EE 720: An Introduction to Number Theory and Cryptography (Spring 2020)

Lecture 11 — February 20, 2020

Instructor: Saravanan Vijayakumaran *Scribe: Saravanan Vijayakumaran*

1 Lecture Plan

- Construction and security proof of a fixed-length MAC
- Challenges in domain extension for MACs
- CBC-MAC

2 Message Authentication Codes

- Message authentication codes prevent *undetected tampering* of messages sent over an open communication channel.
- A MAC consists of three PPT algorithms $(\texttt{Gen}, \texttt{Mac}, \texttt{Vrfy})$.
- Consider the following **message authentication experiment** $\texttt{Mac-forge}_{\mathcal{A},\Pi}(n)$:
  1. A key k is generated by running $\texttt{Gen}(1^n)$.
  2. The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\texttt{Mac}_k(\cdot)$. The adversary eventually outputs (m, t). Let $\mathcal{Q}$ denote the set of all queries that $\mathcal{A}$ asked its oracle.
  3. $\mathcal{A}$ succeeds if and only if (1) $\texttt{Vrfy}_k(m, t) = 1$ and (2) $m \notin \mathcal{Q}$. If $\mathcal{A}$ succeeds, the output of the experiment is 1. Otherwise, the output is 0.
- A MAC is secure if no efficient adversary can succeed in the above experiment with non-negligible probability.

Definition. *A message authentication code* $\Pi = (\texttt{Gen}, \texttt{Mac}, \texttt{Vrfy})$ *is* ***existentially unforgeable under an adaptive chosen-message attack****, or just* ***secure****, if for all PPT adversaries* $\mathcal{A}$*, there is a negligible function* $\texttt{negl}$ *such that:*

$$\Pr\left[\texttt{Mac-forge}_{\mathcal{A},\Pi}(n) = 1\right] \leq \texttt{negl}(n).$$

- The above definition of MAC security offers no protection against *replay attacks*. These can be prevented using sequence numbers or timestamps.

2.1 Fixed-Length MAC Construction

- Let F be a pseudorandom function. Define a fixed-length MAC for messages of length n as follows:
  - $\texttt{Mac}$: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, output the tag $t \coloneqq F_k(m)$.
  - $\texttt{Vrfy}$: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, and a tag $t \in \{0,1\}^n$, output a 1 if and only if $t = F_k(m)$. If $t \neq F_k(m)$, output 0.

Theorem 1. *If F is a pseudorandom function, then the above construction is a secure fixed-length MAC for messages of length n.*

Proof. See pages 117–118 in Katz/Lindell. □

3 Domain Extension for MACs

- The above secure MAC construction works only for fixed-length messages. What about arbitrary-length messages?
- Suppose the message m can be broken up into a sequence of d blocks $m_1, m_2, \ldots, m_d$ each of which is an element of $\{0,1\}^n$.
- Let us ignore efficiency of the scheme in terms of the tag length. Suppose we are only interested in authenticating arbitrary-length messages. The discussion will help illustrate some canonical attacks.
- Let $\Pi' = (\texttt{Mac}', \texttt{Vrfy}')$ be a secure fixed-length MAC for messages of length n. We want to construct a secure MAC $\Pi = (\texttt{Mac}, \texttt{Vrfy})$ for messages of length dn.
- If we simply compute a per-block tag $t_i = \texttt{Mac}'_k(m_i)$ and output $\langle t_1, \ldots, t_d \rangle$ as the tag for m, then an adversary can perform a *block reordering attack*.
- We can prevent block reordering attacks by authenticating the block index along with the message. After reducing the size of the blocks, we can compute $t_i = \texttt{Mac}'_k(i \| m_i)$. But this does not prevent a *truncation attack* where an attacker simply drops blocks from the end of the message.
- To prevent truncation attacks, the message length could be authenticated. After further reducing the size of the blocks, we compute $t_i = \texttt{Mac}'_k(l \| i \| m_i)$ and output $\langle t_1, \ldots, t_d \rangle$ as the tag for m. Here l is the length of the message in bits. This is still vulnerable to a *mix-and-match attack*.
- To prevent mix-and-match attacks, we include a random *message identifier* in the authentication of each block. The following is a construction of a secure MAC if Π' is a secure MAC.
  - Let $m \in \{0,1\}^*$ be a message of length $l < 2^{n/4}$. Parse m into d blocks $m_1, m_2, \ldots, m_d$ of length $n/4$ bits each.
  - Choose r uniformly from $\{0,1\}^{n/4}$.

– For $i = 1, 2, \ldots, d$, compute $t_i \leftarrow \texttt{Mac}'_k(r \| l \| i \| m_i)$ where i and l are encoded as $n/4$-bit strings.
– Output the tag $t \coloneqq \langle r, t_1, t_2, \ldots, t_d \rangle$.

4 CBC-MAC

- If the tag length of $\texttt{Mac}'$ is n bits long, the above construction is inefficient as it generates a tag which is more than 4 times longer than the message length.
- CBC-MACs are widely used in practice.
- We first present a basic construction of a CBC-MAC which is secure only when authenticating messages of fixed length. We then extend it to a more general construction which is secure for authenticating arbitrary-length messages.

4.1 Basic Construction

- Let F be a length-preserving pseudorandom function with key/input/output length equal to n bits. Let $m \in \{0,1\}^{dn}$ be a message for a fixed $d > 0$.
  – `Mac`: Parse the message m in to d blocks $m_1, \ldots, m_d$ of length n bits each.
  Set $t_0 = 0^n$. For $i = 1, \ldots, d$, set $t_i = F_k(t_{i-1} \oplus m_i)$.
  Output t_d as the tag.
  – `Vrfy`: For a message-tag pair (m, t) output 0, if the message is not of length dn.
  Otherwise, output 1 if and only if $t = \texttt{Mac}_k(m)$.

Theorem. *If $d = l(n)$ for some polynomial l and F is a pseudorandom function, then the above construction is secure for messages of length dn.*

5 References and Additional Reading

- Sections 4.3, 4.4 from Katz/Lindell