Symmetric Encryption: Symmetric encryption, also known as secret-key encryption or conventional encryption, involves using the same key for both the encryption and decryption processes. The key, known as the secret key, is shared between the sender and receiver in a secure manner before communication begins. Symmetric encryption is generally faster and more efficient than asymmetric encryption for encrypting large amounts of data. However, a significant challenge with symmetric encryption is securely distributing the secret key between parties. If an unauthorized person gains access to the key, they can decrypt the data.

Examples of symmetric encryption algorithms include the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES).

Asymmetric Encryption (Public-Key Encryption): Asymmetric encryption involves using a pair of mathematically related keys: a public key and a private key. The public key is widely distributed and can be freely shared with others. It is used for encryption, allowing anyone to encrypt data or messages intended for the owner of the corresponding private key. The private key, on the other hand, is kept secret and is used for decryption. Information encrypted with the public key can only be decrypted with the corresponding private key. Asymmetric encryption is primarily used for secure key exchange and digital signatures.

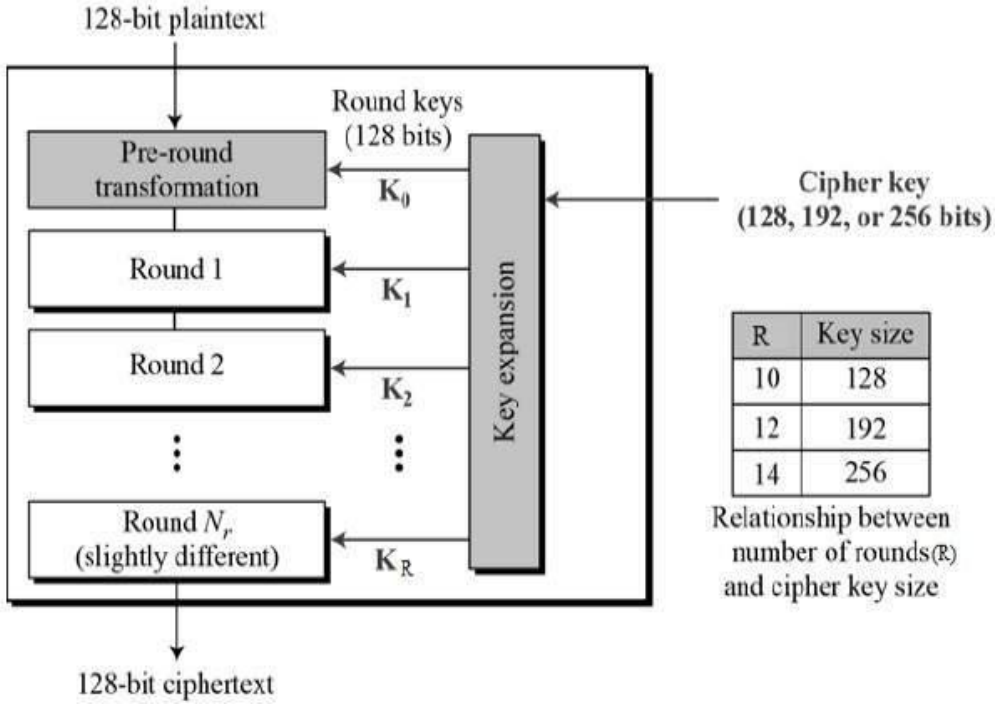
Examples of asymmetric encryption algorithms include the Rivest-Shamir-Adleman (RSA) algorithm and Elliptic Curve Cryptography (ECC).

Hybrid Encryption: Hybrid encryption combines both symmetric and asymmetric encryption to address the challenges of each approach. In this scheme, the sender generates a random symmetric key, also known as a session key, for data encryption. The actual data is encrypted using the symmetric key, which is much faster than asymmetric encryption. After encrypting the data, the sender then encrypts the symmetric key itself using the recipient's public key. This encrypted symmetric key is sent alongside the encrypted data. Upon receiving the message, the recipient uses their private key to decrypt the symmetric key and subsequently decrypt the data.

Hybrid encryption provides the best of both worlds: the efficiency of symmetric encryption and the security of asymmetric encryption. It ensures secure key exchange without the need for a secure channel to transmit the symmetric key.

Hashing: Hashing is a one-way process that converts data of any size into a fixed-size string of characters, known as a hash value or digest. Hash functions are designed to be irreversible, meaning it is computationally infeasible to obtain the original data from its hash value. Hashing is often used to verify the integrity of data, ensuring that the data has not been altered or tampered with during transmission or storage. Additionally, it is commonly used in password storage to securely store user passwords.

Examples of hashing algorithms include MD5, SHA-1, SHA-256, and SHA-3.



R	Key size
10	128
12	192
14	256

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

End-to-End Encryption: End-to-End Encryption (E2EE) is a communication method that ensures data is encrypted from the sender's device until it reaches the intended recipient's device. It means that the data is encrypted at the source and remains encrypted throughout its journey, ensuring that only the sender and the recipient can access the plaintext data. Even service providers or intermediaries cannot access the decrypted data. E2EE is widely used in messaging apps, email services, and cloud storage to provide enhanced privacy and security to users' data.
E2EE can be achieved using either symmetric or asymmetric encryption, or a combination of both (hybrid encryption).

Transport Layer Security (TLS) / Secure Sockets Layer (SSL): TLS and SSL are cryptographic protocols used to secure communication over the internet. They establish encrypted channels between web browsers and servers to ensure that data transmitted during online interactions remains confidential and secure. TLS has superseded SSL, but the term SSL is still used colloquially. TLS/SSL employs a combination of asymmetric and symmetric encryption during the initial handshake to establish a secure connection between client and server. Once the secure connection is established, data is transmitted using symmetric encryption, ensuring fast and secure communication.

Homomorphic Encryption: Homomorphic encryption is a specialized encryption technique that allows performing computations on encrypted data without decrypting it. It enables privacy-preserving data analysis, as data can remain encrypted while being processed by a third party. Homomorphic encryption has potential applications in secure cloud computing and data sharing, where data owners can share encrypted data with others for processing without exposing sensitive information.

While homomorphic encryption offers strong privacy guarantees, it is computationally intensive and slower compared to traditional encryption techniques. Ongoing research aims to improve the efficiency of homomorphic encryption for practical applications.