EE 720: An Introduction to Number Theory and Cryptography (Spring 2019)

Lecture 12 — February 11, 2019

Instructor: Saravanan Vijayakumaran *Scribe: Saravanan Vijayakumaran*

1 Lecture Plan

- Challenges in domain extension for MACs
- CBC-MAC

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 $\{0,1\}^n$.
- 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 $\langle t_1, \dots, t_d \rangle$ 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 $\langle t_1, \ldots, t_d \rangle$ 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.*

- For a message length dn, only n bits of tag is required.
- This construction is secure only if the sender and the receiver agree upon the length of the messages being authenticated in advance.
  - Suppose we have a sender and receiver who do not fix the length of the messages being authenticated. Additionally, assume that the sender only authenticates messages of length $2n$ but the receiver performs verification for arbitrary-length messages. Show that an adversary can forge a tag on a message of length $4n$.
- CBC-MAC vs CBC-mode encryption
  - F needs to be pseudorandom permutation of the CBC-mode encryption. For CBC-MAC, F is only required to be a pseudorandom function.
  - CBC-mode encryption uses a random IV which is crucial for security. CBC-MAC uses a fixed IV of 0^n which is also crucial for security. *A CBC-MAC with a random IV is not secure even for fixed-length message authentication. (Why?)*
  - In CBC-mode encryption, all the outputs of the F_k blocks are revealed. In CBC-MAC, only the final F_k block output is revealed. *A CBC-MAC with a fixed IV* 0^n *but with* F_k *outputs revealed is not secure even for fixed-length message authentication. (Why?)*

4.2 Secure CBC-MAC for Arbitrary-Length Messages

- Suppose m is a message of length dn. Parse m into $m_1, m_2, \ldots, m_d$ each of which is n bits long.
- Prepend the message length $|m|$ as an n-bit block resulting in message block sequence $|m|, m_1, \ldots, m_d$. Apply the CBC-MAC to this message and output the resulting tag t for m.
- Appending the message length $|m|$ and then computing the basic CBC-MAC is *not secure.*

5 References and Additional Reading

- Sections 4.3, 4.4 from Katz/Lindell