EE 720: An Introduction to Number Theory and Cryptography (Spring 2018)

Lecture 13 — February 23, 2018

Instructor: Saravanan Vijayakumaran *Scribe: Saravanan Vijayakumaran*

1 Lecture Plan

- Challenges in domain extension for MACs
- Authenticated Encryption

2 Recap

- Message authentication codes prevent *undetected tampering* of messages sent over an open communication channel.
- A MAC consists of three PPT algorithms $(\texttt{Gen}, \texttt{Mac}, \texttt{Vrfy})$.
- We defined a message authentication experiment $\texttt{Mac-forge}_{\mathcal{A},\Pi}(n)$.
- A MAC is secure if for all PPT adversaries $\mathcal{A}$ we have

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

- We proved the security of a fixed-length MAC construction.

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 $m_1, m_2, \dots, m_d$ each of which is an element of $\{0,1\}^n$.
- If we simply compute a per-block tag $t_i = \texttt{Mac}_k(m_i)$ and output $\langle t_1, \dots, t_d \rangle$ 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 $\langle t_1, \dots, t_d \rangle$ 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.*

- **We will skip the construction of MACs for arbitrary-length messages due to lack of time.** We will see one construction later in our discussion of hash functions.

4 Authenticated Encryption

- In many applications where secrecy is needed it turns out that integrity is also essential. It is a best practice to ensure *both secrecy and integrity* of the messages.
- Let $\Pi = (\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ be private-key encryption scheme.
  - We require Π to be CCA-secure.
  - We want the scheme to satisfy existential unforgeability under an adaptive chosen-message attack
- But Π does not have the syntax of a message authentication code. So we introduce a definition of integrity for this case.
- **The unforgeable encryption experiment** $\texttt{Enc-Forge}_{\mathcal{A},\Pi}(n)$:
  1. Run $\texttt{Gen}(1^n)$ to obtain a key k.
  2. The adversary $\mathcal{A}$ is given input 1^n and access to an encryption oracle $\texttt{Enc}_k(\cdot)$. The adversary outputs a ciphertext c.
  3. Let $m := Dec_k(c)$, and let $\mathcal{Q}$ denote the set of all queries that $\mathcal{A}$ asked its encryption oracle. The output of the experiment is 1 if and only if (1) $m \neq \perp$ and (2) $m \neq \mathcal{Q}$.

Definition. *A private-key encryption scheme Π is **unforgeable** if for all PPT adversaries $\mathcal{A}$, there is a negligible function $\texttt{negl}$ such that:*

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

Definition. *A private-key encryption scheme Π is an **authenticated encryption scheme** if it is CCA-secure and unforgeable.*

4.1 How to Construct AE Schemes?

- One should be careful while combining a secure encryption scheme and secure MAC to construct a secure authenticated encryption scheme.
- Let $\Pi_E = (\texttt{Enc}, \texttt{Dec})$ be a CPA-secure encryption scheme and let $\Pi_M = (\texttt{Mac}, \texttt{Vrfy})$ denote a MAC, where key generation in both schemes involves choosing a uniform n-bit key.[1]
- Assume independent keys k_E and k_M for Π_E and Π_M, respectively. There are three natural approaches to combining encryption and authentication.

[1]Note that we do not assume Π_E is CCA-secure. We have not seen a construction of such a scheme. The point of introducing MACs was to construct CCA-secure schemes.

1. *Encrypt-and-authenticate:* Given a plaintext message m, the sender transmits ciphertext $\langle c, t\rangle$ where:
$$c \leftarrow \mathtt{Enc}_{k_E}(m) \text{ and } t \leftarrow \mathtt{Mac}_{k_M}(m).$$
The receiver decrypts c to recover m; assuming no error occurred, it then verifies the tag t.
2. *Authenticate-then-encrypt:* A MAC tag t is first computed, and then the message and tag are encrypted together. Given a message m, the sender transmits the ciphertext c computed as:
$$t \leftarrow \mathtt{Mac}_{k_M}(m) \text{ and } c \leftarrow \mathtt{Enc}_{k_E}(m\|t).$$
The receiver decrypts c to recover $m\|t$; assuming no error occurred, it then verifies the tag t.
3. *Encrypt-then-authenticate:* Given a plaintext message m, the message is first encrypted and then a MAC tag is computed over the result. The ciphertext is the pair $\langle c, t\rangle$ where:
$$c \leftarrow \mathtt{Enc}_{k_E}(m) \text{ and } t \leftarrow \mathtt{Mac}_{k_M}(c).$$
The receiver first verifies the tag t; assuming no error occurred it decrypts c to recover m.

- The first two approaches are not secure.
  - In encrypt-and-authenticate, if a *deterministic* MAC is used then the resulting scheme is not CPA-secure.
  - In authenticate-then-encrypt, the ciphertext c is not authenticated and is vulnerable to padding oracle attacks. There are now two sources of decryption failure: the padding may be incorrect or the MAC tag may not verify. Even if one error is returned for both failures, timing attacks can be used to figure out which failure occurred.
- The encrypt-then-authenticate scheme is an authenticated encryption scheme if the MAC is *strongly secure.*
- A secure MAC guarantees that any PPT adversary will not be able to forge a valid tag t for a message $m \notin \mathcal{Q}$ (the set of messages queried). The definition does not say anything about the situation when the adversary can generate a different valid tag $t' \neq t$ for some message in $\mathcal{Q}$. A secure MAC allows such an adversary to exist. But a strongly secure MAC guarantees excludes such adversaries.
- Consider the experiment $\mathtt{Mac\text{-}sforge}_{\mathcal{A},\Pi}(n)$:
  1. A key k is generated by running $\mathtt{Gen}(1^n)$.
  2. The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\mathtt{Mac}_k(\cdot)$. The adversary eventually outputs (m, t). Let $\mathcal{Q}$ denote the set of all pairs (m', t') of oracle queries and their associated responses.
  3. $\mathcal{A}$ succeeds if and only if (1) $\mathtt{Vrfy}_k(m, t) = 1$ and (2) $(m, t) \notin \mathcal{Q}$. If $\mathcal{A}$ succeeds, the output of the experiment is 1. Otherwise, the output is 0.

Definition. *A message authentication code $\Pi = (\mathtt{Gen}, \mathtt{Mac}, \mathtt{Vrfy})$ is* ***strongly secure****, or* ***a strong MAC****, if for all PPT adversaries $\mathcal{A}$, there is a negligible function $\mathtt{negl}$ such that:*
$$\Pr\left[\mathtt{Mac\text{-}sforge}_{\mathcal{A},\Pi}(n) = 1\right] \leq \mathtt{negl}(n).$$

- **Encrypt-then-authenticate construction:** Let $\Pi_E = (\texttt{Enc}, \texttt{Dec})$ be a CPA-secure encryption scheme and let $\Pi_M = (\texttt{Mac}, \texttt{Vrfy})$ denote a MAC, where key generation in both schemes involves choosing a uniform n-bit key. Define a private-key encryption schem $(\texttt{Gen}', \texttt{Enc}', \texttt{Dec}')$ as follows:
  - $\texttt{Gen}'$: on input 1^n, choose independent, uniform $k_E, k_M \in \{0,1\}^n$ and output the key (k_E, k_M).
  - $\texttt{Enc}'$: on input a key (k_E, k_M) and a plaintext message m, compute $c \leftarrow \texttt{Enc}_{k_E}(m)$ and $t \leftarrow \texttt{Mac}_{k_M}(c)$. Output the ciphertext $\langle c, t\rangle$.
  - $\texttt{Enc}'$: on input a key (k_E, k_M) and a ciphertext $\langle c, t\rangle$, first check $\texttt{Vrfy}_{k_M}(c, t) = 1$. If yes, then output $\texttt{Dec}_{k_E}(c)$; if no, then output $\perp$.

Theorem. *Let Π_E be a CPA-secure private-key encryption scheme, and let Π_M be a strongly secure MAC. Then the above construction is an authenticated encryption scheme.*

5 References and Additional Reading

- Sections 4.4, 4.5 from Katz/Lindell