Background – MAC Forgery and Length Extension Attack

1. What is a Message Authentication Code (MAC)?

A Message Authentication Code (MAC) is a short cryptographic checksum used to ensure the **integrity** and **authenticity** of a message. It is generated using a **secret key** and a **message**, and is verified on the receiver side by recomputing the MAC and comparing it with the one received. If the MACs match, the message is considered valid and untampered.

MACs are widely used in secure communications, including APIs, network protocols, and data validation processes.

2. What is a Length Extension Attack?

A length extension attack targets certain cryptographic hash functions (like **MD5** and **SHA1**) that are based on the **Merkle–Damgård construction**.

In such constructions, an attacker can exploit the fact that:

Hash(secret || message) leaks internal state that can be extended to compute Hash(secret || message || padding || new_data)

This allows an attacker who knows the original hash(secret || message) and the message to:

- Add extra data to the end of the message.
- Forge a valid hash for the extended message.
- Do all of this **without knowing the secret key**.

This kind of attack works because Merkle–Damgård hash functions allow you to continue hashing from a known internal state.

3. Why is MAC = hash(secret || message) Insecure?

The naive MAC construction MAC = hash(secret || message) is insecure **because it is vulnerable to length extension attacks**.

In this scheme:

- If an attacker can see the output of hash(secret || message), and knows message, they can exploit the hash's internal state to **forge a MAC** for message || extra_data.

This breaks the main purpose of a MAC — **ensuring authenticity and integrity** — since the attacker can craft a message that will be accepted by the system as valid, without knowledge of the secret key.