Feb 14, 2016: MAC and CPA/CCA Security

February 15, 2017

MAC

Message Authentication Codes(MAC) are used to prevent attackers from forging messages. The MAC is calculated by the sender using the message contents and a shared key. Upon recieving the message, the recipient calculates a MAC on their own and confirms that it matches the one that accompanies the message.

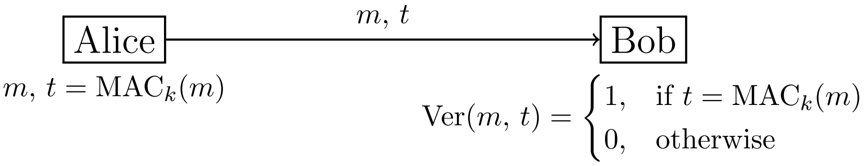

Alice
m, t
Bob
$m, t = \mathrm{MAC}_k(m)$
$\mathrm{Ver}(m, t) = \begin{cases} 1, & \text{if } t = \mathrm{MAC}_k(m) \\ 0, & \text{otherwise} \end{cases}$

Figure 1: Bob verifies that Alice sent m by calculating $\mathrm{MAC}_k(m)$ on his own, and ensuring that it matches the t sent by Alice

Mac needs to be deterministic since the recipient must be able to calculate the same t as the sender.

MAC Schemes

MD5($k \parallel m$)

This is MAC scheme is easily breakable using a length extension attack. The attacker can pad and modify an intercepted message and update the hash to reflect the changes without knowing the key. The recipient's calculation of MAC for the tampered message will match the hash.

HMAC

HMAC is a much better scheme, which is believed to be secure. For HMAC:

$$\mathrm{MAC}_k(m) = \mathrm{hash}(k \oplus \mathrm{st1} \parallel \mathrm{hash}(k \oplus \mathrm{st2} \parallel m))$$

With HMAC an attacker cannot perform a length extension in the same way because HMAC involves two layers of hashing - both involving the secret key.
Even with an MD5 hashing function, HMAC is thought to be secure, although it is recommended to use a stronger hashing function such as SHA256.

Security Definition for MAC

To define secure MAC, model a game where a forger has access to MAC and Verification oracles. The first computes and returns the MAC for any message the forger selects. The Verification oracle accpets a message and MAC and responds with a confirmation(or denial) that the MAC correspondds to the message. To win the game, the forger must provide a message and tag(that he has not submitted to the MAC oracle) which pass verification This definition is strong and sufficient because a forger can choose any message. With a secure scheme, the best the forger can do is select a random HMAC.

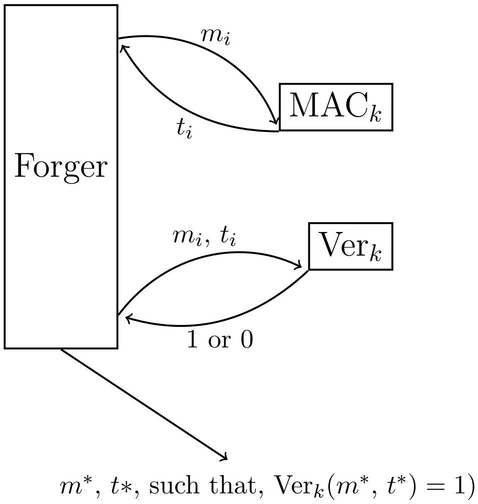


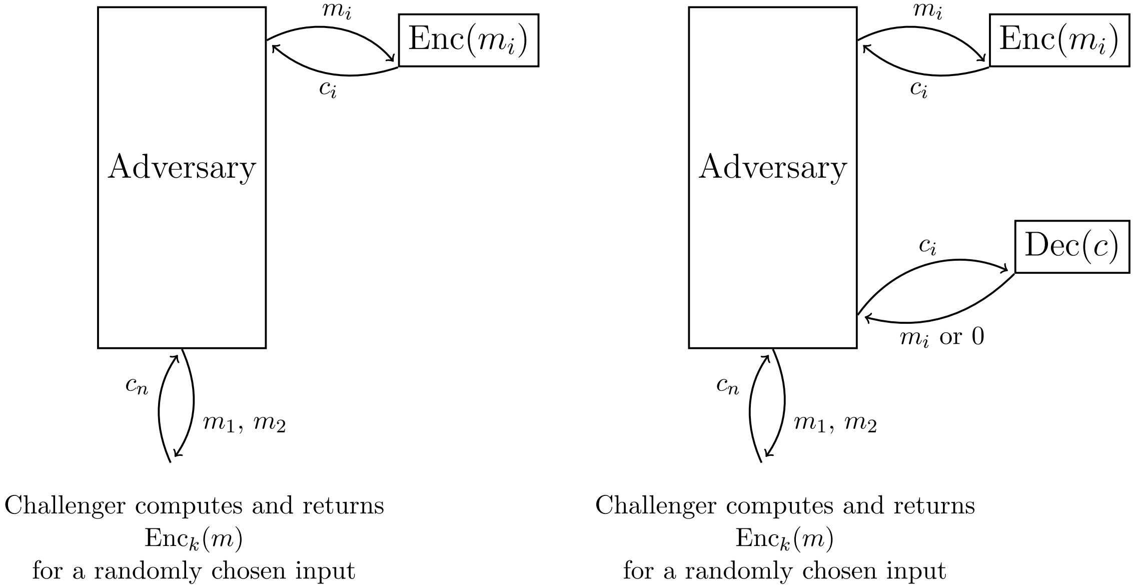
Figure 2: This game is sufficient to define a secure MAC. An attacker could win the game if the MAC scheme is MD5($k \parallel m$) using length extension

CPA and CCA security

CPA security is defined by a scenario where an attacker has access to an Encryption oracle and must have a greater than 0.5 probability of correctly guessing which of two message a challenger encrypted.
CCA secuirty is defined by a game which extends the CPA game to allow the attacker access to a Decryption Oracle.
See Figure 3 Because the conditions of the CCA game provide the Attacker with more resources than the CPA game, CPA security does not imply CCA security.

Achieving CCA Security

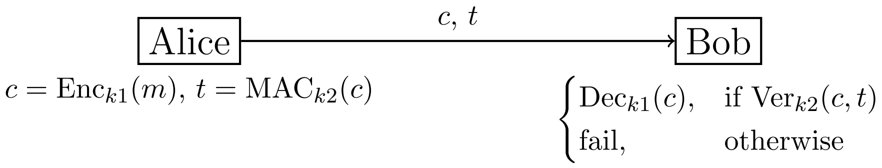
To achieve CCA security, a CPA secure scheme must be extended with a secure MAC. Let Enc, Dec be a CPA secure scheme(eg. CBC mode). Let MAC, Ver be a secure MAC. To make a CCA scheme, augment Enc to calculate and append a $\mathrm{MAC}_{k2}(m)$ to the encrypted message as t using a second key k_2. Modify Dec to Verify that $t = \mathrm{MAC}_{k2}(m)$ prior to proceding with the decryption. See Figure 4.
In practice, Galois Counter Mode(GCM) is used today.
Takeaway: Anytime we encrypt we need MAC to ensure confidentiality.



(a) CPA security Game

(b) CCA security Game

Figure 3: To win the games the Adversary must correctly guess which message the Challenger encrypted with a greater than 0.5 probabilty of success.



Alice
c, t
Bob
$c = \mathrm{Enc}_{k1}(m)$, $t = \mathrm{MAC}_{k2}(c)$
$\begin{cases} \mathrm{Dec}_{k1}(c), & \text{if } \mathrm{Ver}_{k2}(c,t) \\ \text{fail}, & \text{otherwise} \end{cases}$

Figure 4: A CCA Secure Transaction