Demonstrating and Mitigating a Message Integrity Attack (MAC Forgery)

Alexandria National University
Faculty of Computers and Data Science Cyber Security Program

DR:Maged Abdelaty

Mohamed Ahmed Aly Mobarak
ID:2205249

AMR KHALED
ID:2205220

MARIEM WALED
ID:2205184

What is a MAC and its purpose?

A Message Authentication Code (MAC) is a cryptographic checksum used to ensure data integrity and authentication. It is computed using a secret key and a hash function and appended to the message. The recipient can verify the MAC to ensure that the message has not been altered and comes from a trusted sender

MAC Forgery and Length Extension Attack

The vulnerability in MAC = hash(secret | message) lies in its exposure to length extension attacks. To mitigate this, we replace the naive construction with HMAC (Hash-based Message Authentication Code), which securely combines the key and message using a proven cryptographic design

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

This construction is insecure because it exposes the hashing process to manipulation. Hash functions like MD5 do not distinguish between the secret and the message during padding, allowing attackers to forge new messages that validate under the original MAC. The proper solution is to use HMAC, a construction specifically designed to prevent such attacks by separating the key and message inputs correctly

Fixing the Vulnerability

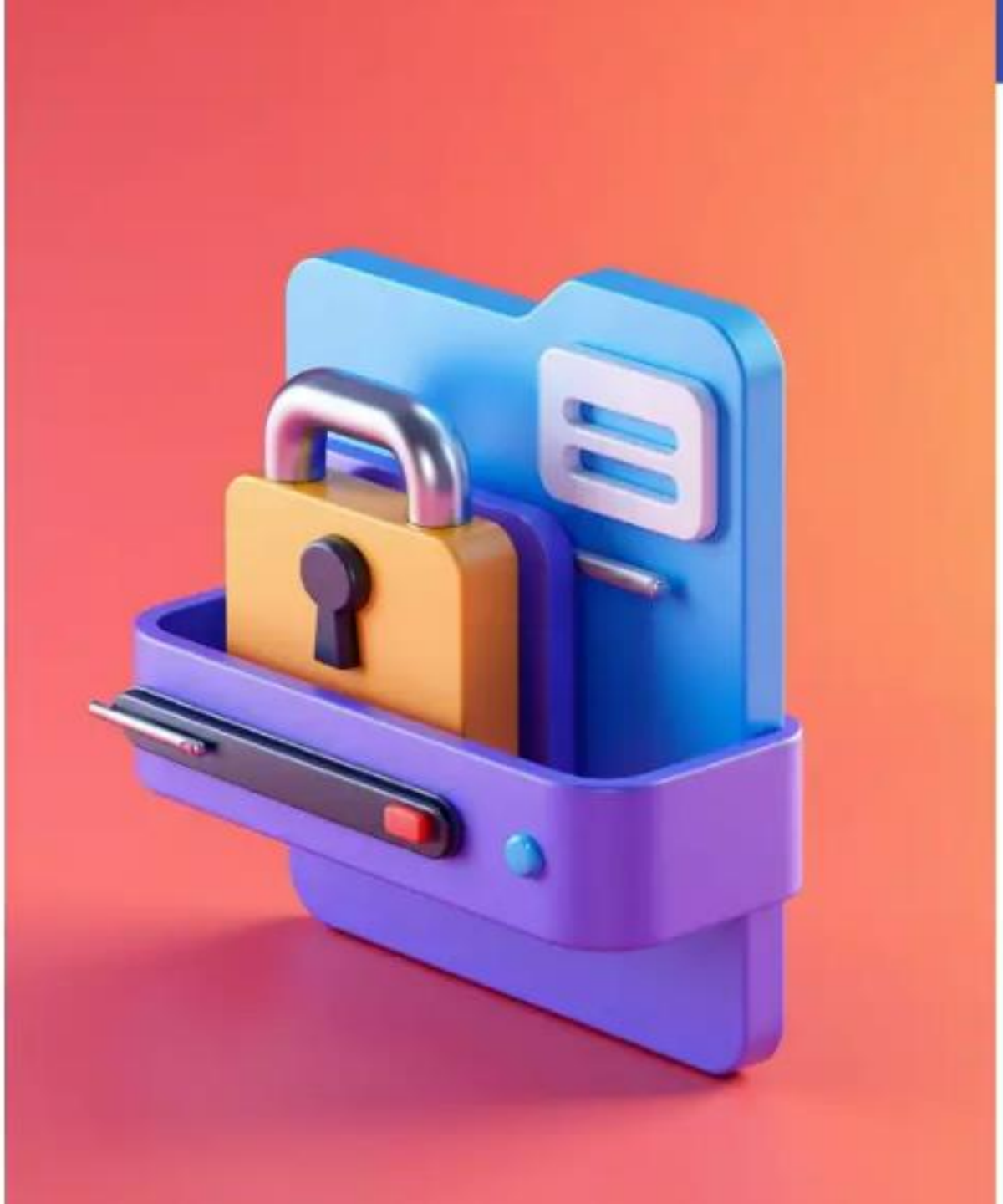
The vulnerability in MAC = hash(secret | message) lies in its exposure to length extension attacks. To mitigate this, we replace the naive construction with HMAC (Hash-based Message Authentication Code), which securely combines the key and message using a proven cryptographic design

What is HMAC?

HMAC stands for Hash-based Message Authentication Code. It uses a cryptographic hash function along with a secret key in a way that separates key processing from the message content. The formula is: HMAC(key, message) = H((key XOR opad) | H((key XOR ipad) | message)). This design ensures the integrity and authenticity of the message, even against length extension attacks

HMAC prevents attacks ensuring data integrity

HMAC prevents length extension attacks because the internal state of the hash function is tied to a padded and hashed key. Even if an attacker knows the message and the original HMAC, they cannot append data or generate a valid HMAC without knowing the secret key.
The separation between key and message, along with the double hashing structure, makes it cryptographically secure



Demonstrating the Defense

After switching the server-side MAC generation from hash(secret | message) to HMAC, we rerun the same attack script. The result is a failure to verify the forged message and MAC, demonstrating that the attack no longer works. This proves that HMAC is a secure alternative and an effective defense against MAC forgery via length extension


ALEXANDRIA NATIONAL UNIVERSITY
جامعة الإسكندرية الأهلية