CSC 591 Cryptography 10/24/2018

Lecture 11: Message Authentication Code (MAC)

Lecturer: Alessandra Scafuro *Scribe: Mingkang Zhuang*

Security of Message Authentication Code

In previous lectures, we learned about security for encryption of messages, we focused on the *confidentiality* of encryption schema, and whether an adversary can *distinguish* the ciphertext. But it raise another problem about *integrity*, how do we *authenticate* message we received is sent from the sender, not replaced or modified by a third party.

In this lecture, we learn the concept and security of *Message Authentication Code (MAC)*.

Definition

We see that encryption does not solve problem of integrity, so additional mechanism is needed to let communicating parties to know whether or not a message was tampered with. MAC is a tool aims to prevent an adversary from modifying or replacing the message sent from one to another, without detecting the message is not from the original sender.

Definition A message Authentication Code is a tuple of algorithms Π = (Gen, Mac, Verify) where Gen(1^n) outputs a key k, Mac(m, k) outputs a tag t and Verify(k, m, t) outputs 1 if the tag is correct (and 0 otherwise).

Formal Representation

The message authentication experiment Mac-Forge($\mathcal{A}$, Π, n)

1. A key k is generated by running Gen(1^n).
2. The adversary $\mathcal{A}$ is given oracle access to the MAC function (Mac(k, m), and can query this oracle with input m_i and obtain outputs t_i. Let $\mathcal{Q}$ be the set of queries asked by $\mathcal{A}$ to the MAC oracle.
3. When the challenger is ready, the adversary outputs a tuple (m^*, t^*).
4. The adversary wins if Verify(m^*, t^*) = 1, and $m^* \notin \mathcal{Q}$.

Unforgeability: even after observing many pairs of (m_i, t_i), the adversary is not able to generate a valid new tag for a new message:

$$Pr[Mac - Forge_{\mathcal{A},\Pi}(n) = 1] \leq negl(n)$$

Scheme

MAC

$Gen(1^n) \rightarrow k$

$Mac(m, k) \rightarrow t$

$Verify(k, m, t)$
outputs 1 if the tag matches the massage
outputs 0 otherwise

Security Proof

We construct a MAC scheme Π using PRF F as follow:

- $Gen(1^n) \rightarrow k$
- Mac(k, m):
  1. $t = F_k(m)$
  2. Output m, t
- Verify(k, m, t):

  Output 1 if $t = F_k(m)$
  Output 0 if $t \neq F_k(m)$

Theorem: If $\mathcal{F}$ is a PRF, then Π is a secure MAC scheme.

Proof: Towards a contradiction, assume there exists an adversary $\mathcal{A}_{Forge}$ wins the forge game, $\mathcal{A}_{Forge}$ outputs a pair (m^*, t^*) such that $t^* = F_k(m^*)$, with probability $negl(n)$

we can construct an adversary $\mathcal{A}_F$ that distinguishes the output of an PRF from truly random function. $\mathcal{A}_F$ has oracle access $\mathcal{O}$ with $\mathcal{F}$ and Truly random function:

1. when $\mathcal{A}_{Forge}$ queries Game Π with input m_i, $\mathcal{A}_F$ sends m_i to oracle $\mathcal{O}$ and obtain y_i and sends to $\mathcal{A}_{Forge}$
2. when $\mathcal{A}_{Forge}$ outputs (m^*, t^*), $\mathcal{A}_F$ sends m^* to oracle and obtain z
3. if $z = t^*$ output 1

   else output 0

Analysis:

$\mathcal{A}_{Forge}$ simulates exactly as $\mathcal{A}_F$, which wins the game with probability of $negl(n)$. So Π is a secure MAC scheme.