Introduction to Modern Cryptography
Recitation 6

Orit Moskovich
Tel Aviv University
December 7, 2016

Public Key Encryption

Definition. A **public key encryption (PKE)** scheme $\mathcal{E} = (Gen, Enc, Dec)$ consists of three algorithms:

1. The randomized key generation algorithm $Gen(r)$ that outputs a pair (pk, sk)
2. The encryption algorithm Enc takes as input a public key pk and a message m and outputs a ciphertext c. Denoted $Enc_{pk}(m) = c$
3. The decryption algorithm Dec takes as input a secret key sk and a ciphertext c and outputs a message m. Denoted $Dec_{sk}(c) = m$

CPA Security

- Define an experiment involving an adversary A
- A is trying to distinguish the encryption of one plaintext from the encryption of another

Adversarial indistinguishability experiment for A:

1. A random key (pk, sk) is generated using Gen
2. The adversary A is given pk and outputs a pair of messages m_0, m_1 of the same length
3. A random bit $b \leftarrow \{0,1\}$ is chosen
4. The ciphertext $c = Enc_{pk}(m_b)$ is computed and given to A
5. A outputs a bit b'

$$A \text{ wins} \Leftrightarrow b = b'$$

CPA Security

What if A is also given an oracle to $Enc_{pk}(\cdot)$

Adversarial indistinguishability experiment for A:

1. A random key (pk, sk) is generated using Gen
2. The adversary A is given pk and outputs a pair of messages m_0, m_1 of the same length
3. A random bit $b \leftarrow \{0,1\}$ is chosen
4. The ciphertext $c = Enc_{pk}(m_b)$ is computed and given to A
5. A outputs a bit b'

$$A \text{ wins} \Leftrightarrow b = b'$$

Definition. A PKE scheme (Gen, Enc, Dec) is **ε-CPA-secure** (chosen plaintext attack) if for every PPT adversary A it holds that $\Pr[A \text{ wins}] \leq \frac{1}{2} + \varepsilon$

Notes Regarding Security

- No deterministic PKE scheme is CPA secure

Semantic Security

- The encryption of any two messages must “look” the same

Definition. A PKE scheme (Gen, Enc, Dec) is
ε-semantically secure if for all messages m_0, m_1 such that $|m_0| = |m_1|$:

$$pk, Enc_{pk}(m_0, r) \approx_{c,\varepsilon} pk, Enc_{pk}(m_1, r)$$

Where $(sk, pk) \leftarrow Gen$ and $r \leftarrow U_n$

Notes Regarding Security

- No deterministic PKE scheme is CPA secure
- Perfectly secret PKE does not exist (HW)

Theorem. A PKE scheme (Gen, Enc, Dec) is ε-semantically secure $\Leftrightarrow$ it is ε-CPA secure

Security for Multiple Encryptions

Adversarial indistinguishability experiment for A_{mult}:

1. A random key (pk, sk) is generated using Gen
2. The adversary A_{mult} is given pk and outputs a pair of vectors $M_0 = (m_0^1, \dots, m_0^t)$ and $M_1 = (m_1^1, \dots, m_1^t)$, where $\forall i. \left|m_0^i\right| = \left|m_1^i\right|$
3. A random bit $b \leftarrow \{0,1\}$ is chosen
4. The vector $C = \left(Enc_{pk}(m_b^1), \dots, Enc_{pk}(m_b^t)\right)$ is given to A_{mult}
5. A_{mult} outputs a bit b'

$$A_{mult} \text{ wins} \Leftrightarrow b = b'$$

Definition. An encryption scheme is **ε-CPA-secure for multiple encryptions** if for every adversary A_{mult} it holds that $\Pr[A_{mult} \text{ wins}] \leq \frac{1}{2} + \varepsilon$

Notes Regarding Security

- No deterministic PKE scheme is CPA secure
- Perfectly secret PKE does not exist (HW)

Theorem. A PKE scheme (Gen, Enc, Dec) is
ε-semantically secure $\Leftrightarrow$ it is ε-CPA secure

- Security for a single message ➔ security for multiple messages (hybrid arguments)
- Given any CPA-secure PKE scheme for fixed length messages, we can construct a CPA-secure PKE scheme for arbitrary length messages

El Gamal Encryption Scheme

- A PKE scheme based on DDH assumption
- Motivation:

Claim. Let G be a finite group and let $m, g' \in G$ arbitrary elements. Then

$$\Pr_{g \leftarrow G}[m \cdot g = g'] = \frac{1}{|G|}$$

Construction. A perfectly secret private key scheme:

1. The messages are $m \in G$
2. Secret random key $g \leftarrow G$
3. $Enc_g(m) = m \cdot g$
4. $Dec_g(c) = c \cdot g^{-1}$

El Gamal Encryption Scheme

- A PKE scheme based on DDH assumption
- Motivation:

Construction. A perfectly secret private key scheme:

1. The messages are $m \in G$
2. Secret random key $g \leftarrow G$
3. $Enc_g(m) = m \cdot g$
4. $Dec_g(c) = c \cdot g^{-1}$

- El Gamal relies on choosing a pseudorandom g during encryption
- Decryption? We will construct g in such a way that it will be easy for the receiver (holding sk) to recover g

El Gamal Encryption Scheme

El Gamal PKE Scheme. Let G be a cyclic group of order q and a generator $g \in G$.

1. Key generation algorithm Gen chooses a random $x \in \mathbb{Z}_q$.
   $pk = (G, q, g, g^x), sk = (G, q, g, x)$
2. To encrypt a message $m \in G$ using $pk = (G, q, g, h)$:
   - Choose a random $y \leftarrow \mathbb{Z}_q$
   - Compute g^y
   - Compute $h^y \cdot m = (g^x)^y \cdot m = g^{xy} \cdot m$
   - Send $Enc_{pk}(m) = (g^y, h^y \cdot m)$
3. To decrypt a message (c_1, c_2) using $sk = (G, q, g, x)$:
   - Compute $c_1^x = (g^y)^x = g^{xy}$
   - Compute $c_2 \cdot (g^{xy})^{-1} = g^{xy} \cdot m \cdot (g^{xy})^{-1} = m$

El Gamal Encryption Scheme- The Short Version

El Gamal PKE Scheme. Let G be a cyclic group of order q and a generator $g \in G$.

1. Key generation algorithm Gen chooses a random $x \in \mathbb{Z}_q$.
   $pk = (G, q, g, g^x), sk = (G, q, g, x)$
2. To encrypt a message $m \in G$ using $pk = (G, q, g, h)$:
   - Choose a random $y \leftarrow \mathbb{Z}_q$
   - Compute g^y
   - Send $Enc_{pk}(m) = (g^y, h^y \cdot m)$
3. To decrypt a message (c_1, c_2) using $sk = (G, q, g, x)$:
   - Compute $Dec_{sk}\big((c_1, c_2)\big) = c_2 \cdot (c_1^x)^{-1}$

El Gamal Encryption Scheme

Theorem. If the DDH problem is hard in G, then El Gamal PKE scheme is ε-semantically secure

(Reminder) Definition. Let G be a cyclic group of order $|G| = m$ and a generator $g \in G$. Define

- $D_0 = \{g^x, g^y, g^{xy} \mid (x, y) \leftarrow \mathbb{Z}_m \times \mathbb{Z}_m\}$
- $D_1 = \{g^x, g^y, g^z \mid (x, y, z) \leftarrow \mathbb{Z}_m \times \mathbb{Z}_m \times \mathbb{Z}_m\}$

Then, we say that **The DDH problem is hard in G** $\Leftrightarrow D_0 \approx_{c,\varepsilon} D_1$

El Gamal Encryption Scheme

Theorem. If the DDH problem is hard in G, then El Gamal PKE scheme is ε-semantically secure

- Let $m_0, m_1 \in G$
- $D_0 = \{g, g^x, g^y, g^{xy} \cdot m_0 | (x,y) \leftarrow \mathbb{Z}_{|G|} \times \mathbb{Z}_{|G|}\} \approx_{c,\varepsilon}$
  $\{g, g^x, g^y, g^z \cdot m_0 | (x,y,z) \leftarrow \mathbb{Z}_{|G|} \times \mathbb{Z}_{|G|} \times \mathbb{Z}_{|G|}\} \equiv$
  $\{g, g^x, g^y, g^{z'} | (x,y,z') \leftarrow \mathbb{Z}_{|G|} \times \mathbb{Z}_{|G|} \times \mathbb{Z}_{|G|}\}$
- $D_1 = \{g, g^x, g^y, g^{xy} \cdot m_1 | (x,y) \leftarrow \mathbb{Z}_{|G|} \times \mathbb{Z}_{|G|}\}$

Example – Coin Flip Using El Gamal

- Define the following protocol between Alice and Bob:
- Alice and Bob wishes to generate a joint, unbiased bit $b \in \{0,1\}$ that they both agree on
  (recall coin flip over the phone)
- Assume Benny is a trusted third party

Example – Coin Flip Using El Gamal

1. Publish a public key $pk = g, g^x$

Alice

Bob

Example – Coin Flip Using El Gamal

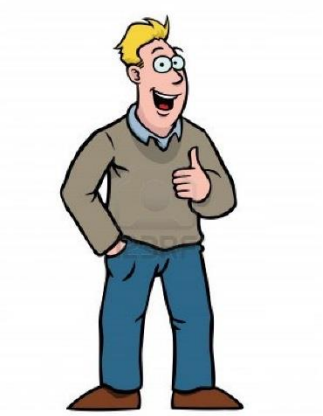
1. Publish a public key $pk = g, g^x$

2. Alice chooses a random bit b_A
and sends $c_A = Enc_{pk}(b_A) = g^y, g^{xy} \cdot g^{b_A}$

Alice

Bob

Example – Coin Flip Using El Gamal



1. Publish a public key $pk = g, g^x$

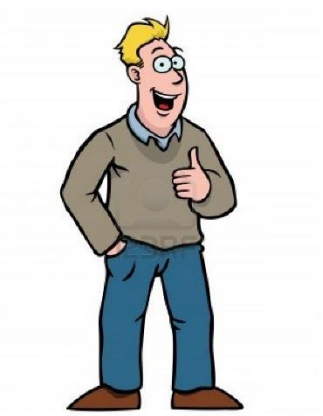
2. Alice chooses a random bit b_A
and sends $c_A = Enc_{pk}(b_A) = g^y, g^{xy} \cdot g^{b_A}$

3. Bob chooses a random bit b_B
and sends $c_B = Enc_{pk}(b_B) = g^{y'}, g^{xy'} \cdot g^{b_B}$

Alice

Bob

Example – Coin Flip Using El Gamal



1. Publish a public key $pk = g, g^x$

2. Alice chooses a random bit b_A
and sends $c_A = Enc_{pk}(b_A) = g^y, g^{xy} \cdot g^{b_A}$

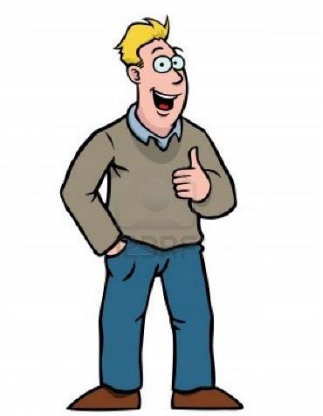
3. Bob chooses a random bit b_B
and sends $c_B = Enc_{pk}(b_B) = g^{y'}, g^{xy'} \cdot g^{b_B}$

Alice

$Dec_{sk}(c_A) = b_A,$
$Dec_{sk}(c_B) = b_B$

Bob

Example – Coin Flip Using El Gamal



1. Publish a public key $pk = g, g^x$

2. Alice chooses a random bit b_A
and sends $c_A = Enc_{pk}(b_A) = g^y, g^{xy} \cdot g^{b_A}$

3. Bob chooses a random bit b_B
and sends $c_B = Enc_{pk}(b_B) = g^{y'}, g^{xy'} \cdot g^{b_B}$

Alice

Bob

$Dec_{sk}(c_A) = b_A,$
$Dec_{sk}(c_B) = b_B$

The joint bit is $b_A \oplus b_B$

Example – Coin Flip Using El Gamal

What if…?

1. Publish a public key $pk = g, g^x$

2. Alice chooses a random bit b_A
and sends $c_A = Enc_{pk}(b_A) = g^y, g^{xy} \cdot g^{b_A}$

Alice

$Dec_{sk}(c_A) = b_A,$
$Dec_{sk}(c_B) = b_B$

3. Bob chooses a random bit b_B
and sends $c_B = Enc_{pk}(b_B) = g^{y'}, g^{xy'} \cdot g^{b_B}$
$g^y g^r, g^{xy} \cdot g^{b_A} g^{xr}$

Bob

The joint bit is $b_A \oplus b_B$