Notes for Lecture 8

Scribed by James Cook, posted February 18, 2009

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

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

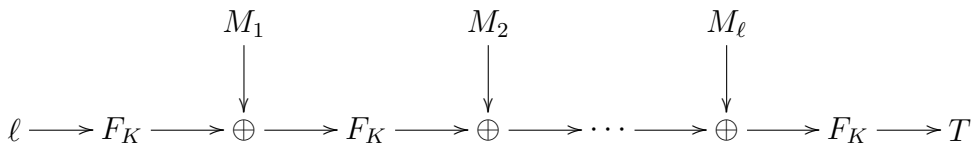
for a random string r. That is, the length of the message is not explicitly authenticated in each block, but we authenticate a single bit that says whether this is, or isn't, the last block of the message.

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.

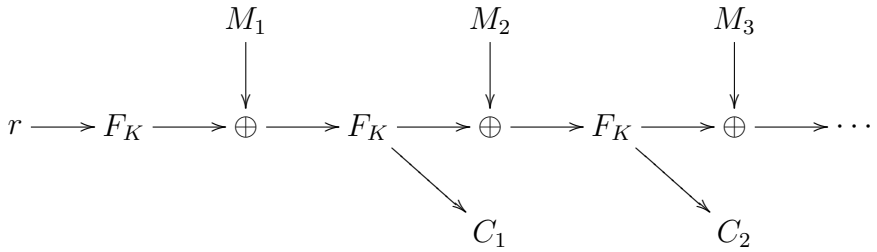
The CBC-MAC scheme has the advantage of producing a tag whose length is only m.

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 $Tag(K, M) == T$

This scheme is similar to in structure to CBC encryption:



We will not prove CBC-MAC to be secure, but the general approach is to show that all the inputs to F_K are distinct with high probability.

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.*

PROOF: Suppose (E', D') is not CCA-secure. Then there exist an algorithm A' of complexity $t' \leq t/(r + O(\ell))$ and two messages M_1 and M_2 such that

$$|\,\mathbb{P}[A'^{E', D'}(E'(M_1)) = 1] - \mathbb{P}[A'^{E', D'}(E'(M_2)) = 1]| > 3\epsilon \tag{1}$$

(In (1), $E'()$ and $D'()$ should take keys (K_1, K_2) as input; we have omitted the keys to simplify notation.)

Without loss of generality, we assume A' never queries D' on any ciphertext previously returned by E'. We can make this assumption because we can modify A' to keep a record of all the queries it makes to E', and to use the record to avoid redundant queries to D'.

We now wish to convert A' to a new algorithm A_1 such that

$$\forall M \;\; \mathop{\mathbb{P}}_{K}[A_1^{E_K}(E_K(M)) = 1] \approx \mathop{\mathbb{P}}_{K_1, K_2}[A'^{E'_{(K_1,K_2)}, D'_{(K_1,K_2)}}(E'_{(K_1,K_2)}(M))].$$

Note that A' is given the oracles E' and D', but A_1 is given as an oracle just the original CPA-secure encryption algorithm E.

Define

- $A_1^E(C)$:
  - pick a random key K_2'
  - $T := T(K_2', C)$
  - simulate $A'^{O_1,O_2}(C, T)$ with these oracles:
    - $O_1(M)$ returns $E'((K_1, K_2'), M)$;
    - O_2 always returns ERROR.

A_1 has to run the tagging algorithm T, which has complexity r, every time A' makes an oracle call. Since A' has complexity at most t/r, A_1 has complexity at most t.

Now, assuming the attack A' works, we can apply the triangle inequality to (1) to obtain:

$$3\epsilon < |\,\mathbb{P}[A'^{E',D'}(E'(M_1)) = 1] - \mathbb{P}[A_1^E(E(M_1)) = 1]| \tag{2}$$

$$+|\,\mathbb{P}[A_1^E(E(M_1) = 1] - \mathbb{P}[A_1^E(E(M_2) = 1]| \tag{3}$$

$$+|\,\mathbb{P}[A_1^E(E(M_2) = 1] - \mathbb{P}[A'^{E',D'}(E'(M_2) = 1]| \tag{4}$$

One of (2), (3) or (4) must be greater than ϵ.

If (3) $> \epsilon$, then algorithm A_1 breaks the CPA-security of E. We assumed E was CPA-secure, so one of (2) or (4) must be greater than ϵ. In either case, there exists a message M with the property that

$$|\,\mathbb{P}_K[A_1^E(E(M)) = 1] - \mathbb{P}_{K_1,K_2}[A'^{E',D'}(E'(M)) = 1]| > \epsilon. \tag{5}$$

If when A_1 is simulating A', A' never makes a call to D' which results in an output other than "ERROR", then A_1 behaves exactly as A' would with the same key K_2. So (5) implies that with probability greater than ϵ, $A'^{E',O'}(E'(M))$ makes a call to the decryption oracle resulting in an output other than "ERROR". This means A' manages to generate valid messages that it has never seen before, and we can use this fact to define an algorithm A_2 that breaks the Message Authentication Code (T, V).

$A'^{E',D'}$ makes at most t oracle queries to D', and with probability ϵ, at least one of those results in an output other than "ERROR". There must exist a number i such that with probability at least ϵ/t, the i-th query $A'^{E',D'}$ makes to D' is the first one that does not result in an error. Then define algorithm A_2 as follows.

- A_2^T (no input)
  - choose a random key K_1
  - $C := E(K_1, M)$
  - $T := T(C)$
  - simulate $A'^{O_1, O_2}(C, T)$, with the following two oracles...
    - $O_1(M_1)$:
      - $C := E(K_1, M_1)$
      - $T := T(C)$
      - return (C, T)
    - O_2 always returns ERROR

    ...until A' makes its i-th query to O_2
  - Let (C_i, T_i) be the i-th query A' made to O_2.
  - return (C_i, T_i)

Note that A_2 has complexity at most t, and by our analysis of algorithm A', A_2^T produces a correct tage for a message it has never seen before with probability at least ϵ/t.

Since we assumed (T, V) was $(T, \epsilon/t)$-secure, we have reached a contradiction: (E', D') is therefore CCA secure. $\square$