Lecture 26: Collision-Resistant Hash-Function Family

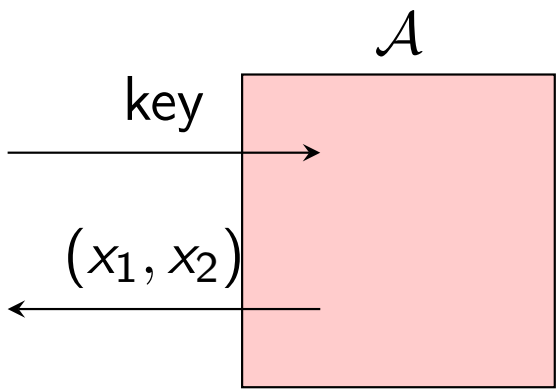
Recall

- One-time MAC: We can construct from 2-wise independent hash functions. This construction is secure even against adversaries with unbounded computational power.
- General MAC. If one-way functions (OWF) exist, then we can construct pseudorandom generators (PRG) and, then, pseudorandom functions (PRF). Once, we have constructed pseudorandom functions, we can construct MACs for fixed-length messages and, then, MAC for arbitrary-length messages.
- Today, for efficiency purposes, we shall introduce the concept of collision-resistant hash-function family (CRHF) to construct MACs for arbitrary-length messages. We emphasize that the assumption that "OWF exist" is strictly weaker than "CRHF exists." So, theoretically, we prefer to construct MAC using OWF, but for practice we construct OWF using CRHF

Definition: Collision-Resistant Hash-Function Family I

- Suppose we have a hash-function family $\mathcal{H} = \{h_1, h_2, \ldots, h_\alpha\}$
- This hash-function family $\mathcal{H}$ is a collision-resistant hash-function family (CRHF) if the success probability (i.e., $\mathbb{P}[z = 1]$) in the following experiment, for any computationally efficient adversary $\mathcal{A}$, is small

$$\text{key} \xleftarrow{\$} \{1, 2, \ldots, \alpha\}$$



$$z = 1 \text{ iff}$$
$$x_1 \neq x_2 \text{ and } h_{\text{key}}(x_1) = h_{\text{key}}(x_2)$$

Definition: Collision-Resistant Hash-Function Family II

- Intuitively, the experiment captures the following property.
  - We are picking a random hash function h_{key} from the hash function family $\mathcal{H}$.
  - We present the hash function to the adversary $\mathcal{A}$ (by sending key to it).
  - The adversary $\mathcal{A}$ replies with two inputs x_1, x_2.
  - The adversary succeeds in this experiment if it has produced a collision of h_{key}, i.e., $x_1 \neq x_2$ and $h_{\text{key}}(x_1) = h_{\text{key}}(x_2)$.
- Tersely, we insist that the following probability is small for every computationally efficient adversary $\mathcal{A}$

$$\mathbb{P}\left[z = 1 : \begin{array}{c} \text{key} \xleftarrow{\$} \{1, 2, \ldots, \alpha\} \\ (x_1, x_2) \xleftarrow{\$} \mathcal{A}(\text{key}) \\ z = 1 \text{ iff } x_1 \neq x_2 \text{ and } h_{\text{key}}(x_1) = h_{\text{key}}(x_2) \end{array}\right]$$

Quick Note

- Note that if P = NP then we can find the solution to the following NP statement

$$\exists(x_1, x_2) \text{ s.t. } x_1 \neq x_2 \text{ and } h_{\text{key}}(x_1) = h_{\text{key}}(x_2)$$

MAC for Fixed-length Messages I

- Suppose we are given a CRHF family

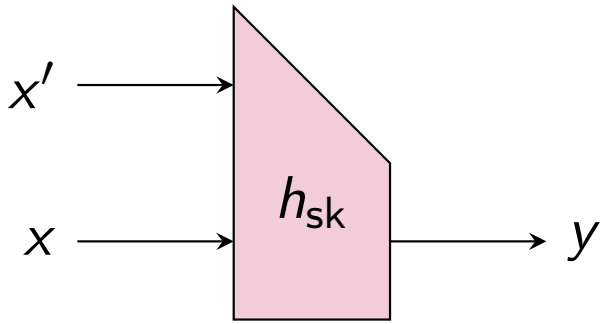
$$\mathcal{H} = \{h_1, h_2, \ldots, h_\alpha\},$$

where $h_i : \{0,1\}^{2B} \rightarrow \{0,1\}^B$, for every $i \in \{1, 2, \ldots, \alpha\}$

- We want to construct MAC for $m \in \{0,1\}^n$

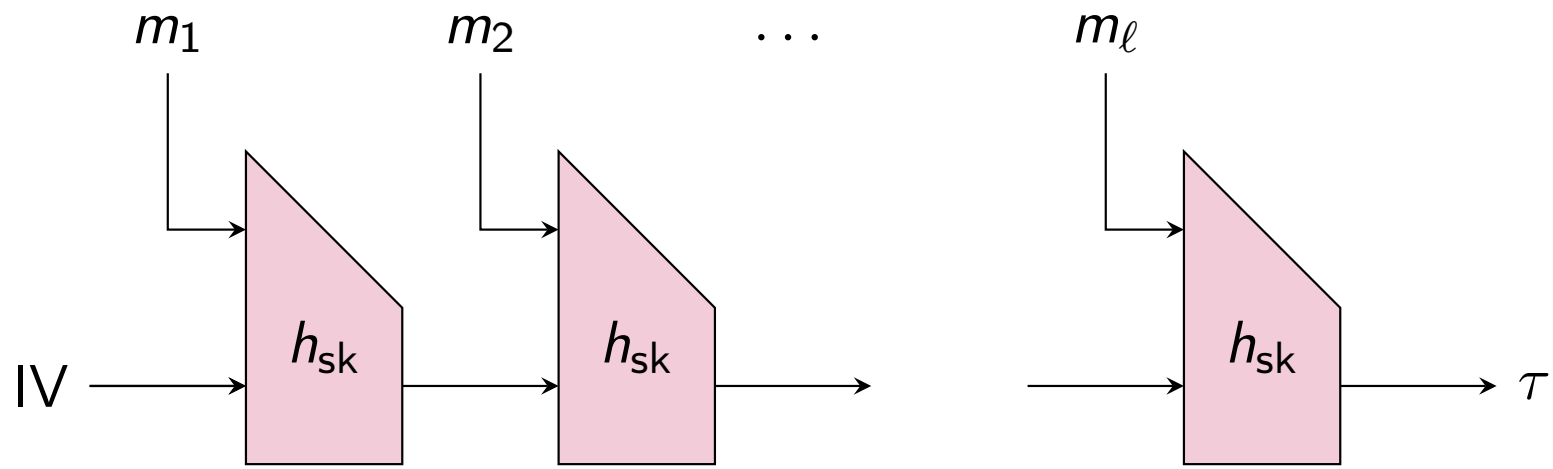
MAC for Fixed-length Messages II

- The secret-key is $\mathsf{sk} \xleftarrow{\$} \{1, 2, \ldots, \alpha\}$
- We will pictorially represent $h_{\mathsf{sk}}(x, x') = y$ as follows



MAC for Fixed-length Messages III

- Suppose the length of m is n and $\ell = \lceil n/B \rceil$
- We interpret $m = (m_1, m_2, \ldots, m_\ell)$, where each m_i is a block of B-bits
- The tag τ of the message m is generated as follows



- Here IV stands for "Initialization Vector" (any string in $\{0,1\}^B$). For example, you can choose $IV = 0^B$

Remarks

- This is known as the "Merkle-Damgård Construction"
- Think: How to extend this MAC algorithm to arbitrary-length messages