

---

## University of Michigan–Ann Arbor

Department of Electrical Engineering and Computer Science

EECS 475 **Introduction to Cryptography**, Winter 2023

### Lecture 22: CPA security continued, El Gamal cryptosystem

March 29, 2023

Lecturer: Mahdi Cheraghchi

Scribe: Yi-Wen Tseng

---

## 1 CPA Security

In continuation of the previous class, we want to show that one-query CPA implies many-query CPA.

Imagine a many-query attacker  $A$  that makes up to  $q$  queries where  $q \in \text{poly}(n)$ . Consider the following worlds:

**Hybrid 0 (Left World)** : all queries  $(m_0, m_1)$  to the LR oracle are answered by  $c \leftarrow \text{Enc}_{pk}(m_0)$ .

**Hybrid 1** : First query  $(m_0, m_1)$  to the LR oracle is answered by  $c \leftarrow \text{Enc}_{pk}(m_1)$ , then  $c \leftarrow \text{Enc}_{pk}(m_0)$  thereafter.

**Hybrid 2** : First 2 queries  $(m_0, m_1)$  to the LR oracle are answered by  $c \leftarrow \text{Enc}_{pk}(m_1)$ , then  $c \leftarrow \text{Enc}_{pk}(m_0)$  thereafter.

$\vdots$

**Hybrid  $q$  (Right World)** : all queries  $(m_0, m_1)$  to the LR oracle are answered by  $c \leftarrow \text{Enc}_{pk}(m_1)$ .

Note here, the only difference between  $\text{Hybrid}(i - 1)$  and  $\text{Hybrid}(i)$  is how the  $i^{\text{th}}$  query is answered.

Now, we build a "simulator"  $S_i^{LR_{pk,b}(\cdot, \cdot)}(pk)$  that gets **one query** and simulates either  $\text{Hybrid}(i - 1)$  or  $\text{Hybrid}(i)$  depending on  $b$ .

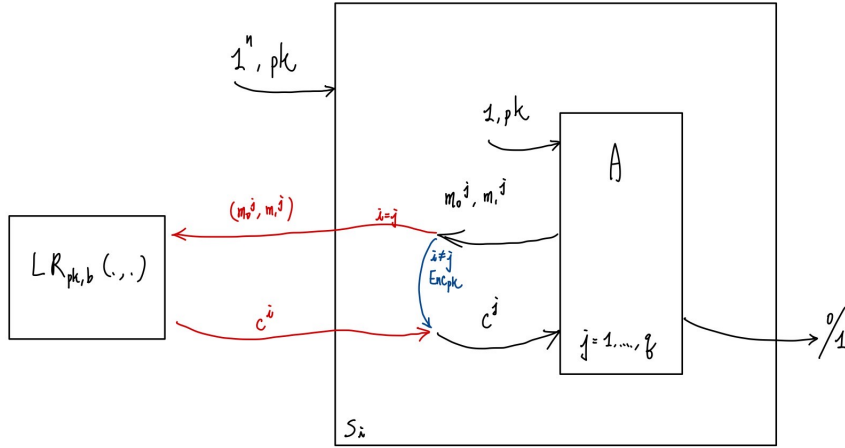


Figure 1: Simulator Model

On  $j^{th}$  query of  $A$  ( $m_0^j, m_1^j$ ):

- If  $j < i$ ,  $S_i$  runs  $c \leftarrow Enc_{pk}(m_1^j)$
- If  $j > i$ ,  $S_i$  runs  $c \leftarrow Enc_{pk}(m_0^j)$
- If  $j = i$ ,  $S_i$  queries to LR oracle and gives the result to  $A$

$$\begin{cases} S_i \text{ is in the left world } (b = 0), \text{ then we perfectly simulate } Hybrid(i-1) \\ S_i \text{ is in the right world } (b = 1), \text{ then we perfectly simulate } Hybrid(i) \end{cases} \quad (1)$$

By triangle inequality,

$$\begin{aligned} Adv_{\pi}^{CPA}(A) &= |Pr(A = 1 \text{ in } Hybrid(0)) - Pr(A = 1 \text{ in } Hybrid(q))| \\ &= |Pr(A = 1 \text{ in } Hybrid(0)) - Pr(A = 1 \text{ in } Hybrid(1)) + Pr(A = 1 \text{ in } Hybrid(1)) \\ &\quad - Pr(A = 1 \text{ in } Hybrid(2)) + Pr(A = 1 \text{ in } Hybrid(2)) \cdots - Pr(A = 1 \text{ in } Hybrid(q))| \\ &\leq \sum_{i=1}^q Adv_{\pi}^{single-CPA}(S_i) = q \cdot negl(n) = negl(n) \end{aligned}$$

The theorem implies we can encrypt long messages bit-by-bit (or block-by-block) or broken up in any other many calls on "short" messages, which is acceptable by the theorem.

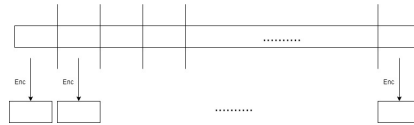


Figure 2: Block-by-block encryption

**Theorem:** Any public key encryption scheme with deterministic  $Enc_{pk}(\cdot)$  can not be CPA secure even for 1 query.

**Proof:** query  $c \leftarrow LR_{pk,b}(m_0, m_1)$  for any  $m_0 \neq m_1$ . Then, run  $c' = Enc_{pk}(m_0)$ . If  $c = c'$  outputs 0, else 1. Because the adversary knows the query  $(m_0, m_1)$ , the adversary has perfect advantage on distinguishing  $c$  and  $c'$ .

## 2 El Gamal Cryptosystem

El Gamal is the public key encryption version of Diffie Hellman. It works as follows:

$$Alice \xrightleftharpoons[B = g^b \in G]{A = g^a \in G} Bob$$

choose random  $a \leftarrow Z_q$

choose random  $b \leftarrow Z_q$

$$K = B^a = g^{ab \bmod q} \in G$$

$$K = A^b = g^{ba \bmod q} \in G$$

where  $G$  is a group of order  $q$  and  $g$  is the generator of  $G$ .

$K$  is the secret key derived by two parties. We use the properties of cyclic group to get random number with multiplication.

We can look at El Gamal Cryptosystem in terms of  $(Gen, Enc, Dec)$ :

**Idea:** Basically, message is  $M \in G$ , the "one-time-pad effect" would involve multiplying  $M$  with something random  $K$ .

- $Gen(1^n)$ : choose random  $a \leftarrow Z_q$  output  $(pk = A = g^a \in G, sk = a) \leftarrow$  at Alice computes
- $Enc(pk = A, M \in G)$ : choose random  $b \leftarrow Z_q$  output ciphertext  $(B = g^b \in G, C = M \cdot A^b \in G) \leftarrow$  what Bob computes
- $Dec(sk = a, (B, C))$ : compute  $K = B^a$ , output  $C \cdot K^{-1} \in G$

**Correctness:**  $\forall M \in G, (pk = g^a, sk = a)$

$$Enc(pk, A) = (B = g^b, C = M \cdot (g^a)^b)$$

$$Dec(B, C) = C \cdot (B^a)^{-1} = M \cdot g^{ab} \cdot (g^{(ab)})^{-1} = M$$

**CPA Security:** Based on the DDH assumption over  $G : (g, g^a, g^b, g^{ab}) \in G^4$ , where  $a, b \leftarrow Z_q$ , is indistinguishable from  $(g, g^a, g^b, g^c) \in G^4$ , where  $a, b, c \leftarrow Z_q$ .

**Theorem:** if DDH holds for  $G$ , then El Gamal is CPA-secure.

Proof: Let  $A$  be any feasible p.p.t attacker against El Galal. Use  $A$  to construct the distinguisher against DDH.

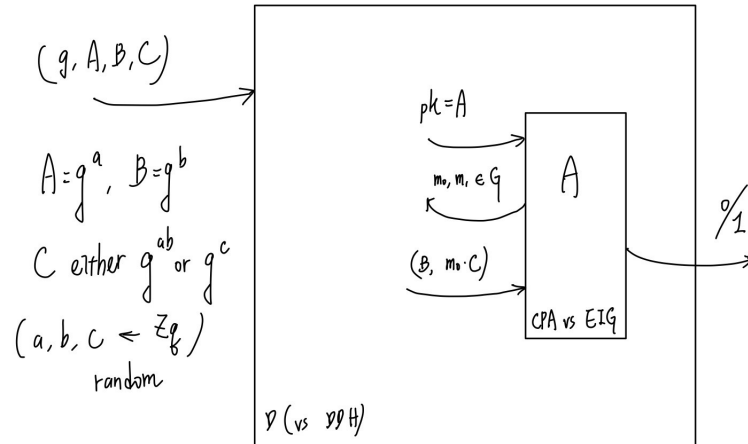


Figure 3: Use the attacker against El Galal to construct the distinguisher against DDH

If  $(g, A, B, C)$  is a DH tuple ("real world"),  $D$  perfectly simulates the left CPA world because  $C = g^{ab}$ .

Ideal world:  $(g, A, B, C)$  is random then  $D$  perfectly simulates a "hybrid" CPA world where the ciphertext is two independent random-group elements (regardless of message).

Symmetrically, we can construct  $D'$  vs DDH that replies  $A$  with  $(B, m_1 \cdot C)$

$$Adv^{CPA}(A) \leq Adv^{DDH}(D) + Adv^{DDH}(D') = negl(n) + negl(n) = negl(n)$$