ElGamal cryptosystem

Ganyu (Bruce) Xu

August 6, 2024

1 The ElGamal cryptosystem

The ElGamal cryptosystem is a public key encryption scheme that mainly operates on the discrete log problem. Each instance of the encryption scheme is parameterized by a cyclic group G with prime order q, a generator g of this cyclic group. The routines of the encryption scheme is shown in figure 1

Algorithm 1 `KeyGen`

1: $x \xleftarrow{\$} \mathbb{Z}_q$
2: $u \leftarrow g^x$
3: $\texttt{pk} \leftarrow u, \texttt{sk} \leftarrow x$
4: **return** $(\texttt{pk}, \texttt{sk})$

Algorithm 2 $\texttt{Enc}(\texttt{pk} = u, m \in G)$

1: $y \xleftarrow{\$} \mathbb{Z}_q$
2: $v \leftarrow g^y$
3: $w \leftarrow u^y$ $\qquad \triangleright\ w = g^{xy}$
4: $c \leftarrow (v, m \cdot w)$
5: **return** c

Algorithm 3 $\texttt{Dec}(\texttt{sk} = x, c)$

1: $(c_1, c_2) \leftarrow c$
2: $\hat{w} \leftarrow c_1^x$
3: $\hat{m} \leftarrow c_2 \cdot \hat{w}^{-1}$
4: **return** $\hat{m}$

Figure 1: ElGamal encryption scheme is IND-CPA secure if DDH holds

The IND-CPA security of the ElGamal cryptosystem depends on the hardness of the following two problems:

Definition 1.1 (Computational Diffie-Hellman Problem)**.** *Let G be a cyclic group with prime order q and generator g. Let $x, y \xleftarrow{\$} \mathbb{Z}_q$ be uniformly random samples. Given g, g^x, g^y, compute g^{xy}*

Definition 1.2 (Decisional Diffie-Hellman Problem)**.** *Let G be a cyclic group with prime order q and generator g. Let $x, y, z \xleftarrow{\$} \mathbb{Z}_q$ be uniformly random samples. Given g, g^x, g^y, distinguish g^{xy} from g^z*

Theorem 1.1. *For every IND-CPA adversary A against the ElGamal cryptosystem, there exists an adversary B against the DDH game such that*

$$\mathit{Adv}(A) = 2 \cdot \mathit{Adv}(B)$$

2 CCA-secure ElGamal construction

The ElGamal cryptosystem presented in figure 1 is not secure against chosen-ciphertext attacks. For a simple attack, consider an adversary who just obtained the challenge encryption $c = (g^y, m \cdot (g^x)^y)$, where $y \xleftarrow{\$} \mathbb{Z}_q^*, m \xleftarrow{\$} G$ are sampled by the challenger, and g^x is the public key. The adversary can pick some non-zero $y' \in \mathbb{Z}_q^*$ and compute a distinct encryption of m using the challenge ciphertext:

$$c' = (g^y \cdot g^{y'}, m \cdot (g^x)^y \cdot (g^x)^{y'}) = (g^{y+y'}, m \cdot (g^x)^{y+y'})$$

The adversary can then query the decryption oracle on c', and the decryption oracle will return m, which can then be returned to win the OW-CCA game.

[BS20] presented a hybrid encryption scheme that combined the ElGamal cryptosystem with an IND-CPA symmetric cipher $\mathcal{E} = (\mathtt{Enc}, \mathtt{Dec})$ into a public-key encryption scheme. This CCA-secure ElGamal cryptosystem (which we will denote by HPKE for short) is parameterized by:

1. A cyclic group G of prime order q with generator g
2. A symmetric cipher $\mathcal{E} = (\mathtt{Enc}_s, \mathtt{Dec}_s)$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
3. A hash function $H : G \to \mathcal{K}$

The routines are listed in figure 2

Algorithm 4 `KeyGen`

$x \leftarrow \mathbb{Z}_q$
$u \leftarrow g^x$
$\texttt{pk} \leftarrow u$
$\texttt{sk} \leftarrow x$
return $(\texttt{pk}, \texttt{sk})$

Algorithm 5 $\texttt{Enc}(\texttt{pk} = u, m \in \mathcal{M})$

$y \xleftarrow{\$} \mathbb{Z}_q$
$v \leftarrow g^y$
$w \leftarrow u^y$ $\qquad \triangleright\ w = g^{xy}$
$k \leftarrow H(w)$
$c' \leftarrow \texttt{Enc}_S(k, m)$
$c \leftarrow (v, c')$
return c

Algorithm 6 $\texttt{Dec}(\texttt{sk} = x, c)$

$(v, c') \leftarrow c$
$\hat{w} \leftarrow v^x$
$\hat{k} \leftarrow H(\hat{w})$
$\hat{m} \leftarrow \texttt{Dec}_S(\hat{k}, c')$
return $\hat{m}$

Figure 2: ElGamal HPKE

In this construction *the decryption oracle can be used to construct a decisional Diffie-Hellman problem oracle*:

1. The DDH adversary A receives g^x, g^y, w and needs to decide whether w is g^{xy} or g^z
2. A samples a random message $m \leftarrow \mathcal{M}$ and computes $k \leftarrow H(w)$ and $c \leftarrow \texttt{Enc}_s(k, m)$
3. A queries the decryption oracle on c and receives some "decryption" $\hat{m}$
4. If $\hat{m} = m$, then w is g^{xy}, otherwise w is g^z. This is because if $w = g^{xy}$, then $k \leftarrow H(w)$ is the correct symmetric key, so the decryption oracle will decrypt correctly. On the other hand, if $w = g^z$, then $k \leftarrow H(w)$ is a uniformly random key, so the decryptino oracle will not decrypt correctly.

Even though the decryption oracle allows an IND-CCA adversary to solve the decisional Diffie-Hellman problem, the hybrid construction remains CCA secure. This is because *to our knowledge today, solving the decisional Diffie-Hellman problem does not give non-negligible advantage to solving the computational Diffie-Hellman problem.* This idea is expressed in a modified assumption:

Definition 2.1 (Interactive computational Diffie-Hellman problem)**.** *Let G be a cyclic group of prime order q with generator g. Let $x, y, z \xleftarrow{\$} \mathbb{Z}_q^*$ be uniformly random samples. Given g, g^x, g^y and a decisional Diffie-Hellman oracle $\mathcal{O} : (g^x, g^y, w \in \{g^{xy}, g^z\}) \mapsto [\![w = g^{xy}]\!]$, there is no efficient adversary who can compute g^{xy} with non-negligible advantage.*

Finally we will put everything together into the security theorem for CCA ElGamal.

Theorem 2.1. *Under the random oracle model, for every IND-CCA adversary A against the HPKE, there exists an* interactive computational Diffie-Hellman problem *adversary B and an IND-CPA adversary C against the symmetric encryption scheme such that*

$$\texttt{Adv}_{\texttt{IND-CCA}}(A) \leq \texttt{Adv}_{\texttt{ICDH}}(B) + \texttt{Adv}_{\texttt{IND-CPA}}(C)$$

Proof. We will prove using a sequence of games. The games are listed in figure 3

Algorithm 7 Games $G_0 - G_1$

$x \xleftarrow{\$} \mathbb{Z}_q^*$ ▷ x is the secret key
$u \leftarrow g^x$ ▷ u is the public key
$(m_0, m_1) \leftarrow A^{\mathcal{O}^{\mathsf{Dec}}}(u)$
$y \xleftarrow{\$} \mathbb{Z}_q^*$
$v \leftarrow g^y$
$w \leftarrow u^y$
$k \leftarrow H(w)$ ▷ Game 0
$k \xleftarrow{\$} \mathcal{K}$ ▷ Game 1
$b \xleftarrow{\$} \{0, 1\}$
$c' \leftarrow \texttt{Enc}_s(k, m_b)$
$c^* \leftarrow (v, c')$
$\hat{b} \leftarrow A^{\mathcal{O}^{\mathsf{Dec}}}(u, c^*, (m_0, m_1))$
return $[\![\hat{b} = b]\!]$

Figure 3: Sequence of games

Game 0 is the standard IND-CCA game: $\texttt{Adv}_0(A) \coloneqq \texttt{Adv}_{\texttt{IND-CCA}}(A)$

Game 1 is identical to game 0, except that in the challenge encryption, the symmetric key $k \xleftarrow{\$} \mathcal{K}$ is uniformly random instead of pseudorandomly derived. Under the random oracle model, the two games are statistically indistinguishable from adversary A's perspective unless A queries H on $w = g^{xy}$. Denote this event by $\texttt{QUERY}^*$, then by the difference lemma:

$$\texttt{Adv}_0(A) - \texttt{Adv}_1(A) \leq P\left[\texttt{QUERY}^*\right]$$

Game 1 can be entirely simulated by an IND-CPA adversary C against the symmetric cipher. C can generate the ElGamal keypair on its own, simulate the hash oracle H, and service A's decryption queries (using the generated keypair) before A outputs the chosen plaintexts m_0, m_1. When A outputs the chosen plaintexts m_0, m_1, C outputs them as its own chosen plaintexts and receives the challenge ciphertext c'. C then samples random $y \xleftarrow{\$} \mathbb{Z}_q^*$, computes $v \leftarrow g^y$, and returns $c^* = (v, c')$ to A as A's challenge encryption. Finally, when A outputs its guess $\hat{b}$, C passes $\hat{b}$ as its own guess. It is easy to see that C wins the IND-CPA game if and only if A wins game 1:

$$\texttt{Adv}_1(A) = \texttt{Adv}_{\texttt{IND-CPA}}(C)$$

We now bound the probability $P\left[\texttt{QUERY}^*\right]$ by constructing an interactive computational Diffie-Hellman problem adversary B using A as a subroutine. To do that, B needs to service A's hash queries and decryption queries. The simulated hash oracles and decryption oracles are listed in figure 4. Note that $\mathcal{L}^H$ is used to record the queries made to H, while $\mathcal{L}^{\mathsf{Dec}}$ is used to record symmetric keys used for decryption queries.

Algorithm 8 $\mathcal{O}_1^H(w)$

if $\exists(\tilde{w},\tilde{k}) \in \mathcal{L}^H : \tilde{w} = w$ **then**
  return $\tilde{k}$
else if $\exists(\tilde{v},\tilde{k}) \in \mathcal{L}^{\texttt{Dec}} : \mathcal{O}^{\texttt{DDH}}(g^x, \tilde{v}, w) = 1$ **then**
  $k \leftarrow \tilde{k}$
else
  $k \xleftarrow{\$} \mathcal{K}$
end if
$\mathcal{L}^H \leftarrow \mathcal{L}^H \cup \{(w,k)\}$
return k

Algorithm 9 $\mathcal{O}_1^{\texttt{Dec}}(v, c')$

if $\exists(\tilde{w},\tilde{k}) \in \mathcal{L}^H : \mathcal{O}^{\texttt{DDH}}(g^x, v, \tilde{w}) = 1$ **then**
  return $\texttt{Dec}_s(\tilde{k}, c')$
else if $\exists(\tilde{v},\tilde{k}) \in \mathcal{L}^{\texttt{Dec}} : \tilde{v} = v$ **then**
  return $\texttt{Dec}_s(\tilde{k}, c')$
else
  $k \xleftarrow{\$} \mathcal{K}$
  $\mathcal{L}^{\texttt{Dec}} \leftarrow \mathcal{L}^{\texttt{Dec}} \cup \{(v,k)\}$
  return $\texttt{Dec}_s(k, c')$
end if

Figure 4: Simulated hash oracle and decryption oracle

Assuming that $\mathcal{O}^{\texttt{DDH}}$ is always correct, the IND-CCA adversary A cannot distinguish the simulated oracles from the true oracles.

When A produces the chosen plaintexts m_0, m_1, B needs to perform the challenge encryption:

1. $v \leftarrow g^y$, where g^y is given to B as part of ICDH input
2. $k \xleftarrow{\$} \mathcal{K}$ as per game 1
3. $b \xleftarrow{\$} \{0,1\}; c' \leftarrow \texttt{Enc}_s(k, m_b)$
4. Return (v, c') as the challenge ciphertext

Afterwards, B continues simulating the oracles for A until A halts. Then, B searches through $\mathcal{L}^H$. If $\texttt{QUERY}^*$ happens, then there exists $\tilde{w} \in \mathcal{L}^H$ such that $\mathcal{O}^{\texttt{DDH}}(g^x, g^y, \tilde{w}) = 1$, and B can return $\tilde{w}$ and win the ICDH game. Therefore:

$$P[\texttt{QUERY}^*] \leq \texttt{Adv}_{\texttt{ICDH}}(B)$$

Finally, putting the equations above give us the desired result. □

3 Does this apply to Kyber-AE?

Unfortuantely not.

In the case of the hybrid ElGamal presented in figure 2, the decryption oracle is converted into a decisional Diffie-Hellman oracle, but it can be safely assumed that having a decisional Diffie-Hellman oracle does not provide non-negligible help. However, there is no comparable "loosening of security assumption" for Kyber-AE, where the decapsulation oracle can be converted into a plaintext-checking oracle against the underlying PKE, which then recovers the secret key.

References

[BS20] Dan Boneh and Victor Shoup. A graduate course in applied cryptography. *Draft 0.5*, 2020.