1. What is the key length of S-DES algorithm?
The key length of S-DES (Simplified DES) is **10 bits**.
From the 10-bit key, two 8-bit subkeys (K1 and K2) are generated through permutation and shifting.
These subkeys are used during the encryption and decryption process.

2. What is the difference between a block cipher and a stream cipher?
A block cipher encrypts fixed-size groups of bits (blocks), such as 64 or 128 bits at a time.
A stream cipher encrypts data bit by bit or byte by byte, ideal for real-time communications.
Block ciphers are generally more secure for bulk data, while stream ciphers are faster for continuous data.

3. What is some common application of S-DES?
S-DES is commonly used for **educational purposes** to explain how encryption algorithms work.
It helps students understand the structure of real encryption like DES without heavy complexity.
S-DES is not used in actual commercial or security applications because it is too simple.

4. What happens when you use a weak key with DES?
Using a weak key with DES can cause identical encryption and decryption behavior, weakening security.
It may produce predictable ciphertext patterns, making it easier for hackers to crack the message.
As a result, weak keys make the entire encryption process vulnerable to attacks.

1. Describe the process of AES key generation.
AES key generation involves taking the original cipher key and expanding it into multiple round keys.
This is done using a process called **key expansion**, which includes operations like SubWord, RotWord, and XOR with round constants.
These round keys are then used in each round of encryption and decryption.

2. What are the main stages of the AES algorithm?
The main stages of AES are **SubBytes**, **ShiftRows**, **MixColumns**, and **AddRoundKey**.
SubBytes substitutes each byte using an S-box; ShiftRows shifts rows of the matrix.
MixColumns mixes data within columns, and AddRoundKey combines data with the round key using XOR.

3. How does key expansion work in AES?
Key expansion starts by copying the original key into the beginning of the expanded key array.
It then generates new key words by transforming previous words using functions like SubWord and RotWord.
Each new word depends on the previous word and a fixed round constant to maintain strong security.

4. What differences exist between the three AES variants, AES-128, AES-192, and AES-256?
The main differences are the **key size** and **number of rounds**: AES-128 uses a 128-bit key and 10 rounds.
AES-192 uses a 192-bit key and 12 rounds, while AES-256 uses a 256-bit key and 14 rounds.
Higher key sizes offer greater security but may slightly impact performance.

1. Explain “Diffie-Hellman key exchange algorithm" with suitable example.
Diffie-Hellman is a method for two parties to securely share a secret key over a public channel.
For example, both pick private numbers, exchange calculated public values, and then compute the shared secret.
Even if an attacker sees the public values, without the private numbers, they can't find the secret key.

2. What is Man in the Middle attack?
A Man-in-the-Middle (MITM) attack happens when an attacker secretly intercepts and alters communication between two parties.
The attacker can read, insert, or modify messages without either party knowing.
This compromises the confidentiality and integrity of the data being exchanged.

3. How to Prevent a Man-in-the-Middle Attack?
Using **strong encryption (like HTTPS)** ensures that data is encrypted and harder to intercept.
Authentication mechanisms like digital certificates verify the identity of the communicating parties.
Avoiding public Wi-Fi and using VPNs also help in preventing MITM attacks.

1. Are strong primes necessary in RSA?
Strong primes make RSA keys harder to attack using certain mathematical methods like factorization.
However, with modern key sizes (like 2048 bits or higher), using random large primes is usually sufficient.
Thus, while helpful, strong primes are not absolutely necessary if the key size is large enough.

2. How fast is RSA?
RSA is relatively slow compared to symmetric encryption algorithms like AES.
It is mainly used for encrypting small pieces of data, like keys or digital signatures, not large files.
For bulk data, symmetric keys are exchanged using RSA and then faster algorithms handle the actual encryption.

3. What would it take to break RSA?
Breaking RSA requires factoring a very large number (hundreds of digits long) into its prime components.
This is computationally infeasible with current technology for properly sized keys (2048 or 3072 bits).
Quantum computers using Shor’s algorithm could break RSA, but practical quantum threats are still years away.

4. How is RSA used for authentication in practice?
RSA is used to create **digital signatures**, where a user signs a message with their private key.
The receiver verifies the signature with the sender’s public key to ensure authenticity and integrity.
It is widely used in secure websites (SSL/TLS), emails, and software signing.

1. What is the general form of an elliptic curve equation used in ECC?
The general form of an elliptic curve is $\mathbf{y^2 = x^3 + ax + b}$ over a finite field.
Here, "a" and "b" are constants that define the specific curve properties.
The curve must satisfy the condition $\mathbf{4a^3 + 27b^2 \neq 0}$ to avoid singular points.

2. How are public and private keys generated in ECC?
A private key is a randomly selected integer within a specific range.
The public key is generated by multiplying the private key with a known point (called the base point) on the curve.
This multiplication operation is known as **scalar multiplication**.

3. What is the process of point addition and point doubling in ECC?
Point addition combines two distinct points on the curve to find a third point.
Point doubling is a special case where a point is added to itself.
Both operations follow specific algebraic formulas based on the curve's equation.

4. How is the shared secret key computed between two parties using ECC (Elliptic Curve Diffie-Hellman)?
Each party multiplies their private key with the other party's public key.
Both calculations result in the same shared point on the curve, used to derive the secret key.
The secret key is then used for secure communication between the two parties.