Notes for Lecture 7

Scribed by Mark Landry, posted February 15, 2009

Summary

Today we start to talk about *message authentication codes* (MACs). The goal of a MAC is to guarantee to the recipient the integrity of a message and the identity of the sender. We provide a very strong definition of security (*existential unforgeability under adaptive chosen message attack*) and show how to achieve it using pseudorandom functions.

Our solution will be secure, but inefficient in terms of length of the required authentication information.

Next time we shall see a more space-efficient authentication scheme, and we shall prove that given a CPA-secure encryption scheme and a secure MAC, one can get a CCA-secure encryption scheme. (That is, an encryption scheme secure against an *adaptive chosen ciphertext and plaintext attack*.)

1 Message Authentication

The goal of message authentication is for two parties (say, Alice and Bob) who share a secret key to ensure the integrity and authenticity of the messages they exchange. When Alice wants to send a message to Bob, she also computes a *tag*, using the secret key, which she appends to the message. When Bob receives the message, he *verifies* the validity of the tag, again using the secret key.

The syntax of an authentication scheme is the following.

Definition 1 (Authentication Scheme) *An authentication scheme is a pair of algorithms $(Tag, Verify)$, where $Tag(\cdot,\cdot)$ takes in input a key $K \in \{0,1\}^k$ and a message M and outputs a tag T, and $Verify(\cdot,\cdot,\cdot)$ takes in input a key, a message, and a tag, and outputs a boolean answers. We require that for every key K, and very message M*

$$Verify(K, M, Tag(K, M)) = True$$

if $Tag(\cdot,\cdot)$ is deterministic, and we require

$$\mathbb{P}[Verify(K, M, Tag(K, M)) = True] = 1$$

if $Tag(\cdot, \cdot)$ *is randomized.*

In defining security, we want to ensure that an adversary who does not know the private key is unable to produce a valid tag. Usually, an adversary may attempt to forge a tag for a message after having seen other tagged messages, so our definition of security must ensure that seeing tagged messages does not help in producing a forgery. We provide a very strong definition of security by making sure that the adversary is able to tag *no* new messages, even after having seen tags of any other messages of *her choice*.

Definition 2 (Existential unforgeability under chosen message attack) *We say that an authentication scheme* $(Tag, Verify)$ *is* (t, ϵ)*-secure if for every algorithm* A *of complexity at most* t

$$\mathbb{P}_K[A^{Tag(K,\cdot)} = (M, T) : (\mathrm{M}, \mathrm{T}) \text{ is a forge}] \leq \epsilon$$

where a pair (M, T) *is a "forge" if* $Verify(K, M, T) = True$ *and* M *is none of the messages that* A *queried to the tag oracle.*

This definition rules out any possible attack by an active adversary except a *replay* attack, in which the adversary stores a tagged message it sees on the channel, and later sends a copy of it. We still are guaranteed that any message we see was sent at some time by the right party. To protect against *replay* attacks, we could include a timestamp with the message, and reject messages that are too old. We'll assume that *replay* attacks are handled at a higher level and will not worry about them.

2 Construction for Short Messages

Suppose $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^m$ is a pseudorandom function. A simple scheme is to use the pseudorandom function as a tag:

- $Tag(K, M) := F_K(M)$
- $Verify(K, M, T) := True$ if $T = F_K(M)$, $False$ otherwise

This construction works only for short messages (of the same length as the input of the pseudorandom function), but is secure.

Theorem 3 *If F is a (t, ϵ)-secure pseudorandom function, then the above construction is a $(t - O(m), \epsilon + 2^{-m})$-secure authentication scheme.*

PROOF: First, let $R : \{0,1\}^m \to \{0,1\}^m$ be a truly random function, and $A^{R(\cdot)}()$ an algorithm with oracle access to $R(\cdot)$ and complexity at most t. Then

$$\mathop{\mathbb{P}}_{R(\cdot)}[A^{R(\cdot)}() = (M,T) : M, T\, is\, a\, forgery] = \mathop{\mathbb{P}}_{R(\cdot)}[R(M) = T] = 2^{-m}.$$

Now, define an algorithm $A'^{O(\cdot)}$ that returns 1 iff $O(M) = T$, where M, T are the values computed by $A^{O(\cdot)}()$. Then

$$|\,\mathbb{P}[A^{R(\cdot)}\, is\, a\, forgery] - \mathbb{P}[A^{F_K(\cdot)}\, is\, a\, forgery]| = |\,\mathbb{P}[A'^{R(\cdot)} = 1] - \mathbb{P}[A'^{F_K(\cdot)} = 1]| \leq \epsilon$$

Where the last inequality is due to the definition of a pseudo-random function. From this it follows that

$$\begin{aligned}
\mathbb{P}[A^{F_K(\cdot)}\, is\, a\, forgery] &\leq \mathbb{P}[A^{R(\cdot)}()\, is\, a\, forgery] \\
&\quad +|\,\mathbb{P}[A^{R(\cdot)}\, is\, a\, forgery] - \mathbb{P}[A^{F_K(\cdot)}\, is\, a\, forgery]| \\
&\leq 2^{-m} + \epsilon
\end{aligned}$$

□

3 Construction for Messages of Arbitrary Length

Suppose we now have a longer message M, which we write as $M := M_1, \ldots, M_\ell$ with each block M_i being of the same length as the input of a given pseudorandom function.

There are various simple constructions we described in class that do not work. Here are some examples:

Example 4 *$Tag(K, M) := F_K(M_1), \ldots, F_K(M_\ell)$. This authentication scheme allows the advisary to rearrange, repeate, or remove blocks of the message. Therefore it is insecure.*

Example 5 *$Tag(K, M) := F_K(1, M_1), \ldots, F_K(\ell, M_\ell)$. This authentication scheme prevents the advisary from reordering blocks of the message, but it still allows the advisary to truncate the message or to interleave blocks from two previously seen messages.*

Example 6 *$Tag(K, M) := r, F_K(r, 1, M_1), \ldots, F_K(r, \ell, M_\ell)$. This scheme adds a randomized message identifier, and it prevents interleaving blocks from different messages, but it still fails to protect the message from being truncated by the advisary.*

The following construction works:

Let $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^m$ be a pseudorandom function, M be the message we want to tag, and write $M = M_1, \ldots, M_\ell$ where each block M_i is $m/4$ bits long.

- $Tag(K, M)$:
  - Pick a random $r \in \{0,1\}^{m/4}$,
  - output $r, T_1, \ldots, T_\ell$, where

$$T_i := F_K(r, \ell, i, M_i)$$

- $Verify(K, (M_1, \ldots, M_\ell), (r, T_1, \ldots, T_\ell)$:
  - Output $True$ if and only if $T_i = F_K(r, \ell, i, M_i)$

Theorem 7 *If F is (t, ϵ)-secure, the above scheme is $(\Omega(t), \epsilon + t^2 \cdot 2^{-m/4} + 2^{-m})$-secure.*

PROOF: Define (T, V) as the authentication scheme above. Define $(\bar{T}, \bar{V})$ in the same way, except using a truly random function R in place of F_K. Let A be an algorithm of complexity at most t.

Consider $A^{\bar{T}}$. Note that A can make at most t oracle queries. Define `FORGE` as the event in which $A^{\bar{T}}$ never queries M, and produces a tag T for which $\bar{V}(M, T) = yes$. Define `REP` as the event in which, for two different oracle queries, A receives tags with same r.

Now,

$$\begin{aligned}
\mathbb{P}[\texttt{REP}] &= \mathbb{P}[\exists \textit{ a repetition among } r_1, \ldots, r_t] \\
&\leq \sum_i \sum_j \mathbb{P}[r_i = r_j] \\
&= t^2 2^{-m/4}
\end{aligned}$$

Consider the event $\texttt{FORGE} \wedge \neg\texttt{REP}$. Suppose our oracle queries, and the resulting random strings, were:

$$\begin{aligned}
M_1^1, \ldots, M_{l_1}^1 &\to r^1 \\
M_1^2, \ldots, M_{l_2}^2 &\to r^2 \\
&\ldots
\end{aligned}$$

$$M_1^1, \ldots, M_{l_1}^1 \to r^1$$
$$M_1^2, \ldots, M_{l_2}^2 \to r^2$$
$$\ldots$$

Then we know $i \neq j \Rightarrow r^i \neq r^j$. Now, the algorithm outputs message

$$M_1, \ldots, M_\ell$$

with a valid tag

$$r, T_1, \ldots, T_\ell$$

Then there are the following cases:

- Case1: $r \neq r^i \forall i$. Then the algorithm computed $T_1 = R(r, \ell, 1, M_1)$ without having seen it before.
- Case2: r was seen before, so it occured exactly once, in the Tag for the j^{th} query.
  - Case 2a: $\ell_j \neq \ell$. Then we computed $T_1 = R(r, \ell, 1, M_1)$ without having seen it before.
  - Case 2b: $\ell_j = \ell$. We know $M \neq M^j$, so $\exists i : M_i^j \neq M_i$. thus we computed $T_i = R(r, \ell, i, M_i)$ without having seen it before

Thus, in the event $\texttt{FORGE} \wedge \neg\texttt{REP}$, we constructed some $T_i = R(r, \ell, i, M_i)$ without sending (r, ℓ, i, M_i) to the oracle. Since R is truely random, this can only occur with prbability 2^{-m}.

Now,

$$\begin{aligned}
\mathbb{P}[A^{\bar{T}}\, is\, a\, forgery] &= \mathbb{P}[\texttt{FORGE}] \\
&= \mathbb{P}[\texttt{FORGE} \wedge \texttt{REP}] + \mathbb{P}[\texttt{FORGE} \wedge \neg\texttt{REP}] \\
&\leq \mathbb{P}[\texttt{REP}] + \mathbb{P}[\texttt{FORGE} \wedge \neg\texttt{REP}] \\
&\leq t^2 2^{-m/4} + 2^{-m}
\end{aligned}$$

So finally we have

$$\begin{aligned}
\mathbb{P}[A^T()\, is\, a\, forgery] &\leq |\, \mathbb{P}[A^T()\, is\, a\, forgery] - \mathbb{P}[A^{\bar{T}}()\, is\, a\, forgery]| + \mathbb{P}[A^{\bar{T}}()\, is\, a\, forgery] \\
&\leq \epsilon + t^2 2^{-m/4} + 2^{-m}
\end{aligned}$$

□