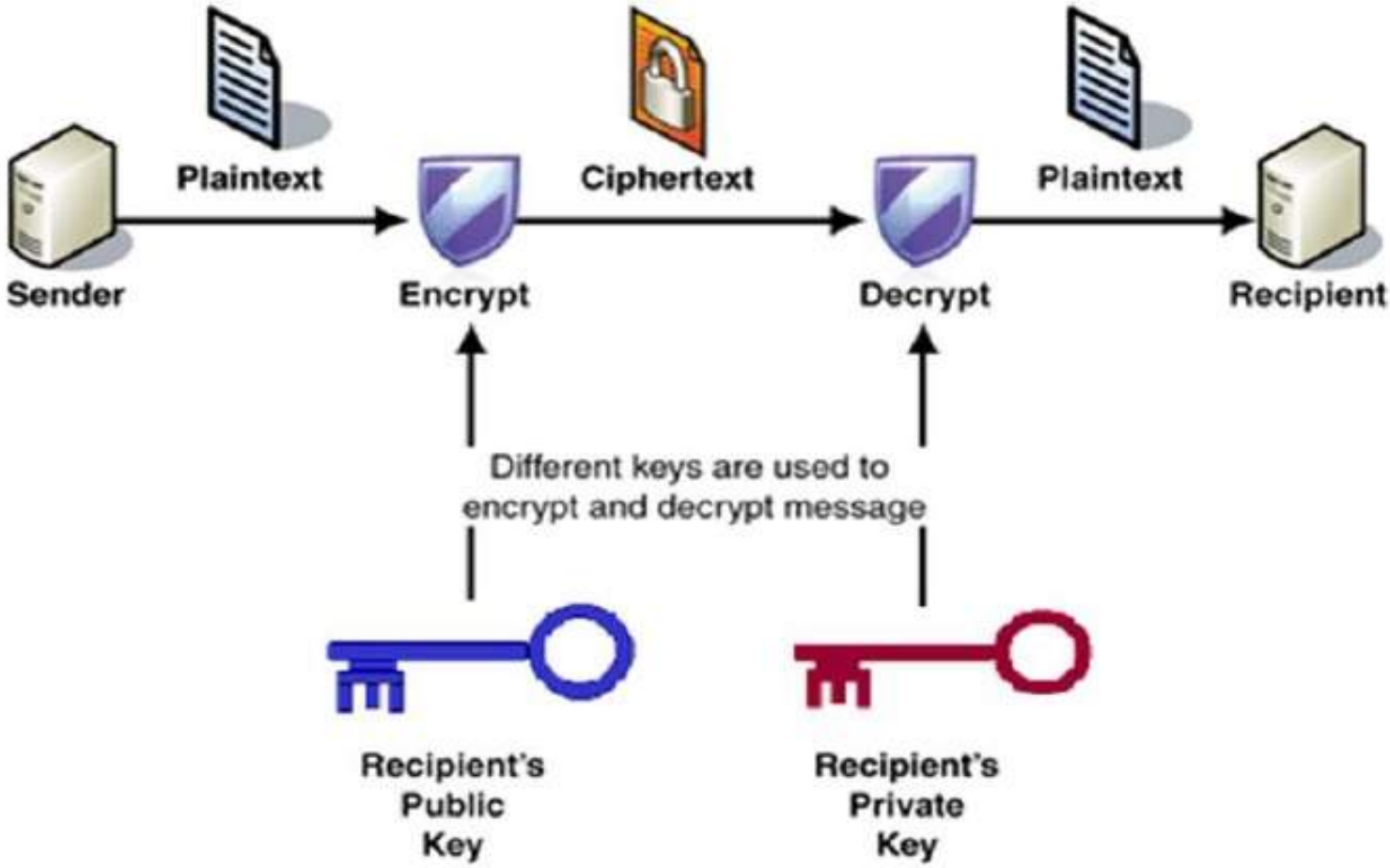
However, CTR mode has almost all advantages of CFB mode. In addition, it does not propagate error of transmission at all.

14_Public Key Encryption

Public Key Cryptography ⚓

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept.
Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations that were involved in the classified communication.
With the spread of more unsecure computer networks in the last few decades, a genuine need was felt to use cryptography at a larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.

The process of encryption and decryption is depicted in the following illustration –



What are the important properties of a public-key encryption scheme?

• Different keys are used for encryption and decryption. This is a property which sets this scheme differently than a symmetric encryption scheme.

• Each receiver possesses a unique decryption key, generally referred to as his private key.

• Receiver needs to publish an encryption key, referred to as his public key.

◇ Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by an adversary as the receiver. Generally, this type of cryptosystem involves a trusted third party which certifies that a particular public key belongs to a specific person or entity only.

• Encryption algorithm is complex enough to prohibit an attacker from deducing the plaintext from the ciphertext and the encryption (public) key.

• Though private and public keys are related mathematically, it is not feasible to calculate the private key from the public key. In fact, the intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

There are three types of Public Key Encryption schemes. We discuss them in following sections −

What are the three types of Public Key Encryption schemes?

1. RSA Cryptosystem

2.

ElGamal Cryptosystem ⚓

3.

Elliptic Curve Cryptography (ECC) ⚓

RSA Cryptosystem ⚓

Explain RSA Cryptosystem?

This cryptosystem is one the initial systems. It remains the most employed cryptosystem even today. The system was invented by three scholars Ron Rivest, Adi Shamir, and Len Adleman and hence, it is termed as RSA cryptosystem.

We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair ⚓

Each person or a party who desires to participate in communication using encryption

needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below −

◇ Generate the RSA modulus (n)

■ Select two large primes, p and q.

■ Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.

◇ Find Derived Number (e)

■ Number e must be greater than 1 and less than (p − 1)(q − 1).

■ There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are coprime.

◇ Form the public key

■ The pair of numbers (n, e) form the RSA public key and are made public.

■ Interestingly, though n is part of the public key, difficulty in factoring a large prime number ensures that an attacker cannot find in finite time the two primes (p & q) used to obtain n. This is the strength of RSA.

◇ Generate the private key

■ Private Key d is calculated from p, q, and e. For given n and e, there is a unique number d.

■ Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).

■ This relationship is written mathematically as follows −

$$ed = 1 \bmod (p - 1)(q - 1)$$

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

Example

An example of generating an RSA Key pair is given below. (For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high).

◇ Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.

◇ Select e = 5, which is a valid choice since there is no number that is common factor of 5 and (p − 1)(q − 1) = 6 × 12 = 72, except for 1.

◇ The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.

◇ Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29.

◇ Check that the d calculated is correct by computing −

$$de = 29 \times 5 = 145 = 1 \bmod 72$$

◇ Hence, the public key is (91, 5) and private keys is (91, 29).

Encryption and Decryption

Explain Encryption and Decryption of RSA system?

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.

Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n.

RSA Encryption

◇ Suppose the sender wishes to send some text message to someone whose public key is (n, e).

◇ The sender then represents the plaintext as a series of numbers less than n.

◇ To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

$C = P^e \bmod n$

◇ In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.

◇ Returning to our Key Generation example with plaintext P = 10, we get ciphertext C −

$C = 10^5 \bmod 91$

RSA Decryption ⚓

◇ The decryption process for RSA is also very straightforward. Suppose that the receiver of a public-key pair (n, e) has received a ciphertext C.

◇ Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

$\text{Plaintext} = C^d \bmod n$

◇ Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 −

$\text{Plaintext} = 82^{29} \bmod 91 = 10$

RSA Analysis ⚓

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is the most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

◇ Encryption Function − It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of private key d.

◇ Key Generation − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if

a technique for factoring efficiently is developed then RSA will no longer be safe.
The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.

ElGamal Cryptosystem

Explain Elgamal Cyptosystem:

Along with RSA, there are other public-key cryptosystems proposed. Many of them are based on different versions of the Discrete Logarithm Problem.
ElGamal cryptosystem, called Elliptic Curve Variant, is based on the Discrete Logarithm Problem. It derives strength from the assumption that the discrete logarithms cannot be found in a practical time frame for a given number, while the inverse operation of the power can be computed efficiently.
Let us go through a simple version of ElGamal that works with numbers modulo p. In the case of elliptic curve variants, it is based on quite different number systems.

Generation of ElGamal Key Pair

Each user of ElGamal cryptosystem generates the key pair through as follows −
◇ Choosing a large prime p. Generally a prime number of 1024 to 2048 bits length is chosen.

◇ Choosing a generator element g.

■ This number must be between 1 and p − 1, but cannot be any number.

■ It is a generator of the multiplicative group of integers modulo p. This means for every integer m co-prime to p, there is an integer k such that $g^k = a \bmod n$.
For example, 3 is the generator of group 5 ($Z_5 = \{1, 2, 3, 4\}$).

N
1
2
3
4

◇

Choosing the private key. The private key x is any number bigger than 1 and smaller than p−1.

◇ Computing part of the public key. The value y is computed from the parameters p, g and the private key x as follows −

$$y = g^x \bmod p$$

◇ Obtaining a Public key. The ElGamal public key consists of the three parameters (p, g, y). For example, suppose that p = 17 and that g = 6 (It can be confirmed that 6 is a generator of group Z_{17}). The private key x can be any number bigger than 1 and smaller than 71, so we choose x = 5. The value y is then computed as follows −

$$y = 6^5 \bmod 17 = 7$$

◇ Thus the private key is 62 and the public key is (17, 6, 7).

Encryption and Decryption⚓

The generation of an ElGamal key pair is comparatively simpler than the equivalent process for RSA. But the encryption and decryption are slightly more complex than RSA.

ElGamal Encryption⚓

Suppose sender wishes to send a plaintext to someone whose ElGamal public key is (p, g, y), then −

◇ Sender represents the plaintext as a series of numbers modulo p.

◇ To encrypt the first plaintext P, which is represented as a number modulo p. The encryption process to obtain the ciphertext C is as follows −

■ Randomly generate a number k;

■ Compute two values C1 and C2, where −

$$C1 = g^k \bmod p$$
$$C2 = (P * y^k) \bmod p$$

◇ Send the ciphertext C, consisting of the two separate values (C1, C2), sent together.

◇ Referring to our ElGamal key generation example given above, the plaintext P = 13 is encrypted as follows −

■ Randomly generate a number, say k = 10

■ Compute the two values C1 and C2, where −

$C1 = 6^{10} \bmod 17$
$C2 = (13 * 7^{10}) \bmod 17 = 9$

◇ Send the ciphertext C = (C1, C2) = (15, 9).

ElGamal Decryption⚓

◇ To decrypt the ciphertext (C1, C2) using private key x, the following two steps are taken −

■ Compute the modular inverse of $(C1)^x$ modulo p, which is $(C1)^{-x}$, generally referred to as the decryption factor.

■ Obtain the plaintext by using the following formula −

$C2 \times (C1)^{-x} \bmod p = \text{Plaintext}$

◇ In our example, to decrypt the ciphertext C = (C1, C2) = (15, 9) using private key x = 5, the decryption factor is

$15^{-5} \bmod 17 = 9$

◇ Extract plaintext P = (9 × 9) mod 17 = 13.

ElGamal Analysis

In the ElGamal system, each user has a private key x. and has three components of public key – prime modulus p, generator g, and public $Y = g^x \bmod p$. The strength of the ElGamal is based on the difficulty of discrete logarithm problems.

The secure key size is generally > 1024 bits. Today even 2048 bits long keys are used. On the processing speed front, Elgamal is quite slow, it is used mainly for key authentication protocols. Due to higher processing efficiency, Elliptic Curve variants of ElGamal are becoming increasingly popular.

Elliptic Curve Cryptography (ECC)

Explain Elliptic Curve Cryptography?

Elliptic Curve Cryptography (ECC) is a term used to describe a suite of cryptographic tools and protocols whose security is based on special versions of the discrete logarithm problem. It does not use numbers modulo p.

◇ ECC is based on sets of numbers that are associated with mathematical objects called elliptic curves. There are rules for adding and computing multiples of these numbers, just as there are for numbers modulo p.

ECC includes variants of many cryptographic schemes that were initially designed for modular numbers such as ElGamal encryption and Digital Signature Algorithm.

It is believed that the discrete logarithm problem is much harder when applied to points on an elliptic curve. This prompts switching from numbers modulo p to points on an elliptic curve. Also an equivalent security level can be obtained with shorter keys if we use elliptic curve-based variants.

The shorter keys result in two benefits –

◇ Ease of key management

◇ Efficient computation

These benefits make elliptic-curve-based variants of encryption schemes highly attractive for applications where computing resources are constrained.

RSA and ElGamal Schemes – A Comparison

Let us briefly compare the RSA and ElGamal schemes on the various aspects.

RSA
It is more efficient for encryption.
It is less efficient for decryption.
For a particular security level, lengthy keys are required in RSA.
It is widely accepted and used.

18_Cryptography Digital signatures

What are Digital Signatures?

• Digital signatures are the public-key primitives of message authentication.

◇ In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message.
◇ Similarly, a digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by the receiver as well as any third party.
◇ Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

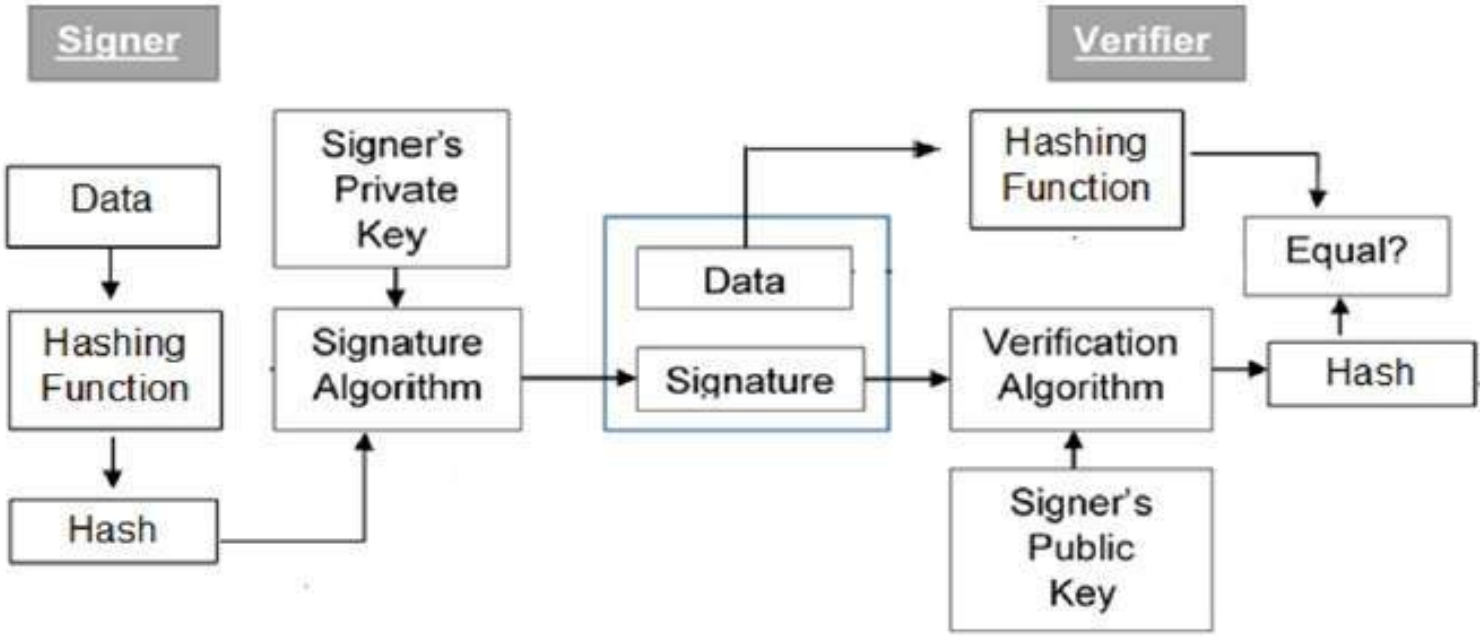
In the real world, the receiver of a message needs assurance that the message belongs to the sender and he should not be able to repudiate the origin of that message. This requirement is very crucial in business applications, since the likelihood of a dispute over exchanged data is very high.

Model of Digital Signature ⚓

Explain the Model of Digial Signature:

◇ As mentioned earlier, the digital signature scheme is based on public key cryptography.

◇ The model of digital signature scheme is depicted in the following illustration −



Explain the digital signature process:

The following points explain the entire process in detail –

◇ Each person adopting this scheme has a public-private key pair.

◇ Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.

◇ Signer feeds data to the hash function and generates a hash of data.

◇ Hash value and signature key are then fed to the signature algorithm which produces the digital signature on a given hash. Signature is appended to the data and then both are sent to the verifier.

◇ Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.

◇ Verifier also runs the same hash function on received data to generate hash value.

◇ For verification, this hash value and output of the verification algorithm are compared. Based on the comparison result, the verifier decides whether the digital signature is valid.

◇ Since a digital signature is created by the 'private' key of the signer and no one else can have this key; the signer cannot repudiate signing the data in future.

It should be noticed that instead of signing data directly by signing algorithm, usually a hash of data is created. Since the hash of data is a unique representation of data, it is

sufficient to sign the hash in place of data. The most important reason for using hash instead of data directly for signing is efficiency of the scheme.

Let us assume RSA is used as the signing algorithm. As discussed in the public key encryption chapter, the encryption/signing process using RSA involves modular exponentiation.
Signing large data through modular exponentiation is computationally expensive and time consuming. The hash of the data is a relatively small digest of the data, hence signing a hash is more efficient than signing the entire data.

Importance of Digital Signature ⚓

What is the Importance of using Digital Signatures?
◇ Out of all cryptographic primitives, the digital signature using public key cryptography is considered as a very important and useful tool to achieve information security.

Apart from the ability to provide non-repudiation of messages, the digital signature also provides message authentication and data integrity. Let us briefly see how this is achieved by the digital signature −
◇ Message authentication − When the verifier validates the digital signature using the public key of a sender, he is assured that the signature has been created only by the sender who possesses the corresponding secret private key and no one else.

◇ Data Integrity − In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.

◇ Non-repudiation − Since it is assumed that only the signer has the knowledge of the signature key, he can only create a unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

By adding public-key encryption to digital signature scheme, we can create a cryptosystem that can provide the four essential elements of security namely − Privacy, Authentication, Integrity, and Non-repudiation.

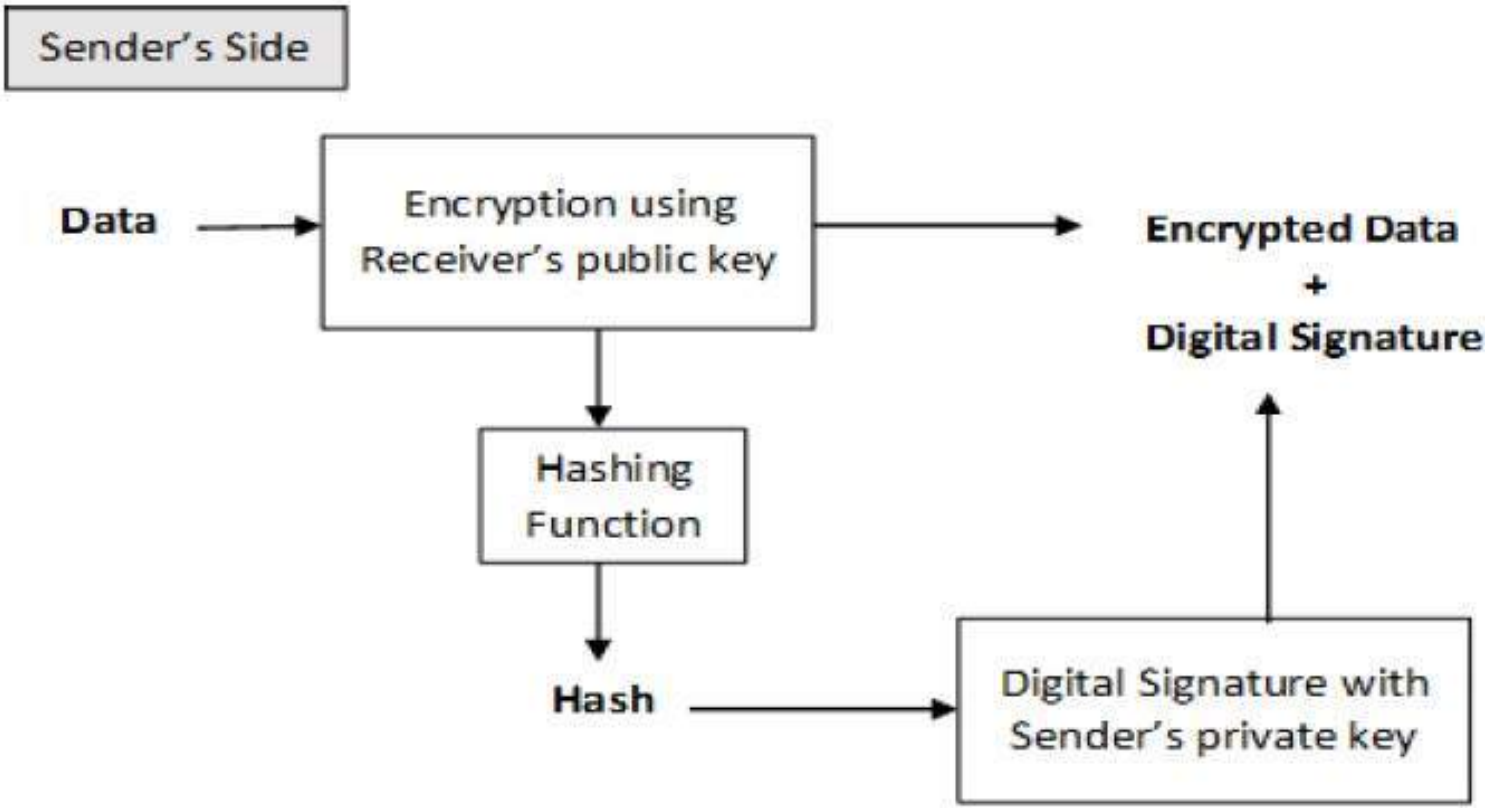
Encryption with Digital Signature ⚓

In many digital communications, it is desirable to exchange encrypted messages rather

than plaintext to achieve confidentiality. In a public key encryption scheme, a public (encryption) key of sender is available in open domain, and hence anyone can spoof his identity and send any encrypted message to the receiver.

This makes it essential for users employing PKC for encryption to seek digital signatures along with encrypted data to be assured of message authentication and non-repudiation. This can be achieved by combining digital signatures with encryption schemes. Let us briefly discuss how to achieve this requirement. There are two possibilities, sign-then-encrypt and encrypt-then-sign.

However, the cryptosystem based on sign-then-encrypt can be exploited by the receiver to spoof the identity of the sender and send that data to a third party. Hence, this method is not preferred. The process of encrypt-then-sign is more reliable and widely adopted. This is depicted in the following illustration –



The receiver, after receiving the encrypted data and signature on it, first verifies the signature using the sender's public key. After ensuring the validity of the signature, he then retrieves the data through decryption using his private key.

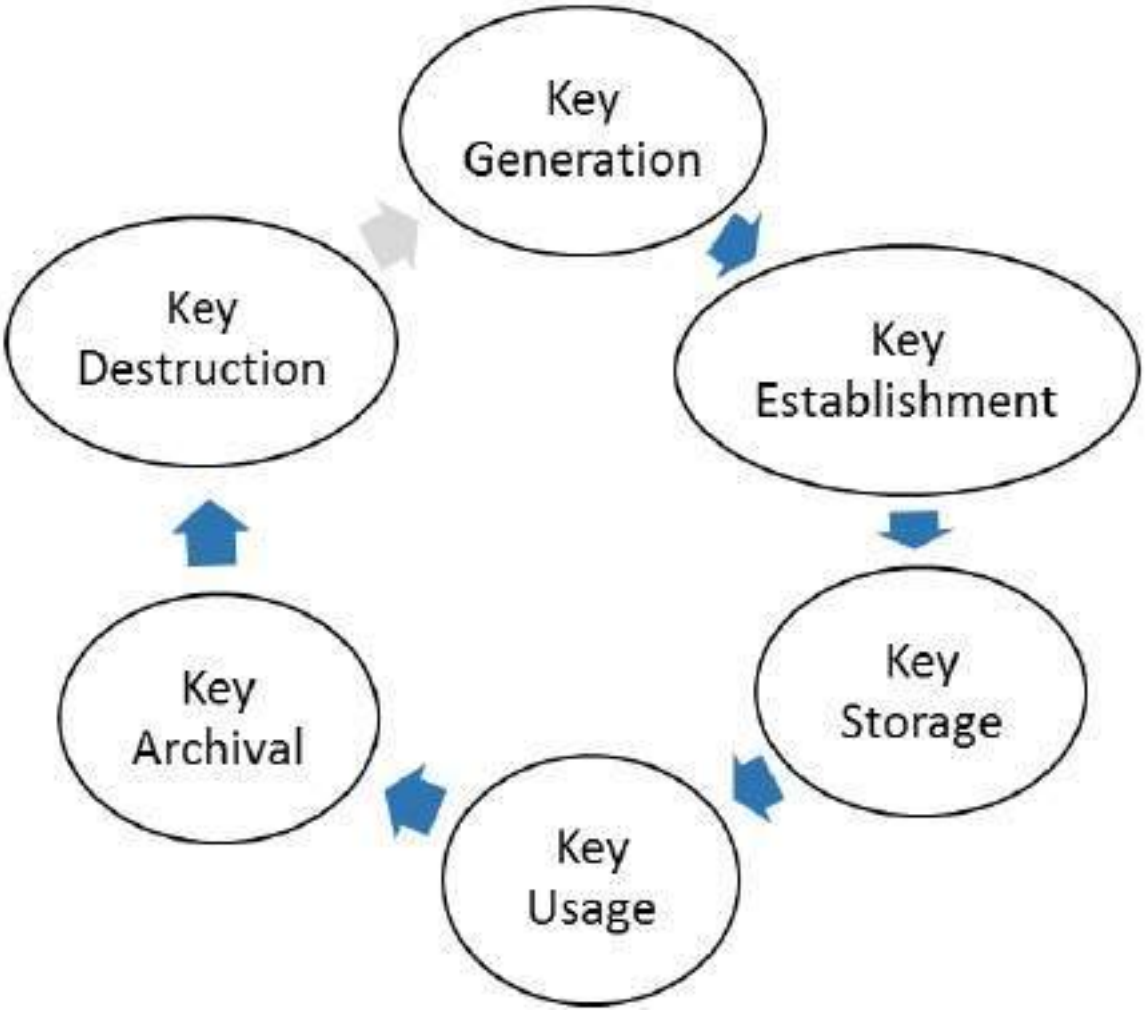
19_Public Key Infrastructure

The most distinct feature of Public Key Infrastructure (PKI) is that it uses a pair of keys to achieve the underlying security service. The key pair comprises private key and public key. Since the public keys are in open domain, they are likely to be abused. It is, thus, necessary to establish and maintain some kind of trusted infrastructure to manage these keys.

Key Management⚓

It goes without saying that the security of any cryptosystem depends upon how securely its keys are managed. Without secure procedures for the handling of cryptographic keys, the benefits of the use of strong cryptographic schemes are potentially lost.
It is observed that cryptographic schemes are rarely compromised through weaknesses in their design. However, they are often compromised through poor key management.
There are some important aspects of key management which are as follows −
• Cryptographic keys are nothing but special pieces of data. Key management refers to the secure administration of cryptographic keys.

• Key management deals with entire key lifecycle as depicted in the following illustration −



◇ There are two specific requirements of key management for public key cryptography.

■ Secrecy of private keys. Throughout the key lifecycle, secret keys must remain secret from all parties except those who are owner and are authorized to use them.

■ Assurance of public keys. In public key cryptography, the public keys are in open domain and seen as public pieces of data. By default there are no assurances of whether a public key is correct, with whom it can be associated, or what it can be used for. Thus key management of public keys needs to focus much more explicitly on assurance of purpose of public keys.

The most crucial requirement of 'assurance of public key' can be achieved through the public-key infrastructure (PKI), a key management systems for supporting public-key cryptography.

Public Key Infrastructure (PKI) ⚓

PKI provides assurance of public keys. It provides the identification of public keys and their distribution. Anatomy of PKI comprises the following components.

◇ Public Key Certificate, commonly referred to as 'digital certificate'.

◇ Private Key tokens.

◇ Certification Authority.

◇ Registration Authority.

◇ Certificate Management System.

Digital Certificate ⚓

For analogy, a certificate can be considered as the ID card issued to the person. People use ID cards such as a driver's license, passport to prove their identity. A digital certificate does the same basic thing in the electronic world, but with one difference.

Digital Certificates are not only issued to people but they can be issued to computers, software packages or anything else that needs to prove the identity in the electronic world.

◇ Digital certificates are based on the ITU standard X.509 which defines a standard certificate format for public key certificates and certification validation. Hence digital certificates are sometimes also referred to as X.509 certificates.

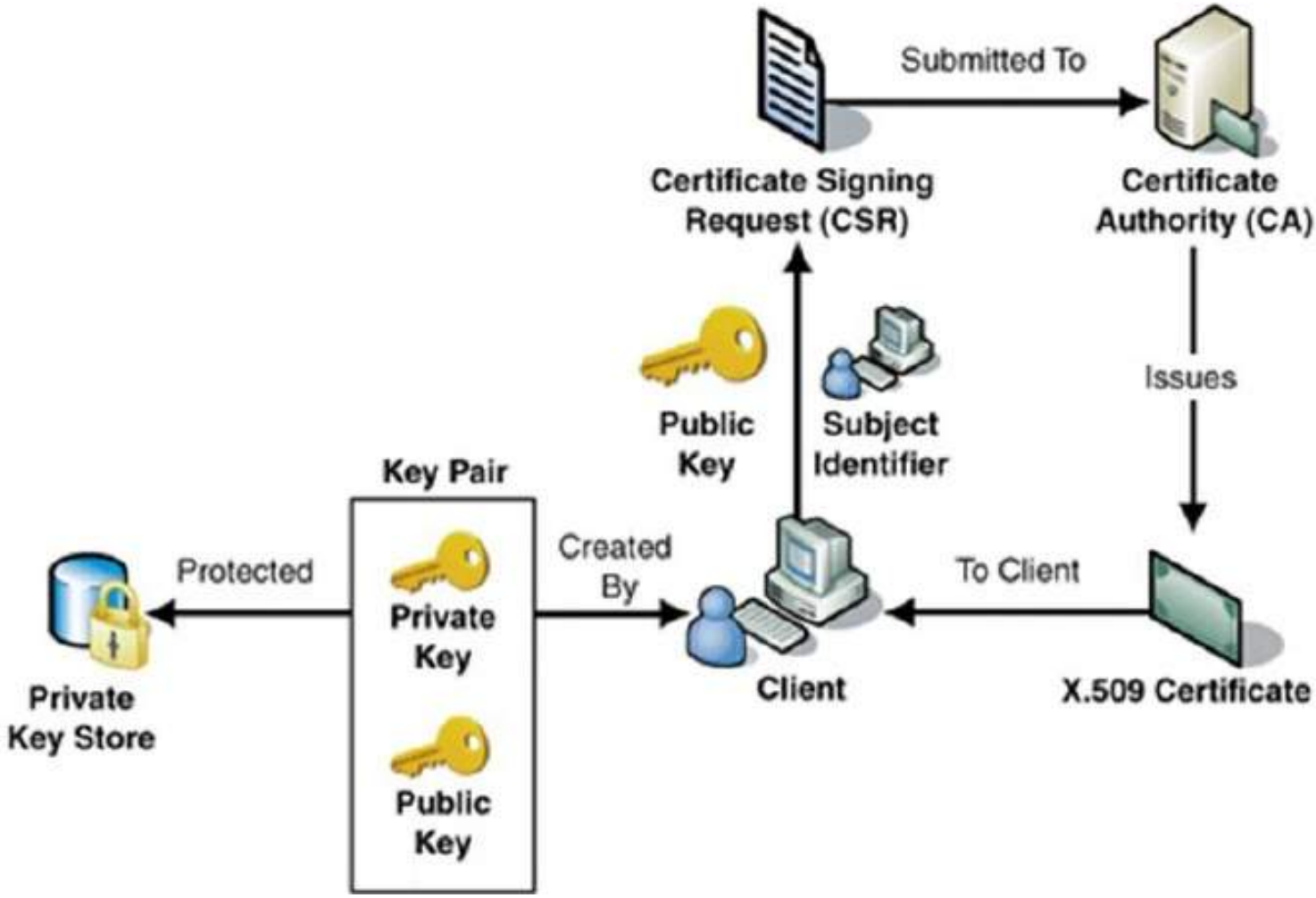
Public keys pertaining to the user client are stored in digital certificates by The Certification Authority (CA) along with other relevant information such as client information, expiration

date, usage, issuer etc.

◇ CA digitally signs this entire information and includes digital signature in the certificate.

◇ Anyone who needs the assurance about the public key and associated information of the client, he carries out the signature validation process using CA's public key. Successful validation assures that the public key given in the certificate belongs to the person whose details are given in the certificate.

The process of obtaining Digital Certificate by a person/entity is depicted in the following illustration.



As shown in the illustration, the CA accepts the application from a client to certify his public key. The CA, after duly verifying the identity of the client, issues a digital certificate to that client.

Certifying Authority (CA) ⚓

As discussed above, the CA issues certificates to a client and assists other users to verify the certificate. The CA takes responsibility for identifying correctly the identity of the client asking for a certificate to be issued, and ensures that the information contained within the

certificate is correct and digitally signs it.

Key Functions of CA

The key functions of a CA are as follows −

◇ Generating key pairs − The CA may generate a key pair independently or jointly with the client.

◇ Issuing digital certificates − The CA could be thought of as the PKI equivalent of a passport agency − the CA issues a certificate after client provides the credentials to confirm his identity. The CA then signs the certificate to prevent modification of the details contained in the certificate.

◇ Publishing Certificates − The CA need to publish certificates so that users can find them. There are two ways of achieving this. One is to publish certificates in the equivalent of an electronic telephone directory. The other is to send your certificate out to those people you think might need it by one means or another.

◇ Verifying Certificates − The CA makes its public key available in environment to assist verification of his signature on clients' digital certificate.

◇ Revocation of Certificates − At times, CA revokes the certificate issued due to some reason such as compromise of private key by user or loss of trust in the client. After revocation, CA maintains the list of all revoked certificate that is available to the environment.

Classes of Certificates

There are four typical classes of certificate −

◇ Class 1 − These certificates can be easily acquired by supplying an email address.

◇ Class 2 − These certificates require additional personal information to be supplied.

◇ Class 3 − These certificates can only be purchased after checks have been made about the requestor's identity.

◇ Class 4 − They may be used by governments and financial organizations needing very high levels of trust.

Registration Authority (RA)

CA may use a third-party Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to confirm their identity. The RA may appear to the client as a CA, but they do not actually sign the certificate that is issued.

Certificate Management System (CMS)

It is the management system through which certificates are published, temporarily or permanently suspended, renewed, or revoked. Certificate management systems do not normally delete certificates because it may be necessary to prove their status at a point in time, perhaps for legal reasons. A CA along with associated RA runs certificate management systems to be able to track their responsibilities and liabilities.

Private Key Tokens

While the public key of a client is stored on the certificate, the associated secret private key can be stored on the key owner's computer. This method is generally not adopted. If an attacker gains access to the computer, he can easily gain access to private key. For this reason, a private key is stored on secure removable storage token access to which is protected through a password.
Different vendors often use different and sometimes proprietary storage formats for storing keys. For example, Entrust uses the proprietary .epf format, while Verisign, GlobalSign, and Baltimore use the standard .p12 format.

Hierarchy of CA

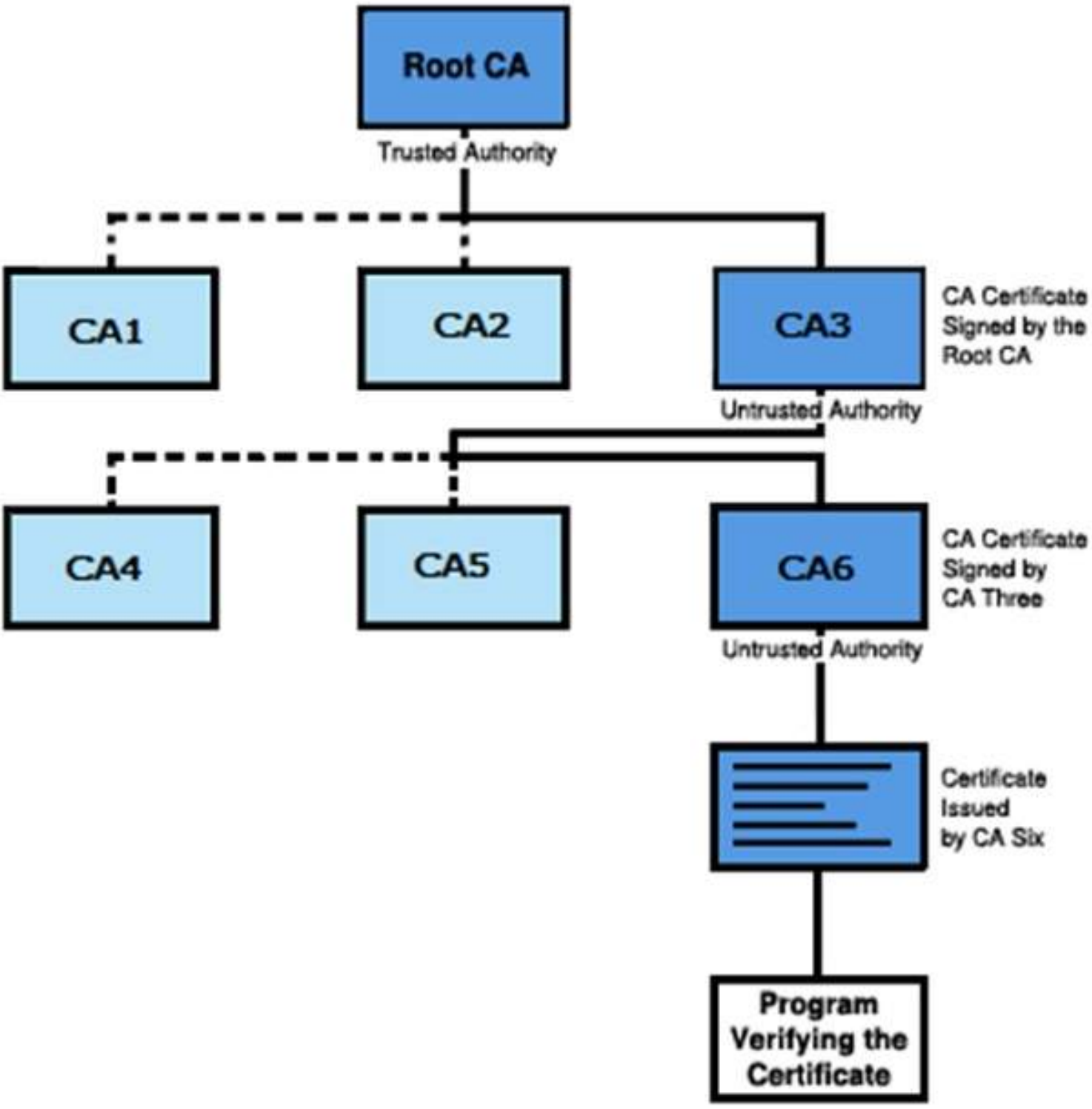
With vast networks and requirements of global communications, it is practically not feasible to have only one trusted CA from whom all users obtain their certificates. Secondly, availability of only one CA may lead to difficulties if CA is compromised.
In such case, the hierarchical certification model is of interest since it allows public key certificates to be used in environments where two communicating parties do not have trust relationships with the same CA.
◇ The root CA is at the top of the CA hierarchy and the root CA's certificate is a self-signed certificate.

◇ The CAs, which are directly subordinate to the root CA (For example, CA1 and CA2) have CA certificates that are signed by the root CA.

◇ The CAs under the subordinate CAs in the hierarchy (For example, CA5 and CA6) have their CA certificates signed by the higher-level subordinate CAs.

Certificate authority (CA) hierarchies are reflected in certificate chains. A certificate chain traces a path of certificates from a branch in the hierarchy to the root of the hierarchy. The following illustration shows a CA hierarchy with a certificate chain leading from an entity certificate through two subordinate CA certificates (CA6 and CA3) to the CA

certificate for the root CA.



Verifying a certificate chain is the process of ensuring that a specific certificate chain is valid, correctly signed, and trustworthy. The following procedure verifies a certificate chain, beginning with the certificate that is presented for authentication –

◇ A client whose authenticity is being verified supplies his certificate, generally along with the chain of certificates up to Root CA.

◇ Verifier takes the certificate and validates by using public key of issuer. The issuer's public key is found in the issuer's certificate which is in the chain next to client's certificate.

◇ Now if the higher CA who has signed the issuer's certificate, is trusted by the verifier,

verification is successful and stops here.

◇ Else, the issuer's certificate is verified in a similar manner as done for client in above steps. This process continues till either trusted CA is found in between or else it continues till Root CA.

20_Cryptography Benefits & Drawbacks

Nowadays, the networks have gone global and information has taken the digital form of bits and bytes. Critical information now gets stored, processed and transmitted in digital form on computer systems and open communication channels.
Since information plays such a vital role, adversaries are targeting the computer systems and open communication channels to either steal the sensitive information or to disrupt the critical information system.
Modern cryptography provides a robust set of techniques to ensure that the malevolent intentions of the adversary are thwarted while ensuring the legitimate users get access to information. Here in this chapter, we will discuss the benefits that we draw from cryptography, its limitations, as well as the future of cryptography.

Cryptography – Benefits ⚓

Cryptography is an essential information security tool. It provides the four most basic services of information security −
• Confidentiality − Encryption technique can guard the information and communication from unauthorized revelation and access of information.

• Data Integrity − The cryptographic hash functions are playing a vital role in assuring the users about the data integrity.

• Authentication − The cryptographic techniques such as MAC and digital signatures can protect information against spoofing and forgeries.

• Non-repudiation − The digital signature provides the non-repudiation service to guard against the dispute that may arise due to denial of a passing message by the sender.

All these fundamental services offered by cryptography have enabled the conduct of business over the networks using the computer systems in extremely efficient and effective manner.

Cryptography – Drawbacks ⚓

Apart from the four fundamental elements of information security, there are other issues that affect the effective use of information −
◇ A strongly encrypted, authentic, and digitally signed information can be difficult to access even for a legitimate user at a crucial time of decision-making. The network or the computer system can be attacked and rendered non-functional by an intruder.

◇ High availability, one of the fundamental aspects of information security, cannot be

ensured through the use of cryptography. Other methods are needed to guard against the threats such as denial of service or complete breakdown of the information system.

◇ Another fundamental need of information security of selective access control also cannot be realized through the use of cryptography. Administrative controls and procedures are required to be exercised for the same.

◇ Cryptography does not guard against the vulnerabilities and threats that emerge from the poor design of systems, protocols, and procedures. These need to be fixed through proper design and setting up of a defensive infrastructure.

◇ Cryptography comes at cost. The cost is in terms of time and money −

■ Addition of cryptographic techniques in the information processing leads to delay.

■ The use of public key cryptography requires setting up and maintenance of public key infrastructure requiring the handsome financial budget.

◇ The security of cryptographic technique is based on the computational difficulty of mathematical problems. Any breakthrough in solving such mathematical problems or increasing the computing power can render a cryptographic technique vulnerable.

Future of Cryptography⚓

Elliptic Curve Cryptography (ECC) has already been invented but its advantages and disadvantages are not yet fully understood. ECC allows to perform encryption and decryption in a drastically lesser time, thus allowing a higher amount of data to be passed with equal security. However, as other methods of encryption, ECC must also be tested and proven secure before it is accepted for governmental, commercial, and private use.
Quantum computation is a new phenomenon. While modern computers store data using a binary format called a "bit" in which a "1" or a "0" can be stored; a quantum computer stores data using a quantum superposition of multiple states. These multiple valued states are stored in "quantum bits" or "qubits". This allows the computation of numbers to be several orders of magnitude faster than traditional transistor processors.
To comprehend the power of quantum computers, consider RSA-640, a number with 193 digits, which can be factored by eighty 2.2GHz computers over the span of 5 months, one quantum computer would factor in less than 17 seconds. Numbers that would typically take billions of years to compute could only take a matter of hours or even minutes with a fully developed quantum computer.
In view of these facts, modern cryptography will have to look for computationally hard

problems or devise completely new techniques of archiving the goals presently served by modern cryptography.

15_Data Integrity in Cryptography

Until now, we discussed the use of symmetric and public key schemes to achieve the confidentiality of information. With this chapter, we begin our discussion on different cryptographic techniques designed to provide other security services.
The focus of this chapter is on data integrity and cryptographic tools used to achieve the same.

Threats to Data Integrity ⚓

When sensitive information is exchanged, the receiver must have the assurance that the message has come intact from the intended sender and is not modified inadvertently or otherwise. There are two different types of data integrity threats, namely passive and active.

What are the two different types of data integrity threats?

1. Passive threats
2. Active Threats

Explain Passive threats ⚓

Passive Threats ⚓

This type of threat exists due to accidental changes in data.
• These data errors are likely to occur due to noise in a communication channel. Also, the data may get corrupted while the file is stored on a disk.

• Error-correcting codes and simple checksums like Cyclic Redundancy Checks (CRCs) are used to detect the loss of data integrity. In these techniques, a digest of data is computed mathematically and appended to the data.

Explain Active Threats ⚓

In this type of threat, an attacker can manipulate the data with malicious intent.
◇ At simplest level, if data is without digest, it can be modified without detection. The system can use techniques of appending CRC to data for detecting any active modification.
◇ At a higher level of threat, an attacker may modify data and try to derive new digest for modified data from existing digest. This is possible if the digest is computed using simple mechanisms such as CRC.
◇ Security mechanisms such as Hash functions are used to tackle the active modification

threats.

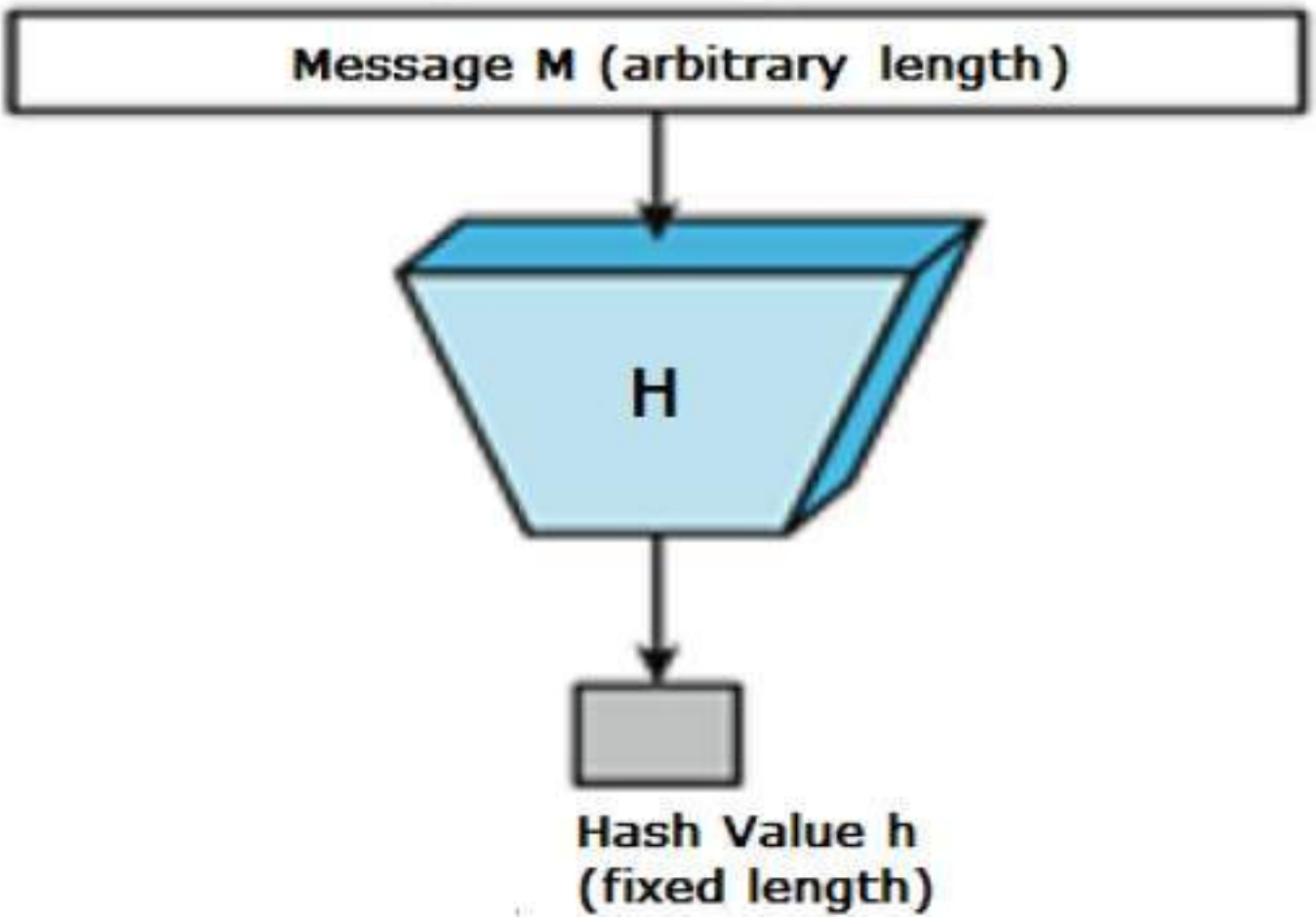
16_Cryptography Hash functions

What is a hash function?

• A hash function is a mathematical function that converts a numerical input value into another compressed numerical value.

• The input to the hash function is of arbitrary length but output is always of fixed length.

Values returned by a hash function are called message digest or simply hash values. The following picture illustrated hash function −



What are the features of a hash function?

Features of Hash Functions ⚓

The typical features of hash functions are −

• Fixed Length Output (Hash Value)

• Hash function converts data of arbitrary length to a fixed length. This process is often referred to as hashing the data.

• In general, the hash is much smaller than the input data, hence hash functions are sometimes called compression functions.

• Since a hash is a smaller representation of a larger data, it is also referred to as a digest.

• Hash function with n bit output is referred to as an n-bit hash function. Popular hash

functions generate values between 160 and 512 bits.
• Efficiency of Operation
• Generally for any hash function h with input x, computation of h(x) is a fast operation.
• Computationally hash functions are much faster than a symmetric encryption.

⚓

What are the properties of Hash functions? ⚓

1. Pre-Image Resistance
2. Second Pre-Image resistance
3. Collision Resistance

Explain the properties of Hash functions:

◇ Pre-Image Resistance

• This property means that it should be computationally hard to reverse a hash function.
• In other words, if a hash function h produces a hash value z, then it should be a difficult process to find any input value x that hashes to z.
• This property protects against an attacker who only has a hash value and is trying to find the input.

◇ Second Pre-Image Resistance

• This property means that with a given input and its hash, it should be hard to find a different input with the same hash.
• In other words, if a hash function h for an input x produces hash value h(x), then it should be difficult to find any other input value y such that h(y) = h(x).
• This property of the hash function protects against an attacker who has an input value and its hash, and wants to substitute a different value as legitimate value in place of original input value.

◇ Collision Resistance

• This property means it should be hard to find two different inputs of any length that result in the same hash.
• This property is also referred to as a collision free hash function.
• In other words, for a hash function h, it is hard to find any two different inputs x and y such that h(x) = h(y).

• Since, a hash function is a compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find.
• This property makes it very difficult for an attacker to find two input values with the same hash.
• Also, if a hash function is collision-resistant then it is second pre-image resistant.

Design of Hashing Algorithms ⚓

Understand the difference between hash function and algorithm correctly.

Hash Function :

◇ The hash function generates a hash value by operating on two blocks of fixed-length binary data.

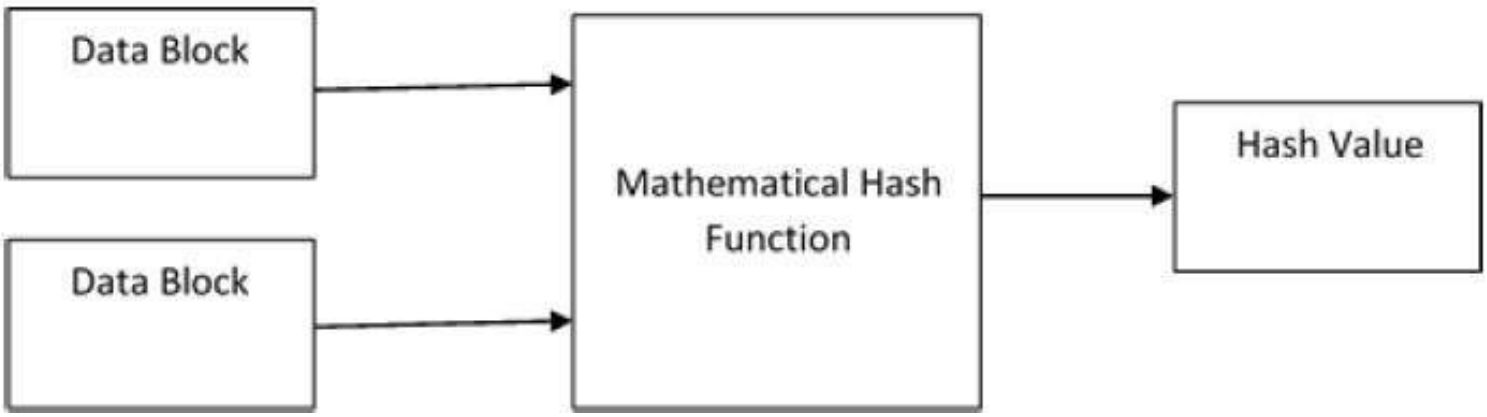
Hashing Algorithm:

◇ Hashing algorithm is a process for using the hash function, specifying how the message will be broken up and how the results from previous message blocks are chained together.

◇ At the heart of a hashing is a mathematical function that operates on two fixed-size blocks of data to create a hash code.
◇ This hash function forms part of the hashing algorithm.

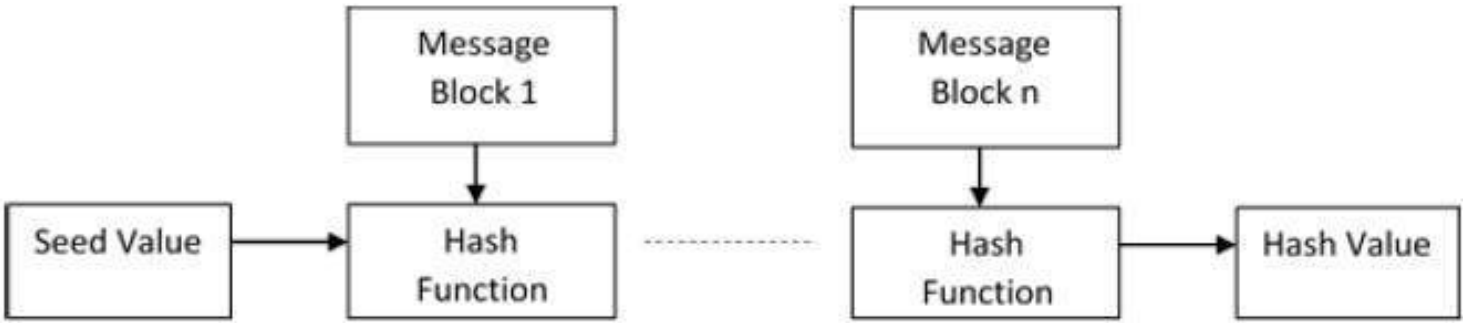
Explain how hash function operates

The size of each data block varies depending on the algorithm. Typically the block sizes are from 128 bits to 512 bits. The following illustration demonstrates hash function −



◇ Hashing algorithm involves rounds of the above hash function like a block cipher. Each round takes an input of a fixed size, typically a combination of the most recent message block and the output of the last round.

◇ This process is repeated for as many rounds as are required to hash the entire message. Schematic of hashing algorithm is depicted in the following illustration



What is the Avalanche effect in hashing?

Avalanche Effect:

◇ The hash value of the first message block becomes an input(seed) to the second hash operation, output of which alters the result of the third operation, and so on. This effect is known as an avalanche effect of hashing.

◇ Avalanche effect results in different hash values for two messages that differ by even a single bit of data.

Understand the difference between hash function and algorithm correctly. The hash function generates a hash code by operating on two blocks of fixed-length binary data. Hashing algorithm is a process for using the hash function, specifying how the message

will be broken up and how the results from previous message blocks are chained together.

What are the popular hash functions?

Popular Hash Functions

Let us briefly see some popular hash functions −

1. Message Digest (MD)
1. MD2, MD4, MD5, MD6

2. SHA (Secure Hash Function)
1. SHA-0, SHA-1, SHA-2, SHA-3

2. RIPEMD
1. RIPEMD, RIPEMD-128 and RIPEMD-160, RIPEMD-256, RIPEMD-320

2. Whirlpool
1. WHIRLPOOL-0, WHIRLPOOL-T, and WHIRLPOOL

Message Digest (MD)

MD5 was the most popular hash function for quite some years.

◇ The MD family comprises hash functions MD2, MD4, MD5 and MD6. It was adopted as Internet Standard RFC 1321. It is a 128-bit hash function.

◇ MD5 digests have been widely used in the software world to provide assurance about integrity of transferred files. For example, file servers often provide a pre-computed MD5 checksum for the files, so that a user can compare the checksum of the downloaded file to it.

◇ In 2004, collisions were found in MD5. An analytical attack was reported to be successful only in an hour by using a computer cluster. This collision attack resulted in compromised MD5 and hence it is no longer recommended for use.

Secure Hash Function (SHA)

Family of SHA comprises four SHA algorithms; SHA-0, SHA-1, SHA-2, and SHA-3. Though from the same family, they are structurally different.

◇ The original version is SHA-0, a 160-bit hash function, was published by the National Institute of Standards and Technology (NIST) in 1993. It had few weaknesses and did not become very popular. Later in 1995, SHA-1 was designed to correct alleged weaknesses of

SHA-0.

◇ SHA-1 is the most widely used of the existing SHA hash functions. It is employed in several widely used applications and protocols including Secure Socket Layer (SSL) security.

◇ In 2005, a method was found for uncovering collisions for SHA-1 within a practical time frame making long-term employability of SHA-1 doubtful.

◇ The SHA-2 family has four further SHA variants, depending on the number of bits in their hash value.

■ SHA-224,

■ SHA-256,

■ SHA-384,

■ SHA-512.

■ No successful attacks have yet been reported on SHA-2 hash function.

◇ Though SHA-2 is a strong hash function. Though significantly different, its basic design still follows the design of SHA-1. Hence, NIST called for new competitive hash function designs.

◇ SHA-3

■ In October 2012, the NIST chose the Keccak algorithm as the new SHA-3 standard. Keccak offers many benefits, such as efficient performance and good resistance for attacks.

RIPEMD ⚓

The RIPEMD is an acronym for RACE Integrity Primitives Evaluation Message Digest. This set of hash functions was designed by the open research community and generally known as a family of European hash functions.
◇ The set includes RIPEMD, RIPEMD-128, and RIPEMD-160. There also exist 256, and 320-bit versions of this algorithm.

◇ Original RIPEMD (128 bit) is based upon the design principles used in MD4 and found to provide questionable security. RIPEMD 128-bit version came as a quick fix replacement to overcome vulnerabilities on the original RIPEMD.

◇ RIPEMD-160 is an improved version and the most widely used version in the family. The 256 and 320-bit versions reduce the chance of accidental collision, but do not have higher levels of security as compared to RIPEMD-128 and RIPEMD-160 respectively.

Whirlpool⚓

This is a 512-bit hash function.

◇ It is derived from the modified version of Advanced Encryption Standard (AES). One of the designers was Vincent Rijmen, a co-creator of the AES.

◇ Three versions of Whirlpool have been released; namely WHIRLPOOL-0, WHIRLPOOL-T, and WHIRLPOOL.

applications of hash functions

What are the applications of hash functions?

1. Password Storage
2. Data Integrity Check

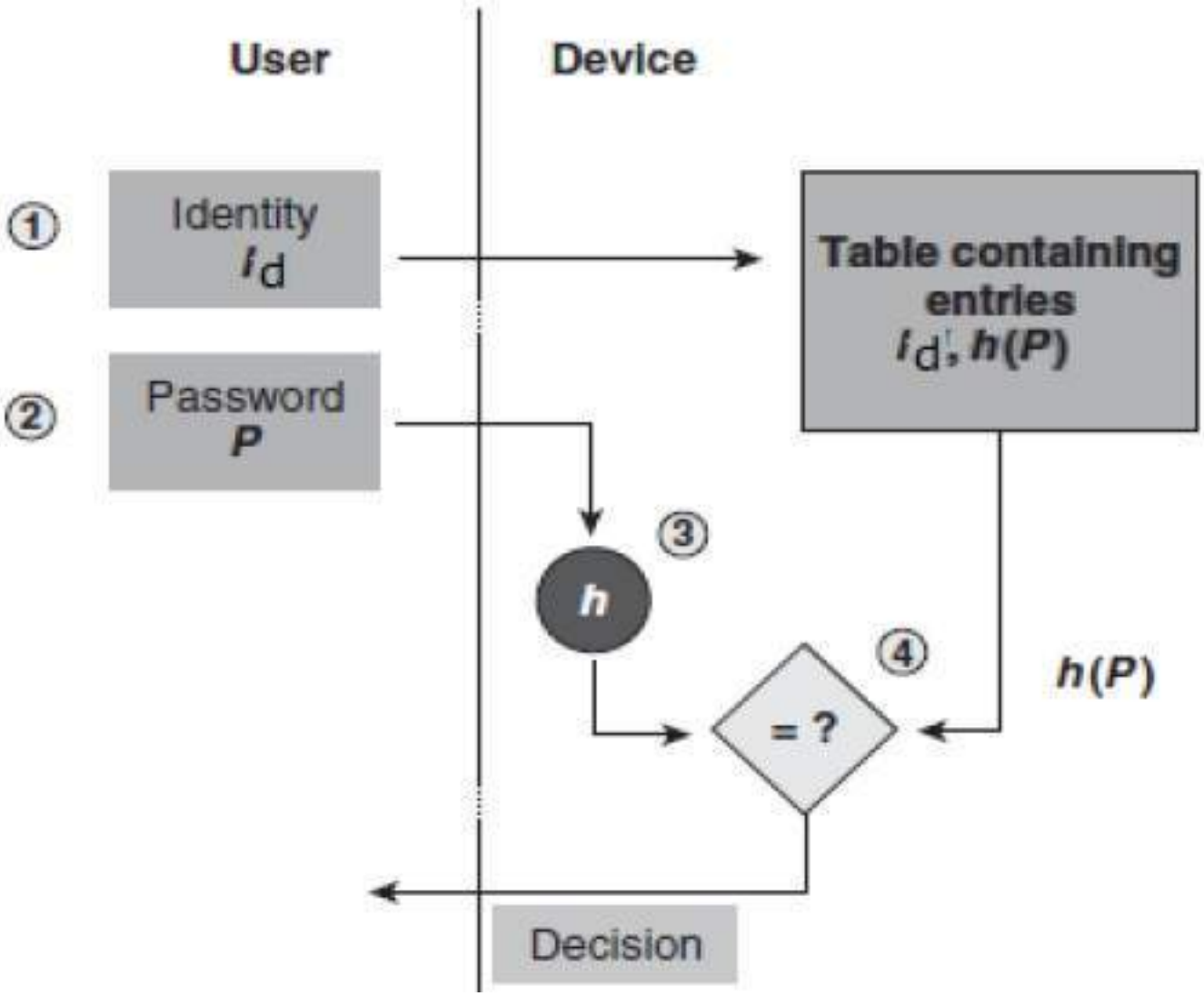
Password Storage ⚓

Hash functions provide protection to password storage.

◇ Instead of storing passwords in clear or encrypted text, mostly all logon processes store the hash values of passwords in the file.

◇ The Password file consists of a table of pairs which are in the form (user id, h(P)).

◇ The process of logon is depicted in the following illustration –



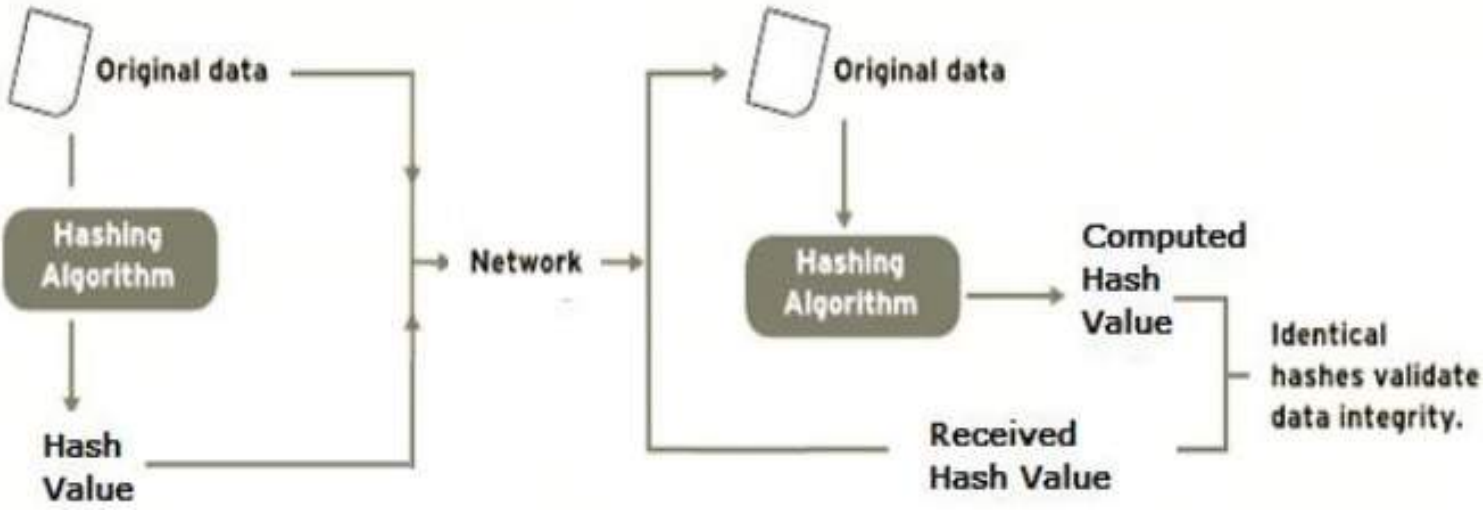
◇ An intruder can only see the hashes of passwords, even if he accessed the password.

◇ He can neither logon using hash nor can he derive the password from hash value since hash function possesses the property of pre-image resistance.

Data Integrity Check ⚓

◇ Data integrity check is a most common application of the hash functions.
◇ Hashing is used to generate the checksums on data files.
◇ This application provides assurance to the user about correctness of the data.

The process is depicted in the following illustration −



◇ The integrity check helps the user to detect any changes made to the original file.
◇ It however, does not provide any assurance about originality. The attacker, instead of modifying file data, can change the entire file and compute all together new hash and send it to the receiver.
◇ This integrity check application is useful only if the user is sure about the originality of the file.

Authentication

17_Message Authentication

Why is message authentication required?

• In the last chapter, we discussed the data integrity threats and the use of hashing techniques to detect if any modification attacks have taken place on the data.
• Another type of threat that exists for data is the lack of message authentication.
• In this threat, the user is not sure about the originator of the message.
• Message authentication can be provided using the cryptographic techniques that use secret keys as done in case of encryption.

Message Authentication Code (MAC)

What is a MAC Algorithm?

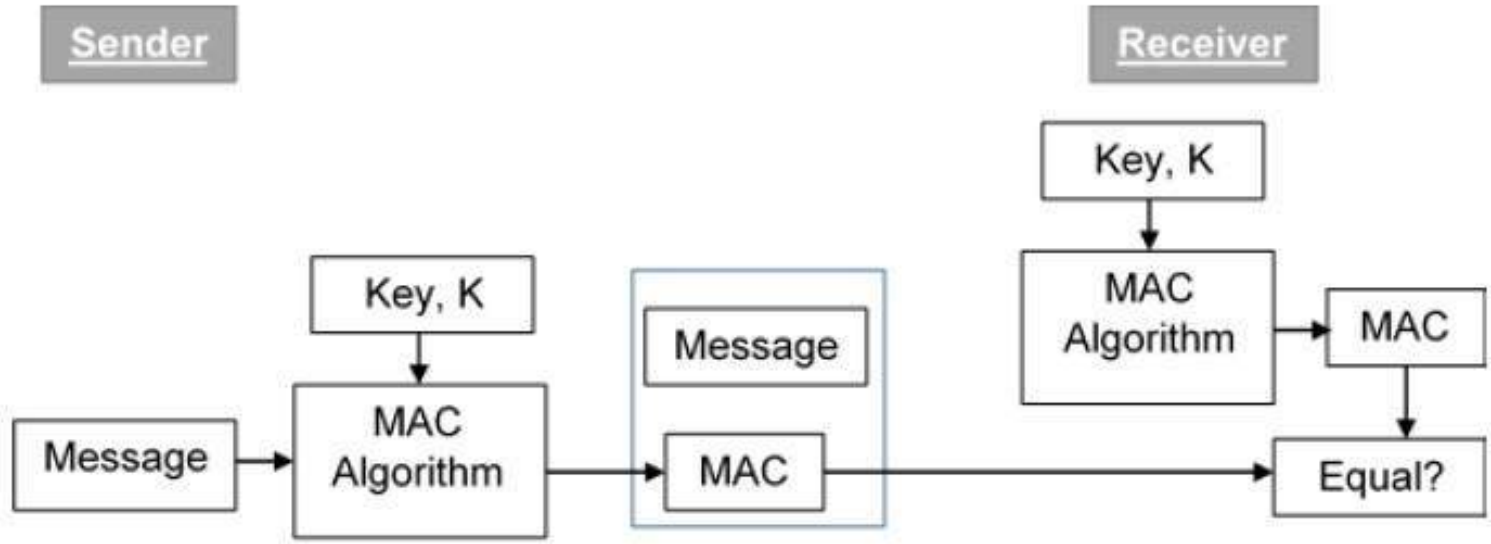
◇ MAC algorithm is a symmetric key cryptographic technique to provide message authentication.

■ For establishing MAC processes, the sender and receiver share a symmetric key K.

◇ Essentially, a MAC is an encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication.

Explain the MAC authentication process

The process of using MAC for authentication is depicted in the following illustration −



Let us now try to understand the entire process in detail –
◇ The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value.

◇ What is the major difference between MAC and HASH?

◇ Similar to hash, MAC function also compresses an arbitrary long input into a fixed length output. The major difference between hash and MAC is that MAC uses a secret key during the compression.

◇ The sender forwards the message along with the MAC. Here, we assume that the message is sent in the clear, as we are concerned with providing message origin authentication, not confidentiality. If confidentiality is required then the message needs encryption.

◇ On receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.

◇ The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures himself that the message has been sent by the intended sender.

◇ If the computed MAC does not match the MAC sent by the sender, the receiver cannot determine whether it is the message that has been altered or it is the origin that has been falsified. As a bottom-line, a receiver safely assumes that the message is not genuine.

What are the Limitations of MAC?

1. Establishment of Shared Secret.

2. Inability to Provide Non-Repudiation

These are two major limitations of MAC, both due to its symmetric nature of operation
1. Establishment of Shared Secret.

◇ It can provide message authentication among pre-decided legitimate users who have shared keys.

◇ This requires establishment of shared secrets prior to use of MAC.

• Inability to Provide Non-Repudiation

◇ Non-repudiation is the assurance that a message originator cannot deny any previously sent messages and commitments or actions.

◇ MAC technique does not provide a non-repudiation service. If the sender and receiver get involved in a dispute over message origination, MACs cannot provide proof that a message was indeed sent by the sender.

◇ Though no third party can compute the MAC, still the sender could deny having sent the message and claim that the receiver forged it, as it is impossible to determine which of the two parties computed the MAC.

How can the limitations of MAC be overcome?

◇ Both these limitations can be overcome by using the public key based digital signatures discussed in the following section.

ADDITIONAL_READINGS

KECCSAC

Keccak and sha-3 are not the same.

In 2007, U.S. National Institute of Standard and Technology (NIST) initiated a competition about SHA-3. In 2012, Keccak team won the competition. From then on, developers implemented lots of “sha3” solution based on Keccak.

However, in 2014, NIST modified Keccak solution and released FIPS 202, and this updated proposal becomes official SHA-3 standard on Aug 2015. Many “old” program still use Keccak, and do not upgrade to official SHA-3 standard.

“old” code based on Keccak does not generate the same hash value as SHA-3 does. So, if using a “sha3” library, you should be crystal clear that the library is based on Keccak or based on standard SHA-3. A simple solution is doing a test for empty input:

SHA-3 standard output is:
a7ffc6f8bf1ed76651c14756a061d662f580ff4de43b49fa82d80a4b80f8434a
Many old Keccak-256 outputs are:
c5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470

The key idea behind SHA-3 is based on unkeyed permutations, as opposed to other typical hash function constructions that used keyed permutations. Keccak also does not make use of the Merkle- Damgard transformation that is commonly used to handle arbitrary-length input messages in hash functions. A newer approach, called sponge and squeeze construction, is used in Keccak. It is a random permutation model.

Different variants of SHA-3 have been standardized, such as SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, and SHAKE256