Lecture 25: CBC-MAC

Recall: Summary of MAC Schemes so far

- One-time MAC: We can construct from 2-wise independent hash function families. These exist even against adversaries with unbounded computational power
- General MAC: We can construct if One-way Functions Exist. For example, we use pseudorandom functions (using the GGM construction) for these constructions. The GGM construction uses length-doubling pseudorandom generators, and pseudorandom generators can be constructed from one-way functions

Today's Summary

Today we shall construct MACs using pseudorandom function (PRF) family and the Cipher Block Chaining (CBC) technique

MAC for Fixed-length Messages I

What we shall use

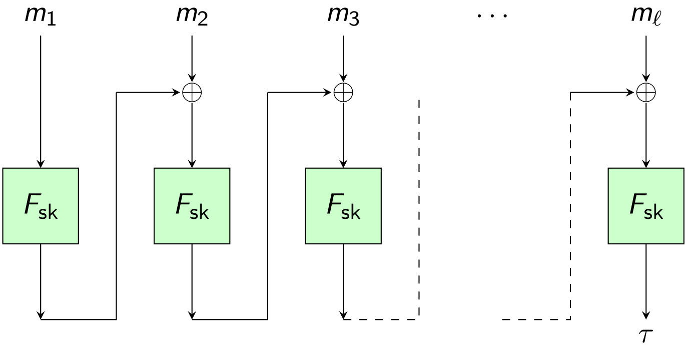
- Pseudorandom Function Family $\mathcal{F} = \{F_1, F_2, \ldots, F_\alpha\}$, where each function $F_i \colon \{0,1\}^B \to \{0,1\}^B$

What we shall construct

- Construct a MAC scheme for n-bit messages

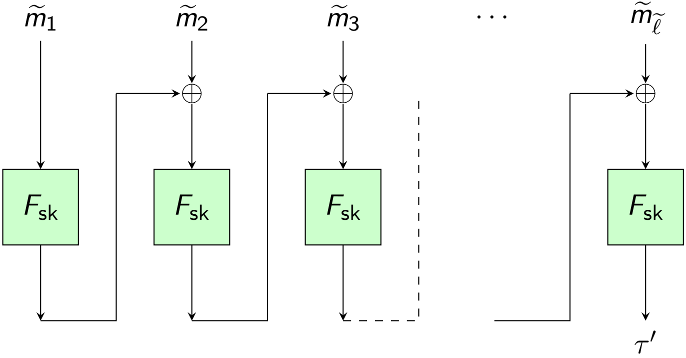
MAC for Fixed-length Messages II

- Gen(). Create a key for the pseudorandom function family. Return $\mathsf{sk} \xleftarrow{\$} \{1, 2, \dots, \alpha\}$
- $\mathsf{Mac}_{\mathsf{sk}}(m)$. Interpret $m = (m_1, m_2, \dots, m_\ell)$, where each m_i is B-bits long and $\ell = \lceil n/B \rceil$



MAC for Fixed-length Messages III

- $\mathrm{Ver}_{\mathsf{sk}}(\widetilde{m}, \widetilde{\tau})$. Let $\widetilde{m} = (\widetilde{m}_1, \widetilde{m}_2, \dots, \widetilde{m}_{\widetilde{\ell}})$, where each $\widetilde{m}_i$ is B-bit long. Return whether $\widetilde{\tau} == \tau'$ or not, where τ' is calculated as below.



MAC for Fixed-length Messages IV

Attack on this Scheme using Arbitrary-length Messages.

- The adversary sees the message-tag pair (m, τ), where $m = (m_1, m_2, \dots, m_\ell)$
- The adversary sees the message-tag pair (m', τ'), where $m' = (m'_1, m'_2, \dots, m'_{\ell'})$
- The adversary computes

$$\widetilde{m} = (m_1, \dots, m_\ell, m'_1 \oplus \tau, m'_2, \dots, m'_{\ell'})$$

- The message-tag pair $(\widetilde{m}, \tau')$ is a forgery (Check that this passes verification)

MAC-ing Arbitrary-length Messages, First Construction I

What we shall use

- Pseudorandom Function Family $\mathcal{F} = \{F_1, F_2, \ldots, F_\alpha\}$, where each function $F_i \colon \{0,1\}^B \to \{0,1\}^B$

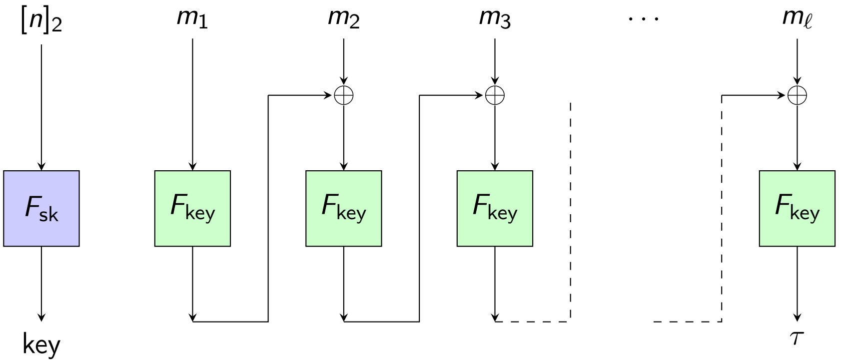
What we shall construct

- Construct a MAC scheme for $\{0,1\}^*$

Intuition for the construction.

- We shall use separate sk for each message length to "chain"
- The Gen() returns a random sk $\stackrel{\$}{\leftarrow} \{1, 2, \ldots, \alpha\}$.
- The pictorial summary of $\mathsf{Mac}_{\mathsf{sk}}(m)$ is provided in the next slide

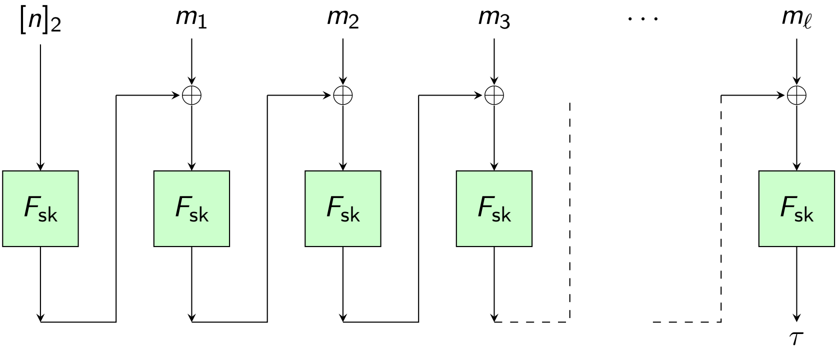
Suppose the message is $m \in \{0,1\}^n$. It is interpreted as $(m_1, m_2, \ldots, m_\ell)$, where each m_i is a B-bit string and $\ell = \lceil n/B \rceil$. Let $[n]_2$ represent the B-bit binary string that represents the length of m in binary.



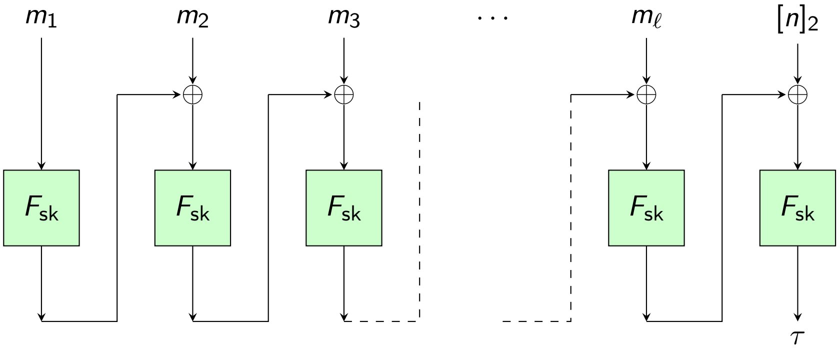
MAC-ing Arbitrary-length Messages, First Construction IV

Note. You can use the same sk to sign messages of different length using the algorithm presented above!

We append the binary representation of the length of m at the beginning and CBC-MAC the new message. See the construction below.



Adding the length at the end is INSECURE! The following scheme is insecure.



Students are strongly recommended to construct the attack on their own

- Suppose the adversary the message-tag pairs on two different n-bit messages p and q. Let the message tag pairs be

$$\begin{gathered}\left(\ p = (p_1, p_2, \dots, p_\ell)\ ,\ \tau_p\ \right)\\ \left(\ q = (q_1, q_2, \dots, q_\ell)\ ,\ \tau_q\ \right)\end{gathered}$$

- The adversary requests to see the tag τ_m for the message m as defined below

$$m = \left(\ p_1, p_2, \dots, p_\ell, [n]_2, r_1, r_2, \dots, r_t\ \right)$$

We emphasize that here the adversary requests to see the signature on a particular message. All previous attacks had the adversary obtain message-tag pairs for arbitrary messages.

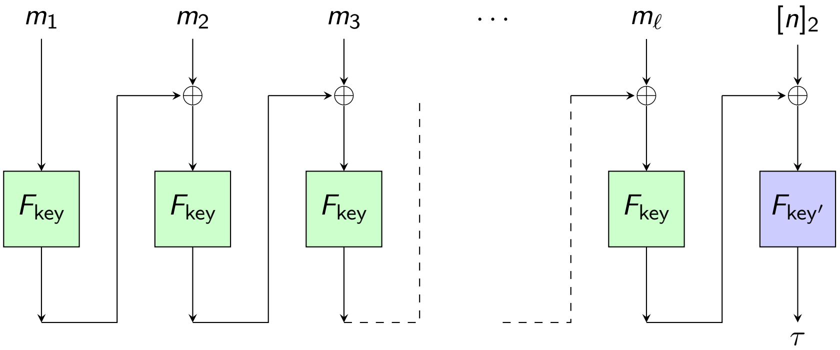
- Now, the adversary can splice out $(p_1, \dots, p_\ell)$ to replace $(q_1, \dots, q_\ell)$ in the message m as follows

$$m' = \left(\ q_1, q_2, \dots, q_\ell, [n]_2, r_1 \oplus \tau_p \oplus \tau_q, r_2, \dots, r_t\ \right)$$

- Note that the tag of the message m' is identical to the tag τ_m

But a small change to the above-mentioned insecure construction is secure.
All we need to ensure is that the key for the pseudorandom function used to chain the message-blocks is <ins>different</ins> from the key for the pseudorandom function used on $[n]_2$. Let $\mathsf{key} = F_{\mathsf{sk}}(0)$ and $\mathsf{key}' = F_{\mathsf{sk}}(1)$. Now, consider the following construction.

MAC-ing Arbitrary-length Messages, Third Construction II



Check how this new construction prevents the adversarial attack where the message length was at the end. This is crucial to ensure that you have a good understanding of this new MAC scheme.

Benefit of having the message-length at the end. We do not need the length of the message ahead of time. We can even MAC messages that are coming as a stream!