

Cryptography is the study of securing communications from outside observers. **Encryption algorithms** take the original message, or **plaintext**, and converts it into ciphertext, which is not understandable.

The key allows the user to **decrypt** the message, thus ensuring on they can read the message. The strength of the randomness of an **encryption** is also studied, which makes it harder for anyone to guess the key or input of the algorithm.

Cryptography is how we can achieve more secure and robust connections to elevate our privacy.

Advancements in cryptography makes it harder to break encryptions so that encrypted files, folders, or network connections are only accessible to authorized users.

Cryptography focuses on four different objectives:

1. **Confidentiality**: Confidentiality ensures that only the intended recipient can decrypt the message and read its contents.
2. **Non-repudiation**: Non-repudiation means the sender of the message cannot backtrack in the future and deny their reasons for sending or creating the message.
3. **Integrity**: Integrity focuses on the ability to be certain that the information contained within the message cannot be modified while in storage or transit.
4. **Authenticity**: Authenticity ensures the sender and recipient can verify each other's identities and the destination of the message.

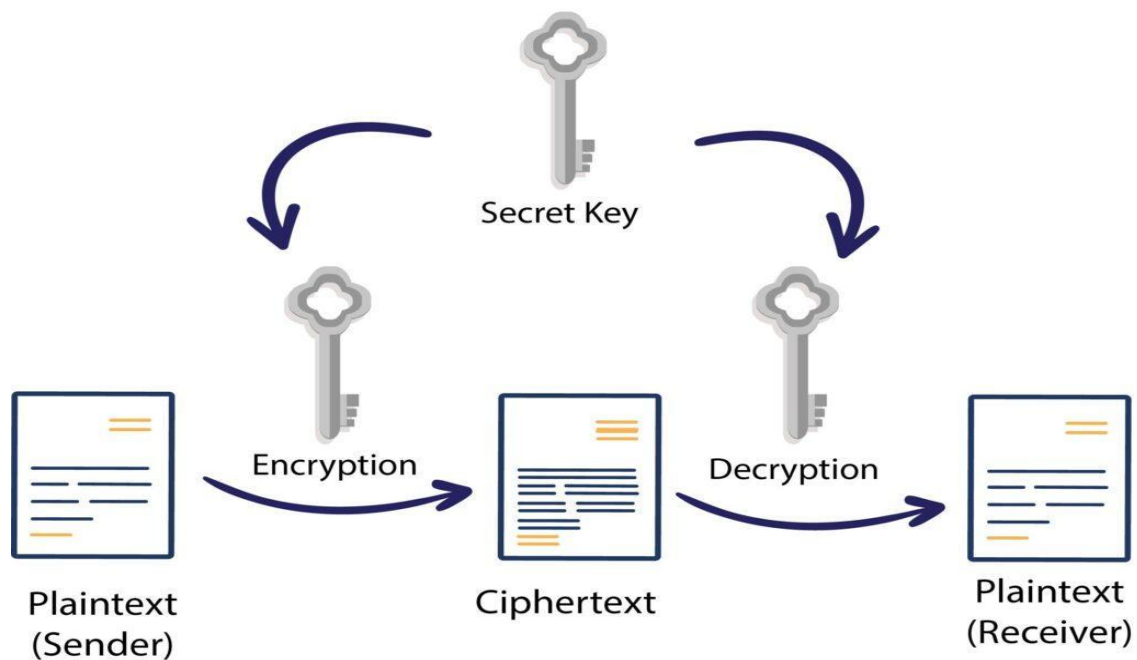
Types of Cryptography

Secret Key Cryptography-

Secret Key Cryptography, or symmetric cryptography, uses a single key to encrypt data. Both encryption and decryption in symmetric cryptography use the same key, making this the easiest form

of-cryptography.

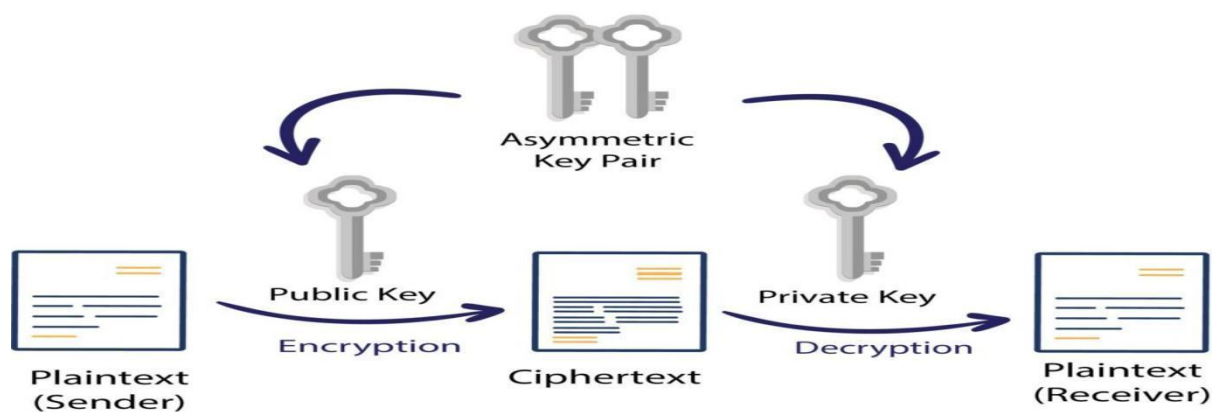
Symmetric Encryption



- **Public Key Cryptography**

Public Key Cryptography, or asymmetric cryptography, uses two keys to encrypt data. One is used for encryption, while the other key can decrypts the message. Unlike symmetric cryptography, if one key is used to encrypt, that same key cannot decrypt the message, rather the other key shall be used.

Asymmetric Encryption



One key is kept private, and is called the “private key”, while the other is shared publicly and can be used by anyone, hence it is known as the “public key”. The mathematical relation of the keys is such that the private key cannot be derived from the public key, but the public key can be derived from the private. The private key should not be distributed and should remain with the owner only. The public key can be given to any other entity.

- **Hash Functions**

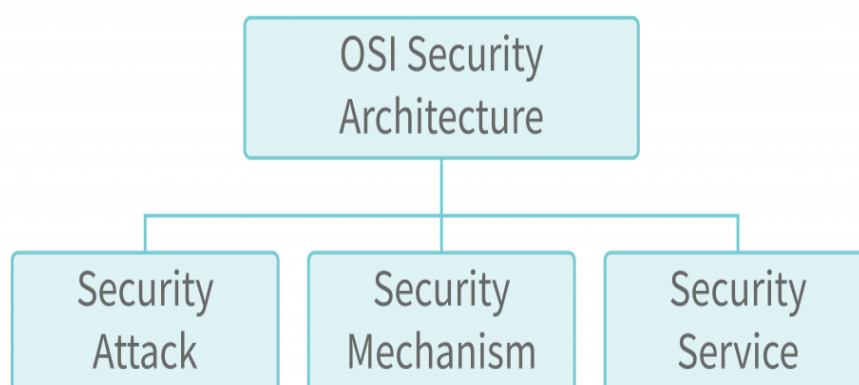
Hash functions are irreversible, one-way functions which protect the data, at the cost of not being able to recover the original message. Hashing is a way to transform a given string into a fixed length string. A good hashing algorithm will produce unique outputs for each input given. The only way to crack a hash is by trying every input possible, until you get the exact same hash. A hash can be used for hashing data (such as passwords) and in certificates.

Some of the most famous hashing algorithms are:

- MD5
- SHA-1

OSI Security Architecture

Classification of OSI Security Architecture



the OSI Security model identifies the attacks on a system (data) and also identifies various security services and the mechanisms to implement those services in various layers of the OSI model. Let us first discuss Security Attacks.

Security Attacks

A security attack means any action that puts the data or overall security of the system at risk. An attack might be successful or unsuccessful.

In case of a successful attack, the attacker can complete his/her motive of breaking the security of the system in any way he/she wants to.

In case of an unsuccessful attack, the system remains secured and no harm to the security is done. There are majorly 2 types of attacks: active attacks and passive attacks. Let's discuss them in detail.

1.Passive Attack

A passive attack is a kind of attack in which the data that is sent from the sender to the receiver is read by the attacker in the middle of the transmission. However, the main point to note here is that the passive attack is the attack in which the attacker does not modify or corrupt the data. No changes are made to the data.

The attacker just observes the data sent to the receiver from the sender and can know a lot of information about the sender and the receiver just by observing the communication between them. There are 2 types of passive attacks.

- **Traffic Analysis:** As the name suggests, this attack focuses on the amount or volume of data sent between the sender and the receiver.¹ The attacker can predict a lot of information about the sender and the receiver by knowing the amount of data sent.
- For example, if a lot of data is being sent from the sender to the receiver, it is assumed as there is an emergency, or a task is happening on an urgent basis. If less data is shared between the sender and the receiver, it is assumed that there is a lack of communication and so on.

- **Eavesdropping:** In this kind of attack, the attacker reads the communication that happens between the sender and the receiver and then can use this information for many things. For instance, an attacker can use the information to know about the financial details of the user. Also, this can be used for criminal activities as the attacker can send a lot of personal information to a criminal.
- The difference between eavesdropping and traffic analysis is that in traffic analysis, the attacker does not even read the data. He/she is just focused on the volume of the data. Whereas on the other hand, in eavesdropping, the focus is on the actual data being exchanged between the sender and the receiver.

2.Active Attack

In an active attack, the focus of the attacker is to modify the data that is being exchanged between the sender and the receiver. The most dangerous thing about this attack is that most of the time, the sender and the receiver do not even know that an attack has happened. There are several types of active attacks. Some of them are as follows:

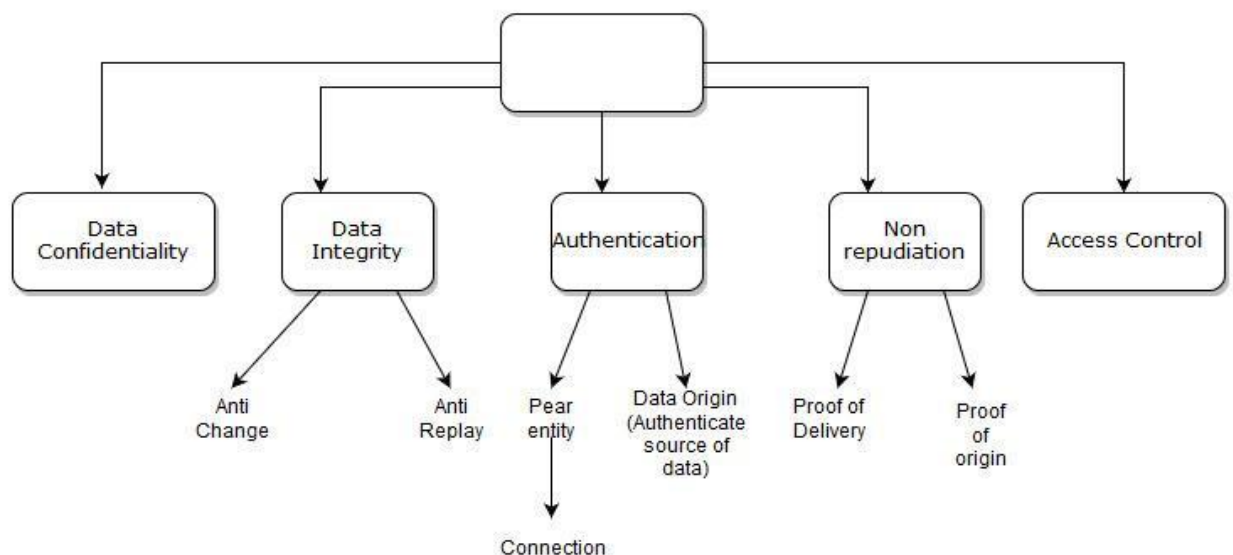
- **Replay:** In a replay attack, the attacker acts as an authorized user and can use the details of the authorized user to log in to a system. This happens as follows. Suppose that there is a user, and he/she wants to log in to a system. So, they enter their username and password, and this detail reaches in the form of a data packet to the server of the system. The attacker can steal this data packet in between and use this data packet later to log in to the system. You might be wondering that the login details are encrypted, so how would the attacker use them? The encryption will not matter in this case as the data packet as it is, has been stolen and the server might not recognize this and give access to the attacker.
- **Masquerade:** In this attack too, the attacker acts to be an authorized user. Now, this is not done by stealing the data packet. It is done by stealing the login details of the user somehow. So, no technical aspect of stealing the details is involved here.
- **Denial of Service (DOS):** The denial-of-service attack is an attack in which a system is attacked by a lot of requests to the system at one time that it is not able to handle. The attacker sends multiple requests to the server at the same time and the server is not able to

handle such requests. However, this attack is easily identifiable as these loads of requests come from a single sender (the attacker) and it is easy to identify the source of the attack.

- **Distributed Denial of Service (DDOS):** As we saw, in the denial-of-service attack, the source of the attack can be easily identified. Now, there is a modified version of this attack i.e., DDOS i.e., distributed version of the DOS attack. In this attack, the attacker first observes the details of a lot of authorized users. Then, the attacker uses these authorized users at the same time to send requests to the system. Now, thousands (or even more) of requests at the same are sent to the system and the system cannot recognize the source of attack as there is each request from a different user, and all the users are authorized. So, the attacker is using the authorized users as victims too. The primary victim is the system, and the secondary victims are the authorized users. The authorized users are called Zombie PCs.

Security Services:

A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. These services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service. It has five categories



Authentication: The assurance that the communicating entity is the one that it claims to be./

This is a very basic and easy service to implement. In authentication, the system (both sender and receiver) identifies the user first. Only the user authorized to enter the system can use it. This can be done using basic password protection.

E.g In case of the single message, such a warning or alarm signal , the function of the authentication service is to assure that recipient that the message is from the source that it claims to be from.

- **Peer Entity Authentication:** Used in association with a logical connection to provide confidence in the identity of the entities connected.
- **Data-Origin Authentication:** In a connectionless transfer, provides assurance that the source of received data is as claimed.

Data Confidentiality: Protects data from unauthorized disclosure.

This is one of the three pillars of the security model CIA (Confidentiality, Integrity, and Availability). Confidentiality means that the data i.e. is shared between a sender and receiver should be confidential to them only. No third party should be able to read the data.

Access Control: The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).

Data Integrity: The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

integrity means that no third party should be able to modify the data i.e. is shared between the sender and the receiver.

Non-repudiation: Protects against denial by one of the entities involved in a communication of having participated in all or part of the communication.

OR

It prevent either sender or receiver from denying a transmitted messages. Thus, when a message sent , the receiver can prove that the sender in fact sent the message.

Similarly when the message is received , the sender can prove that the receiver in fact receive the message.

Proof of Origin: Proof that the message was sent by the specified party.

Proof of Delivery: Proof that the message was received by the specified party.

Security Mechanisms

The mechanisms that help in setting up the security services in different layers of the OSI model and that help in identifying any attack or data breach are called security mechanisms. The security mechanisms provide a way of preventing, protecting, and detecting attacks. Some of the security mechanisms are as follows

- **Encipherment (Encryption):** One of the most popular security mechanisms is encryption. The message/data sent from the sender to the receiver is usually encrypted to some format that even if the message is stolen, cannot be decrypted easily by the attacker. Some of the popular encryption algorithms are AES, RSA, Triple DES, etc.
- **Traffic Padding:** The sender and receiver send the data to each other. Now, sometimes there is a gap between the sender and receiver. This means that for some time when the sender and receiver are not sharing the data, the attacker can act as if it is the sender and send some data to the receiver to attack it. So, this can be avoided if the gap (empty time) between the sender and the receiver is not known to the attacker. For this, during the gap duration, the sender keeps on sending some dummy data to the receiver and the receiver knows that this is the dummy data by using some identification. Hence, no gap is created between the sender and the receiver and the attacker cannot attack the system.
- **Routing Control:** The messages that a sender sends to a receiver travel different routes. However, in some cases, the sender and receiver might communicate mostly via the same route. In this case, the attacker tracks this route and can make changes to the data or take advantage of this. So, routing should be controlled in such a way that mostly, a different route is selected between the sender and the receiver to deliver the message.

So, these were some of the security mechanisms that can be used to detect/prevent attacks. This is the complete OSI Security model. Let us now discuss some benefits of OSI Security Architecture.

Symmetric cipher- Symmetric Encryption is the most basic and old method of encryption. It uses only one key for the process of both the encryption and decryption of data. Thus, it is also known as Single-Key Encryption. A few basic terms in Cryptography are as follows:

Plain Text: original message to be communicated between sender and receiver

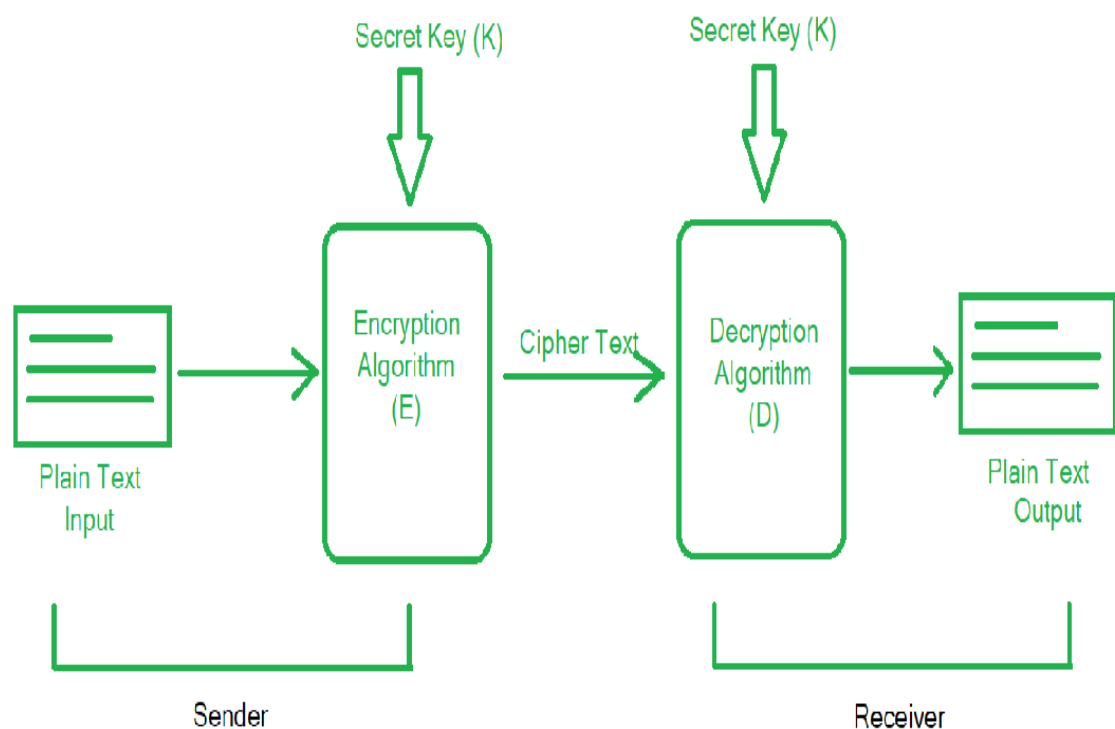
Cipher Text: encoded format of the original message that cannot be understood by humans

Encryption (or Enciphering): the conversion of plain text to cipher text

Decryption (or Deciphering): the conversion of cipher text to plain text, i.e., reverse of encryption

Symmetric Cipher Model:

A symmetric cipher model is composed of five essential parts:



1. **Plain Text (x):** This is the original data/message that is to be communicated to the receiver by the sender. It is one of the inputs to the encryption algorithm.

2. **Secret Key (k):** It is a value/string/textfile used by the encryption and decryption algorithm to encode and decode the plain text to cipher text and vice-versa respectively. It is independent of the encryption algorithm. It governs all the conversions in plain text. All the substitutions and transformations done depend on the secret key.

3. **Encryption Algorithm (E):** It takes the plain text and the secret key as inputs and produces Cipher Text as output. It implies several techniques such as substitutions and transformations on the plain text using the secret key. $E(x, k) = y$

4. **Cipher Text (y):** It is the formatted form of the plain text (x) which is unreadable for humans, hence providing encryption during the transmission. It is completely dependent upon the secret key provided to the encryption algorithm. Each unique secret key produces a unique cipher text.

5. **Decryption Algorithm (D):** It performs reversal of the encryption algorithm at the recipient's side. It also takes the secret key as input and decodes the cipher text received from the sender based on the secret key. It produces plain text as output. $D(y, k) = x$

Requirements for Encryption:- There are only two requirements that need to be met to perform encryption. They are,

1. Encryption Algorithm: There is a need for a very strong encryption algorithm that produces cipher texts in such a way that the attacker should be unable to crack the secret key even if they have access to one or more cipher texts.

2. Secure way to share Secret Key: There must be a secure and robust way to share the secret key between the sender and the receiver. It should be leakproof so that the attacker cannot access the secret key.

Mono-Alphabetic Substitution Cipher

Plain text alphabet:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Cipher text alphabet (key):	M	U	A	L	V	O	Z	K	R	N	J	X	Q	D	F	S	H	P	E	B	C	T	I

A **mono-alphabetic cipher** (aka **simple substitution cipher**) is a substitution cipher where each letter of the plain text is replaced with another letter of the alphabet. It uses a **fixed key** which consist of the 26 letters of a “shuffled alphabet”.

Polyalphabetic Cipher?

The polyalphabetic cipher refers to the ciphers that are based on substitution with multiple substitution alphabets. The most prominent example of this type of cipher is the Vigenère cipher, although it is basically a special simplified case.

Vigenere Cipher

Introduction

The vigenere cipher is an algorithm that is used to encrypting and decrypting the text. The vigenere cipher is an algorithm of encrypting an alphabetic text that uses a series of interwoven caesar ciphers. It is based on a keyword's letters. It is an example of a polyalphabetic substitution cipher. This algorithm is easy to understand and implement. This

algorithm was first described in 1553 by **Giovan Battista Bellaso**. It uses a Vigenere table or Vigenere square for encryption and decryption of the text. The vigenere table is also called the tabula recta.

Two methods perform the vigenere cipher.

Method 1

When the vigenere table is given, the encryption and decryption are done using the vigenere table (26 * 26 matrix) in this method.

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

Example: The plaintext is "JAVATPOINT", and the key is "BEST".

To generate a new key, the given key is repeated in a circular manner, as long as the length of the plain text does not equal to the new key.

J	A	V	A	T	P	O	I	N	T
B	E	S	T	B	E	S	T	B	E

Encryption

The first letter of the plaintext is combined with the first letter of the key. The column of plain text "J" and row of key "B" intersects the alphabet of "K" in the vigenere table, so the first letter of ciphertext is "K".

Similarly, the second letter of the plaintext is combined with the second letter of the key. The column of plain text "A" and row of key "E"

intersects the alphabet of "E" in the vigenere table, so the second letter of ciphertext is "E".

This process continues continuously until the plaintext is finished.

Ciphertext = KENTUTGBOX

Decryption

Decryption is done by the row of keys in the vigenere table. First, select the row of the key letter, find the ciphertext letter's position in that row, and then select the column label of the corresponding ciphertext as the plaintext.

K	E	N	T	U	T	G	B	O	X
B	E	S	T	B	E	S	T	B	E

For example, in the row of the key is "B" and the ciphertext is "K" and this ciphertext letter appears in the column "J", that means the first plaintext letter is "J".

Next, in the row of the key is "E" and the ciphertext is "E" and this ciphertext letter appears in the column "A", that means the second plaintext letter is "A".

This process continues continuously until the ciphertext is finished.

Plaintext = JAVATPOINT

Method 2

When the vigenere table is not given, the encryption and decryption are done by Vigenar algebraically formula in this method (convert the letters (A-Z) into the numbers (0-25)).

Formula of encryption is,

$$E_i = (P_i + K_i) \bmod 26$$

Formula of decryption is,

$$D_i = (E_i - K_i) \bmod 26$$

If any case (D_i) value becomes negative (-ve), in this case, we will add 26 in the negative value.

Where,

E denotes the encryption.

D denotes the decryption.

P denotes the plaintext.

K denotes the key.

Note: "i" denotes the offset of the ith number of the letters, as shown in the table below.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example: The plaintext is "JAVATPOINT", and the key is "BEST".

Encryption: $E_i = (P_i + K_i) \bmod 26$

Plaintext	J	A	V	A	T	P	O	I	N	T
Plaintext value (P)	09	00	21	00	19	15	14	08	13	19
Key	B	E	S	T	B	E	S	T	B	E
Key value (K)	01	04	18	19	01	04	18	19	01	04
Ciphertext value (E)	10	04	13	19	20	19	06	01	14	23
Ciphertext	K	E	N	T	U	T	G	B	O	X

Decryption: $D_i = (E_i - K_i) \bmod 26$

If any case (D_i) value becomes negative (-ve), in this case, we will add 26 in the negative value. Like, the third letter of the ciphertext;

$N = 13$ and $S = 18$

$D_i = (E_i - K_i) \bmod 26$

$D_i = (13 - 18) \bmod 26$

$D_i = -5 \bmod 26$

$D_i = (-5 + 26) \bmod 26$

$D_i = 21$

Ciphertext	K	E	N	T	U	T	G	B	O	X
Ciphertext value (E)	10	04	13	19	20	19	06	01	14	23
Key	B	E	S	T	B	E	S	T	B	E
Key value (K)	01	04	18	19	01	04	18	19	01	04
Plaintext value (P)	09	00	21	00	19	15	14	08	13	19
Plaintext	J	A	V	A	T	P	O	I	N	T

Playfair Cipher

Playfair cipher is an encryption algorithm to encrypt or encode a message. It is the same as a traditional cipher. The only difference is that it encrypts a **digraph** (a pair of two letters) instead of a single letter.

It initially creates a key-table of 5*5 matrix.

The matrix contains alphabets that act as the key for encryption of the plaintext.

Note that any alphabet should not be repeated.

Another point to note that there are 26 alphabets and we have only 25 blocks to put a letter inside it. Therefore, one letter is excess so, a letter will be omitted (usually J) from the matrix.

Nevertheless, the plaintext contains J, then **J** is replaced by **I**. It means treat I and J as the same letter, accordingly.

- o **Plaintext:** It is the original message that is to be encrypted. It is also known as a **message**.
- o **Ciphertext:** It is an encrypted message.
- o **Cipher:** It is an algorithm for transforming plaintext to ciphertext.
- o **Key:** It is the key to encrypt or decrypt the plaintext. It is known only to the sender and receiver. It is filled character by character in the matrix that is called **key-table** or **key-matrix**.
- o **Encipher:** The process of converting plaintext into ciphertext is called **encipher**.
- o **Decipher:** The process of removing ciphertext from plaintext is called **decipher**.
- o **Cryptanalysis:** It is the study of the methods and principles of deciphering ciphertext without knowing the key.

Playfair Cipher Encryption Rules

1. First, split the plaintext into **digraphs** (pair of two letters). If the plaintext has the odd number of letters, append the letter **Z** at the end of the plaintext. It makes the plaintext of **even**

For example, the plaintext **MANGO** has five letters. So, it is not possible to make a digraph. Since, we will append a letter **Z** at the end of the plaintext, i.e. **MANGOZ**.

2. After that, break the plaintext into **digraphs** (pair of two letters). If any letter appears twice (side by side), put **X** at the place of the second occurrence.

Suppose, the plaintext is **COMMUNICATE** then its digraph becomes **CO MX MU NI CA TE**. Similarly, the digraph for the

plaintext **JAZZ** will be **JA ZX ZX**, and for plaintext **GREET**, the digraph will be **GR EX ET**.

3. To determine the cipher (encryption) text, first, build a 5*5 key-matrix or key-table and filled it with the letters of alphabets


4. If KEY value is repeating take once at a time....like AAbhm===In matrix box it will come Abhm

4. There may be the following three conditions:

i) If a pair of letters (digraph) appears in the same row

In this case, replace each letter of the digraph with the letters immediately to their right. If there is no letter to the right, consider the first letter of the same row as the right letter. Suppose, **Z** is a letter whose right letter is required, in such case, **T** will be right to **Z**.

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



Right of Z

ii) If a pair of letters (digraph) appears in the same column

In this case, replace each letter of the digraph with the letters immediately below them. If there is no letter below, wrap around to the top of the same column. Suppose, **W** is a letter whose below letter is required, in such case, **V** will be below **W**.

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

Below of W is V

iii) If a pair of letters (digraph) appears in a different row and different column

In this case, select a 3*3 matrix from a 5*5 matrix such that pair of letters appear in the 3*3 matrix. Since they occupy two opposite corners of a square within the matrix. The other corner will be a cipher for the given digraph.

In other words, we can also say that intersection of H and Y will be the cipher for the first letter and

Suppose, a digraph is **HY** and we have to find a cipher for it. We observe that both H and Y are placed in different rows and different columns. In such cases, we have to select a 3*3 matrix in such a way that both H and Y appear in the 3*3 matrix (highlighted with yellow color). Now, we will consider only the selected matrix to find the cipher.

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

Now to find the cipher for HY, we will consider the diagonal **opposite** to HY, i.e. **LU**. Therefore, the cipher for **H** will be **L**, and the cipher for **Y** will be **U**.

Example of Playfair Cipher

Suppose, the plaintext is **COMMUNICATE** and the key that we will use to encipher the plaintext is **COMPUTER**.

1. First, split the plaintext into digraph (by rule 2) i.e. **CO MX MU NI CA TE**.
2. Construct a 5*5 key-matrix (by rule 3). In our case, the key is **COMPUTER**.

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

3. Now, we will traverse in key-matrix pair by pair and find the corresponding encipher for the pair.
 - o The first digraph is **CO**. The pair appears in the same row. By using **Rule 4(i)** **CO** gets encipher into **OM**.
 - o The second digraph is **MX**. The pair appears in the same column. By using **Rule 4(ii)** **MX** gets encipher into **RM**.
 - o The third digraph is **MU**. The pair appears in the same row. By using **Rule 4(i)** **MU** gets encipher into **PC**.
 - o The fourth digraph is **NI**. The pair appears in different rows and different columns. By using **Rule 4(iii)** **NI** gets encipher into **SG**.
 - o The fifth digraph is **CA**. The pair appears in different rows and different columns. By using **Rule 4(iii)** **CA** gets encipher into **PT**.
 - o The sixth digraph is **TE**. The pair appears in the same row. By using **Rule 4(i)** **TE** gets encipher into **ER**.

Therefore, the plaintext **COMMUNICATE** gets encipher (encrypted) into **OMRMPCSGPTER**.

Hill cipher

In hill cipher algorithm every letter (A-Z) is represented by a number moduli 26. Usually, the simple substitution scheme is used where A = 0, B = 1, C = 2...Z = 25 in order to use 2x2 key matrix.

Note: The complexity of the algorithm increases with the size of the key matrix.

Encryption

To encrypt the text using hill cipher, we need to perform the following operation.

$$1. E(K, P) = (K * P) \bmod 26$$

Where **K** is the key matrix and **P** is plain text in **vector form**. Matrix multiplication of K and P generates the encrypted ciphertext.

Steps For Encryption

Step 1: Let's say our key text (2x2) is **DCDF**. Convert this key using a substitution scheme into a 2x2 key matrix as shown below:

$$\text{DCDF} \longrightarrow \begin{bmatrix} D & D \\ C & F \end{bmatrix} \longrightarrow \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$$

Step 2: Now, we will convert our plain text into vector form. Since the key matrix is 2x2, the vector must be 2x1 for matrix multiplication. (Suppose the key matrix is 3x3, a vector will be a 3x1 matrix.)

In our case, plain text is **TEXT** that is four letters long word; thus we can put in a 2x1 vector and then substitute as:

$$\text{TEXT} \longrightarrow \begin{bmatrix} T \\ E \end{bmatrix} \quad \begin{bmatrix} X \\ T \end{bmatrix} \longrightarrow \begin{bmatrix} 19 \\ 4 \end{bmatrix} \quad \begin{bmatrix} 23 \\ 19 \end{bmatrix}$$

Step 3: Multiply the key matrix with each 2x1 plain text vector, and take the modulo of result (2x1 vectors) by 26. Then concatenate the results, and we get the encrypted or ciphertext as **RGWL**.

$$\begin{bmatrix} D & D \\ C & F \end{bmatrix} \times \begin{bmatrix} T \\ E \end{bmatrix} \longrightarrow \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 19 \\ 4 \end{bmatrix} = \begin{bmatrix} 69 \\ 58 \end{bmatrix} \% 26 = \begin{bmatrix} 17 \\ 6 \end{bmatrix} \longrightarrow \begin{bmatrix} R \\ G \end{bmatrix}$$

$$\begin{bmatrix} D & D \\ C & F \end{bmatrix} \times \begin{bmatrix} X \\ T \end{bmatrix} \longrightarrow \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 23 \\ 19 \end{bmatrix} = \begin{bmatrix} 126 \\ 141 \end{bmatrix} \% 26 = \begin{bmatrix} 22 \\ 11 \end{bmatrix} \longrightarrow \begin{bmatrix} W \\ L \end{bmatrix}$$

} **RGWL**

Decryption

To encrypt the text using hill cipher, we need to perform the following operation.

$$1. D(K, C) = (K^{-1} * C) \bmod 26$$

Where **K** is the key matrix and **C** is the ciphertext in **vector form**. Matrix multiplication of inverse of key matrix K and ciphertext C generates the decrypted plain text.

Steps For Decryption

Step 1: Calculate the inverse of the key matrix. First, we need to find the determinant of the key matrix (must be between 0-25). Here the Extended Euclidean algorithm is used to get modulo multiplicative inverse of key matrix determinant

$$K^{-1} \bmod 26 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \xrightarrow[\text{solve by Extended Euclidean Algorithm}]{\% 26 = ((3 \times 2) - (3 \times 2))^{-1} \times} \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix} \xrightarrow{\% 26 = 3} \begin{bmatrix} 5 & 23 \\ 24 & 3 \end{bmatrix} = \begin{bmatrix} 15 & 69 \\ 72 & 9 \end{bmatrix} \xrightarrow{\% 26 =} \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}$$

Step 2: Now, we multiply the 2x1 blocks of ciphertext and the inverse of the key matrix. The resultant block after concatenation is the plain text that we have encrypted i.e., **TEXT**.

$$\begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} \times \begin{bmatrix} 17 \\ 6 \end{bmatrix} = \begin{bmatrix} 357 \\ 394 \end{bmatrix} \xrightarrow{\% 26} \begin{bmatrix} 19 \\ 4 \end{bmatrix} \rightarrow \begin{bmatrix} T \\ E \end{bmatrix}$$

$$\begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} \times \begin{bmatrix} 22 \\ 11 \end{bmatrix} = \begin{bmatrix} 517 \\ 539 \end{bmatrix} \xrightarrow{\% 26} \begin{bmatrix} 23 \\ 19 \end{bmatrix} \rightarrow \begin{bmatrix} X \\ T \end{bmatrix}$$

} **TEXT**

transposition techniques

Transposition Cipher is a cryptographic algorithm where the order of alphabets in the plaintext is rearranged to form a cipher text. In this process, the actual plain text alphabets are not included.

Keyless Transposition Ciphers Simple transposition ciphers, which were used in the past, are keyless. There are two methods for permutation of characters. In the first method, the text is written into a table column by column and then transmitted row by row. In the second method, the text is written into a table row by row and then transmitted column by column.

EXAMPLE 3.22 A good example of a keyless cipher using the first method is the rail fence cipher. In this cipher, the plaintext is arranged in two lines as a zigzag pattern (which means column by column); the ciphertext is created reading the pattern row by row. For example, to send the message "Meet me at the park" to Bob, Alice writes

```

m   e   m   a   t   e   a   k
 \  / \  / \  / \  / \  / \
  e   t   e   t   h   p   r

```

She then creates the ciphertext "MEMATEAKETETHPR" by sending the first row followed by the second row. Bob receives the ciphertext and divides it in half (in this case the second half has one less character). The first half forms the first row; the second half, the second row. Bob reads the result in zigzag. Because there is no key and the number of rows is fixed (2), the cryptanalysis of the ciphertext would be very easy for Eve. All she needs to know is that the rail fence cipher is used.

EXAMPLE 3.23 Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

```

m e e t
m e a t
t h e p
a r k

```

She then creates the ciphertext "MMTAEHREAEKTP" by transmitting the characters column by column. Bob receives the ciphertext and follows the reverse process. He writes the received message, column by column, and reads it row by row as the plaintext. Eve can easily decipher the message if she knows the number of columns.

Its not necessary to take only 4 row and 4 column .We can take any row and column

Keyed Transposition Ciphers The keyless ciphers permute the characters by using writing plaintext in one way (row by row, for example) and reading it in another way (column by column, for example). The permutation is done on the whole plaintext to create the whole ciphertext. Another method is to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.

EXAMPLE 3.25 Alice needs to send the message "Enemy attacks tonight" to Bob. Alice and Bob have agreed to divide the text into groups of five characters and then permute the characters in each group. The following shows the grouping after adding a bogus character at the end to make the last group the same size as the others.

```

e n e m y   a t t a c   k s t o n   i g h t z

```

e n e m y a t t a c k s t o n i g h t z

The key used for encryption and decryption is a permutation key, which shows how the characters are permuted. For this message, assume that Alice and Bob used the following key:

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

The third character in the plaintext block becomes the first character in the ciphertext block; the first character in the plaintext block becomes the second character in the ciphertext block; and so on.

For enemy

Sr.no	1	2	3	4	5
Plain text	e	n	e	m	y

Sr.no	3	1	4	5	2
Cipher text	e	e	m	y	n

For attac

Sr.no	1	2	3	4	5
Plain text	a	t	t	a	c

Sr.no	3	1	4	5	2
Cipher text	t	a	a	c	t

For kston

Sr.no	1	2	3	4	5
Plain text	k	s	t	o	n

Sr.no	3	1	4	5	2
Cipher text	t	k	o	n	s

For ightz

Sr.no	1	2	3	4	5
Plain text	i	g	h	t	z

Sr.no	3	1	4	5	2
-------	---	---	---	---	---

Cipher text	h	i	t	z	g
----------------	---	---	---	---	---

The permutation yields
 E E M Y N T A A C T T K O N S H I T Z G

Steganography

The word **Steganography** is derived from two Greek words- ‘stegos’ meaning ‘to cover’ and ‘graphia’, meaning ‘writing’, thus translating to ‘covered writing’, or ‘hidden writing’.

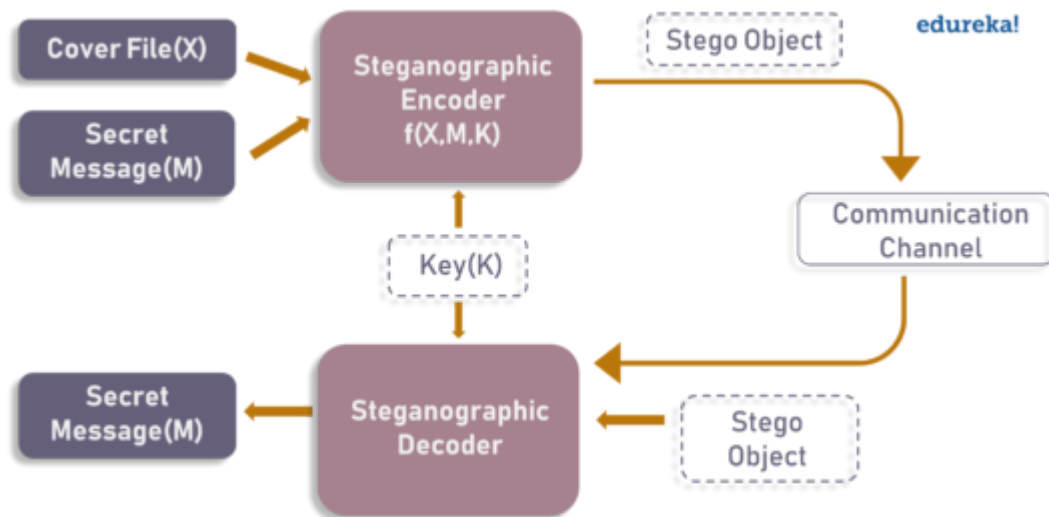
Steganography is a method of hiding secret data, by embedding it into an audio, video, image, or text file.

It is one of the methods employed to protect secret or sensitive data from malicious attacks.

cryptography is similar to writing a letter in a secret language: people can read it, but won’t understand what it means.

steganography in the same situation, you would hide the letter inside a pair of socks that you would be gifting the intended recipient of the letter. To those who don’t know about the message, it would look like there was nothing more to your gift than the socks. But the intended recipient knows what to look for, and finds the message hidden in them.

The diagram below depicts a basic steganographic model



As the image depicts, both cover file(X) and secret message(M) are fed into steganographic encoder as input. Steganographic Encoder function, $f(X,M,K)$ embeds the secret message into a cover file. Resulting Stego Object looks very similar to your cover file, with no visible changes. This completes encoding. To retrieve the secret message, Stego Object is fed into Steganographic Decoder.

Steganography Techniques

1. Text Steganography
2. Image Steganography
3. Video Steganography
4. Audio Steganography
5. Network Steganography

Text Steganography

Text Steganography is hiding information inside the text files. It involves things like changing the format of existing text, changing words within a text, generating random character sequences or using context-free grammars to generate readable texts. Various techniques used to hide the data in the text are:

- Format Based Method
- Random and Statistical Generation
- Linguistic Method

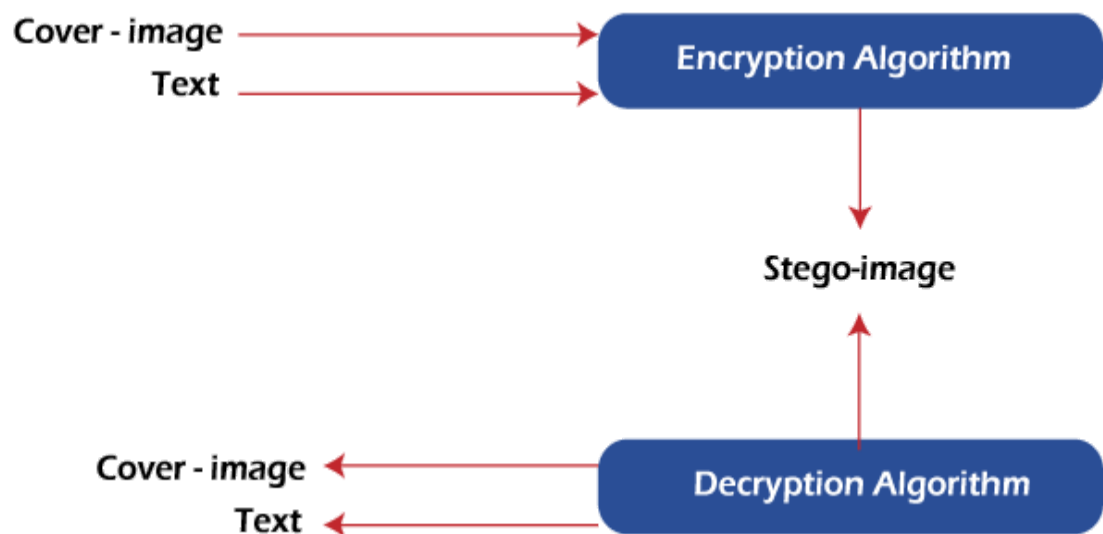
Image Steganography

Hiding the data by taking the cover object as the image is known as image steganography. In digital steganography, images are widely used

cover source because there are a huge number of bits present in the digital representation of an image.

How does it work?

In memory, an image is represented as a $N \times M$ (for greyscale images) or $N \times M \times 3$ (for colour images) matrix, with each entry representing the intensity value of a pixel. Image steganography embeds a message into an image by changing the values of some pixels chosen by an encryption algorithm.



There are a lot of ways to hide information inside an image. Common approaches include:

- Least Significant Bit Insertion
- Masking and Filtering
- Redundant Pattern Encoding
- Encrypt and Scatter
- Coding and Cosine Transformation

Audio Steganography

In audio steganography, the secret message is embedded into an audio signal which alters the binary sequence of the corresponding audio file. Hiding secret messages in digital sound is a much more difficult process when compared to others, such as Image Steganography. Different methods of audio steganography include:

- Least Significant Bit Encoding

- Parity Encoding
- Phase Coding
- Spread Spectrum

This method hides the data in WAV, AU, and even MP3 sound files.

Video Steganography

In Video Steganography you can hide kind of data into digital video format. The advantage of this type is a large amount of data can be hidden inside and the fact that it is a moving stream of images and sounds. You can think of this as the combination of Image Steganography and Audio Steganography. Two main classes of Video Steganography include:

- Embedding data in uncompressed raw video and compressing it later
- Embedding data directly into the compressed data stream

Network Steganography (Protocol Steganography)

It is the technique of embedding information within network control protocols used in data transmission such TCP, UDP, ICMP etc. You can use steganography in some covert channels that you can find in the OSI model. For Example, you can hide information in the header of a TCP/IP packet in some fields that are either optional.

In today's digitalized world, various software tools are available for Steganography. In the remainder of this Steganography Tutorial, we will explore some of the popular steganographic tools and their capabilities.