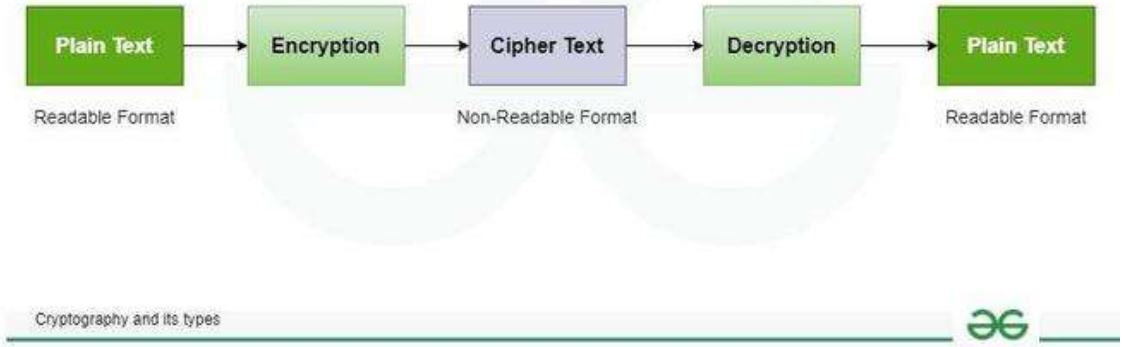
Cryptography and its Types

Cryptography is a technique of securing information and communications using codes to ensure confidentiality, integrity and authentication. Thus, preventing unauthorized access to information. The prefix "crypt" means "hidden" and the suffix "graphy" means "writing". In Cryptography, the techniques that are used to protect information are obtained from mathematical concepts and a set of rule-based calculations known as algorithms to convert messages in ways that make it hard to decode them.



These algorithms are used for cryptographic key generation, digital signing, and verification to protect data privacy, web browsing on the internet and to protect confidential transactions such as credit card and debit card transactions.

Features Of Cryptography

The features of cryptography that makes it a popular choice in various applications could be listed down as

Confidentiality: Information can only be accessed by the person for whom it is intended and no other person except him can access it.

2. **Non-repudiation:** The creator/sender of information cannot deny his intention to send information at a later stage.
3. **Integrity:** Information cannot be modified in storage or transition between sender and intended receiver without any addition to information being detected.
4. **Adaptability:** Cryptography continuously evolves to stay ahead of security threats and technological advancements.
5. **Interoperability:** Cryptography allows for secure communication between different systems and platforms.
6. **Authentication:** The identities of the sender and receiver are confirmed. As well destination/origin of the information is confirmed.

Working of Cryptography

As we all know that cryptography technique is use to convert plain text into ciphertext. This technique is done by cryptographic key. Basically cryptographic key is a string of characters which is used to encrypts the data and decrypt the data.

Here,

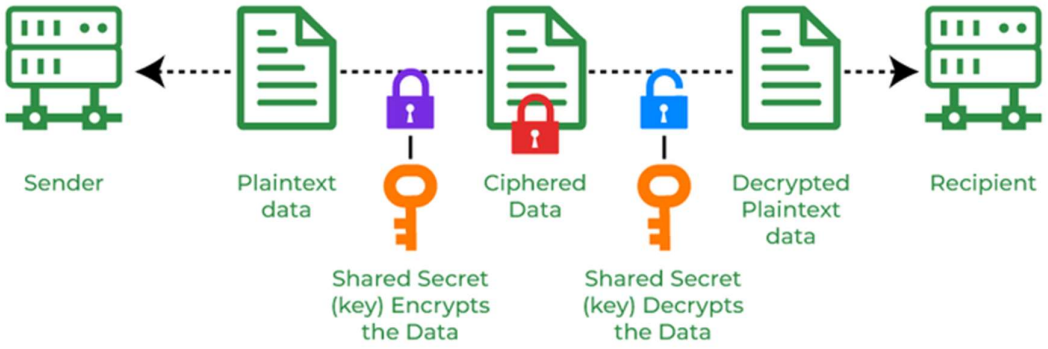
"Hello" is a plaintext and convert into ciphertext "jknnq" with the help of cryptographic key and then decrypt into "Hello".

Types Of Cryptography

There are three types of cryptography, namely Symmetric Key Cryptography, Asymmetric Key Cryptography and Hash functions, here's a detailed explanation below:

1. Symmetric Key Cryptography

Symmetric Key Cryptography is an encryption system where the sender and receiver of a message use a single common key to encrypt and decrypt messages. Symmetric Key cryptography is faster and simpler but the problem is that the sender and receiver have to somehow exchange keys securely. The most popular symmetric key cryptography systems are Data Encryption Systems (DES) and Advanced Encryption Systems (AES) .



Symmetric Key Cryptography

Example:

1. Data Encryption Standard (DES)

DES (Data encryption standard) is an older encryption algorithm that is used to convert 64-bit plaintext data into 48-bit encrypted ciphertext. It uses symmetric keys (which means same key for encryption and decryption). It is kind of old by today's standard but can be used as a basic building block for learning newer encryption algorithms.

Example of DES Encryption and Decryption tool:

2. AES (Advanced Encryption Standard):

AES s a popular encryption algorithm which uses the same key for encryption and decryption It is a symmetric block cipher algorithm with block size of 128 bits, 192 bits or 256 bits. AES algorithm is widely regarded as the replacement of DES (Data encryption standard) algorithm.

Example for Encryption and Decryption of AES tool.

Hash Functions

Hash functions do not require a key. Instead, they use mathematical algorithms to convert messages of any arbitrary length into a fixed-length output, known as a hash value or digest. Hash functions are designed to be one-way, meaning the original input cannot be derived from the output. Given Below, Some of the most widely used hash functions include:

SHA-1:

SHA-1 stands for Secure Hash Algorithm 1. It is a cryptographic hash function developed by the National Security Agency (NSA) and published by NIST in 1995. It was designed to ensure data integrity by converting any input into a fixed-size 160-bit (20-byte) hash valueSHA-1

Example of SHA-1 Hash Generator

Example

SHA-256:

The SHA-256 algorithm belongs to the family of the SHA 2 algorithms. It generates a fixed 256-bit (32-byte) hash value from input data of any length. It is widely used in various security applications, including data integrity verification, digital signatures, blockchain technology, and password hashing, due to its strength and resistance to known cryptographic attacks.

SHA-256

Example:

Data : geeksforgeeks

SHA256 : f8d59362da74ffe833332dc20508f12de6da6a9298c98b3b42873e7298fced78

MD5:

MD5 produces a 128-bit hash value (32-character hexadecimal number) from input data of any size. Originally designed for data integrity verification, MD5 was widely used for checksums, digital signatures, and file verification. It is developed by Ronald Rivest in 1991.

MD5

MD6:

MD6 is a cryptographic hash function designed by Ronald Rivest and his team in 2008 as a successor to the MD5 and MD4 algorithms. It was created to be highly secure and suitable for modern computing systems, including multi-core processors.

What is a Symmetric Encryption?

Last Updated : 07 Aug, 2025

-
-
-

When the plain text is encrypted and decrypted using the same key, it is known as symmetric encryption. It is also known as "shared-key" or "private-key" encryption. It ensures confidentiality by ensuring only authorized parties with the key can access the original data.

The key is a piece of a shared secret between the two parties involved, hence it is '**shared-key**' and is kept secret, hence the name '**private-key**' is justified.

How does Symmetric Encryption Work?

From key generation to decryption, multiple steps are involved when symmetric encryption is applied. These are the steps involved in sharing a message securely over the network using the symmetric encryption technique.

1. Key Generation:

The first step involves selecting a private key. A secure key is generated using algorithms like PBKDF2 (Password-Based Key Derivation Function 2) or hardware random number generators. This key must be securely shared or transferred over the network for future use.

Example: A 256-bit AES key: 3A7F2B4E... (32-byte hexadecimal string).

2. Encryption:

In this step, the original message (plain text) is transformed into unreadable text (ciphertext) ,and the plaintext is processed in blocks or streams using an encryption algorithm and the secret key.

Example: AES-256 in CBC (Cipher Block Chaining) mode encrypts a 128-bit block of plaintext with the key and an initialization vector (IV) to produce ciphertext.

3. Transfer of Cipher text:

The encrypted message (ciphertext) is then sent over the network. Even if intercepted, it remains unreadable to the attacker unless they have access to the shared secret key and the algorithm used for encryption.

4. Decryption:

In the final step, the recipient uses the same secret key and a reverse encryption algorithm to convert the cipher text back into the original message (plain text).

Challenges of Symmetric Encryption

While symmetric encryption is widely valued for its speed and efficiency, it also comes with several challenges regarding the storing and sharing of the key, here are the following causes:

- **Key Sharing Security**: The primary challenge lies in securely sharing the secret key. If the key is compromised, the entire communication becomes vulnerable. So use asymmetric encryption (e.g., RSA) for initial key exchange (hybrid systems)
- **Key Storage**: Safely storing the secret key is another significant concern, as unauthorized access to the key can compromise the security of the encrypted data. So use the Hardware Security Modules (HSMs) for tamper-resistant storage and for cloud use Cloud-based Key Management Services (KMS) like AWS KMS or Azure Key Vault.
- **Scalability**: As the number of users grows, the complexity of securely managing and sharing secret keys increases exponentially, making it more difficult to maintain a robust security framework, so implement hierarchical key structures or automated rotation policies.

Application of Symmetric Encryption

Due to its speed and efficiency, Symmetric encryption is widely used for securing communications. Some of the most common use-cases of symmetric encryption includes:

- **File and Disk Encryption**: Symmetric encryption is the preferred choice for securing files, databases, and entire drives due to its robust performance and simplicity.

- **Bulk Data Encryption**: For encrypting large volumes of data, symmetric encryption is the go-to method because of its faster processing time compared to asymmetric encryption.
- **Hybrid Algorithms**: While asymmetric encryption is crucial for securing keys and verifying identities, it is not ideal for data encryption. Hybrid systems combine the strengths of both, using asymmetric encryption for key exchange and symmetric encryption for the actual data encryption.

Real-life Examples

Here are some most uses real-life examples of SymmetricEncryption:

1. Wi-fi Password (WPA2)

When you connect to a Wi-Fi network, both your device and the Wi-Fi router use the same password to authenticate each other. This password is not directly used to encrypt the data but is used during a handshake process (like the 4-way handshake in WPA2) to generate a shared secret encryption key. Once this key is established, it is used to securely encrypt and decrypt all the data transmitted between your device and the router.

2. BitLocker

When BitLocker is enabled, it generates a single encryption key (called the Full Volume Encryption Key, or FVEK) to encrypt and decrypt all the data on the disk. This key is stored securely and is protected by a trusted platform module (TPM), a PIN, USB key, or password, depending on the configuration. Since the same key is used for both encryption and decryption, this makes BitLocker an example of symmetric encryption.

Symmetric Encryption Algorithms

Symmetric encryption has various types, depending on their features, strengths and effectiveness. Some of the most popular symmetric encryption algorithms includes:

Algorithm	Description	Key Features
AES (Advanced Encryption Standard)	Widely adopted symmetric encryption standard endorsed by NIST for national and industrial use.	Available in 128-bit, 192-bit, and 256-bit key sizes. High performance and security.
DES (Data Encryption Standard)	Formerly popular, now obsolete due to vulnerability to brute-force attacks.	56-bit key size. Superseded by more secure alternatives like AES and 3DES.
Triple DES (3DES)	Improved version of DES, applies DES three times to each data block.	Stronger than DES, but slower and less efficient than AES.
Blowfish	Block cipher designed as an alternative to DES, known for speed and effectiveness in many applications.	64-bit block size. Flexible key lengths up to 448 bits.
Twofish	Successor to Blowfish and finalist in AES competition. Offers robust security and flexibility.	128-bit block size. Key sizes up to 256 bits.

Cryptography - Traditional Ciphers

Traditional ciphers encode messages in a way that only those with the secret key can decode it. Throughout history, it has been used for private communication.

These ciphers were used extensively before the advent of modern computer encryption techniques. Although relatively simple, when used properly, it can also be effective and provide interesting insights into the history of encryption.

Here are a few common types −

- Caesar Cipher
- Vigenere Cipher
- Simple Substitution Cipher
- Monoalphabetic and Polyalphabetic Cipher
- Transposition Cipher
- Playfair Cipher

Earlier cryptographic systems

Before proceeding further, you should know some facts about cryptographic systems −

- These systems are all built using symmetric key encryption.
- These systems simply offer information confidentiality as a security feature.
- In comparison with modern digital systems that handle data as binary numbers, previous systems used alphabets as the basic building block.

These early cryptography schemes are also called Ciphers. In general, a cipher is simply the steps (algorithms) that can be used to perform encryption, and the corresponding decryption.

Now let us discuss the above types of Tranditional Ciphers one by one.

Caesar Cipher

Caesar cipher is a type of substitution cipher in which every letter in the given plaintext is shifted with a certain number of places down or up the alphabet. The shift number is called the key. It is create by Julius Caesar. It is believed that this cipher is to communicate secretly with his generals.

For this type of method both the sender and the receiver should agree on a 'secret shift key' for shifting the alphabet of the given plaintext. The number which can be between 0 and 25 is the key of encryption.

This techniques sometimes used to describe the Shift Cipher when the 'shift of three' is used.

For example - If plaintext is shift of 4:

- 'A' will become 'E'

- 'B' will become 'F'
- 'C' will become 'G'
- and so on...

The Caesar cipher is a very simple and easy to understand encryption method. But it is also very easy to break as there are only 25 possible keys which is easy to guess using brute force.

Vigenere Cipher

Vigenere is also a traditional cipher algorithm which uses a text string (Let us say a word) as a key. Then it is used for shifting a number of shifts on the given plaintext.

For example − let us say the key is 'tutor'. In this each alphabet of the key is changed to its respective numeric value: In our case,

t → 20, u → 21, t → 20, o → 15, and r → 18.

Thus, the key is: 20 21 20 15 18.

Vigenere Cipher Steps

First the sender and the receiver decide a key. Let us say the key is 'tutor'. So '20 21 20 15 18' is the Numeric representation of this key.

Now the sender will encrypt the message. And the message is 'This is ethical hacker'. Then it will be arranged as numeric key as follows −

```
T  h  i  s  i  s  e  t  h  i  c  a  l  h  a  c  k  e  r
20 21 20 15 18 20 21 20 15 18 20 21 20 15 18 20 21 20 15
```

Now we will each plaintext letter by the number written in below the letter:

```
T  h  i  s  i  s  e  t  h  i  c  a  l  h  a  c  k  e  r
20 21 20 15 18 20 21 20 15 18 20 21 20 15 18 20 21 20 15
n  c  c  h  a  m  z  n  x  a  w  v  f  w  s  w  f  y  g
```

In the above image, every letter of the plaintext is shifted by a different amount and that amount is found by the key. The key should be less than or equal to the size of the message.

In the decryption process, the receiver will use the same key and shifts received cipher text in opposite order to get the plaintext.

```
n  c  c  h  a  m  z  n  x  a  w  v  f  w  s  w  f  y  g
20 21 20 15 18 20 21 20 15 18 20 21 20 15 18 20 21 20 15
T  h  i  s  i  s  e  t  h  i  c  a  l  h  a  c  k  e  r
```

The Vigenere Cipher is designed to tweak the standard Caesar cipher to reduce the efficiency of cryptanalysis on the ciphertext and increase the robustness of the cryptosystem It is much more secure than a regular Caesar cipher.

Throughout history, it was regularly used to protect important political and military information. It is described as an unbreakable cipher due to the difficulty of dividing the secret.

The Vigenere cipher − has two main points

The length of the keyword is the same as the plaintect message. This statement is called varnam cipher. This is more secure than the standard Vigenere cipher. The Vigenere cipher would be a cryptosystem with complete privacy, called a One-time pad.

Simple Substitution Cipher

Simple substitution cipher is a form of encryption in which the letters in Plaintext is replaced by another character in cipher text according to a fixed order. In other words it is a method of encoding with any letter and characters are mapped to other characters.

For example, in Simple Substitution Cipher −

- 'A' can be replaced by 'X'
- 'B' can be replaced by 'K'
- 'C' can be replaced by 'Q'
- So on... until each character in the alphabet is replaced by another one.

The key to decrypting a message is knowing the substitution model used. These ciphers are relatively easy to use, but can be break by frequency analysis which requires analyzing the frequency of the characters in the cipher text and removing the mapping.

The simple alternative cipher is a vast improvement over the Caesar cipher. The number of possible keys is enormous (26!) and even modern computer systems do not yet have the ability to comfortably perform brute force attacks to crack the system However, a simple alternative cipher has a simple and common structure system flaws, say choose obvious changes.

Monoalphabetic and Polyalphabetic Cipher

A Monoalphabetic cipher is a type of substitution cipher in which fixed characters are always replaced in the ciphertext which means that every character in the plaintext is always replaced by the same corresponding character in the ciphertext. The replacement remains constant throughout the encryption process.

For example, if 'A' is replaced by 'E' in the encryption process, any occurrence of 'A' in plain text will be replaced by 'E' in ciphertext and if 'B' is replaced by 'F'. replace 'any B ' in plaintext with 'F' in ciphertext.

A Polyalphabetic Cipher is a type of substitution cipher in which plaintext characters are replaced by ciphertext characters varying with the encryption process. This means that characters in the plaintext can be replaced by characters in the ciphertext, depending on the encryption key and the position of the character in the plaintext.

The most popular Polyalphabetic Cipher is the Vigenaire cipher, where the encryption key determines where the letters will be located in the ciphertext. Unlike single-letter ciphers, Polyalphabetic Cipher are more secure because they introduce changes to the encryption scheme, making frequency analysis and other attacks more difficult.

Transposition Cipher

This is a new type of cipher in which the sequence of letters is rearranged in plain text to create ciphertext. They are not replaced by actual plaintext.

An example is a simple column substitution cipher in which simple increments are written in a specific alphabet width. The ciphertext is then read directly as indicated.

For example, the plain text is "The tajmahal is in agra" and the randomly chosen hidden key is Four. We format this text directly in a table with a number of columns including key value. The resulting text is shown below.

T	h	e	t
a	j	m	a
h	a	l	i
s	i	n	a
g	r	a	

So now we can get the ciphertext just reading columns from first to last by vertically downward. And the ciphertext will be 'Tahsghjairemlnataia'.

Playfair Cipher

The Playfair cipher uses a 5x5 grid of letters unless it is double and usually 'J' for encryption. Messages are encrypted by pairing letters and using rules: if on the same letter, turn right; If it is in one column, turn down; If there are rows and columns, make a triangle and replace them with opposite corners. Decryption follows the opposite rule. Playfair encrypts character pairs, making it more secure than single-character ciphers, but still vulnerable to an attack.

Cryptography - Need for Encryption

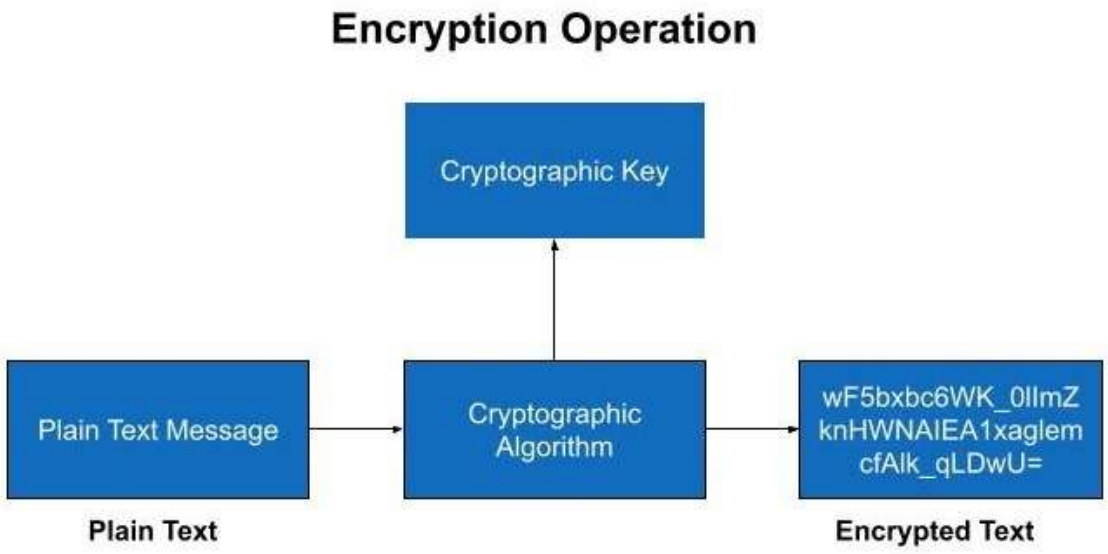
A large volume of private information is sent over the Internet, including passwords entered into login screens, emails carrying personal information, and tax documents uploaded to servers.

Private data is transmitted over the Internet in packets with the help of the same pathways as public data. Unfortunately, malicious individuals have found methods to intercept this data during its journey across the Internet.

We will explore the importance of encryption and its role in safeguarding sensitive information.

What is Encryption?

Encryption is the process of coding our data to hide its true meaning. Cryptography is the study of data encryption and decryption. Encryption has been safeguarding sensitive data for a long time and was once primarily used by governments and military forces. Encryption secures data both when it is being transmitted across networks and devices and when it is stored on computers and other storage devices. Encryption helps secure sensitive information such as passwords, financial data, and personal details from unauthorized access and manipulation.



What data should be Encrypted?

Lack of proper data encryption can lead to complex issues. When confidential information gets leaked, it can severely damage your business and data in various ways, such as causing economic disasters, harming your brand, and resulting in a loss of customer trust.

You need to back up all encrypted data. Data encryption should cover both at-rest and in-transit data for complete protection. Below are some categories of data that require encryption −

- Credit Card Details
- Account Passwords
- Social Numbers
- Mobile Phone Numbers
- House Addresses
- Bank Account Details
- Patient Health Details

- And many other confidential details

The Core Principles of Cryptography and Encryption

Encryption is crucial for the protection of various IT assets and private identified data. With this, encryption meets four important principles −

Confidentiality

Encryption makes sure that only authorized parties can access and understand the encrypted data. Encodes the data to prevent it from being understood if it is intercepted by a third party. By changing the plaintext into ciphertext using cryptographic algorithms to secure the private data. For example: It is like having a secret language that only you and your friends understand.

Authentication

It is the process of making the piece of data being claimed by the user belong to it. It verifies the origin of the data that has been encrypted. Digital certificates and cryptographic protocols authenticate users and devices by validating their credentials and confirming their identity before establishing secure communication channels. Authentication prevents impersonation attacks and unauthorized access by ensuring that only trusted entities can decrypt and access encrypted data.

Integrity

Encryption helps maintain the integrity of data by finding any unauthorized changes or modifications. Using techniques like digital signatures and message authentication codes (MACs), encryption ensures that the encrypted data is unchanged during transmission or storage. So we can say, it is like putting a special seal on a letter. If someone tries to open it before it gets to you, you'll know because the seal will be broken.

Nonrepudiation

Nonrepudiation prevents senders from denying they sent the encrypted data. Encryption provides non-repudiation by making sure that parties cannot deny their actions or transactions. For example - It is like getting a receipt when you buy something. If you try to say you didn't buy it later, the receipt proves you did.

Benefits of Cryptography and Encryption for Security

Protecting the confidentiality of digital data kept on computer systems or sent over the internet or other computer networks is the main goal of encryption. It is used to protect many types of data, like private and confidential company information, and military and public records.

- Encrypting data helps organizations reduce their risk of serious penalties, long legal proceedings, revenue loss, and a damaged reputation.
- Many firms use encryption not only to protect their data but also to meet rules that require sensitive data to be protected.
- In rare cases where unauthorized parties or hackers manage to access the data, encryption ensures they are unable to decipher it.

For example, sellers must encrypt consumer credit card data when it is at rest and when it is being sent over public networks, in keeping with the credit Card Industry Data Security Standard.

Challenges in Encryption

Encryption presents several challenges, particularly in managing encryption keys, which are crucial for accessing encrypted data. The process of key management can be complex and poses risks if not handled properly.

- With the help of encryption, the data is safe from attackers, but in some cases, it can stop the owners of the data from being able to access their information.
- If encryption keys are lost or destroyed, owners might never access their data again.
- Hackers might target encryption keys instead of the data itself, making it easy for them to decode information.
- Managing encryption keys is hard because they have to be stored securely, but attackers know where to look.
- There are best practices for managing encryption keys, but they make backup and restore processes complicated.
- Retrieving and transferring keys to a new server can delay recovering data.
- Just setting up a key management system is not enough; managers need a plan to protect it.
- This includes backing up the key management system separately and arranging backups so keys are easy to get in a disaster.

Cryptosystems - Overview

A cryptosystem uses cryptographic methods and the support structures around them to secure information. It's also known as a cipher system.

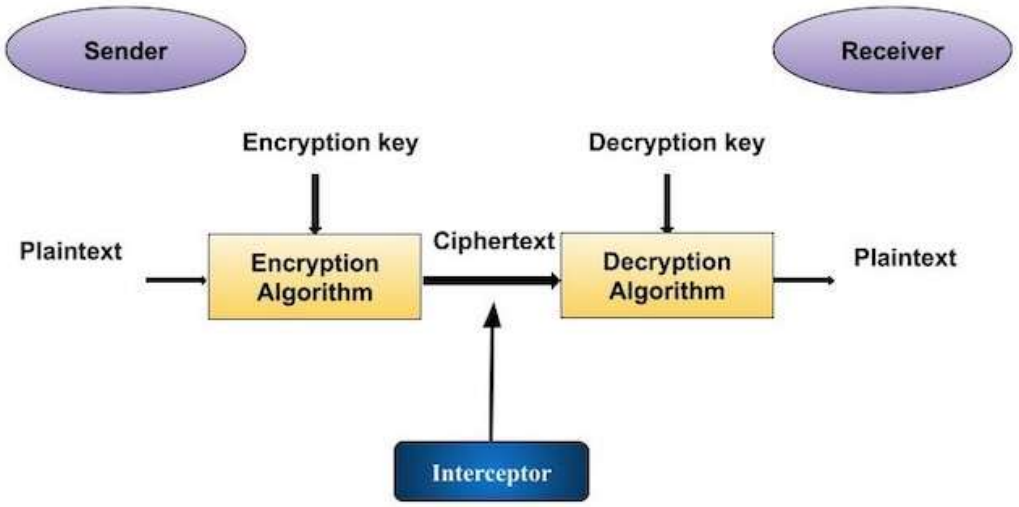
Cryptosystem uses algorithms to convert plain text into ciphertext to securely encode or decode messages. The term cryptosystem refers to a computer system that uses cryptography, codes to protect information and communications so only the intended can read and process it.

Cryptosystems use algorithms for key creation, encryption, and decoding to keep data secure. Cryptographic keys, bit strings used by algorithms, transform plain text into coded text and vice versa.

The key and variable data are provided as input to an algorithm for this operation. The algorithm's security relies on secure keys.

Cryptosystems use private data like credit cards securely online. Secure email uses signatures, hashes, and key management.

We will look at a basic cryptosystem model. This model keeps transmitted information confidential. You can see this simple model in the picture below −



The figure above shows a sender transmitting a sensitive piece of information to a receiver in such a way that the data cannot be extracted by any authenticated or third party in the communication channel.

The goal of this simple crypto scheme is that, at the end of the process, only the sender and receiver will know the private information.

Cryptographic Keys

The key plays an important role in the variable information submitted as input to a cryptographic algorithm to perform an encryption or decryption process. The security of the cryptographic system relies heavily on how secure the keys are that are used. If an unauthorized party were able to access the keys, they could potentially decrypt encrypted messages, pretending to be someone else, or encrypt misleading messages to pose as another individual. It is vital that keys are kept private and are generated, distributed, and stored securely to maintain the integrity of the cryptographic system. The strength of the encryption also depends on the length and complexity of the keys.

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

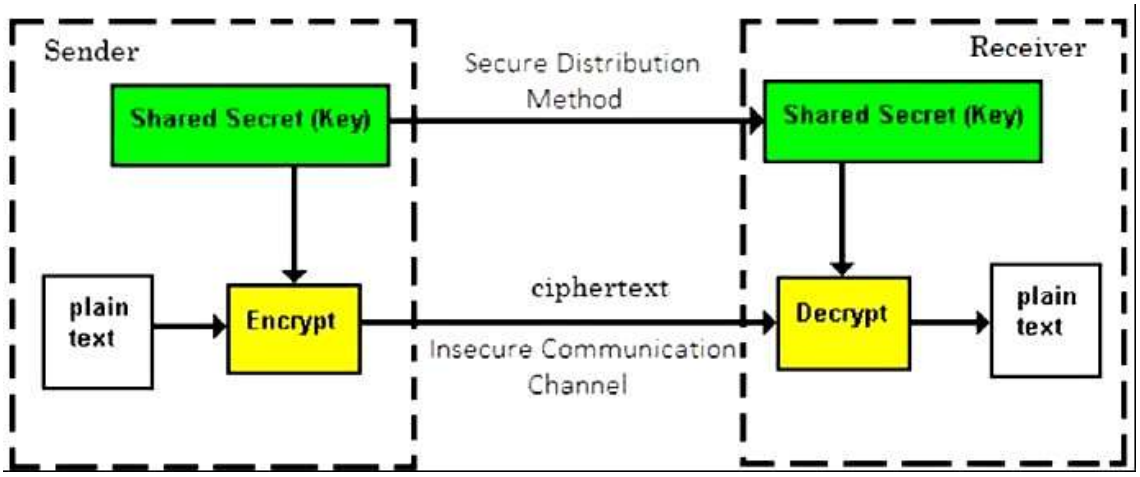
The biggest difference between these cryptosystems is the link between encryption and decryption keys. In any cryptosystem, both keys are logically connected. It is almost impossible to decrypt the ciphertext using a key unrelated to the encryption key.

Symmetric Key Encryption

The encryption process where the same keys are used for encrypting and decrypting the information is known as Symmetric Key Encryption.

The study of symmetric cryptosystems is referred to as symmetric cryptography. Symmetric cryptosystems are also sometimes referred to as secret key cryptosystems.

A few well-known examples of symmetric key encryption methods are - Digital Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH.



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 need to be changed on a regular basis, this technique becomes expensive and cumbersome.
- In a group of n people, to enable two-party communication between any two persons, the number of keys required for the group is n x (n − 1)/2.
- Length of Key (number of bits) in this encryption is smaller and hence, process of encryption-decryption is faster than asymmetric key encryption.
- Processing power of a computer system required to run a symmetric algorithm is less.

Challenge of Symmetric Key Cryptosystem

There are two main challenges of using symmetric key cryptography.

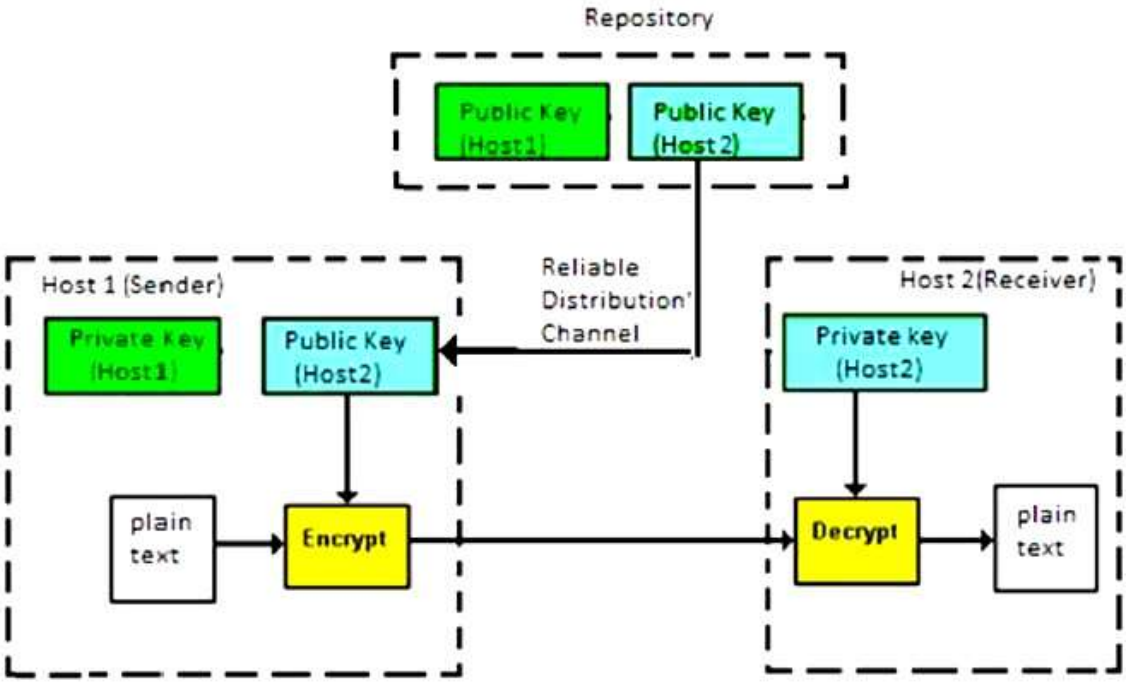
- Key establishment − Before any communication, both the sender and the receiver need to agree on a secret symmetric key. It needs a secure key management scheme in place.

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

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

Asymmetric(Public) Key Encryption

Asymmetric Key Encryption is the process of encrypting and decrypting information using various keys. Though the keys differ, they are mathematically similar, therefore getting plaintext via decrypting ciphertext is possible. The process is depicted in the following illustration −



Asymmetric Key Encryption was invented in the 20th century to overcome 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. Because of their mathematical relationship, these keys can be used to decrypt the ciphertext to retrieve the original plaintext when either is used for encryption.
- It requires putting the public key in a public repository and the private key as a well-guarded secret. Hence, this scheme of encryption is also called Public Key Encryption.
- Though public and private keys of the user are related, it is computationally not feasible to find one from another. This is a strength of this scheme.
- When Host1 has to send data to Host2, he gets the public key from the repository, encrypts the file, and sends it.

- Host2 uses his private key to extract the plaintext.
- Length of Keys (number of bits) in this encryption is large and hence, the process of encryption-decryption is slower than symmetric key encryption.
- Processing power of a computer system required to run an asymmetric algorithm is higher.

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

You may think, how can the encryption key and the decryption key be related', and yet it is impossible to determine the decryption key from the encryption key? The answer lies in the mathematical concepts. It is possible to create a cryptosystem whose keys possess this feature. The concept of public-key cryptography is relatively new. There are fewer public-key algorithms known than symmetric algorithms.

Challenge of Public Key Cryptosystem

One fundamental problem for public-key cryptosystems is that the user has to believe that the public key he is using in communication with another person is actually that person's public key and has not been created by a third party.

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

The third party ensures user identity by authorization, signing, or another procedure - 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 −

Heading	**Relation between Keys**	**Encryption Key**	**Decryption Key**
Symmetric Cryptosystems	Same	Symmetric	Symmetric
Public Key Cryptosystems	Different, but mathematically related	Public	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

Kerckhoff's principle is an important concept in the field of encryption. It suggests that the strength of an encrypted system should not depend on keeping the encryption method a secret. Rather,

security should come from protecting the encryption key. A well designed encrypted system needs to remain secure even if details about the encryption process become public knowledge.

This idea is crucial for modern encryption standards - it promotes transparent algorithm design and analysis. Knowing the specific encryption process is allowed, as long as the key is concealed. This encourages collaboration from security experts to thoroughly review encryption methods. It also means new algorithms can be openly proposed and tested, with confidence they will not automatically be broken simply by publication. Ultimately, Kerckhoffs' principle is foundational because it shifts focus to diligently shielding the encryption trigger rather than fruitlessly attempting to cloak the encryption mechanism itself.

The concept of information security through cryptography originated in the late 19th century. It was first proposed by Dutch cryptographer Auguste Kerckhoffs, who laid the groundwork for modern cryptanalysis and the design of secure encryption systems. Later in the 20th century, American mathematician and cryptographer Claude Shannon further developed the theory of cryptography and security.

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 insecure channel.
- The encryption apparatus and documents should be portable and operable by a single person.
- Finally, the system has to be easy to utilize, required neither mental strain nor knowledge of a long set of rules to follow.

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

Importance of Cryptosystems

Cryptosystems play an important role in protecting sensitive information sent over the internet. They allow users to transmit private data, like credit card numbers, in a secure way. Cryptography has various applications beyond just messaging. For example, a secure email system may incorporate digital signatures to verify a sender's identity. It could also use cryptographic hash functions to validate that a message has not been altered during transit. Additionally, such a system would manage encryption keys to encrypt and decrypt correspondence. Overall, the techniques of cryptography help guarantee privacy and integrity in digital communications.

Cryptography - Data Encryption Standard

The DES which stands for Data Encryption Standard algorithm, is a symmetric key block cipher created by an IBM team in the early 1970s and some time later it is adopted by the National Institute of Standards and Technology. In this method we divides plaintext into 64-bit blocks and transforms it to ciphertext with the help of 48-bit keys.

As you may know, it is a symmetric-key method, it uses the same key to encrypt and decrypt the data which we have already discussed in the previous chapter. If it were an asymmetrical algorithm, it will require different keys for encryption and decryption.

Some systems can break the DES algorithm. This algorithm uses a 56-bit key. DES uses this key to convert a block of 64-bit plaintext into a block of 64-bit ciphertext.

The DES process has several steps, each of which is referred to as a "round." The number of rounds vary depending on to the size of the key we have used. For example, a 128-bit key takes 10 rounds, a 192-bit key takes 12 rounds, and so on.

History of DES Algorithm

DES is based on the LUCIFER Feistel block cipher, which was invented in 1971 by IBM cryptography researcher Horst Feistel. DES uses 16 rounds of the Feistel structure, with each round using a unique key.

In November 1976, DES was approved as the government encryption standard, which was later reaffirmed in the years 1983, 1988, and 1999.

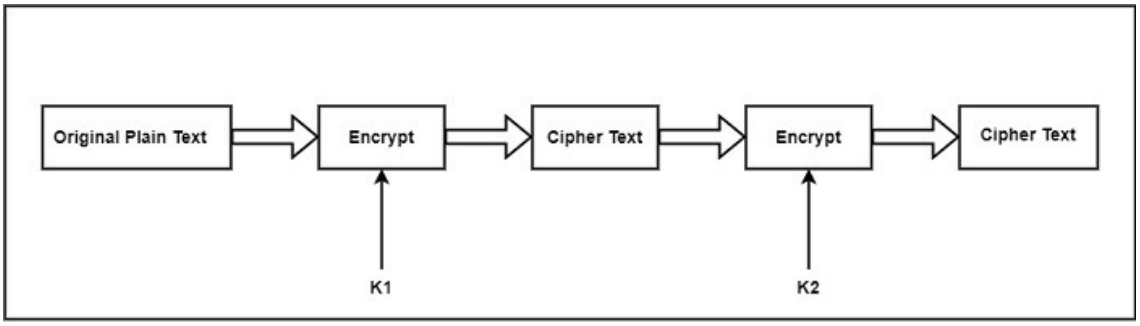
After a public competition to develop a replacement, the Advanced Encryption Standard (AES) took over as the acknowledged standard in 2002, ending DES's monopoly. In May 2005, the NIST officially revoked FIPS 46-3 (the 1999 reaffirmation), however Triple DES (3DES) is still allowed for sensitive government information until 2030.

Types/Variations of DES

There are two main variations of Data Encryption Standard are as follows −

Double DES

Double DES is a type of encryption in which two instances of DES must be present in the same plaintext. In both scenarios, the plaintext is encrypted with the help of a number of keys. For the purpose of decryption, both keys are required.



Double DES Encryption

Triple DES

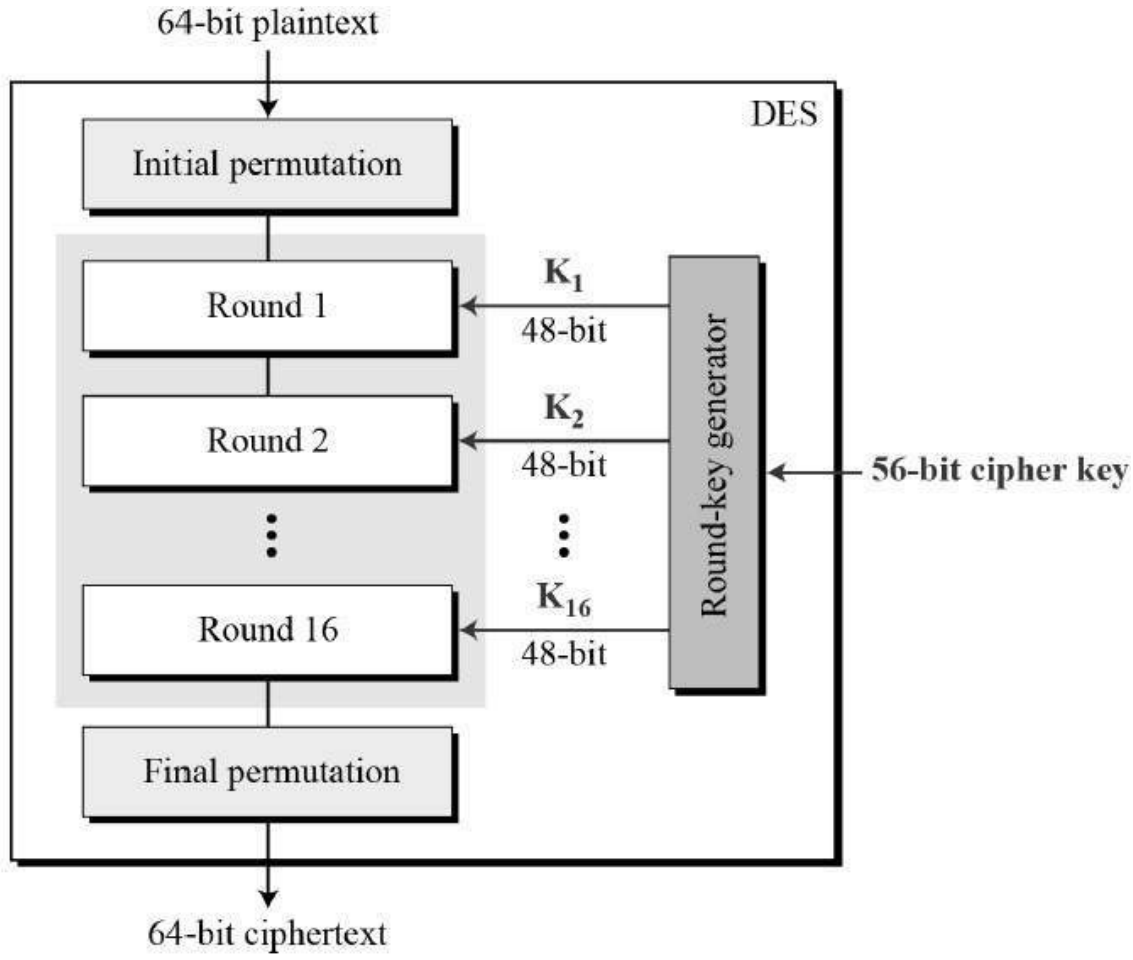
There are two types of Triple DES which are as follows −

Triple DES with Two Keys − In triple DES with two keys there are only two keys K1 used by the first and third process and K2 used in the second process.

Triple DES with Three Key − In Triple DES, the plaintext block P is first encrypted with a key K1 then encrypts with a second key K2 and finally with a third key K3 where K1, K2, and K3 are distinct from each other.

Structure of DES

DES uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −

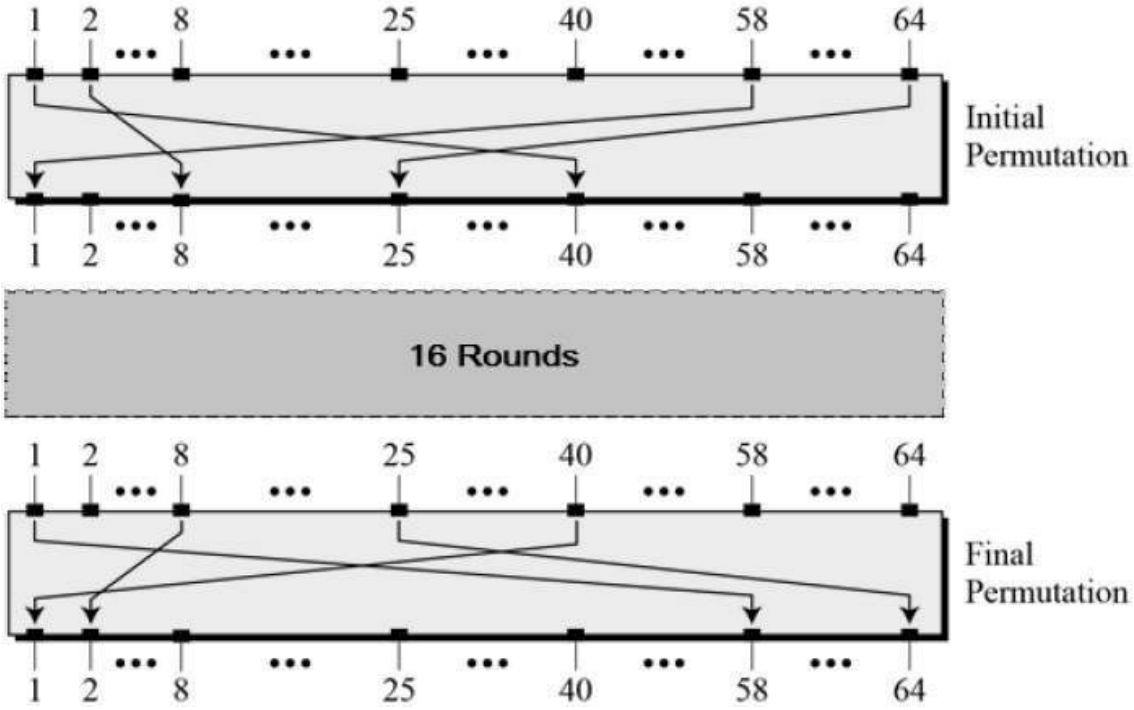


As DES is based on the Feistel Cipher, all that is required to specify DES is −

- Round function
- Key schedule
- Any additional processing - Initial and final permutation

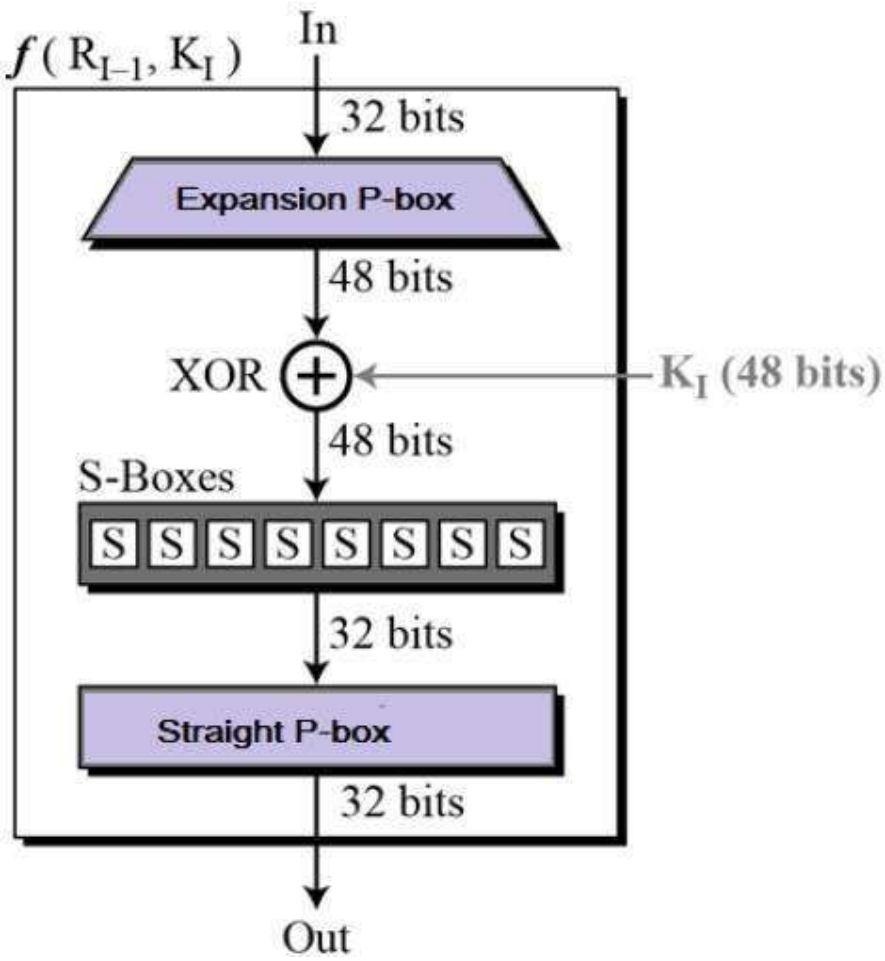
Initial and Final Permutation

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



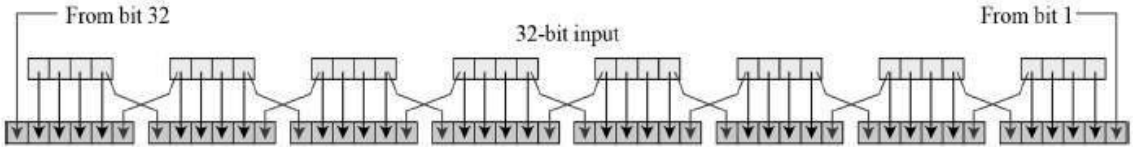
Round Function

The heart of this cipher is the DES function, f. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



Now let us discuss the above image and its terminologies −

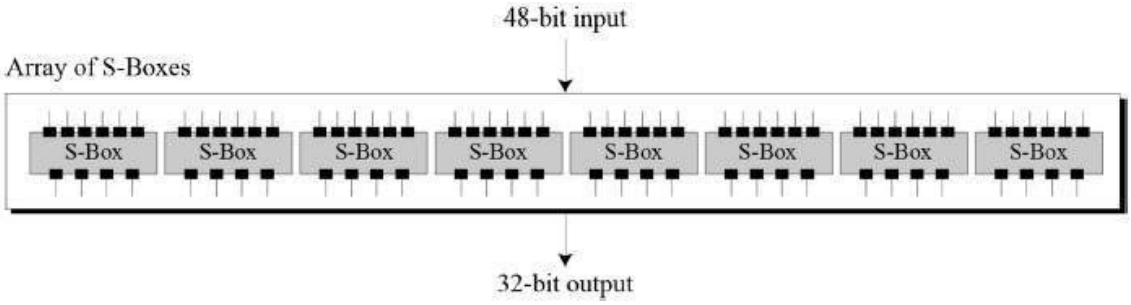
- Expansion Permutation Box − Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration −



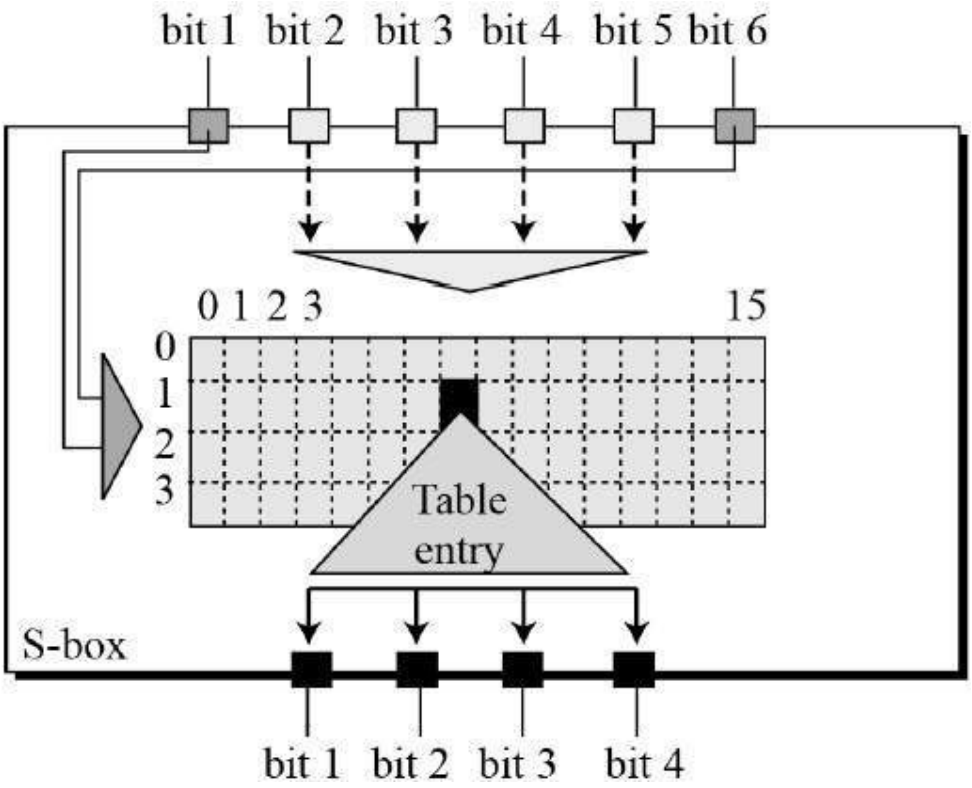
- The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown −

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

- **XOR (Whitener)** − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. This is the only type of operation that uses the round key.
- **Substitution Boxes** − The S-boxes carry out the real mixing (confusion). Eight S-boxes, each with a 6-bit input and a 4-bit output, are used in DES. Refer the following illustration −



- The S-box rule is illustrated below −



- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.
- **Straight Permutation** − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration −

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

Algorithm of DES

The following steps define the algorithm process −

- **Step 1** − The 64-bit plaintext block is initially provided to an initial permutation (IP) function to start the process.
- **Step 2** − After that, the plaintext undergoes to the initial permutation (IP).
- **Step 3** − The first permutation (IP) then generates Left Plain Text (LPT) and Right Plain Text (RPT), which are the two sides of the permuted block.
- **Step 4** − The encryption procedure consists of sixteen cycles for each LPT and RPT.

- **Step 5** − Lastly, the LPT and RPT become one, and the newly combined block is subjected to a Final Permutation (FP).
- **Step 6** − This procedure yields the intended 64-bit ciphertext.

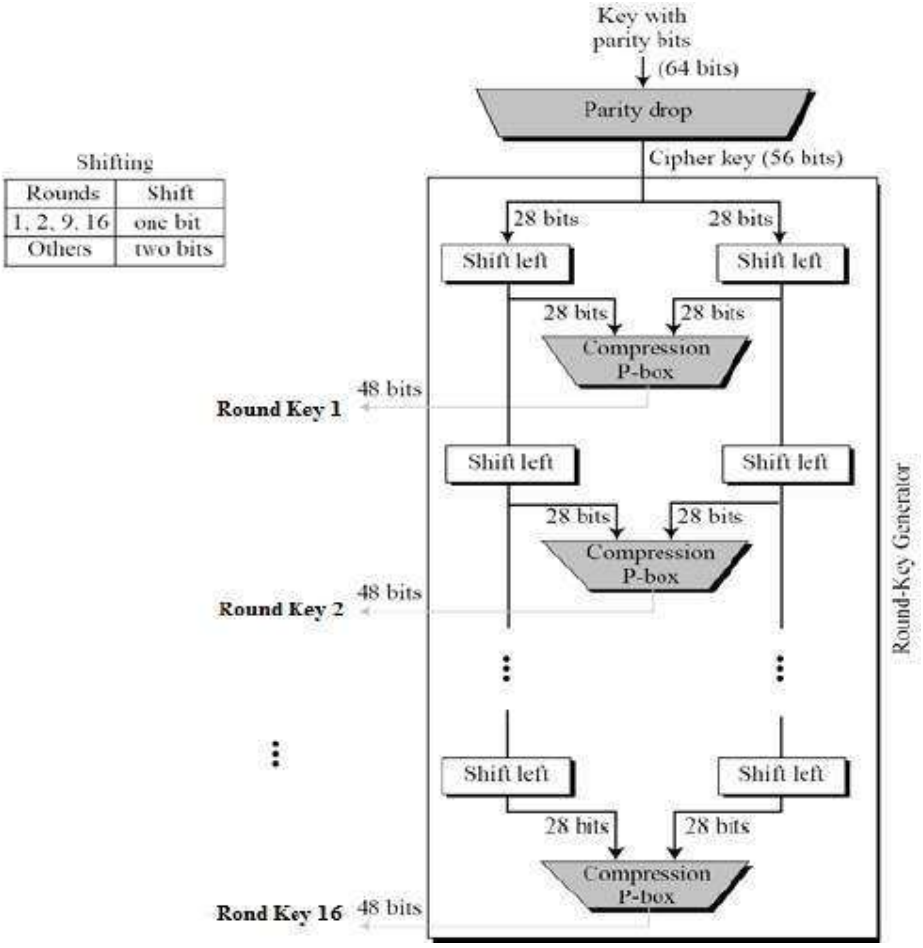
Step 4 of the encryption process can be further divided into five stages − Key transformation, Expansion permutation, S-Box permutation, P-Box permutation, XOR and swap.

We apply the same process for decryption, with the exception we reverse the of 16 round keys.

Next, we will explore the several DES modes of operation to get a better understanding of what DES is.

Key Generation

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



The algorithm implement 16 rounds of encryption and for each round, a unique key is produced. Before transforming to the steps, it is essential to understand that in plaintext the bits are labeled from 1 to 64 where 1 is the most significant bit and 64 is the least significant bit. The process of generating keys are as follows −

- The round key generator produce sixteen 48-bit keys out of a 56-bit cipher key. The cipher key is provided as 64 bit key in which 8 extra bits are parity bits, which are discarded before the actual key generation process begins.
- The parity bit drop process drops the parity bits (bit 8, 16, 24, 32...64) from the 64-bit key and permutes the remaining bit according to the pre-defined rules as display in the parity bit drop table below.
- These remaining 56 bits are generally used for key generation.
- After the permutation, the keys are divided into two 28 bits parts. Each part is changed left one or two bits is depend on the rounds.
- In round 1, 2, 9, and 16 shifting is one bit and in the other rounds it is two bits. The two parts are integrate to build a 56 bit part.
- Thus the compression D-box transform it into 48 bit. These 48 bits are being utilized as a key for a round.

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

DES Analysis

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

- **Avalanche effect** − The ciphertext changes significantly in reaction to even minor changes in the plaintext. It can change in one bit of plaintext creates change in some bits of the ciphertext. The advantage of Avalanche effect can be made accessible in securing embedded software wherever DES and AES algorithms are used. The attackers are attempting a lot to smuggle the data saved in the databases.
- **Completeness** − Completeness effect defines that each bit of the ciphertext required to base on some bits on the plaintext. The diffusion and confusion developed by Dboxes and S-boxes in DES, show a very strong completeness effect. The completeness feature tightness the avalanche concept even more. It needed that change in the ciphertext is consistently distributed for each changed bit of the input plaintext or the key.

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

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

DES Strength

The strength of DES is as follows −

- **Use of 56 bit keys** − A 56-bit encryption key provides 256 possible keys, making brute force attacks impossible due to the vast number of combinations. Even with a machine performing one DES encryption per microsecond, it would take thousands of years to decrypt the cipher.

Diffie and Hellman suggested the feasibility of a parallel machine with one million encryption units, each capable of one encryption per microsecond. However, decrypting messages may require more

than just trying all possible keys, especially if the message is in English plaintext, which can be identified easily. If the text is compressed before encryption, decryption becomes more challenging.

- **The nature of algorithm** − Cryptanalysts can conduct cryptanalysis on the DES algorithm by exploiting its features. They focus on weaknesses in the eight substitution tables, or S-boxes, used in each iteration. While researchers have discovered many regularities and unexpected behaviors in the S-boxes, no one has yet found weaknesses in them.

This ongoing pursuit has led to the identification of numerous irregularities and unexpected behaviors in the S-boxes over the years.

- **Timing Attacks** − A timing attack is a security exploit within a broader class known as side-channel attacks, where an attacker measures the response time of a system to different inputs to extract sensitive data. Kocher designed a timing attack specifically targeting RSA decryption secret keys, primarily used in hardware security tokens like smartcards.

These attacks focus on public key algorithms, extracting information about keys or plaintexts by observing decryption times of multiple ciphertexts. To mitigate timing attacks, techniques like "blinding" are used to ensure consistent computation times regardless of the key or message being processed.

Advantages of DES

The DES method's advantages are −

- The US authority has set it as a standard.
- It works on hardware faster than it does on software.
- Triple DES utilises a highly difficult-to-break 168-bit key.

Disadvantages of DES

The drawbacks or disadvantages of DES are as follows −

- It is an algorithm with weak security.
- It can be break using Brute force attacks.
- There is a DES breaking device on the market called Deep Crack.

Attacks on DES

There are various attacks on DES which are as follows −

- **Differential Cryptanalysis** − The main objective of differential cryptanalysis is to view for statistical distributions and patterns in ciphertext to provide deduce element about the key used in the cipher.

Differential cryptanalysis is a section of study in cryptography that compares the method differences in input associated to the differences in encrypted output. It can be used basically in the study of block ciphers to decide if changes in plaintext result in any non-random outcomes in the encrypted ciphertext.

- **Related-Key Cryptanalysis** − Related-key cryptanalysis consider that the attacker understand the encryption of specific plaintexts not only under the original (unknown) key K, but also below some derived keys K0 = f (K).

In a chosen-related key attack, the attacker defines how the key is to be modified and known-related-key attacks are those where the key difference is acknowledged, but cannot be selected by the attacker.

It can emphasize that the attacker understand or select the relationship between keys, and not only the actual key values.

Related-key cryptanalysis is a factual attack on key-exchange protocols that do not provide key-integrity an attacker can be capable to ip bits in the key without understanding the keypad key-update protocols that update keys utilizing a known function such as K, K + 1, K + 2, etc. Related-key attacks were also utilized against rotor devices such as operators consistently set rotors incorrectly.

- **Linear Cryptanalysis** − Linear cryptanalysis is a general form of cryptanalysis depend on discovering affine approximations to the element of a cipher. Attacks have been produced for block ciphers and stream ciphers. Linear cryptanalysis is one of the two most generally used attacks on block ciphers and the other being differential cryptanalysis.

Linear approximate equations is depend on the best (n-2) round expression, and dependability of the key candidates changed from these equations. The former decrease the number of needed plaintexts, whereas the latter enhance the success rate of the attack.

- **Brute Force Attack** − In cryptanalysis, a brute force attack is an approach of defeating a cryptographic scheme by attempting a huge number of possibilities.

For example, it can be exhaustively working through all possible keys in order to decode a message. The selection of an appropriate key length based on the practical feasibility of implementing a brute force attack.

For symmetric-key ciphers, a brute force attack generally means a brute-force search of the key area; that is, checking all possible keys in order to find the plaintext used to create a specific cipher text.

In a brute force attack, the expected number of trials before the proper key is discovered is similar to half the size of the key space. For instance, if there are 264 possible keys, a brute force attack can generally be normal to discover a key after 263 trials.

Factors Affecting Security of DES

There are some factors which affects the security of DES is as follows −

- **Weak Keys** − Because of the method the initial key is changed to receive a subkey for each round of the algorithm, specific initial keys are weak keys. The initial keys value is divided into two halves and each half is changed independently.

If all the bits in each half are either 0 or 1, thus the key can be used for some cycle of the algorithm is the same for all the cycles of the algorithm. This can appear if the key is completely 1s, completely 0s, or if one half of the key is completely 1s and the other half is completely 0s. So that creates DES less secure.

- **Algebraic Structure** − The DES encryption operation can form a group, and encrypting a group of plaintext blocks with k1 followed by k2 can be equal to encrypting the blocks with k3.

Even worse, DES can be vulnerable to a meet-in-the-middle known-plaintext attack that runs in only 228 steps. If DES were closed then for any k1 and k2 there would be a k3 such that −

Ek2(Ek1(P))=Ek3(P)

- **Key Length** − If there is a possibility to speed up the searching procedure by timespace tradeoff. The possibility of calculating and saving 256 possible results of encrypting an individual plaintext block under each possible key and then to break an unknown key, and it is required to add data blocks into the encryption stream, recover the resulting cipher text and view the key up.

No. of Rounds − No of rounds kept 16 because reduced number of rounds has been strongly attacked. DES with three or four rounds was simply broken .DES with any number of rounds fewer than 16 can be broken with a known plaintext attack more effectively than by a brute-force attack.

Testing and Implementing DES

Implementing DES needs a security provider. Choosing a supplier is an essential first step in the process of implementation, even if there are multiple choices. The language that you work in, like MATLAB, Java, Python, or C, can have an impact on the choice you make.

After choosing a provider, you have to decide whether to build the key that the Key Generator will produce at random using a byte array or plaintext.

Testing the encryption is also important for making sure that it is being used correctly.

Design Issues of DES

Design issues in DES include S-Boxes, D-Boxes, and the number of rounds.

- S-Boxes are used to substitute 48-bit inputs into 32-bit outputs, adding non-linearity and complexity to the encryption process. They follow specific properties such as permutations within rows and non-affine transformations.
- D-Boxes, similar to transposition ciphers, permute input bits from one round to the next, ensuring each S-box input comes from a different S-box output and avoiding repetition within rounds.
- DES employs sixteen rounds of Feistel ciphers, but it's been shown that after eight rounds, each ciphertext becomes a function of every plaintext and key bit, indicating sufficient security.

Cryptography - Triple DES (Data Encryption Standards)

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

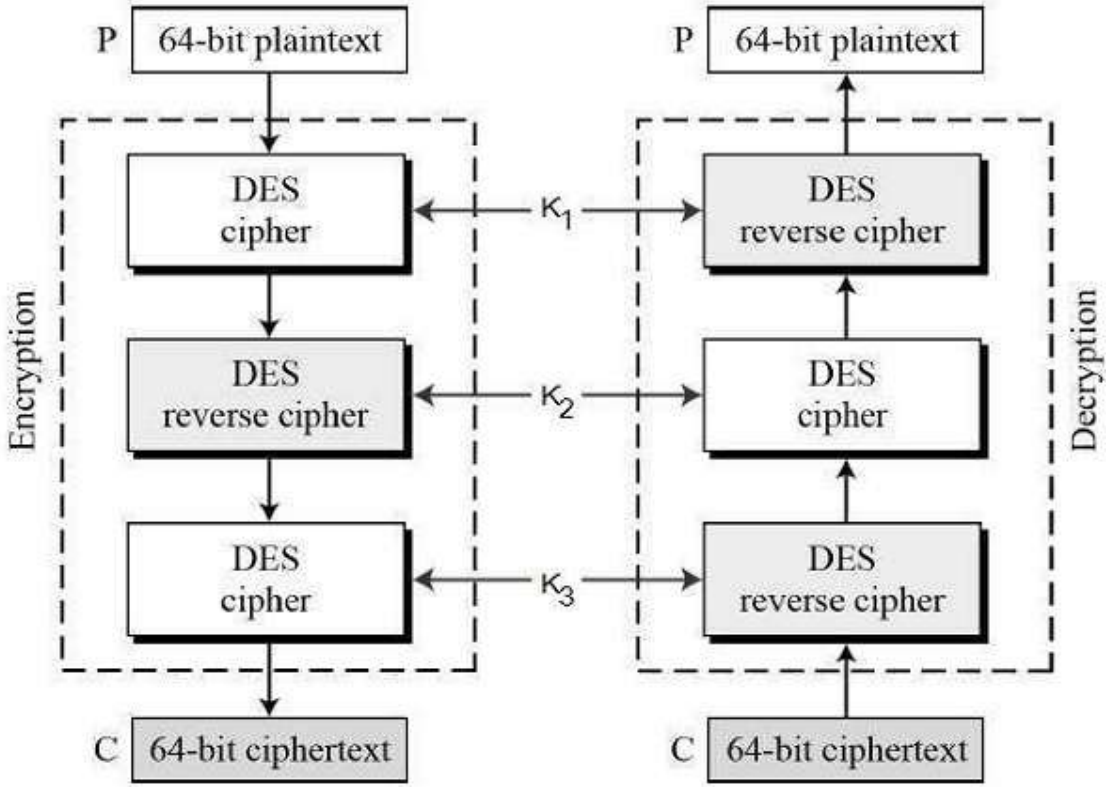
The pragmatic approach was not to abandon the DES completely, but to change the manner in which DES is used. This led to the modified schemes of Triple DES (sometimes known as 3DES).

It should be noted that there are two versions of Triple DES: 2-key Triple DES (2TDES) and 3-key Triple DES (3TDES).

What is Triple DES?

Triple Data Encryption Algorithm, often known as Triple DES, TDEA and 3DES is an encryption method that encrypts a single piece of text using three separate versions of DES. In the first, each key used is unique; in the second, two keys are same and one is different; and in the third, every key is similar. These are the different key selecting methods that are used.

Triple DES was first introduced in 1998. For a longer effective key length, it runs the DES cipher algorithm three times over each data block.



Encryption Process of Triple DES

- The three stages of Triple DES operation are Encrypt-Decrypt-Encrypt (EDE). It functions by using a key bundle, made up of three 56-bit keys (K1, K2, and K3).
- Initially K1 is used for encryption; K2 is then used for decryption; and K3 is used for the final encryption. There is a two-key Triple DES version where K1 is used for the first and end steps but the same algorithm is done three times. In 2015, this two-key version was discontinued.
- Because of the limitation to use double enciphering, the algorithm is executed three times. Meet-in-the-middle attacks are a type of attacks that encrypt from one end, decrypt from

the other, and search for collisions, or keys resulting in the same outcome in both directions. Double DES, or any other cipher run twice, would only be twice as strong as the base cipher if memory were enough.

- In simple terms, the double cipher is going to have a shorter key and be equally powerful as the identical cipher run once.
- Not only this, but encrypting twice with two keys is equivalent to encrypting once with an alternate key if the cipher combines together. However determining the other key is not simple, using every single key in a brute-force attack will result to the discovery of the third key.
- So multiple encrypting is a waste of time if the cipher is a group.
- An operator and a set have a connection known as a group. They form a group if their behaviour with addition is nearly the same as that of integers.
- A group is also formed if you continue to encrypt a block and it finishes a full circuit over the set of potential blocks.
- DES does not belong to a group. But DES has popular structural characteristics that lead some to maintain that it's not definitely not a group, in other words, it can be a group.
- For example, there are known DES loops in which you can get stuck in an indefinite loop by continuously encrypting with the same key.

Encryption Modes

Triple DES allows the DES algorithm to be used in each of the three rounds in either direction, for encryption or decryption. As a result, Triple DES has eight different possible modes.

Mode	**Encryption Sequence**
DDD	Decrypt-Decrypt-Decrypt
DDE	Decrypt-Decrypt-Encrypt
DED	Decrypt-Encrypt-Decrypt
DEE	Decrypt-Encrypt-Encrypt
EDD	Encrypt-Decrypt-Decrypt
EDE	Encrypt-Decrypt-Encrypt
EED	Encrypt-Encrypt-Decrypt

EEE	Encrypt-Encrypt-Encrypt

If there was a better choice, you are unlikely to want to use EEE or DDD mode for the same structural reasons that you could not want to use EED, DEE, DDE, or EDD. The most effective compositions are EDE or DED due to the weak nongroupness of DES. EDE additionally makes more sense. You have to clarify how Triple DES begins with decryption if you decide to use DED.

Modules of Triple DES

There are some modules of Triple DES which are as follows −

- **Admin Login** − In this project, admin can get in the username and password to validate himself to access the account panel modules.
- **User Login** − In this module, users can get in their username and password to authenticate themselves to access their account panel modules.
- **User Registration Module** − In this module, users can get in their username and password and address, mobile, email id to register themselves to access the account panel modules.
- **Create Message Module for Admin** − In this module, admin can choose the username and then enter the message along with the subject and also the input encryption key which can be used for encrypt the message and the subject and then send it to the selected user and message and subject are both saved into the user inbox.
- **Check Suspicious Mails for Admin** − In this module, admin can verify the suspicious mails which is not generally stored into the user inbox rather than marked as suspicious status and sent it to the admin as suspicious mails with the user element.
- **Data Dictionary for Admin** − In this module, admin can add the suspicious words into current data dictionary to detect more directly and efficiently the suspicious mails sent by the users.
- **View Data Dictionary for Admin** − In this module, admin can see the suspicious words exists into the data dictionary and also has access to remove the suspicious words from the current data dictionary of suspicious words.
- **View Users List for Admin** − In this module, admin can see the registered users and their full element and has access to remove the users if any of the registered users are discovered to do the suspicious event on the website.
- **Create Message Module for Users** − In this module, users can choose the other users and then enter the message forward with the subject and then send it to the selected user and message and subject are both saved into that inbox of taking user and at back end of the website. Suspicious mail detection module is processed which is identified the sent mails marked as suspicious or normal.
- **Inbox Module for Users** − In this module, users can see the inbox messages which are sent by the users those who are already registered if the user is any registered user then this message will be checked without any decryption module and also has access to remove that mail.

Security of Triple DES

The Triple DES method uses keys to encrypt data in order to secure it. Triple DES can be used in two different ways. The first method must be highly secure because it uses three different keys. It is not as safe as we figured out, however, due to a manner in which attackers can get in. It is similar to a door having three locks on it, but an attacker can still get in easily.

The second approach is weak because one of the keys is the same as the first. Think it as if a door had two locks, just one of which is simple to open. Because of this, it is no longer recommended to use because it is not very secure.

Also, there is an issue with Triple DES when multiple pieces of data are secured by the same key. This is similar to having a too short code, which makes it easy for someone to decipher. The problem exists in TLS and OpenVPN systems.

Triple DES is no longer advised because to these problems, and it is no longer present in more recent iterations of security programs like OpenSSL.

Advantages

Here are some advantages of the Triple DES −

- As compared to the triple-layered encryption to the original DES, security is enhanced.
- 3DES allows for smooth migrations through preserving connectivity with current DES implementations.
- By setting each of all three keys to the same value, 3DES can be used for a single DES with respect to backward compatibility.
- 3DES is widely used and integrated with a wide range of hardware, protocols, and applications.

Disadvantages

Below are some disadvantages of Triple DES which we have to consider while using it −

- The slower speed of 3DES compared to recent encryption methods like AES decreases processing efficiency.
- Even while 3DES is more powerful than DES, its efficient key length is limited, especially if three 56-bit keys are used.
- In scenarios with limited resources, the triple encryption process reduces performance because it needs more computing power.
- 3DES has a lower security margin than more recent encryption methods like AES, but being more secure than DES.

Implementation of Triple DES

The following applications can secure data with Triple DES −

Botan, Bouncy Castle, cryptlib, Crypto++, Libgcrypt, Nettle, OpenSSL, wolfSSL, Trusted Platform Module (TPM).

But keep in mind that in their most recent revisions, some of these programs may not have Triple DES as a default option. It is important that you confirm its availability in the particular version you are using.

Cryptography - Double DES

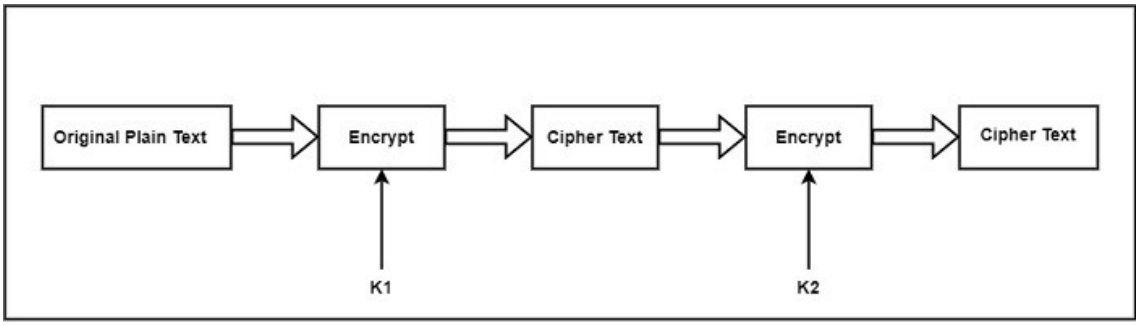
In the previous chapter we saw what is data encrytption standard in this chapter we will see detailed information about Double DES.

Double DES is a type of encryption where the same plaintext is encrypted using a pair of DES. Different keys are provided in both cases to encrypt the plaintext. Learning double DES is easy.

Double DES uses two keys, k1 and k2. For it to obtain the encrypted text, it can apply DES to the original plaintext using k1. With a different key, k2, it can apply DES to the encrypted text this time. The encrypted text that is displayed is the final output.

How does Double DES Work?

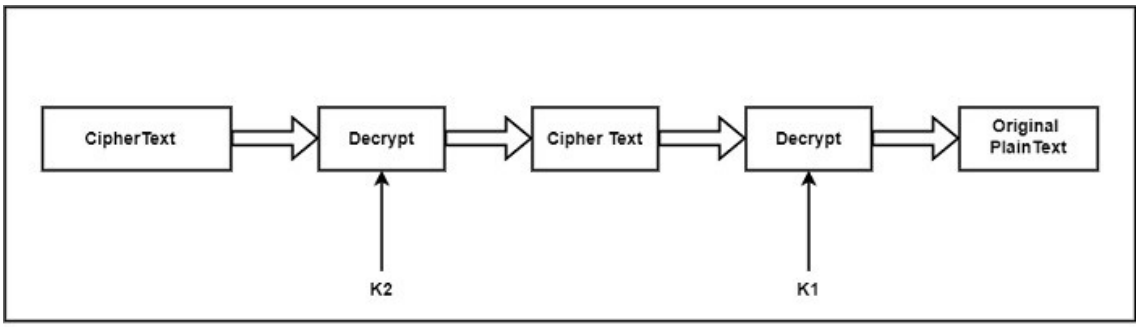
To create the singly encrypted ciphertext, first use the key K2 to decrypt the double encrypted cipher-text block. The original plaintext block can be retrieved by decrypting this ciphertext block with key K1.



Double DES Encryption

If it is capable of using a single-bit key, then 0 and 1 are the two available keys. There are four possible key values, like (00, 01, 10 and 11), if it can use a two-bit key.

In most cases, the cryptanalyst must implement 2^n operations in order to try out every possible key if it can use an n-bit key. The cryptanalyst will have to make 2^{2n}n attempts to crack the key if it is possible to use two distinct keys, each with n bits.



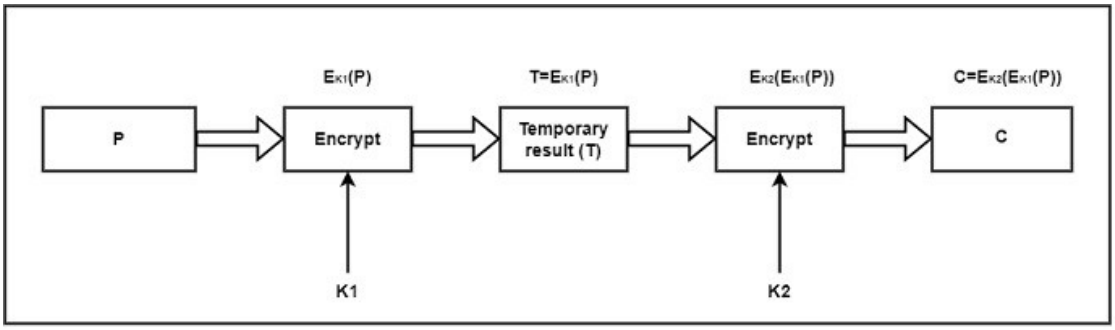
Double DES Decryption

Think about a message's P (plain-text block) and C (corresponding final cipher-text block) as the two basic elements of information that the cryptanalyst is aware of. Double DES stated numerically, as seen in the figure.

The result of the first encryption is known as T and is indicated as $T = E_{k1}(P)$ [i.e., encrypt the block P with key K1]. After this encrypted block is encrypted with another key K2, it indicate the result as −

$C = E_{k2}(E_{k1}(P))$

[i.e., encrypt the already encrypted block T, with a different key K2, and call the final ciphertext as C].



Mathematical Expression of Double DES

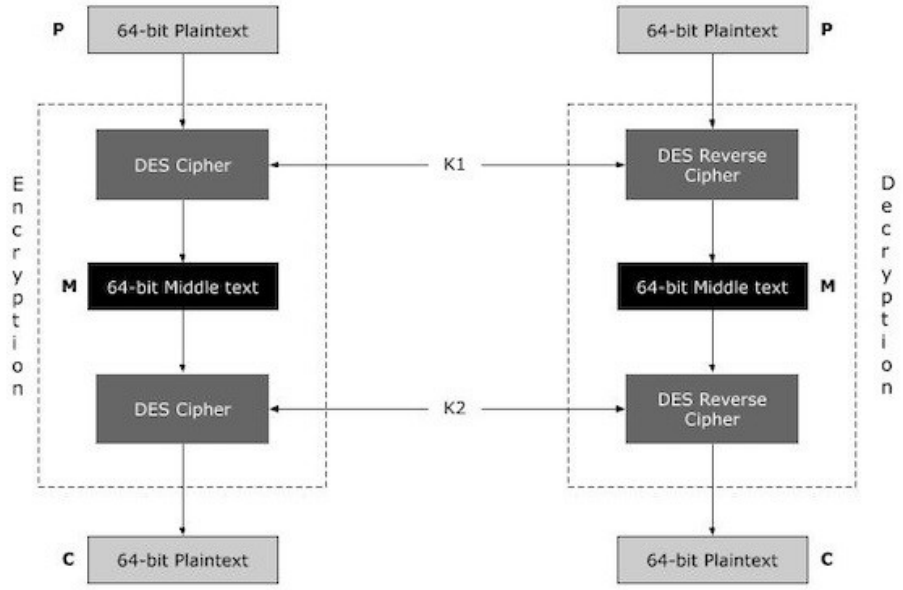
Advertisement

Meet in Middle Attack

In the double DES can be destroyed by known plaintext attack known as meet-in-themiddle attack.

Given a plaintext P and two encryption keys K1 and K2, ciphertext C is produced as $C = E_{k2}(E_{k1}, (m))$ decryption needed that the keys be used in reverse order −

$P = D_{k1}(D_{k2}, (C))$



A Meet-in-the-Middle (MitM) Attack is a type of cryptanalytic attack where the attacker need some type of space or time tradeoff to support the attack. MITM attempt can decrease the amount of difficulty needed to perform the assault in its original state.

Merkle and Hellman introduced the terms of meet-in-the-middle attack. This attack contains encryption from one end and decryption from another and connecting the result in the middle, therefore is the name meet-in-the-middle.

MITM can create the form of dividing the target connection into two so that each element can be addressed independently. It can mean changing an attack requiring X amount of time into one requiring Y time and Z space. The goal is to significantly decrease the effort required to implement a brute-force attack.

Meet-in-the-Middle attackers try to reconcile the difficulty contained in a high cryptanalytic attack by meeting in the middle, or halving the area of what they are analyzing to create the effort applicable or reasonable in their view.

The main aim of an attack is to steal personal information, including login credentials, account details and credit card numbers. Targets are frequently the users of monetary applications, SaaS businesses, e-commerce sites and other websites where logging in is needed.

Data acquired at the time of an attack can be used for several goals, such as identity theft, unapproved fund transfers or an illicit password change.

There are two keys including K1 and K2 are used for encrypt plaintext P into ciphertext C and the similar K1 and K2 are used for decryption. The intermediate text produced by first encryption and of first decryption, M should be the similar i.e., the two relationship must hold.

Let us say a cryptanalyst have a previous pair of P and C then it can use all possible values (256) of K1 and record all values of M. Similarly for all values of K2 access all M and thus compare these M's of K1 and K2 and discover a pair of K1 and K2 for which M is same.

If only one such pair occur then K1 and K2 are the desired keys. If more than one pair exists for which K1 and K2 are equal, another intercepted plaintext/ciphertext pair is utilized.

Advantages

The Double DES (Data Encryption Standard) method of data encryption involves quickly performing the DES algorithm twice. The following are Double DES's advantages −

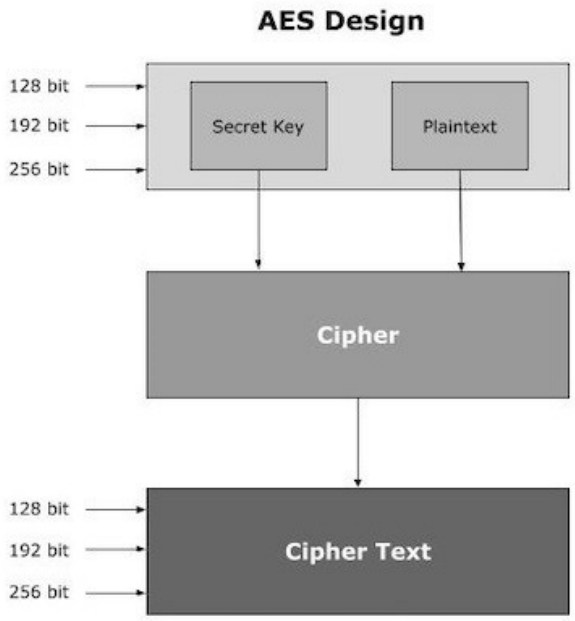
- **Enhanced Security** − By doubling the length of the key, Double DES improves the security of DES and makes it more difficult for attackers to decrypt encrypted data via brute-force attacks.
- **Compatibility with Current Systems** − Double DES can be implemented with current DES hardware and software, making it a simple option for businesses who currently use DES.
- **Widely Studied** − Due to the in-depth study and analysis that DES has received over the years, double DES can be applied and understood with the help of a variety of information and resources.

Disadvantages

- **Vulnerable to Meet-in-the-Middle Attack** − Double DES is vulnerable to a meet-in-the-middle attack, which involves an attacker intercepting the ciphertext and trying every key for the first encryption and decryption phases while storing the intermediate results. They then try each key for the second encryption stage until they find one that corresponds with the previously saved results. Therefore, the effective key length decreases to 112 bits, significantly weakening the theoretical 168-bit key length.
- **Performance Overhead** − Because double DES encryption takes more time and computing power than single DES encryption, it may have a performance overhead.
- **Limited Key Length** − While double DES has doubled the key length compared to single DES, it has a relatively limited key length (112 effective bits after calculating for the meet-in-the-middle attack). This may not be enough for handling sophisticated attacks, especially with the development of more powerful computing technologies.

Cryptography - Advanced Encryption Standards

An FIPS-approved cryptographic technique that can be used for securing electronic data is specified by the Advanced Encryption Standard (AES). Information can be encrypted and decrypted with the symmetric block cipher method known as the AES algorithm. Data that has been encrypted is transformed into ciphertext, an unreadable form; data that has been decrypted is returned to plaintext, the original form. Data can be encrypted and decrypted in blocks of 128 bits using the AES method with cryptographic keys of 128, 192, and 256 bits.



How AES encryption works?

Three block ciphers, or cryptographic keys, are part of AES −

- AES-128 encrypts and decrypts message blocks using keys of a length of 128 bits.
- Message blocks are encrypted and decrypted using a 192-bit key length by AES-192.
- Message blocks are encrypted and decrypted using a 256-bit key length using AES-256.

Each cipher uses cryptographic keys of 128, 192, and 256 bits, respectively, to encrypt and decrypt data in blocks of 128 bits. Ten, twelve, and fourteen encryption rounds are carried out to the 128-bit, 192-bit, and 256-bit keys, respectively.

A round of processing involves a number of processes, like mixing, transposition, and substitution of the plaintext input to create the final ciphertext output. The original data is secure and the encryption becomes more difficult to break the more rounds there are.

Many data transformations are carried out in AES. The data is first placed into an array, and then several encryption rounds are performed with the cipher transformations. Using a substitution table and an existing cipher, data substitution is the initial transformation.

With the exception of the first row, every data row is moved by one in the second transformation. The Hill cipher is used in the third transformation to combine columns. Each column, or data block, undergoes the final transformation using a distinct encryption key or a subset of it. Larger keys are needed for more rounds to complete.

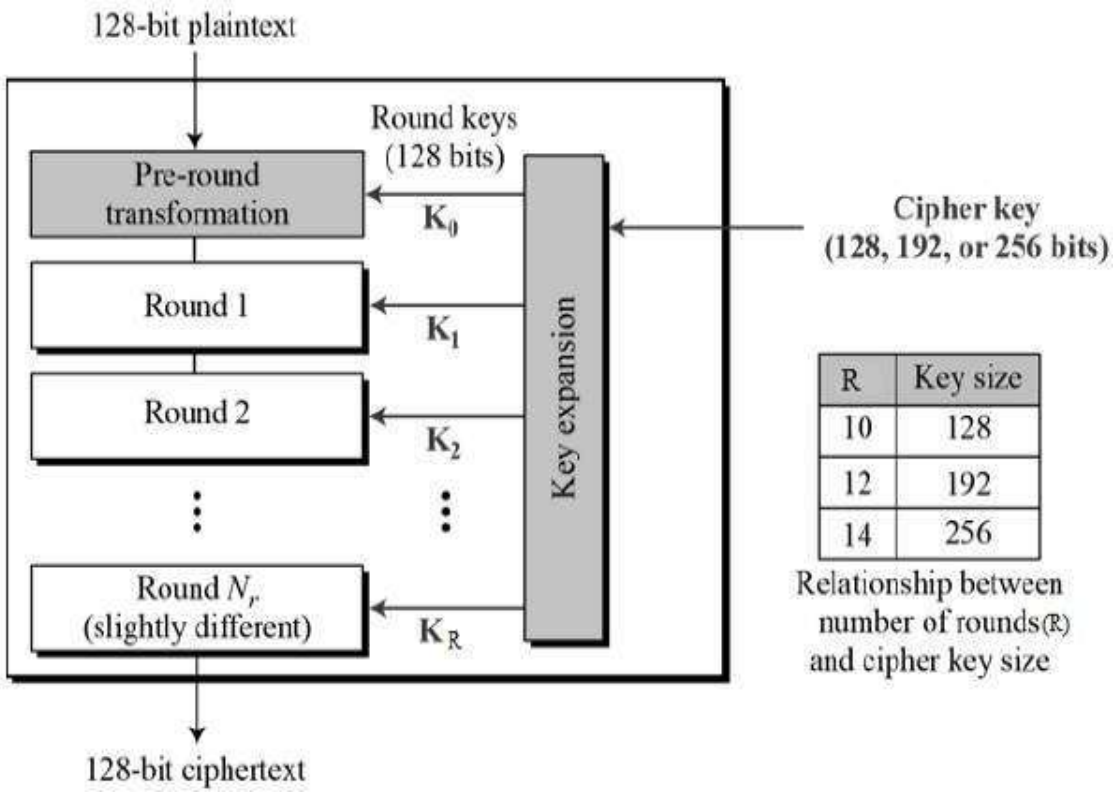
The message recipient decrypts the message by using a copy of the cipher to remove the many encryption layers and return the ciphertext to plaintext. They can read the communication after conversion and be sure that no one else has intercepted or read it.

AES 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 −

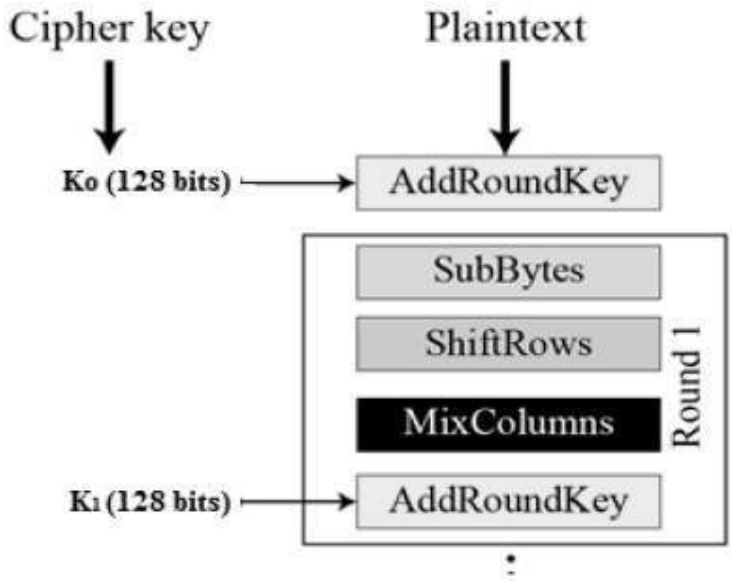


A plaintext block size of 128 bits, or 16 bytes, is required by the cipher. 16, 24, or 32 bytes (128, 192, or 256 bits) can make up the key length. AES-128, AES-192, or AES-256 are the names of the algorithm, depending on the key length.

A single 128-bit block serves as the input for both the encryption and decryption procedures. This block is represented as a 4 * 4 square matrix of bytes in FIPS PUB 197. At each step of encryption or decryption, this block is copied into the State array, which is updated. Following the final phase, an output matrix contains a copy of the current state.

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 −



Key Expansion

The round keys are calculated from the cipher key using Rijndael's block cipher schedule.

Pre−Transformation

This comprises of only 1 process namely Add_Round_Key. Here, XOR operation is performed on each data byte with a byte of the round key.

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 needs to be separately implemented, although they are very closely related.

Public Key Encryption

Public key cryptography provides a secure way to exchange information and authenticate users by using pairs of keys. The public key is used for encryption and signature verification, while the private key is used for decryption and signing.

When the two parties communicate with each other to transfer the intelligible or sensible message, referred to as plaintext, is converted into apparently random unreadable for security purposes referred to as ciphertext.

Public Key Cryptography

Public key cryptography is a method of secure communication that uses a pair of keys, a public key, which anyone can use to encrypt messages or verify signatures, and a private key, which is kept secret and used to decrypt messages or sign documents.

This system ensures that only the intended recipient can read an encrypted message and that a signed message truly comes from the claimed sender. Public key cryptography is essential for secure internet communications, allowing for confidential messaging, authentication of identities, and verification of data integrity.

Cryptographic Key

A cryptographic key is a piece of information used by cryptographic algorithms to encrypt or decrypt data, authenticate identities, or generate digital signatures. It serves as a parameter to control cryptographic operations, ensuring the security and privacy of digital communications and transactions.

How Does TLS/SSL Use Public Key Cryptography

TLS/SSL uses public key cryptography to keep our internet connections secure. It does this in two main ways:

- **Encryption**: When you visit a secure website (HTTPS), TLS/SSL helps encrypt data exchanged between your browser and the website's server.
  It uses a combination of public and private keys to create a secure connection. Your browser and the server agree on a secret key for this session, which keeps your data safe from eavesdroppers.
- **Authentication**: TLS/SSL verifies the identity of websites. When you connect to a site, it presents a digital certificate signed by a trusted authority. Your browser checks this certificate to ensure you're really connecting to the right site and not a fake one trying to steal your information.

By using public key cryptography, TLS/SSL protects our privacy online and ensures that the websites we visit are genuine and trustworthy.

Components of Public Key Encryption

- **Plain Text:** This is the message which is readable or understandable. This message is given to the Encryption algorithm as an input.
- **Cipher Text:** The cipher text is produced as an output of Encryption algorithm. We cannot simply understand this message.
- **Encryption Algorithm:** The encryption algorithm is used to convert plain text into cipher text.
- **Decryption Algorithm:** It accepts the cipher text as input and the matching key (Private Key or Public key) and produces the original plain text.
- **Public and Private Key:** One key either Private key (Secret key) or Public Key (known to everyone) is used for encryption and other is used for decryption.

Public Key Encryption Working

Key Pair Generation : A user generates a pair of keys :

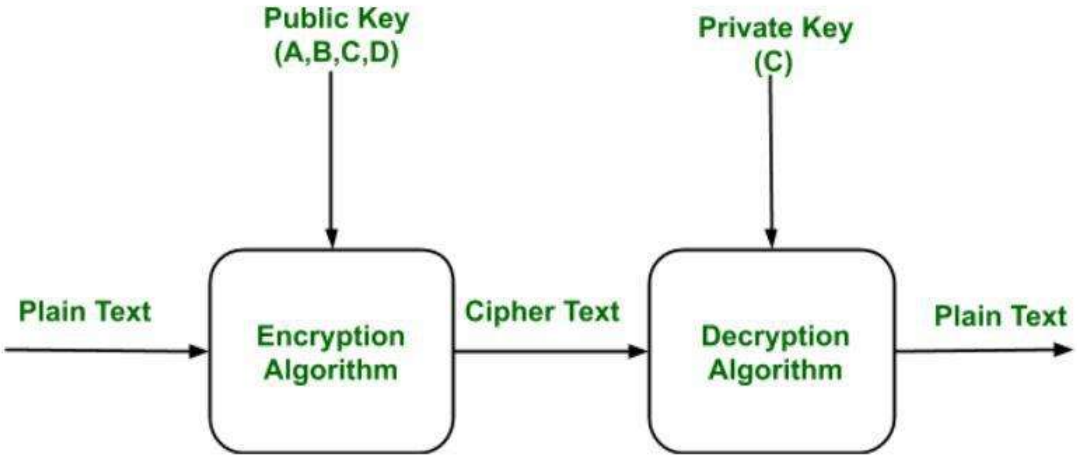
- **Public Key**: Shared openly. Anyone can use it to send an encrypted message.
- **Private Key**: Kept secret. Only the key owner can decrypt messages encrypted with the public key.

Encryption : If someone wants to send a private message:

- They obtain the recipient’s public key.
- They encrypt the message using that public key.
- The encrypted message is sent over a network.

Decryption : Upon receiving the message:

- The recipient uses their private key to decrypt the message and recover the original plaintext



Public Key Encryption

Public Key Encryption Practical Example: Secure Website (HTTPS)

When you visit a secure website like https://www.bank.com, public key encryption is used behind the scenes to encrypt data between your browser and the bank's server.

Bank's Server Has a Key Pair

- **Private Key**: Secret, stored securely on the server.
- **Public Key**: Shared with anyone via an SSL certificate.

You Connect to the Website

- Your browser gets the bank's public key from its SSL certificate.
- It verifies the certificate is valid (issued by a trusted certificate authority).

Encrypting the Session Key

- Your browser creates a random symmetric key (used for actual data encryption).
- It encrypts this key using the bank's public key.
- Only the bank can decrypt it using its private key.

Secure Communication Begins

- Now both your browser and the bank share a secret symmetric key.
- All further communication (login info, account data, etc.) is encrypted using this key.

Why Public Key Encryption is Used

- It ensures that only the server (with the private key) can read the symmetric key.
- Even if someone intercepts the traffic, they can't decrypt the session key or data.

Characteristics of Public Encryption key

- **Security Assurance**:
  It is computationally infeasible to determine the private (decryption) key from the public (encryption) key and algorithm alone.
- **Key Pair Flexibility**:
  Either key (public or private) can be used for encryption, with the other used for decryption supporting both confidentiality and authentication.
- **Easy Public Key Distribution**:
  Public keys can be shared freely, enabling convenient encryption and digital signature verification.
- **Private Key Confidentiality**:
  Private keys are kept secret, ensuring that only the key owner can decrypt content or create valid digital signatures.
- **Foundation of RSA**:
  The most widely used public-key cryptosystem, **RSA**, is based on the difficulty of factoring large composite numbers into primes.

Limitations of the Public Key Encryption

- **Susceptible to Brute-Force Attacks**: Although computationally hard, public key encryption algorithms can be theoretically brute forced if key lengths are too short or computational power advances (e.g., quantum computing).
- **Private Key Loss**: If a user loses their private key, they can no longer decrypt data or prove their identity, making the system non-recoverable and highly vulnerable.
- **Man-in-the-Middle (MitM) Attack Risk**: A third party could intercept and alter public keys during transmission, leading to unauthorized decryption or spoofed signatures if keys aren't verified through a trusted channel.
- **PKI Chain of Trust Vulnerability**: If a private key higher in the PKI hierarchy (e.g., a root certificate authority) is compromised, it can invalidate all subordinate certificates, enabling widespread MitM attacks.

Applications of the Public Key Encryption

- **SSL/TLS protocols** : They use public key encryption to securely exchange symmetric session keys between a web browser and a server.
- **Digital signature:** Digital signature is for senders authentication purpose. In this sender encrypt the plain text using his own private key. This step will make sure the authentication of the sender because receiver can decrypt the cipher text using senders public key only.
- **Key exchange:** This algorithm can use in both Key-management and securely transmission of data.
- **SSH keys** : For secure login to remote servers use public/private key pairs for authentication.
- **Blockchain and Cryptocurrencies**: Users control wallets with private keys , public keys serve as wallet addresses.

Cryptography Hash Functions

Cryptographic hash functions are mathematical algorithms that transform input data into a fixed-length sequence of characters, referred to as a hash value. Cryptographic hash functions are intended to be fast, deterministic, and one-way, meaning that even a minor change in input yields a very different hash. These functions are important for securing information digitally by allowing data verification and authentication.

- Cryptographic hash functions protect data integrity by creating identifying hash values, which enable systems to identify any unauthorized changes to messages or files in real time.
- Within cybersecurity, they are the foundation for digital signatures and certificate validation, giving a secure means to verify the authenticity of software and communications.
- They provide safe password storage through hashing passwords prior to saving, stopping direct exposure of sensitive credentials even if there happens to be a breach of data.
- Hash functions facilitate blockchain and other distributed ledgers by connecting blocks based on hash values to allow transparency and tamper-proof record-keeping.

Working of Cryptography Hash Function

- **Input Processing:** Cryptographic hash functions process an input of any length—whether text, file, or data stream—and subject it to a sequence of mathematical operations. The input can range from several bytes to gigabytes of information.
- **Fixed-Size Output Generation:** No matter what the length of the input, the function generates a fixed-size hash value, normally in the form of a hexadecimal string. This uniform output size provides equality regardless of the inputs.
- **Deterministic Operation:** The hash function consistently computes the same hash for the same input. Such a property enables uncompromising data authentication, as any alteration in the input leads to a totally unique hash.
- **Avalanche Effect:** A minor alteration in the input, even the flipping of one bit, significantly alters the resultant hash. The sensitivity ensures that collisions among hashes (two inputs having the same hash) are highly unlikely.
- **One-Way Computation:** The algorithm is made irreversible in the sense that it is computationally impossible to recover the original input from its hash value. This one-way feature protects sensitive information such as passwords and digital signatures.
- **Collision Resistance:** Hash functions used in cryptography are designed to minimize the probability of two distinct inputs generating the same hash value, upholding the integrity and trustworthiness of verification processes.

Properties of Cryptographic Hash Functions

- **Deterministic:** The same input always generates the exact same hash output, ensuring consistent and reliable verification of data.
- **Fast Computation:** Cryptographic hash functions are designed to process inputs quickly and efficiently, making them practical for handling large datasets and real-time applications.

- **Pre-image Resistance:** It is computationally infeasible to reverse-engineer or retrieve the original input data from its hash value, protecting sensitive information from exposure.
- **Second Pre-image Resistance:** Given an input and its hash, it is extremely difficult to find a different input that produces the same hash, preventing impersonation or forgery.
- **Collision Resistance:** The function minimizes the chance that two distinct inputs will produce identical hash values, ensuring unique data fingerprints for security and integrity.
- **Avalanche Effect:** Even a tiny change in the input, such as flipping a single bit, causes a significant and unpredictable change in the hash output, enhancing the function's sensitivity to data modifications.

Applications of Cryptographic Hash Functions

Below are some applications of cryptography hash functions

Message Authentication

- Message authentication is a system or service that verifies the integrity of a communication.
- It ensures data is received precisely as transmitted, with no modifications, insertions, or deletions, a hash function is used for message authentication, and the value is sometimes referred to as a message digest.
- Message authentication often involves employing a message authentication code (MAC).
- MACs are widely used between two parties that share a secret key for authentication purposes. A MAC function uses a secret key and data block to generate a hash value, that identifies the protected communication.

Data Integrity Check

- Hash functions are most commonly used to create checksums for data files.
- This program offers the user with assurance that the data is correct.
- The integrity check allows the user to detect any modifications to the original file.
- It does not assure uniqueness. Instead of altering file data, the attacker can update the entire file, compute a new hash, and deliver it to the recipient.

Digital Signatures

- The digital signature application is comparable to message authentication.
- Digital signatures operate similarly to MACs.
- Digital signatures encrypt message hash values using a user's private key.
- The digital signature may be verified by anybody who knows the user's public key.

Popular Cryptographic Hash Algorithms

MD5 (Message Digest Algorithm 5)

Once widely used for data integrity and digital signatures, MD5 is now considered insecure due to vulnerabilities that allow attackers to generate hash collisions easily. Its speed and simplicity made it popular historically, but it is no longer recommended for security-critical applications.

SHA-1 (Secure Hash Algorithm 1)

SHA-1 improved upon MD5 with a longer hash length and better resistance to collisions. However, advances in computational power and cryptanalysis exposed weaknesses, leading to practical collision attacks. Consequently, SHA-1 is deprecated for most security uses, including SSL/TLS certificates and digital signatures.

SHA-2 Family (SHA-256, SHA-512)

The SHA-2 family is currently the industry standard for cryptographic hashing, offering robust security with longer hash outputs of 256 and 512 bits. These algorithms provide strong collision and pre-image resistance, making them the preferred choice for secure communication protocols, blockchain technologies, and password hashing.

SHA-3 (Keccak)

Adopted as the latest NIST standard, SHA-3 uses a unique sponge construction different from SHA-2, enhancing security and flexibility. It offers comparable hash lengths with improved resistance to certain types of attacks, making it suitable for applications demanding long-term security.

BLAKE2 & BLAKE3

Designed as high-speed, secure alternatives to SHA-2 and SHA-3, BLAKE2 and BLAKE3 deliver faster hashing without compromising security. BLAKE3, in particular, supports parallel processing and incremental updates, making it ideal for modern systems requiring both speed and strong cryptographic guarantees.

Cryptography Digital signatures

Digital signatures are the public-key primitives of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message.

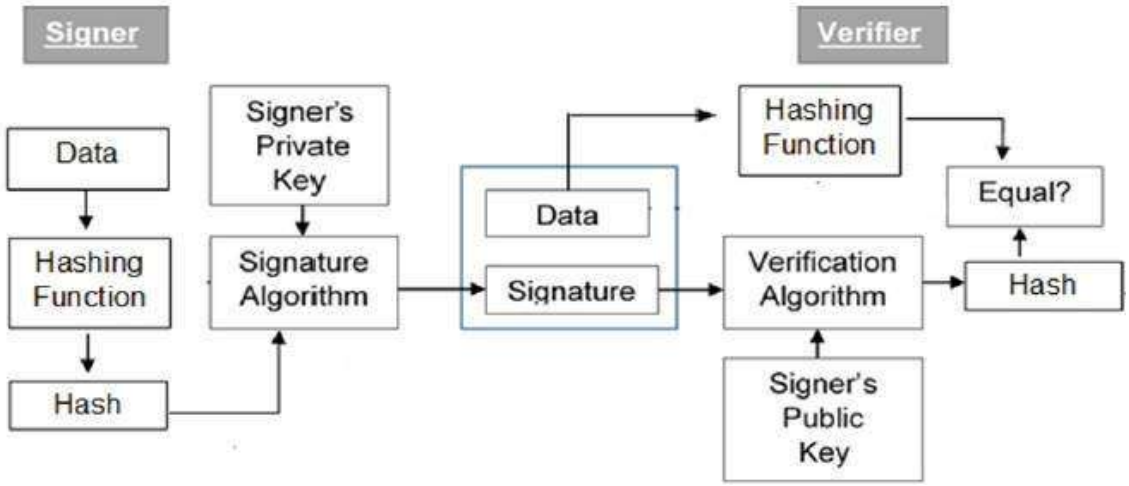
Similarly, a digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by receiver as well as any third party.

Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

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

Model of Digital Signature

As mentioned earlier, the digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration −



The following points explain the entire process in detail −

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- Signer feeds data to the hash function and generates hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.
- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- Verifier also runs same hash function on received data to generate hash value.
- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.
- Since digital signature is created by private key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

It should be noticed that instead of signing data directly by signing algorithm, usually a hash of data is created. Since the hash of data is a unique representation of data, it is sufficient to sign the hash in place of data. The most important reason of using hash instead of data directly for signing is efficiency of the scheme.

Let us assume RSA is used as the signing algorithm. As discussed in public key encryption chapter, the encryption/signing process using RSA involves modular exponentiation.

Signing large data through modular exponentiation is computationally expensive and time consuming. The hash of the data is a relatively small digest of the data, hence **signing a hash is more efficient than signing the entire data**.

Importance of Digital Signature

Out of all cryptographic primitives, the digital signature using public key cryptography is considered as very important and useful tool to achieve information security.

Apart from ability to provide non-repudiation of message, the digital signature also provides message authentication and data integrity. Let us briefly see how this is achieved by the digital signature −

- **Message authentication** − When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.
- **Data Integrity** − In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.
- **Non-repudiation** − Since it is assumed that only the signer has the knowledge of the signature key, he can only create unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

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

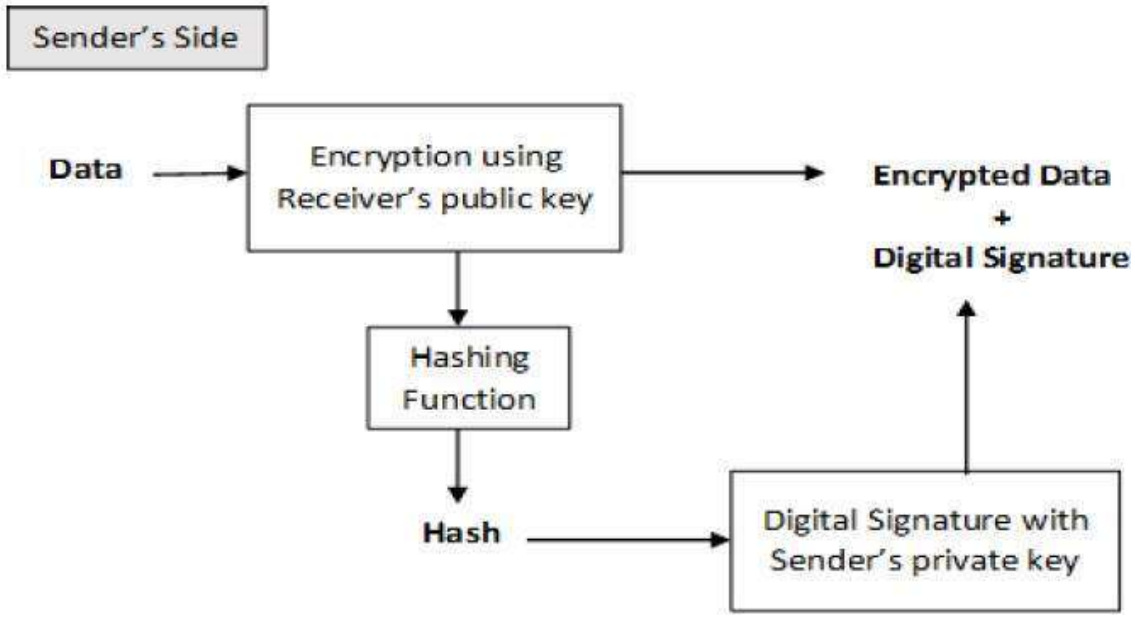
Encryption with Digital Signature

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

This makes it essential for users employing PKC for encryption to seek digital signatures along with encrypted data to be assured of message authentication and non-repudiation.

This can archived by combining digital signatures with encryption scheme. Let us briefly discuss how to achieve this requirement. There are **two possibilities, sign-then-encrypt** and **encrypt-then-sign**.

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



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

Cryptography in Blockchain

One of the important questions that always comes to our mind is How blockchain is secure? and What makes blockchain secure? Blockchain security is built on two concepts Cryptography and Hashing. This article focuses on discussing these two important concepts in detail.

Cryptography in Blockchain

Cryptography is a method of securing data from unauthorized access. In the blockchain, cryptography is used to secure transactions taking place between two nodes in a blockchain network. As discussed above, in a blockchain there are two main concepts cryptography and hashing. Cryptography is used to encrypt messages in a P2P network and hashing is used to secure the block information and the link blocks in a blockchain.

Cryptography primarily focuses on ensuring the security of participants, transactions, and safeguards against double-spending. It helps in securing different transactions on the blockchain network. It ensures that only the individuals for whom the transaction data is intended can obtain, read and process the transaction.

Role of Cryptography in Blockchain

Blockchain is developed with a range of different cryptography concepts. The development of cryptography technology promotes restrictions for the further development of blockchain.

- In the blockchain, cryptography is mainly used to protect user privacy and transaction information and ensure data consistency.
- The core technologies of cryptography include symmetric encryption and asymmetric encryption.

- Asymmetric cryptography uses digital signatures for verification purposes, every transaction recorded to the block is signed by the sender by digital signature and ensures that the data is not corrupted.

Cryptography plays a key role in keeping the public network secure, so making it fit to maintain the integrity and security of blockchain.

Cryptography

Cryptography is a technique or a set of protocols that secure information from any third party during a process of communication. It is also made up of two Greek terms, Kryptos term meaning "hidden" and Graphein, a term meaning "to write". Some terminologies related to Cryptography:

- **Encryption:** Conversion of normal text to a random sequence of bits.
- **Key:** Some amount of information is required to get the information of the cryptographic algorithm.
- **Decryption:** The inverse process of encryption, conversion of a Random sequence of bits to plaintext.
- **Cipher:** The mathematical function, i.e. a cryptographic algorithm which is used to convert plaintext to ciphertext(Random sequence of bits).

Types of Cryptography

The two types of cryptography are:

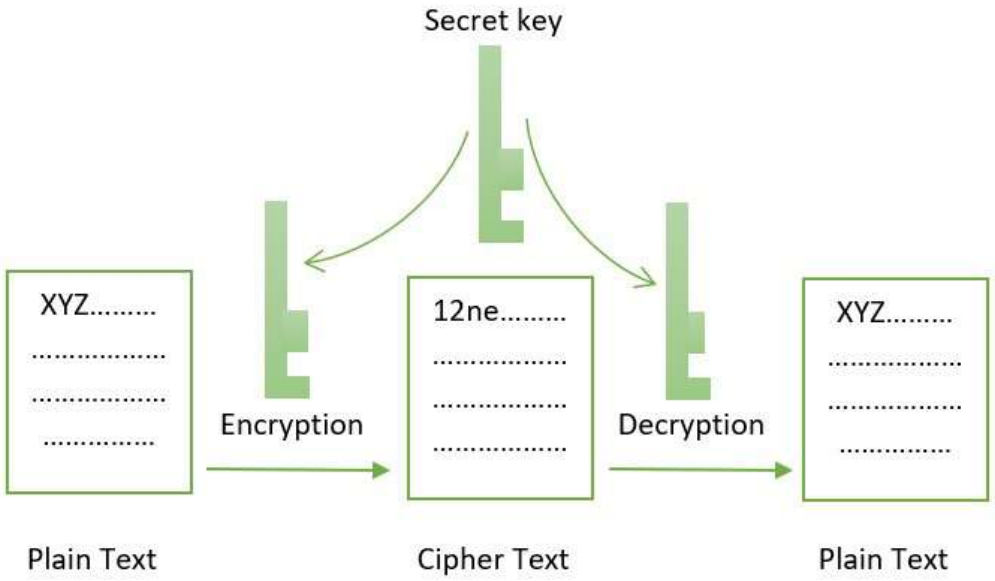
- **Symmetric-key cryptography.**
- **Asymmetric-key cryptography.**

Let's discuss each of these topics in detail.

1. Symmetric-key Encryption: It focuses on a similar key for encryption as well as decryption. Most importantly, the symmetric key encryption method is also applicable to secure website connections or encryption of data. It is also referred to as secret-key cryptography. The only problem is that the sender and receiver exchange keys in a secure manner. The popular symmetric-key cryptography system is Data Encryption System(DES). The cryptographic algorithm utilizes the key in a cipher to encrypt the data and the data must be accessed. A person entrusted with the secret key can decrypt the data. Examples: AES, DES, etc.

Features:

- It is also known as Secret key cryptography.
- Both parties have the same key to keeping secrets.
- It is suited for bulk encryptions.
- It requires less computational power and faster transfer.

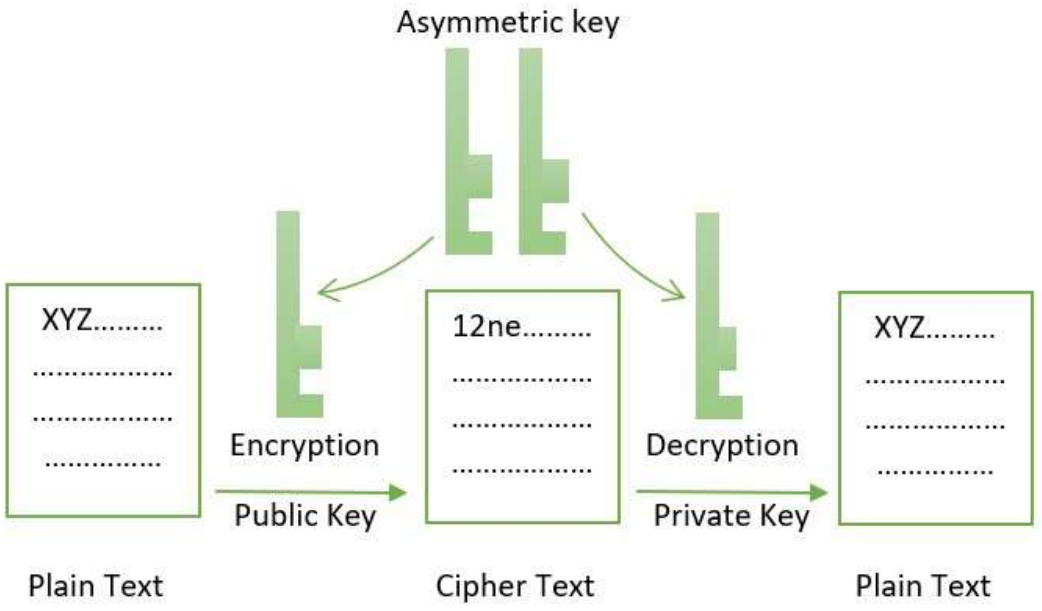


Symmetric Cryptography

2. Asymmetric-key Encryption: This cryptographic method uses different keys for the encryption and decryption process. This encryption method uses public and private key methods. This public key method help completely unknown parties to share information between them like email id. private key helps to decrypt the messages and it also helps in the verification of the digital signature. The mathematical relation between the keys is that the private key cannot be derived from the public key, but the public key can be derived from the private key. **Example:** ECC,DSS etc.

Features:

- It is also known as Public-key cryptography.
- It is often used for sharing secret keys of symmetric cryptography.
- It requires a long processing time for execution.
- Plays a significant role in website server authenticity.



Asymmetric Cryptography

Wallets And Digital Signatures

A blockchain wallet is a special software or a hardware device that is used to keep the transaction information and personal information of the user. Blockchain wallets do not contain the actual currency. The wallets are used to keep private keys and maintain a transaction balance.
Wallets are only a communication tool to communicate to carry out transactions with other users. The real data or currency is stored in blocks in the blockchain.

Digital signatures are like proofs that the user gives to the recipient and other nodes in the network to prove that it is a legitimate node in the network to carry out transactions. While initiating a transaction with other nodes in the blockchain network, the user first has to create a unique digital signature by combining the transaction data with the user's private key using a special algorithm. This process will guarantee the authenticity of the node and the integrity of the data.

Cryptography Hash Function in Blockchain

One of the most notable uses of cryptography is cryptographic hashing. Hashing enables immutability in the blockchain. The encryption in cryptographic hashing does not involve any use of keys. When a transaction is verified hash algorithm adds the hash to the block, and a new unique hash is added to the block from the original transaction. Hashing continues to combine or make new hashes, but the original footprint is still accessible. The single combined hash is called the root hash. Hash Function helps in linking the block as well as maintaining the integrity of data inside the block and any alteration in the block data leads to a break of the blockchain. Some commonly used hashed function is MD5 and SHA-1.

Properties of Cryptographic Hash:

- For a particular message hash function does not change.
- Every minor change in data will result in a change in a major change in the hash value.

- The input value cannot be guessed from the output hash function.
- They are fast and efficient as they largely rely on bitwise operations.

Benefits of Hash function in Blockchain:

1. Reduce the bandwidth of the transaction.
2. Prevent the modification in the data block.
3. Make verification of the transaction easier.

Use of Cryptographic Hash Functions

As the blockchain is also public to everyone it is important to secure data in the blockchain and keeps the data of the user safe from malicious hands. So, this can be achieved easily by cryptography.

- When the transaction is verified through a hash algorithm, it is added to the blockchain, and as the transaction becomes confirmed it is added to the network making a chain of blocks.
- Cryptography uses mathematical codes, it ensures the users to whom the data is intended can obtain it for reading and processing the transaction.
- Many new tools related to the application of cryptography in blockchain have emerged over the years with diverse functionalities.

Benefits of Cryptography in Blockchain

There are a huge number of benefits of cryptography in blockchain some of them are stated below:

- **Encryption:** Cryptography uses asymmetric encryption to ensure that the transaction on their network guards the information and communication against unauthorized revelation and access to information.
- **Immutability:** This feature of cryptography makes it important for blockchain and makes it possible for blocks to get securely linked by other blocks and also to ensure the reliability of data stored in the blockchain, it also ensures that no attacker can derive a valid signature for unposed queries from previous queries and their corresponding signatures.
- **Security:** Cryptography makes the records of transactions easier using encryption of data, and accessing of data using public and private keys. Cryptographic hashing tampering with data is not possible, making blockchain more secure.
- **Scalability:** Cryptography makes the transaction irreversible giving the assurance that all users can rely on the accuracy of the digital ledger. It allows limitless transactions to be recorded securely in the network.
- **Non-repudiation:** The digital signature provides the non-repudiation service to guard against any denial of a message passed by the sender. This benefit can be associated with collision resistance i.e.; since every input value has a unique hash function so there is no clash between the messages that are sent and one message can be easily differentiated from the other.
- **Prevent hackers:** The digital signature prevents hackers from altering the data because if the data changes, the digital signature becomes invalid. With the help of cryptography, it protects the data from hackers and makes cryptography in blockchain unstoppable.

Limitations of Cryptography in Blockchain

Below are some of the limitations of cryptography in the blockchain:

- **Information difficult to access:** Strongly encrypted and digitally signed information can be difficult to access even for a legitimate user at the most critical time of decision-making. The network can be attacked and rendered non-functional by an intruder.
- **High availability:** It is one of the fundamental aspects of information security, and cannot be ensured through the use of cryptography. Other methods are needed to guard against the threats such as denial of service or complete breakdown of the information systems.
- **No protection against vulnerabilities:** Cryptography does not guard against the vulnerabilities and threats that emerge from the poor design of protocols, procedures, and systems. These issues need to be fixed with the proper design of the defense infrastructure.
- **Expensive:** Cryptography needs huge time and money investments. Public key cryptography needs setting up and maintenance of public key infrastructure which requires huge investment. Addition of cryptographic techniques while sending messages and information processing adds to the delay.
- **Vulnerability:** The security of cryptographic techniques depends on the complexity and difficulty of the mathematical problem. Any breakthrough in solving such mathematical problems can make cryptographic techniques vulnerable.

What is Bitcoin?

There are a number of currencies in this world used for trading amenities. Rupee, Dollar, Pound Euro, and Yen are some of them. These are printed currencies and coins and you might be having one of these in your wallet. But bitcoin is a currency you can not touch, you can not see but you can efficiently use it to trade amenities. It is an electronically stored currency. It can be stored in your mobiles, computers, or any storage media as a virtual currency.

Bitcoin is an innovative digital payment system. It is an example of a cryptocurrency and the next big thing in finance.

- It is a virtual currency designed to act as money and outside the control of any person or group thus eliminating the need for third-party in financial transactions.
- It is used as a reward for the miners in bitcoin mining.
- It can be purchased on several exchanges.

There are 3 ways you can get a bitcoin in your electronic storage:

1. **Trade Money For Bitcoin:** Say that the value of a bitcoin is 1 lakh rupees, so if you want a bitcoin, you can trade a bitcoin in place of 1 lakh rupees. This Bitcoin will further be stored in your electronic storage media which you can further use.
2. **Trade Goods For Bitcoin:** Say that the value of a bitcoin is 1 lakh rupees and you have a commodity that has its value as 1 lakh rupees, so you can trade that commodity in place of a bitcoin, and the bitcoin will be stored in your electronic storage media.

3. **Mine Bitcoins:** Other than trading, you can also mine bitcoins. Since it is a decentralized currency, there is no authority that brings bitcoins into the market. Bitcoins only come into the market by mining them.

Brief History:

"Satoshi Nakamoto" is presumed to be the pen name for the person or people who designed the original bitcoin. Bitcoin was first introduced in the year 2009 as a medium of exchange. Bitcoin then started as a peer-to-peer network to generate a system for electronic transactions. Since then, there has been a rapid growth in the usage as well as the value of bitcoin which is a popular system of digital currency.

Features:

- **Distributed:** All bitcoin transactions are recorded in a public ledger known as the blockchain. There are nodes in the network that maintain copies of the ledger and contribute to the correct propagation of the transactions following the rules of the protocols making it impossible for the network to suffer downtime.
- **Decentralized:** There is no third party or no CEO who controls the bitcoin network. The network consists of willing participants who agree to the rules of a protocol and changes to the protocol are done by the consensus of its users. This makes bitcoin a quasi-political system.
- **Transparent:** The addition of new transactions to the blockchain ledger and the state of the bitcoin network is arrived upon by consensus in a transparent manner according to the rules of the protocol.
- **Peer-to-peer:** In Bitcoin transactions, the payments go straight from one party to another party so there is no need for any third party to act as an intermediary.
- **Censorship resistant:** As bitcoin transactions are pseudo-anonymous and users possess the keys to their bitcoin holdings, so it is difficult for the authorities to ban users from using their assets. This provides economic freedom to the users.
- **Public:** All bitcoin transactions are available publicly for everyone to see. All the transactions are recorded, which eliminates the possibility of fraudulent transactions.
- **Permissionless:** Bitcoin is completely open access and ready to use for everyone, there are no complicated rules of entry. Any transaction that follows the set algorithm will be processed with certainty.
- **Pseudo-anonymous:** Bitcoin transactions are tied to addresses that take the form of randomly generated alphanumeric strings.

Value of Bitcoin

A normal piece of paper and a currency note is physically the same but the value of the note is decided by an authority or a centralized government. But Bitcoin is a currency that does not have any centralized government or authority to control and decide its value. It is a decentralized digital currency.

- As of now, the value of 1 bitcoin is 23,54,953.68 Indian Rupees but this value fluctuates as there is no centralized authority to control it.

- In December 2011, the value of a Bitcoin was estimated to be 2 US dollars, in December 2013, it went up to about 1000 US dollars.

How Do Bitcoin Transactions Work?

Bitcoin transactions are digitally signed for security. Everyone on the network gets to know about a transaction. Anyone can create a bitcoin wallet by downloading the bitcoin program. Each bitcoin wallet has two things:

- **Public key:** It is like an address or an account number via which any user or account can receive bitcoins.
- **Private key:** It is like a digital signature via which anyone can send bitcoins.

The public key can be shared with anyone but the private key must be held by the owner. If the private key gets hacked or stolen then bitcoin gets lost.

A bitcoin transaction contains three pieces of information:

- **Private key:** The first part contains the bitcoin wallet address of the sender i.e. the private key.
- **Amount of bitcoin to be transferred:** The second part contains the amount that has been sent.
- **Public key:** The third part contains the bitcoin wallet address of the recipient i.e. the public key.

Bitcoin transactions are verified by the nodes on the network. Once the transaction is verified and executed successfully, the transaction is recorded in a distributed public ledger called a blockchain. A bitcoin can also be considered as an invisible currency with only the transaction records between different addresses.

How Do Bitcoins Come Into Market?

Bitcoins are a decentralized currency, they aren't printed, like rupees, they're produced by people, and big companies, running computers all around the world, using software that solves mathematical problems.

- Bitcoins are mined using the computing power of the distributed network. This network also processes transactions made using Bitcoin.
- Bitcoins are mined on the basis of computing power, so they take time to be generated.
- To keep it valuable, it has been stated that only 21 million bitcoins can be created by miners. By the year 2140, all the bitcoins will be created.
- Around the world, thousands of computers with very high computing power are processing transactions and securing the network by solving complex mathematical calculations and collecting new bitcoins in exchange.

How Does Bitcoin Mining Work?

In the Bitcoin network, there are nodes that use the computing power of their CPU to process the transactions. The following are the steps followed while mining a bitcoin:

- The user initiates the bitcoin transaction by listing the details like the number of bitcoins to be sent, and the public address, and affixing the private key to generate a digital signature. The encrypted information to the miners present on the network.
- The miners will verify the transaction to check whether there is sufficient balance to carry out the transaction.
- The faster the CPU of a miner, the greater the chances for the miner to get rewarded for verifying the transaction. The miner's job is only to provide the CPU, there is no manual intervention from the miner. The bitcoin program will run automatically on the system.
- Once the transaction is verified, the number of transactions is broadcasted to the network of miners who can copy or download the block.
- These blocks through timestamps are stored in sequential order to form a blockchain.
- Each miner in the network must have an updated copy of the blockchain ledger in order to earn bitcoins.

How Do You Buy Bitcoin?

There are three ways to get a bitcoin:

1. Buying: Many marketplaces like Bitcoin exchanges allow users to buy or sell bitcoins using different currencies. If one does not want to mine a bitcoin, it can be bought using a cryptocurrency exchange. Most people will not be able to purchase the entire BTC due to its price, so it is possible to buy portions of BTC on these exchanges in fiat currency like U.S. Dollars. The following steps can be followed to buy bitcoin outside the online exchanges:

- Each person who joins the bitcoin network is issued a public key and a private key.
- When a person buys a bitcoin or sends/receives it, the person will receive a public key.
- The person can only access the bitcoin using the private key (it has) with the public key (it received).

2. Mining: People on the bitcoin network compete among themselves to mine bitcoins using computers to solve complex maths puzzles. This is how bitcoins are created.

3. Transfer: Bitcoin can be transferred from one account to another just like digital cash using mobile applications or computers.

How is Bitcoin Used?

Below are some of the ways of using bitcoin:

- **Payment:** Bitcoin is accepted as a mode of payment for goods and services at many merchants, and retailers. To use bitcoin, wallets are required. cryptocurrency wallets contain private keys to the bitcoin, which need to be entered while conducting a transaction.
- **Investing: portfolio:** Bitcoin grew in popularity which made Investors and Individuals interested in investing in the cryptocurrency Bitcoin. Individuals can invest in Bitcoin to help diversify their portfolio of stocks and bonds.

Benefits of Bitcoin

The following are some of the advantages of using bitcoins:

- **User anonymity:** Bitcoin users can have multiple public keys and are identified by numerical codes. This ensures that the transactions cannot be traced back to the user. Even if the wallet address becomes public, the user can generate a new wallet address to keep information safe.
- **Transparency:** Bitcoin transactions are recorded on the public ledger blockchain. The transactions are permanently viewable, which gives transparency to the system but they are secure and fraud-resistant at the same time due to blockchain technology.
- **Accessibility:** Bitcoin is a very versatile and accessible currency. It takes a few minutes to transfer bitcoins to another user, so it can be used to buy goods and services from a variety of places accepting bitcoins. This makes spending bitcoin easy in another country with little or no fees applied.
- **Independence from central authority:** Bitcoin is a decentralized currency, which means there is no dependence on any single governing authority for verifying transactions. This means that the authorities are not likely to freeze or demand back the bitcoins.
- **Low transaction fees:** Standard wire transfers involve transaction fees and exchange costs. Since bitcoin transactions do not involve any government authority so the transaction fees are very low compared to bank transfers.

Drawbacks of Bitcoin

The following are some of the cons of using bitcoin:

- **Volatility:** There are various factors that contribute to the bitcoin's volatility like uncertainty about its future value, security breaches, headline-making news, and one of the most important reasons is the scarcity of bitcoins. It is known that there is a limit of 21 million bitcoins that could ever exist which is why some regard bitcoin as a scarce resource. This scarcity makes bitcoin's price variable.
- **No government regulations:** Unlike the investments that are done through central banks, bitcoins transactions are not regulated by any central authority due to a decentralized framework. This means that bitcoin's transactions don't come with legal protection and are irreversible which makes them susceptible to crimes.
- **No buyer protection:** If the goods are bought using bitcoins and the seller does not send the promised goods then nothing can be done to reverse the transactions and since there is no central authority so no legal protection can be provided in this case.
- **Not widely accepted:** Bitcoins are still only accepted by a small group of online merchants. This makes it unfeasible to rely completely on bitcoin as a currency and replace it with traditional bank transactions.
- **Irreversible:** There is a lack of security in bitcoin transactions due to the anonymous and non-regulated nature of the bitcoin transactions. If the wrong amount is sent or the amount is sent to the wrong recipient then nothing can be done to reverse the transactions.

Blockchain Hash Function

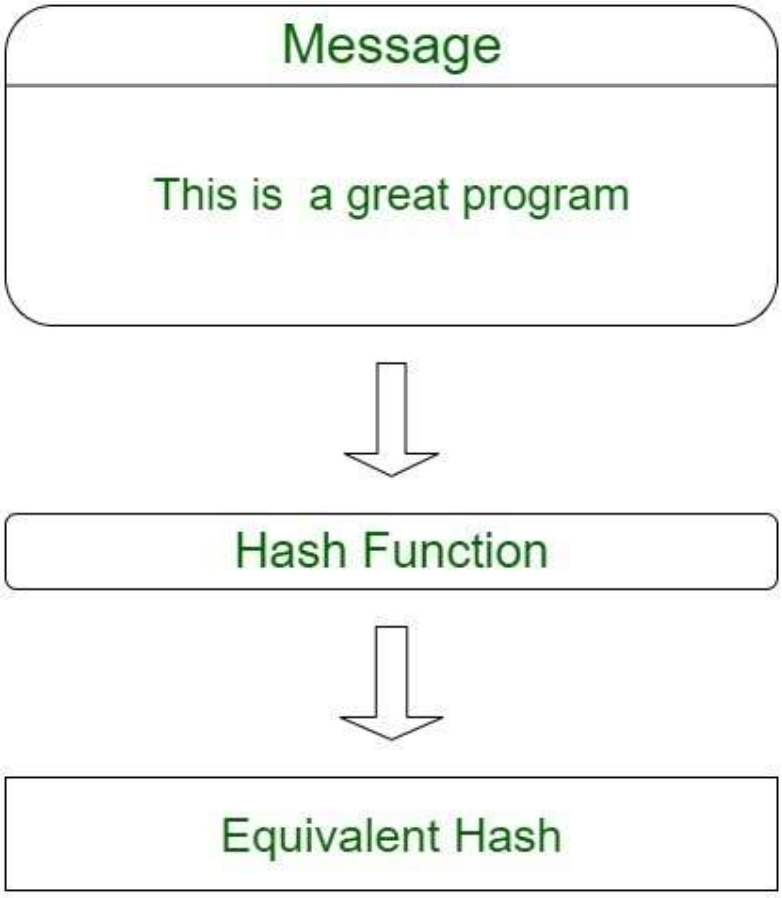
A hash function is a mathematical function that takes an input string of any length and converts it to a fixed-length output string. The fixed-length output is known as the hash value. To be cryptographically secure and useful, a hash function should have the following properties:

- **Collision resistant:** Give two messages m1 and m2, it is difficult to find a hash value such that hash(k, m1) = hash(k, m2) where k is the key value.
- **Preimage resistance:** Given a hash value h, it is difficult to find a message m such that h = hash(k, m).
- **Second preimage resistance:** Given a message m1, it is difficult to find another message m2 such that hash(k, m1) = hash(k, m2).
- **Large output space:** The only way to find a hash collision is via a brute force search, which requires checking as many inputs as the hash function has possible outputs.
- **Deterministic:** A hash function must be deterministic, which means that for any given input a hash function must always give the same result.
- **Avalanche Effect:** This means for a small change in the input, the output will change significantly.
- **Puzzle Friendliness:** This means even if one gets to know the first 200 bytes, one cannot guess or determine the next 56 bytes.
- **Fixed-length Mapping:** For any input of fixed length, the hash function will always generate the output of the same length.

How do Hash Functions work?

The hash function takes the input of variable lengths and returns outputs of fixed lengths. In cryptographic hash functions, the transactions are taken as inputs and the hash algorithm gives an output of a fixed size.

The below diagram shows how hashes work.



Types of Cryptographic Hash Functions

There are several different classes of hash functions. Some of the popular classes are:

1. RACE Integrity Primitives Evaluation Message Digest (RIPEMD): This set includes RIPEMD, RIPEMD-128, RIPEMD-160, RIPEMD-256, RIPEMD-320.

- Out of these RIPEMD-160 is the most common.
- The original RIPEMD-128 is based on the design principles used in MD4.
- The RIPEMD-256 and 320 have fewer chances of the accidental collision but do not have higher levels of security as compared to RIPEMD-128 and RIPEMD-160.

2. Message-Digest Algorithm: This family comprises hash functions MD2, MD4, MD5, and MD6.

- MD5 is the most widely used cryptographic hash function.
- It is used to generate a 128-bit digest from a 512-bit string broken down into 16 words composed of 32 bits each.
- Ronald Rivest designed this algorithm in 1991 to use for digital signature verification.
- These are no longer considered cryptographically secure methods and should not be used for cryptographic authentication.

3. BLAKE2: It was announced on December 21, 2012.BLAKE2 is a cryptographic hash function based on BLAKE, designed with the aim to replace MD5 and SHA-1 algorithms in applications requiring high performance in software. It provides better security than SHA-2 and is similar to that of SHS-3. It provides the following features:

- It provides immunity to length extensions.
- It removes the addition of constants to message words.
- It simplifies padding and reduces the number of rounds from 16 to 12.

4. BLAKE3: It was announced on January 9, 2020. BLAKE3 is a cryptographic function based on Bao and BLAKE2. It is a few times faster than BLAKE2. This algorithm provides many features like parallelism, XOF, KDF, etc.

5. Whirlpool: It is a cryptographic hash function, first described in 2000. It is a modified version of the Advanced Encryption Standard (AES). Whirlpool produces a hash of 512 bits.

6. Secure Hashing Algorithm: The family of SHA comprises four SHA algorithms: SHA-0, SHA-1, SHA-2, and SHA-3.

- SHA-0 is a 160-bit hash function that was published by the National Institute of Standards and Technology in 1993.
- SHA-1 was designed in 1995 to correct the weaknesses of SHA-0. In 2005, a method was found to uncover collisions in the SHA-1 algorithm due to which long-term employability became doubtful.
- SHA-2 has the following SHA variants, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256. It is a stronger hash function and it still follows the design of SHA-1.
- In 2012, the Keccak algorithm was chosen as the new SHA-3 standard.
- SHA-256 is the most famous of all cryptographic hash functions because it's used extensively in blockchain technology. The SHA-256 Hashing algorithm was developed by the National Security Agency (NSA) in 2001.

Uses of Hash Functions in Blockchain

The blockchain has a number of different uses for hash functions. Some of the most common uses of the hash function in blockchain are:

- **Merkle Tree:** This uses hash functions to make sure that it is infeasible to find two Merkle trees with the same root hash. This helps to protect the integrity of the block header by storing the root hash within the block header and thus protecting the integrity of the transactions.
- **Proof of Work Consensus:** This algorithm defines a valid block as the one whose block header has a hash value less than the threshold value.
- **Digital signatures:** Hash functions are the vital part of digital signatures that ensures data integrity and are used for authentication for blockchain transactions.
- **The chain of blocks:** Each block header in a block in the blockchain contains the hash of the previous block header. This ensures that it is not possible to change even a single block in a

blockchain without being detected. As modifying one block requires generating new versions of every following block, thus increasing the difficulty.

Blockchain - Block Hashing

Blockchain is the backbone Technology of Digital CryptoCurrency BitCoin. The blockchain is a distributed database of records of all transactions or digital events that have been executed and shared among participating parties. Each transaction is verified by the majority of participants of the system. It contains every single record of each transaction. In order to understand how blockchain works, it's very important to understand in detail the concept of Hashing.

There are three main parts of any blockchain network:

- **Node/ Block:** It is the main building block of any blockchain. It acts as a database to store the information related to all transactions. The size, period, and triggering event for blocks are different for every blockchain. Each block or node contains a complete record of all the transactions that were ever recorded in that blockchain.
- **Network:** The network is composed of "full nodes". Think of them as the computer running an algorithm that is securing the network.
- **Hash:** It acts as a chain that links one block to another, mathematically one can say that it's "chaining" all blocks together. This is one of the most difficult concepts in blockchain to comprehend. It's like magic that glues blockchains together and allows them to create mathematical trust and maintain privacy as well as security in the network. The hash in the blockchain is created from the data that was in the previous block. So we can say that the hash is a fingerprint of this data and locks blocks in order and time.

Hashing In Blockchain:

In simple words, hashing refers to the process of having an input of any length but showing an output item of a fixed length**.** If we take the example of blockchain use in online transactions(using bitcoins), transactions of different lengths are run through a given hashing algorithm, and all give an output that is of a fixed length. This output is independent of the length of the input transaction.

- **Secure Hashing Algorithm 256 (SHA-256):** This hashing algorithm always gives an output of fixed length 256-bits or 32 bytes, no matter whatever is the size of the input transaction. It means if we hash two different input using SHA-256, let's say one is a movie of 1 gigabyte and another is an image of 5 kilobytes, then in both cases, the output hash will be 256-bits in length. The only difference between the two will be the hash pattern. Currently, this algorithm is used in the Bitcoin network.
- **Keccak-256:** This hashing algorithm always gives an output of fixed length 256-bit; currently it is used in the Ethereum network.

Hash Functions:

Basically, the process of using a given hash function to produce a transaction is called hashing. A hash function, will take any transaction or data input and rehash it to produce an output of a fixed size. The transaction output of that given hash function is what we call a hash.

	H(X)		H(x)	
INPUT DATA --->	HASH FUNCTION	---->HASH---->	HASH FUNCTION	----> OUTPUT

This complete process is known as Hashing. It consists of two subparts, one is encryption of data (the process of generating a hash from input data) and the second one is the decryption of data (the process of generating output from hash using a cryptographic hash function).

Properties of Hash Function: Three main properties of cryptographic hash functions are:

- Its Input can be any string of any size.
- It produces a fixed size output. We had already seen the example of SHA-256 regarding this property.
- And the third one is, It is efficiently Computable**,** this means that for a given input string, we can figure out what the output of the hash function is in a reasonable amount of time. If I talk more technically, computing the hash of an n-bit string has a running time that is O(N).

For a hash function to be strong and more secure, it has the following three additional properties:

- **Collision-resistance:** A collision occurs when two distinct inputs produce the same output. A hash function H(x) is collision-resistant if nobody can find a collision. It means for two values x and y, such that $x \neq y$, if h(x) =H(y), it means the function h(.) is collision-resistant. We want to increase data security then it's very important to use a strong hash function that is not collision-resistant.
- **Hiding:** This property of hash functions ensures that if we're given the output of hash function y=H(x), then there's no feasible way to figure out what the input, x, was. In a more technical and mathematical way "A hash function H is hiding if: when a secret value r is chosen from a probability distribution that has high min-entropy, then given H(r // x) it is infeasible to find x.
- **Puzzle friendliness:** This property of the hash function is a bit complicated to understand, but I will try to make you understand in simple words. According to this property, if someone wants to target the hash function to come out to some particular output value y, that if there's part of the input that is chosen in a suitably randomized way, it's very difficult to find another value that hits exactly that target. In a more technical way, this property states that " A hash function H is said to be a puzzle - friendly if for every possible n-bit output value y, if k is chosen from a distribution with high min-entropy, then it is infeasible to find x such that H(k // x)=y in time significantly less than 2^n.

Applications:

1. Signature generation and verification: Almost all digital signature schemes require a cryptographic hash to be calculated over the message. The message is considered authentic if the signature verification succeeds given the signature and recalculated hash digest over the message. So the message integrity property of the cryptographic hash is used to create secure and efficient digital signature schemes.

2. Password verification: To authenticate a user, the password presented by the user is hashed and compared with the stored hash. A password hashing is performed; the original password cannot be recalculated from the stored hash value.

3. Verification of files and messages: An important application of secure hashes is the verification of **message integrity.** Comparing messages digests(hash digest over the message) calculated before, and after, transmission can determine whether any changes have been made to the message or file.

Conclusion: The cryptographic hash function is an integral part of blockchain innovation. It is essentially a feature that gives security capabilities to the processed transactions, making them immutable. Hashing is also at the center of "Merkle Trees", which is an advanced approach to blockchain hashing. It is useful in issues of scalability, and mobile/ light wallets.

Blockchain and Distributed Ledger Technology (DLT)

Last Updated : 23 Jul, 2025

-
-
-

A blockchain is a digital ledger of transactions distributed across the entire network of computers (or nodes) on the blockchain. Distributed ledgers use independent nodes to record, share, and synchronize transactions in their respective electronic ledgers instead of keeping them in one centralized server. A blockchain uses several technologies like digital signatures, distributed networks, encryption/ decryption methods, and distributed ledger technology to enable blockchain applications. Blockchain is one of the types of DLT in which transactions are recorded with an unchangeable cryptographic signature called a hash. That is why distributed ledgers are often called blockchains.

What is Distributed Ledger Technology (DLT)?

Distributed Ledger Technology (DLT) is centered around an encoded and distributed database where records regarding transactions are stored. A distributed ledger is a database spread across various computers, nodes, institutions, or countries and accessible by multiple people around the globe.

Key Features:

1. **Decentralized:** It is a decentralized technology and every node will maintain the ledger, and if any data changes happen, the ledger will get updated. The process of updating takes place independently at each node. Even small updates or changes made to the ledger are reflected and the history of that change is sent to all participants in a matter of seconds.
2. **Immutable:** Distributed ledger uses cryptography to create a secure database in which data once stored cannot be altered or changed.
3. **Append only:** Distributed ledgers are append-only in comparison to the traditional database where data can be altered.
4. **Distributed:** In this technology, there is no central server or authority managing the database, which makes the technology transparent. To counter the weaknesses of having one ledger to rule all, So that there is no one authoritative copy and have specific rules

around changing them. This would make the system much more transparent and will make it a more decentralized authority. In this process, every node or contributor of the ledger will try to verify the transactions with the various consensus algorithms or voting. the voting or participation of all the nodes depends on the rules of that ledger. In the case of bitcoin, the Proof of Work consensus mechanism is used for the participation of each node.

5. **Shared:** The distributed ledger is not associated with any single entity. It is shared among the nodes on the network where some nodes have a full copy of the ledger while some nodes have only the necessary information that is required to make them functional and efficient.
6. **Smart Contracts:** Distributed ledgers can be programmed to execute smart contracts, which are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. This allows for transactions to be automated, secure, and transparent.
7. **Fault Tolerance:** Distributed ledgers are highly fault-tolerant because of their decentralized nature. If one node or participant fails, the data remains available on other nodes.
8. **Transparency:** Distributed ledgers are transparent because every participant can see the transactions that occur on the ledger. This transparency helps in creating trust among the participants.
9. **Efficiency:** The distributed nature of ledgers makes them highly efficient. Transactions can be processed and settled in a matter of seconds, making them much faster than traditional methods.
10. **Security:** Distributed ledgers are highly secure because of their cryptographic nature. Every transaction is recorded with a cryptographic signature that ensures that it cannot be altered. This makes the technology highly secure and resistant to fraud.

How DLT Can Replace Traditional Book-Keeping Methods?

1. **Decentralization:** Unlike centralized systems, DLT operates on a network where multiple participants maintain synchronized copies of the ledger, reducing the risk of data manipulation and single points of failure.
2. **Real-time Updates:** DLT enables immediate transaction recording and updates across all network nodes, enhancing accuracy and providing real-time visibility into financial status.
3. **Enhanced Transparency:** Transactions on a DLT are visible to authorized participants, fostering trust and reducing the need for extensive audits.
4. **Improved Security:** DLT utilizes cryptographic techniques to secure data, creating an immutable chain of transactions that is resistant to tampering and cyberattacks.
5. **Cost Reduction:** By automating processes and minimizing the need for intermediaries, DLT can lower operational costs and improve efficiency.
6. **Streamlined Processes:** DLT enables automation through smart contracts, reducing manual intervention and errors in transaction processing.
7. **Secure Audit Trail:** DLT automatically creates a secure and immutable record of all transactions, simplifying audits and enhancing accountability.

8. **Interoperability:** DLT facilitates seamless data sharing and collaboration across different accounting systems, reducing data silos.

Types of Distributed Ledger Technology

1. **Blockchain:** In this type of DLT, transactions are stored in the form chain of blocks and each block produces a unique hash that can be used as proof of valid transactions. Each node has a copy of the ledger which makes it more transparent.
2. **Directed Acyclic Graphs (DAG):** This uses a different data structure to organize the data that brings more consensus. In this type of DLT, validation of transactions mostly requires the majority of support from the nodes in the network. Every node on the network has to provide proof of transactions on the ledger and then can initiate transactions. In this nodes have to verify at least two of the previous transactions on the ledger to confirm their transaction.
3. **Hashgraph:** In this type of DLT, records are stored in the form of a directed acyclic graph. It uses a different consensus mechanism, using virtual voting as the form consensus mechanism for gaining network consensus. Hence nodes do not have to validate each transaction on the network.
4. **Holochain:** Holochain is termed as the next level of blockchain by some people because it is much more decentralized than blockchain. It is a type of DLT that simply proposes that each node will run on a chain of its own. Therefore nodes or miners have the freedom to operate autonomously. It basically moves to the agent-centric structure. Here agent means computer, node, miner,etc.
5. **Tempo or Radix:** Tempo uses the method of making a partition of the ledger this is termed sharding and then all the events that happened in the network are ordered properly. Basically, transactions are added to the ledger on basis of the order of events than the timestamp.

Types of DLT

Advantages Of Distributed Ledger Technology

1. **High Transparency:** Distributed ledger presents a high level of transparency because all the transaction records are visible to everyone. The addition of data needs to be validated by nodes by using various consensus mechanisms. and if anyone tries to alter or change data in the ledger then it is immediately reflected across all nodes of the network which prevents invalid transactions.
2. **Decentralized:** In a centralized network, there may be a single point of failure and it can disrupt the whole network because of mistakes at the central authority level. But in the case of distributed networks, there is no risk of a single point of failure. because of the decentralized structure trust factor also increases in participating nodes. This decentralized nature of validation reduces the cost of transactions drastically.
3. **Time Efficient:** As this network is decentralized so there is no need for a central authority to validate transactions every time. Hence this time for validation of each transaction reduces drastically. In the case of DLT, transactions can be validated by members of the network itself by using various consensus mechanisms.

4. **Scalable:** Distributed ledger technology is more scalable because many different types of consensus mechanisms can be used to make it more reliant, fast, and updated. Because these many advanced DLT technologies are introduced in the last few years. Such as Holochain, hashgraph are considered to be advanced and more secure versions of Blockchain DLT. Blockchain itself is advanced and secure but DLT provides a way to more advanced technologies.

Applications of Distributed Ledger Technology

Because of all these benefits of distributed ledger technology and this technology has the potential to revolutionize many sectors like Financial, energy, healthcare, governance, supply chain management, real estate, cloud computing, etc.

Applications of DLT

1. **Banking**: In the banking sector right now transfer of money can be both expensive and time-consuming. Also sending money overseas becomes even more complex due to exchange rates and other hidden fees included. Here DLT can provide a decentralized secure network that will help to reduce the time, complexity, and costs required to transfer money. This decentralized network will eliminate the need for third parties which makes this system more complex and time-consuming.

2. **Cyber Security:** Nowadays cyber security has been emerging as a big threat to governments, enterprises, and individual people also. So it is essential to find an effective solution to secure our data and privacy against unauthorized access. In DLT, all information is authorized and securely encrypted by various cryptographic algorithms. This provides a transparent and secure environment and none of the data can be tempered by any entity.

3. **Supply chain management**: Supply chain is one of the complex structures itself. In this structure, it is hard to trace where the fault happened. So here Distributed ledger technology comes into the picture, Using DLT, you can easily trace the supply chain from the beginning to the end and can easily find out where a mistake or fault has happened. All the data added to the DLT is validated and permanent and can not be altered. This transparency of data enables us to trace from the beginning to the end of the ledger.

4. **Healthcare:** Distributed Ledger eliminates central authority and ensures rapid access to secured and untempered data. Here important medical can be stored securely and no one can change this data, even if someone tries to change it will be reflected everyone immediately. DLT can be used in the insurance sector to trace false claims because of its decentralized system.

5. **Governance:** DLT can be used in the government system to make it transparent among citizens. Many governments have adopted blockchain in the governance system because of the robustness of this system. It can be used as a voting system too. The traditional voting system has many flaws and sometimes it is found that there are many false voting and illegal activities that happen during voting. Online voting systems can be used to vote and with security and fake votes can be easily checked. everyone will have their own identity. So that any person sitting anywhere in the world can cast his vote.

How are Blockchain And Distributed Ledger Different?

In general blockchain and Distributed Ledger Technology are considered as same, but there are some differences between these two technologies. Blockchain can be classified as a type of Distributed Ledger Technology. We can say that Blockchain is a type of DLT, but every Distributed Ledger can not be called a blockchain.

Blockchain is the parent technology of DLT. But the idea behind them is the same. Blockchain technology has the potential to solve many problems in the banking and financial industry. Here, blockchain is the advanced version of Distributed Ledger Technology with many useful functionalities. Developers have many other variants of DLTs in the technology world. However, they do not have the many real-life implementations and applications that blockchain has been able to do.

Aspect	Distributed Ledger	Blockchain Technology
Block Structure	In DLT, blocks can be organized in different forms.	In Blockchain, blocks are added in the form of a chain.
Power of Work	It is more scalable because it does not need the power of a work consensus mechanism for the validation of each transaction.	It is a subset of DLT, the power of the work consensus mechanism adds more functionalities and security.
Tokens	It does not require any tokens or digital currency.	In it, tokens must be considered while working with Blockchain.
Sequence	It does not require any specific sequence of data.	All blocks are arranged in a particular series.
Trustability	Trust among participating nodes is high.	Trust among participating nodes is less than DLT. Decision-making powers can be on one hand because everyone can mine.

Advantages of Using Distributed Ledger Technology In Blockchain

1. **Security:** All records of every transaction are securely encrypted. Once the transaction is validated, it is completely secure and no one can update or change it. It is a permanent process.
2. **Decentralization:** All network members or nodes have a copy of the ledger for complete transparency. A decentralized private distributed network improves the reliability of the system and gives assurance of continuous operations without any interruption. It gives control of information and data in the hand of the user.

3. **Anonymity:** The identity of each participant is anonymous and does not possibly reveal their identity.
4. **Immutable:** Any validated transactions can not be changed as they are irreversible.
5. **Transparency:** Distributed technologies offer a high level of transparency. Which is necessary for the sectors like finance, medical science, banking, etc.
6. **Speed:** Distributed Ledger Technology can handle large transactions faster than traditional methods.
7. **Smart Contracts:** Distributed Ledger Technology supports smart contracts which are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. Smart contracts reduce the need for intermediaries and offer transparency and automation in the execution of the contract terms.
8. **Lower Costs:** Distributed Ledger Technology eliminates intermediaries and reduces the costs associated with intermediaries, which makes the system more cost-effective.
9. **Improved Efficiency:** Distributed Ledger Technology reduces the time and costs associated with traditional transaction methods. It offers faster settlement times, reduced paperwork, and increased efficiency.
10. **Auditing:** Distributed Ledger Technology makes auditing easier as every transaction is recorded and the ledger cannot be altered. This improves the transparency and accuracy of financial audits.
11. **Resilience:** Distributed Ledger Technology is more resilient than traditional databases as it is spread across multiple nodes. This means that even if one node goes down, the network can still function as the rest of the nodes can continue to validate transactions.
12. **Traceability:** Distributed Ledger Technology offers complete traceability of assets, from their creation to their current ownership. This improves accountability and reduces the risks of fraud and theft.

Disadvantages Of Distributed Ledger Technology

1. **51% Attack:** The 51% attack is a bit concerning part of this distributed ledger technology that is to be checked routinely.
2. **Costs of Transaction:** The connected nodes are expected to validate the transaction of a given Distributed Ledger Technology which gives high transaction cost as the other nodes are paid incentives to validate the transaction.
3. **Slow Transaction Speed:** The major disadvantage of this DLT is the slow speed of transactions as multiple nodes are attached to this network and it takes time to validate the transaction by all the other nodes.
4. **Scalability Issues:** Due to low speed and high transaction costs DLT faces very difficulties to expand on a large scale.
5. **Lack of Regulation:** As DLT is a decentralized technology, it operates outside the control of any centralized authority which can lead to a lack of regulation, making it difficult to hold accountable any wrongdoings or fraudulent activities on the network.

6. **Energy Consumption:** Distributed Ledger Technology requires a significant amount of energy to maintain the network and validate transactions, especially in the case of Proof of Work consensus mechanisms, which can lead to a negative impact on the environment.
7. **Complexity:** Implementing and managing Distributed Ledger Technology can be complex and requires a high level of technical expertise, which can be a barrier to entry for many organizations and individuals.
8. **Privacy Concerns:** While the anonymity of participants on the network is considered an advantage, it can also be a disadvantage as it can lead to privacy concerns and illicit activities on the network.
9. **Lack of Interoperability:** Different Distributed Ledger Technologies may use different protocols, which can lead to interoperability issues, making it difficult for different networks to communicate and transact with each other.

Future of Distributed Ledger Technology

1. Experts in this area promote DLT as a solution for many problems that are present on the internet and will drastically be able to solve all these problems. Distributed Ledger Technology is termed the "Internet of Value". Transactions and processes will occur in real-time with the help of the internet.
2. Distributed Ledger Technology has the potential to impact problems in financial or banking, cyber security, healthcare, government, data security, etc. sectors with effective solutions.
3. Enterprises and visionaries are now faced with the challenge of establishing networks of entities that together can take advantage of DLT to radically change how they share and keep records, and innovate where DLT can enable entirely new processes and business models.

Conclusion

In conclusion, Blockchain and Distributed Ledger Technology (DLT) offer innovative solutions for secure and transparent record-keeping. By enabling decentralized and tamper-resistant data management, they enhance trust among participants while improving efficiency across various industries. With different types and categories available, organizations can choose the right technology to meet their specific needs. As these technologies continue to evolve, they promise to revolutionize how we conduct transactions and manage data.