Question no.1

Will the mechanism Message Integrity and Authentication Via Message Authentication Code (MAC) achieve the following services?

- Confidentially
  - ⇨ Message Integrity and Authentication Via Message Authentication Code (MAC) is a technique used to ensure that a message has not been tampered with or altered during transmission. This is accomplished by generating a unique code, or tag, based on the contents of the message and a secret key. The MAC is then appended to the message and sent to the recipient, who can then use the secret key to verify the authenticity and integrity of the message.

    However, this mechanism does not provide confidentiality, meaning that the message itself is not encrypted and can be read by anyone who intercepts

it. Confidentiality refers to the protection of the contents of a message from unauthorized access or exposure. This means that the message can still be intercepted and read by an attacker, who may be able to see sensitive or private information.

In conclusion, while MAC provides an important service of ensuring message integrity and authentication, it does not provide confidentiality, so additional measures must be taken to protect the contents of a message from unauthorized access or exposure.

- Integrity
  - Case 1: Eve changes msg, but not MAC.

⇨ Yes, that's correct. If an attacker, such as Eve, intercepts and try to modify a message which is protected by a MAC, they will not be able to modify the message without knowing the secret

key, which is used for the original message in the MAC. As a result, the recipient will detect tampering since the received MAC will not match the MAC calculated using the secret key and the received message.

This is because a MAC is a unique code generated based on the message's contents and the secret key. Even a small change to the message will result in a completely different MAC value. By verifying the MAC, the recipient can ensure that the message has not been altered in transit and it has arrived in its original form.

- Case 2: Eve change MAC, but not msg.

⇨ If an attacker like Eve tries to modify the MAC without changing the message, the recipient will still detect the modification. This is because the recipient will calculate a new MAC for

the received message using the secret key, and the newly calculated MAC will not match the modified MAC that was sent with the message. The recipient will then know that the message has been tampered with and will not trust the authenticity of the received message.

It's important to note that the MAC value acts as a digital signature for the message, providing a way to verify the authenticity and integrity of the message. If the MAC value is modified, even if the message itself is not changed, the recipient will know that the message has been tampered with and will not trust it.

- Eve changes both msg and MAC.

⇨ If an attacker, such as Eve, changes both the message and the MAC, the recipient will still detect the tampering. The recipient will calculate

a new MAC for the received message using the secret key, and the newly calculated MAC will not match the modified MAC that was sent with the message. The recipient will know that the message and the MAC have been tampered with and will not trust the authenticity or integrity of the received message.

It's important to use strong cryptographic techniques, such as secure key management and digital signatures, to protect against tampering and ensure messages' confidentiality, authenticity, and integrity during transmission.

Question no.2

What are the advantages and disadvantages of Symmetric-Key Cryptography?

- The advantages of Symmetric-Key Cryptography :

1. Speed: Symmetric key cryptography is much faster than asymmetric key cryptography, making it well-suited for the encryption and decryption of large amounts of data.
2. Simplicity: Symmetric key cryptography is much simpler to implement than asymmetric key cryptography, making it easier to use and less prone to implementation errors.
3. Security: Symmetric key cryptography algorithms are generally considered to be highly secure if the key used for encryption and decryption is kept secret.
4. Key management: In symmetric key cryptography, only one key is needed for encryption and decryption, making key management much simpler than

asymmetric key cryptography, where a different key is used for each direction of communication.

- Disadvantages of Symmetric-Key Cryptography:
  1. Key distribution: Symmetric key cryptography requires that the same secret key be shared between the sender and receiver, which can be a challenge when there are many parties involved in the communication.
  2. Key management: Symmetric key cryptography requires secure key management, as the same secret key is used for both encryption and decryption. If the key is lost, stolen, or otherwise compromised, the security of the encrypted data is at risk.
  3. Scalability: As the number of parties involved in the communication increases, the

number of keys required for secure communication also increases, which can become unwieldy and difficult to manage.

4. Security: Symmetric key cryptography can be less secure than asymmetric key cryptography, especially if the key used for encryption and decryption is not properly managed or if the encryption algorithm used is weak or vulnerable to attack.

Question no.3

Please explain how Kerberos solves the n^2 issue and key exchange issue.

⇨ Kerberos solves the "n-squared issue" and the key exchange issue by using a centralized authentication server to manage and distribute keys.

The "n-squared issue" is a problem in symmetric key cryptography where

the number of keys required for secure communication between n parties grows quadratically with the number of parties involved.

In other words, for n parties, n(n-1)/2 keys are required.

Kerberos solves this issue by using a centralized authentication server, known as the Key Distribution Center (KDC), to manage and distribute keys. The KDC is trusted by all parties in the network and acts as an intermediary to provide authentication and key distribution services.

When a client wants to communicate with a server, it first authenticates itself to the KDC, which then provides the client with a session key that can be used to encrypt communications between the client and server. This eliminates the need for direct key exchange between the client and

server and reduces the number of keys required to secure communication.
Kerberos also solves the key exchange issue by providing a secure and trusted mechanism for key distribution. The KDC acts as a trusted third party, ensuring that only authorized parties can access the shared keys. This eliminates the need for direct key exchange between parties, which can be vulnerable to interception or tampering by attackers.
We can explain with the example,
Suppose party A wants to talk to party B in the service. First party A and party B will send their password respectively to the KDC, which means Key Distribution Server. Then party A will send the request to KDC. Then, KDC will provide a ticket to party A where A can decrypt and encrypt the message. After that, KDC sends the ticket to party B along with an id. After

that, B can decrypt the ticket, which includes the secret key and id of A. Finally, parties A and B can securely communicate with one another.

Overall, Kerberos provides a secure and efficient solution for managing and distributing keys in symmetric key cryptography and helps to solve the "n-squared issue" and a key exchange issue.