Authentication

Recommended reading:

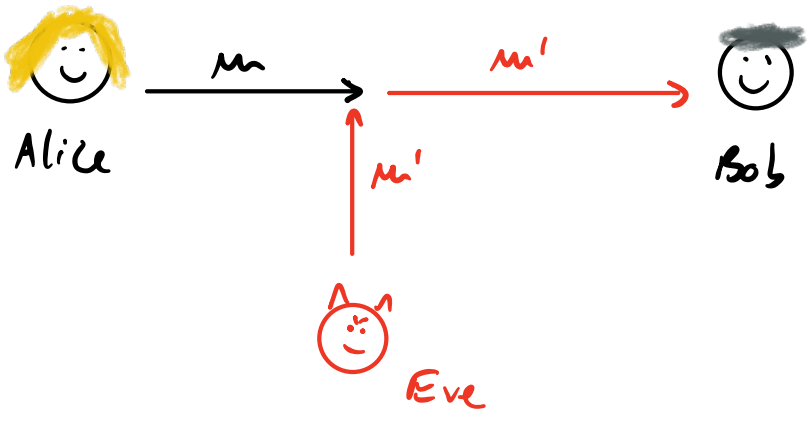
KL, Sections 4.1, 4.2, 4.3.1

Barak, Sections 4.2, 4.3

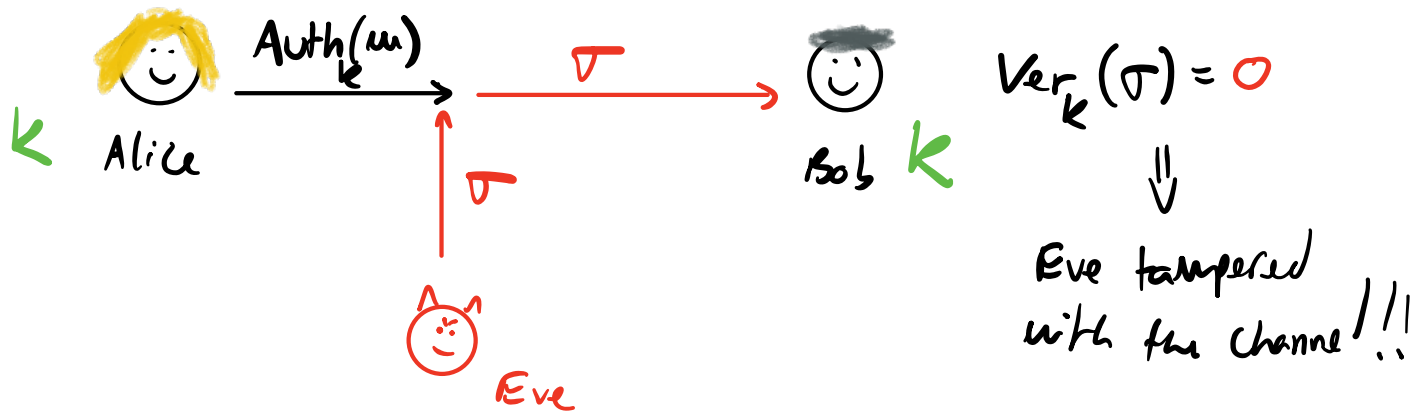
Up until now we have been discussing how to hide messages from an eavesdropper. However, we have not taken into account the integrity of the messages being transmitted. For example, what if the adversary replaces the message being transmitted by one of their own? How can we detect this?

This is the problem of authentication!

Setting: we want to avoid this



We assume, as before, that Alice and Bob share a secret key



Note that here we don't enforce privacy! Authentication is a problem separate from privacy, although sometimes people erroneously conflate the two. Encryption schemes don't provide authentication by default.

Let's define the desired objects more formally.

Def (Message authentication code).

A message authentication code (MAC) is a tuple of algorithms (Gen, MAC, Ver) such that:

- → Gen is the PPT key generation algorithm that samples $k \leftarrow Gen(1^n)$, $|k| \geq n$.
- → MAC is the "tag generation" PPT algorithm that on input a key k and a message m outputs a tag $t \leftarrow MAC(k, m)$.
  ↳ $\in \{0,1\}^{\ell(n)}$

→ Ver is the deterministic polynomial-time algorithm that takes as input a key k, message m, and tag t. It outputs $b = Ver(k, m, t) \in \{0,1\}$.

We require correctness: True tags verify correctly.

For every message m, key k,

$$Pr(Ver(k, m, MAC(k,m)) = 1) = 1.$$

Canonical verification: If MAC is deterministic, we can verify a tag t by recomputing $\tilde{t} = MAC(k,m)$ and checking if $\tilde{t} = t$. This is called canonical verification.

How can we define security for MACs?

Intuition: An adversary that sees many tagged messages should not be able to produce a tag for a new message

The MAC experiment

Parameterized by the security parameter n

- A key is generated by running $k \leftarrow Gen(1^n)$.
- The adversary $A(1^n)$ gets to interact with a "MAC oracle" $O_{MAC,k}$ that on input m outputs $t = MAC(k, m)$.

Let Q denote the set of all queries made by A.

- A outputs a pair (m', t') and wins if $m' \notin Q$ (i.e., A didn't ask the oracle about m') and $Ver(k, m', t') = 1$. (t' is a valid tag for m').

Def (EUFCMA security). We say that (Gen, MAC, Ver) is EUFCMA-secure, or just <u>secure</u>, if for any PPT adv A there is a negl. function $\varepsilon(n)$ such that

$$Pr\left(A(1^n) \text{ wins the MAC experiment}\right) \leq \varepsilon(n).$$

Note: This definition doesn't preclude "replay attacks", where the adversary replaces a later message by an earlier message already sent by Alice (for which they know the tag). But replay attacks are not problematic if we add unique identifiers to messages (counter, timestamps ...).

How to Construct MACs

Spoiler: Use a PRF! PRFs are basically MACs, in fact. :)

Here's the MAC: Let F be a PRF

- $\text{Gen}(1^n)$ samples $k \leftarrow \{0,1\}^n$
- $\text{MAC}(k, m) = F_k(m)$
- $\text{Ver}(k, m, t)$ = canonical verification
  i.e., output 1 if and only if $F_k(m) = t$

Thm: The scheme above is a secure MAC.

Proof. Correctness is clear, so we focus on security.

At a high level, we follow the same two steps as in the proof of CPA-security of the PRF-based encryption scheme.

1) Show that the claim is true when F is replaced by a random function
2) Argue that the adversary won't notice if we replace the random function by F.

1) Consider the idealized $\widetilde{MAC}$ which is exactly like MAC but uses a random function instead of F.
Fix an arbitrary adversary A.
We claim that $\Pr(A(1^n) \text{ wins the } \widetilde{MAC} \text{ experiment}) = 2^{-n}$.

To see this, note that for $m' \notin Q$ the output $\tilde{t} = \widetilde{MAC}(k, m')$ is independent of the adversary's view
↳ is uniformly random over $\{0,1\}^n$.

Suppose that A outputs (m', t'). Then, since $\tilde{t}$ is independent of (m', t'), we have

$$\Pr(A \text{ wins } \widetilde{MAC} \text{ experiment}) = \Pr(\tilde{t} = t') = 2^{-n}$$

2) Suppose that there is a PPT adv A and a polynomial $p(n)$ s.t

$$\left| \Pr(A(1^n) \text{ wins the MAC exp}) - \Pr(A(1^n) \text{ wins the } \widetilde{MAC} \text{ exp}) \right| \geq \frac{1}{p(n)}$$

for infinitely many n's.

We use this A to break the PRF F. Consider the PPT adv $D^{\mathcal{O}}(1^n)$ for the PRF security game as follows:

↳ either $\mathcal{O}_{real}$ or $\mathcal{O}_{ideal}$

→ D runs $A(1^n)$

→ If A asks a query m to the MAC oracle, D queries $\mathcal{O}$ on m to get z, then gives z to A. Let Q be the list of queries.

→ If A outputs (m', t'), then D queries $\mathcal{O}$ on m' and gets back z'. D outputs 1 iff $t' = z'$.

If $\mathcal{O} = \mathcal{O}_{real,k}$ then D plays the MAC experiment with A, and so

$$\Pr\left(D^{\mathcal{O}_{real,k}}(1^n) = 1\right) = \Pr(A \text{ wins the MAC exp.}).$$

If $\mathcal{O} = \mathcal{O}_{ideal}$, then D plays the $\widetilde{MAC}$ experiment with A, and so

$$\Pr\left(D^{\mathcal{O}_{ideal}}(1^n) = 1\right) = \Pr(A \text{ wins the } \widetilde{MAC} \text{ exp.})$$

$$\Rightarrow \left|\Pr_k\left(D^{\mathcal{O}_{real,k}}(1^n) = 1\right) - \Pr\left(D^{\mathcal{O}_{ideal}}(1^n) = 1\right)\right| \geq \frac{1}{p(n)},$$

a contradiction.

Therefore, there is a negl. function $\varepsilon(n)$ s.t.

$$\left|\Pr(A(1^n) \text{ wins the MAC exp}) - \underbrace{\Pr(A(1^n) \text{ wins the } \widetilde{MAC} \text{ exp})}_{2^{-n}}\right| \leq \varepsilon(n)$$

$$\Rightarrow \Pr(A(1^n) \text{ wins the MAC exp.}) \leq \underbrace{\varepsilon(n) + 2^{-n}}_{\text{negligible}}$$

$\blacksquare$