Notes for Lecture 7 (Draft)

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.

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* $(t, \epsilon + 2^{-m})$*-secure authentication scheme.*

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.

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