Notes for Lecture 8 (Draft)

Summary

Last time we described a secure MAC (message authentication code) based on pseudorandom functions. Its disadvantage was the length of the tag, which grew with the length of the message.

Today we describe the CBC-MAC, also based on pseudorandom functions, which has the advantage of short tags. We skip its security analysis.

Next, we show that combining a CPA-secure encryption with a secure MAC gives a CCA-secure encryption scheme.

1 CBC-MAC

Suppose we have a pseudorandom function $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^m$.

Last time we described a provably secure MAC in which a message M is broken up into blocks $M_1, \ldots, M_\ell$, each of length $m/4$, and the tag of M is the sequence

$$(r, F_K(r, \ell, 1, M_1), F_K(r, \ell, 2, M_2), \ldots, F_K(r, \ell, \ell, M_\ell))$$

where r is a random string and K is the key of the authentication scheme. Jonah suggested a more compact scheme, in which M is broken into blocks $M_1, \ldots, M_\ell$ of length $m/3$ and the tag is

$$(r, F_K(r, 0, 1, M_1), F_K(r, 0, 2, M_2), \ldots, F_K(r, 1, \ell, M_\ell))$$

for a random string r. That is, the length of the message is not explicitly authenticated in each block, but we authenticate a single bit that says whether this is, or isn't, the last block of the message.

Exercise 1 *Prove that if F is (t, ϵ)-secure then this scheme is $(t/O(\ell m), \epsilon + t^2 \cdot 2^{-m/3} + 2^{-m})$-secure, where ℓ is an upper bound to the number of blocks of the message that we are going to authenticate.*

A main disadvantage of such schemes is the length of the final tag.

The CBC-MAC scheme has the advantage of producing a tag whose length is only m.

CBC-MAC scheme:

- $Tag(K, M_1, \ldots, M_\ell)$:
  - $T_0 := F_K(\ell)$
  - for $i := 1$ to ℓ: $T_i := F_K(T_{i-1} \oplus M_i)$
  - return T_ℓ
- $Verify(K, M, T)$: check that $Tag(K, M) == T$

2 Combining MAC and Encryption

Suppose that we have an encryption scheme (E, D) and a MAC (T, V). We can combine them to produce the following encryption scheme, in which a key is made of pair (K_1, K_2) where K_1 is a key for (E, D) and K_2 is a key for (T, V):

- $E'((K_1, K_2), M)$:
  - $C := E(K_1, M)$
  - $T := T(K_2, C)$
  - return (C, T)
- $D'((K_1, K_2), (C, T))$:
  - if $V(K_2, C, T)$: return $D(K_1, C)$
  - else return ERROR

The scheme (E', D') is an encrypt-then-authenticate scheme in which we first encrypt the plaintext with key K_1 and then authenticate the ciphertext with key K_2. The decryption aborts if given an incorrectly tagged ciphertext.

The idea of this scheme is that an adversary mounting a CCA attack (and hence having access to both an encryption oracle and a decryption oracle) has *no use* for the decryption oracle, because the adversary *already knows* the answer that the decryption oracle is going to provide for each oracle query:

1. if the adversary queries a ciphertext previously obtained from the encryption oracle, then it already knows the corresponding plaintext

2. if the adversary queries a ciphertext not previously obtained from the encryption oracle, then almost surely (assuming the security of the MAC), the tag in the ciphertext will be incorrect, and the oracle answer is going to be "ERROR"

This intuition is formalized in the proof of the following theorem.

Theorem 1 *If (E, D) is (t, ϵ) CPA secure, and (T, V) is $(t, \epsilon/t)$ secure, then (E', D') is $(t/(r + O(\ell)), 3\epsilon)$ CCA secure, where r is an upper bound to the running time of the encryption algorithm E and the tag algorithm T, and ℓ is an upper bound to the length of the messages that we encrypt.*