Course Title: Data Integrity & Authentication

Instructor: Dr. Maged Abdelaty

Team Members:

- Maryam Waheed Zamel - 2205154
- Amina Ahmed Ferra - 2205225
- Mayssoune Hussein Elmasry - 2205251

Background Writeup

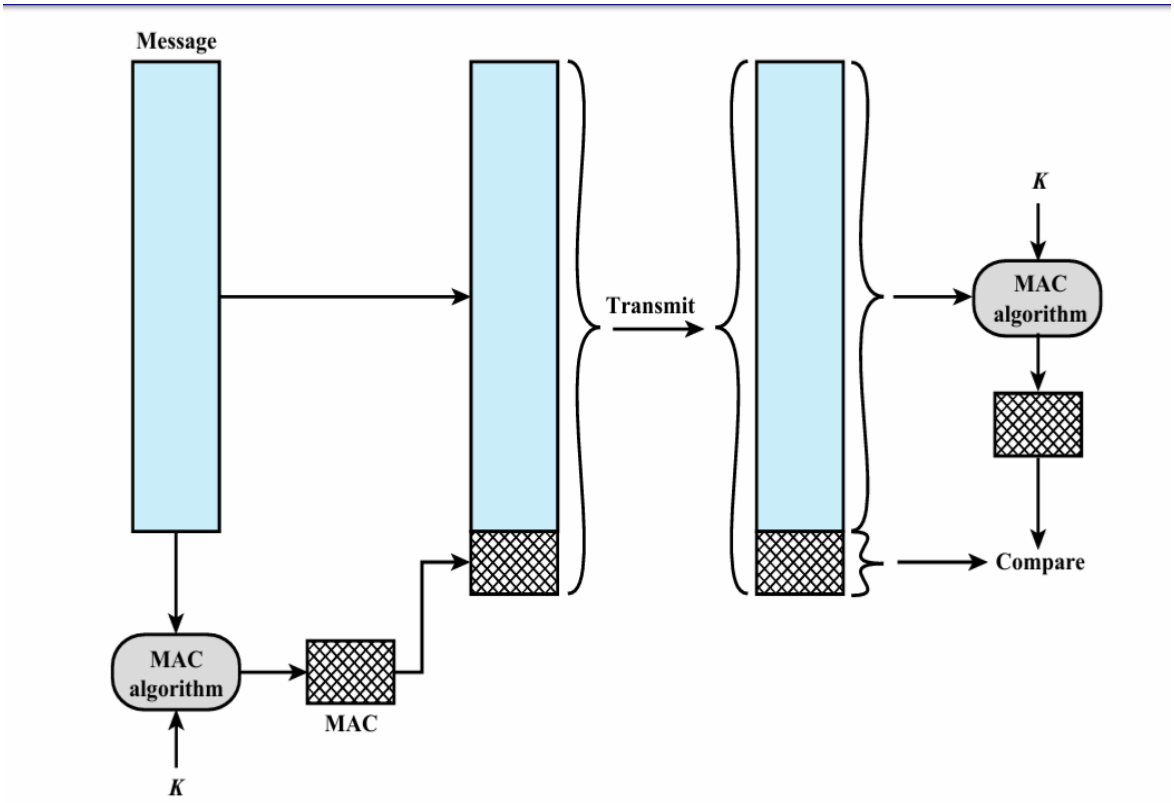
a. What is MAC and its purpose in data integrity/authentication?

Message Authentication Code (MAC) is a short piece of information derived from a message and a shared secret key using MAC algorithm. It is sent along with the message to the receiver, who uses the same key and algorithm to compute the MAC independently and verify the integrity and authenticity of the message.

Purpose of MAC:

- **Data Integrity:**
  MACs ensure that the message has not been tampered with during transmission. If even a single bit is altered, the computed MAC will not match the received MAC.

- **Authentication:**
  It authenticates the message's origin because only someone with access to the shared secret key can compute a valid MAC. This confirms that the message came from a legitimate sender.



Background Writeup

b. How does a length extension attack work in hash functions like MD5/SHA1?
Length extension attack targets insecure MAC constructions like MAC = hash (secret || message). Hash functions like MD5 and SHA1 process data in blocks and expose their internal state.

If an attacker knows the message and its MAC, they can:
- Guess the length of the secret.
- Append new data (e.g., &admin=true).
- Use the previous hash state to generate a valid MAC for the new message.

This works because these hash functions allow continued hashing from an exposed internal state. The forged MAC is accepted by the server even though the attacker doesn't know the secret.

c. Why is MAC = hash (secret || message) insecure?

Using MAC = hash(secret || message) may seem simple but is insecure when used with hash functions like MD5 or SHA1 because:

- It is vulnerable to length extension attacks, allowing attackers to forge valid MACs for extended messages.
- The hash function's internal state can be reused to compute hashes for longer messages.
- An attacker with the original message and MAC can generate a forged message + MAC that passes verification.

To avoid this, it's recommended to use HMAC, which applies two rounds of hashing with inner and outer pads.

HMAC is resistant to length extension and securely integrates the secret key.