1 Message Authentication Code (MAC)

1.1 Authentication Protocol

Since we saw encryption is not enough for data integrity, we need some authentication protocol. The components of this message authentication protocol is as follows:

- A key generation algorithm that returns a secret key k
- A Mac generating algorithm that returns a tag for a given message m. Tag $t = MAC_k(m)$
- A verification algorithm that returns a bit $b = Verify_k = (m_1, t_1)$ and a tag t_1
- If the message is not modified then with high probability, the value **b** is true, otherwise false.

1.2 Construction of MAC using PRF

- Gen (1^n) chooses k to be a random n-bit string
- MAC $k(m) = Fk(m) = t$ (the tag)
- Verify $k(m, t) = Accept$, if and only if $t = Fk(m)$

2 CBC-MAC construction

CBC-MAC is fairly similar to the original CBC mode for encryption. The Initialization Vector (IV) is a fixed value, usually zero. CBC-MAC only outputs the cipher-text's final block, which serves as the MAC.

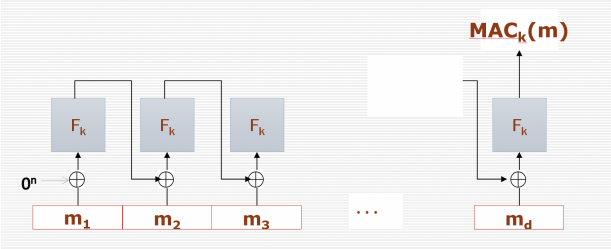

Figure 1: CBC-MAC scheme

There is a simple attack that allows us to forge new messages.

- First we get a MAC t on message m_1

- Now we do XOR the tag t into the first block of some arbitrary second message m_2, and get a MAC on the modified version of m_2.
- The resulting tag $t^{'}$ turns out to be a valid MAC for message $(m1||m2)$

The standard fix to pre-pend the message length to the first block of the message before MAC-ing it, as shown below:

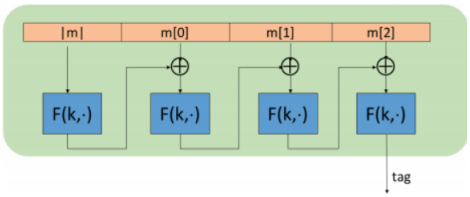

Figure 2: CBC-MAC scheme handling variable length messages