COMS W4995 Introduction to Cryptography **November 18, 2003**

Lecture 22: Message Authentication Codes

Lecturer: Tal Malkin *Scribes: Gopal Ananthraman*

Summary

Digital Signature scheme summary.Definitions of Message Authentication Codes and security for MAC.Construction of fixed length MAC using PRF, extensions of fixed length MAC to larger sized messages.

1 Digital Signature schemes summary

We saw that Digital Signatures that were stateful and based on CRHF but more general constructions exist.

Theorem 1 *There exists a secure signature scheme that is stateless and based on any One Way Function. (OWF $\iff$ SIG)*

Proof: We will not show this construction. ■

Categories of Digital signatures

The ones that are provably secure, that are provably secure in the Random oracle model and the ones that are not provably secure at all (but are typically more efficient).

- Provably Secure
  - First stateful construction was given by Goldwasser,Micali and Rivest
  - First staeless construction was given by Goldreich

  - Merkle signatures(which we saw last time)
  - Naor-Yung (A tree based signature scheme which we didn't see)

  The most efficient Provably Secure Signature scheme based on strong RSA assumption was presented by Cramer-Shoup in 2000 with improvement from Fischilin in 2002. A strong RSA assumption states it is hard to take *any* root for *any* exponent instead of a fixed or randomly chosen exponent as in regular RSA assumption.

- Provably secure in the Random Oracle model These are an attempt to get more efficient as those that are not provably secure.
  - Full domain hash
  - Fiat Shamir
  - Schnorr signatures
  - Guillou-Quisquater (GQ) signatures
- Not provably secure at all but used practically
  - Digital Signature Standard/ Digital Signature Algortithm (DSS/DSA)
    Variation of ElGamal.We don't know how to show they are insecure, based heuristically (but not provably) on DLOG Assumption.
  - PKCS#1 RSA Signatures
    Variation of Full-Domain Hash(but not even provably secure in RO model)

Digital Signature Schemes is to show authenticity based on Public key model (PKE). We will move on to Message Authentication Code(MAC) which is based on Private Key model. MAC is better efficiency wise.

2 Message Authentication Code (MAC)

Similar to digital signatures, Message authentication code sign a message. Unlike digital signature schemes MACs use a Private/Secret Key Setting with same key used for signing and verification.

Definition 1 *A Message Authentication code consists of three algorithms* $\{GEN, TAG, VER\}$ *such that:*

- *GEN (The key generation algorithm) - a possibly randomized algorithm that on input* 1^k *outputs a key* K *(note that this is different from the security parameter* k*).*

  $K \overset{\mathrm{R}}{\leftarrow} GEN(1^k)$

- *TAG - a possibly randomized algorithm that on input* m*,*K *output a signature* σ*.*

$$\sigma \leftarrow TAG_K(m)$$

- *VER (The Verification algorithm) - A deterministic algorithm takes as input* K*,*m*,*σ *outputs accept or reject.*

$$\{accept, reject\} \leftarrow VER_K(m, \sigma)$$

Correctness Requirement.
Associated with the scheme is a message space M_k *we require* $\forall m \in M_k$ *and* $\forall K \leftarrow GEN(1^k)$*,*

$$VER_K(m, TAG_K(m)) = accept.$$

For every TAG that VER accepts, the TAG is valid, otherwise, the TAG is invalid.

Sometimes the tag will be very short. We want to define the security of the MAC scheme. Same as for Digitial Signatures. We want security against existential forgery under adaptive chosen message attack.

Security Definition

Definition 2 *A MAC* $\{GEN, TAG, VER\}$ *is secure*
(existentially unforgeable under adaptive chosen message attack)
if $\forall$ *PPT* A $\exists$ *a negligible* ε*, such that*

$$\Pr\left[K \overset{\mathrm{R}}{\leftarrow} GEN(1^k) \ : \ A^{TAG()}(1^k) forges\right] \leq \varepsilon(k)$$

Where "A forges" means A outputs (m, σ) *such that:*

- $VER_K(m, \sigma) = accept.$
- m *was never a query to the TAG oracle*

Note that A doesn't get the key K.

3 MAC Construction

Idea is to just use PRF. They are simple deterministic, stateless and operate on fixed-length messages and produce fixed-length MAC.

Let $\mathcal{F}_k = \{f_s : \{0,1\}^k \to \{0,1\}^k\}_{s \in \{0,1\}^k}$ be a PRF family.

- $GEN(1^k)$ choose random $s \in \{0,1\}^k$
- $TAG_s(m) = f_s(m)$ for $m \in \{0,1\}^k$.
- $VER_s(m, \sigma) =$ Accept, iff $\sigma = f_s(m)$, reject otherwise.

shows why PRFs are useful.

Theorem 2 *If $F = \cup_k \mathcal{F}_k$ is a PRF. Then the above construction is secure.*

Proof sketch: First prove that if a truly random function is used , any adversary A's probability to forge is $\frac{1}{2^k}$. Why? because if you use a truly random function the TAG is just $f(m)$ for truly random f. For a truly random function since every output is equaly likely we have the probability of finding the right σ for message m is $\frac{1}{2^k}$. Now prove that for PRF, A can forge with probability $\leq \frac{1}{2^k}+$ negligible function $=$ negligible. Way you would you prove, is assuming if that's not the case then you can distinguish truly random function from PRF.

■ It follows frome the above theorem, if you believe a block cipher (e.g AES, DES) to be a Pseudo Random Permutation(PRP) then this immediately gives you a MAC (use them as TAG)

4 MAC for Arbitrary length messages

Assume we have a MAC $\{GEN, TAG, VER\}$ is secure for k-bit messages, how do we construct a MAC for longer messages?

- Attempt 1. (not secure)
  Write $m = m_1 m_2 ... m_l$ where $|m_i| = k$ sized blocks and do $TAG_K(m) = TAG_K(m_1 \oplus m_2 ... \oplus m_l)$ Not secure. why? because what you are authenticating is the $\oplus$ of the message blocks but not the message itself. So you can always choose a message whose $\oplus$ is the same as some other message.

- Attempt 2. (not secure)
  Concatenate the TAG values of the blocks. Not Secure. Because you can rearrange the message blocks/ TAG block and get a different MAC. Given one message with a TAG you can forge a valid TAG for another message

- Attempt 3.(not secure)
  $TAG_K(m) = TAG_K(1m_1)TAG_K(2m_2)TAG_K(3m_3)...$ concatenate the index of the block to each block and then call TAG. This prevents the reordering attack from above however it still fails under chosen message attack. Given two messages with valid TAGs one can forge a valid TAG for a third message.

- Attempt 4. (Secure)
  See Secure MAC based on random identifier subsection

Secure MAC based on random identifier

Write $m = m_1 m_2 ... m_l$ in l blocks where $|m_i| = k/3$
Let $r \in \{0,1\}^{\frac{k}{3}}$ be a random identifier. We will add the index of a block, the random identifier r and the length of the block along with the message block for calculating the TAG.

- Output.
  $TAG_K(m) = (r, TAG_K(m_1, 1, l, r), TAG_K(m_2, 2, l, r), ..., TAG_K(m_l, l, l, r))$

- Verification.
  Will use the key K and r and check whether the TAG is correct.

We need to fix the length of inputs to TAG so that they fit in TAG's domain. You can choose the block size l,r of size $k/3$. Why do we need l? Otherwise You can forge a shorter message. This works for messages of length upto $\frac{k2^{k/6}}{3}$ bits which is sufficient. l is utmost $k/6$ bits so can have a value of l upto $2^{k/6}$.

Theorem 3 *This is a Secure MAC for messages of length* $\frac{k2^{k/6}}{3}$.

We will not prove this, but the proof is based on the fact that there is a very high probability that every TAG will have a different identifier.

Hash-Then-MAC

Use $TAG'_K(m) = TAG_K(h(m))$. using hash function h. Works for h being a CRHF. Use same proof as for signatures. Infact, don't need CRHF (and not even UOWHF). Since h_i need not be known to adversary all you need is a hash family $\{h_i\}_I$: $\forall x \neq x' Prob\big[i \overset{R}{\in} I \ : \ h_i(x) = h_i(x')\big] \leq \varepsilon$ is negligible. This is possible without any cryptographic assumptions.For e.g h_i could be a random linear mapping etc. (Note that this even stronger because the definition is for any x and x' not just those generated by a poly-time adversary. But this still can be achieved cryptographically).

CBC-MAC

Another secure MAC is the Cipher Block Chaining MAC (CBC-MAC), which is described in the next lecture.