COM S 6830 – Cryptography Nov 15, 2011

Lecture 20: Authentication

Instructor: Rafael Pass *Scribe: Sidharth Telang*

1 Motivation

Imagine the scenario Alice communicating with Bob in the presence of mailicious Eve. Bob wishes to make sure what he recieves from the communication channel is indeed what Alice said. Sending the message alone will not work as Eve can tamper with this message and send another message instead. We define and construct authentication primitives to help Alice and Bob in this scenario.

2 MAC

A MAC is an authentication primitive that can be used if Alice and Bob can share a secret key.

Definition 1 *(Gen, Tag, Ver) is a MAC if Gen and Tag are PPT algorithms, Ver is a polynomial time algorithm whose output is in $\{0,1\}$ and $\forall n \in N$, $m \in \{0,1\}^n$ we have $Pr[k \leftarrow Gen(1^n), \sigma \leftarrow Tag_k(m) : Ver(m, \sigma) = 1] = 1$*

Intuitively, we should say our MAC is secure when Eve cannot create a valid tag for any message other than messages whose tags are already known to Eve. These could be messages that have already been sent to Bob by Alice, and it's possible Eve could choose these messages for Alice.

Definition 2 *A MAC (Gen, Tag, Ver) is secure if for every n.u.PPT A there exists negligible function ϵ such that $\forall n \in N$ we have $Pr[k \leftarrow Gen(1^n), (m, \sigma) \leftarrow A^{Tag_k(), Ver_k()} :$ A didn't query $Tag_k()$ on $m \wedge Ver_k(m, \sigma) = 1]$*

In the above definition the attacker is given access to a tagging oracle $Tag_k()$ and a verifying oracle $Ver_k()$.

Theorem 1 *The existence of pseudorandom functions implies the existence of secure MAC*

Proof. Let $\{f_s : \{0,1\}^{|s|} \to \{0,1\}^{|s|}\}_{s \in \{0,1\}^*}$ be a family of pseudorandom functions. We define (Gen, Tag, Ver) as follows. $Gen(1^n)$ outputs $k \leftarrow \{0,1\}^n$. $Tag_k(m)$ outputs $f_k(m)$. $Ver(m, \sigma)$ outputs 1 iff $\sigma = f_k(m)$. Security follows from the pseudorandomness of $\{f_s\}$, the fact that if Tag was a random function the success probability of any attacker would be negligible and the closure of oracle indistinguishability under efficient operations.

3 Digital Signatures

Digital signatures can be thought of as the public key analogue of authentication. It can be used by Alice and Bob if Alice has a secret key and the corresponding verification key is public.

Definition 3 *$(Gen, Sign, Ver)$ is a Digital Signature scheme (DSS) if Gen and $Sign$ are PPT algorithms, Ver is a polynomial time algorithm whose output is in $\{0,1\}$ and $\forall n \in N$, $m \in \{0,1\}^n$ we have $Pr[(vk, sk) \leftarrow Gen(1^n), \sigma \leftarrow Sign_{sk}(m) : Ver_{vk}(m, \sigma) = 1] = 1$*

Security for digital signatures is defined in a way similar to MAC

Definition 4 *A DSS $(Gen, Sign, Ver)$ is secure if for every n.u.PPT A there exists negligible function ϵ such that $\forall n \in N$ we have $Pr[(vk, sk) \leftarrow Gen(1^n), (m, \sigma) \leftarrow A^{Sign_{sk}()} : A$ didn't query $Sign_{sk}()$ on $m \wedge Ver_{vk}(m, \sigma) = 1]$*

A first attempt to achieve a DSS would be to use a family of trapdoor permutations (TDP) $\{f_i\}_{i \in I}$ such that signing a message m under secret key sk should be computing $f_i^{-1}(m)$ using the trapdoor sk, where i is the verification key vk. However we require this to be hard to invert for all m, while TDP guarantees only average case hardness.

One could argue that the messages on which the trapdoor permutation is easy to invert could be messages of no practical importance i.e. messages that won't be sent with the authentication protocol. The same reasoning could justify using the RSA TDP as it is as a DSS. With RSA, given valid (m_1, σ_1) and (m_2, σ_2), it is easy to find a valid signature $\sigma_3 = \sigma_1 \sigma_2$ for message $m_1 m_2$. However, a better heuristic to deal with this is to hash the message before using RSA. *Gen* generates RSA key pair (vk, sk) where $vk = (N, e)$ and $sk = d$. $Sign(sk, m)$ returns $H(m)^d \ mod \ N$ and $Ver(vk, m, \sigma)$ returns 1 iff $\sigma^e = H(m) \ mod \ N$. Assuming this hash function behaves like a random function, we can prove this DSS is secure.

It is possible to construct a secure DSS assuming one way functions. We begin by first constructing a one-time secure DSS from one way functions

Definition 5 *A DSS $(Gen, Sign, Ver)$ is one time secure if for every n.u.PPT A that makes a single query to the signing oracle there exists negligible function ϵ such that $\forall n \in N$ we have $Pr[(vk, sk) \leftarrow Gen(1^n), (m, \sigma) \leftarrow A^{Sign_{sk}()} : A$ didn't query $Sign_{sk}()$ on $m \wedge Ver_{vk}(m, \sigma) = 1]$*

Theorem 2 *The existence of one way functions implies one time secure DSS*

Proof. Let f be a OWF. $Gen(1^n)$ returns (sk, vk) where $sk = \{(X_i^0, X_i^1)\}_{i=1}^n$, $vk = \{(f(X_i^0), f(X_i^1))\}_{i=1}^n$ and for every $i \in [n]$, $X_i^0, X_i^1 \leftarrow \{0,1\}^n$. To sign a message $m \in \{0,1\}^n$ $Sign_sk(m)$ just returns $\{\sigma_i\}_{i=1}^n$ where $\sigma_i = X_i^{m_i}$ where m_i is the i^{th} bit of m. Ver outputs 1 iff $f(\sigma_i) = f(X_i^{m_i})$ for every $i \in [n]$.
Given an n.u.PPT A that breaks the one time security of the above DSS with probability $1/p(n)$ for some polynomial p, we can invert $f(X)$ where $X \leftarrow \{0,1\}^n$ with probability $1/2np(n)$ in the following way: Set $f(X)$ to be Y_i^b for a random $i \in [n]$ and $b \in \{0,1\}$. With probability 1/2 A asks the signing oracle a query whose i^{th} bit is not b, in which case we can answer the oracle query, otherwise we fail. A outputs a message different from the oracle query on atleast one bit position. With probability $\geq 1/n$ i is among those bit positions.

We wish to define the security of Hash functions in a way that even non uniform adversaries cannot find collisions in them. We hence define collision resistance for a family of hash functions.

Definition 6 $H = \{h_i : D_i \rightarrow R_i\}_{i \in I}$ *is a collection of Collision resistant hash functions (CRH) if there exists PPT* Gen : $Gen(1^n) \in I$, $|R_i| < |D_i|$, h_i *can be efficiently computed and for every n.u.PPT* A *there exists negligible function* ϵ *such that* $\forall n$, $Pr[i \leftarrow Gen(1^n), (X, X') \leftarrow A(1^n, i) : h_i(X) = h_i(X')] \leq \epsilon(n)$