Symmetric Key Encryption

- is a form of encryption whereby the same key is used to encrypt and decrypt the message.
- It's the oldest and most well-known technique for encryption. The secret key can be a word, a number, or a string of letters, and it's applied to a message. The message is changed following the rules in the key. Sender and receiver know the key, and can thus code and decode any message that would use that specific key.
- The main drawback of the symmetric key encryption is that all individuals engaged in the activity have to exchange the key used to encrypt the data before they can decrypt it, which isn't always convenient.
- Examples of Symmetric Encryption: Blowfish, AES (Advanced Encryption Standard), RC4 (Rivest Cipher 4), DES (Data Encryption Standard), RC5 (Rivest Cipher 5), and RC6 (Rivest Cipher 6)

Types of Symmetric Encryption

- **Block algorithms**
    - are used to encrypt blocks of electronic data. Specified set lengths of bits are altered, while continuing to use the designated private key. This key is then used for each block. When network stream data is being encrypted, the encryption system retains the data in its memory components while waiting for the complete blocks. The time in which the system waits can lead to a certain security gap, and may undermine data security and integrity.
- **Stream algorithms**
    - are not retained in the encryption system's memory, but arrive in data stream algorithms. This type of procedure is considered somewhat safer, since a disk or system is not retaining the data without encryption in the memory components.

Asymmetric key encryption

- also known as public key encryption, is a cryptographic technique that uses two different keys for encrypting and decrypting data. These two keys are mathematically related but different, and are referred to as the public key and the private key.
- In asymmetric key encryption, the public key is used to encrypt data, and the private key is used to decrypt the data. This means that anyone can use the public key to encrypt data that only the owner of the private key can decrypt.

Types of Asymmetric key encryption

- **RSA Algorithm (Rivest-Shamir-Adleman)**
    - This is the most widely used asymmetric key encryption algorithm. It was developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman.
    - An asymmetric key encryption algorithm widely used for secure data transmission and digital signatures.

- **Diffie-Hellman**
    - This algorithm is used for secure key exchange between two parties, and was developed by Whitfield Diffie and Martin Hellman in 1976.
    - An asymmetric key encryption algorithm used for secure key exchange between two parties without transmitting the key itself.
- **Elliptic Curve Cryptography (ECC)**
    - This algorithm is used for digital signatures and key exchange, and is based on the mathematics of elliptic curves.
    - An asymmetric key encryption algorithm that uses elliptic curves to provide equivalent security to traditional encryption algorithms but with shorter key lengths.
- **Digital Signature Algorithm (DSA)**
    - This algorithm is used for digital signatures and was developed by the National Institute of Standards and Technology (NIST).
    - An asymmetric key encryption algorithm used for digital signatures to verify the authenticity of digital documents or messages.

Hashing

- is a process of converting input data into a fixed-size output, which is usually a sequence of alphanumeric characters. The output generated by the hash function is called a hash value, hash code, or checksum. Hashing is commonly used in computer science for various purposes, such as data encryption, data validation, and data indexing.

Types of Hashing Algorithms

- **MD5**
    - is a widely-used hashing algorithm that produces a 128-bit hash value. It is commonly used for data validation and data integrity checks. However, it is not recommended for cryptographic purposes since it is vulnerable to collision attacks.
- **SHA-1**
    - is a widely-used hashing algorithm that produces a 160-bit hash value. It is commonly used for data validation, digital signatures, and message authentication codes. However, it is also vulnerable to collision attacks.
- **SHA-256**
    - is a widely-used hashing algorithm that produces a 256-bit hash value. It is commonly used for data encryption, data validation, and digital signatures. It is considered more secure than MD5 and SHA-1.
- **HMAC**
    - is a type of hash function that uses a secret key to generate a hash value. It is commonly used for message authentication, digital signatures, and data validation.
- **PBKDF2**

    - is a type of hash function that is used for password-based key derivation. It is commonly used to derive a cryptographic key from a password. It is designed to be computationally expensive to make brute force attacks more difficult.

Quantum Encryption

- also known as quantum cryptography, is a method of secure communication that uses the principles of quantum mechanics. It relies on the fundamental laws of physics to provide security, rather than relying on complex mathematical algorithms.

Main types of Quantum Encryption

- **Quantum Key Distribution (QKD)**
    - In this method, two parties - the sender and the receiver - use a shared secret key to encrypt and decrypt messages. The key is generated using a quantum channel, which is used to transmit quantum bits or qubits. The qubits are then measured, and the results are used to generate the key. Any attempt to intercept or measure the qubits will disturb their state, alerting the parties to the presence of an eavesdropper.
- **Quantum Secure Direct Communication (QSDC)**
    - This method allows for the direct transmission of messages without the need for a shared secret key. Instead, it relies on the properties of entangled particles to ensure the security of the communication. Entangled particles are two particles that are connected in such a way that the state of one particle is dependent on the state of the other. This means that any attempt to intercept or measure one of the particles will change the state of the other, alerting the parties to the presence of an eavesdropper.

Steganography

- is the practice of concealing a message or information within another non-secret data in such a way that the existence of the secret message remains undetected.

Types of Steganography

- **Image Steganography**
    - This technique involves hiding a secret message or information within an image file, without affecting the original image's quality. Image steganography can use different approaches such as Least Significant Bit (LSB) insertion, where the message is hidden in the least significant bits of the image's pixels, or the Discrete Cosine Transform (DCT) technique, which is used to hide the message in the frequency domain of the image.
- **Audio Steganography**
    - This technique is similar to image steganography, but it hides the message within an audio file. Audio steganography can use different approaches, such as hiding the message in the low frequencies or changing the phase of the audio signal to hide the message.

- **Video Steganography**
    - This technique is similar to audio and image steganography, but it hides the message within a video file. Video steganography can use different approaches, such as hiding the message in the motion vectors or the compressed video data.
- **Text Steganography**
    - This technique involves hiding a secret message or information within a text file. Text steganography can use different approaches, such as hiding the message in the white spaces or using special characters to represent the hidden message.

Homomorphic Encryption

- is a cryptographic technique that allows computations to be performed on encrypted data without the need to decrypt it first. This provides a significant advantage in terms of privacy and security, as the data remains encrypted throughout the entire computation process.

Types of Homomorphic Encryption

- **Fully Homomorphic Encryption (FHE)**
    - This type of homomorphic encryption allows arbitrary computations to be performed on encrypted data. In other words, it enables the evaluation of any function on encrypted data, without the need to decrypt it first. FHE is the most powerful form of homomorphic encryption, but it is also the most computationally expensive and complex to implement.
- **Partially Homomorphic Encryption (PHE)**
    - This type of homomorphic encryption allows computations on encrypted data, but only for specific operations. For example, it may allow the computation of either addition or multiplication on encrypted data, but not both. PHE is less powerful than FHE but is also less computationally expensive.
- **Somewhat Homomorphic Encryption (SHE)**
    - This is a variant of PHE that allows for a limited number of computations to be performed on encrypted data. It provides a balance between the power of FHE and the efficiency of PHE.