Introduction to Cryptology

Lecture 23

Announcements

- Optional HW11 due next class (5/7)
- Course Evaluations at the end of next class (5/7)
  - Please bring laptop or mobile device to next class
- Stay tuned for survey about review session for final exam.

Agenda

- Last time:
  - Diffie-Hellman Key Exchange (10.3)
  - Public Key Encryption Definitions (11.2)
  - El Gamal Encryption (11.4)

- This time:
  - RSA Encryption and Weaknesses (11.5)
  - Digital Signatures (12.2-12.3)

RSA Encryption

CONSTRUCTION 11.25

Let GenRSA be as in the text. Define a public-key encryption scheme as follows:

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

The plain RSA encryption scheme.

RSA Example

$$p = 3, q = 7, N = 21$$
$$\phi(N) = 12$$
$$e = 5$$
$$d = 5$$
$$Enc_{(21,5)}(11) = 4^5 \; mod \; 21 = 16 \; mod \; 21$$
$$Dec_{21,5}(16) = 16^5 \; mod \; 21 = 4^5 \cdot 4^5 mod \; 21$$
$$= 16 \cdot 16 \; mod \; 21 = 4$$

Is Plain-RSA Secure?

- It is deterministic so cannot be secure!

Additional Attacks

Additional Attacks

Encrypting short messages using small e:

- When $m < N^{1/e}$, raising m to the e-th power modulo N involves no modular reduction.
- Can compute $m = c^{1/e}$ over the integers.

Additional Attacks

Encrypting a partially known message:

Coppersmith's Theorem: Let $p(x)$ be a polynomial of degree e. Then in time $poly(\log(N), e)$ one can find all m such that $p(m) = 0 \ mod \ N$ and $m \leq N^{1/e}$.

In the following, we assume $e = 3$.
Assume message is $m = m_1 || m_2$, where m_1 is known, but not m_2.
So $m = 2^k \cdot m_1 + m_2$.
Define $p(x) \coloneqq \left(2^k \cdot m_1 + x\right)^3 - c$.
This polynomial has m_2 as a root and $m \leq 2^k \leq N^{1/3}$.

Additional Attacks

Encrypting related messages:

Assume the sender encrypts both m and $m + \delta$, giving two ciphertexts c_1 and c_2.

Define $f_1(x) \coloneqq x^e - c_1$ and $f_2(x) \coloneqq (x + \delta)^e - c_2$.

$x = m$ is a root of both polynomials.

$(x - m)$ is a factor of both.

Use algorithm for finding gcd of polynomials.

Additional Attacks

Sending the same message to multiple receivers:
$pk_1 = \langle N_1, 3\rangle, pk_2 = \langle N_2, 3\rangle, pk_3 = \langle N_3, 3\rangle$.
Eavesdropper sees:
$c_1 = m^3 \; mod \; N_1, c_2 = m^3 \; mod \; N_2, c_3 = m^3 \; mod \; N_3$
Let $N^* = N_1 \cdot N_2 \cdot N_3$.
Using Chinese remainder theorem to find $\hat{c} < N^*$ such that:

$$\hat{c} = c_1 \; mod \; N_1$$
$$\hat{c} = c_2 \; mod \; N_2$$
$$\hat{c} = c_3 \; mod \; N_3.$$

Note that m^3 satisfies all three equations. Moreover, $m^3 < N^*$. Thus, we can solve for $m^3 = \hat{c}$ over the integers.

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.

PKCS #1 v1.5

- Issued by RSA Labs in 1993
- Let k denote the length of N in bytes.
- Messages m to be encrypted are assumed to be a multiple of 8 bits long and can have length anywhere from 1 to $k - 11$ bytes.
- Encryption of a message m that is D-bytes long is computed as:

$$\left[\left(0 \times 00 \middle|\middle| 0 \times 02 \middle|\middle| r \middle|\middle| 0 \times 00 \middle|\middle| m\right)^e \, mod \; N\right]$$

Where r is a randomly generated $(k - D - 3)$-byte string with none of its bytes equal to 0×00.

Insecurity of PKCS #1 v1.5

- The random padding is too short.
- Attack:
  - Set $m = b||0 \ldots 0$ (with L 0's), $b \in \{0,1\}$
  - Encryption gives a ciphertext c with $c = (0 \times 00||0 \times 02||r||0 \times 00||b||0 \cdots 0)^e \ mod \ N$
  - Compute $c' = \frac{c}{(2^L)^e} mod \ N$
  - $c' = \left(0 \times 00\middle||0 \times 02\middle||r\middle||0 \times 00\middle|| b\right)^e \ mod \ N$
  - This is only 75 bits long so an attacker can apply the "short message attack."
- r should be of length at least $k/2$ for security.

Insecurity of PKCS #1 v1.5

- Due to a chosen-ciphertext attack, this version should not be used.
- Updated versions should be used instead.
- Now up to v2.2

Digital Signatures

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 $Vrfy_{pk}\left(m, Sign_{sk}(m)\right) = 1$ 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:

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