COM S 6830 – Cryptography Nov 10, 2009

Lecture 20: Digital Signatures

Instructor: Rafael Pass *Scribe: Edward Lui*

1 Definitions

Digital signatures are the digital equivalent of hand-written signatures. We want digital signatures to have the following properties:

- Public verification: Anyone can verify the correctness of a digital signature.
- Non-reputability: The signer of a message cannot deny that he or she had signed the message.
- "Hierarchical": Can be used in a public key infrastructure. (e.g. certificates)

Definition 1 *$(Gen, Sign, Ver)$ is a* digital signature scheme *over the message space $\{M_n\}_{n\in\mathbb{N}}$ if*

- *Gen is a PPT algorithm: $(pk, sk) \leftarrow Gen(1^n)$*
- *$Sign$ is a PPT algorithm: $\sigma \leftarrow Sign_{sk}(m)$*
- *Ver is a deterministic PT algorithm: $Ver_{pk}(m, \sigma) \in \{0, 1\}$*
- $\forall n \in \mathbb{N}$, $\forall m \in M_n$, $\Pr[(pk, sk) \leftarrow Gen(1^n); \sigma \leftarrow Sign_{sk}(m) : Ver_{pk}(m, \sigma) = 1] = 1$.

For a digital signature scheme to be *secure*, we also want the following property:

- Unforgeability: $\forall$ non-uniform PPT machine A, $\exists$ a negligible function ϵ s.t. for all n,

$$\Pr[(pk, sk) \leftarrow Gen(1^n); (m, \sigma) \leftarrow A^{Sign_{sk}(\cdot)}(1^n, pk) : \\ (Ver_{pk}(m, \sigma) = 1) \wedge (A \text{ didn't query } m)] \leq \epsilon(n).$$

2 Possible Constructions

We first show some constructions of digital signature schemes that are not secure.

Trapdoor Permutation (TDP):

- $Gen(1^n)$: Run $Gen_{TDP}(1^n)$ to get (i, t), where i is the index (of the TDP) and t is the trapdoor. Let $pk = i$ and $sk = t$. Output (pk, sk).

- $Sign_{sk}(m) = f_{pk}^{-1}(m)$ [using the trapoor sk]
- $Ver_{pk}(m, \sigma)$: Check $f_{pk}(\sigma) = m$.

"Attack": A picks r, computes $f_{pk}(r)$, and outputs $(f_{pk}(r), r)$.

RSA:

- $Gen(1^n)$: Let $pk = (N, e)$ and $sk = d = e^{-1} \mod \phi(N)$, where N, e, and d are chosen as in RSA. Output (pk, sk).
- $Sign_{sk}(m) = m^d \mod N$.
- $Ver_{pk}(m, \sigma)$: Check $\sigma^e = m \pmod N$.

"Attack": Query $Sign$ oracle on m_1 and m_2 to get ${m_1}^d$ and ${m_2}^d$. Multiply ${m_1}^d$ and ${m_2}^d$ to get $(m_1 m_2)^d$. Output $(m_1 m_2, (m_1 m_2)^d)$. (This attack is even better than the previous one, since $m_1 m_2$ is not "random".)

Construction in Practice:

- TDP (trapdoor permutation): f
- RO (random oracle): $\mathcal{O}$
- $Gen(1^n)$: $(i, t) \leftarrow Gen_{TDP}(1^n)$. Let $pk = i$ and $sk = t$. Output (pk, sk).
- $Sign_{sk}(m) = f_i^{-1}(\mathcal{O}(m))$
- $Ver_{pk}(m, \sigma)$: Check $\mathcal{O}(m) = f_i(\sigma)$.

In practice, substitute $\mathcal{O}$ by some candidate random function.

3 One-Time Signatures

Definition 2 *A digital signature scheme is said to be* one-time secure *if it is secure when the adversary only queries the signing oracle once.*

Construction: Based on OWF f.

- $Gen(1^n)$:
  $m_1^0, m_2^0, \ldots, m_n^0 \leftarrow \{0,1\}^n$
  $m_1^1, m_2^1, \ldots, m_n^1 \leftarrow \{0,1\}^n$
  $sk = \begin{pmatrix} m_1^0 & m_2^0 & m_3^0 & \ldots & m_n^0 \\ m_1^1 & m_2^1 & m_3^1 & \ldots & m_n^1 \end{pmatrix}$

$$pk = f(sk) = \begin{pmatrix} f(m_1^0) & f(m_2^0) & f(m_3^0) & \dots & f(m_n^0) \\ f(m_1^1) & f(m_2^1) & f(m_3^1) & \dots & f(m_n^1) \end{pmatrix}$$

Output (pk, sk).

- $Sign_{sk}(s)$: For $i = 1, \dots, n$, let $\sigma_i = m_i^{s_i}$. Output $\sigma = (\sigma_1, \dots, \sigma_n)$.
- $Ver_{pk}(s, \sigma)$: For $i = 1, \dots, n$, check that $f(\sigma_i) = f(m_i^{s_i})$.

This signature scheme is clearly not 2-time secure. E.g., by querying the signing oracle on 0^n and 1^n, one can recover the secret key sk; then, using sk, one can sign any message one desires. This signature scheme, however, is one-time secure.

Intuition: If A queries $Sign_{sk}(s)$ and outputs $(s', Sign_{sk}(s'))$, then let i be s.t. $s_i \neq s'_i$. Then, A has "inverted $f(m_i^{s'_i})$".

Proof. Suppose a non-uniform PPT machine A succeeds with probability $\epsilon(n)$ in breaking the one-time signature scheme. WLOG, we can assume that A always makes at least one query, and A never outputs the signature of a message that it has already queried. Using A, we will construct a non-uniform PPT machine B that inverts f with probability $\frac{\epsilon(n)}{2n}$. On input $(1^n, y)$, B chooses a random $i \in \{1, \dots, n\}$ and $b \in \{0, 1\}$. Then, B runs $Gen(1^n)$ to get (pk, sk), but B replaces $f(m_i^b)$ in pk by y. Then, B runs A on input $(1^n, pk)$, and if A makes the query m, B answers the query with $Sign_{pk}(m)$ if $m_i \neq b$; otherwise, B aborts and outputs $\perp$, since B does not know the inverse of y. A eventually outputs (m', σ'). If $m'_i = b$, output σ'_i; otherwise, B outputs $\perp$.

We note that the distribution of pk is the same regardless of the value of i and b that B chose. I.e., pk is independent of i and b. Thus, with probability $\frac{1}{2}$, we have $m_i \neq b$, and with probability at least $\frac{1}{n}$, we have $m'_i = b$ (since m and m' must differ in at least one position). Thus, with probability $\frac{1}{2n}$, B does not output $\perp$; in this case, A outputs (m', σ') s.t. $\sigma'_i = f^{-1}(y)$ with probability at least $\epsilon(n)$. Thus, B successfully inverts y (under f) with probability at least $\frac{\epsilon(n)}{2n}$. Since f is a one-way function, ϵ must be negligible.

■

This signature scheme is bad because pk and sk are too long: $O(|m_i^b| \times \text{message length})$. We will now construct a better one-time secure signature scheme, using a cryptographic primitive called a collision-resistant hash function.

4 Collision-Resistant Hash Functions

Definition 3 *A CRH (collision-resistant hash) function* $h\colon \{0,1\}^n \to \{0,1\}^m$ *satisfies*

1. *Length compression:* $m < n$ *(typically* $m = \frac{n}{2}$*)*

2. *Easy to evaluate: h can be computed in PPT.*

3. *Hard to find collisions: For every uniform PPT machine A, there exists a negligible function ϵ s.t. for all n, $\Pr((x,x') \leftarrow A(1^n) : x,x' \in \{0,1\}^m$ and $h(x) = h(x')$ and $x \neq x') \leq \epsilon(n)$.*

Note: For non-uniform adversaries, we require a family of CRH functions, since for every n, a non-uniform adversary can simply hardcode $x, x' \in \{0,1\}^m$ s.t. $h(x) = h(x')$ and $x \neq x'$.

Definition 4 *A collection of functions $H = \{h_i \colon D_i \to R_i\}_{i \in I}$ is a family of collision-resistant hash (CRH) functions if the following hold:*

- *Easy to sample: Gen is a PPT algorithm: $Gen(1^n) \in I$*
- *Length compression: $|R_i| < |D_i|$.*
- *Easy to evaluate: There exists a PPT algorithm that, given $i \in I$ and $x \in D_i$, computes $f_i(x)$.*
- *Hard to find collisions: For every non-uniform PPT algorithm A, there exists a negligible function ϵ s.t. for all n, $\Pr(i \leftarrow Gen(1^n); (x,x') \leftarrow A(1^n, i) : h_i(x) = h_i(x')$ and $x \neq x') \leq \epsilon(n)$.*

e.g. Java: $H(x) = \sum_{i=1}^{n} x_i \cdot 31^{n-i} \mod 2^{32}$

Let $(Gen, Sign, Ver)$ be the one-time secure signature scheme that we constructed above. Let $\mathcal{H} = \{h_i \colon \{0,1\}^* \to \{0,1\}^n\}_i$ be a family of CRH functions.

Damgård's Construction:

- $Gen'(1^n)$: $(pk, sk) \leftarrow Gen(1^n)$; $i \leftarrow Gen_{CRH}(1^n)$; let $pk' = (pk, i)$ and $sk' = (sk, i)$. Output (pk', sk').
- $Sign'_{sk'}(m) = Sign_{sk}(h_i(m))$
- $Ver'_{pk'}(m, \sigma) = Ver_{pk}(h_i(m), \sigma)$

Intuition: Suppose A finds a signature σ' for a message m' by querying m.
If $h(m) = h(m')$, A breaks the collision-resistance of the CRH function; otherwise, A breaks the one-time security of $(Gen, Sign, Ver)$.