2201-COL759
IIT Delhi Lecture 19 Venkata K

1 Plan for today's lecture

Last class, we saw two equivalent definitions for message authentication codes. Next, we saw a construction based on PRFs. Today, we will complete the proof of the construction, and discuss how (not) to construct MACs with unbounded message space.

2 MACs with bounded message space, from PRFs

Recall the following MAC construction using PRFs. Let $F : \mathcal{X} \times \mathcal{K} \to \mathcal{Y}$ be a secure PRF. We will construct a MAC scheme with message space $\mathcal{M} = \mathcal{X}$, key space $\mathcal{K}$, and signature space $\mathcal{Y}$.

MAC with bounded message space

- KeyGen : choose a random key $k \leftarrow \mathcal{K}$.
- $\mathsf{Sign}(m, k)$: Given message $m \in \mathcal{M} = \mathcal{X}$, output $y = F(m, k)$ as the signature.
- $\mathsf{Verify}(m, \sigma, k)$: Output 1 iff $F(m, k) = \sigma$.

Correctness follows immediately from the construction. Here, the signing algorithm is deterministic. Moreover, for any message m, there is a **unique** signature that verifies. Note that the signing algorithm being deterministic DOES NOT IMPLY every message has a unique signature that verifies (why?).

Before we get into the security proof, note the following two potential vulnerabilities:

- If F is a predictable function (that is, given many values $\{(x_i, F(x_i, k))\}_i$, it is easy to compute $F(x^*, k)$ for a new input x^*), then this signature scheme is not secure.
- If F is a secure PRF, but $|\mathcal{Y}|$ is polynomial in the security parameter, then also the signature scheme is broken. The adversary can simply pick some message m^*, and send a random element of $\mathcal{Y}$ as the signature. The guess will be correct with probability $1/|\mathcal{Y}|$, which is non-negligible if $|\mathcal{Y}|$ is polynomial in n.

The security proof below shows that these are the only possible vulnerabilities. If the function F is a secure PRF, and if $\mathcal{Y}$ is superpolynomial in the security parameter, then the resulting signature scheme is strongly unforgeable.

2.1 Proof of security

In this section, we will show that the above scheme satisfies strong unforgeability (as defined in the previous lecture).

Intuition: As a first step, can the adversary win the game without making any queries? In order to do so, the adversary must output $(m^*, F(m^*, k))$. But without making any queries, the adversary has no idea about k, and therefore, its best bet is to pick some m^*, pick a random $y \leftarrow \mathcal{Y}$, and output (m^*, y). The guess will be correct with probability $1/|\mathcal{Y}|$, and therefore, without any queries, the adversary can win with probability at most $1/|\mathcal{Y}|$.

Let us now consider general adversaries. The adversary is receiving some information about the key. But using PRF security, we can replace $F(\cdot, k)$ with a uniformly random function f. As a result, the adversary is playing a modified security game where it interacts with a random function, and must finally output m^* together with $f(m^*)$.

Formal proof: The formal proof will go through a sequence of games, where the original game is the strong-unforgeability game. We will gradually modify the games, such that the adversary's winning probability only degrades by a negligible additive factor. Finally, we will reach a game where the adversary has little/no chance of winning.

Game 0: This is the original security game.

- **Setup phase:** Challenger chooses a PRF key $k \leftarrow \mathcal{K}$.
- **Query phase:** Adversary can send polynomially many queries. For the i^{th} query m_i, the adversary receives $\sigma_i = F(m_i, k)$.
- **Forgery:** Finally, the adversary must output (m^*, σ^*) such that $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ and $\sigma^* = F(m^*, k)$.

First, note that for every message, there is exactly one signature that gets accepted (this follows from the construction). As a result, if the adversary sends (m^*, σ^*) that is not equal to any of the (m_i, σ_i) and it also verifies, then $m^* \neq m_i$ for all i.

Let p_0 denote the probability of $\mathcal{A}$ winning Game 0.

Game 1: In this game, the challenger picks a uniformly random function instead of a PRF key.

- **Setup phase:** Challenger chooses a uniformly random function $f \leftarrow \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
- **Query phase:** Adversary can send polynomially many queries. For the i^{th} query m_i, the adversary receives $\sigma_i = f(m_i)$.
- **Forgery:** Finally, the adversary must output (m^*, σ^*) such that $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ and $\sigma^* = f(m^*)$.

Let p_1 denote the probability of $\mathcal{A}$ winning Game 1.

2.1.1 Analysis

Claim 18.02. Suppose there exists a p.p.t. adversary $\mathcal{A}$ that makes q signature queries, and $p_0 - p_1 = \epsilon$. Then there exists a p.p.t. algorithm $\mathcal{B}$ that makes $q + 1$ queries to the PRF challenger, and wins the PRF security game with probability $1/2 + \epsilon/2$.

Proof. The reduction algorithm receives signing queries from $\mathcal{A}$, which it forwards to the PRF challenger. It then forwards the challenger's response to the adversary.

After polynomially many queries, the adversary sends (m^*, σ^*). The reduction algorithm makes a final query to the PRF challenger. It sends m^* to the PRF challenger, gets y^* in response. If $y^* = \sigma^*$, the reduction guesses '0', else it guesses '1'.

If the PRF challenger chose a pseudorandom function, then $\mathcal{B}$ outputs '0' with probability p_0. If the PRF challenger chose a random function, then $\mathcal{B}$ outputs '0' with probability p_1. Hence, it wins the bit-guessing game with probability $1/2 + (p_0 - p_1)/2$.

□

Claim 18.03. For any adversary $\mathcal{A}$, $p_1 = 1/|\mathcal{Y}|$.

Proof. In Game 1, the challenger chooses a random function f. This is equivalent to choosing the random function 'on-the-fly' as follows: for each new query m_i, the challenger picks a uniformly random string y_i and sends y_i to the adversary. It also maintains a database and adds (m_i, y_i) to the database.

In order to win, the adversary must output (m^*, σ^*) such that $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ for all i and $\sigma^* = f(m^*)$. Note that every message has a unique verifiable signature. Therefore, if $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ and σ^* is a valid signature for m^*, then $m^* \neq m_i$ for all i. However, since the random function f is chosen 'on-the-fly', $\sigma^* = f(m^*)$ with probability $1/|\mathcal{Y}|$.

□

Putting together the above claims, we get that if F is a secure PRF, and $1/|\mathcal{Y}|$ is negligible, then the signature scheme is strongly unforgeable.

3 MACs with long messages

The above construction gave a MAC scheme with bounded length messages. In real-world scenarios, we want the MAC to support unbounded length messages (suppose you are e-signing a digital document. You wouldn't want the document size to be restricted to 128 bytes).

Here are a few attempts discussed in class, and why they do not work. Let $(\mathsf{Sign}_{\text{bdd}}, \mathsf{Verify}_{\text{bdd}})$ be a MAC scheme that works for bounded message space. For all the attempts below, let $m = m_1 \mid\mid \ldots \mid\mid m_\ell$, where each m_i is n bits long. Also, for brevity, I am only describing the signing algorithm, verification can be defined appropriately.

3.1 Attempts discussed in class

1. Compute $z = \oplus_i \; m_i$, and output $\sigma = \mathsf{Sign}_{\text{bdd}}(z, k)$.

   This construction is not secure. If σ is a valid signature on $m = m_1 \mid\mid \ldots \mid\mid m_\ell$, then it is also a valid signature on $m' = m_\ell \mid\mid \ldots \mid\mid m_1$. Hence the adversary can query for a signature on m, and output (m', σ) as a forgery.

2. Output $\sigma = \sigma_1 \mid\mid \ldots \mid\mid \sigma_\ell$, where $\sigma_i = \mathsf{Sign}_{\text{bdd}}(m_i, k)$.

   This construction is also not secure. If σ is a signature on m, then $\sigma' = \sigma'_\ell \mid\mid \ldots \mid\mid \sigma'_1$ is a valid signature on $m' = m_\ell \mid\mid \ldots \mid\mid m_1$.

3. Output $\sigma = \sigma_1 \mid\mid \ldots \mid\mid \sigma_\ell$, where $\sigma_i = \mathsf{Sign}_{\text{bdd}}(i \mid\mid m_i, k)$.

   In this construction, we have appended a counter to each block before signing it. At first sight, this seems to take care of the issue in previous attempt. [1]

   This construction is also not secure, and can be broken using two signing queries. We can take $\ell = 2$ for this. First, it queries on $m_1 \mid\mid m_2$ and receives $\sigma_1 \mid\mid \sigma_2$. Next, it queries on $m'_1 \mid\mid m'_2$ and receives $\sigma'_1 \mid\mid \sigma'_2$. It then outputs $m^* = (m_1 \mid\mid m'_2)$ as the message and $\sigma^* = (\sigma_1 \mid\mid \sigma'_2)$ as the forgery.

   Check that m^* is not equal to either m or m', but the signature verifies.

   In Question 19.01, we present a modification of the above scheme which is provably secure.

4. Output $\sigma = \oplus_i \; \sigma_i$, where $\sigma_i = \mathsf{Sign}_{\text{bdd}}(i \mid\mid m_i, k)$. The adversary can query for signature on $m = m_1 \mid\mid m_2$, receives σ, next queries on $m' = m'_1 \mid\mid m'_2$, receives σ'. It then outputs message $m^* = m_1 \mid\mid m_2 \mid\mid m'_1 \mid\mid m'_2$ and signature $\sigma \oplus \sigma'$.

   Check that this signature verifies.

3.2 CBC-MAC: An approach for signing long messages

We have used cipher-block chaining for encrypting long messages. A similar approach can be tried for signing long messages. Let $F : \{0,1\}^n \times \mathcal{K} \to \{0,1\}^n$ be a secure PRF/PRP (one can use AES in practice), and suppose $m = m_1 \mid\mid \ldots \mid\mid m_\ell$ is the message to be signed (where each $m_i \in \{0,1\}^n$). The CBC-MAC procedure works as follows.

[1] During the lecture, my initial response was that this is a secure candidate. However, as was pointed out by one of the students, this construction can be broken using two signing queries. Hopefully this was a 'live demo' why it is important to write a formal proof, and not go by 'intuition'.

CBC-MAC: Basic Version

Let $t_0 = 0^n$. For each $i \geq 1$, compute $t_i = F(m_i \oplus t_{i-1}, k)$. Finally, output t_ℓ as the signature.

This construction is not secure. Given a signature σ on $m \in \{0,1\}^n$, consider the following forgery: set $m' = (m, m \oplus \sigma)$, and check that the signature on m' is also σ.

However, this is an interesting construction for the following reasons:

- If the message length is fixed (that is, we want a MAC for long but fixed-length messages), then this is a secure MAC. The attack above crucially used the fact that message space consists of messages of different lengths.
- The construction can be made provably secure by setting $t_0 = F(\ell, k)$. Here, ℓ is the number of blocks in the message, and it is expressed as an n bit string.
- The construction can be made secure by using another PRF key. That is, the signing key consists of two PRF keys (k_1, k_2). The signing algorithm is described below: Let $t_0 = 0^n$. For each $i \geq 1$, compute $t_i = F(m_i \oplus t_{i-1}, k_1)$. Finally, output $F(t_\ell, k_2)$ as the signature.

  First, note that the previous attack does not work. This is because the signature only contains $F(t_\ell, k_2)$, the signature does not contain t_ℓ. With this small modification, we get a **provably secure MAC that can handle long messages**. An additional advantage of this approach (as compared to the previous one) is that this does not require knowing the number of blocks before you start signing, and as a result, can be used in the *streaming setting*.

4 Lecture summary, plan for next lecture, additional resources

Summary: We completed the proof of MAC security. This construction required two things: PRF security, large co-domain. Next, we saw a few attempts towards building a secure MAC with unbounded message space. We finally saw the CBC-MAC and why it is not secure.

Next lecture: We will discuss how to 'fix' CBC-MAC to make it a secure MAC. We will end this topic with a summary of MACs. If time permits, we will discuss Question 19.01.

5 Questions

Question 19.01. Consider the following variant of Attempt 3. We are given a MAC scheme $(\mathsf{Sign}_{\text{bdd}}, \mathsf{Verify}_{\text{bdd}})$ which supports $4n$-bit messages. Suppose we wish to sign $m = m_1 \mid\mid \ldots \mid\mid m_\ell$ where each m_i is an n-bit string.
The signing algorithm chooses a uniformly random n-bit string r, computes $\sigma_i = \mathsf{Sign}_{\text{bdd}}(i \mid\mid m_i \mid\mid \ell \mid\mid r, k)$ for each i. Here i and ℓ are expressed as n-bit strings.
Finally, it outputs $\sigma = (r, \sigma_1 \mid\mid \ldots \mid\mid \sigma_\ell)$.

Prove that this is a secure MAC scheme, assuming $\mathsf{Sign}_{\text{bdd}}$ is a secure MAC for bounded-length messages.