Introduction to Cryptology

Lecture 24

Announcements

- HW 10 due on 5/10
- Scholarly Paper EC due on 5/10

Agenda

- Last time:
  - Public Key Encryption (11.3)
  - El Gamal Encryption (11.4)
  - RSA Encryption and Weaknesses (11.5)

- This time:
  - Digital Signatures Definitions (12.2-12.3)
  - RSA Signatures (12.4)
  - Dlog-based signatures (12.5)

Padded RSA

CONSTRUCTION 11.29

Let GenRSA be as before, and let ℓ be a function with $\ell(n) \leq 2n-4$ for all n. Define a public-key encryption scheme as follows:

- Gen: on input 1^n, run GenRSA(1^n) to obtain (N, e, d). Output the public key $pk = \langle N, e \rangle$, and the private key $sk = \langle N, d \rangle$.
- Enc: on input a public key $pk = \langle N, e \rangle$ and a message $m \in \{0,1\}^{\|N\| - \ell(n) - 2}$, choose a random string $r \leftarrow \{0,1\}^{\ell(n)}$ and interpret $\hat{m} := 1 \| r \| m$ as an element of $\mathbb{Z}_N^*$. Output the ciphertext

$$c := [\hat{m}^e \bmod N].$$

- Dec: on input a private key $sk = \langle N, d \rangle$ and a ciphertext $c \in \mathbb{Z}_N^*$, compute

$$\hat{m} := [c^d \bmod N],$$

and output the $\|N\| - \ell(n) - 2$ least-significant bits of $\hat{m}$.

The padded RSA encryption scheme.

Digital Signatures Definition

A digital signature scheme consists of three ppt algorithms $(Gen, Sign, Vrfy)$ such that:

1. The key-generation algorithm Gen takes as input a security parameter 1^n and outputs a pair of keys (pk, sk). We assume that pk, sk each have length at least n, and that n can be determined from pk or sk.
2. The signing algorithm $Sign$ 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 $\sigma \leftarrow Sign_{sk}(m)$.
3. The deterministic verification algorithm $Vrfy$ 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 $b \coloneqq Vrfy_{pk}(m, \sigma)$.

Correctness: It is required that except with negligible probability over (pk, sk) output by $Gen(1^n)$, it holds that $Vrfy_{pk}\big(m, Sign_{sk}(m)\big) = 1$ for every message m.

Digital Signatures Definition: Security

Experiment $SigForge_{A,\Pi}(n)$:

1. $Gen(1^n)$ is run to obtain keys (pk, sk).
2. Adversary A is given pk and access to an oracle $Sign_{sk}(\cdot)$. The adversary then outputs (m, σ). Let Q denote the set of all queries that A asked to its oracle.
3. A succeeds if and only if
   1. $Vrfy_{pk}(m, \sigma) = 1$
   2. $m \notin Q$.

In this case the output of the experiment is defined to be 1.

Definition: A signature scheme $\Pi = (Gen, Sign, Vrfy)$ is existentially unforgeable under an adaptive chosen-message attack, if for all ppt adversaries A, there is a negligible function neg such that:

$$\Pr\left[SigForge_{A,Pi}(n) = 1\right] \leq neg(n).$$

RSA Signatures

CONSTRUCTION 12.5

Let GenRSA be as in the text. Define a signature scheme as follows:

- Gen: on input 1^n run GenRSA(1^n) to obtain (N, e, d). The public key is $\langle N, e \rangle$ and the private key is $\langle N, d \rangle$.
- Sign: on input a private key $sk = \langle N, d \rangle$ and a message $m \in \mathbb{Z}_N^*$, compute the signature

$$\sigma := [m^d \bmod N].$$

- 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

$$m \stackrel{?}{=} [\sigma^e \bmod N].$$

The plain RSA signature scheme.

Attacks

No message attack:

Choose $s \in Z_N^*$, compute s^e.

Ouput $(m = s^e, \sigma = s)$ as the forgery.

Attacks

Forging a signature on an arbitrary message:

To forge a signature on message m, choose arbitrary $m_1, m_2 \neq 1$ such that $m = m_1 \cdot m_2$.
Query oracle for $(m_1, \sigma_1), (m_2, \sigma_2)$.
Output (m, σ), where $\sigma = \sigma_1 \cdot \sigma_2$.

RSA-FDH

CONSTRUCTION 12.6

Let GenRSA be as in the previous sections, and construct a signature scheme as follows:

- Gen: on input 1^n, run GenRSA(1^n) to compute (N, e, d). The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$.

  As part of key generation, a function $H : \{0,1\}^* \to \mathbb{Z}_N^*$ is specified, but we leave this implicit.
- Sign: on input a private key $\langle N, d\rangle$ and a message $m \in \{0,1\}^*$, compute
$$\sigma := [H(m)^d \bmod N].$$
- Vrfy: on input a public key $\langle N, e\rangle$, a message m, and a signature σ, output 1 if and only if $\sigma^e \stackrel{?}{=} H(m) \bmod N$.

The RSA-FDH signature scheme.

Random Oracles

- Assume certain hash functions behave exactly like a random oracle.
- The "oracle" is a box that takes a binary string as input and returns a binary string as output.
- The internal workings of the box are unknown.
- All parties (honest parties and adversary) have access to the box.
- The box is consistent.
- Oracle implements a random function by choosing values of $H(x)$ "on the fly."

Principles of RO Model

1. If x has not been queried to H, then the value of $H(x)$ is uniform.
2. If A queries x to H, the reduction can see this query and learn x.
3. The reduction can set the value of $H(x)$ to a value of its choice, as long as this value is correctly distributed, i.e., uniform.

Security of RSA-FDH

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

PKCS #1 v2.1

- Uses an instantiation of RSA-FDH for signing.
- SHA-1 should not be used "off-the-shelf" as an instantiation of H because output length is too small and so practical short-message attacks apply.
- In PKCS #1 v2.1, H is constructed via repeated application of an underlying cryptographic hash function.