Message Authentication Codes, MACs

The goal of MACs is to provide *integrity* and *authenticity*.

Definition 1. A *MAC* is a triple of poly-time algorithms

$$(\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verif})$$

such that:

- ▷ $\mathrm{KeyGen}(1^\lambda)$ takes as input the security parameter (in unary) and outputs a key $k \in \{0,1\}^s$;
- ▷ $\mathrm{Sign}(k, \mu)$ takes as inputs a key k, and a message $\mu \in \{0,1\}^n$, and outputs a tag $t \in \{0,1\}^m$;
- ▷ $\mathrm{Verify}(k, \mu, t)$ that takes as input a key k, a message μ and a tag t, and outputs a bit in $\{0,1\}$.

We say that a MAC is *correct* if, for every key k output by KeyGen, for all message μ,

$$\mathrm{Verify}(k, \mu, \mathrm{Sign}(k, \mu)) = 1.$$

The security is defined with an experiment:

- ▷ A challenger $\mathscr{C}$ creates a key k with $\mathrm{KeyGen}(\ )$.
- ▷ An adversary $\mathscr{A}$ gives a message μ_1 to $\mathscr{C}$.
- ▷ Then $\mathscr{C}$ sends back $t_1 := \mathrm{Sign}(k, \mu_1)$.

- After, $\mathscr{A}$ gives a message μ_2 to $\mathscr{C}$.
- And $\mathscr{C}$ sends back $t_2 := \mathrm{Sign}(k, \mu_2)$.
- *etc.*
- Finally, $\mathscr{A}$ sends a pair $(\mu^\star, t^\star)$ to $\mathscr{C}$.

The goal of $\mathscr{A}$ is to create (forge) a new valid message-tag pair. The adversary $\mathscr{A}$ will win if $\mathrm{Verify}(k, \mu^\star, t^\star) = 1$ and $(\mu^\star, t^\star) \neq (\mu_i, t_i)$ for every i.

The MAC is secure if, for any poly-time adversary $\mathscr{A}$, the probability that $\mathscr{A}$ wins is negligible. We call this *sEU-CMA security* (strong existential unforgeability under chosen message attacks).

We also define *EU-CMA* security: it is a variant where the success conditions are

$$\mathrm{Verify}(k, \mu^\star, t^\star) = 1 \qquad \text{and} \qquad \mu^\star \neq \mu_i \quad \forall i.$$

We have that sEU-CMA security implies EU-CMA security.

PRF-base MAC for fixed-length messages.

We can proceed like the following:

- KeyGen( ), it samples $k \leftarrow \mathscr{U}(\{0,1\}^s)$;
- $\mathrm{Sign}(k, \mu)$, it returns $t \leftarrow F(k, \mu)$;
- $\mathrm{Verify}(k, \mu, t)$, it tests if $t \stackrel{?}{=} F(k, \mu)$.

This way, a PRF is a MAC.

Why is it a secure MAC ? Let's assume we have a sEU-CMA adversary $\mathscr{A}$ and see if we can use it to break the PRF.

Consider the experiment Exp_0—the genuine sEU-CMA experiment—where $\mathscr{C}$ samples a key $k \leftarrow \mathscr{U}(\{0,1\}^s)$, then $\mathscr{A}$ makes queries μ_i (than can depend on results of previous ones) and gets back $t_i \leftarrow F(k, \mu_i)$.

Finally $\mathscr{A}$ sends $\mathscr{C}$ a "forged signature" $(\mu^\star, t^\star)$. The adversary will win if $F(k, \mu^\star) = t^\star$ and $(\mu_i, t_i) \neq (\mu^\star, t^\star)$.

Now, consider experiment Exp_1, where $\mathscr{C}$ (lazily) gets a uniform $f : \{0,1\}^n \to \{0,1\}^m$. When answering $\mathscr{A}$'s queries, $\mathscr{C}$ will use $t_i \leftarrow f(\mu_i)$. Finally $\mathscr{A}$ sends $\mathscr{C}$ a "forged signature" $(\mu^\star, t^\star)$. The adversary will win if $f(\mu^\star) = t^\star$ and $(\mu_i, t_i) \neq (\mu^\star, t^\star)$.

I will stop taking notes for the Cryptography and Security course, as I will no longer be following it. Some great lecture notes can be found in the AliENS GitLab (ENS students only):

`https://gitlab.aliens-lyon.fr/di-students/cours-m1/-/tree/2020-2021/s2/CS/2019-2020`

Farewell everyone!