Cryptography

Lecture 23

Announcements

- HW8 due today
- HW9 now up on course webpage. Due 5/6.
- One additional “optional” homework due on 5/13.

Agenda

- Last time:
  - Elliptic Curve Groups
  - Key Exchange Definitions and Diffie-Hellman Key Exchange (10.3)
  - Public Key Encryption Definitions (11.2)
- This time:
  - El Gamal Encryption (11.4)
  - RSA Encryption and Weaknesses (11.5)
  - Class Exercise

Public Key Encryption

Definition: A public key encryption scheme is a triple of ppt algorithms (Gen, Enc, Dec) such that:

1. The key generation algorithm Gen takes as input the security parameter 1^n and outputs a pair of keys (pk, sk). We refer to the first of these as the public key and the second as the private key. We assume for convenience that pk and sk each has length at least n, and that n can be determined from pk, sk.
2. The encryption algorithm Enc takes as input a public key pk and a message m from some message space. It outputs a ciphertext c, and we write this as $c \leftarrow Enc_{pk}(m)$.
3. The deterministic decryption algorithm Dec takes as input a private key sk and a ciphertext c, and outputs a message m or a special symbol $\perp$ denoting failure. We write this as $m \coloneqq Dec_{sk}(c)$.

Correctness: It is required that, except possibly with negligible probability over (pk, sk) output by $Gen(1^n)$, we have $Dec_{sk}\left(Enc_{pk}(m)\right) = m$ for any legal message m.

CPA-Security

The CPA experiment ${PubK^{cpa}}_{A,\Pi}(n)$:

1. $Gen(1^n)$ is run to obtain keys (pk, sk).
2. Adversary A is given pk, and outputs a pair of equal-length messages m_0, m_1 in the message space.
3. A uniform bit $b \in \{0,1\}$ is chosen, and then a challenge ciphertext $c \leftarrow Enc_{pk}(m_b)$ is computed and given to A.
4. A outputs a bit b'. The output of the experiment is 1 if $b' = b$, and 0 otherwise.

Definition: A public-key encryption scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is CPA-secure if for all ppt adversaries A there is a negligible function neg such that

$$\Pr\left[{PubK^{cpa}}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + neg(n).$$

El Gamal Encryption

--Show how we can derive El Gamal PKE from Diffie-Hellman Key Exchange

Important Property

Lemma: Let G be a finite group, and let $m \in G$ be arbirary. Then choosing uniform $k \in G$ and setting $k' := k \cdot m$ gives the same distribution for k' as choosing uniform $k' \in G$. Put differently, for any $\hat{g} \in G$ we have

$$\Pr[k \cdot m = \hat{g}] = 1/|G|.$$

El Gamal Encryption Scheme

CONSTRUCTION 11.16

Let $\mathcal{G}$ be as in the text. Define a public-key encryption scheme as follows:

- Gen: on input 1^n run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. Then choose a uniform $x \leftarrow \mathbb{Z}_q$ and compute $h := g^x$. The public key is $\langle \mathbb{G}, q, g, h \rangle$ and the private key is $\langle \mathbb{G}, q, g, x \rangle$. The message space is $\mathbb{G}$.
- Enc: on input a public key $pk = \langle \mathbb{G}, q, g, h \rangle$ and a message $m \in \mathbb{G}$, choose a uniform $y \leftarrow \mathbb{Z}_q$ and output the ciphertext

$$\langle g^y,\ h^y \cdot m \rangle.$$

- Dec: on input a private key $sk = \langle \mathbb{G}, q, g, x \rangle$ and a ciphertext $\langle c_1, c_2 \rangle$, output

$$\hat{m} := c_2 / c_1^x.$$

The El Gamal encryption scheme.

El Gamal Example

Security Analysis

Theorem: If the DDH problem is hard relative to G, then the El Gamal encryption scheme is CPA-secure.

Textbook 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

$$p = 3, q = 7, N = 21$$

$$\phi(N) = 12$$

$$e = 5$$

$$d = 5$$

$$Enc_{(21,5)}(4) = 4^5 \; mod \; 21 = 16 \; mod \; 21$$

$$Dec_{21,5}(16) = 16^5 \; mod \; 21 = 4^5 \cdot 4^5 mod \; 21$$

$$= 16 \cdot 16 \; mod \; 21 = 4$$

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.