CS6222 Cryptography

Topic: MAC and Cryptographic Hash Function — Date: Oct 31, 2024

Lecturer: Wei-Kai Lin (TA: Arup Sarker) — Scriber: Liran Li and Mikhail Kornilov

1 Message Authentication Code

Firstly we recap the definition of the security of MAC:

Definition 1. Security of MAC
For all NUPPT A, there exists a negligible $\epsilon(\cdot)$ such that for all $n \in \mathbb{N}$:

$$\Pr[k \leftarrow Gen(1^n), (m', \sigma') \leftarrow A^{Tag_k(\cdot)}(1^n) : A \text{ never queried } m' \text{ and } Ver_k(m', \sigma') = Acc] \le \epsilon(n)$$

Also, we have the security challenge game:

$$\begin{aligned} A(1^n) &\leftrightarrow Challenge(1^n) : k \leftarrow Gen(1^n) \\ x &\rightarrow \\ &\leftarrow Tag_k(x) \\ x' &\rightarrow \\ &\leftarrow Tag_k(x') \\ &\vdots \\ (m', \sigma') &\rightarrow \end{aligned}$$

A wins if and only if $Tag_k(m')$ is not queried and $Ver_k(m', \sigma') = Acc$.

Claim 2. ***MAC from PRF***
Suppose we have a family of PRF $F = \{f_k : \{0,1\}^n \rightarrow \{0,1\}^n, |k| = n, n \in \mathbb{N}\}$. We construct the followings:

1. *$Gen(1^n) : k \leftarrow \{0,1\}^n$.*
2. *$Tag_k(m) : m \in M = \{0,1\}^n$, $\sigma \leftarrow f_k(m)$.*
3. *$Ver_k(m, \sigma) : Acc$ if and only if $\sigma = f_k(m)$.*

We claim that (Gen, Tag, Ver) is a MAC.

Before proceeding to prove the construction in Claim 2, let's review the challenge game for the security of PRF:

$$\begin{aligned} & B \leftrightarrow PRF - Chal : b \leftarrow \{0,1\}, k \leftarrow \{0,1\}^n, R \leftarrow RF_n \\ & x \rightarrow \\ & \quad \leftarrow f(x) : f(\cdot) := \begin{cases} f_k(\cdot) & \text{if } b = 0 \\ R(\cdot) & \text{if } b = 1 \end{cases} \\ & b' \rightarrow \end{aligned}$$

and B wins if and only if $b' = b$.

Proof. The correctness is straightforward. Because $Ver_k(m, \sigma = Tag_k(m)) = Acc$ and $f_k(m) = \sigma = f_k(m)$, which is trivial. Then we want to show the security. Assume for contradiction, the MAC we constructed is not secure, that means there exists an NUPPT A and a polynomial $p(n)$ such that A wins the security challenge game with probability $\frac{1}{p(n)}$ for infinitely many $n \in \mathbb{N}$. We want to show that there exists another adversary B such that B can break the PRF scheme F. For the description of algorithm B, we have it runs A directly. At some point, A outputs x, B will forward this x to the challenger of PRF in the security game above, and we have the corresponding $f(x)$, which is essentially the tag σ using the PRF. Note that A can repeat this game multiple times, B can also query the information multiple times. By our assumption A will output a pair (m', σ'), B will forward m' and ask the oracle to output $f(m')$, if $\sigma' = f(m')$ and m' is not queried by A, it outputs "It is PRF", otherwise it outputs "It is random function". We see that if f is f_k, then B outputs "It is PRF" with probability at least $\frac{1}{p(n)}$. If f is R and m' is never queried, then

$$\Pr[\sigma' = R(m')] = \frac{1}{2^n}$$

and this concludes the proof. □

Note 3. *The hardness of predicting the output of PRF makes the prediction of the signature σ hard.*

Here comes one question, in the construction above, we need to know the message length in order to generate the key. In real world, this scenario is very limited. For very long message, say $|m| = 100n$, one potential solution will be firstly break $m = (m_1, ..., m_{100})$ and we get $\sigma_i \leftarrow Tag_k(m_i)$ for $i \in \{1, ..., 100\}$. However, this construction is wrong, the adversary can see all the σ_i's. The Adv can manipulate arbitrary pairs (m_i, σ_i). One potential way to fix this is:

$$\sigma_i \leftarrow Tag_k(m_i \| i \| 100)$$

and tell the receiver B the information: $\sigma_1, ..., \sigma_{100}, 100$. Again, this still cannot defend against the Adv, it can basically query twice to break.

2 Cryptographic Hash Function

Now we want to solve the problem for arbitrarily long message.

Claim 4. *Suppose we have a MAC scheme $MAC = (Gen, Tag, Ver)$, we have the message space $M = \{0,1\}^n$, $|k| = n$, we want the new MAC scheme $MAC' = (Gen', Tag', Ver')$ for $M = \{0,1\}^*$, which is an arbitrarily large message space. We claim MAC' should be constructed as:*

1. *For Gen', we have $k \leftarrow Gen(1^n)$.*
2. *For $Tag'_k(m)$, we have $s \leftarrow \{0,1\}^n$, $v \leftarrow h_s(m)$, $\sigma := (s, t = Tag_k(v))$. Where $h_s : \{0,1\}^* \to \{0,1\}^n$ is a hash function.*
3. *For $Ver'_k(m, \sigma)$, $\sigma = (s, t)$, $v := h_s(m)$, it outputs Acc if and only if $Ver_k(v, t) = Acc$.*

Definition 5. Cryptographic Hash Function
Now we introduce the *cryptographic hash function* $(GenH, H)$ describing how to compute $h_s(x)$:

1. (Compressing): $|h(x)| << |x|$. Which implies that the collision is possible: there exist $x \neq x'$ such that $h_s(x) = h(x')$.

2. For all NUPPT A, there exists a negligible function $\epsilon(\cdot)$ such that for all $n \in \mathbb{N}$,

$$\Pr[(x, st) \leftarrow A(1^*), s \leftarrow GenH(1^n), x' \leftarrow A(st, s) : x \neq x' \wedge h_s(x) = h_s(x')] \leq \epsilon(n)$$

3. $H(s, t)$ computes $h_s(x)$ in polynomial time.

Note that st stands for "state", indicating A is a stateful machine and $GenH$ means $s \leftarrow \{0, 1\}^n$.

Note 6. *In other words, the definition above means the function is compression (reducing length), easy to compute, and given any key, it is hard to find collision.*

What's coming next: we also have the object called Universal One-Way Hash Functions (UOWHF), we claim that if we have UOWHF and one standard MAC, we can get arbitrarily large MAC, we will discuss this next time.

Acknowledgement

References