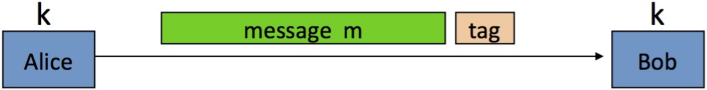
Message Authentication Codes - RECAP

Definition of Message Authentication Code (MAC)

A **MAC**$= (S, V)$ is a pair of algorithms defined over $(\mathcal{K}, \mathcal{M}, \mathcal{T})$ with the following properties:

- $S : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a **signing** algorithm that takes as input a key k and a message m and outputs a tag $t = S(k, m)$.
- $V : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{\text{yes}, \text{no}\}$ is a **verification** algorithm that checks if t is a *valid* tag for m under the key k. If so, the verification outputs **“yes”**, otherwise it outputs **“no”**.

Consistency requirement: $\forall k \in \mathcal{K}, \forall m \in \mathcal{M} : \quad V\Big(k, m, S(k, m)\Big) = \text{yes}$



Generate tag:
tag $\leftarrow$ S(k, m)

Verify tag:
V(k, m, tag) $\stackrel{?}{=}$ ‘yes’

The main property of MACs is **integrity**: *without* knowing the secret key k it is *hard* to generate a *valid* tag t^* for a *new* message.

Security Game for MACs (chosen message attack)

$$\begin{array}{ccc}
\mathcal{A} & & \mathcal{C} \\
\text{choose } m_i, & & k \\
i = 1, \ldots, q \in \mathcal{M} & \xrightarrow{m_i} & \\
& & \text{compute } S(k, m_i) = t_i \\
& \xleftarrow{t_i} & \\
(m^*, t^*) \neq (m_i, t_i) & & \\
& \xrightarrow{(m^*, t^*)} & \\
& & V(k, m^*, t^*) = D
\end{array}$$

If $D =$ "yes" then $\mathcal{A}$ wins the security game (i.e. the MAC is **not** secure against chosen message attack). If $D =$"no", $\mathcal{A}$ has lost the security game (i.e. the MAC is secure).

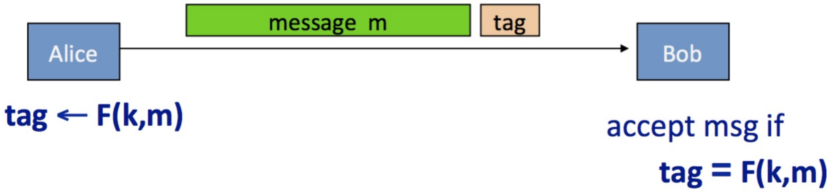
How to build a MAC from a block cipher?

Secure PRF $\Rightarrow$ Secure MAC

For a PRF $F : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{T}$ define a $MAC = (S, V)$ as

- $S(k, m) := F(k, m) = t$.
- $V(k, m, t)$ output **"yes"** if $t = F(k, m)$ and **"no"** otherwise.

If $|T|$ is large and the PRF is **secure** then the MAC is also **secure**.



How to build a MAC from a Hash function?

HMAC (Hash-MAC)

HMAC are the most widely used MAC on the Internet. Let H be a hash function (e.g. H is SHA-256), define a HMAC as

$$HMAC : S(k, m) = H(k \oplus \text{ opad}||H(k \oplus \text{ipad}||m)$$

Where opad and ipad are respectively an outer and an inner pad. Both pads are fixed constants of size equal to the block size for H