**CSE 599b: Cryptography** **(Winter 2006)**

# Lecture 15: Message Authentication

22 February 2006

*Lecturer: Paul Beame* *Scribe: Paul Beame*

## 1 Message Authentication

Recall that the goal of message authentication is to develop a tagging scheme that so that an adversary cannot modify the message being sent on a channel without being detected.

**Definition 1.1.** *A* message authentication scheme *is a triple of PPT algorithms,*

- *a key generation algorithm $\mathcal{K}$, such that $\mathcal{K}(1^k)$ is a key $K \in \{0,1\}^k$.*
- *a tagging algorithm $Tag : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ that maps a pair consisting of a key $K$ and message $M$ to a tag $\sigma = Tag_K(M) = Tag(K, M)$.*
- *a verification algorithm $Verify : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ that takes a key $K$, a message $M$, and a string $\sigma$ and produces a bit of output. We require that $Verify_K(M, \sigma) = Verify(K, M, \sigma)$ if and only if $\sigma$ is a possible output of $Tag_K(M)$.*

For $K \in \{0,1\}^k$, our definitions assume that $Tag_K : \{0,1\}^* \to \{0,1\}^*$. We also allow restricted authentication schemes in which keys of a given size can only be used to authenticate messages of up to a given size. An $L(k) \to \ell(k)$*-message authentication scheme* is one in which for each $k$, $Tag : \{0,1\}^k \times \{0,1\}^{L(k)} \to \{0,1\}^{\ell(k)}$.

A *message authemtication code* is a special case of a message authentication scheme in which the algorithm $Tag$ is deterministic and $Verify_K(M, \sigma)$ simply evaluates the predicate "$\sigma = Tag_K(M)$". We usually write $MAC_K(M)$ instead of $Tag_K(M)$.

### 1.1 Attacks on Message Authentication Schemes

**Replay Attack** In this attack the adversary, who acts as an intermediary for all traffic, simply re-sends a tagged message that has been sent previously. Unless we substantially complicate the definition above this is unavoidable. However, we can easily combat this by including sequence numbers and time stamps along with the messages that are tagged (making this a stateful scheme). Therefore we will assume that replay attacks are handled separately.

**(Adaptive) Chosen Message Attack** The adversary gets access to the tagging algorithm $Tag_K(\cdot)$ as well as $Verify_K(\cdot, \cdot)$ and afterwards must successfully produce the tag for a new message for which the tagging algorithm has not been called.

This corresponds to a situation in which the adversary by out-of-band interaction causes the sender to send tagged messages of the adversary's choosing and tries out passing some of these to the receiver to observe whether or not the receiver acted as if the message was correctly verified.

## 1.2 Existential Unforgeability under Chosen Message Attacks

**Definition 1.2.** *A message authentication scheme $(\mathcal{K}, Tag, Verify)$ is UF-CMA secure if and only if for every oracle PPT $A$, the function*

$$\epsilon(k) = \Pr[Verify_K(M,\sigma) = 1 \mid K \leftarrow \mathcal{K}(1^k);\ (M,\sigma) \leftarrow^{new} A^{Tag_K(\cdot),Verify_K(\cdot,\cdot)}(1^k)]$$

*is negligible, where the 'new' $M$ has not been given as an argument to $Tag_K(\cdot)$.*

Since $A$ is given an oracle for $Verify_K$ we can without loss of generality simplify the UF-CMA definition to simply require that $Verify_K$ never be called on the output of any input sent to $Tag_K$ and return 0 for all but (possibly) its last call which is on its output $(M, \sigma)$ and the probability $\epsilon(k)$ is simply the probability that $A$ produces a call to its $Verify_K(\cdot,\cdot)$ oracle that returns 1.

Note also that the same definition applies to $L(k) \rightarrow \ell(k)$ message authentication schemes as well as full message authentication schemes.

## 1.3 MACs based on PRFFs

The simplest way to define a MAC is using an ideal block cipher or PRFF. Given a length $\ell$ PRFF $\mathcal{F}$, define $MAC_K(M) = F_K(M)$.

**Theorem 1.3.** *If $\mathcal{F}$ is a length $\ell(k)$ PRFF and $\ell(k)$ is $\omega(\log k)$ then $MAC^{\mathcal{F}}$ is an $\ell(k) \rightarrow \ell(k)$ UF-CMA secure message authentication scheme.*

*Proof.* Let $A$ be a PPT with two oracles and let $\epsilon(k)$ be its advantage as in the UF-CMA definition. Define $B$ on input $1^k$ with oracle $f$ that simulates $A$ on input $1^k$ and whenever $A$:

- queries its $MAC_K(\cdot)$ oracle with $M_0$, $B$ passes it $f(M_0)$, or
- queries its $Verify_K(\cdot,\cdot)$ oracle on $(M_0, \sigma_0)$, $B$ passes $A$ the value of the predicate "$f(M_0) = \sigma_0$".

and $B$ outputs 1 if and only if $f(M) = \sigma$ where $(M, \sigma)$ is the output of $A$ (and has not been made as a call to $MAC_K(\cdot)$).

By the fact that $\mathcal{F}$ is a PRFF,

$$\epsilon'(k) = \Pr[B^{\mathcal{F}_k}(1^k) = 1] - \Pr[B^{\mathcal{F}unc(\ell(k),\ell(k))}(1^k) = 1]$$

is negligible. By construction, $\Pr[B^{\mathcal{F}_k}(1^k) = 1] = \epsilon(k)$.

By construction $B^f(1^k) = 1$ if and only if $A^{f(\cdot),v(\cdot,\cdot)}$ creates a new call $(M, \sigma)$ to $v$ such that $f(M) = \sigma$ having only previously received 0 as an output from $v$. For $f \leftarrow \mathcal{F}unc(\ell(k), \ell(k))$, the chance that a given call is successful the probability is $1/2^{\ell(k)}$, which yields a total value of

$\Pr[B^{\mathcal{F}unc(\ell(k),\ell(k))}(1^k) = 1]$ is at most $q(k)/2^{\ell(k)}$ where $q(k)$ is the polynomial upper bound on the number of calls that $A$ makes to its $v$ oracle. Therefore $\epsilon(k) \leq \epsilon'(k) + q(k)/2^{\ell(k)}$ which is negligible since $\ell(k)$ is $\omega(\log k)$. □

**Remark 1.4.** *This proof is typical of the proofs that we shall give, namely given a use of a PRFF in a construction, the key analysis we need to do is to determine how the construction would work if a truly random function were used in place of a PRFF.*

Although there are a number of different constructions of MACs have been given directly, there is a common thread between these constructions in that they really are constructions of PRFFs with different sizes of their domain and range. Instead of directly expressing them as MACs we will describe them as PRFF constructions as apply the above theorem to show that they are UF-CMA secure.

**Definition 1.5.** *We say that a function family $\mathcal{F} = \{\mathcal{F}_k\}_{k\geq 1}$ is an $L(k) \to \ell(k)$ PRFF if and only if for all PPT $B$,*

$$\epsilon(k) = \Pr[B^{\mathcal{F}_k}(1^k) = 1] - \Pr[B^{\mathcal{F}unc(L(k),\ell(k))}(1^k) = 1]$$

*is negligible.*

## 1.4 Cipher Block Chaining, CBC-MAC

So far, we only have a construction such that the $MAC_K(M)$ and $M$ have the same length (or a PRFF with input and output lengths the same). Using cipher block chaining we can obtain tags that are much smaller than their input size.

For $f : \{0,1\}^\ell \to \{0,1\}^\ell$ and $m$ with $|M_i| = \ell$ for $i = 1, \ldots, m$ define

$CBC_f^m(M_1 \ldots M_m)$
  $C_0 \leftarrow 0^{\ell(k)}$
  `For` $i = 1$ `to` $m$
    $C_i \leftarrow f(C_{i-1} \oplus M_i)$
  `EndFor`
  `Return` $C_m$

**Definition 1.6.** *Given a PRFF $\mathcal{F}$ and $m : \mathbb{N}^+ \to \mathbb{N}^+$, define $CBC\text{-}PRFF^m(\mathcal{F})$ by $CBC\text{-}PRFF_K^m(M_1 \ldots M_m) = CBC_{F_K}^{m(k)}(M_1 \ldots M_{m(k)})$ for $K \leftarrow \mathcal{K}(1^k)$, and $|M_i| = \ell(k)$ for $i = 1, \ldots, m = m(k)$.*

**Theorem 1.7.** *If $\mathcal{F}$ is a length $\ell(k)$ PRFF and $\ell(k)$ is $\omega(\log k)$ then for polynomial $m : \mathbb{N}^+ \to \mathbb{N}^+$ $CBC\text{-}PRFF^m(\mathcal{F})$ is an $L(k) \to \ell(k)$ PRFF for $L(K) = m(k)\ell(k)$.*

*Proof.* Let $A$ be an oracle PPT and define

$$\epsilon(k) = \Pr[A^{CBC\text{-}PRFF^m(\mathcal{F}_k)}(1^k) = 1] - \Pr[A^{\mathcal{F}unc(L(k),\ell(k))}(1^k) = 1].$$

Define $B$ that on input $1^k$ runs $A$ on input $1^k$ and uses its oracle $f : \{0,1\}^{L(k)} \rightarrow \{0,1\}^{\ell(k)}$ to run $A$ with oracle $CBC_f^{m(k)}$.

By definition,

$$\Pr[B^{\mathcal{F}_k}(1^k) = 1] = \Pr[A^{CBC\text{-}PRFF^m(\mathcal{F}_k)}(1^k) = 1]$$

and

$$\Pr[B^{\mathcal{F}unc(\ell(k),\ell(k))}(1^k) = 1] = \Pr[A^{CBC\text{-}PRFF^m(\mathcal{F}unc(\ell(k),\ell(k)))}(1^k) = 1]$$

Also, by definition,

$$\epsilon'(k) = \Pr[B^{\mathcal{F}_k}(1^k) = 1] - \Pr[B^{\mathcal{F}unc(\ell(k),\ell(k))}(1^k) = 1]$$

is negligible. Therefore

$$\epsilon(k) = \epsilon'(k) + \Pr[A^{CBC\text{-}PRFF^m(\mathcal{F}unc(\ell(k),\ell(k)))}(1^k) = 1] - \Pr[A^{\mathcal{F}unc(L(k),\ell(k))}(1^k) = 1].$$

Let $q(k)$ be the polynomial upper bound on the number of queries that $A$ makes on input $1^k$. Consider the case when $A$ receives an oracle for $CBC_f^m$ for $f \leftarrow \mathcal{F}unc(\ell(k),\ell(k))$. The claim is that, except for an event of negligible probability, at some point in each evaluation of $CBC_f^m$ the function $f$ will be called on an input on which it has never previously been called. This would imply that the output of $CBC_f^m$ will be distributed randomly in $\{0,1\}^{\ell(k)}$ independent of all previous values output which is precisely what is true for functions from $\mathcal{F}unc(L(k),\ell(k))$.

The key observation is that since $A$ does not have access to the intermediate $C_{i-1}$ values in evaluating $CBC_f^m$ for $i > 1$, and each new query yields a new random output of $\ell(k)$ bits, $A$ only can attack the psuedorandomness of the $CBC_f^m$ oracle by

- Producing known inputs to $f$ by choosing the same first coordinates $M_1$ since $C_0$ is fixed.
- Producing unknown but equal inputs to $f$ by choosing two messages $M$ and $M'$ that agree up to some prefix.
- Being lucky that the message $M$ it chose for $i > 1$ satisfies $M_i \oplus C_{i-1} = M'_j \oplus C'_{j-1}$ for some previous $M'$ and $j$.

Observe that once a call to $f$ in computing $CBC_f^m$ on its $j$-th query $M_1 \ldots M_m$ is different from every previous call to $f$ the probability that the remainder of the computation of $CBC_f^m$ yields a call that agrees with a previous call is at most $m(k)j/2^{\ell(k)}$ since each different input yields an output that is a random string. Consider $CBC_f^m$ on its $j$-th query $M_1 \ldots M_m$. Consider the sequence $C_0, C_1, \ldots, C_m$ produced by this call and consider the query $M' = M'_1 \ldots M'_m$ that has the longest agreement with $M_1 \ldots M_m$ and let $i$ be an index such $M_i \neq M'_i$. If $i = 1$ then no other query begins with the same $M_1$. By construction, the only inputs on which $A$ knows the input that has been passed to $f$ are those in the first position (since $C_0$ is fixed) and all others have been random and unknown to $A$. Since no other query begins with the same $M_1$, the chance that $A$ produces an $M_1$ that is that the same as some previous input to $f$ is only $m(k)j/2^{\ell(k)}$ since there are fewer than $m(k)j$ previous calls.

Suppose now that $i > 1$. Then by assumption $C_{i-1}$ was a randomly chosen string unknown to $A$. For any prior query $M''$ and any $i' \leq m$, $M''_1 \dots M''_{i'} \neq M_1 \dots M_i$ and in creating $M$, the value of $C''_{i'-1}$ is independent of $C_{i-1}$. Therefore the chance that $A$ can produce $M$ such that $M''_{i'} \oplus C''_{i'-1} = M_i \oplus C_{i-1}$ is only $1/2^{\ell(k)}$. There at most $m(k)j$ previous choices of $M''$ and $i'$. Therefore, in total, despite the fact that the queries to $f$ are made adaptively the chance of detection that the oracle is not a truly random function from $L(k)$ bits to $\ell(k)$ bits is at most the number of pairs of queries to $f$, which is $\leq (m(k)q(k))^2$, divided by the chance a given pair collides which is $1/2^{\ell(k)}$. Thus $\epsilon(k) \leq \epsilon'(k) + (q(k)m(k))^2/2^{\ell(k)}$ which is negligible as required. □

Define $CBC\text{-}MAC^{\mathcal{F}}_m$ to be $MAC^{CBC\text{-}PRFF^m(\mathcal{F})}$.

**Corollary 1.8.** *If $\mathcal{F}$ is a length $\ell(k)$ PRFF and $\ell(k)$ is $\omega(\log k)$ then for polynomial $m : \mathbb{N}^+ \to \mathbb{N}^+$, then $CBC\text{-}MAC^{\mathcal{F}}_m$ is an UF-CMA secure $L(k) \to \ell(k)$ MAC for $L(K) = m(k)\ell(k)$.*

**Remark 1.9.** *Note that this construction does not yield a general UF-CMA secure MAC since the input length $L(k)$ is fixed (and must be a multiple of $\ell(k)$ although this restriction is less severe). For example, given $CBC\text{-}MAC^{\mathcal{F}}$ applied to $M = M_1 \dots M_m$, one can forge $CBC\text{-}MAC^{\mathcal{F}}$ applied to $M' = M_1 \dots M_{m+1}$ by computing $CBC\text{-}MAC^{\mathcal{F}}$ applied to $C_m \oplus M_{m+1}$.*

## 1.5 Universal Hashing Approach: Hash and Hide

An alternative way to produce PRFFs that compress a wider variety of strings is to use (almost) unversal hashing. Recall the following definition of almost universal hash function families.

**Definition 1.10.** *An* almost universal $L(k) \to \ell(k)$ hash function family $\mathcal{H} = \{\mathcal{H}_k\}_{k \geq 1}$ *is given by a family of functions $\{H^k\}$ such that $H^k : \{0,1\}^k \times \{0,1\}^{L(k)} \to \{0,1\}^{\ell(k)}$ for $\ell(k) < L(k)$ such that*

- *there is a PPT algorithm $\mathcal{K}'$ that on input $1^k$ produces a random key $K$ from $\{0,1\}^k$.*
- *there is a deterministic polynomial time algorithm $H$ that for any $k$ on input a key $K \in \{0,1\}^k$ and $x \in \{0,1\}^{L(k)}$ outputs $H_K(x) = H^k_K(x)$.*
- *The function*

$$\delta(k) = \max_{x_1 \neq x_2 \in \{0,1\}^{L(k)}} \Pr[H_K(x_1) = H_K(x_2) \mid K \leftarrow \mathcal{K}'(1^k)]$$

*is negligible.*

Given an almost universal $L(k) \to \ell(k)$ hash function family $\mathcal{H}$ and a length $\ell(k)$ PRFF $\mathcal{F}$ define the function family $\mathcal{F} \circ \mathcal{H}$ by

$$(F \circ H)_{(K,K')}(x) = F_K(H_{K'}(x)).$$

**Theorem 1.11.** *If $\mathcal{H}$ is an almost universal $L(k) \to \ell(k)$ hash function family and $\mathcal{F}$ is a length $\ell(k)$ PRFF with $\ell(k)$ that is $\omega(\log k)$ then $\mathcal{F} \circ \mathcal{H}$ is an $L(k) \to \ell(k)$ PRFF.*

*Proof.* Let $A$ be a PPT algorithm that on input $1^k$ with oracle for $g$ that maps $L(k)$ bits to $\ell(k)$ bits that can distinguish $g$ chosen from $\mathcal{F}_k \circ \mathcal{H}_k$ from $g$ chosen from $\mathcal{F}unc(L(k), \ell(k))$ with probability $\epsilon(k)$. Define $\delta(k)$ to be the collision probability of $\mathcal{H}$ as above.

Define $B$ that on oracle $f$ and input $1^k$ runs $A$ on input $1^k$ and oracle for $f \circ \mathcal{H}_k$. Clearly since $\mathcal{F}$ is a PRFF, $B$ can only distinguish between $f \leftarrow \mathcal{F}_k$ from $f \leftarrow \mathcal{F}unc(\ell(k), \ell(k)$ with negligible probability $\epsilon'(k)$.

Therefore

$$\epsilon(k) = \epsilon'(k) + (\Pr[A^{\mathcal{F}unc(\ell(k),\ell(k))\circ\mathcal{H}}(1^k) = 1] - \Pr[A^{\mathcal{F}unc(L(k),\ell(k))}(1^k) = 1]).$$

As in the previous argument, the difference in the parenthesized term will be 0 except for the fraction of the time that the oracle computation of $f \circ H_{K'}$ repeats an argument to $f$. Let the sequence of queries to $f \circ H_{K'}$ be $x_1, x_2, \ldots$. By construction, no matter what $x_1$ is, the fact that $f$ is random ensures that $f \circ H_{K'}$ is independent of $x_1$ and independent of $H_{K'}(x_1)$. Therefore, since $A$ does not see $H_{K'}(x_1)$, the choice of $x_2$ that $A$ makes is independent of $K'$, etc. Provided $H_{K'}(x_i) \neq H_{K'}(x_j)$ for all $i \neq j$, the values of $f$ produced will all be random and independent of the key $K'$. Therefore, the probability of a collision for $H_{K'}$ is at most $q(k)^2\delta(k)$ where $q(k)$ is the number of different $x_j$. This quantity is negligible as required. □