EE 720: An Introduction to Number Theory and Cryptography (Spring 2018)

Lecture 24 — April 18, 2018

Instructor: Saravanan Vijayakumaran *Scribe: Saravanan Vijayakumaran*

1 Lecture Plan

- Digital Signatures

2 Digital Signature Schemes

- Digital signature schemes allow a *signer* S to sign a message using his private key sk. Anyone who has the signer's public key pk can verify that the message originated from S and was not modified in transit.
- For example, a software company like Microsoft can sign updates to its operating system. Users who know Microsoft's public key can verify that these updates are authentic, i.e. they are generated by an entity who knows the corresponding private key.
- Both MACs and digital signature schemes ensure integrity of the transmitted messages. But signature schemes have the added advantage of *public verifiability* and *non-repudiation*.

Definition. *A **digital signature scheme** consists of three PPT algorithms* $(\texttt{Gen}, \texttt{Sign}, \texttt{Vrfy})$ *such that:*

1. *The **key-generation algorithm** $\texttt{Gen}$ takes as input the security parameter 1^n and outputs a pair of keys (pk, sk). These are called the **public key** and the **private key**, respectively. We assume that pk and sk each has length at least n, and that n can be determined from pk or sk.*
2. *The **signing algorithm** $\texttt{Sign}$ takes as input a private key sk and a message m from some message space (that may depend on pk). It outputs a signature σ, and we write this as $\sigma \leftarrow \texttt{Sign}_{sk}(m)$.*
3. *The deterministic **verification algorithm** $\texttt{Vrfy}$ takes as input a public key pk, a message m, and a signature σ. It outputs a bit b, with $b = 1$ meaning **valid** and $b = 0$ meaning **invalid**. We write this as $b = \texttt{Vrfy}_{pk}(m, \sigma)$.*

It is required that except with negligible probability over (pk, sk) output by $\texttt{Gen}(1^n)$, it holds that $\texttt{Vrfy}_{pk}(m, \texttt{Sign}_{sk}(m)) = 1$ for every (legal) message m.

*If there is a function l such that for every (pk, sk) output by $\texttt{Gen}(1^n)$ the message space is $\{0,1\}^{l(n)}$, then we say that $(\texttt{Gen}, \texttt{Sign}, \texttt{Vrfy})$ is a **signature scheme for messages of length** $l(n)$.*

2.1 Security Definition

- For a fixed public key pk generated by a signer S, a *forgery* is a message m along with a valid signature σ, where m was not previously signed by S.
- Security of a signature scheme means that an adversary should be unable to output a forgery even if it obtains signatures on many other messages of its choice. It is formalized in the following experiment.
- **The signature experiment** $\texttt{Sig-forge}_{\mathcal{A},\Pi}(n)$:
  1. $\texttt{Gen}(1^n)$ is run to obtain keys (pk, sk).
  2. Adversary $\mathcal{A}$ is the public key pk and oracle access to $\texttt{Sign}_{sk}(\cdot)$. The adversary eventually outputs (m, σ). Let $\mathcal{Q}$ denote the set of all queries that $\mathcal{A}$ asked its oracle.
  3. $\mathcal{A}$ succeeds if and only if (1) $\texttt{Vrfy}_{pk}(m, \sigma) = 1$ and (2) $m \notin \mathcal{Q}$. If $\mathcal{A}$ succeeds, the output of the experiment is 1. Otherwise, the output is 0.

Definition. *A signature scheme* $\Pi = (\texttt{Gen}, \texttt{Sign}, \texttt{Vrfy})$ *is* ***existentially unforgeable under an adaptive chosen-message attack****, or just* ***secure****, if for all PPT adversaries* $\mathcal{A}$*, there is a negligible function* $\texttt{negl}$ *such that:*

$$\Pr\left[\texttt{Sig-forge}_{\mathcal{A},\Pi}(n) = 1\right] \leq \texttt{negl}(n).$$

3 RSA Signatures

3.1 Plain RSA

- Let $\texttt{GenRSA}$ be a PPT algorithm that on input 1^n, outputs a modulus N that is the product of two n-bit primes, along with integers $e, d > 1$ satisfying $ed = 1 \bmod \phi(N)$.
- Define a signature scheme as follows:
  - $\texttt{Gen:}$ On input 1^n run $\texttt{GenRSA}(1^n)$ to obtain N, e, and d. The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$.
  - $\texttt{Sign:}$ On input a private key $sk = \langle N, d\rangle$ and message $m \in \mathbb{Z}_N^*$, compute the signature
    $$\sigma = m^d \bmod N.$$
  - $\texttt{Vrfy:}$ On input a public key $pk = \langle N, e\rangle$, a message $m \in \mathbb{Z}_N^*$, and a signature $\sigma \in \mathbb{Z}_N^*$, output 1 if and only if
    $$m = \sigma^e \bmod N.$$

3.1.1 Attacks on Plain RSA

- **A no-message attack:** Given an public key $\langle N, e\rangle$, adversary chooses a uniform $\sigma \in \mathbb{Z}_N^*$ and computes $m = \sigma^e \bmod N$. He then outputs (m, σ) as a forgery. This is a forgery as no messages were issued by the owner of the public key.
- **Attack using two signatures:** Suppose the adversary can make the signer sign two messages $m_1, m_2 \in \mathbb{Z}_N^*$ and get the corresponding signatures σ_1, σ_2. The adversary outputs $(m_1 m_2, \sigma_1 \sigma_2)$ as the forgery.

3.2 RSA Full-Domain Hash

- To prevent the attacks on plain RSA, a deterministic function H which acts as a *random oracle* is specified as part of the public key.
- A random oracle is a function $H : \{0,1\}^* \mapsto \{0,1\}^*$ which satisfies the following properties:
  - For a given input x which has not been queried before and given output length l, the output $H(x)$ is generated by randomly picking l uniformly distributed bits.
  - If x is a **repeated query**, the same $H(x)$ is returned.
- Define a signature scheme as follows:
  - `Gen:` On input 1^n run $\texttt{GenRSA}(1^n)$ to obtain N, e, and d. The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$.

    As part of the key generation, a function $H : \{0,1\}^* \mapsto \mathbb{Z}_N^*$ is specified.
  - `Sign:` On input a private key $sk = \langle N, d\rangle$ and message $m \in \{0,1\}^*$, compute the signature
    $$\sigma = H(m)^d \bmod N.$$
  - `Vrfy:` On input a public key $pk = \langle N, e\rangle$, a message m, and a signature $\sigma \in \mathbb{Z}_N^*$, output 1 if and only if
    $$\sigma^e = H(m) \bmod N.$$
- What properties should H have?
  - H should be difficult to invert, otherwise $(m, \sigma) = \left(H^{-1}(\sigma^e), \sigma\right)$ is a forgery.
  - It should be difficult to find three messages m, m_1, m_2 such that $H(m) = H(m_1)H(m_2)$.
  - It should be difficult to find **distinct** messages m_1, m_2 such that $H(m_1) = H(m_2)$.

Theorem. *If the RSA problem is hard relative to* ***GenRSA*** *and H is modeled as a random oracle, then the above construction is secure.*

4 References and Additional Reading

- Sections 12.1,12.2,12.4 from Katz/Lindell
- Section 5.5 from Katz/Lindell