Security Proofs of the Compact ElGamal Encryption Scheme

oblivious-file-sharing

1 Syntax

We provide a simple syntax of the compact ElGamal scheme that suffices our security proofs. Consider an ElGamal encryption scheme that are specialized to encrypt a plaintext vector that comprises n Ristretto points, as follows.

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}\left(1^\lambda\right)$ generates the public key and the private key.
- $\mathsf{ct} \leftarrow \mathsf{Encrypt}\left(\mathsf{pk}, \mathsf{pt}\right)$ encrypts a plaintext vector, which comprises n Ristretto points.
- $\mathsf{pt} \leftarrow \mathsf{Decrypt}\left(\mathsf{sk}, \mathsf{ct}\right)$ decrypts a ciphertext vector, which comprises $(n+1)$ Ristretto points.
- $\mathsf{ct}' \leftarrow \mathsf{Rerand}\left(\mathsf{pk}, \mathsf{ct}\right)$ rerandomizes a ciphertext vector.

2 Security Definition for Semantic Security

We consider semantic security: a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, which consists of two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, tries to distinguish the ciphertexts in the chosen-plaintext setting.

For a pair of randomly generated keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, we let the first algorithm $\mathcal{A}_1(\mathsf{pk})$ choose two plaintexts $(\mathsf{ptA}, \mathsf{ptB})$ and some state information st; then, the second algorithm $\mathcal{A}_2$ cannot distinguish the following two distributions with a non-negligible advantage.

$$(\mathsf{st}, \mathsf{Encrypt}\left(\mathsf{pk}, \mathsf{ptA}\right))$$
$$(\mathsf{st}, \mathsf{Encrypt}\left(\mathsf{pk}, \mathsf{ptB}\right))$$

Thus, we write

$$(\mathsf{st}, \mathsf{Encrypt}\left(\mathsf{pk}, \mathsf{ptA}\right)) \overset{\mathsf{c}}{\approx} (\mathsf{st}, \mathsf{Encrypt}\left(\mathsf{pk}, \mathsf{ptB}\right)) .$$

where $\overset{\mathsf{c}}{\approx}$ refers to computational indistinguishability. In this definition, we require that each of the two plaintexts provided by $\mathcal{A}_1$, which is ptA or ptB, is correctly encoded. This check can be conducted in polynomial time.

3 Security Definition of Rerandomization

Rerandomization is another property of the compact ElGamal encryption. For simplicity, we consider a notion of perfect security, as follows.

We consider a PPT adversary $\mathcal{A}$, which consists of two algorithms A_1 and A_2. We simply let A_1 output a key pair $(\mathsf{sk}, \mathsf{pk})$, one plaintext pt, one ciphertext ct, and some state information st. As long as the key pair and the ciphertext are legitimate, which means that there exists some random tape using which $\mathsf{KeyGen}(1^\lambda)$ outputs that key pair, and there exists some random tape using which $\mathsf{Encrypt}(\mathsf{pk}, \mathsf{pt})$ outputs that ciphertext, we want

$$(\mathsf{st}, \mathsf{Rerand}(\mathsf{pk}, \mathsf{ct})) \overset{\mathsf{p}}{\approx} (\mathsf{st}, \mathsf{Encrypt}(\mathsf{pk}, \mathsf{pt})) \ ,$$

where $\overset{\mathsf{p}}{\approx}$ refers to perfect indistinguishability.

4 Construction

The underlying curve for Ristretto, Curve25519, is a curve of the order $8q$. The Ristretto group is a subgroup with order q, in which $q \geq 2^\ell$. We let G be a base point of the Ristretto subgroup. The compact ElGamal encryption scheme has the following construction:

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}\left(1^\lambda\right)$.
  - Samples $\mathsf{sk}_i \leftarrow_{\$} \{0, 1, ..., q-1\}$ for $i \in \{1, 2, ..., n\}$.
  - Computes $\mathsf{pk}_i \leftarrow \mathsf{sk}_i \cdot G$ for $i \in \{1, 2, ..., n\}$.
  - Lets $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2, ..., \mathsf{sk}_n)$.
  - Lets $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, ..., \mathsf{pk}_n)$.
  - Outputs $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{pt})$.
  - Parses pt as an array of n Ristretto points: $\mathsf{pt} = (\mathsf{pt}_1, \mathsf{pt}_2, ..., \mathsf{pt}_n)$.
  - Parses pk as an array of n Ristretto epoints: $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, ..., \mathsf{pk}_n)$.
  - Samples $r \leftarrow_{\$} \{0, 1, ..., q-1\}$.
  - Computes $\mathsf{ct}_i \leftarrow \mathsf{pt}_i + r \cdot \mathsf{pk}_i$ for $i \in \{1, 2, ..., n\}$.
  - Computes $\mathsf{ct}_{n+1} \leftarrow r \cdot G$.
  - Lets $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, ..., \mathsf{ct}_{n+1})$.
  - Outputs ct.
- $\mathsf{pt} \leftarrow \mathsf{Decrypt}\left(\mathsf{sk}, \mathsf{ct}\right)$.
  - Parses sk as an array of n scalar values: $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2, ..., \mathsf{sk}_n)$.
  - Parses ct as an array of $(n+1)$ points: $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, ..., \mathsf{ct}_{n+1})$.

  - Computes $\mathsf{pt}_i \leftarrow \mathsf{ct}_i - \mathsf{sk}_i \cdot \mathsf{ct}_{n+1}$ for $i \in \{1, 2, ..., n\}$.
  - Lets $\mathsf{pt} = (\mathsf{pt}_1, \mathsf{pt}_2, ..., \mathsf{pt}_n)$.
  - Outputs pt.

- $\mathsf{ct}' \leftarrow \mathsf{Rerand}\,(\mathsf{pk}, \mathsf{ct})$.
  - Parses pk as an array of n points: $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, ..., \mathsf{pk}_n)$.
  - Parses ct as an array of $(n+1)$ points: $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, ..., \mathsf{ct}_{n+1})$.
  - Samples $r' \leftarrow_{\$} \{0, 1, ..., q-1\}$.
  - Computes $\mathsf{ct}'_i \leftarrow \mathsf{ct}_i + r' \cdot \mathsf{pk}_i$ for $i \in \{1, 2, ..., n\}$.
  - Computes $\mathsf{ct}'_{n+1} \leftarrow \mathsf{ct}_{n+1} + r' \cdot G$.
  - Lets $\mathsf{ct}' = (\mathsf{ct}'_1, \mathsf{ct}'_2, ..., \mathsf{ct}'_{n+1})$.
  - Outputs ct'.

5 Security Proof for Semantic Security

We first recall the decisional Diffie-Hellman assumption in the Ristretto group with a base point G, as follows. For random $k, x, r \leftarrow_{\$} \{0, 1, ..., q-1\}$, we have:

$$\begin{pmatrix} k \cdot G, \\ x \cdot G, \\ k \cdot x \cdot G \end{pmatrix} \overset{c}{\approx} \begin{pmatrix} k \cdot G, \\ x \cdot G, \\ r \cdot G \end{pmatrix}.$$

Extending from this assumption, we want to claim the following result:

Lemma 1. *In the Ristretto group (of prime-order q), for random $k, x_1, x_2, ..., x_n, r_1, r_2, ...,$ $r_n \leftarrow_{\$} \{0, 1, ..., q-1\}$, we have:*

$$\begin{pmatrix} k \cdot G, \\ x_1 \cdot G, \\ x_2 \cdot G, \\ ..., \\ x_n \cdot G, \\ k \cdot x_1 \cdot G, \\ k \cdot x_2 \cdot G, \\ ..., \\ k \cdot x_n \cdot G \end{pmatrix} \overset{c}{\approx} \begin{pmatrix} k \cdot G, \\ x_1 \cdot G, \\ x_2 \cdot G, \\ ..., \\ x_n \cdot G, \\ r_1 \cdot G, \\ r_2 \cdot G, \\ ..., \\ r_n \cdot G \end{pmatrix}.$$

Proof. We prove the lemma by hybrid argument. Consider the following hybrid distribution H_i.

$$H_i : \begin{pmatrix} k \cdot G, \\ x_1 \cdot G, \\ x_2 \cdot G, \\ ..., \\ x_n \cdot G, \\ r_1 \cdot G, \\ r_2 \cdot G, \\ ..., \\ r_i \cdot G, \\ k \cdot x_{i+1} \cdot G, \\ ..., \\ k \cdot x_n \cdot G \end{pmatrix}$$

□

We know that the left-hand side of Lemma 1 equals H_0, and the right-hand side equals H_n. We want to prove that for any $i \in \{0, 1, ..., n-1\}$, we have:

$$H_i \overset{c}{\approx} H_{i+1}.$$

If there is no contradiction, by hybrid argument, we know that H_0 and H_n are indistinguishable, and Lemma 1 holds.

If there is any contradiction, we suppose that there is a probabilistic polynomial-time (PPT) algorithm $\mathcal{D}_i$ that can distinguish H_i and H_{i+1} with a non-negligible advantage. However, this supposition implies that there is also a PPT algorithm $\mathcal{E}$ that violates the decisional Diffie-Hellman assumption. The construction of $\mathcal{E}$ is as follows:

- The challenge for $\mathcal{E}$ to solve is either from the left-hand side distribution (denoted by D_L) or the right-hand side distribution (denoted by D_R), as follows:

  $$D_L : (k \cdot G, x \cdot G, k \cdot x \cdot G),$$
  $$D_R : (k \cdot G, x \cdot G, r \cdot G)$$

  Let us write the challenge as $Q = (Q_1, Q_2, Q_3)$, where $Q_1 = k \cdot G$, $Q_2 = x \cdot G$, and Q_3 equals $k \cdot x \cdot G$ or $r \cdot G$.

- The algorithm $\mathcal{E}$'s idea is to map the challenge into either the distribution H_i or the distribution H_{i+1}. The difference between these two distributions are highlighted as follows.

  $$H_i : (k \cdot G, ..., r_i \cdot G, \boxed{k \cdot x_{i+1} \cdot G}, k \cdot x_{i+2} \cdot G, ..., k \cdot x_n \cdot G),$$
  $$H_{i+1} : (k \cdot G, ..., r_i \cdot G, \boxed{r_{i+1} \cdot G}, k \cdot x_{i+2} \cdot G, ..., k \cdot x_n \cdot G).$$

- To do that, the algorithm $\mathcal{E}$ samples $x_1, x_2, ..., x_i, x_{i+2}, x_{i+3}, ..., x_n, r_1, r_2, ..., r_i, r_{i+2}, r_{i+3}, ..., r_n \leftarrow_{\$} \{0, 1, ..., q-1\}$ and generates the following new challenge Q':

$$Q' : \begin{pmatrix} Q_1, \\ x_1 \cdot G, \\ x_2 \cdot G, \\ ..., \\ x_i \cdot G, \\ Q_2, \\ x_{i+2} \cdot G, \\ x_{i+3} \cdot G, \\ ..., \\ x_n \cdot G, \\ r_i \cdot G, \\ Q_3, \\ x_{i+2} \cdot Q_1, \\ x_{i+3} \cdot Q_1, \\ ..., \\ x_n \cdot Q_1 \end{pmatrix},$$

  which is either equivalent to sampling from H_i or equivalent to sampling from H_{i+1}, as we can see by comparing the distributions.

- The algorithm $\mathcal{E}$ outputs whatever D_i outputs for the challenge Q'. Recall that the algorithm D_i has a non-negligible advantage in distinguishing H_i and H_{i+1}; it implies that the algorithm $\mathcal{E}$ also has that advantage.

However, the decisional Diffie-Hellman assumption says that no such probabilistic polynomial-time algorithm $\mathcal{E}$ exists, a contradiction. Thus, there is no such attacker D_i, and the lemma holds.

Now, we claim the following lemma.

Lemma 2. *In the compact ElGamal algorithm constructed over the Ristretto group, for a randomly sampled key pair* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and for arbitrary two plaintexts* $(\mathsf{ptA}, \mathsf{ptB})$ *chosen by the previously mentioned PPT algorithm* $\mathcal{A}_1$ *together with the state* st*, the following two distributions are indistinguishable.*

$$\begin{pmatrix} \mathsf{st}, \\ \mathsf{ptA}_1 + k \cdot x_1 \cdot G, \\ \mathsf{ptA}_2 + k \cdot x_2 \cdot G, \\ ..., \\ \mathsf{ptA}_n + k \cdot x_n \cdot G, \\ k \cdot G \end{pmatrix} \overset{c}{\approx} \begin{pmatrix} \mathsf{st}, \\ \mathsf{ptB}_1 + k \cdot x_1 \cdot G, \\ \mathsf{ptB}_2 + k \cdot x_2 \cdot G, \\ ..., \\ \mathsf{ptB}_n + k \cdot x_n \cdot G, \\ k \cdot G \end{pmatrix}$$

where ptA *is parsed as* $(\mathsf{ptA}_1, \mathsf{ptA}_2, ..., \mathsf{ptA}_n)$*, and* ptB *is parsed as* $(\mathsf{ptB}_1, \mathsf{ptB}_2, ..., \mathsf{ptB}_n)$*.*

Proof. Recall that st is computed by the algorithm A_1 with the public key as input, which comprises $(x_1 \cdot G, x_2 \cdot G, ..., x_n \cdot G)$ here. By applying Lemma 1, we know:

$$\begin{pmatrix} \mathsf{st}, \\ \mathsf{ptA}_1 + k \cdot x_1 \cdot G, \\ \mathsf{ptA}_2 + k \cdot x_2 \cdot G, \\ ..., \\ \mathsf{ptA}_n + k \cdot x_n \cdot G, \\ k \cdot G \end{pmatrix} \overset{\mathsf{c}}{\approx} \begin{pmatrix} \mathsf{st}, \\ \mathsf{ptA}_1 + r_1 \cdot G, \\ \mathsf{ptA}_2 + r_2 \cdot G, \\ ..., \\ \mathsf{ptA}_n + r_n \cdot G, \\ k \cdot G \end{pmatrix}$$

where $r_1, r_2, ..., r_n$ are individually sampled from $\{0, 1, ..., q-1\}$.

And,

$$\begin{pmatrix} \mathsf{st}, \\ \mathsf{ptB}_1 + k \cdot x_1 \cdot G, \\ \mathsf{ptB}_2 + k \cdot x_2 \cdot G, \\ ..., \\ \mathsf{ptB}_n + k \cdot x_n \cdot G, \\ k \cdot G \end{pmatrix} \overset{\mathsf{c}}{\approx} \begin{pmatrix} \mathsf{st}, \\ \mathsf{ptB}_1 + r_1 \cdot G, \\ \mathsf{ptB}_2 + r_2 \cdot G, \\ ..., \\ \mathsf{ptB}_n + r_n \cdot G, \\ k \cdot G \end{pmatrix}$$

where $r_1, r_2, ..., r_n$ are individually sampled from $\{0, 1, ..., q-1\}$.

However, the right-hand sides of both results about are actually the same distribution as follows:

$$\begin{pmatrix} \mathsf{st}, \\ r_1 \cdot G, \\ r_2 \cdot G, \\ ..., \\ r_n \cdot G, \\ k \cdot G \end{pmatrix}.$$

As a result, we know that the both distributions on the left-hand sides are computationally indistinguishable, and therefore, the lemma is correct. □

6 Security Proof of Rerandomization

We assume that r was the random number used during the encryption that generates ct. We can write ct using this random number, the public key $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, ..., \mathsf{pk}_n)$, and the plaintext $\mathsf{pt} = (\mathsf{pt}_1, \mathsf{pt}_2, ..., \mathsf{pt}_n)$, as follows:

$$\begin{aligned} \mathsf{ct} &= (\mathsf{ct}_1, \mathsf{ct}_2, ..., \mathsf{ct}_{n+1}). \\ \mathsf{ct}_i &= \mathsf{pt}_i + r \cdot \mathsf{pk}_i \quad (i \in \{1, 2, ..., n\}). \\ \mathsf{ct}_{n+1} &= r \cdot G. \end{aligned}$$

Let us consider the left-hand side where we use rerandomization. The new ciphertext ct' can be expressed as follows:

$$\begin{aligned} \mathsf{ct}' &= (\mathsf{ct}'_1, \mathsf{ct}'_2, ..., \mathsf{ct}'_{n+1}). \\ \mathsf{ct}'_i &= \mathsf{pt}_i + (r + r') \cdot \mathsf{pk}_i \quad (i \in \{1, 2, ..., n\}). \\ \mathsf{ct}'_{n+1} &= (r + r') \cdot G. \end{aligned}$$

where r' is randomly sampled from $\{0, 1, ..., q - 1\}$.

Let us consider the right-hand side where we do the encryption again. The new ciphertext ct'' can be expressed as follows:

$$\begin{aligned}
\mathsf{ct}'' &= (\mathsf{ct}''_1, \mathsf{ct}''_2, ..., \mathsf{ct}''_{n+1}). \\
\mathsf{ct}''_i &= \mathsf{pt}_i + r'' \cdot \mathsf{pk}_i \quad (i \in \{1, 2, ..., n\}). \\
\mathsf{ct}''_{n+1} &= r'' \cdot G.
\end{aligned}$$

where r'' is randomly sampled from $\{0, 1, ..., q - 1\}$.

These two ciphertext distributions are the same. For this reason, we have the perfect indistinguishability for the rerandomization.