INTRODUCTION TO SECURITY REDUCTION II: PUBLIC KEY ENCRYPTION

SHUANGJUN ZHANG

Content

1. *Number Theory and Cryptographic Hardness Assumptions*
2. *Indistinguishable Encryption in Public-Key Model*
3. *ElGamal Encryption*
4. *Hashed ElGamal*

Content

1. *Number Theory and Cryptographic Hardness Assumptions*
2. *Indistinguishable Encryption in Public-Key Model*
3. *ElGamal Encryption*
4. *Hashed ElGamal*

1. Number Theory and Cryptographic Hardness Assumptions

- Modern cryptosystems are invariably based on an assumption that some problem is hard.
- For example, in our first slide we construct some indistinguishable encryptions by PRG and PRF. However, the assumption that pseudorandom permutations exist seems quite strong and unnatural. It can be proved that PRG/PRF exist based on the much milder assumption that one-way functions exist.
- One goal of this chapter is to introduce various problems believed to be “hard”, and to present conjectured one-way functions based on those problems.

1. Number Theory and Cryptographic Hardness Assumptions

- 1.1 The Factoring Assumption
  - *Let* $GenModulus$ *be a polynomial-time algorithm that, on input* 1^n, *outputs*(N, p, q) *where* $N\ = pq$, *and* p *and* q *are* n*-bit primes.*
  - *Consider the following game played between an adversary and challenger:*
    - 1. The challenger runs $GenModulus(1^n)$ to obtain (N, p, q).
    - 2. The adversary A is given N, and outputs $p', q' > 1$.
  - *We define* S *to be the event that* $p' \cdot q' = N$
  - *Factoring is hard relative to* $GenModulus$ *if for all probabilistic polynomial-time adversary* A *there exists a negligible function* $negl$ *such that*

$$\Pr[S] \le negl(n)$$

1. Number Theory and Cryptographic Hardness Assumptions

- 1.2 The Discrete Logarithm Assumption
  - *Let G be a group-generation algorithm that, on input 1^n, outputs$(\mathbb{G}, g, q)$, where $\mathbb{G}$ is a cyclic group of order q (with $\| q \| = n$), and g is a generator of $\mathbb{G}$.*
  - *Consider the following game played between an adversary and challenger:*
    - 1. The challenger runs $G(1^n)$ to obtain $(\mathbb{G}, q, g)$ and choose a uniform $h \in \mathbb{G}$.
    - 2. The adversary A is given $\mathbb{G}, q, g, h$, and output $x \in \mathbb{Z}_q$.
  - *We define S to be the event that $g^x = h$.*
  - *The Discrete Logarithm Problem is hard relative to G if for all probabilistic polynomial-time adversary A there exists a negligible function $negl$ such that*

$$\Pr[S] \leq negl(n)$$

1. Number Theory and Cryptographic Hardness Assumptions

- Two important variants: the computational Diffie–Hellman (CDH) problem and the decisional Diffie–Hellman (DDH) problem.
- The computational Diffie–Hellman (CDH) Assumption
  - *Let G be a group-generation algorithm that, on input 1^n, outputs$(\mathbb{G}, g, q)$, where $\mathbb{G}$ is a cyclic group of order q (with $\| q \| = n$), and g is a generator of $\mathbb{G}$.*
  - *Consider the following game played between an adversary and challenger:*
    - 1. The challenger runs $G(1^n)$ to obtain $(\mathbb{G}, q, g)$ and choose uniform $h_1, h_2 \in \mathbb{G}$. Let $x_1 = log_g h_1$, $x_2 = \log_g h_2$
    - 2. The adversary A is given $\mathbb{G}, q, g, h_1, h_2$, and output $h_3 \in Z_q$
  - *We define S to be the event that $h_3 = g^{x_1 \cdot x_2}$.*
  - *The computational Diffie–Hellman (CDH) Problem is hard relative to G if for all probabilistic polynomial-time adversary A there exists a negligible function $negl$ such that*

$$\Pr[S] \leq negl(n)$$

1. Number Theory and Cryptographic Hardness Assumptions

- The decisional Diffie–Hellman (DDH) Assumption
  - *Let G be a group-generation algorithm that, on input 1^n, outputs$(\mathbb{G}, g, q)$, where $\mathbb{G}$ is a cyclic group of order q (with $\| q \| = n$), and g is a generator of $\mathbb{G}$.*
  - *Consider the following game played between an adversary and challenger:*
    - 1. The challenger runs $G(1^n)$ to obtain $(\mathbb{G}, q, g)$ and choose uniform $h_1, h_2 \in \mathbb{G}$. Let $x_1 = log_g h_1$, $x_2 = \log_g h_2$. A random bit $b \in \{0,1\}$ is chosen by the challenger. Choose uniform $h_3 \in \mathbb{G}$ if $b = 0$, set $h_3 = g^{x_1 \cdot x_2}$ if $b = 1$.
    - 2. The adversary A is given $\mathbb{G}, q, g, h_1, h_2, h_3$, and output a bit b'
  - *We define S to be the event that $b' = b$.*
  - *The decisional Diffie–Hellman (DDH) Problem is hard relative to G if for all probabilistic polynomial-time adversary A there exists a negligible function $negl$ such that*

$$\left|\Pr[S] - \frac{1}{2}\right| \leq negl(n)$$

1. Number Theory and Cryptographic Hardness Assumptions

- *The* decisional *Diffie–Hellman (DDH) Problem is hard relative to G if for all probabilistic polynomial-time adversary A there exists a negligible function $negl$ such that*

$$|\Pr[A(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - \Pr[A(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]| \leq negl(n)$$

$where\ x, y, z\ is\ random\ chosen\ from\ \mathbb{Z}_q$

- *The hardness of the CDH problem relative to G implies hardness of the DLP relative to G, and that hardness of the DDH problem relative to G implies hardness of the CDH problem relative to G.*

Content

1. *Number Theory and Cryptographic Hardness Assumptions*
2. *Indistinguishable Encryption in Public-Key Model*
3. *ElGamal Encryption*
4. *Hashed ElGamal*

2. Indistinguishable Encryption in Public-Key Model

- Indistinguishable Encryption in Public-Key Model
- Consider the following game played between an adversary and challenger:
  - *1. The challenger B runs $Gen(1^n)$ and gets (pk, sk).*
  - *2. The adversary A is given input 1^n, and the public key pk, and outputs a pair of messages m_0, m_1 with $|m_0| = |m_1|$.*
  - *3. The challenger B flips a random coin $b \in \{0,1\}^*$ and computes ciphertext $c \leftarrow Enc_k(m_b)$, gives the ciphertext to A. We refer to c as the challenge ciphertext.*
  - *4. The adversary A outputs a guess b' of b.*

  The advantage of an adversary A in this game is defined as $|\Pr[b = b'] - \frac{1}{2}|$.

  Theorem 2.1 A public encryption scheme is semantic security if and only if all PPT adversaries have at most a negligible advantage in the above game.
- If a public-key encryption scheme is an indistinguishable encryption, then it is CPA-secure.

Content

1. *Number Theory and Cryptographic Hardness Assumptions*
2. *Indistinguishable Encryption in Public-Key Model*
3. *ElGamal Encryption*
4. *Hashed ElGamal*

3.Elgamal Encryption

- Construction 3.1
- Define a public-key encryption scheme as follows :

Let G be a group of prime order q, and let $g \in G$ be a generator.

- *1. $Gen(1^n)$: Choose uniform $x \in \mathbb{Z}_q$, set $\alpha = g^x$, output (sk, pk), where pk is α, sk is x.*
- *2. $Enc(m, pk)$:To encryption a message $m \in G$ using the pk, the algorithm first choose uniform $y \in \mathbb{Z}_q$, output the ciphertext $c =< c_1, c_2 >$, where*

$$c_1 = g^y, c_2 = m \cdot \alpha^y$$

- *3. $Dec(c, sk)$: on input a key sk and a ciphertext $c =< c_1, c_2 >$, output the message*

$$m = c_2 / c_1^x$$

Correctness is easy to see

$$m = \frac{c_2}{c_1^x} = \frac{m \cdot \alpha^y}{(g^y)^x} = \frac{m g^{xy}}{g^{xy}} = m$$

- Theorem 3.2 If the DDH problem is hard relative to G, then the El Gamal encryption scheme is CPA-secure.

3.Elgamal Encryption

- Proof of theorem 3.2
  - *Game 0. This is the original attack game. We may describe the attack game algorithmically as follows:*
    - $x \leftarrow \mathbb{Z}_q, \alpha \leftarrow g^x$
    - $(m_0, m_1) \leftarrow A(\alpha)$
    - $b \leftarrow \{0,1\}, y \leftarrow \mathbb{Z}_q, c_1 \leftarrow g^y, \beta \leftarrow \alpha^y, c_2 \leftarrow m_b \cdot \beta$
    - $b' = A(\alpha, c_1, c_2)$

  If we define S_0 to be the event that $b = b$' in Game 0, then the adversary's advantage is $\left|\Pr[S_0] - \frac{1}{2}\right|$.
  - *Game 1. We now make one small change to the above game. Namely, instead of computing $\beta \leftarrow \alpha^y$, we compute it as g^z for randomly chosen $z \in \mathbb{Z}_q$.*
    - $x \leftarrow \mathbb{Z}_q, \alpha \leftarrow g^x$
    - $(m_0, m_1) \leftarrow A(\alpha)$
    - $b \leftarrow \{0,1\}, y \leftarrow \mathbb{Z}_q, c_1 \leftarrow g^y, z \leftarrow \mathbb{Z}_q, \beta \leftarrow g^z, c_2 \leftarrow m_b \cdot \beta$
    - $b' = A(\alpha, c_1, c_2)$

  If we define S_1 to be the event that $b = b$' in Game 1.

3.Elgamal Encryption

- Claim 1. $\Pr[S_1] = \frac{1}{2}$
- Proof of Claim 1.
  - *Since z is random number in $\mathbb{Z}_q$, $\beta = g^z$ is a random element in G.*
  - *Fixing a group element $g' \in G$, the probability of c_2 equals to g' is*

$$\Pr[c_2 = g'] = \Pr[m \cdot \beta = g'] = \Pr[\beta = m^{-1} \cdot g'] = \frac{1}{|G|}.$$

  - *It follow that c_2 is a random element in G from A’s view. A can’t get any information about b, so $\Pr[S_1] = \frac{1}{2}$*

3.Elgamal Encryption

- Claim 2. $|\Pr[S_0] - \Pr[S_1]| \; = \; \epsilon_{ddh}$ where ϵ_{ddh} is the DDH-advantage of some efficient algorithm. (which is negligible under the DDH assumption)
- Proof of Claim 2.
  - *Consider a distinguishing algorithm D works as follows.*
  - *Algorithm* $D(g^x, g^y, \beta)$// x, y, z *is chosen at random*
    - $\alpha \leftarrow g^x, c_1 \leftarrow g^y$
    - $(m_0, m_1\ ) \leftarrow A(\alpha)$
    - $b \leftarrow \{0,1\};\ c_2 = m_b \cdot \beta$
    - $b' \leftarrow A(\alpha, c_1, c_2)$
    - If $b \; = \; b$' output 1, else output 0
  - *If* $\beta = g^{xy}$*, then the adversary A is in Game 0, and therefore* $\Pr[D(g^x, g^y, g^{xy}) = 1] = \Pr[S_0]$
  - *If* $\beta = g^z$*, where* z *is chosen at random, then the adversary A is in Game 1, and therefore* $\Pr[D(g^x, g^y, g^z) = 1] = \Pr[S_1]$
  - *So, we get* $|\Pr[S_0] - \Pr[S_1]| = |\Pr[D(g^x, g^y, g^{xy}) = 1] - \Pr[D(g^x, g^y, g^z) = 1]| = \epsilon_{ddh}$

3.Elgamal Encryption

- *Combining Claim 1 and Claim 2, we see that*

$$\left|\Pr[S_0] - \frac{1}{2}\right| = \epsilon_{ddh}$$

- *Since ϵ_{ddh} is negligible, we get the advantage of A in game 0 is negligible.*

Content

1. *Number Theory and Cryptographic Hardness Assumptions*
2. *Indistinguishable Encryption in Public-Key Model*
3. *ElGamal Encryption*
4. *Hashed ElGamal*

4. Hashed ELGamal

- For a number of reasons, it is convenient to work with messages that are bit strings, say, of length l, rather than group elements. Because of this, one may choose to use a “hashed” version of the ElGamal encryption scheme.
- This scheme makes use of a family of keyed “hash” functions $H := \{H_k\}_{k \in K}$, where each H_k is a function mapping G to $\{0, 1\}^l$

4. Hashed ELGamal

- Construction 4.1
- Define the Hashed ELGamal scheme as follows :

Let G be a group of prime order q, and let $g \in G$ be a generator.

- *1. $Gen(1^n)$: Choose uniform $x \in \mathbb{Z}_q$, $k \in K$, set $\alpha = g^x$, output (sk, pk), where pk is α, k, sk is x.*
- *2. $Enc(m, pk)$:To encryption a message $m \in \{0,1\}^l$ using the pk, the algorithm first choose uniform $y \in \mathbb{Z}_q$, computer $h = H_k(\alpha^y)$, output the ciphertext $c =< c_1, c_2 >$, where*

$$c_1 = g^y, c_2 = m \oplus h$$

- *3. $Dec(c, sk)$: on input a key sk and a ciphertext $c =< c_1, c_2 >$, output the message*

$$m = H_k(c_1^x) \oplus c_2$$

Correctness is easy to see

$$m = H_k(c_1^x) \oplus c_2 = H_k(g^{xy}) \oplus m \oplus H_k(\alpha^y) = m$$

- Theorem 4.2 If the DDH problem is hard relative to G and the family of hash functions H is "entropy smoothing." then the El Gamal encryption scheme is CPA-secure.

4. Hashed ELGamal

- “entropy smoothing” hash functions
- Loosely speaking, this means that it is hard to distinguish $(k, H_k(s))$ from (k, r), where k is a random element of K, s is a random element of G, and r is a random element of $\{0, 1\}^l$
- The hash functions $H := \{H_k\}_{k \in K}$ is “entropy smoothing” if for all probabilistic polynomial-time adversary A there exists a negligible function $negl$ such that
$$|\Pr\left[A\left(k, H_k(s)\right) = 1\right] - \Pr[A(k, r) = 1]| \leq negl(n)$$

4. Hashed ELGamal

- Proof of theorem 4.2
  - *Game 0. This is the original attack game. We may describe the attack game algorithmically as follows:*
    - $x \leftarrow \mathbb{Z}_q, \alpha \leftarrow g^x, k \leftarrow K$
    - $(m_0, m_1) \leftarrow A(\alpha, k)$
    - $b \leftarrow \{0,1\}, y \leftarrow \mathbb{Z}_q, c_1 \leftarrow g^y, \beta \leftarrow \alpha^y, h = H_k(\beta), c_2 \leftarrow m_b \oplus h$
    - $b' = A(\alpha, k, c_1, c_2)$

If we define S_0 to be the event that $b = b$' in Game 0, then the adversary's advantage is $\left|\Pr[S_0] - \frac{1}{2}\right|$.

  - *Game 1. We now make one small change to the above game. Namely, instead of computing $\beta \leftarrow \alpha^y$, we compute it as g^z for randomly chosen $z \in \mathbb{Z}_q$.*
    - $x \leftarrow \mathbb{Z}_q, \alpha \leftarrow g^x, k \leftarrow K$
    - $(m_0, m_1) \leftarrow A(\alpha, k)$
    - $b \leftarrow \{0,1\}, y \leftarrow \mathbb{Z}_q, c_1 \leftarrow g^y, z \leftarrow \mathbb{Z}_q, \beta \leftarrow g^z, h = H_k(\beta), c_2 \leftarrow m_b \oplus h$
    - $b' = A(\alpha, k, c_1, c_2)$

If we define S_1 to be the event that $b = b$' in Game 1.

4. Hashed ELGamal

- Claim 1. $|\Pr[S_0] - \Pr[S_1]| = \epsilon_{ddh}$ where ϵ_{ddh} is the DDH-advantage of some efficient algorithm. (which is negligible under the DDH assumption)
- The proof is almost identical to the proof of the corresponding claim for "plain" ElGamal.
  - *Consider a distinguishing algorithm D works as follows.*
  - *Algorithm $D(g^x, g^y, \beta)$// x, y, z is chosen at random; $\beta = g^{xy}$ or g^z*
    - $k \leftarrow K, \alpha \leftarrow g^x, c_1 \leftarrow g^y$
    - $(m_0, m_1) \leftarrow A(k, \alpha)$
    - $b \leftarrow \{0,1\}, h = H_k(\beta), c_2 = m_b \oplus h$
    - $b' \leftarrow A(k, \alpha, c_1, c_2)$
    - If $b = b'$ output 1, else output 0
  - *If $\beta = g^{xy}$, then the adversary A is in Game 0, and therefore $\Pr[D(g^x, g^y, g^{xy}) = 1] = \Pr[S_0]$*
  - *If $\beta = g^z$, where z is chosen at random, then the adversary A is in Game 1, and therefore $\Pr[D(g^x, g^y, g^z) = 1] = \Pr[S_1]$*
  - *So, we get $|\Pr[S_0] - \Pr[S_1]| = |\Pr[D(g^x, g^y, g^{xy}) = 1] - \Pr[D(g^x, g^y, g^z) = 1]| = \epsilon_{ddh}$*

4. Hashed ELGamal

- *Game 2. We now make one small change to the game 1. Namely, instead of computing a hash function, we using a random string r.*
  - $x \leftarrow \mathbb{Z}_q, \alpha \leftarrow g^x, k \leftarrow K$
  - $(m_0, m_1) \leftarrow A(\alpha, k)$
  - $b \leftarrow \{0,1\}, y \leftarrow \mathbb{Z}_q, c_1 \leftarrow g^y, z \leftarrow \mathbb{Z}_q, \beta \leftarrow g^z, r \leftarrow \{0,1\}^l, c_2 \leftarrow m_b \oplus r$
  - $b' = A(\alpha, k, c_1, c_2)$

  If we define S_2 to be the event that $b = b$' in Game 2.

- Claim 2: $|\Pr[S_1] - \Pr[S_2]| = \epsilon_{es}$ where ϵ_{es} is the ES-advantage of some efficient algorithm. (which is negligible assuming H is entropy smoothing)

4. Hashed ELGamal

- Proof of Claim 2:
  - *Consider a distinguishing algorithm D' works as follows.*
  - *Algorithm* $D'(k,\delta)$// *k is chosen at random, δ is a hash value or random string*
    - $x \leftarrow \mathbb{Z}_q, \alpha = g^x$
    - $(m_0, m_1\ ) \leftarrow A(k,\alpha)$
    - $b \leftarrow \{0,1\}, y \leftarrow \mathbb{Z}_q, c_1 \leftarrow g^y, c_2 = m_b \oplus \delta$
    - $b' \leftarrow A(k,\alpha,c_1,c_2)$
    - If $b\ =\ b$' output 1, else output 0
  - *If $\delta = H_k(s)$ for some random element in G, then the adversary A is in Game 1, and therefore $\Pr[D'(k,H_k(s)) = 1] = \Pr[S_1]$*
  - *If $\delta = r$ where r is a random string, then the adversary A is in Game 2, and therefore $\Pr[D'(k,r) = 1] = \Pr[S_2]$*
  - *So, we get* $|\Pr[S_1] - \Pr[S_2]| = |\Pr[D'(k,H_k(s)) = 1] - \Pr[D'(k,r) = 1]| = \epsilon_{es}$

4. Hashed ELGamal

- Claim 3: $\Pr[S_2] = \frac{1}{2}$ (Since c_2 is a random uniform string in $\{0,1\}^l$)
- Combining Claim (1), (2), (3), we obtain
  - $\left|\Pr[S_0] - \frac{1}{2}\right| = |\Pr[S_0] - \Pr[S_2]|$

    $= |\Pr[S_0] - \Pr[S_1] + \Pr[S_1] - \Pr[S_2]|$

    $\leq |\Pr[S_0] - \Pr[S_1]| + |\Pr[S_1] - \Pr[S_2]|$

    $= \epsilon_{ddh} + \epsilon_{dh}$
- Since ϵ_{ddh}+ ϵ_{dh} is negligible, we get the advantage of A in game 0 is negligible.

5.