MAC (Message Aunthentican Codes)

The aim of the message authentication code is to present an adversary from modifying a message sent by one party to another without detecting that the modification has been made.

<u>Definition</u> :- A MAC is a tuple of probabilitie polynomial time algorithms (Gen, Mac, Vrfy) fulfilling the following:

1.) Upon input 1^n, the algorithm Gen outputs a uniformly distributed key k of length n ; $k \leftarrow Gen(1^n)$.

2.) Algorithm MAC receives for input, $k \in \{0,1\}^n$ and $m \in \{0,1\}^*$ and outputs $t \in \{0,1\}^*$. The value of t is called MAC tag.

3.) The algorithm Vrfy receives for input $k \in \{0,1\}^n$, $m \in \{0,1\}^*$ & $t \in \{0,1\}^*$ and output a bit $b \in \{0,1\}$.

4.) For every n, every $k \in \{0,1\}^n$ & every $m \in \{0,1\}^*$ it holds that

$$Vrfy_k(m, MAC_k(m)) = 1$$

If there exists a function $l(\cdot)$ such that $MAC_k(\cdot)$ is defined only over messages of length $l(n)$ & $Vrfy_k(m,t)$ outputs 0 for every m that is not of length $l(n)$, then we say that (Gen, Mac, Vrfy) is a fixed length MAC with length parameter l.

The idea behind the security of MAC is that no polynomial time adversary should be able to generate ~~a~~ a valid MAC tag on any new message.

Fixed length MAC using PRF:-

Let function F is a pseudorandom function. Define fixed length MAC as follows

- $Gen(1^n)$ - upon input 1^n choose $k \leftarrow \{0,1\}^n$
- $Mac_k(m)$ - upon input key $k \in \{0,1\}^n$, message $m \in \{0,1\}^n$, compute

$$t = F_k(m) \quad [\,|k| = |m|\,]$$

$Vrfy_k(m,t)$ - upon input key $k \in \{0,1\}^n$,

message $m \in \{0,1\}^n$ & tag $t \in \{0,1\}^n$,
output 1 iff $t = F_k(m)$.

To prove :- Above construction results in secure MAC.

Proof :- This can be proved if we prove that fixed length msg authentica^n code with length parameter $l(n) = n$ is existentially unforgeable under chosen ~~msg~~ msg attack.
Let A be a probabilistic polynomial time adversary & ~~&~~ let $\epsilon(\cdot)$ be a function so that

$$Pr[\text{Mac-forge}_{A,\pi}(n) = 1] = \epsilon(n).$$

This implies the existence of a polynomial time algorithm that can distinguish the pseudorandom from a random one with advantage $\epsilon(n)$. This will imply that ϵ must be negligible as required.
Consider MAC $\tilde{\pi} = (\tilde{Gen}, \tilde{Enc}, \tilde{Dec})$ which is same as $\pi = (Gen, Enc, Dec)$ with truly random function instead of PRF.

$$Pr[\text{Mac forge}_{A,\tilde{\pi}}(n) = 1] \leq 1/2^n$$

because for any msg $m \notin q$, the value $t = f_n(m)$ is uniformly distributed in $\{0,1\}^n$ from the point of view of A.
Construct a polynomial-time distinguished D, upon input 1^n algorithm D involves A upon input 1^n. Then, when A queries its oracle with a message m', D queries its oracle with m' & ~~&~~ set t' to be the oracle reply. D hands t' to A & continues. At the end when A output a pair (m, t),

distinguisher D checks that m was not asked during execution (i.e. $m \notin \mathcal{Q}$) & that t is a "valid" MAC. D does this by querying m to it's oracle & checking that the response equals t. If both of the above check pass, then D outputs 1 otherwise 0.

$$\Pr[D^{F_k(\cdot)}(1^n) = 1] = \Pr[\text{Mac-Forge}_{A,\Pi}(n) = 1]$$
$$= \epsilon(n)$$

$$\Pr[D^{f_n(\cdot)}(1^n) = 1] = \Pr[\text{Mac-forge}_{A,\tilde{\Pi}}(n) = 1] \leq \frac{1}{2^n}$$

$$\therefore \left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f_n(\cdot)}(1^n) = 1]\right| \geq \epsilon(n) - \frac{1}{2^n}$$

$\because$ F is a pseudorandom function, $\epsilon(n) - \frac{1}{2^n}$ must be negligible.

$\Rightarrow$ $\epsilon(\cdot)$ must be negligible function.

$\Rightarrow$ A succeeds in Mac-forge with at most negligible probability & MAC constructed above is existentially unforgeable under chosen message attacks, i.e. provably secure.