Data Integrity and Authentication

Background Write-Up: MAC Forgery and Length Extension Attacks

Submitted by:

Mohammed Abdulrahman Awad Khaled (ID: 2205114)

Mohammed Ahmed Ramadan Al-Arjawi (ID: 2205043)

Omar Ahmed Hameed Mohammed (ID: 2205213)

Submitted to:

Dr. Maged Abdelaty

May, 2025

1 Introduction to Message Authentication Codes (MACs)

A Message Authentication Code (MAC) is a cryptographic tool designed to guarantee the integrity and authenticity of a message. By combining a secret key with the message through a hash function or similar algorithm, a MAC produces a fixed-length tag. The recipient, possessing the same secret key, can recompute the MAC to confirm that the message is unaltered and originates from a trusted source. For instance, in a transaction such as `amount=100&to=alice`, a MAC prevents unauthorized modifications to the amount or recipient. MACs are essential in securing financial transactions, API authentication, and communication protocols.

2 Length Extension Attacks on Hash Functions

Hash functions like MD5 and SHA-1 employ the Merkle-Damgård construction, processing data in fixed-size blocks (512 bits for MD5) and updating an internal state. The final state forms the hash output. A length extension attack exploits this design: given a hash $H = \text{hash}(secret||message)$, an attacker can use H as the state after processing $secret||message$. By appending new data and calculating appropriate padding, the attacker can compute $\text{hash}(secret||message||p$ without the secret key. The attack requires:

- The original message and its hash.
- The length of $secret||message$ for correct padding.

MD5 padding adds a '1' bit (0x80), followed by zeros to align with 512 bits minus 64 bits, and a 64-bit length field. This vulnerability enables attackers to forge messages in systems using insecure MAC constructions.

3 Insecurity of Naive MAC Construction

The construction $\text{MAC} = \text{hash}(secret||message)$ is vulnerable to length extension attacks. The hash output reveals the internal state, allowing an attacker to append data and generate a valid MAC for the extended message without knowing the secret key. For example, given a valid pair (`amount=100&to=alice`, MD5($secret$||message)), an attacker can append `&admin=true` to create a new valid MAC. This undermines integrity (the message is modified) and authenticity (the forged message appears legitimate). Secure alternatives like HMAC, discussed in the mitigation write-up, prevent such attacks through a robust construction that protects the hash state.