Message integrity: Confidentiality alone not sufficient, also need message integrity. Otherwise adversary can tamper with the message (e.g., "Send \$100 to Bob" $\to$ "Send \$100 to Eve")

In some cases (e.g., software patches), integrity more important than confidentiality

Idea: Append a "tag" (also called a "signature") to the message to prove integrity (property we want is tags should be hard to forge)

Observation: The tag should be computed using a keyed-function

↳ Example of keyless integrity check: CRC (cyclic redundancy check) [simple example is to set tag to be the parity]

this tolerates a single error (better error-correcting codes can do much better)

↳ this was used in SSH v1 (1995) for data integrity! Fixed in SSH v2 (1996)

↳ also used in WEP (802.11b) protocol for integrity — also broken!

Problem: If there is no key, anyone can compute it! Adversary can tamper with message and compute the new tag.

Definition. A message authentication code (MAC) with key-space $\mathcal{K}$, message space $\mathcal{M}$ and tag space $\mathcal{T}$ is a tuple of algorithms $\Pi_{MAC} = (\text{Sign}, \text{Verify})$:

$$\left.\begin{array}{l}\text{Sign}: \mathcal{K} \times \mathcal{M} \to \mathcal{T} \\ \text{Verify}: \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0,1\}\end{array}\right\} \text{Must be efficiently-computable}$$

Correctness: $\forall k \in \mathcal{K}, \forall m \in \mathcal{M}$:

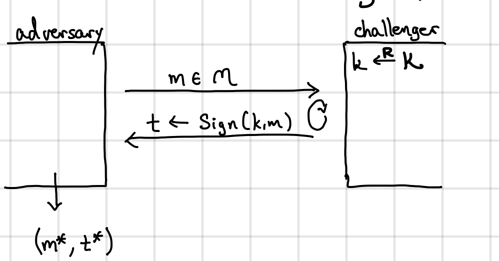
$$\Pr[\text{Verify}(k, m, \text{Sign}(k,m)) = 1] = 1$$

Sign can be a randomized algorithm

Defining security: Intuitively, adversary should not be able to compute a tag on any message without knowledge of the key

↳ Moreover, since adversary might be able to see tags on existing messages (e.g., signed software updates), it should not help towards creating a new MAC

Definition. A MAC $\Pi_{MAC} = (\text{Sign}, \text{Verify})$ satisfies existential unforgeability against $\underbrace{\text{chosen message attacks}}_{\text{adversary gets to choose messages to be signed}}$ (EUF-CMA) if for all efficient adversaries $\mathcal{A}$, $\text{MACAdv}[\mathcal{A}, \Pi_{MAC}] = \Pr[W=1] = \text{negl}(\lambda)$, where W is the output of the following security game:



As usual, λ denotes the length of the MAC secret key (e.g., $\log|\mathcal{K}| = \text{poly}(\lambda)$)

Note: the key can also be sampled by a special KeyGen algorithm (for simplicity, we just define it to be uniformly random)

Let $m_1, \ldots, m_Q$ be the signing queries the adversary submits to the challenger, and let $t_i \leftarrow \text{Sign}(k, m_i)$ be the challenger's responses. Then, $W = 1$ if and only if:

$$\text{Verify}(k, m^*, t^*) = 1 \text{ and } (m^*, t^*) \notin \{(m_1, t_1), \ldots, (m_Q, t_Q)\}$$

MAC security notion says that adversary cannot produce a new tag on any message even if it gets to obtain tags on messages of its choosing.

First, we show that we can directly construct a MAC from any PRF.

<u>MACs from PRFs</u>: Let $F: \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be a PRF. We construct a MAC Π_{MAC} over $(\mathcal{K}, \mathcal{M}, \mathcal{T})$ as follows:

Sign(k, m): output $t \leftarrow F(k, m)$

Verify(k, m, t): output 1 if $t = F(k, m)$ and 0 otherwise

<u>Theorem</u>. If F is a secure PRF with a sufficiently large range, then Π_{MAC} defined above is a secure MAC. Specifically, for every efficient MAC adversary $\mathcal{A}$, there exists an efficient PRF adversary $\mathcal{B}$ such that

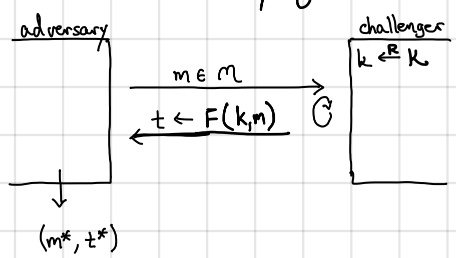
$$\mathrm{MACAdv}[\mathcal{A}, \Pi_{MAC}] \leq \mathrm{PRFAdv}[\mathcal{B}, F] + \frac{1}{|\mathcal{T}|}.$$

<u>Intuition for proof</u>: 1. Output of PRF is computationally indistinguishable from that of a truly random function.

2. If we replace the PRF with a truly random function, adversary wins the MAC game only if it correctly predicts the random function at a new point. Success probability is then exactly $1/|\mathcal{T}|$.

<u>Proof</u>. We define the following sequence of hybrid experiments:

Hyb_0: This is the MAC security game:



Goal: Show for all efficient $\mathcal{A}$:

$$\Pr[\mathrm{Hyb}_0(\mathcal{A}) = 1] = \mathrm{negl}.$$

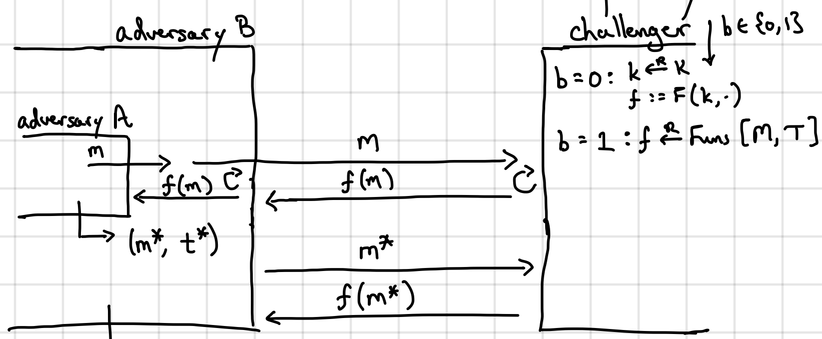
Experiment outputs 1 if adversary did not query on m^* and $t^* = F(k, m^*)$

Hyb_1: Same as Hyb_0 except we replace $F(k, \cdot)$ with $f(\cdot)$ where $f \xleftarrow{R} \mathrm{Funs}[\mathcal{M}, \mathcal{T}]$

<u>Lemma 1</u>. If F is a secure PRF, then for all efficient adversaries $\mathcal{A}$,

$$|\Pr[\mathrm{Hyb}_0(\mathcal{A}) = 1] - \Pr[\mathrm{Hyb}_1(\mathcal{A}) = 1]| = \mathrm{negl}.$$

<u>Proof</u>. Suppose there exists efficient $\mathcal{A}$ such that above probability is ε. We construct $\mathcal{B}$ as follows:



output 1 if $\mathcal{A}$ does not query on m^* and $t^* = f(m^*)$

$$\left.\begin{array}{l} \Pr[\mathcal{B} \text{ outputs } 1 \mid b = 0] = \Pr[\mathrm{Hyb}_0(\mathcal{A}) = 1] \\ \Pr[\mathcal{B} \text{ outputs } 1 \mid b = 1] = \Pr[\mathrm{Hyb}_1(\mathcal{A}) = 1] \end{array}\right\} \mathrm{PRFAdv}[\mathcal{B}, F] = \varepsilon$$

<u>Lemma 2</u>. For all adversaries $\mathcal{A}$, $\Pr[\mathrm{Hyb}_1(\mathcal{A}) = 1] = \frac{1}{|\mathcal{T}|}$.

<u>Proof</u>. $\mathrm{Hyb}_1(\mathcal{A})$ outputs 1 if $\mathcal{A}$ predicts value of f at m^*. Since f is uniform, $\mathcal{A}$ succeeds with probability at most $1/|\mathcal{T}|$.

Implication: Any PRF with large output space can be used as a MAC.

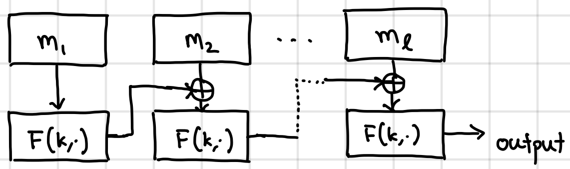
↳ AES has 128-bit output space, so can be used as a MAC

Drawback: Domain of AES is 128-bits, so can only sign 128-bit (16-byte) messages

How do we sign longer messages? We will look at two types of constructions:

1. Constructing a large-domain PRF from a small-domain PRF (i.e., AES)
2. Hash-based constructions

Approach 1: use CBC (without IV)



Not encrypting messages so no need for IV (or intermediate blocks)

↳ Mode often called "raw-CBC"

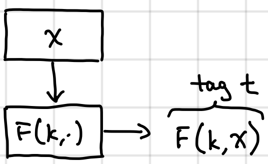
Raw-CBC is a way to build a large-domain PRF from a small-domain one

↳ Can show security for "prefix-free" messages [more precisely, raw-CBC is a prefix-free PRF: pseudorandom as long as PRF never evaluated on two values where one is a prefix of other]

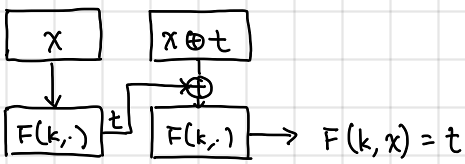
↳ includes fixed-length messages as a special case

But not secure for variable-length messages: "Extension attack"

1. Query for MAC on arbitrary block x:



2. Output forgery on message $(x, x \oplus t)$ and tag t



$\Rightarrow$ t is a valid tag on extended message $(x, t \oplus x)$

↳ Adversary succeed with advantage 1