

## **NOTES**

# Unit -01

## Cryptography Introduction

**Cryptography** is the study and practice of techniques for secure communication in the presence of third parties called adversaries. It deals with developing and analyzing protocols which prevents malicious third parties from retrieving information being shared between two entities thereby following the various aspects of information security. Secure Communication refers to the scenario where the message or data shared between two parties can't be accessed by an adversary. In Cryptography, an Adversary is a malicious entity, which aims to retrieve precious information or data thereby undermining the principles of information security. Data Confidentiality, Data Integrity, Authentication and Non-repudiation are core principles of modern-day cryptography.

1. **Confidentiality** refers to certain rules and guidelines usually executed under confidentiality agreements which ensure that the information is restricted to certain people or places.
2. **Data integrity** refers to maintaining and making sure that the data stays accurate and consistent over its entire life cycle.
3. **Authentication** is the process of making sure that the piece of data being claimed by the user belongs to it.
4. **Non-repudiation** refers to ability to make sure that a person or a party associated with a contract or a communication cannot deny the authenticity of their signature over their document or the sending of a message.

Consider two parties Alice and Bob. Now, Alice wants to send a message  $m$  to Bob over a secure channel. So, what happens is as follows. The sender's message or sometimes called the Plaintext, is converted into an unreadable form using a Key  $k$ . The resultant text obtained is called the Ciphertext. This process is known as Encryption. At the time of receipt, the Ciphertext is converted back into the plaintext using the same Key  $k$ , so that it can be read by the receiver. This process is known as Decryption.

Alice (Sender)                      Bob (Receiver)

$C = E(m, k)$        $m = D(C, k)$

Here,  $C$  refers to the Ciphertext while  $E$  and  $D$  are the Encryption and Decryption algorithms respectively.

Let's consider the case of Caesar Cipher or Shift Cipher as an example. As the name suggests, in Caesar Cipher each character in a word is replaced by another character under some defined rules. Thus, if A is replaced by D, B by E and so on. Then, each character in the word would be shifted by a position of 3.

For example:

Plaintext : Geeksforgeeks

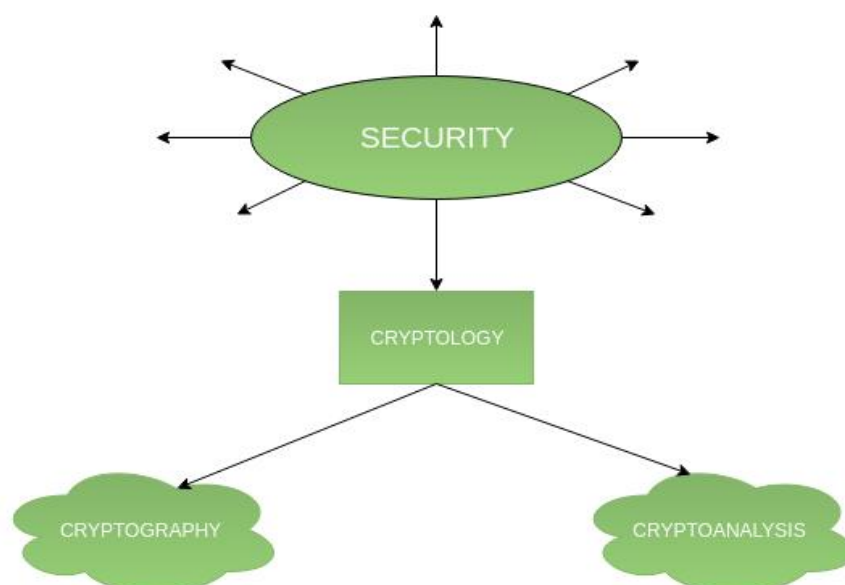
Ciphertext : Jhhnvirujhhnv

**Note** that even if the adversary knows that the cipher is based on Caesar Cipher, it cannot predict the plaintext as it doesn't have the key in this case which is to shift the characters back by three places.

**Cryptography** is an important aspect when we deal with network security. 'Crypto' means secret or hidden. Cryptography is the science of secret writing with the intention of keeping the data secret. Cryptanalysis, on the other hand, is the science or sometimes the art of breaking cryptosystems. Both terms are a subset of what is called **Cryptology**.

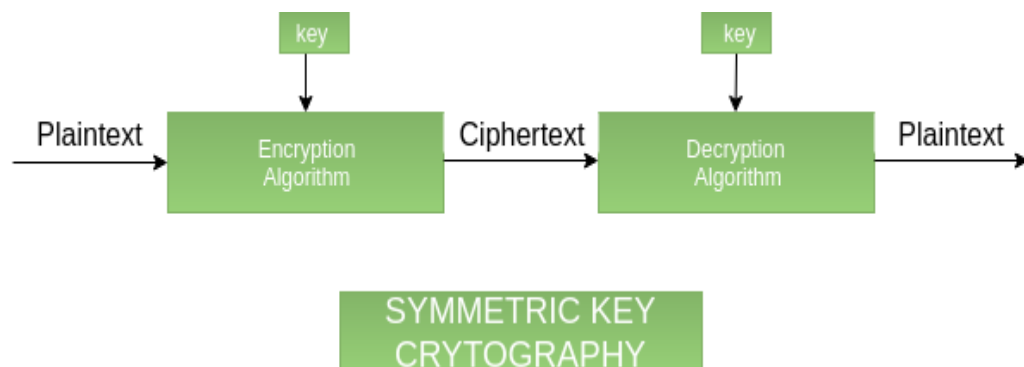
**Classification:** The flowchart depicts that cryptology is only one of the factors involved in securing networks. Cryptology refers to the study of codes, which involves both writing (cryptography) and solving (cryptanalysis) them. Below is a classification of the crypto terminologies and their various types.

Cryptography:

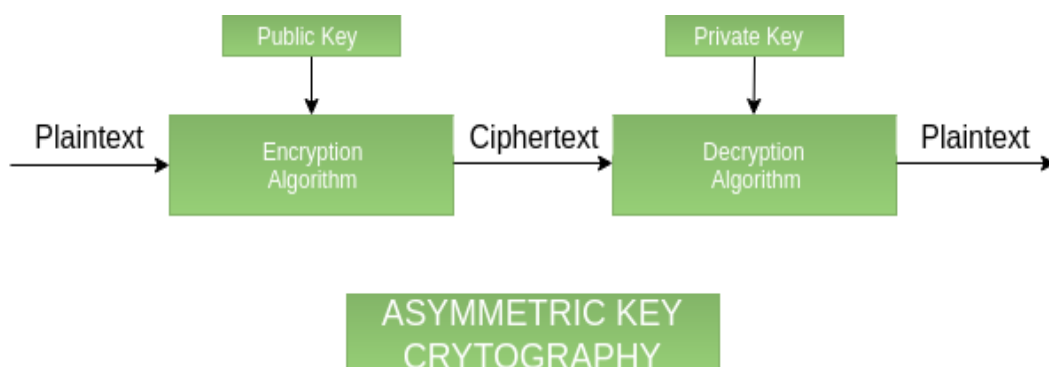


Cryptography is classified into symmetric cryptography and asymmetric cryptography. Below are the description of these types.

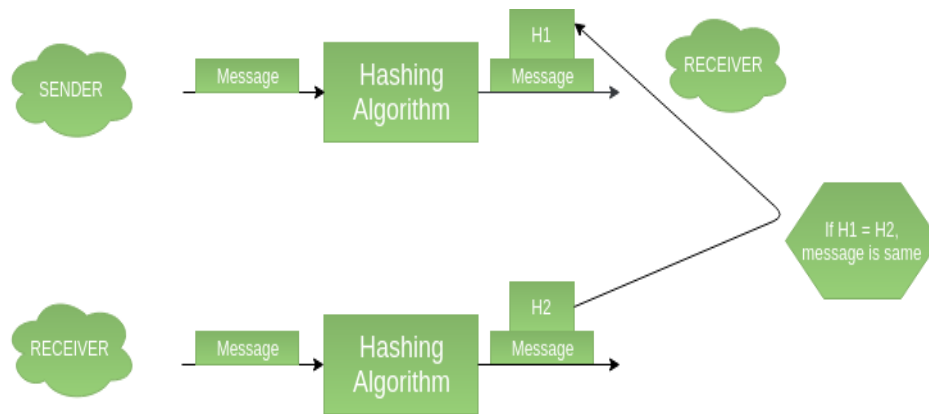
1. **Symmetric key cryptography** – It involves the usage of one secret key along with encryption and decryption algorithms which help in securing the contents of the message. The strength of symmetric key cryptography depends upon the number of key bits. It is relatively faster than asymmetric key cryptography. There arises a key distribution problem as the key has to be transferred from the sender to the receiver through a secure channel.



2. **Asymmetric key cryptography:** It is also known as public-key cryptography because it involves the usage of a public key along with the secret key. It solves the problem of key distribution as both parties use different keys for encryption/decryption. It is not feasible to use for decrypting bulk messages as it is very slow compared to symmetric key cryptography.



3. **Hashing:** It involves taking the plain text and converting it to a hash value of fixed size by a hash function. This process ensures the integrity of the message as the hash value on both, the sender's and receiver's sides should match if the message is unaltered.



Difference between Hash functions, Symmetric, and Asymmetric algorithms:

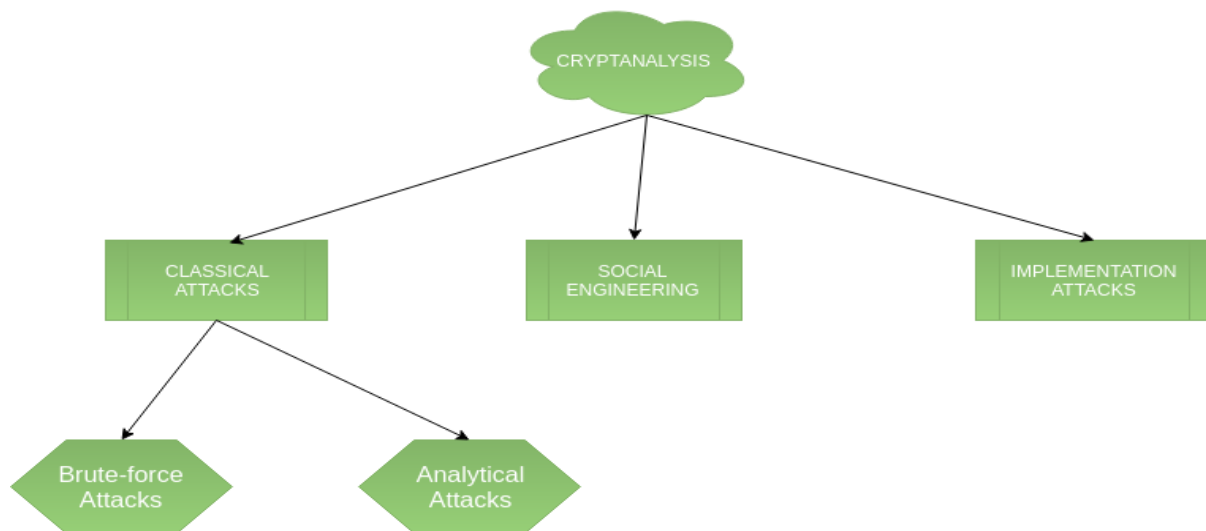
Feature	Hash functions	Symmetric algorithms	Asymmetric algorithms
Number of Keys	0	1	2
Length of keys recommended by NIST	256 bits	128 bits	2048 bits
Example	SHA-256, SHA3-256, SHA-512	AES or 3DES	RSA, DSA, ECC

Cryptanalysis:

*The art and science of breaking the cipher text is known as cryptanalysis.*

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

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



1. **Classical attacks:** It can be divided into:
  - a) **Mathematical analysis:** It's a type of attack that takes advantage of structural flaws in a specific algorithm.
  - b) **Brute-force attacks:** The attacker uses a Brute Force Attack (BFA) to try all potential keys in order to figure out the key. If the key is long, the attack will take a long time to execute. Brute-force attacks run the encryption algorithm for all possible cases of the keys until a match is found. The encryption algorithm is treated as a black box. Analytical attacks are those attacks that focus on breaking the cryptosystem by analyzing the internal structure of the encryption algorithm.
2. **Social Engineering attack:** It is something that is dependent on the human factor. Tricking someone to reveal their passwords to the attacker or allowing access to the restricted area comes under this attack. People should be cautious when revealing their passwords to any third party which is not trusted.
3. **Implementation attacks:** Implementation attacks such as side-channel analysis can be used to obtain a secret key. They are relevant in cases where the attacker can obtain physical access to the cryptosystem.

**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.

## Characteristics of Modern Cryptography

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

Classic Cryptography	Modern Cryptography
It manipulates traditional characters, i.e., letters and digits directly.	It operates on binary bit sequences.
It is mainly based on 'security through obscurity'. The techniques employed for coding were kept secret and only the parties involved in communication knew about them.	It relies on publicly known mathematical algorithms for coding the information. Secrecy is obtained through a secret key which is used as the seed for the algorithms. The computational difficulty of algorithms, absence of secret key, etc., make it impossible for an attacker to obtain the original information even if he knows the algorithm used for coding.
It requires the entire cryptosystem for communicating confidentially.	Modern cryptography requires parties interested in secure communication to possess the secret key only.

## Context of Cryptography

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

- Cryptography
- Cryptanalysis

## Security Services of Cryptography

The primary objective of using cryptography is to provide the following four fundamental information security services. Let us now see the possible goals intended to be fulfilled by cryptography.

## Confidentiality

Confidentiality is the fundamental security service provided by cryptography. It is a security service that keeps the information from an unauthorized person. It is sometimes referred to as **privacy** or **secrecy**.

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

## Data Integrity

It is security service that deals with identifying any alteration to the data. The data may get modified by an unauthorized entity intentionally or accidentally. Integrity service confirms that 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.

## Authentication

Authentication 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.

Authentication service has two variants –

- **Message authentication** identifies the originator of the message without any regard 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.

## Non-repudiation

It 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.





# Cryptography Primitives

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

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

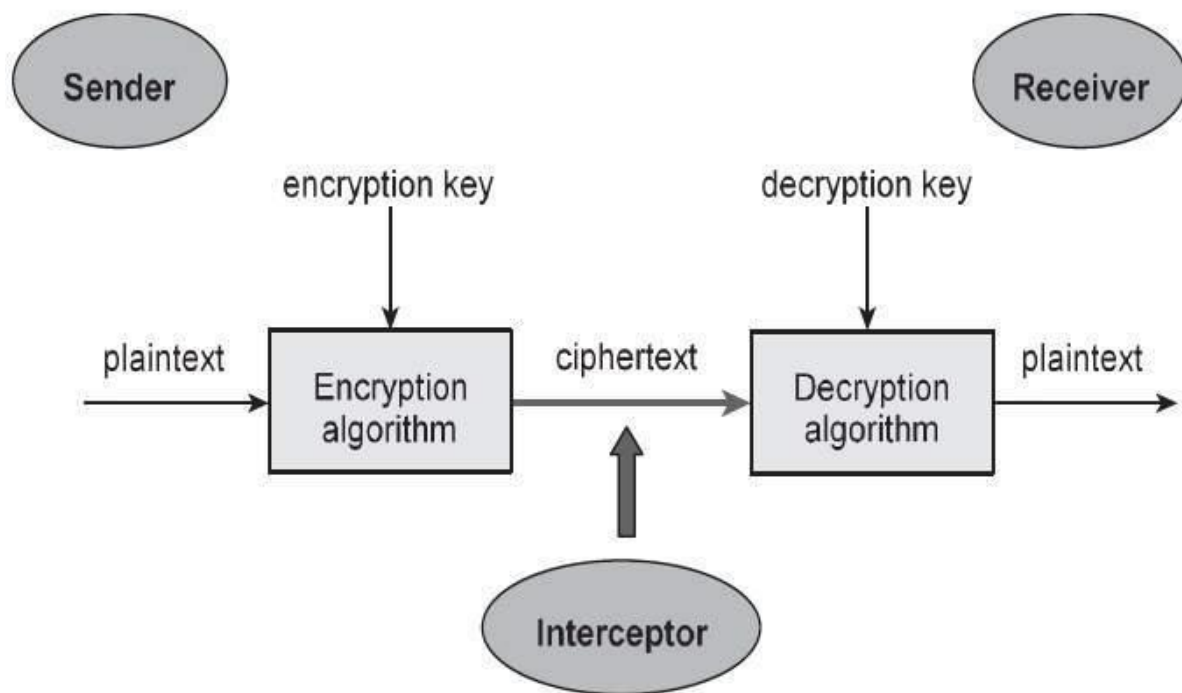
The following table shows the primitives that can achieve a particular security service on their own.

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

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

A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. A cryptosystem is also referred to as a **cipher system**.

Let us discuss a simple model of a cryptosystem that provides confidentiality to the information being transmitted. This basic model is depicted in the illustration below –



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.

## Components of a Cryptosystem or cipher system

The various components of a basic cryptosystem are as follows –

- **Plaintext.** It is the data to be protected during transmission.
- **Encryption Algorithm.** It is a mathematical process that produces a ciphertext for any given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.
- **Ciphertext.** It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.
- **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.

- **Encryption Key.** It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.
- **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.

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

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.

## Types of Cryptosystems

Fundamentally, there are two types of cryptosystems based on the manner in which encryption-decryption is carried out in the system –

- Symmetric Key Encryption
- Asymmetric Key Encryption

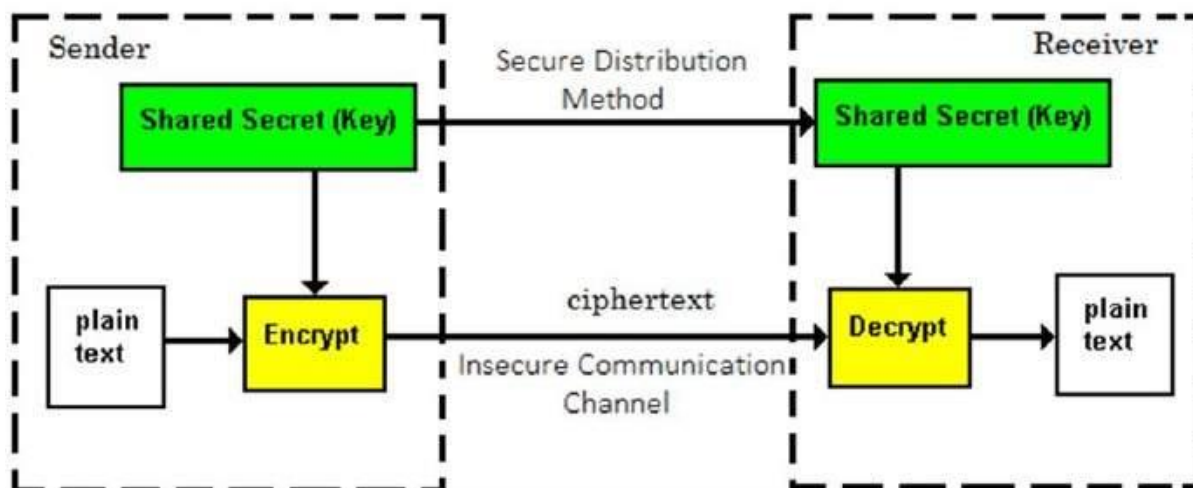
The main difference between these cryptosystems is the relationship between the encryption and the decryption key. Logically, in any cryptosystem, both the keys are closely associated. It is practically impossible to decrypt the ciphertext with the key that is unrelated to the encryption key.

### Symmetric Key Encryption

The encryption process where **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.



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.

The salient features of cryptosystem based on symmetric key encryption are –

- 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 group is  $n \times (n - 1)/2$ .
- 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 computer system required to run symmetric algorithm is less.

## Challenge of Symmetric Key Cryptosystem

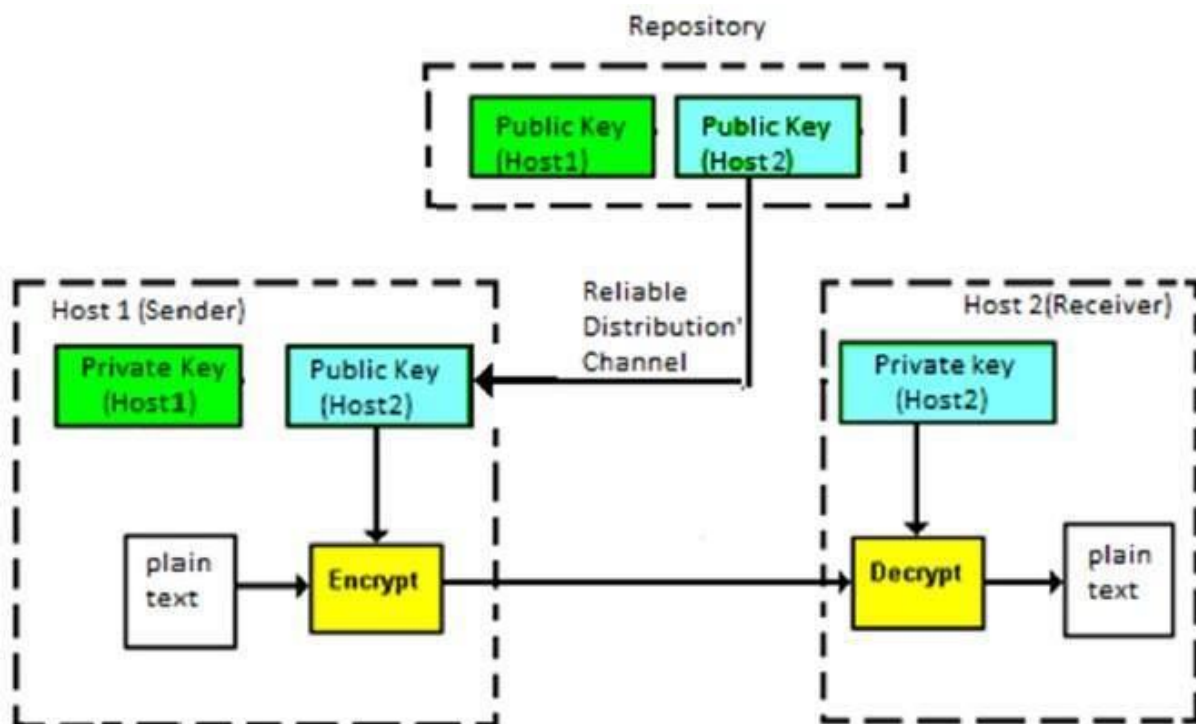
There are two restrictive challenges of employing symmetric key cryptography.

- **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, a communication between online seller and customer. These limitations of symmetric key encryption gave rise to asymmetric key encryption schemes.

## 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 –



Asymmetric Key Encryption was invented in the 20<sup>th</sup> century to come over the necessity of pre-shared secret key between communicating persons. 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 to put the public key in 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 computer system required to run 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.

## Challenge of Public Key Cryptosystem

Public-key cryptosystems have one significant challenge – 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 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.

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. 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

A summary of basic key properties of two types of cryptosystems is given below –

	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 disadvantage 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

In the 19<sup>th</sup> 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 –

- The cryptosystem should be unbreakable practically, if not mathematically.
- 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.
- The key should be easily communicable, memorable, and changeable.
- The ciphertext should be transmissible by telegraph, an unsecure channel.
- The encryption apparatus and documents should be portable and operable by a single person.
- Finally, 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 **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 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.

## Attacks On Cryptosystems

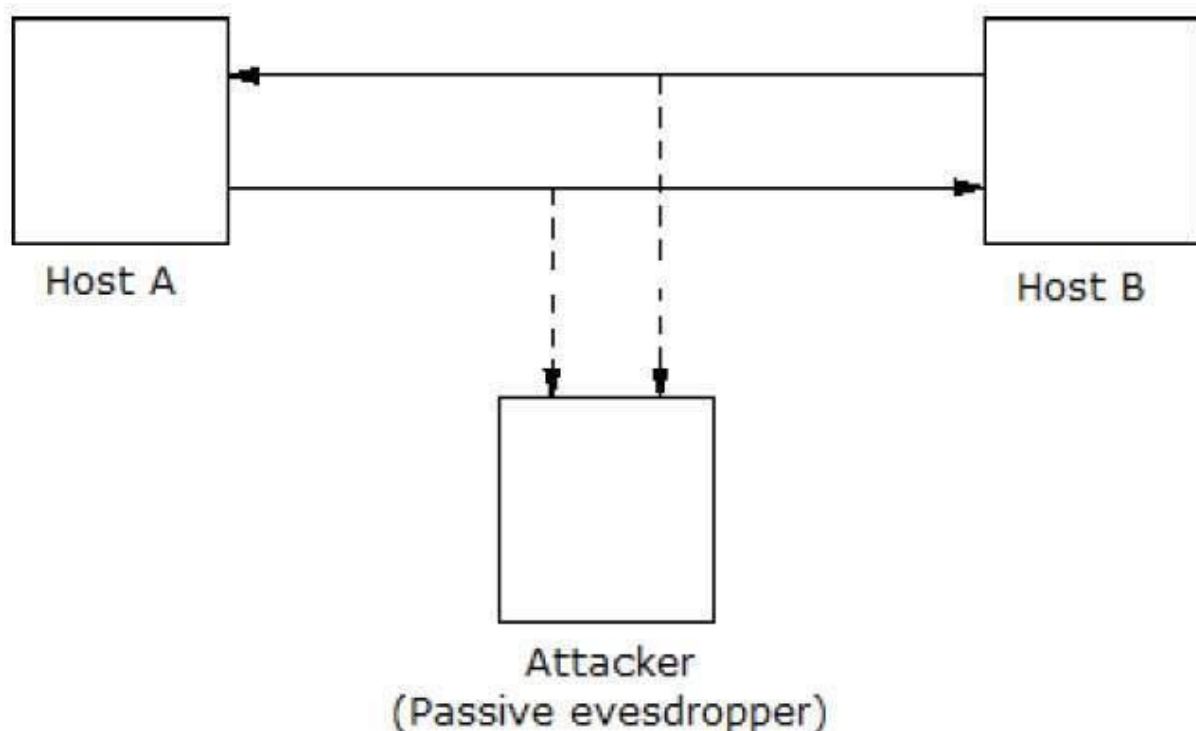
In the present era, not only business but almost all the aspects of human life are driven by information. Hence, it has become imperative to protect useful information from malicious activities such as attacks. Let us consider the types of attacks to which information is typically subjected to.

Attacks are typically categorized based on the action performed by the attacker. An attack, thus, can be **passive** or **active**.

## Passive Attacks

The main goal of a passive attack is to obtain **unauthorized access to the 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. A passive attack is often seen as *stealing* information. 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.



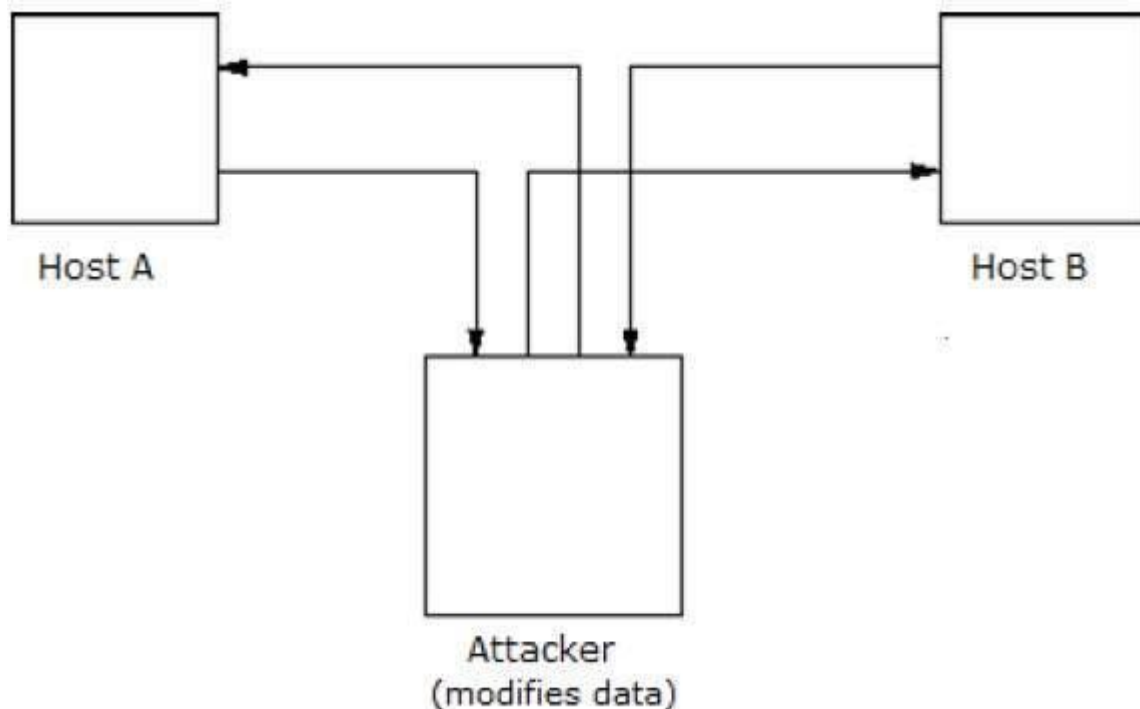
## Active Attacks

An active attack involves changing the information in some way by conducting some process on the information. For example,

- Modifying the information in an unauthorized manner.
- Initiating unintended or unauthorized transmission of information.



- Alteration of authentication data such as originator name or timestamp associated with information
- Unauthorized deletion of data.
- 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.

## Cryptographic Attacks

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 –

- **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 simplest method of this attack, attacker builds a dictionary of ciphertexts and corresponding plaintexts that he has learnt over a period of time. In 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 3<sup>rd</sup> Aug. Then to find the next student whose birthdate is 3<sup>rd</sup> Aug, we need to enquire  $1.25 \cdot \sqrt{365} \approx 25$  students.  
Similarly, if the hash function produces 64 bit hash values, the possible hash values are  $1.8 \times 10^{19}$ . By repeatedly evaluating the function for different inputs, the same output is expected to be obtained after about  $5.1 \times 10^9$  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 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 processor. By measuring such timings, it is 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.

## Substitution Technique in Cryptography

**Substitution technique** is a classical encryption technique where the characters present in the **original message** are **replaced** by the **other characters or numbers or by symbols**. If the plain text (original message) is considered as the string of bits, then the substitution technique would replace bit pattern of plain text with the bit pattern of cipher text.

## Some Substitution Technique:

1. Caesar Cipher
2. Monoalphabetic Cipher
3. Playfair Cipher
4. Hill Cipher
5. Polyalphabetic Cipher
6. One-Time Pad

## Caesar Cipher

This is the simplest substitution cipher by Julius Caesar. In this substitution technique, to encrypt the plain text, each alphabet of the plain text is replaced by the alphabet three places further it. And to decrypt the cipher text each alphabet of cipher text is replaced by the alphabet three places before it.

Let us take a simple example:

**Plain Text:** meet me tomorrow

**Cipher Text:** phhw ph wrpruurz

Look at the example above, we have replaced, 'm' with 'p' which occurs three places after, 'm'. Similarly, 'e' is replaced with 'h' which occurs in three places after 'e'.

**Note:** If we have to replace the letter 'z' then the next three alphabets counted after 'z' will be 'a' 'b' 'c'. So, while counting further three alphabets if 'z' occurs it circularly follows 'a'.

There are also some drawbacks of this simple substitution technique. If the hacker knows that the Caesar cipher is used then to perform brute force cryptanalysis, he has only to try 25 possible keys to decrypt the plain text.

The hacker is also aware of the encryption and decryption algorithm.

## Monoalphabetic Cipher

Monoalphabetic cipher is a substitution cipher, where the cipher alphabet for each plain text alphabet is fixed, for the entire encryption.

In simple words, if the alphabet 'p' in the plain text is replaced by the cipher alphabet 'd'. Then in the entire plain text wherever alphabet 'p' is used, it will be replaced by the alphabet 'd' to form the ciphertext.

## Monoalphabetic and Polyalphabetic 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 occurrence 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.

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 is Substitution Cipher Technique?

In the Substitution Cipher technique, each character is substituted with other character/number or other symbol. This technique changes the identity of a character but not its position in the string.

A substitution cipher encrypts a text sequence by replacing letters or units of text with other letters or units of text. Substitution ciphers are an early form of cryptography that predates the development of computers and are now largely outdated.

Like B or C, a letter is converted into another letter in a substitution cipher, encrypting the sequence for a human reader. The problem is that basic substitution ciphers do not encrypt adequately in terms of computer assessment.

Substitution ciphers became relatively easy for computers to crack with the emergence of the personal computer. However, some of the substitution cipher's concepts have persisted, for example, some versions of current encryption may encrypt data using an extensive text set and a very clever substitution.

### Merits:

#### 1. Economical:

It is economical, because we have not to collect all data. Instead of getting data from 5000 farmers, we get it from 50-100 only.

#### 2. Less Time Consuming:

As no of units is only a fraction of the total universe, time consumed is also a fraction of total time. Number of units is considerably small, hence the time.

### 3. Reliable:

If sample is taken judiciously, the results are very reliable and accurate.

### 4. Organisational Convenience:

As samples are taken and the number of units is smaller, the better (Trained) enumerators can be employed by the organisation.

### 5. More Scientific:

According to Prof R.A. Fisher, "The sample technique has four important advantages over census technique of data collection. They are Speed, Economy, Adaptability and Scientific approach."

It is based on certain laws such as:

- (a) Law of Statistical Regularity
- (b) Law of Inertia of Large numbers
- (c) Law of Persistence
- (d) Law of Validity.

### 6. Detailed Enquiry:

A detailed study can be undertaken in case of the units included in the sample. Size of sample can be taken according to time and money available with the investigator.

### 7. Indispensable Method:

If universe is bigger, there remains no option but to proceed for this method. It is specially used for infinite, hypothetical and perishable universes.

## **Demerits:**

#### 1. Absence of Being Representative:

Methods, such as purposive sampling may not provide a sample, that is representative.

#### 2. Wrong Conclusion:

If the sample is not representative, the results will not be correct. These will lead to the wrong conclusions.

#### 3. Small Universe:

Sometimes universe is so small that proper samples cannot be taken not of it. Number of units are so less.

#### 4. Specialised Knowledge:

It is a scientific method. Therefore, to get a good and representative sample, one should have special knowledge to get good sample and to perform proper analysis so that reliable result may be achieved.

#### 5. Inherent defects:

The results which are achieved through the analysis of sampling data may not be accurate as this method has inherent defects. There is not even a single method of sampling which has no demerit.

#### 6. Sampling Error:

This method of sampling has many errors.

#### 7. Personal Bias:

As in many cases the investigator, chooses samples, such as convenience method, chances of personal bias creep in.

## **What is Transposition Cipher Technique?**

In the Transposition Cipher Technique, each character's position is shifted to a different position. A transposition cipher is an encryption method in which plaintext units (usually letters or groups of characters) are shifted in a predictable way resulting in the ciphertext being a permutation of the plaintext. That is, the units' order is altered (the plaintext is reordered).

To encrypt, a bijective function is applied to the locations of the characters, and to decrypt, an inverse function is used.

The Rail Fence encryption is a type of transposition cipher named after how it is encoded. The plaintext is written on successive "rails" of an imaginary fence in the rail fence, downwards and diagonally, then moved up when we reach the bottom. The message is then read aloud in a series of rows.

The Rail Fence Cipher follows a scytale-like pattern, an old Greek mechanical device for constructing a transposition cipher. A cylinder and a ribbon wrapped around the cylinder comprised the system. The coiled ribbon was used to write the encrypted message. When the ribbon was uncoiled from the cylinder, the letters of the original message were rearranged. The message was decoded when the ribbon recoiled on a cylinder with the same diameter as the encrypting cylinder.

## Difference between Substitution Cipher and Transposition Cipher

The following table highlights the important differences between Substitution Cipher and Transposition Cipher.

Key	Substitution Cipher Technique	Transposition Cipher Technique
Algorithm	Each character is replaced with a different character, integer, or symbol.	Each character has been repositioned from its original place.
Forms	It comes in two forms: Mono-Alphabetic Substitution Cipher and Poly-Alphabetic Substitution Cipher.	It has two forms: Key-less transposition cipher and keyed transposition cipher.
Change	Character identity is changed but position remains same.	Character position is changed but identity remains same.
Detection	A letter less frequently used can be easily traced.	A letter near to original position can be easily traced.
Example	Caesar Cipher is an example of Substitution Cipher.	Reil Fence Cipher is an example of Transposition Cipher.

## Conclusion

Substitution cipher and Transposition cipher are traditional methods in cryptography, which are now outdated techniques. As their names imply, substitution ciphers substitute each character of a string with another character, number, or symbol; whereas transposition ciphers transpose each character of a string to a different position.

## Modern Symmetric Key Encryption

### Block Ciphers

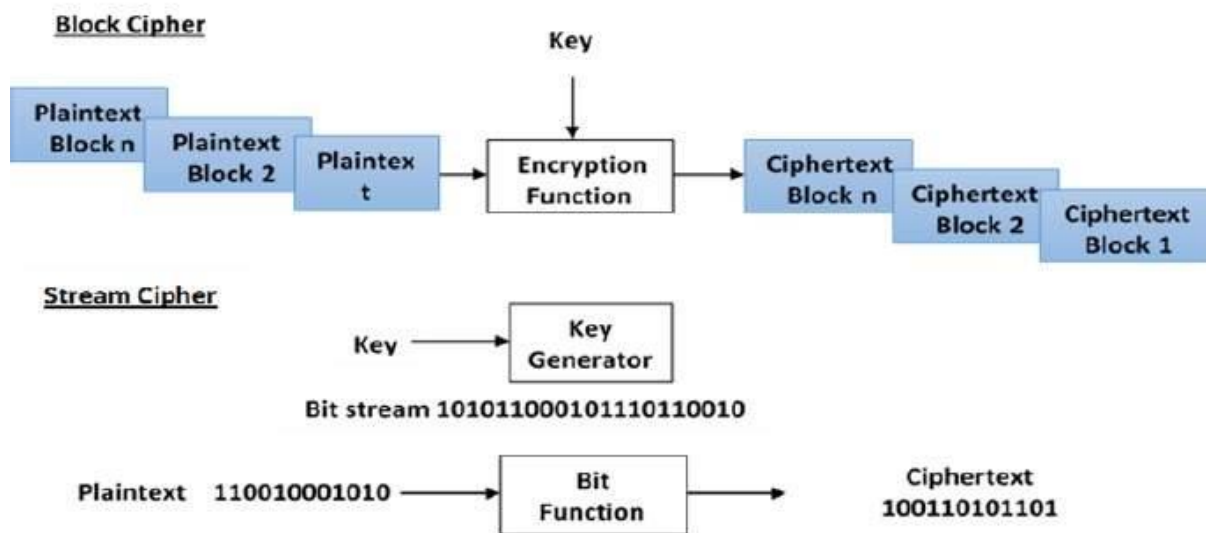
In this scheme, the plain binary text is processed in blocks (groups) of bits at a time; i.e. a block of plaintext bits is selected, a series of operations is performed on this



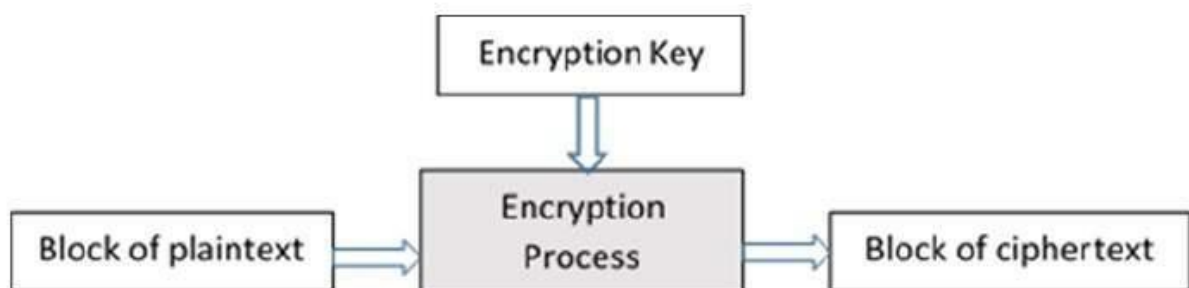
block to generate a block of ciphertext bits. The number of bits in a block is fixed. For example, the schemes DES and AES have block sizes of 64 and 128, respectively.

## 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.



The basic scheme of a **block cipher** is depicted as follows –



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

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  $2^m$ . 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 size, 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 processor handle data in multiple of 8 bits.

## 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**.

Too much padding makes the system inefficient. Also, padding may render the system insecure at times, if the padding is done with same bits always.

## Block Cipher Schemes

There is 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.

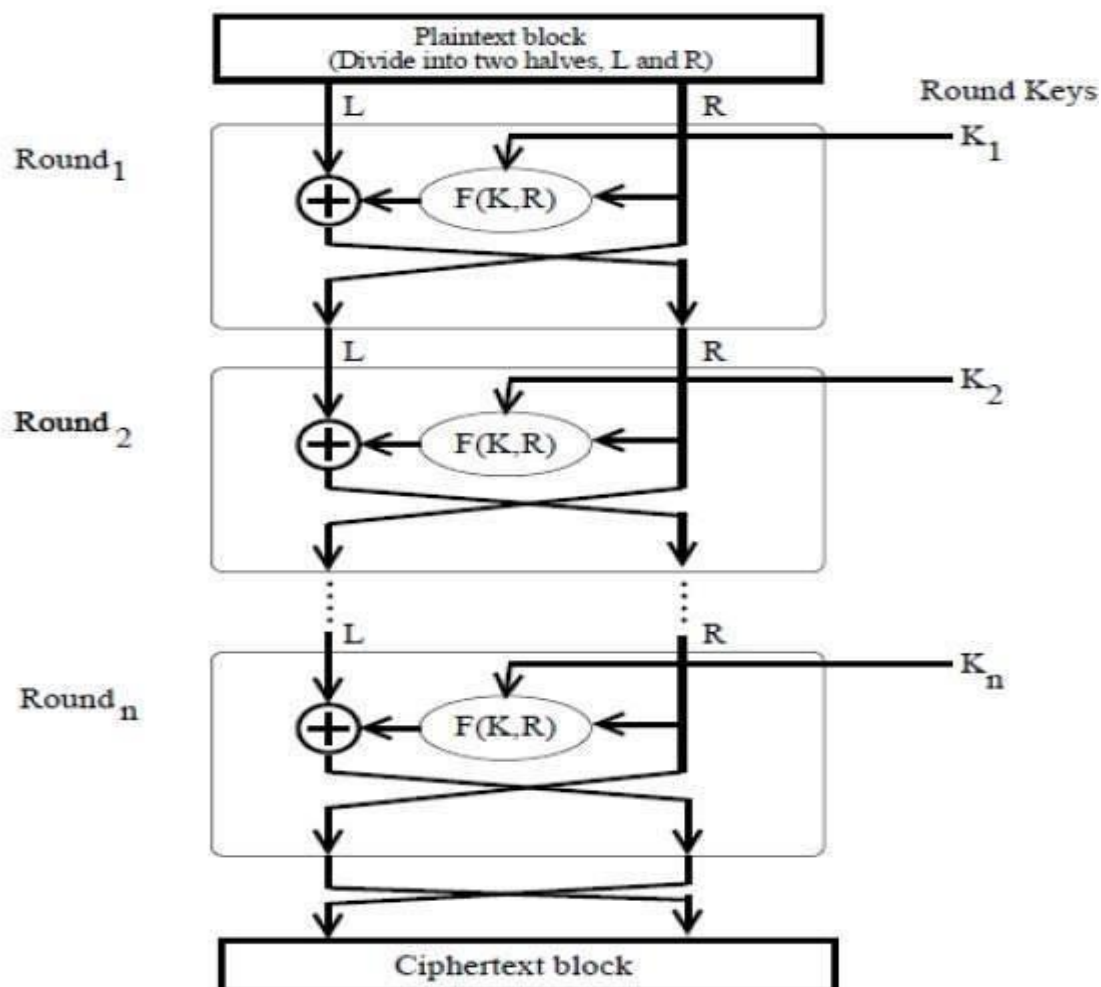
# Feistel Block Cipher

Feistel Cipher is not a specific scheme of block cipher. It is a design model from which many different block ciphers are derived. DES is just one example of a Feistel Cipher. A cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption.

## Encryption Process

The encryption process uses the Feistel structure consisting 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 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 selection of round function 'f'. In order to be unbreakable scheme, this function needs to have several important properties that are beyond the scope of our discussion.

## Decryption Process

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 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 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.

## FERMAT'S AND EULER'S THEOREMS

Two theorems that play important roles in public-key cryptography are Fermat's theorem and Euler's theorem.

### Fermat's Theorem

Fermat's theorem states the following: If  $p$  is prime and  $a$  is a positive integer not divisible by  $p$ , then

$$a^{p-1} \equiv 1 \pmod{p} \quad (8.2)$$

## Euler's Totient Function

Before presenting Euler's theorem, we need to introduce an important quantity in number theory, referred to as **Euler's totient function**, written  $\phi(n)$ , and defined as the number of positive integers less than  $n$  and relatively prime to  $n$ . By convention,  $\phi(1) = 1$ .

**Euler's theorem** is a generalization of Fermat's little theorem dealing with powers of integers modulo positive integers. It arises in applications of elementary number theory, including the theoretical underpinning for the RSA cryptosystem.

Let  $n$  be a positive integer, and let  $a$  be an integer that is relatively prime to  $n$ . Then

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

where  $\phi(n)$  is Euler's totient function, which counts the number of positive integers  $\leq n$  which are relatively prime to  $n$ .

Suppose  $a$  is relatively prime to 10. Since  $\phi(10)=4$ , Euler's theorem says that  $a^4 \equiv 1 \pmod{10}$ , i.e. the units digit of  $a^4$  is always 1. See the wiki on finding the last digit of a power for similar problems.

## Unit-02

### Cryptographic Algorithms

In cryptography, the strength of a transaction is based on the key. In general, the longer the key, the stronger it is. Many different systems use the same cryptographic algorithm, but they all use different keys. It's important that you keep the key safe and confidential. If the key gets lost, you will no longer be able to decrypt data that was encrypted with it. If someone else comes in possession of the key, then he or she will be able to decrypt your encrypted data.

All cryptographic algorithms are based on some sort of mathematical function. Many times, mathematical functions can have numbers or series of numbers that cause the function to behave improperly. In cryptography, this number or series of numbers is called a weak key. When cryptographic algorithms are designed, the creators try to reduce or eliminate the possible number of weak keys. But, they are not always successful.

### Symmetric Encryption

Symmetric key algorithms are sometimes referred to as secret key algorithms. This is because these types of algorithms generally use one key that is kept secret by the systems engaged in the encryption and decryption processes. This single key is used for both encryption and decryption.

**Symmetric key algorithms** tend to be very secure. In general, they are considered more secure than asymmetric key algorithms. There are some symmetric key algorithms that are considered virtually unbreakable. Symmetric key algorithms are also very fast. This is why they are often used in situations where there is a lot of data that needs to be encrypted.

**In symmetric key algorithms**, the key is shared between the two systems. This can present a problem. You have to figure out a way to get the key to all systems that will have to encrypt or decrypt data using a symmetric key algorithm. Having to manually

distribute a key to all systems can be a quite cumbersome task. Sometimes, this can only be done by copying the key from a central location. You can imagine how troublesome that can be. On Windows systems, you do have the option of possibly using a group policy or a script of some kind to copy the key to the necessary systems. This helps, but the administrator is still responsible for making sure the group policy or the script functions properly.

### Symmetric Key Algorithms

There are hundreds of different symmetric key algorithms available. Each has its own strengths and weaknesses. Some of the more common examples are DES, 3DES, AES, IDEA, RC4, and RC5.

### Encryption modes

When encrypting multiple blocks of data using a block cipher, there are various encryption modes that may be employed, each having particular advantages and disadvantages. We will look at some of these here.

#### ECB (Electronic Code Book) mode

This is the simplest mode, whereby each block of data is simply encrypted with the same key. There is thus a one-to-one mapping between the plaintext block and corresponding ciphertext block for any particular key, analogous to looking up the plaintext in a (very large!) code book and reading off the matching ciphertext. Multiple blocks can therefore be encrypted in parallel. However, this mode leaks information about the plaintext and is thus rarely used. Also, the plaintext data must be padded to an integral number of blocks.

#### CBC (Cipher Block Chaining) mode

CBC tries to improve on ECB by making the encryption of each block dependent not just on the key but also on ciphertext of the previous block. Each block of ciphertext thus depends on all the plaintext blocks processed up to that point, which prevents parallelization of the encryption process. Another downside is that any error can propagate to the subsequent block. Furthermore, CBC is also vulnerable to what is known as a “bit flipping” attack. As with ECB, padding of the last block is necessary.

#### OFB (Output Feedback) mode

OFB turns a block cipher into a synchronous stream cipher. Based on an IV and the key, it generates keystream blocks which are then simply XORed with the plaintext data. As with CFB, the encryption and decryption processes are identical, and no padding is required.

#### CTR (Counter) mode

CTR shares many characteristics with OFB, but it generates the next keystream block by encrypting successive values of a counter (which must be synchronized at both ends). CTR mode does not propagate transmission errors and lends itself to parallelization.

## 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.

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 attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows –

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

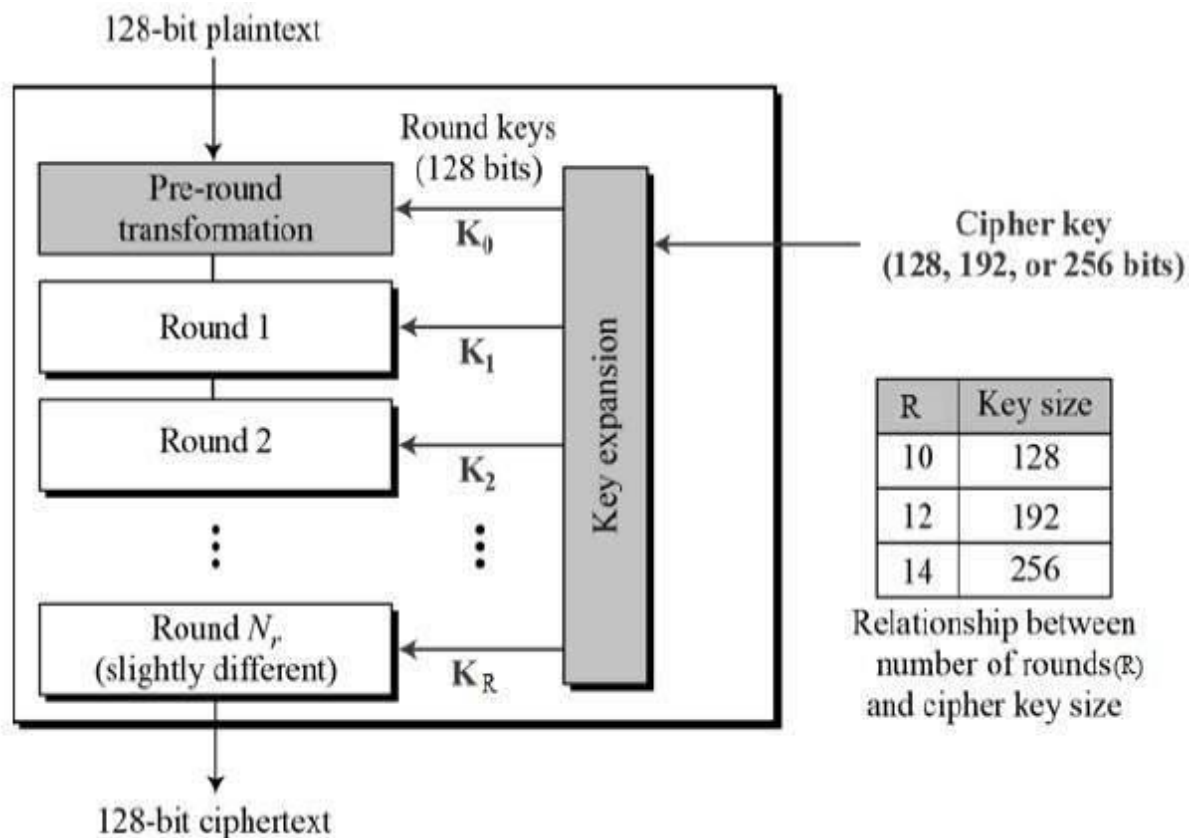
## **Operation of AES**

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

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 –

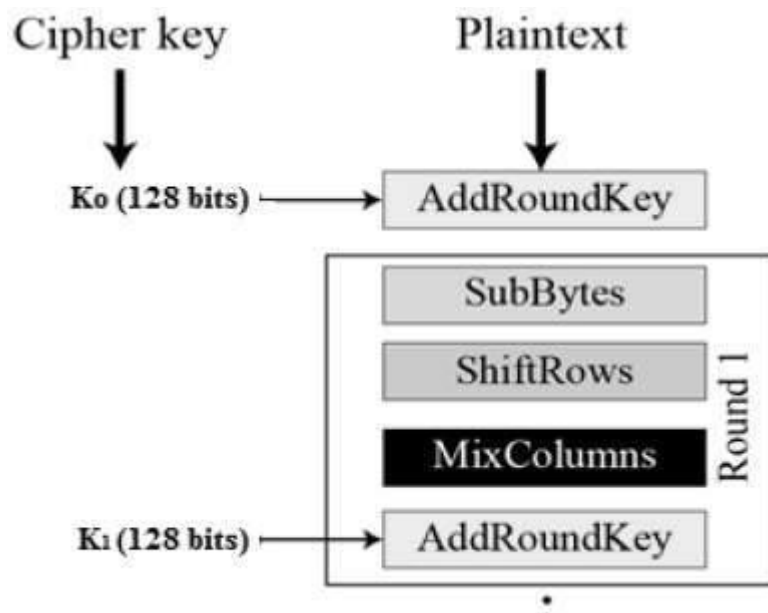
Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration –



## Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of 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 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.

## 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.

**Following are the benefits or advantages of AES:**

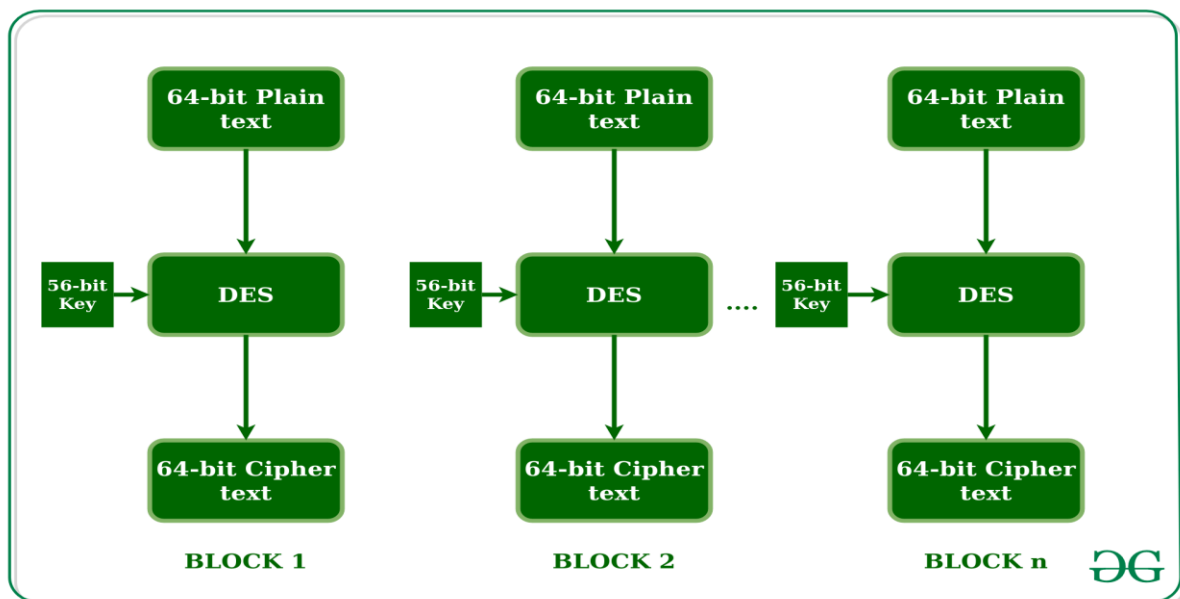
- ➡ As it is implemented in both hardware and software, it is most robust security protocol.
- ➡ It uses higher length key sizes such as 128, 192 and 256 bits for encryption. Hence it makes AES algorithm more robust against hacking.
- ➡ It is most common security protocol used for wide various of applications such as wireless communication, financial transactions, e-business, encrypted data storage etc.
- ➡ It is one of the most spread commercial and open source solutions used all over the world.
- ➡ No one can hack your personal information.
- ➡ For 128 bit, about  $2^{128}$  attempts are needed to break. This makes it very difficult to hack it as a result it is very safe protocol.

**Following are the disadvantages of AES:**

- ➡ It uses too simple algebraic structure.
- ➡ Every block is always encrypted in the same way.
- ➡ Hard to implement with software.
- ➡ AES in counter mode is complex to implement in software taking both performance and security into considerations.

## **Data encryption standard (DES) |**

**Data encryption standard (DES)** has been found vulnerable to very powerful attacks and therefore, the popularity of DES has been found slightly on the decline. DES is a block cipher and encrypts data in blocks of size of **64 bits** each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is **56 bits**. The basic idea is shown in the figure:



We have mentioned that DES uses a 56-bit key. Actually, the initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

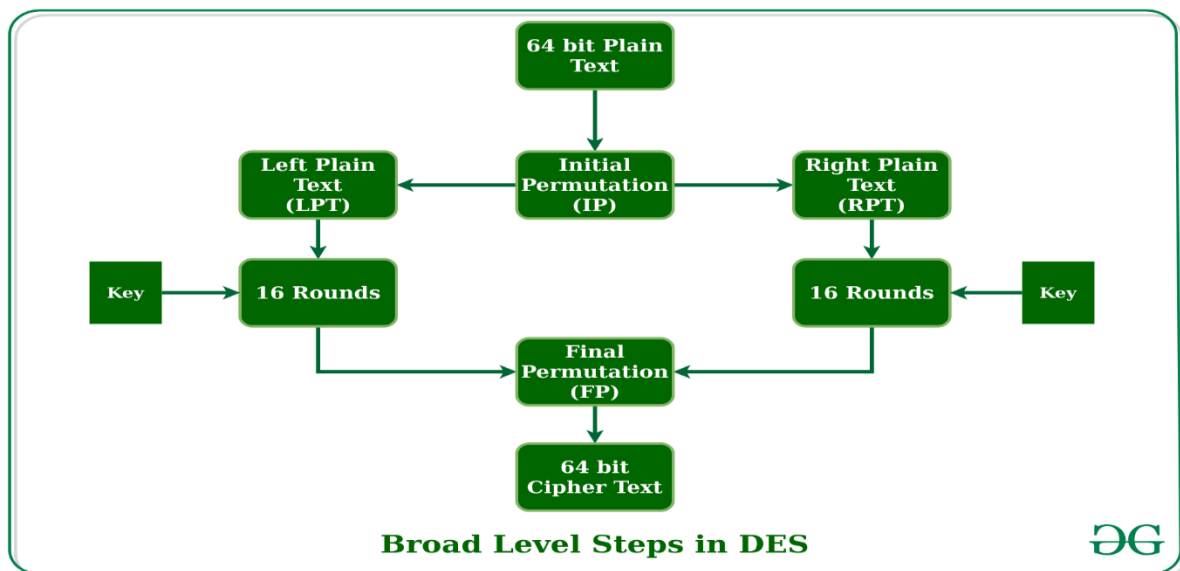
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Figure - discarding of every 8<sup>th</sup> bit of original key

Thus, the discarding of every 8th bit of the key produces a **56-bit key** from the original **64-bit key**.

DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

- In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP) function.
- The initial permutation is performed on plain text.
- Next, the initial permutation (IP) produces two halves of the permuted block; saying Left Plain Text (LPT) and Right Plain Text (RPT).
- Now each LPT and RPT go through 16 rounds of the encryption process.
- In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
- The result of this process produces 64-bit ciphertext.



### Initial Permutation (IP):

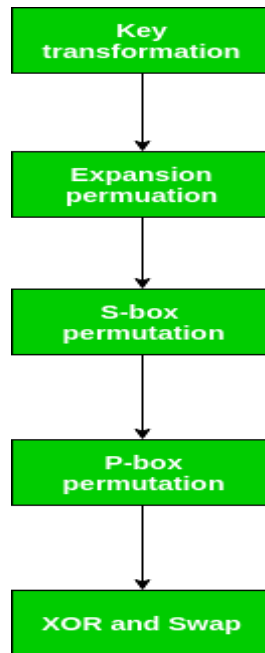
As we have noted, the initial permutation (IP) happens only once and it happens before the first round. It suggests how the transposition in IP should proceed, as shown in the figure. For example, it says that the IP replaces the first bit of the original plain text block with the 58th bit of the original plain text, the second bit with the 50th bit of the original plain text block, and so on.

This is nothing but jugglery of bit positions of the original plain text block. the same rule applies to all the other bit positions shown in the figure.

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

**Figure - Initial permutation table**

As we have noted after IP is done, the resulting 64-bit permuted text block is divided into two half blocks. Each half-block consists of 32 bits, and each of the 16 rounds, in turn, consists of the broad-level steps outlined in the figure.



#### Step-1: Key transformation:

We have noted initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key. Thus, for each a 56-bit key is available. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called key transformation. For this, the 56-bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round.

**For example:** if the round numbers 1, 2, 9, or 16 the shift is done by only one position for other rounds, the circular shift is done by two positions. The number of key bits shifted per round is shown in the figure.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
#key bits shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

**Figure - number of key bits shifted per round**

After an appropriate shift, 48 of the 56 bits are selected. for selecting 48 of the 56 bits the table is shown in the figure given below. For instance, after the shift, bit number 14 moves to the first position, bit number 17 moves to the second position, and so on. If we observe the table carefully, we will realize that it contains only 48-bit positions. Bit number 18 is discarded (we will not find it in the table), like 7 others, to reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as a selection of a 48-bit subset of the original 56-bit key it is called Compression Permutation.

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Figure - compression permutation

Because of this compression permutation technique, a different subset of key bits is used in each round. That makes DES not easy to crack.

#### Step-2: Expansion Permutation:

Recall that after the initial permutation, we had two 32-bit plain text areas called Left Plain Text(LPT) and Right Plain Text(RPT). During the expansion permutation, the RPT is expanded from 32 bits to 48 bits. Bits are permuted as well hence called expansion permutation. This happens as the 32-bit RPT is divided into 8 blocks, with each block consisting of 4 bits. Then, each 4-bit block of the previous step is then expanded to a corresponding 6-bit block, i.e., per 4-bit block, 2 more bits are added.

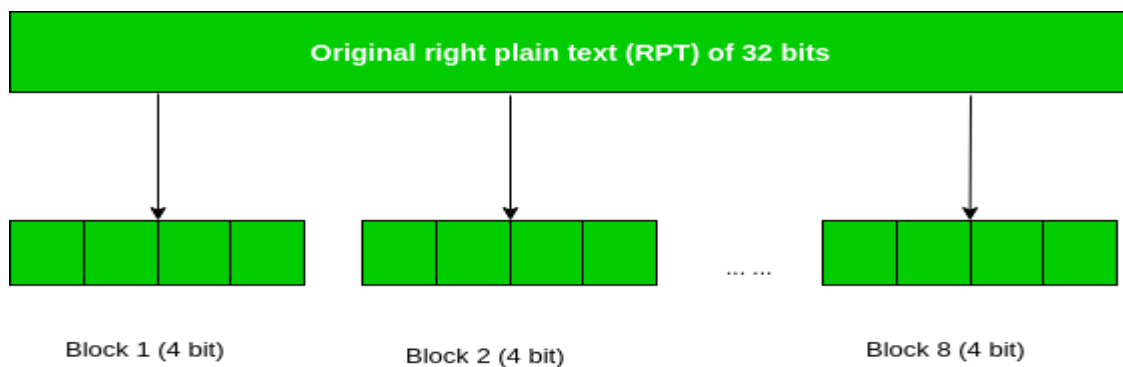


Figure - division of 32 bit RPT into 8 bit blocks

This process results in expansion as well as a permutation of the input bit while creating output. The key transformation process compresses the 56-bit key to 48 bits. Then the expansion permutation process expands the **32-bit RPT to 48-bits**. Now the 48-bit key is XOR with 48-bit RPT and the resulting output is given to the next step, which is the **S-Box substitution**.

#### Strength:

The strength of DES lies on two facts:

- The use of 56-bit keys: 56-bit key is used in encryption, there are 256 possible keys. A brute force attack on such number of keys is impractical.
- The nature of algorithm: Cryptanalyst can perform cryptanalysis by exploiting the characteristic of DES algorithm but no one has succeeded in finding out the weakness.

**Weakness-** Weakness has been found in the design of the cipher:  
a. Two chosen input to an S-box can create the same output.  
b. The purpose of initial and final permutation is not clear.

## Difference between AES and DES ciphers

AES and DES are both examples of symmetric block ciphers but have certain dissimilarities.

AES	DES
AES stands for Advanced Encryption Standard	DES stands for Data Encryption Standard
The date of creation is 1999.	The date of creation is 1976.
Byte-Oriented.	Bit-Oriented.
Key length can be 128-bits, 192-bits, and 256-bits.	The key length is 56 bits in DES.
Number of rounds depends on key length: 10(128-bits), 12(192-bits), or 14(256-bits)	DES involves 16 rounds of identical operations
The structure is based on a substitution-permutation network.	The structure is based on a Feistel network.
The design rationale for AES is open.	The design rationale for DES is closed.
The selection process for this is secret but accepted for open public comment.	The selection process for this is secret.
AES is more secure than the DES cipher and is the de facto world standard.	DES can be broken easily as it has known vulnerabilities. 3DES(Triple DES) is a variation of DES which is secure than the usual DES.
The rounds in AES are: Byte Substitution, Shift Row, Mix Column and Key Addition	The rounds in DES are: Expansion, XOR operation with round key, Substitution and Permutation
AES can encrypt 128 bits of plaintext.	DES can encrypt 64 bits of plaintext.

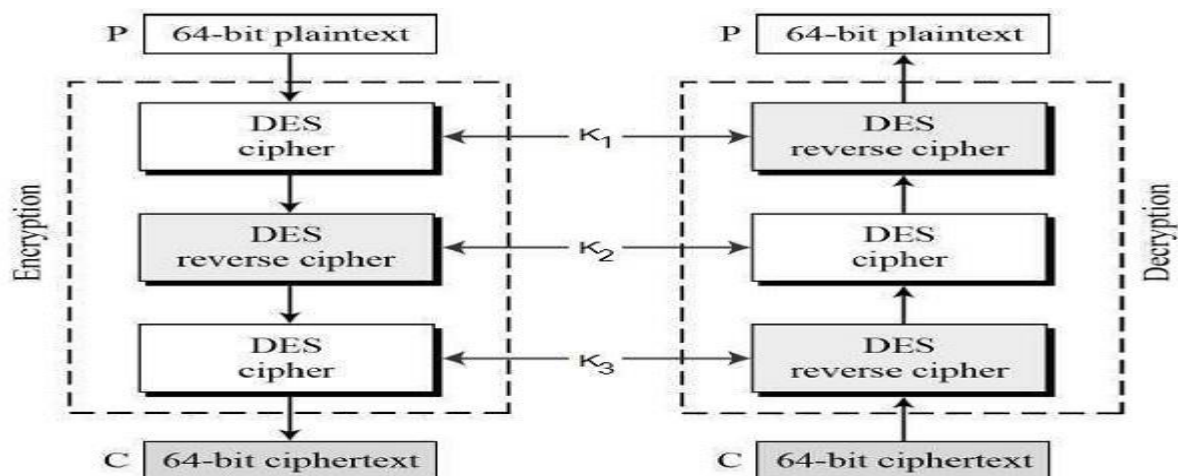
AES	DES
AES cipher is derived from an aside-channel square cipher.	DES cipher is derived from Lucifer cipher.
AES was designed by Vincent Rijmen and Joan Daemen.	DES was designed by IBM.
No known crypt-analytical attacks against AES but side channel attacks against AES implementations possible. Biclique attacks have better complexity than brute force but still ineffective.	Known attacks against DES include Brute-force, Linear crypt-analysis, and Differential crypt-analysis.

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).

### 3-KEY Triple DES

Before using 3TDES, user 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 –



The encryption-decryption process is as follows –

- Encrypt the plaintext blocks using single DES with key  $K_1$ .



- Now decrypt the output of step 1 using single DES with key  $K_2$ .
- Finally, encrypt the output of step 2 using single DES with key  $K_3$ .
- The output of step 3 is the ciphertext.
- Decryption of a ciphertext is a reverse process. User first decrypt using  $K_3$ , then encrypt with  $K_2$ , and finally decrypt with  $K_1$ .

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  $K_1$ ,  $K_2$ , and  $K_3$  to be the same value. This provides backwards compatibility with DES.

Second variant of Triple DES (2TDES) is identical to 3TDES except that  $K_3$  is replaced by  $K_1$ . In other words, user 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.

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

## Asymmetric-key algorithms

Also referred to as public-key algorithms, asymmetric-key algorithms use paired keys (a public and a private key) in performing their function. The public key is known to all, but the private key is controlled solely by the owner of that key pair. The private key cannot be mathematically calculated through the use of the public key even though they are cryptographically related. Asymmetric algorithms are used for:

- Computing digital signatures
- Establishing cryptographic keying material
- Identity Management

*Asymmetric-key algorithms* work in a similar manner to symmetric-key algorithms, where plaintext is combined with a key, input to an algorithm, and outputs ciphertext. The major difference is the keys used for the encryption and decryption portions are different, thus the asymmetry of the algorithm.

The *key pair* is comprised of a *private key* and a *public key*. As the names imply, the public key is made available to everyone, whereas the private key is kept secret. Which key is used for encryption and which key is used for decryption varies depending on the intended use of asymmetric-key algorithm in question.

The two main uses of asymmetric-key algorithms are *public-key encryption* and *digital signatures*. Public-key encryption is a method where anyone can send an encrypted message within a trusted network of users. The sender encrypts the message using the receiver's public key, allowing only the receiver to decrypt the message using his or her own private key. Anyone could intercept the encrypted message, but only the receiver can decrypt it. This makes public-key encryption an ideal method for protecting messages sent over unsecured mediums, such as the World Wide Web, where the sender has no control over how a message is routed to the sender.

The biggest vulnerability to asymmetric-key encryption is key management. Along with symmetric-key encryption, a compromised key is very bad, as it could be used to

disclose all information encrypted with that key. However, the additional threat faced by asymmetric-key encryption is the risk of an attacker using a compromised private key to send message on the victim's behalf. The message would encrypt and decrypt correctly, so there would be no indication of wrongdoing. In this sense, key management is even more critical in asymmetric-key encryption. Great care should be taken to manage the encryption key lifecycles from issuance, to renewal, and revocation.

## RSA Algorithm in Cryptography

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. **Public Key** and **Private Key**. As the name describes that the Public Key is given to everyone and the Private key is kept private.

### An example of asymmetric cryptography :

1. A client (for example browser) sends its public key to the server and requests some data.
2. The server encrypts the data using the client's public key and sends the encrypted data.
3. The client receives this data and decrypts it.

Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of the browser.

**The idea!** The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is a multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future. But till now it seems to be an infeasible task.

### Let us learn the mechanism behind RSA algorithm : >> Generating Public Key

Select two prime no's. Suppose  $P = 53$  and  $Q = 59$ .

Now First part of the Public key :  $n = P*Q = 3127$ .

We also need a small exponent say  $e$  :

But  $e$  Must be

An integer.

Not be a factor of  $n$ .

$1 < e < \phi(n)$  [ $\phi(n)$  is discussed below],

Let us now consider it to be equal to 3.

Our Public Key is made of  $n$  and  $e$

### >> Generating Private Key :

We need to calculate  $\phi(n)$  :

Such that  $\phi(n) = (P-1)(Q-1)$

so,  $\phi(n) = 3016$

Now calculate Private Key,  $d$  :

$d = (k*\phi(n) + 1) / e$  for some integer  $k$

For  $k = 2$ , value of  $d$  is 2011.

Now we are ready with our – Public Key (  $n = 3127$  and  $e = 3$ ) and Private Key( $d = 2011$ ) Now we will encrypt “HI” :

Convert letters to numbers :  $H = 8$  and  $I = 9$

Thus **Encrypted Data**  $c = 89^{e \bmod n}$ .

Thus our Encrypted Data comes out to be 1394

Now we will decrypt **1394** :

**Decrypted Data** =  $c^{d \bmod n}$ .

Thus our Encrypted Data comes out to be 89

$8 = H$  and  $I = 9$  i.e. "HI".

## Digital Signatures and Certificates

**Encryption** – Process of converting electronic data into another form, called ciphertext, which cannot be easily understood by anyone except the authorized parties. This assures data security.

**Decryption**– Process of translating code to data.

- The message is encrypted at the sender's side using various encryption algorithms and decrypted at the receiver's end with the help of the decryption algorithms.
- When some message is to be kept secure like username, password, etc., encryption and decryption techniques are used to assure data security.

### Types of Encryption

1. **Symmetric Encryption**– Data is encrypted using a key and the decryption is also done using the same key.
2. **Asymmetric Encryption**-Asymmetric Cryptography is also known as public-key cryptography. It uses public and private keys to encrypt and decrypt data. One key in the pair which can be shared with everyone is called the public key. The other key in the pair which is kept secret and is only known by the owner is called the private key. Either of the keys can be used to encrypt a message; the opposite key from the one used to encrypt the message is used for decryption.

**Public key**– Key which is known to everyone. Ex-public key of A is 7, this information is known to everyone.

**Private key**– Key which is only known to the person who's private key it is.

**Authentication**-Authentication is any process by which a system verifies the identity of a user who wishes to access it.

**Non- repudiation**– Non-repudiation means to ensure that a transferred message has been sent and received by the parties claiming to have sent and received the message. Non-repudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

**Integrity**– to ensure that the message was not altered during the transmission.

**Message digest** -The representation of text in the form of a single string of digits, created using a formula called a one way hash function. Encrypting a message digest with a private key creates a digital signature which is an electronic means of authentication..

## Digital Signature

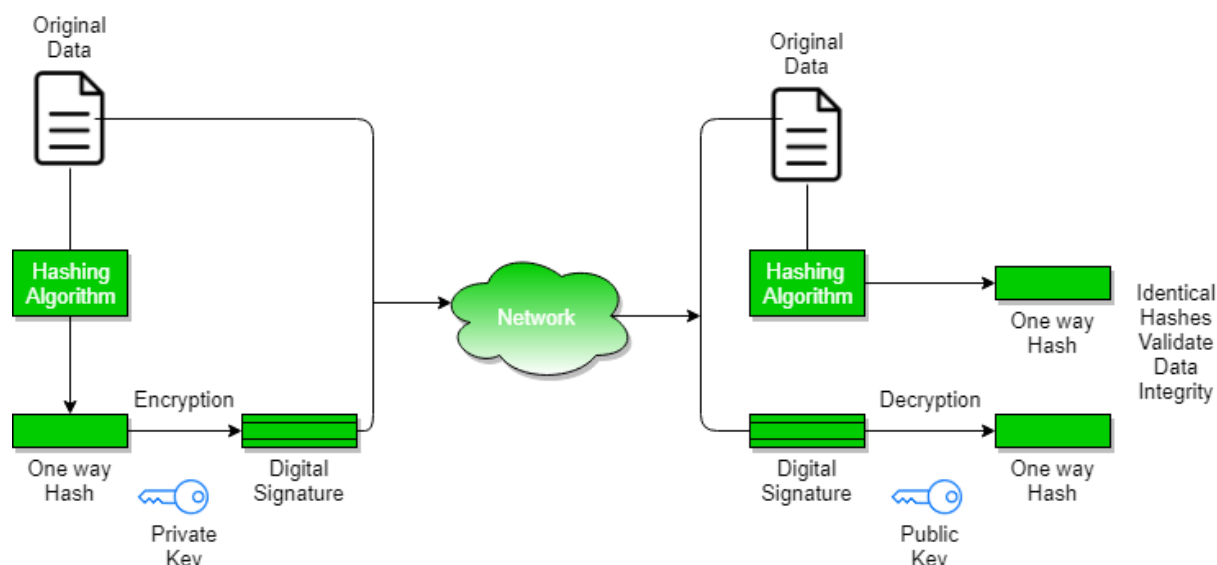
A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software, or digital document.

1. **Key Generation Algorithms:** Digital signature is electronic signatures, which assure that the message was sent by a particular sender. While performing digital transactions authenticity and integrity should be assured, otherwise, the data can be altered or someone can also act as if he was the sender and expect a reply.
2. **Signing Algorithms:** To create a digital signature, signing algorithms like email programs create a one-way hash of the electronic data which is to be signed. The signing algorithm then encrypts the hash value using the private key (signature key). This encrypted hash along with other information like the hashing algorithm is the digital signature. This digital signature is appended with the data and sent to the verifier. The reason for encrypting the hash instead of the entire message or document is that a hash function converts any arbitrary input into a much shorter fixed-length value. This saves time as now instead of signing a long message a shorter hash value has to be signed and moreover hashing is much faster than signing.
3. **Signature Verification Algorithms :** Verifier receives Digital Signature along with the data. It then uses Verification algorithm to process on the digital signature and the public key (verification key) and generates some value. It also applies the same hash function on the received data and generates a hash value. Then the hash value and the output of the verification algorithm are compared. If they both are equal, then the digital signature is valid else it is invalid.

### The steps followed in creating digital signature are :

1. Message digest is computed by applying hash function on the message and then message digest is encrypted using private key of sender to form the digital signature. (digital signature = encryption (private key of sender, message digest) and message digest = message digest algorithm(message)).
2. Digital signature is then transmitted with the message.(message + digital signature is transmitted)
3. Receiver decrypts the digital signature using the public key of sender.(This assures authenticity, as only sender has his private key so only sender can encrypt using his private key which can thus be decrypted by sender's public key).
4. The receiver now has the message digest.
5. The receiver can compute the message digest from the message (actual message is sent with the digital signature).
6. The message digest computed by receiver and the message digest (got by decryption on digital signature) need to be same for ensuring integrity.

Message digest is computed using one-way hash function, i.e. a hash function in which computation of hash value of a message is easy but computation of the message from hash value of the message is very difficult.



## Digital Certificate

Digital certificate is issued by a trusted third party which proves sender's identity to the receiver and receiver's identity to the sender. A digital certificate is a certificate issued by a Certificate Authority (CA) to verify the identity of the certificate holder. The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. Digital certificate is used to attach public key with a particular individual or an entity.

### Digital certificate contains:-

1. Name of certificate holder.
2. Serial number which is used to uniquely identify a certificate, the individual or the entity identified by the certificate
3. Expiration dates.
4. Copy of certificate holder's public key.(used for decrypting messages and digital signatures)
5. Digital Signature of the certificate issuing authority.

Digital certificate is also sent with the digital signature and the message.

### Digital certificate vs digital signature :

Digital signature is used to verify authenticity, integrity, non-repudiation ,i.e. it is assuring that the message is sent by the known user and not modified, while digital certificate is used to verify the identity of the user, maybe sender or receiver. Thus, digital signature and certificate are different kind of things but both are used for security. Most websites use digital certificate to enhance trust of their users.

Feature	Digital Signature	Digital Certificate
Basics / Definition	Digital signature is like a fingerprint or an attachment to a digital document that ensures its authenticity and integrity.	Digital certificate is a file that ensures holder's identity and provides security.

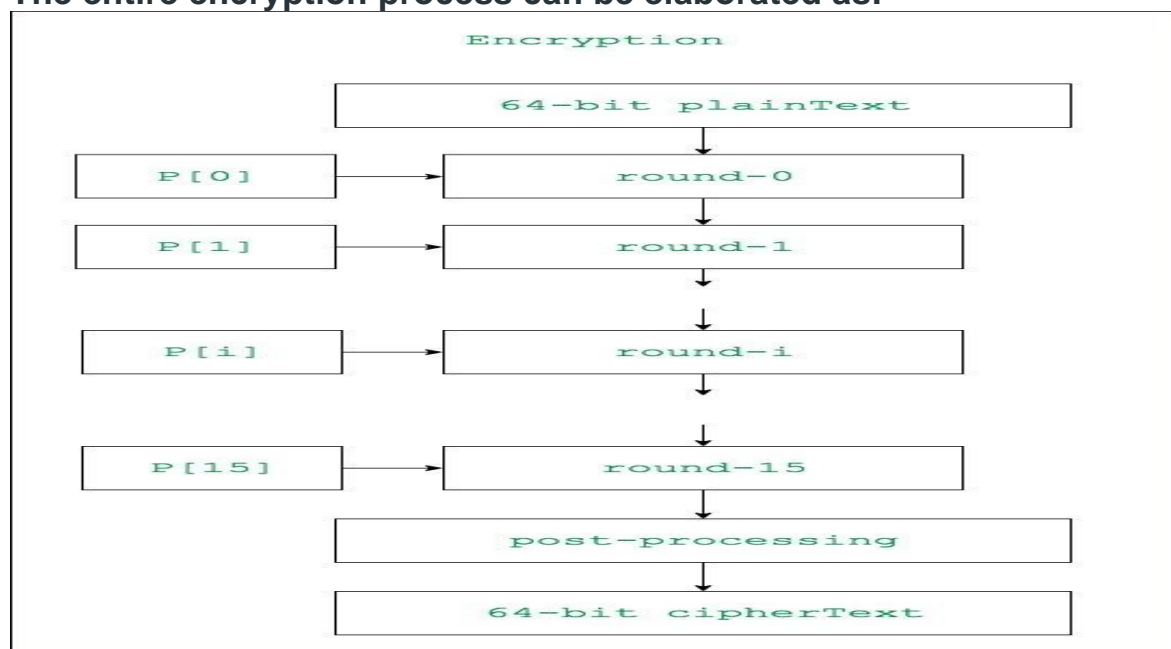
Process / Steps	Hashed value of original message is encrypted with sender's secret key to generate the digital signature.	It is generated by CA (Certifying Authority) that involves four steps: Key Generation, Registration, Verification, Creation.
Security Services	<b>Authenticity</b> of Sender, <b>integrity</b> of the document and <b>non-repudiation</b> . It follows Digital Signature Standard (DSS).	It provides security and <b>authenticity</b> of certificate holder.
Standard		It follows X.509 Standard Format

## Blowfish Algorithm with Examples

**Blowfish** is an encryption technique designed by **Bruce Schneier** in 1993 as an alternative to [DES Encryption Technique](#). It is significantly faster than DES and provides a good encryption rate with no effective [cryptanalysis technique](#) found to date. It is one of the first, secure block cyphers not subject to any patents and hence freely available for anyone to use.

1. **blockSize**: 64-bits
2. **keySize**: 32-bits to 448-bits variable size
3. **number of subkeys**: 18 [P-array]
4. **number of rounds**: 16
5. **number of substitution boxes**: 4 [each having 512 entries of 32-bits each]

The entire encryption process can be elaborated as:



Lets see each step one by one:

### Step1: Generation of subkeys:

- 18 subkeys{P[0]...P[17]} are needed in both encryption as well as decryption process and the same subkeys are used for both the processes.
- These 18 subkeys are stored in a P-array with each array element being a 32-bit entry.
- It is initialized with the digits of pi(?).



- The hexadecimal representation of each of the subkeys is given by:

P[0] = "243f6a88"

P[1] = "85a308d3"

.

.

P[17] = "8979fb1b"

32-bit hexadecimal representation of initial values of sub-keys					
P[0]	:	243f6a88	P[9]	:	38d01377
P[1]	:	85a308d3	P[10]	:	be5466cf
P[2]	:	13198a2e	P[11]	:	34e90c6c
P[3]	:	03707344	P[12]	:	c0ac29b7
P[4]	:	a4093822	P[13]	:	c97c50dd
P[5]	:	299f31d0	P[14]	:	3f84d5b5
P[6]	:	082efa98	P[15]	:	b5470917
P[7]	:	ec4e6c89	P[16]	:	9216d5d9
P[8]	:	452821e6	P[17]	:	8979fb1b

- Now each of the subkey is changed with respect to the input key as:

P[0] = P[0] xor 1st 32-bits of input key

P[1] = P[1] xor 2nd 32-bits of input key

.

.

P[i] = P[i] xor (i+1)th 32-bits of input key

(roll over to 1st 32-bits depending on the key length)

.

.

P[17] = P[17] xor 18th 32-bits of input key

(roll over to 1st 32-bits depending on key length)

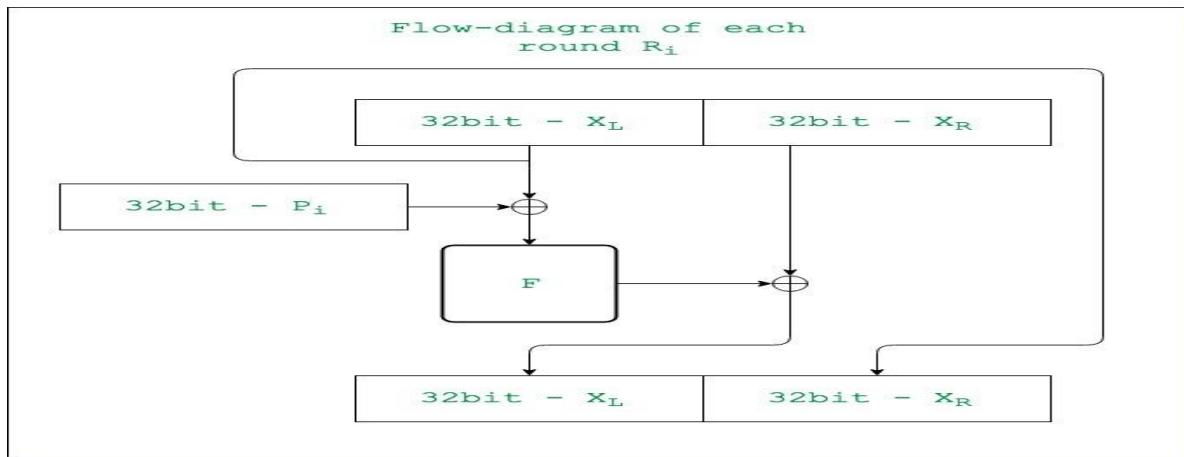
The resultant P-array holds 18 subkeys that is used during the entire encryption process

### **Step2: initialise Substitution Boxes:**

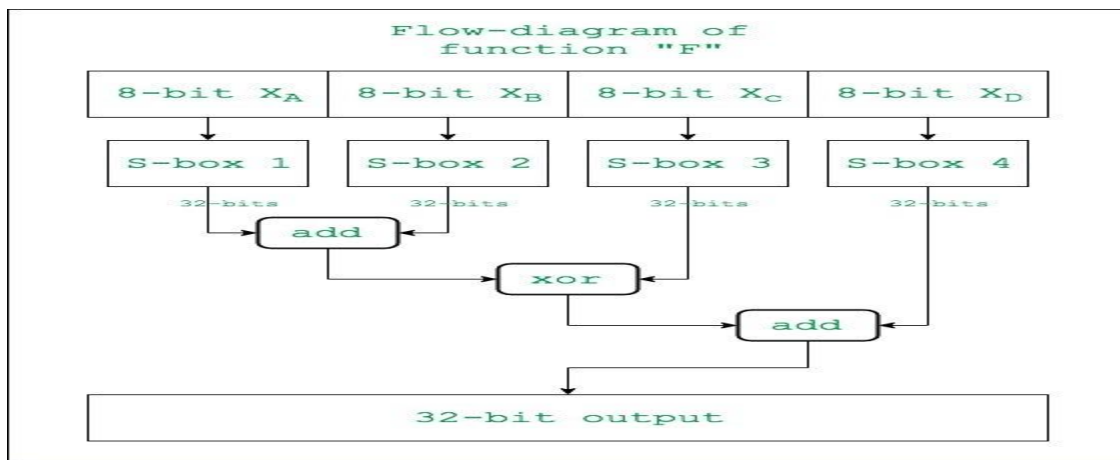
- 4 Substitution boxes(S-boxes) are needed{S[0]...S[4]} in both encryption as well as decryption process with each S-box having 256 entries{S[i][0]...S[i][255], 0≤i≤4} where each entry is 32-bit.
- It is initialized with the digits of pi(?) after initializing the P-array. [You may find the s-boxes in here!](#)

### **Step3: Encryption:**

- The encryption function consists of two parts:  
**a. Rounds:** The encryption consists of 16 rounds with each round(Ri) taking inputs the plainText(P.T.) from previous round and corresponding subkey(Pi). The description of each round is as follows:

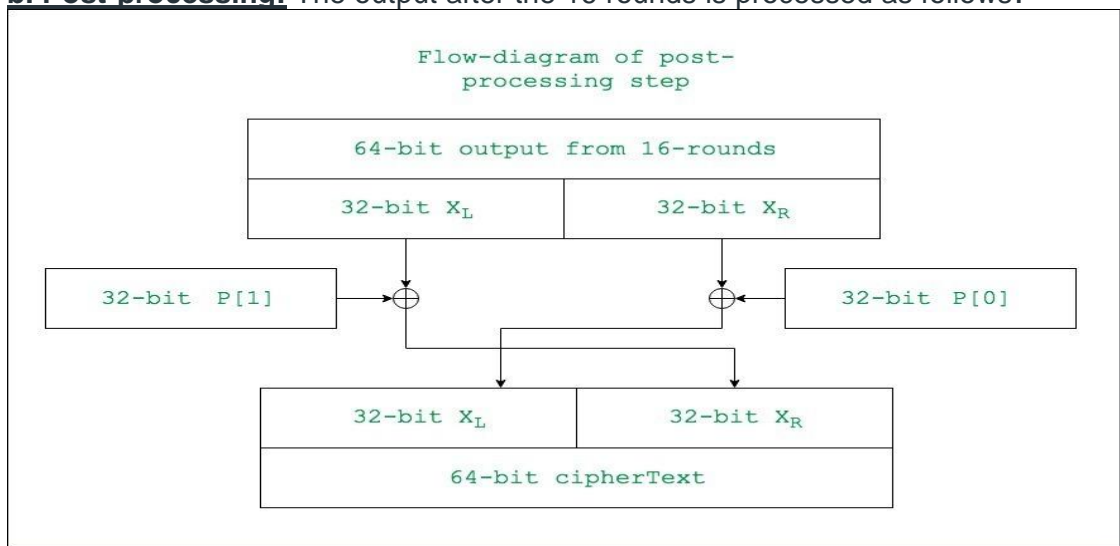


The description of the function "F" is as follows:



Here the function "add" is addition modulo  $2^{32}$ .

**b. Post-processing:** The output after the 16 rounds is processed as follows:



## UNIT-03

In today's world, we transfer the data in bulk, and the security of this data is very important, so Internet security provides that feature i.e., protection of data. There are different types of protocol exist like routing, mail transfer, and remote communication



protocol. But the Internet security protocol helps in the security and integrity of data over the internet. There are many protocols that exist that help in the security of data over the internet such as [Secure Socket Layer \(SSL\)](#), [Transport Layer Security \(TLS\)](#).

**Now, let us look at the various types of Internet Security Protocols :**

1. [SSL Protocol](#) :

- SSL Protocol stands for Secure Sockets Layer protocol, which is an encryption-based Internet security protocol that protects confidentiality and integrity of data.
- SSL is used to ensure the privacy and authenticity of data over the internet.
- SSL is located between the application and transport layers.
- At first, SSL contained security flaws and was quickly replaced by the first version of TLS that's why SSL is the predecessor of the modern TLS encryption.
- TLS/SSL website has "HTTPS" in its URL rather than "HTTP".
- SSL is divided into three sub-protocols: the Handshake Protocol, the Record Protocol, and the Alert Protocol.

2. [TLS Protocol](#) :

- Same as SSL, TLS which stands for Transport Layer Security is widely used for the privacy and security of data over the internet.
- TLS uses a pseudo-random algorithm to generate the master secret which is a key used for the encryption between the protocol client and protocol server.
- TLS is basically used for encrypting communication between online servers like a web browser loading a web page in the online server.
- TLS also has three sub-protocols the same as SSL protocol – Handshake Protocol, Record Protocol, and Alert Protocol.

3. **SHTTP :**

- SHTTP stands for Secure HyperText Transfer Protocol, which is a collection of security measures like Establishing strong passwords, setting up a firewall, thinking of antivirus protection, and so on designed to secure internet communication.
- SHTTP includes data entry forms that are used to input data, which has previously been collected into a database. As well as internet-based transactions.
- SHTTP's services are quite comparable to those of the SSL protocol.
- Secure HyperText Transfer Protocol works at the application layer (that defines the shared communications protocols and interface methods used by hosts in a network) and is thus closely linked with HTTP.
- SHTTP can authenticate and encrypt HTTP traffic between the client and the server.
- SHTTP operates on a message-by-message basis. It can encrypt and sign individual messages.

S-HTTP (Secure HTTP) is an extension to the Hypertext Transfer Protocol ([HTTP](#)) that allows the secure exchange of files on the World Wide Web. Each S-HTTP file is either encrypted, contains a digital certificate, or both. For a given document, S-HTTP is an alternative to another well-known security protocol, Secure Sockets Layer (SSL). A

major difference is that S-HTTP allows the client to send a certificate to authenticate the user whereas, using SSL, only the server can be authenticated. S-HTTP is more likely to be used in situations where the server represents a bank and requires authentication from the user that is more secure than a userid and password.

S-HTTP does not use any single encryption system, but it does support the Rivest-Shamir-Adleman public key infrastructure encryption system. SSL works at a program layer slightly higher than the Transmission Control Protocol (TCP) level. S-HTTP works at the even higher level of the HTTP application. Both security protocols can be used by a browser user, but only one can be used with a given document. Terisa Systems includes both SSL and S-HTTP in their Internet security tool kits.

A number of popular Web servers support both S-HTTP and SSL. Newer browsers support both SSL and S-HTTP. S-HTTP has been submitted to the Internet Engineering Task Force (IETF) for consideration as a standard

#### 4. Set Protocol :

- Secure Electronic Transaction (SET) is a method that assures the security and integrity of electronic transactions made using credit cards.
- SET is not a payment system; rather, it is a secure transaction protocol that is used via the internet.
- The SET protocol provides the following services:
  - It establishes a safe channel of communication between all parties engaged in an e-commerce transaction.
  - It provides confidentiality since the information is only available to the parties engaged in a transaction when and when it is needed.
- The SET protocol includes the following participants:
  - **Cardholder**
  - **Merchant**
  - **Issuer**
  - **Acquire**
  - **Payment Gateway**
  - **Certification Authority**

#### • PEM Protocol :

- PEM Protocol stands for privacy-enhanced mail and is used for email security over the internet.
- RFC 1421, RFC 1422, RFC 1423, and RFC 1424 are the four particular papers that explain the Privacy Enhanced Mail protocol.
- It is capable of performing cryptographic operations such as encryption, nonrepudiation, and message integrity.

#### 5. **PGP Protocol** :

- PGP Protocol stands for Pretty Good Privacy, and it is simple to use and free, including its source code documentation.
- It also meets the fundamental criteria of cryptography.
- When compared to the PEM protocol, the PGP protocol has grown in popularity and use.
- The PGP protocol includes cryptographic features such as encryption, non-repudiation, and message integrity.

## Socket Layer (SSL)

**Secure Socket Layer (SSL)** provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

### Secure Socket Layer Protocols:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol
- Alert protocol

SSL Protocol Stack:

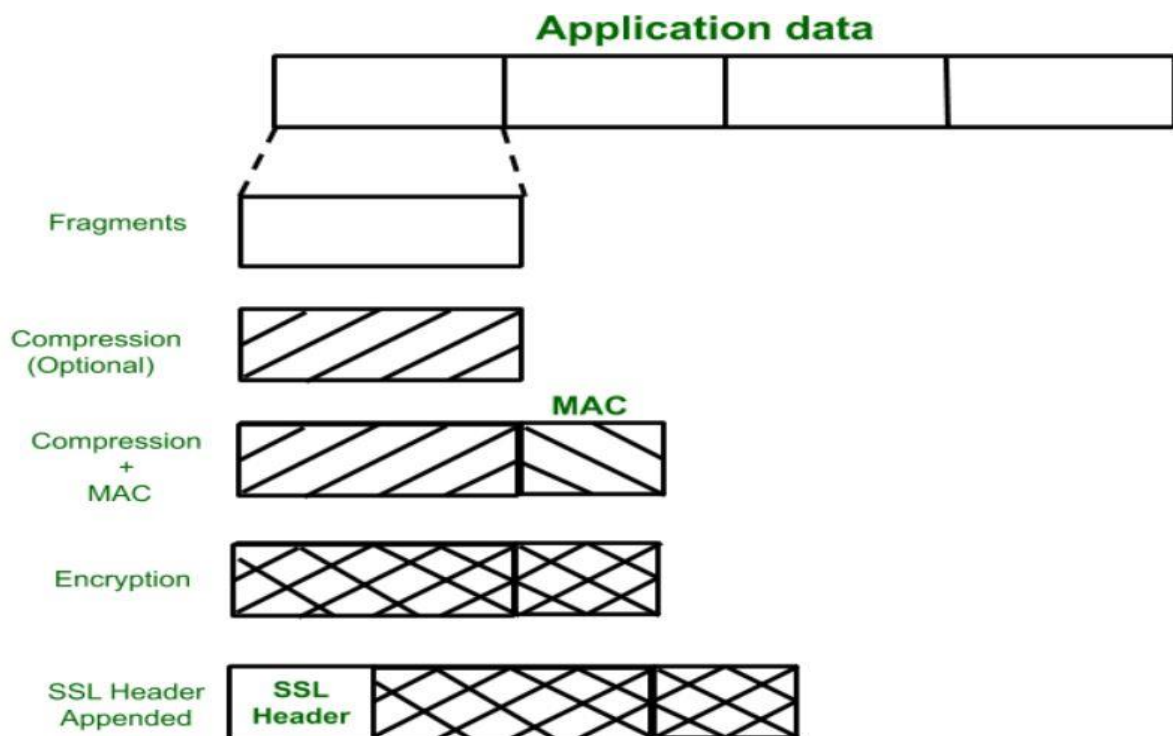
Handshake Protocol	Change Cipher Spec Protocol	Alert Protocol	HTTP
SSL Record Protocol			
TCP			
IP			

SSL Record Protocol:

SSL Record provides two services to SSL connection.

- Confidentiality
- Message Integrity

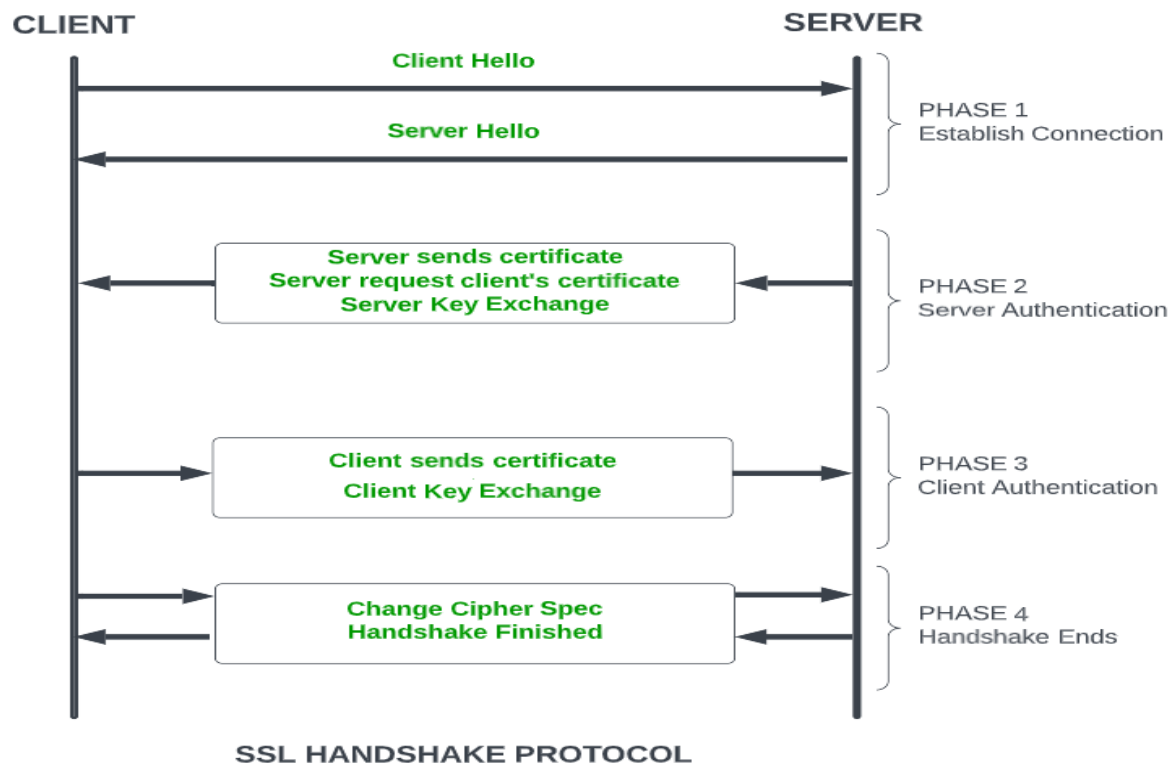
In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.



### Handshake Protocol:

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

- **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
- **Phase-2:** Server sends his certificate and Server-key-exchange. The server ends phase-2 by sending the Server-hello-end packet.
- **Phase-3:** In this phase, Client replies to the server by sending his certificate and Client-exchange-key.
- **Phase-4:** In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends.



*SSL Handshake Protocol Phases diagrammatic representation*

#### Change-cipher Protocol:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current state. Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.



#### Alert Protocol:

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.



The level is further classified into two parts:

#### Warning (level = 1):

This Alert has no impact on the connection between sender and receiver. Some of them are:

**Bad certificate:** When the received certificate is corrupt.

**No certificate:** When an appropriate certificate is not available.

**Certificate expired:** When a certificate has expired.

**Certificate unknown:** When some other unspecified issue arose in processing the certificate, rendering it unacceptable.

**Close notify:** It notifies that the sender will no longer send any messages in the connection.

**Fatal Error (level = 2):**

This Alert breaks the connection between sender and receiver. The connection will be stopped, cannot be resumed but can be restarted. Some of them are :

**Handshake failure:** When the sender is unable to negotiate an acceptable set of security parameters given the options available.

**Decompression failure:** When the decompression function receives improper input.

**Illegal parameters:** When a field is out of range or inconsistent with other fields.

**Bad record MAC:** When an incorrect MAC was received.

**Unexpected message:** When an inappropriate message is received.

The second byte in the Alert protocol describes the error.

Silent Features of Secure Socket Layer:

- The advantage of this approach is that the service can be tailored to the specific needs of the given application.
- Secure Socket Layer was originated by Netscape.
- SSL is designed to make use of TCP to provide reliable end-to-end secure service.
- This is a two-layered protocol.

## Transport Layer Security (TLS)

Transport Layer Securities (TLS) are designed to provide security at the transport layer. TLS was derived from a security protocol called Secure Socket Layer (SSL).

TLS ensures that no third party may eavesdrop or tampers with any message.

There are several benefits of TLS:

- **Encryption:**  
TLS/SSL can help to secure transmitted data using encryption.
- **Interoperability:**  
TLS/SSL works with most web browsers, including Microsoft Internet Explorer and on most operating systems and web servers.
- **Algorithm flexibility:**  
TLS/SSL provides operations for authentication mechanism, encryption algorithms and hashing algorithm that are used during the secure session.
- **Ease of Deployment:**  
Many applications TLS/SSL temporarily on a windows server 2003 operating systems.
- **Ease of Use:**  
Because we implement TLS/SSL beneath the application layer, most of its operations are completely invisible to client.

**Working of TLS:**

The client connect to server (using TCP), the client will be something. The client sends number of specification:

1. Version of SSL/TLS.
2. which cipher suites, compression method it wants to use.

The server checks what the highest SSL/TLS version is that is supported by them both, picks a cipher suite from one of the clients option (if it supports one) and

optionally picks a compression method. After this the basic setup is done, the server provides its certificate. This certificate must be trusted either by the client itself or a party that the client trusts. Having verified the certificate and being certain this server really is who he claims to be (and not a man in the middle), a key is exchanged. This can be a public key, "PreMasterSecret" or simply nothing depending upon cipher suite.

Both the server and client can now compute the key for symmetric encryption. The handshake is finished and the two hosts can communicate securely. To close a connection by finishing. TCP connection both sides will know the connection was improperly terminated. The connection cannot be compromised by this through, merely interrupted.

### **Time Stamping Protocols:**

The timestamp protocols ensures that each transaction in the system has in advance a timestamp that has been associated with each transaction that is being helpful to the transaction to be executed in the system that time only.

It is most helpful in the case when large number of concurrent processes are running in the system simultaneously. So, it assign a unique timestamp to each transaction in the system with the help of ts counter.

#### **ts counter:**

ts counter is a counter that is used for time-stamping protocols. It increment its value by 1 when each commit operation has been occur in the system. If a transaction  $T_i$  has been assigned timestamp  $TS[T_i]$ , and a new transaction enters in the system, then it must hold a condition that  $TS[T_i] < TS[T_i]$

The **Time-Stamp Protocol**, or

**TSP** is a cryptographic protocol for certifying timestamps using X.509 certificates and public key infrastructure. The timestamp is the signer's assertion that a piece of electronic data existed at or before a particular time. The protocol is defined in RFC 3161. One application of the protocol is to show that a digital signature was issued before a point in time, for example before the corresponding certificate was revoked.

The TSP protocol is an example of trusted timestamping. It has been extended to create the ANSI ASC X9.95 Standard.

### **Protocol**

---

In the protocol a Time Stamp Authority (TSA) is a trusted third party that can provide a timestamp to be associated with a hashed version of some data. It is a request-response protocol, where the request contains a hash of the data to be signed. This is sent to the TSA and the response contains a Time Stamp Token (TST) which itself includes the hash of the data, a unique serial number, a timestamp and a digital signature. The signature is generated using the private key of the TSA. The protocol can operate over a number of different transports, including email, TCP sockets or HTTP.

When presented with a TST, someone may verify that the data existed at the timestamp in the TST by verifying the signature using the public key of the TSA and that the hash of the data matches that included in the TST.

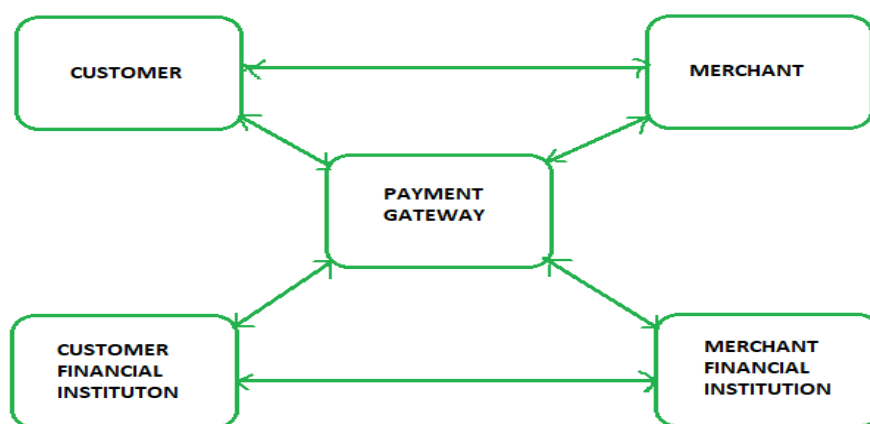
**Secure Electronic Transaction** or SET is a system that ensures the security and integrity of electronic transactions done using credit cards in a scenario. SET is not



some system that enables payment but it is a security protocol applied to those payments. It uses different encryption and hashing techniques to secure payments over the internet done through credit cards. The SET protocol was supported in development by major organizations like Visa, Mastercard, Microsoft which provided its Secure Transaction Technology (STT), and Netscape which provided the technology of Secure Socket Layer (SSL).

SET protocol restricts the revealing of credit card details to merchants thus keeping hackers and thieves at bay. The SET protocol includes Certification Authorities for making use of standard Digital Certificates like X.509 Certificate.

Before discussing SET further, let's see a general scenario of electronic transactions, which includes client, payment gateway, client financial institution, merchant, and merchant financial institution.



### Requirements in SET :

The SET protocol has some requirements to meet, some of the important requirements are :

- It has to provide mutual authentication i.e., customer (or cardholder) authentication by confirming if the customer is an intended user or not, and merchant authentication.
- It has to keep the PI (Payment Information) and OI (Order Information) confidential by appropriate encryptions.
- It has to be resistive against message modifications i.e., no changes should be allowed in the content being transmitted.
- SET also needs to provide interoperability and make use of the best security mechanisms.

### Participants in SET :

In the general scenario of online transactions, SET includes similar participants:

1. **Cardholder** – customer
2. **Issuer** – customer financial institution
3. **Merchant**
4. **Acquirer** – Merchant financial
5. **Certificate authority** – Authority that follows certain standards and issues certificates (like X.509V3) to all other participants.

### SET functionalities :

- **Provide Authentication**



- **Merchant Authentication** – To prevent theft, SET allows customers to check previous relationships between merchants and financial institutions. Standard X.509V3 certificates are used for this verification.
- **Customer / Cardholder Authentication** – SET checks if the use of a credit card is done by an authorized user or not using X.509V3 certificates.
- **Provide Message Confidentiality:** Confidentiality refers to preventing unintended people from reading the message being transferred. SET implements confidentiality by using encryption techniques. Traditionally DES is used for encryption purposes.
- **Provide Message Integrity:** SET doesn't allow message modification with the help of signatures. Messages are protected against unauthorized modification using RSA digital signatures with SHA-1 and some using HMAC with SHA-1,

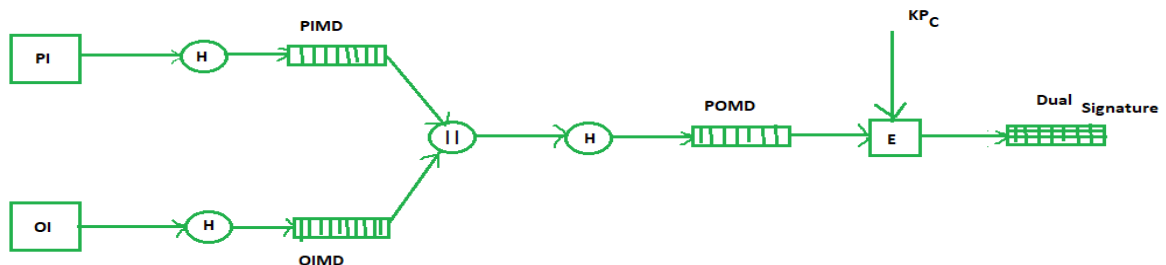
### Dual Signature :

The dual signature is a concept introduced with SET, which aims at connecting two information pieces meant for two different receivers :

#### Order Information (OI) for merchant

#### Payment Information (PI) for bank

You might think sending them separately is an easy and more secure way, but sending them in a connected form resolves any future dispute possible. Here is the generation of dual signature:



Where,

PI stands for payment information  
 OI stands for order information  
 PIMD stands for Payment Information Message Digest  
 OIMD stands for Order Information Message Digest  
 POMD stands for Payment Order Message Digest  
 H stands for Hashing  
 E stands for public key encryption  
 KpC is customer's private key  
 || stands for append operation  
 Dual signature,  $DS = E(KpC, [H(H(PI) || H(OI))])$

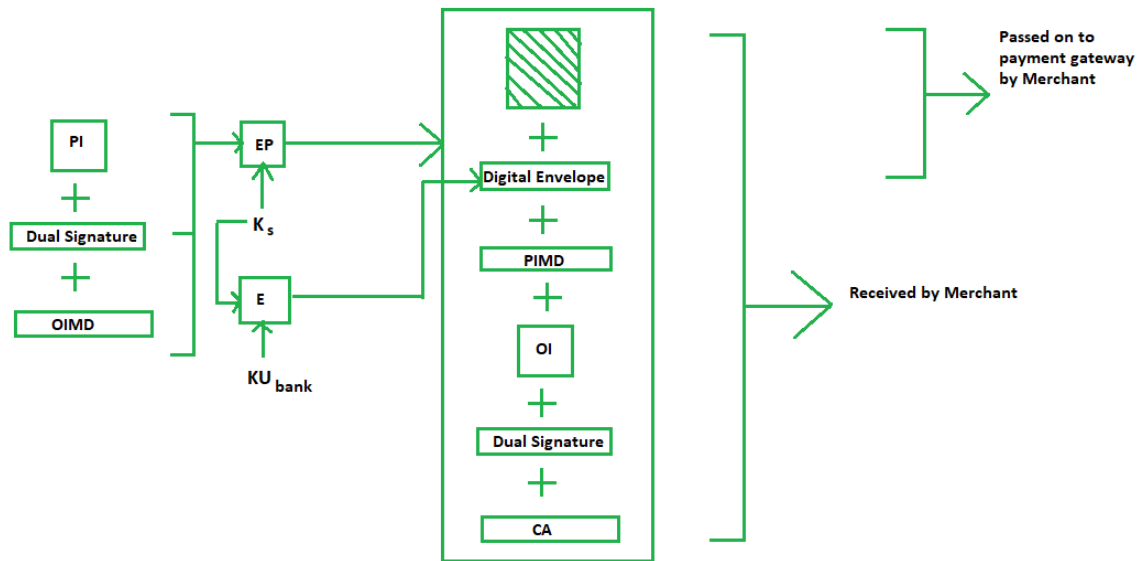
#### Purchase Request Generation :

The process of purchase request generation requires three inputs:

- Payment Information (PI)

- Dual Signature
- Order Information Message Digest (OIMD)

The purchase request is generated as follows:



Here,

PI, OIMD, OI all have the same meanings as before.

The new things are :

EP which is symmetric key encryption

Ks is a temporary symmetric key

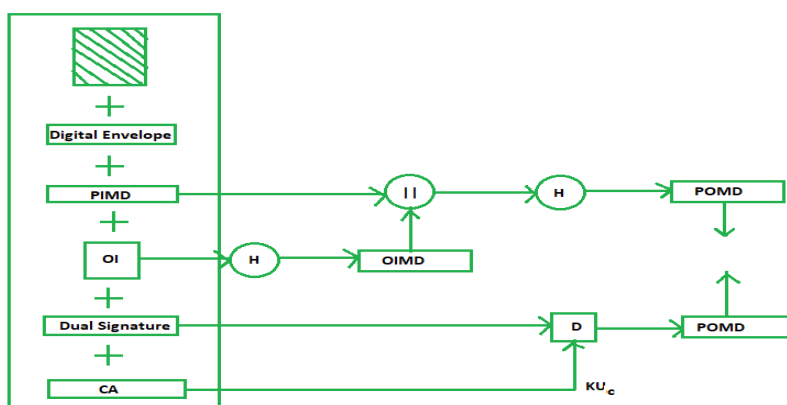
KU<sub>bank</sub> is public key of bank

CA is Cardholder or customer Certificate

Digital Envelope =  $E(KU_{bank}, K_s)$

### Purchase Request Validation on Merchant Side :

The Merchant verifies by comparing POMD generated through PIMD hashing with POMD generated through decryption of Dual Signature as follows:



Since we used Customer's private key in encryption here we use KUC which is the public key of the customer or cardholder for decryption 'D'.

**Payment Authorization and Payment Capture :**

Payment authorization as the name suggests is the authorization of payment information by the merchant which ensures payment will be received by the merchant. Payment capture is the process by which a merchant receives payment which includes again generating some request blocks to gateway and payment gateway in turn issues payment to the merchant.

**Difference between Secure Socket Layer (SSL) and Secure Electronic Transaction (SET):**

S.  
No.

## Secure Socket Layer

### Basics-

SSL is an encryption mechanism for order taking, queries, and other applications and is available on the customer's browser. It does not protect against all security hazards and is naturally simple and widely used. SSL is a protocol for general-purpose secure message exchange. SSL protocol may use a certificate, but the payment gateway is not available. So, the merchant needs to receive both the ordering information and credit card information because the capturing process should be generated by the merchant. SSL protocol has been the industry standard for securing internet communication.

1.

### Developed by-

SSL protocol was developed by Netscape for the secure online transaction.

2.

### Working-

SSL uses a combination of public-key and symmetric-key encryption to safeguard data transactions. The handshake technique is used by the SSL protocol, which permits the server to verify its identity to the client. In case of unsuccessful authentication, the connection will not be formed.

3.

### Integrity-

The technique of Hash functions is used for this purpose.

4.

5. **Acceptability-**

## Secure Electronic Transaction

### Basics-

SET is a very comprehensive protocol. It provides privacy, integration, and authenticity. It is not used frequently due to its complexity and the need for a special card reader by the user. It may be abandoned if it is not simplified. SET is tailored to the credit card payment to the merchant. SET protocols hide the customer's credit card information from merchant and also hides the order information from banks to protect privacy called a **dual signature**. The SET protocol is complex and more secure.

### Developed by-

The SET protocol was jointly developed by MasterCard and visa to secure web browsers for a bank card transaction.

### Working-

The dual signature mechanism is deployed by SET to safeguard a transaction. To use an e-commerce site, SET requires the purchase of software. The design of the protocol necessitates the client's installation of an e-wallet.

### Integrity-

The technique of digital signatures is used for this purpose.

**Acceptability-**

S.  
No.

### Secure Socket Layer

Its acceptability is more as compared to SET.

#### **Functionality-**

The Secure Sockets Layer (SSL) is not a payment protocol. SSL encrypts the communication channel between the cardholder and the merchant website and is not backed by any financial institution. As a result, SSL is unable to ensure the security of a transaction.

6.

#### **Encryption-**

The purpose of SSL lies in prevention of data tampering in client/server applications and has considerably weaker encryption, with a maximum of 128-bit encryption.

7.

#### **Authentication-**

SSL certificates are not endorsed by any financial institution or payment brand association, so they cannot effectively validate all parties.

8.

#### **Security-**

SSL only protects the cardholder and the merchant, which is insufficient to prevent fraud. SSL transactions, in other words, are never assured.

9.

### Secure Electronic Transaction

SET acceptability is less because it's necessary to build an open PKI.

#### **Functionality-**

SET was created with the sole purpose of securing and ultimately guaranteeing a payment transaction. For example, increase in the possibilities for online retail growth only when consumer confidence grows in online shopping.

#### **Encryption-**

SET, which was created expressly to address the security of all parties involved in an electronic payment transaction, uses 1024-bit encryption throughout the transaction.

#### **Authentication-**

Here, all parties get authentication to the transaction because SET's certificates are backed not just by a Certificate Authority, but also by financial institutions and MasterCard International.

#### **Security-**

SET enables transaction security from the cardholder's desktop to the merchant via bank approvals and back through the gateway, leaving an indisputable audit trail and, as a result, a guaranteed transaction.

- **Electronic money** refers to the currency electronically stored on electronic systems and digital databases used to make it easier to transact electronically. It is popularly referred to by many names, including digital cash, digital currency, e-money, and so on.
- Fiat money, simply put, is a legal tender, whose value as a currency is established by an issuing government and consequently, is also regulated by it.
- Electronic money can be classified into two broad categories: hard and soft.

### What is Fiat Currency (or Fiat Money)?

Fiat money, simply put, is a legal tender, whose value as a currency is established by an issuing government and consequently, is also regulated by it. Fiat money is the exact opposite of commodity money, whose value is based on an underlying asset, such as gold or silver.

### Classifications of Electronic Money

Electronic money can be classified into two broad categories:

#### 1. *Hard*

Hard electronic money is when e-money is used for irreversible transactions, ones that are highly securitized, and are more or less procedural in nature. They may include transactions that are drawn through a bank.

#### 2. *Soft*

Soft electronic money is when e-money is used for reversible or flexible transactions. There is an increased level of flexibility offered, and users are allowed to manage their transactions even after payment is processed, like canceling a transaction or modifying the payment price, etc.

The changes can be made post-transaction within a defined period. They may include transactions that are passed through payment mechanisms like PayPal, PayTM, Interac, credit cards, and so on.

### Features of Electronic Money

Just like physical paper currency, electronic money also includes the following four features:

- **Store of value:** Just like physical currency, electronic money is also a store of value, the only difference being, that with electronic money, the value is stored electronically unless and until withdrawn physically.
- **Medium of exchange:** Electronic money is a medium of exchange, i.e., it is used to pay for the purchase of a good or when acquiring a service.
- **Unit of account:** Just like paper currency, electronic money provides a common measure of the value of the goods and/or services being transacted.
- **Standard of deferred payment:** Electronic money is used as a means of deferred payment, i.e., used for the tools of providing credit for repayment at a future date.

### Advantages of Electronic Money

Electronic money offers several advantages for the global economy, including:

### *1. Increased flexibility and convenience*

The use of electronic money brings increased flexibility and convenience to the table. Transactions can be entered into from anywhere in the world, at any given time, with one click of a button. It removes the hassle and tediousness involved with the physical delivery of payments.

### *2. Historical record*

The usage of electronic money is becoming increasingly popular because it stores a digital historical record of each and every transaction made. It makes tracing back payments easier and also helps with making detailed expenditure reports, [budgeting](#), and so on.

### *3. Prevents fraudulent activities*

Since electronic money makes available a detailed historical record of each and every transaction made, it is very easy to keep track of transactions and trace them back through the economy. It increases security and helps prevent fraudulent activities and malpractices.

### *4. Instantaneous*

The use of electronic money brings with it a kind of instantaneousness that has not been experienced before in the economy. Transactions can be completed in split seconds with the click of a button from virtually anywhere in the world. It eliminates problems of physical delivery of payments, including long queues, wait times, etc.

### *5. Increased security*

The use of e-money also brings with it an increased sense of security. To prevent loss of personal information while transacting online, advanced security measures are implemented like authentication and tokenization. Stringent verification measures are also employed to ensure the full authenticity of the transaction.

### **Disadvantages of Electronic Money**

Electronic money comes with the following disadvantages:

#### *1. Necessity of certain infrastructure*

To use electronic money, the availability of certain infrastructure is necessary. It includes a computer or a laptop, or a smartphone, and a stable internet connection.

#### *2. Possible security breaches/hacks*

The internet always comes with the inevitability of possible security breaches and hacks. A hack can leak sensitive personal information and can lead to fraud and money laundering.

#### *3. Online scams*

Online scamming is also possible. All it takes for a scammer is to pretend to be from a certain organization or a bank, and consumers are easily convinced to give away their bank/card details. Despite the increased security and presence of authentication measures to counter online scams, they are still something to be looked after.

## What Is Email Security?

Email security can be defined as the use of various techniques to keep sensitive information in email communication and accounts secure. These precautions are taken chiefly against unauthorized access, loss, or compromise. It allows an individual or an organization to protect the overall access to one or more email addresses or accounts.

Email security safeguards the content of an email account or service that generally serves as a popular medium for the spread of malware, spam, and phishing attacks. This is usually done using deceptive messages to entice recipients to divulge sensitive information, open attachments, or click on hyperlinks that install malware on the victim's device.

### Stopping attacks at the entry point:

Email is also a common entry point for attackers looking to gain a foothold in an enterprise network and breach valuable business data. Hence, email security is necessary for both individual and business email accounts, and there are multiple measures organizations should take to enhance email security. Some of the proactive email security measures, from an end user's standpoint, include:

1.
  - Strong passwords
  - Password rotations
  - Spam filters
  - Desktop-based anti-virus or anti-spam application

Similarly, a service provider ensures email security by using strong passwords and access-control mechanisms on an email server. Here, the email messages are encrypted and digitally signed as they are placed in the inbox or are in transit to or from a subscriber's email address.

The service provider also implements firewall and software-based spam filtering applications to restrict unsolicited, untrustworthy, and malicious email messages from being delivered to a user's inbox.

## Why Is Email Security Important?

Email is a popular attack vector. Therefore, enterprises and individuals must secure their email accounts against common attacks and attempt to gain unauthorized access to the communications' accounts or content. Email security is significant due to the following plausible reasons:

### 1. Email is a common target for cyber-criminals

The first thing that employees get as soon as they join any firm is their official email account. The employees use their respective email IDs to access the company information and communicate with fellow employees daily. Every official communication to or from the company uses email as a medium.

Therefore, when employees work remotely, as observed in the COVID-19 pandemic situation, they tend to use their official emails for almost all communication. Such communications are vulnerable, and the employees are at the risk of being attacked by cyber-criminals. Cyber-criminals often use phishing, baits, social engineering, and many other types of attacks to exploit cracks in the security system.



## **2. A small loophole can affect the entire organization**

A small loophole in an email's security can allow deadly malware or spyware to sneak into the entire communication network, wreaking havoc in the entire organization. The situation worsens when the organization's network is hit by deadly ransomware.

Even seasoned professionals and experienced employees fall victim to such tactics. These cyber thieves may also leak sensitive information by bringing it into the public domain or by selling it to bidders in the case of a personal vendetta.

## **3. Crucial for organizations to protect sensitive information**

The company's confidential information may include highly sensitive information that can be used against the organization or for criminal purposes. Cyber-criminals can also target day-to-day communication and change messages, which can create miscommunication and compel the communicators (the employees) to release or hide relevant information.

Such forgery may result in identity theft that can ultimately cause a breach of sensitive information. Both employees and organizations have to bear in mind that cyber-criminals only need a [thin crack in security](#) to jeopardize the entire workflow.

## **4. Below-average standard security measures**

Although email service providers employ standard security measures, cyber-criminals can easily circumvent many of these measures. Generally, standard email defenses can only stop threats that are already known to them. In some situations, the email system also prompts its users to decide whether the received messages are secure or not and act accordingly. Advanced threat detection systems currently use artificial intelligence databases, real-time analysis, and machine learning for better protection.

However, such sophisticated advanced techniques are not employed under standard email security measures. That's why it has become essential for users to use advanced security measures to identify a wide range of threats well in advance and stop them from entering the system.

## **5. Cyber-criminals use sophisticated & advanced methods**

The technological advancement in online security systems can also empower cyber-criminals and hackers as they tend to use more and more advanced methods to breach various security firewalls.

These methods include AI fuzzing (AIF) and machine learning poisoning (MLP), enabling hackers to automate cyber-attacks. Besides, cyber-criminals can exploit many cloud vulnerabilities, damaging an organization's workflow, business, image, and credibility in the industry.

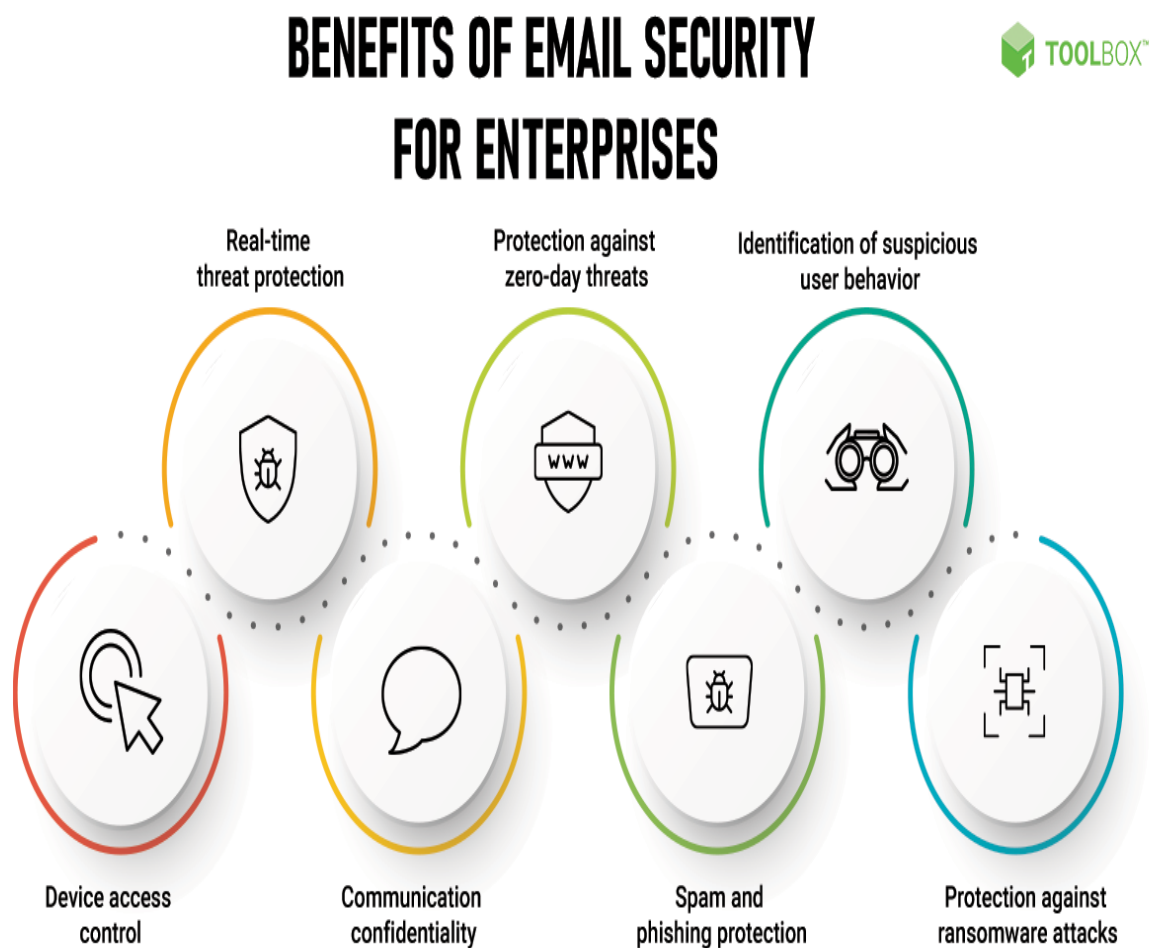
All of these threats are grave enough for any organization, and hence it is crucial to install advanced email security measures. Organizations can also opt for security service providers who offer potent and improved email and overall online security.

## Benefits of Email Security for Enterprises

In today's digital world, email has become an indispensable part of our lives, as almost every communication in an enterprise happens here. Even though emails are used so freely, it is important for an organization not to become complacent about protecting the data shared via emails, as they can cause severe damage to its business.

With the growing threat of hackers, viruses, spam, phishing, identity theft, and ransomware attacks, organizations have an added responsibility to secure their business data and treat email security as a priority.

Following are some of the benefits of email security for enterprises:



## Benefits of Email Security for Enterprises

### **1. Control device access**

Email security can prevent total access to sensitive email attachments on vulnerable unmanaged devices while permitting full access to secure managed devices. It ensures that there is control over an email account's content access, thereby securing the content communicated during an email message exchange.

### **2. Identify suspicious user behavior**

Email security can assist in the assessment of outbound messages for detecting known spam and threats. This control on outbound email messages can help discover account compromise or other suspicious user behavior. The right kind of email security platform should detect and alert users who have started to send or receive large volumes of outbound email, including bulk messages. Such activity can be an indicator of compromise.

### **3. Improve spam and phishing protection**

Powerful phishing protection solutions are required to defend an organization against ever-evolving phishing attacks. In phishing attacks, hackers send an email that appears to be from a legitimate sender to deceive the employees into revealing sensitive information like bank account numbers, account passwords, or credit card information. Sophisticated email security measures help detect unwanted spam and unsafe phishing emails, thereby allowing customers to block or take other actions against suspicious emails.

### **4. Maintain communication confidentiality**

Email security techniques such as email encryption software or programs keep the confidentiality of email messaging intact. Email encryption is critical, specifically when sending any confidential information or during sensitive communications. Thus, email encryption can serve as a tangible investment that can save the organization from compromising its sensitive data.

### **5. Protection against zero-day threats**

Email security is of paramount importance for protecting an organization against a zero-day threat, as attacks are generally initiated via a malicious link or rogue attachment. Preventing a zero-day attack requires multiple layers of protection to defend against malware, viruses, and spam, as well as targeted attacks such as phishing, spear-phishing, or whaling attack. Email security such as a Secure Email Gateway helps prevent a zero-day attack by providing anti-malware and anti-spam protection.

### **6. Real-time threat protection**

Email security provides real-time threat protection capability by using a unique blend of detection technologies, such as machine learning, sandboxing, and predictive analytics, to effectively stop advanced threats such as ransomware.

### **7. Stop ransomware attacks and other threats**

Ransomware attacks involve gaining access to an organization's computer systems or sensitive data and blocking access to the data by or encrypting it. Attackers allow the organizations to access their data only after making a payment, or "ransom." Because ransomware attacks are often initiated via email, blocking malicious URLs and weaponized attachments is the most effective way to stop ransomware attacks. That's why, when email security is in place, it helps in identifying threats ranging from annoying spam to advanced malware, phishing, and Business Email Compromise (BEC) attacks.

## UNIT-04

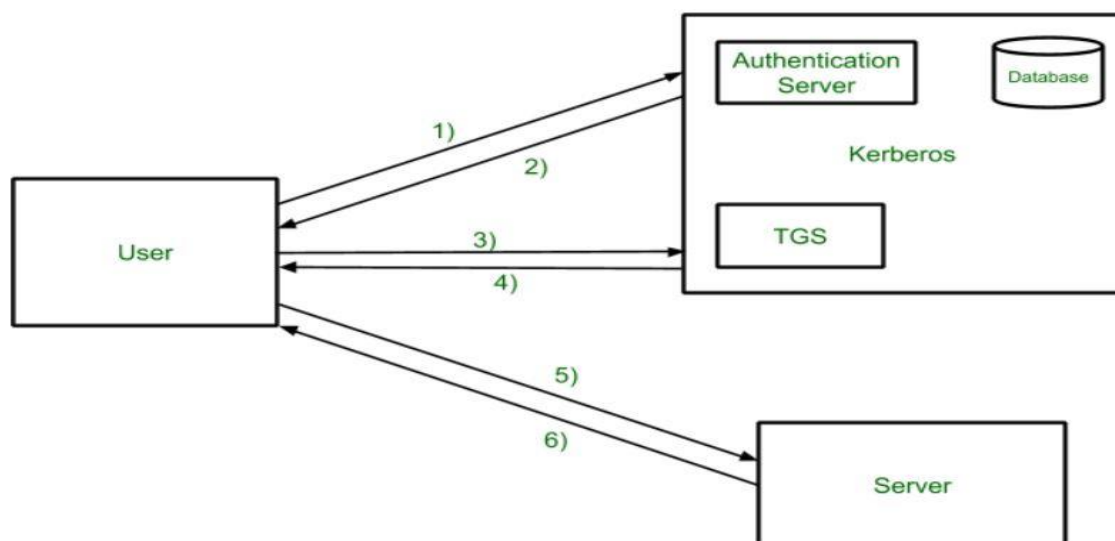
### Kerberos

**Kerberos** provides a centralized authentication server whose function is to authenticate users to servers and servers to users. In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC). Each user and service on the network is a principal.

The main components of Kerberos are:

- **Authentication Server (AS):**  
The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.
- **Database:**  
The Authentication Server verifies the access rights of users in the database.
- **Ticket Granting Server (TGS):**  
The Ticket Granting Server issues the ticket for the Server

### Kerberos Overview:



- **Step-1:**  
User login and request services on the host. Thus user requests for ticket-granting service.
- **Step-2:**  
Authentication Server verifies user's access right using database and then gives ticket-granting-ticket and session key. Results are encrypted using the Password of the user.

- **Step-3:**  
The decryption of the message is done using the password then send the ticket to Ticket Granting Server. The Ticket contains authenticators like user names and network addresses.
- **Step-4:**  
Ticket Granting Server decrypts the ticket sent by User and authenticator verifies the request then creates the ticket for requesting services from the Server.
- **Step-5:**  
The user sends the Ticket and Authenticator to the Server.
- **Step-6:**  
The server verifies the Ticket and authenticators then generate access to the service. After this User can access the services.

#### Kerberos Limitations

- Each network service must be modified individually for use with Kerberos
- It doesn't work well in a timeshare environment
- Secured Kerberos Server
- Requires an always-on Kerberos server
- Stores all passwords are encrypted with a single key
- Assumes workstations are secure
- May result in cascading loss of trust.
- Scalability

#### Is Kerberos Infallible?

No security measure is 100% impregnable, and Kerberos is no exception. Because it's been around for so long, hackers have had the ability over the years to find ways around it, typically through forging tickets, repeated attempts at password guessing (brute force/credential stuffing), and the use of malware, to downgrade the encryption.

Despite this, Kerberos remains the best access security protocol available today. The protocol is flexible enough to employ stronger encryption algorithms to combat new threats, and if users employ good password-choice guidelines, you shouldn't have a problem!

#### What is Kerberos Used For?

Although Kerberos can be found everywhere in the digital world, it is commonly used in secure systems that rely on robust authentication and auditing capabilities. Kerberos is used for Posix, Active Directory, NFS, and Samba authentication. It is also an alternative authentication system to SSH, POP, and SMTP.

#### What is User Authentication in Network Security?

Networks can sometimes be the weak links in the modern day computing world. They are among the most vulnerable and easily hijacked section of the entire setup. This is why different typologies and network security protocols put so much emphasis on the ability to recognize any user trying to make a connection. The recognition process doesn't necessarily identify who the user is. It just verifies the validity of the credentials on the user to determine if that user is cleared to use the resources. This, in essence, is the authentication process in network security.

Authentication happens in two levels. A user or human visible level and a machine level. The human-level authentication is a simple login where you provide a net ID and a password to gain access. Machine level authentication is however more complex and involves a predetermined ID and password that only a machine authorized to access the network can know.

This could occur every time the computer or node in question tries to access the network after the user has finished the initial human authentication. The router or server, in this case, must remember that the machine is authorized to access the network and the machine trying to connect needs to provide its identity (IP address or MAC address) and an accompanying secret key to prove its authority to access the network.

Each authentication process comprises of three main factors:

### **Knowledge Factors**

This is a set of information specific to a user. It's the information that will identify you to the network. It could be a personal identification number and a password or a user name and an answer to a secret challenge. The complexity of these combinations changes depending on how delicate the network in question is hence how much the stake holders are willing to do to protect it from unauthorized use.

### **Possession Factors**

[Possession factors address the items a specific user has in possession](#). This could be hardware devices, for instance a specific MAC address on the network interface card, a security token, or a mobile phone that can receive a one time verification pin.

### **Inheritance factors**

Inheritance factors could either be biometric data that is specific to someone using biometric analysis to access a network, or a cookie left in a computer to identify it in future. A good example is when your computer or phone remembers a network and automatically connects to it when it identifies it.

**Authentication** is the process of verifying the identity of a user or information. User authentication is the process of verifying the identity of a user when that user logs in to a computer system.

There are different types of authentication systems which are: –

1. Single-Factor authentication: – This was the first method of security that was developed. On this authentication system, the user has to enter the username and the password to confirm whether that user is logging in or not. Now if the username or password is wrong, then the user will not be allowed to log in or access the system.

Advantage of the Single-Factor Authentication System: –

- It is a very simple to use and straightforward system.
- it is not at all costly.
- The user does not need any huge technical skills.

The disadvantage of the Single-Factor Authentication

- It is not at all password secure. It will depend on the strength of the password entered by the user.
- The protection level in Single-Factor Authentication is much low.

2. Two-factor Authentication: – In this authentication system, the user has to give a username, password, and other information. There are various types of



authentication systems that are used by the user for securing the system. Some of them are: – wireless tokens and virtual tokens. OTP and more.

#### Advantages of the Two-Factor Authentication

- The Two-Factor Authentication System provides better security than the Single-factor Authentication system.
- The productivity and flexibility increase in the two-factor authentication system.
- Two-Factor Authentication prevents the loss of trust.

#### Disadvantages of Two-Factor Authentication

- It is time-consuming.

3. Multi-Factor authentication system,: – In this type of authentication, more than one factor of authentication is needed. This gives better security to the user. Any type of keylogger or phishing attack will not be possible in a Multi-Factor Authentication system. This assures the user, that the information will not get stolen from them.

The advantage of the Multi-Factor Authentication System are: –

- No risk of security.
- No information could get stolen.
- No risk of any key-logger activity.
- No risk of any data getting captured.

The disadvantage of the Multi-Factor Authentication System are: –

- It is time-consuming.
- it can rely on third parties. The main objective of authentication is to allow authorized users to access the computer and to deny access to unauthorized users. Operating Systems generally identify/authenticates users using the following 3 ways: Passwords, Physical identification, and Biometrics. These are explained as following below.

1. **Passwords:** Password verification is the most popular and commonly used authentication technique. A password is a secret text that is supposed to be known only to a user. In a password-based system, each user is assigned a valid username and password by the system administrator. The system stores all usernames and Passwords. When a user logs in, their user name and password are verified by comparing them with the stored login name and password. If the contents are the same then the user is allowed to access the system otherwise it is rejected.
2. **Physical Identification:** This technique includes machine-readable badges(symbols), cards, or smart cards. In some companies, badges are required for employees to gain access to the organization's gate. In many systems, identification is combined with the use of a password i.e the user must insert the card and then supply his /her password. This kind of authentication is commonly used with ATMs. Smart cards can enhance this scheme by keeping the user password within the card itself. This allows authentication without the storage of passwords in the computer system. The loss of such a card can be dangerous.

3. **Biometrics:** This method of authentication is based on the unique biological characteristics of each user such as fingerprints, voice or face recognition, signatures, and eyes.
4. A scanner or other devices to gather the necessary data about the user.
5. Software to convert the data into a form that can be compared and stored.
6. A database that stores information for all authorized users.
7. **Facial Characteristics** – Humans are differentiated on the basis of facial characteristics such as eyes, nose, lips, eyebrows, and chin shape.
8. **Fingerprints** – Fingerprints are believed to be unique across the entire human population.
9. **Hand Geometry** – Hand geometry systems identify features of the hand that includes the shape, length, and width of fingers.
10. **Retinal pattern** – It is concerned with the detailed structure of the eye.
11. **Signature** – Every individual has a unique style of handwriting, and this feature is reflected in the signatures of a person.
12. **Voice** – This method records the frequency pattern of the voice of an individual speaker.

## Why Is User Authentication Important?

There is no organization, system, network, website, or server in today's modern world that does not require some form of authentication. If they are not, they are putting themselves at risk of attacks that could result in the misappropriation of their resources and sensitive data at the very least. A single blunder may expose your organization's data to cybercriminals, as they are always prepared with a variety of cyber weaponry, such as (Phishing, Data breaches, spoofing, etc). When your authentication system isn't up to par, they can quickly get access and steal information. A few of the most recent major attacks will lead you to the conclusion that, whether you are a little business or a large corporation, authentication using the finest security techniques is a must to stay stable in this technological environment.

## Different types of Authentication :

When it comes to authentication and security, there is a vast ocean of different authentication options to choose from. Before adopting or choosing any of the authentication methods for your Organization's employees or end-users, you should be aware of a few key factors that will help you choose the most appropriate authentication technique for you:

1. Security capability of that Authentication Method
2. Usability interface

Let's take a closer look at the many sorts of authentication techniques available:



### **1. Password Based Login :**

The most commonly utilized regular login authentication system that you will employ on a daily basis while utilizing an online service is password-based login. You need to input a combination of your username/mobile number and a password when using the Password-Based Authentication technique. The individual is authorized only when both of these elements have been verified. However, because today's customers use multiple online services (apps and websites), it's tough to keep track of all of their usernames and passwords. As a result of this, end-users engage in unethical behaviors such as forgetting passwords, using the same password for several services, and so on. Cybercriminals enter at this point and begin actions such as phishing, data breaches, and so on. That is the fundamental reason why standard password-based authentication is losing favor and more organizations are turning to advanced additional security authentication factors.

### **2. Multi-Factor Authentication:**

Multi-Factor Authentication (MFA) is an authentication method in which an individual must pass multiple factors in order to gain access to a service or network. It's an extra layer of security on top of the standard password-based login. Individuals must also submit a second factor in the form of a one-time code that they will receive through phone or email in addition to their Username and Password.

You may quickly configure several Multi-Factor Authentication (MFA) methods to give an extra layer of security to your resources. OTP/TOTP via SMS, OTP/TOTP over Email, Push notification, Hardware Token, and Mobile Authenticator are all examples of MFA methods (Google, Microsoft, Authy, etc). You can choose any of the MFA techniques and implement them for organizational security based on your needs and requirements. After traditional password-based login, Multi-Factor Authentication is the most trusted authentication mechanism. For improved security, password-based traditional authentication and Multi-Factor Authentication methods are usually used simultaneously.

### **3. Biometric Authentication:**

Individual physical attributes such as fingerprints, palms, retinas, voice, face, and voice recognition are used in biometric authentication. Biometric authentication works in the following way: first, the physical characteristics of individuals are saved in a database. Individuals' physical features are checked against the data contained in the database whenever a user wants to access any device or physically enter any premises (Organization, School, Colleges, Workplace). Biometric authentication technology is mostly employed by private organizations, airports, and border crossing points where security is a top priority. Because of its capacity to create a high level of security and a user-friendly frictionless flow, biometrics is one of the most often used security technologies. Among the most common biometric authentication methods are:

**1.Fingerprint:** To enable access, fingerprint authentication matches the unique pattern of an individual's print. In some advanced Fingerprint authentication systems, the vascular structure of the finger is also sensed. Because it is one of the most user-friendly and accurate biometric systems, fingerprint authentication is

currently the most common biometric technology for ordinary customers. Biometrics' popularity can be due to the fact that you use your mobile phones with fingerprints on a regular basis, as well as companies or institutions that use Fingerprint authentication.

**2.Retina & Iris :** Scanners shine a strong light into the eye and look for distinctive patterns in the colourful ring around the pupil of the eye in this biometric. After that, the scanned pattern is compared to data recorded in a database. When a person wears spectacles or contact lenses, eye-based authentication can be inaccurate.

**3. Facial:** In facial authentication, multiple aspects of an individual's face are scanned while they try to get access to a certain resource. When comparing faces from different angles or persons that look similar, such as family members, face recognition results can be inconsistent.

**4. Voice Recognition:** Your voice tone is stored with a standardized secret code in the same way that the above-mentioned approach does. A check occurs because you must speak off each time you want access.

#### **4. Certificate-based authentication:**

Certificate-based authentication identifies people, servers, workstations, and devices by using an electronic digital identity. In our daily lives, a digital certificate functions similarly to a driver's licence or a passport. A certificate is made up of a user's digital identity, which contains a public key and a certification authority's digital signature. This certificate verifies that the public key and the person who issued the certificate are both the same person. When a user attempts to log in to a server, they must first present their digital certificate. The server checks the digital certificate's identity and credibility by confirming that the user has a correctly associated private key with the certificate using cryptography.

#### **5. Token-Based Authentication:**

Token-Based Authentication allows users to enter their credentials only once and obtain a one-of-a-kind encrypted string exchange in return. After that, you won't have to input your credentials every time you want to log in or acquire access. The digital token ensures that you have already been granted access. Most use cases, such as Restful APIs that are accessed by many frameworks and clients, require token-based authentication.

*How will miniOrange advance Authentication technologies will help you out?*

As we get to know more about different types of Authentication, we will move forward with some advanced Authentication Solutions that miniOrange provides with **"All in one Approach"**. It will help you to Manage Identities and access both with advanced security features.

Single Sign-On Authentication (SSO) :

Single Sign-On is a subset of basic username-password-based Authentication. Going with SSO authentication will provide advanced security and multiple features with frictionless experience to your end-users. Single Sign-On as the name depicts allows individuals to enter their username and password once and

get access to all configured applications. Simply stating you will have the provision to configure “N no apps” with miniOrange and you can set a single password for all these apps. With this, you don't need to remember multiple passwords for different applications and you just need to login once and you will automatically get access to all applications. The benefit for this will include – As users need to remember just a single password they will not forget it or write it on any sticky notes type of stuff. Access to multiple applications will become easier which will improve efficiency and boost productivity. From the admin end, they will receive fewer support calls for password resets and login issues.

2nd Factor Authentication (Two-Factor Authentication):

As the name implies, “two-factor authentication” requires an individual to pass two separate authentication procedures in order to gain access to a certain resource. Consider the following scenario: you have a website/app/group of applications and you want to add more protection to it to prevent current cyber assaults such as data breaches, phishing, or the use of keyloggers. With miniOrange, you can configure any app/website built on any platform and enable 2FA for that application. One Time Passwords (OTP through SMS/ Email), Push Notifications, Biometrics, Authenticators (Google Microsoft, Authy), Yubikey and Hardware Token, and more 2FA options are available from miniOrange. According to one of the most recent security surveys, 2FA can prevent 80% of data breaches.

Adaptive Authentication:

Adaptive Authentication is a type of authentication that adapts to the circumstances.” Adaptive Authentication,” a more advanced kind of 2FA/MFA authentication, is introduced. You can authenticate users depending on their “IP, Device, Location, Device, and Time of Access” in this section. If IP and Location-based authentication are enabled, after entering the username and password, Adaptive Authentication will check if the user's IP is the same as the one used by the administrator and whether he is in the location to which he has been assigned. If he does not comply, he will be denied access to the resources. This is one of the most advanced authentication methods used by businesses to ensure their security.

## **Techopedia Explains Key Distribution Center (KDC)**

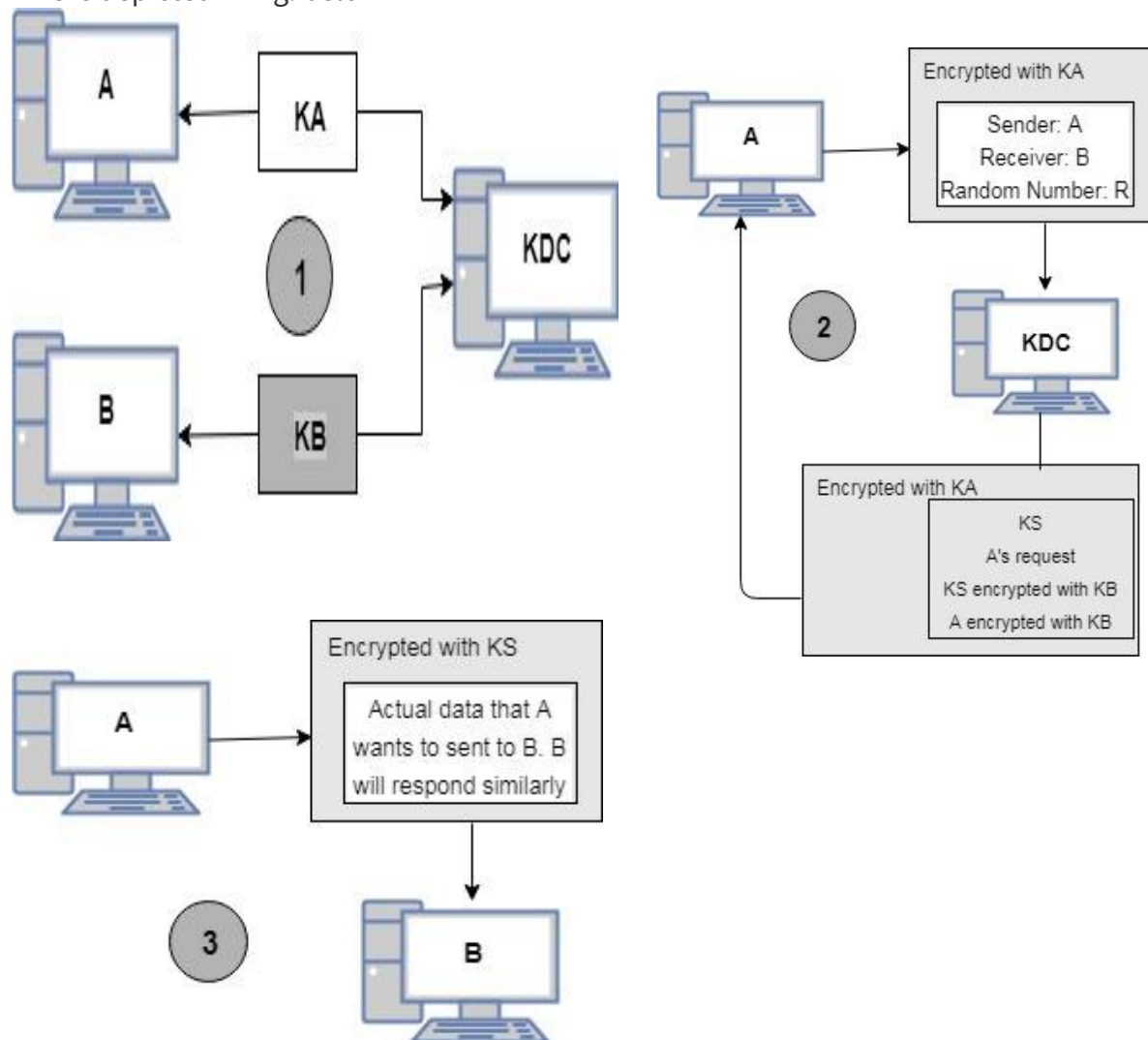
A key distribution center is a form of symmetric encryption that allows the access of two or more systems in a network by generating a unique ticket type key for establishing a secure connection over which data is shared and transferred. KDC is the main server which is consulted before communication takes place. Due to its central infrastructure, KDC is usually employed in smaller networks where the connection requests do not overwhelm the system. KDC is used instead of standard key encryption because the key is generated every time a connection is requested, which minimizes the chances of attack.

**Key Distribution Center (KDC)** is a central authority dealing with keys for individual computers (nodes) in a computer network. It is similar to the concept of the Authentication Server (AS) and Ticket Granting Server (TGS) in Kerberos.

The basic idea is that every node shares a unique secret key with the KDC. Whenever user A wants to communicate securely with user B, the following happens:

1. The background is that A has shared secret key  $K_A$  with KDC. Similarly, B is assumed to share a secret key  $K_B$  with the KDC.
2. A sends a request to KDC encrypted with  $K_A$ , which includes
  - (a) Identities of A and B
  - (b) A random number  $R$ , called a *nonce*
3. KDC responds with a message encrypted with  $K_A$ , containing
  - (a) One-time symmetric key  $K_S$
  - (b) Original request that was sent by A, for verification
  - (c) Plus,  $K_S$  encrypted with  $K_B$  and  $ID$  of A encrypted with  $K_B$
4. A and B can now communicate by using  $K_S$  for encryption.

This is depicted in Fig. below



### What Is a Honeypot?

A honeypot is a cybersecurity mechanism that uses a manufactured attack target to lure cybercriminals away from legitimate targets. They also gather intelligence about the identity, methods and motivations of adversaries.

A honeypot can be modeled after any digital asset, including software applications, servers or the network itself. It is intentionally and purposefully designed to look like a legitimate target, resembling the model in terms of structure, components and content. This is meant to convince the adversary that they have accessed the actual system and encourage them to spend time within this controlled environment.

The honeypot serves as a decoy, distracting cybercriminals from actual targets. It can also serve as a reconnaissance tool, using their intrusion attempts to assess the adversary's techniques, capabilities and sophistication.

The intelligence gathered from honeypots is useful in helping organizations evolve and enhance their cybersecurity strategy in response to real-world threats and identify potential blind spots in the existing architecture, information and network security.

### What Is a Honeynet?

A honeynet is a network of honeypots that is designed to look like a real network, complete with multiple systems, databases, servers, routers and other digital assets. Since the honeynet, or honeypot system, mimics the sprawling nature of a typical network, it tends to engage cybercriminals for a longer period of time.

Given the size of the honeynet, it is also possible to manipulate the environment, luring adversaries deeper into the system in order to gather more intelligence about their capabilities or their identities.

### How Does a Honeypot Work in Cybersecurity?

The basic premise of the honeypot is that it should be designed to look like the network target an organization is trying to defend.

A honeypot trap can be manufactured to look like a payment gateway, which is a popular target for hackers because it contains rich amounts of personal information and transaction details, such as encoded credit card numbers or bank account information. A honeypot or honeynet can also resemble a database, which would lure actors that are interested in gathering intellectual property (IP), trade secrets or other valuable sensitive information. A honeypot may even appear to contain potentially compromising information or photos as a way to entrap adversaries whose goal is to harm the reputation of an individual or engage in ransomware techniques.

Once inside the network, it is possible to track cybercriminals' movements to better understand their methods and motivations. This will help the organization adapt existing security protocols in order to thwart similar attacks on legitimate targets in the future.

To make honeypots more attractive, they often contain deliberate but not necessarily obvious security vulnerabilities. Given the advanced nature of many digital adversaries, it is important for organizations to be strategic about how easily a honeypot can be accessed. An insufficiently secured network is unlikely to trick a sophisticated adversary and may even result in the bad actor providing misinformation or otherwise manipulating the environment to reduce the efficacy of the tool.

## Benefits and Risks of Using a Cybersecurity Honeypot

Honeypots are an important part of a comprehensive cybersecurity strategy. Their main objective is to expose vulnerabilities in the existing system and draw a hacker away from legitimate targets. Assuming the organization can also gather useful intelligence from attackers inside the decoy, honeypots can also help the organization prioritize and focus their cybersecurity efforts based on the techniques being used or the most commonly targeted assets. Additional benefits of a honeypot include:

- Ease of analysis. Honeypot traffic is limited to nefarious actors. As such, the infosec team does not have to separate bad actors from legitimate web traffic – all activity can be considered malicious in the honeypot. This means that the cybersecurity team can spend more time analyzing the behavior of cybercriminals, as opposed to segmenting them from regular users.
- Ongoing evolution. Once deployed, honeypots can deflect a cyberattack and gather information continuously. In this way, it is possible for the cybersecurity team to record what types of attacks are occurring and how they evolve over time. This gives organizations an opportunity to change their security protocols to match the needs of the landscape.
- Internal threat identification. Honeypots can identify both internal and external security threats. While many cybersecurity techniques focus on those risks coming from outside the organization, honeypots can also lure inside actors who are attempting to access the organization's data, IP or other sensitive information.

It is important to remember that honeypots are one component in a comprehensive cybersecurity strategy. Deployed in isolation, the honeypot will not adequately protect the organization against a broad range of threats and risks.

Cyber criminals can also use honeypots just like organizations. If bad actors recognize that the honeypot is a decoy, they can flood the honeypot with intrusion attempts in an effort to draw attention away from real attacks on the legitimate system. Hackers can also deliberately provide misinformation to the honeypot. This allows their identity to remain a mystery while confusing the algorithms and machine-learning models used to analyze activity. It is crucial for an organization to deploy a range of monitoring, detection and remediation tools, as well as preventative measures to protect the organization.

Another honeypot risk occurs when the decoy environment is misconfigured. In this case, advanced adversaries may identify a way to move laterally from the decoy to other parts of the network. Using a honeywall to limit the points of entry and exit for all honeypot traffic is an important aspect of the honeypot design. This is another reason why the organization must enable prevention techniques such as firewalls and cloud-based monitoring tools to deflect attacks and identify potential intrusions quickly.

## Production vs. Research Honeypots

Honeypots can be categorized in many different ways. On the most basic level, honeypots are classified by purpose as either a production honeypot or a research honeypot .

### Production Honeypot

The production honeypot is the most common honeypot type. This decoy is used by businesses to collect information and intelligence about cyberattacks within the production network. This may include IP addresses, intrusion attempt time and dates, traffic volume and other attributes. Production honeypots are relatively simple to design and deploy, but they are less sophisticated than research honeypots in terms of the intelligence produced. They are most commonly used by corporations, private companies and even high-profile individuals, such as celebrities, political figures and business leaders.

### Research Honeypot

A research honeypot is designed to collect information about the specific methods and techniques used by adversaries and what possible vulnerabilities exist within the system in reference to such tactics. Research honeypots are typically more complex than production honeypots. They are often used by government entities, the intelligence community and research organizations to get a better sense of the organization's security risks.

## Security handshake pitfalls

A lot of the existing protocols were designed in an area where eavesdropping was not a concern. The world has changed and so the protocols have to be updated.

- Via a **shared secret**. Alice asks Bob to send her a challenge. She then encrypts the challenge with the shared key between Alice and Bob and sends it back. Alternatively Bob can send an encrypted challenge and Alice has to decrypt it and send it back. The number of connections can be reduced by Alice sending an encrypted timestamp to Bob. All these protocols allow Trudy to impersonate Bob if she can read his database;
- via a **one-way public key**. Alice transforms the challenge with her private key and Bob can verify the validity with her public key. The database of Bob must be protected from modification but disclosure is no longer a problem.

With **mutual authentication** you can validate both participants. The basic protocol requires Alice to ask Bob to send her a challenge. She encrypts the challenge, sends it and Alice creates a second challenge and sends this also to Bob. Bob then encrypts it and sends the result back. In a more simplified version, Alice sends to Bob a challenge, Bob replies with the encrypted challenge and another challenge. Alice then encrypts the second challenge and sends it back. This simplified protocol allows for a **reflection attack** by Trudy and it can also lead to password guessing (Bob requires with the encrypted value that Trudy chooses). Mutual authentication can also be done with private/public keys and instead of using challenges we can also use the timestamp.

Once authentication is done we also need to assure the integrity of the data. Alice and Bob have to agree on a session key either via a shared secret or via private/public keys. You can use sequence numbers to make sure the messages arrive in the right order. You can change



the keys in the middle of a conversation (**key rollover**) to make sure you don't run out of sequence numbers and an attacker waits until the sees the same number reappearing. A **ticket** given by a KDC is an encrypted piece of information that Alice has to hand over to Bob. A **nonce** is a number (sequence number, timestamp, large random number) that is only used once. Note the difference between random (the numbers are picked unpredictably) and pseudorandom (an algorithm where the outcome is determined by its initial state). **Needham-Schroeder** is a classic KDC-arbitrated authentication protocol (Kerberos is based on it with additionally expiration dates in the messages).

1. Alice contacts the KDC. The message contains a nonce that assures Alice that she is talking to the KDC;
2. The KDC returns the nonce, a string "Bob" so the request can not be tampered, a key for Alice and Bob to share and also a ticket (the key and Alice's name encrypted with Bob's key) for Alice to use to Bob. Alice can not read the ticket but Bob can decrypt it.;
3. Alice sends a request to Bob containing a new nonce that is encrypted with the shared key and the ticket;
4. Bob can decrypt the ticket to get the shared key and can so get the nonce;
5. Bob sends the nonce (decremented with one) back to Alice and a also sends a new nonce encrypted with the shared key;
6. Alice returns the decremented version of the decrypted nonce from Bob.

To prevent Trudy from reading recorded messages with a captured (older) key from Alice an extra step is introduced where Alice requests Bob to send a nonce, the nonce is send to the KDC and packaged in the ticket for Bob.

**Otway-Rees** is an improved authentication protocol.

1. Alice sends a clear-text nonce to Bob and another nonce encrypted with the key of Alice;
2. Bob forwards the encrypted nonce to the KDC along with his own encrypted nonce and the clear-text nonce;
3. The KDC replies to Bob the shared key and a message he has to forward to Alice (to assure to Alice that both the KDC and Bob are legitimate);
4. Bob forwards that message to Alice and Alice then replies by sending something encrypted with the shared key (proving that she now knows the key).

It is important that the nonces are unpredictable.

### Authentication Protocol Checklist

- Protect against **eavesdrop**;
- **Impersonation** of one of the correspondents;
- Protect against a situation when the keys are stolen;



- Intercept and **modify** messages.

## Password protocols

A **credential** is something that can be used to prove who you are or prove that you are authorized to do something.

The **Lamport's hash** provides a password protocol that protect against eavesdropping and impersonation and does not use public key cryptography. Alice has a password. Bob remembers the username, an integer (n) that gets decremented with every authentication and n-times the hash of the password. When the integer reaches 1 then Alice has to reset the password. A **salt** can be chosen at password installation time and concatenated to the password before the hashing takes place. Alice can reuse the password with different Bob's as long as different salt is used when setting up the password with the different Bob's.

The **EKE – Encrypted Key Exchange** is a password protocol where

- Alice and Bob share a weak secret (the hash of the password);
- Bob knows the secret because he stores it;
- Alice knows the secret because she computes it.

A Diffie-Hellman exchange happens where the D-H numbers are encrypted with the weak secret and Alice and Bob then do mutual authentication based on the D-H shared secret.

Other strong password protocols were later introduced **SPEKE – Simple Password Exponential Key Exchange**, **SRP – Secure Remote Passwords** and **PDM – Password Derived Moduli**.

## Single Sign On

SSO is an organization access control solution that allows users to authenticate once (typically once per session) and get access to all enterprise resources connected to the SSO system, the solution provides federated access to multiple independent software with one set of credentials . Typically to achieve this magic, multiple techniques are used behind the scenes, depending on what methods each resource supports.

Type of Single Sign On Configurations:

Kerberos Authentication

- Initial sign-on prompts the user for credentials, and gets a [Kerberos ticket-granting ticket](#) (TGT).
- Additional software applications requiring authentication, such as [email clients](#), [wikis](#), and [revision-control](#) systems, use the ticket-granting ticket to acquire service tickets, proving the user's identity to the mailserver / wiki server / etc. without prompting the user to re-enter credentials.

Kerberos is a client-server authentication protocol that allows mutual authentication – both the user and the server verify each other's identity – over non-secure network connections.

The Kerberos protocol uses a symmetric key derived from the user password to securely exchange a session key for the client and server to use. A server component is known as a Ticket Granting Service (TGS) then issues a security token (AKA Ticket-

Granting-Ticket TGT) that can be later used by the client to gain access to different services provided by a Service Server.

[Learn More About Single Sign-On \(SSO\)](#)

### **Smart-card-based Authentication**

A smart card is a secure microcontroller that is typically used for generating, storing and operating on cryptographic keys. Smart card authentication provides users with smart card devices for the purpose of authentication. Users connect their smart card to a host computer. Software on the host computer interacts with the keys material and other secrets stored on the smart card to authenticate the user.

In order for the smart card to operate, a user needs to unlock it with a user-PIN.

Smart cards are considered a very strong form of authentication because cryptographic keys and other secrets stored on the card are very well protected both physically and logically, and are therefore extremely hard to steal.

The added security provided by the smart card comes at the expense of the user experience, as smart cards need to be physically carried around by the user and inserted into the host computer every time they want to authenticate with it. Users are also limited to host devices that have the card interface software installed.

Smart cards are also expensive to administrate, as they require software installation on the host computer and physical distribution to the users.

### **Integrated Windows Authentication (IWA)**

Integrated Windows Authentication uses the security features of Windows clients and servers. Unlike Basic or Digest authentication, initially, it does not prompt users for credentials. The current Windows user information on the client computer is supplied by the web browser through a certificate exchange involving hashing with the Web server. If the authentication exchange initially fails to identify the user, the web browser will request the user for credentials.

Integrated Windows Authentication itself is not a standard or an authentication protocol. When IWA is selected as an option of a program implies underlying security mechanisms should be used in a preferential order. If the Kerberos provider is functional and a [Kerberos ticket](#) can be obtained for the target, and any associated settings permit Kerberos authentication to occur (e.g. Intranet sites settings in Internet Explorer), the Kerberos 5 protocol will be attempted. Otherwise [NTLMSSP](#) authentication is attempted. Similarly, if Kerberos authentication is attempted, yet it fails, then NTLMSSP is attempted. IWA uses SPNEGO to allow initiators and acceptors to negotiate either Kerberos or NTLMSSP. Third party utilities have extended the Integrated Windows Authentication paradigm to UNIX, Linux and Mac systems.

### **Security Assertion Markup Language (SAML)**

Security Assertion Markup Language Security Assertion Markup Language (SAML) is a standard for logging users into applications based on their sessions in another context. This single sign-on (SSO) login standard has significant advantages over logging in using a username/password: No need to type in credentials No need to remember and renew passwords No weak passwords Most organizations already know the identity of users because they are logged in to their Active Directory domain or intranet.

It makes sense to use this information to log users in to other applications, such as web-based applications, and one of the more elegant ways of doing this is by using SAML. SAML is very powerful and flexible, but the specification can be quite a handful. OneLogin's open-source SAML toolkits can help you integrate SAML in hours, instead of months. We've come up with a simple setup that will work for most applications.

Security Assertion Markup Language (SAML) is an open standard that defines a XML-based framework for exchanging authentication and authorization information between an identity provider (IdP) and a service provider (SP), to enable web-based single sign-on (SSO) and identity federation.

Security information is expressed in the form of portable SAML assertions that applications working across security domain boundaries can trust. In a typical scenario, a user requesting access to a service provider is redirected to an identity provider capable of authenticating the user and providing a SAML assertion that allows the service provider to make its access control decisions. The SP and IdP must have a trust relationship established prior to exchanging SAML assertions.

- *WHAT IS SSO?*

---

Single Sign On (SSO) is a solution that allows a user to authenticate once and gain access to all applications/resources supported by the SSO, without having to sign in separately to each application/resource.

---

*WHAT IS AN EXAMPLE OF SINGLE SIGN ON?*

There are many SSO solutions in the market. Active Directory (AD) is an example of a SSO because all domain resources joined to AD can be accessed without the need for additional authentication. SAP, Oracle, IBM and others offer SSO solutions for enterprise use. Okta, OneLogin and others specialize in single sign on for web applications

- *WHAT ARE THE ADVANTAGES OF SINGLE SIGN ON (SSO) ?*

---

To name a few of the many advantages provides an organization

**Access logs** – an SSO portal provides detail reporting on who accessed what

**Session time** – by eliminating credential reauthentication users spend less time on the authentication process leading to improved productivity.

**Centralized database** – one database that includes logs for authentication and authorization to support compliance and administration.

**Less credentials means less chance of phishing** – phishing emails and social engineering are nearly impossible

**Reduce help desk costs** – reducing the amount of credentials (passwords) translates to less help desk calls which are estimated at 20 – 50% of all help desk calls.

---

- ***WHAT ARE THE DISADVANTAGES OF SINGLE SIGN ON (SSO) ?***

- The main disadvantage of SSO is in its use of one set of credentials, if those credentials are not protected correctly and are stolen the thief has access to your entire kingdom.

Companies should always use a 2nd factor to login to SSO (at the very less), companies who take security seriously will use multi factor authentication (MFA).

- The second less talked about disadvantage to SSO is the fact that while it provides single sign on it does not provide single sign off, the logoff process varies from application to application and depends on the settings of the application, user sessions usually stay active long after the user has completed his/hers use which can easily lead to session hijacking.

## What is Single Sign-On?

Single sign-on (SSO) is an authentication method that enables users to securely authenticate with multiple applications and websites by using just one set of credentials.

## How Does SSO Work?

SSO works based upon a trust relationship set up between an application, known as the service provider, and an identity provider, like OneLogin. This trust relationship is often based upon a certificate that is exchanged between the identity provider and the service provider. This certificate can be used to sign identity information that is being sent from the identity provider to the service provider so that the service provider knows it is coming from a trusted source. In SSO, this identity data takes the form of tokens which contain identifying bits of information about the user like a user's email address or a username.

The login flow usually looks like this:

1. A user browses to the application or website they want access to, aka, the Service Provider.
2. The Service Provider sends a token that contains some information about the user, like their email address, to the SSO system, aka, the Identity Provider, as part of a request to authenticate the user.
3. The Identity Provider first checks to see whether the user has already been authenticated, in which case it will grant the user access to the Service Provider application and skip to step 5.
4. If the user hasn't logged in, they will be prompted to do so by providing the credentials required by the Identity Provider. This could simply be a username and password or it might include some other form of authentication like a One-Time Password (OTP).
5. Once the Identity Provider validates the credentials provided, it will send a token back to the Service Provider confirming a successful authentication.
6. This token is passed through the user's browser to the Service Provider.
7. The token that is received by the Service Provider is validated according to the trust relationship that was set up between the Service Provider and the Identity Provider during the initial configuration.
8. The user is granted access to the Service Provider.

When the user tries to access a different website, the new website would have to have a similar trust relationship configured with the SSO solution and the authentication flow would follow the same steps.

## What is an SSO Token?

An SSO token is a collection of data or information that is passed from one system to another during the SSO process. The data can simply be a user's email address and information about which system is sending the token. Tokens must be digitally signed for the token receiver to verify that the token is coming from a trusted source. The certificate that is used for this digital signature is exchanged during the initial configuration process.