LIT-SiGamal: An efficient isogeny-based PKE based on a LIT diagram

Tomoki Moriya

School of Computer Science, University of Birmingham, UK
t.moriya@bham.ac.uk

Abstract. In this paper, we propose a novel isogeny-based public key encryption (PKE) scheme named LIT-SiGamal. This is based on a LIT diagram and SiGamal. SiGamal is an isogeny-based PKE scheme that uses a commutative diagram with an auxiliary point. LIT-SiGamal uses a LIT diagram which is a commutative diagram consisting of large-degree horizontal isogenies and relatively small-degree vertical isogenies, while the original SiGamal uses a CSIDH diagram.
A strength of LIT-SiGamal is efficient encryption and decryption. QFESTA is an isogeny-based PKE scheme proposed by Nakagawa and Onuki, which is a relatively efficient scheme in isogeny-based PKE schemes. In our experimentation with our proof-of-concept implementation, the computational time of the encryption of LIT-SiGamal is as efficient as that of QFESTA, and that of the decryption of LIT-SiGamal is about 5x faster than that of QFESTA.

Keywords: isogeny-based cryptography; Kani's theorem; public key encryption;

1 Introduction

Isogeny-based cryptography is one of the candidates for post-quantum cryptography, which is based on the Isogeny problem of supersingular elliptic curves. The strength of isogeny-based cryptosystems is that their key sizes are short. An isogeny-based digital signature SQISign [11] is considered the most compact post-quantum digital signature. However, isogeny-based cryptosystems generally take longer to execute than other post-quantum cryptosystems.

SIDH [15] was a promising efficient isogeny-based key exchange scheme. However, several researchers showed that SIDH can be broken in polynomial time in 2022 [6,17,24]. The core of these attacks is to use Kani's theorem [16], which describes a relationship between isogenies of elliptic curves and those of abelian varieties of dimension 2.

After breaking SIDH, several isogeny-based schemes have been proposed, which are possible alternatives to SIDH. M-SIDH [13] and bin-SIDH [4] are isogeny-based key exchange schemes constructed by making SIDH more complex and resistant to the SIDH attacks. FESTA proposed by Basso, Maino, and Pope

[5] is constructed in another direction from M-SIDH and bin-SIDH, which is the first isogeny-based public key encryption (PKE) scheme that uses Kani's theorem as a trapdoor. QFESTA [21] is an improvement to FESTA, and IS-CUBE [18] is a key encapsulation mechanism (KEM) constructed by using a technique in FESTA. Thus, various schemes have already been proposed. However, these schemes are still less efficient than SIDH. Therefore, it is worthwhile to attempt to construct more efficient isogeny-based schemes.

1.1 Contribution

We propose a novel isogeny-based PKE scheme named LIT-SiGamal. This scheme is constructed by merging already known isogeny-based schemes SiGamal [20] and IS-CUBE. More precisely, SiGamal is an ElGamal-like PKE scheme that is constructed by a commutative diagram with an auxiliary point of smooth order. The outline of SiGamal is as follows:

$$\begin{array}{ccc} (E, R) & \xrightarrow{\phi_1} & (E_1, \phi_1(R)) \\ \downarrow{\scriptstyle \phi_2} & & \downarrow{\scriptstyle \phi_2'} \\ (E_2, \phi_2(R)) & \xrightarrow[\phi_1']{} & (E_3, \phi_2'(\phi_1(R))) \end{array}$$

1. Suppose that the above diagram is commutative, and R is a point of smooth order in E.
2. Alice computes $(E, R, E_1, \phi_1(R))$ as her public key.
3. Bob computes $(E_2, \phi_2(R), E_3, \boldsymbol{\mu}\phi_2'(\phi_1(R)))$ as a ciphertext. The value μ is his plaintext.
4. Alice computes $\phi_1' \colon E_2 \to E_3$ and obtains μ by solving the Discrete Logarithm problem for $\mu\phi_2'(\phi_1(R)))$ and $\phi_1'(\phi_2(R)))$.

LIT-SiGamal is a SiGamal-based PKE scheme that uses a LIT diagram as the base commutative diagram. A LIT diagram is a commutative diagram with $\deg \phi_1 \gg \deg \phi_2 = \deg \phi_2'$ introduced for constructing IS-CUBE, and the overhead of the prime required to construct a LIT diagram is relatively small.

One beneficial property of LIT-SiGamal is that the encryption and decryption are efficient. In our experimentation with our PoC implementation, the encryption of LIT-SiGamal is as efficient as that of QFESTA, and the decryption of LIT-SiGamal is about 5x faster than that of QFESTA. Moreover, the total time of the encryption and decryption of LIT-SiGamal is the shortest for the security parameters 192 and 256 among isogeny-based schemes compared in this study. Our comparison is summarized in Table 5.

Organization.

We first introduce some background knowledge associated with our study in Section 2. In Section 3, we introduce some isogeny-based schemes that are strongly

related to LIT-SiGamal. Section 4 provides the scheme of LIT-SiGamal. In particular, Section 4.1 gives an overview of LIT-SiGamal, Section 4.2 provides the precise scheme of LIT-SiGamal, Section 4.3 explains the method to generate the public key of LIT-SiGamal, and Section 4.4 provides the compressed LIT-SiGamal. We discuss the security of LIT-SiGamal in Section 5. We define security assumptions and prove that LIT-SiGamal is IND-CPA secure in Section 5.1 and estimate the size of the prime for LIT-SiGamal in Section 5.2. Section 5.3 provides an adaptive attack for LIT-SiGamal. Section 6 shows the results related to our PoC implementation. Section 6.1 provides actual primes for LIT-SiGamal and Section 6.2 shows our experimental results of the PoC implementation. We finally conclude our study in Section 7.

2 Preliminaries

2.1 Isogenies

In this subsection, we introduce some mathematical backgrounds related to this study. In particular, we introduce basic knowledge about elliptic curves. See [25] for more details of elliptic curves.

Let p be a prime, and let k be a field of characteristic p. An *elliptic curve defined over* k is an abelian variety defined over k of dimension 1. Let n be an integer, and let $\overline{k}$ be an algebraic closure of k. The n-torsion subgroup of E is the subgroup of E defined by

$$E[n] = \{P \in E(\overline{k}) \mid nP = 0\}.$$

If n is coprime to p, it holds that $E[n] \cong (\mathbb{Z}/n\mathbb{Z})^2$. If $E[p] = \{0\}$, we call E a *supersinglar* elliptic curve.

Let E and E_1 be elliptic curves defined over k. An *isogeny* $\phi\colon E \to E_1$ is a surjective morphism between algebraic varieties E and E_1 that is also a group morphism and whose kernel is a finite subgroup of E. If ϕ is separable as a morphism of algebraic varieties, we call ϕ a *separable* isogeny. If ϕ is a separable isogeny, then it holds that $\deg \phi = \# \ker \phi$. An *ℓ-isogeny* is a separable isogeny whose kernel is a cyclic group of order ℓ. For an isogeny $\phi\colon E \to E'$, there is the isogeny $\hat{\phi}\colon E' \to E$ such that $\phi \circ \hat{\phi} = [\deg \phi]$ and $\hat{\phi} \circ \phi = [\deg \phi]$, where $[n]$ is the multiplication-by-n map. We call $\hat{\phi}$ the *dual isogeny* of ϕ. Let G be a finite subgroup of E. Then, there is a separable isogeny $\phi\colon E \to E'$ with $\ker \phi = G$. The codomain curve E' of ϕ is unique up to the isomorphism, and we denote a representative by E/G. From an elliptic curve E and its finite subgroup G, we can compute a separable isogeny $\phi\colon E \to E/G$ with $\ker \phi = G$ [27]. Generally, from a principally polarized abelian variety A and its finite subgroup G, we can compute a separable isogeny $\phi\colon A \to A/G$ with $\ker \phi = G$. If $\dim A = 2$, we can use formulas provided by [26] or [10]. An *isogeny diamond* or *SIDH diagram* is

the following commutative diagram

$$\begin{array}{ccc} E & \xrightarrow{\phi_1} & E_1 \\ {\scriptstyle \phi_2}\downarrow & & \downarrow{\scriptstyle \phi_2'} \\ E_2 & \xrightarrow[\phi_1']{} & E_3 \end{array}$$

that satisfies $\gcd(\deg\phi_1, \deg\phi_2) = 1$, $\deg\phi_1 = \deg\phi_1'$, and $\deg\phi_2 = \deg\phi_2'$.

Proposition 1 (Kani's theorem [16]). *Let the following diagram be an isogeny diamond:*

$$\begin{array}{ccc} E & \xrightarrow{\phi_1} & E_1 \\ {\scriptstyle \phi_2}\downarrow & & \downarrow{\scriptstyle \phi_2'} \\ E_2 & \xrightarrow[\phi_1']{} & E_3 \end{array}$$

Then the isogeny $\Psi\colon E \times E_3 \to E_1 \times E_2$ *defined by*

$$\Psi = \begin{pmatrix} \hat{\phi}_1 & \hat{\phi}_2 \\ -\phi_2' & \phi_1' \end{pmatrix}$$

satisfies $\ker\Psi = \langle(\phi_1(P), \phi_2(P)) \mid P \in E[\deg\phi_1 + \deg\phi_2]\rangle$.

2.2 Isogeny problems with torsion points information

In this subsection, we introduce basic mathematical problems related to our study. In particular, we define the Isogeny problem and its some variations with torsion points information.

Definition 1 ((Supersingular) Isogeny problem)**.** Let E_1 and E_2 be random supersingular elliptic curves.

We call the following problem the *Isogeny problem*:

Find an isogeny $\phi_1\colon E_1 \to E_2$ *from* (E_1, E_2).

Definition 2 (CSSI problem [15])**.** Let d_1 and d_2 be coprime smooth integers. Let E_1 and E_2 be random supersingular elliptic curves such that there is a separable isogeny $\phi_1\colon E_1 \to E_2$ of degree d_1, and let $\{P, Q\}$ be a random basis of $E_1[d_2]$.

We call the following problem the *CSSI problem*:

Compute ϕ_1 *from* $(d_1, d_2, E_1, E_2, P, Q, \phi_1(P), \phi_1(Q))$.

The hardness of this problem guaranteed the security of SIDH [15]; however, it can be solved in polynomial time if $d_2^2 \geq d_1$ by the SIDH attacks based on Kani's theorem [6,17,24]. Although the Isogeny problem is still considered as hard to solve, it is not easy to construct a cryptosystem based on this problem

because it is too simple. Therefore, to construct isogeny-based schemes, some variants of the Isogeny problem have been proposed. We now introduce the CIST and LIT problem.

The CIST problem is one variant of the Isogeny problem proposed in [5], which is a problem to compute an isogeny from given two elliptic curves and torsion points that are masked by an appropriate matrix.

Definition 3 (CIST problem [5])**.** Let d_1 and d_2 be coprime integers. Let $\mathcal{M}$ is a sufficient large abelian subgroup of $\mathrm{GL}_2(\mathbb{Z}/d_2\mathbb{Z})$. Let E_1 and E_2 be random supersingular elliptic curves such that there is an isogeny $\phi_1 \colon E_1 \to E_2$ of degree d_1, and let $\{P, Q\}$ be a random basis of $E_1[d_2]$. Put ${}^t(P', Q') := \mathbf{A} \cdot {}^t(\phi_1(P), \phi_1(Q))$, where $\mathbf{A}$ is a random matrix in $\mathcal{M}$.

We call the following problem the *CIST problem*:

Compute ϕ_1 *from* $(d_1, d_2, E_1, E_2, P, Q, P', Q', \mathcal{M})$.

As the subgroup $\mathcal{M}$ of $\mathrm{GL}_2(\mathbb{Z}/d_2\mathbb{Z})$, the group of diagonal matrices and that of circulant matrices are suggested in [5].

The LIT problem is another variant of the Isogeny problem. In the setting of the LIT problem, torsion points are revealed, while in the CIST problem setting, torsion points are masked by a matrix. Instead, we assume that the degree of the isogeny ϕ_1 is much larger than the order of the torsion points in the setting of the LIT problem.

Definition 4 (LIT problem [18])**.** Let d_1 and d_2 be coprime integers such that $d_1 \gg d_2$. Let E_1 and E_2 be random supersingular elliptic curves such that there is a separable isogeny $\phi_1 \colon E_1 \to E_2$ of degree d_1, and let $\{P, Q\}$ be a random basis of $E_1[d_2]$.

We call the following problem the *LIT problem*:

Compute ϕ_1 *from* $(d_1, d_2, E_1, E_2, P, Q, \phi_1(P), \phi_1(Q))$.

From the discussion in [18, Section 4.2], if $d_1 > d_2^2 \cdot 2^{2\lambda}$ for the security parameter λ, then the LIT problem may be hard to solve. A *LIT diagram* is a SIDH diagram satisfying the degree inequality $d_1 > d_2^2 \cdot 2^{2\lambda}$.

3 Related schemes

In this section, we explain some isogeny-based schemes related to the construction of LIT-SiGamal.

3.1 FESTA

FESTA [5] is the first isogeny-based PKE scheme that uses Kani's theorem in the construction. The security of FESTA relies on the hardness of the CIST problem. Notably, they introduced the method of sending a SIDH diagram without revealing isogenies by masking torsion points by appropriate matrices. We provide a brief explanation of an outline of FESTA.

Assume that Bob tries to send a message to Alice by FESTA. The outline of FESTA proceeds as follows. Let ℓ_A be a small prime, let $d_{A,1}, d_{A,2}, d_1, d_2$ be integers such that $d_{A,1}d_1 + d_{A,2}d_2 = \ell_A^a$ for some $a \in \mathbb{Z}_{\geq 1}$, let E_0 be a supersingular elliptic curve, let $\{P_A, Q_A\}$ be a basis of $E_0[\ell_A^a]$, and let $\mathcal{M}_a$ be a commutative subgroup of $\mathrm{GL}_2(\mathbb{Z}/\ell_A^a\mathbb{Z})$ of sufficiently large order.

Public key / Secret key: Alice first computes an isogeny $\phi_A \colon E_0 \to E_A$ of degree $d_{A,1}d_{A,2}$ and computes $\phi_A(P_A), \phi_A(Q_A)$. Alice takes a random matrix $\mathbf{A}$ from $\mathcal{M}_a$ and computes ${}^t(P'_A, Q'_A) := \mathbf{A} \cdot {}^t(\phi_A(P_A), \phi_A(Q_A))$. She publishes E_A and (P'_A, Q'_A) as her public key. Let $\mathbf{A}$ be her secret key.

Encryption: Bob computes an isogeny $\phi_1 \colon E_0 \to E_2$ of order d_1 and an isogeny $\phi_2 \colon E_A \to E_2$ of degree d_2. Here, he embeds his plaintext to ϕ_1 appropriately. He takes a random matrix $\mathbf{B}$ from $\mathcal{M}_a$ and computes ${}^t(P_1, Q_1) := \mathbf{B} \cdot {}^t(\phi_1(P_A), \phi_1(Q_A))$ and ${}^t(P_2, Q_2) := \mathbf{B} \cdot {}^t(\phi_2(P'_A), \phi_2(Q'_A))$. He publishes $(E_1, P_1, Q_1, E_2, P_2, Q_2)$ as the ciphertext.

Decryption: Alice computes ${}^t(P'_2, Q'_2) := \mathbf{A}^{-1} \cdot {}^t(P_2, Q_2)$. Note that it follows from the commutativity of $\mathbf{A}$ and $\mathbf{B}$ that

$$(d_1P'_2, d_1Q'_2) = ((\phi_2 \circ \phi_A \circ \hat{\phi}_1)(P_1), (\phi_2 \circ \phi_A \circ \hat{\phi}_1)(Q_1)).$$

Alice computes a (ℓ_A^a, ℓ_A^a)-isogeny Φ from $E_0 \times E_1$ with

$$\ker \Phi = \langle (d_{A,1}P_1, P'_2), (d_{A,1}Q_1, Q'_2) \rangle.$$

It follows from Kani's theorem and $d_{A,1}d_1 + d_{A,2}d_2 = \ell_A^a$ that

$$\Phi = \begin{pmatrix} \phi_{A,1} \circ \hat{\phi}_1 & \hat{\phi}_{A,2} \circ \hat{\phi}_2 \\ * & * \end{pmatrix},$$

where $\phi_{A,1}$ and $\phi_{A,2}$ are isogenies satisfying $\phi_{A,2} \circ \phi_{A,1} = \phi_A$, $\deg \phi_{A,1} = d_{A,1}$, and $\deg \phi_{A,2} = d_{A,2}$. Alice finally obtains ϕ_1 from Φ.

3.2 IS-CUBE

IS-CUBE is an isogeny-based KEM proposed in [18]. The security of IS-CUBE relies on both the LIT and CIST problems. One feature of IS-CUBE is to construct a novel SIDH diagram by constructing random LIT diagrams from the public SIDH diagram. To send the novel SIDH diagram, the user of IS-CUBE uses a technique provided in FESTA (*i.e.,* masking torsion points by appropriate matrices). In this subsection, we explain an outline of IS-CUBE.

Assume that Bob tries to share a key with Alice. The procedure of IS-CUBE proceeds as follows. Let ℓ_A, ℓ_B, ℓ_C be small distinct primes, let a, b, c be integers such that $\ell_A^a > \ell_B^b \gg \ell_C^c$, and let $\mathcal{M}_a$ be a commutative subgroup of $\mathrm{GL}_2(\mathbb{Z}/\ell_A^a\mathbb{Z})$ of sufficiently large order. Let E_0 be a random supersingular elliptic curve, and let $\{P_A, Q_A\}$ be a basis of $E_0[\ell_A^a]$, and let $\{P_C, Q_C\}$ be a basis of $E_0[\ell_C^c]$. Let $\psi \colon E_0 \to \tilde{E}_0$ be an isogeny of degree $\ell_A^a - \ell_B^b$.

Public key / Secret key: Alice first computes an ℓ_B^b-isogeny $\phi_1 \colon E_0 \to E_1$ at random. Moreover, she computes $\phi_1(P_A), \phi_1(Q_A)$ and $\phi_1(P_C), \phi_1(Q_C)$. Alice takes a random matrix $\mathbf{A}$ from $\mathcal{M}_a$ and computes ${}^t(P_1, Q_1) := \mathbf{A} \cdot {}^t(\phi_1(P_A), \phi_1(Q_A))$. She publishes $(E_1, P_1, Q_1, \phi_1(P_C), \phi_1(Q_C))$ as her public key and lets $\mathbf{A}$ be her secret key.

Encapsulation: Bob takes a random element s from $(\mathbb{Z}/\ell_C^c\mathbb{Z})^\times$. Then, he computes three isogenies:

$$\begin{aligned} \phi_{0,B} \colon \tilde{E}_0 &\to \tilde{E}_0' := \tilde{E}_0/\langle \psi(P_C) + s\psi(Q_C) \rangle, \\ \phi_B \colon \tilde{E}_0 &\to E := E_0/\langle P_C + sQ_C \rangle, \\ \phi_{1,B} \colon E_1 &\to E_1' := E_1/\langle \phi_1(P_C) + s\phi_1(Q_C) \rangle. \end{aligned}$$

Let E be Bob's shared key. He takes a random matrix $\mathbf{B}$ from $\mathcal{M}_a$ and computes

$$\begin{pmatrix} P_0' \\ Q_0' \end{pmatrix} := \mathbf{B} \cdot \begin{pmatrix} \phi_{0,B}(\psi(P_A)) \\ \phi_{0,B}(\psi(Q_A)) \end{pmatrix}, \quad \begin{pmatrix} P_1' \\ Q_1' \end{pmatrix} := \mathbf{B} \cdot \begin{pmatrix} \phi_{1,B}(P_1) \\ \phi_{1,B}(Q_1) \end{pmatrix}.$$

He publishes $(\tilde{E}_0', P_0', Q_0', E_1', P_1', Q_1')$ as the ciphertext.

Decapsulation: Alice first computes ${}^t(P_1'', Q_1'') := \mathbf{A}^{-1} \cdot {}^t(P_1', Q_1')$. Note that there are isogenies $\psi' \colon E \to \tilde{E}_0'$ of degree $\ell_A^a - \ell_B^b$ and $\phi_1' \colon E \to E_1'$ of degree ℓ_B^b such that

$$((\ell_A^a - \ell_B^b)P_1'', (\ell_A^a - \ell_B^b)Q_1'') = ((\phi_1' \circ \hat{\psi}')(P_0'), (\phi_1' \circ \hat{\psi}')(Q_0'))$$

because Bob constructs LIT diagrams to compute $\tilde{E}_0'$ and E_1'. Alice computes an (ℓ_A^a, ℓ_A^a)-isogeny Φ with $\ker \Phi = \langle (P_0', P_1''), (Q_0', Q_1'') \rangle$. It follows from Kani's theorem that

$$\Phi = \begin{pmatrix} \hat{\psi}' & \hat{\phi}_1' \\ * & * \end{pmatrix} : \tilde{E}_0' \times E_1' \longrightarrow E \times *.$$

Therefore, Alice can obtain E from Φ.

3.3 SiGamal

SiGamal is an isogeny-based PKE proposed in [20]. We provide a brief explanation of SiGamal in this subsection.

SiGamal is constructed by adding information on points to a commutative diagram. The following diagram shows the basic structure of SiGamal. Here, we let E_0, E_1, E_2, E_3 be elliptic curves, and let R be a point of E_0. Let $\phi_1 \colon E_0 \to E_1$, $\phi_2 \colon E_0 \to E_2$, $\phi_1' \colon E_2 \to E_3$, $\phi_2' \colon E_1 \to E_3$ be isogenies satisfying $\phi_1' \circ \phi_2 = \phi_2' \circ \phi_1$. We suppose that anyone who knows ϕ_1 and E_2 can compute ϕ_1'.

$$\begin{array}{ccc} (E_0, R) & \xrightarrow{\phi_1} & (E_1, \phi_1(R)) \\ {\scriptstyle \phi_2}\downarrow & & \downarrow{\scriptstyle \phi_2'} \\ (E_2, \phi_2(R)) & \xrightarrow[\phi_1']{} & (E_3, \phi_2'(\phi_1(R))) \end{array}$$

Assume that Bob tries to send a message to Alice by using SiGamal. The whole process of SiGamal is as follows:

Public key / Secret key: First, Alice computes the top isogeny ϕ_1 and obtains (E_0, R) and $(E_1, \phi_1(R))$. She publishes $(E_0, R, E_1, \phi_1(R))$ and lets ϕ_1 be her secret key.

Encryption: Bob computes vertical isogenies ϕ_2 and ϕ_2' and obtains $(E_2, \phi_2(R))$ and $(E_3, \phi_2'(\phi_1(R)))$. Denote Bob's message by μ. Bob computes $\mu\phi_2'(\phi_1(R))$ and sends $(E_2, \phi_2(R), E_3, \mu\phi_2'(\phi_1(R)))$ to Alice.

Decryption: Finally, Alice computes ϕ_1' by using her secret key ϕ. She also computes $\phi_1'(\phi_2(R)) = \phi_2'(\phi_1(R))$. The message μ is recovered by solving the Discrete Logarithm Problem for $\phi_2'(\phi_1(R))$ and $\mu\phi_2'(\phi_1(R))$ via the Pohlig-Hellman algorithm [22].

The above construction is the core of SiGamal; however, this construction does not work in general. It is because it is hard to compute ϕ_1' from ϕ_1 and E_2 without knowing ϕ_2 and ϕ_2' generally. In other words, it is not true that *anyone who knows ϕ_1 and E_2 can compute ϕ_1'*. To solve this problem, the original SiGamal uses a CSIDH diagram. CSIDH is an isogeny-based key exchange scheme proposed in [7], which is based on a group action of a specific commutative group (an ideal class group) on a set of supersingular elliptic curves. The action of a group element $[\mathfrak{a}]$ on an elliptic curve E_0 is computed by $E_0/E_0[\mathfrak{a}]$, where $E_0[\mathfrak{a}]$ is a finite subgroup of E_0 derived from $[\mathfrak{a}]$; therefore, we can obtain a commutative diagram of isogenies by considering the group action as follows:

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_1} & [\mathfrak{a}]E_0 = E_0/E_0[\mathfrak{a}] \\ {\scriptstyle\phi_2}\downarrow & & \downarrow{\scriptstyle\phi_2'} \\ [\mathfrak{b}]E_0 = E_0/E_0[\mathfrak{b}] & \overset{}{\dashrightarrow}_{\phi_1'} & [\mathfrak{a}][\mathfrak{b}]E_0 \end{array}$$

The isogenies ϕ_1 and ϕ_1' correspond to the same element of the ideal class group; hence, anyone who knows ϕ_1 and E_2 can compute ϕ_1' by considering the group action on E_2 of the element related to ϕ_1.

4 LIT-SiGamal

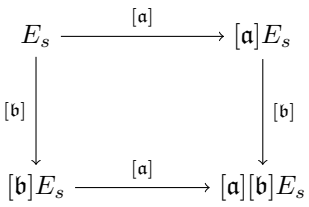
In this section, we provide the precise construction of LIT-SiGamal.

4.1 Overview

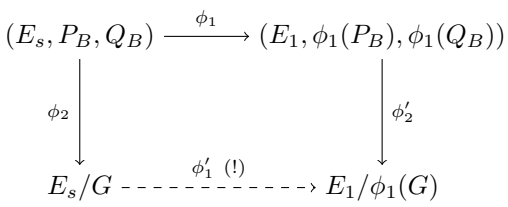
We provide a brief explanation of LIT-SiGamal in this subsection.

LIT-SiGamal is a SiGamal-based public key encryption scheme based on a LIT diagram, while the original SiGamal is based on a CSIDH diagram. The following diagrams show CSIDH and LIT diagrams.

A CSIDH diagram (SiGamal)



A LIT diagram (LIT-SiGamal)



The use of a LIT diagram instead of a CSIDH diagram offers the advantage of reducing the size of the prime p and making the scheme more efficient. In the setting of the LIT problem, anyone can compute two parallel vertical isogenies without revealing the isogeny ϕ_1 as the CSIDH setting; therefore, it seems to be able to construct a SiGamal-based PKE by using a LIT diagram. However, we generally cannot compute the bottom isogeny ϕ_1' of the LIT diagram from the top isogeny ϕ_1 without revealing ϕ_2 and ϕ_2' because the diagram does not rely on a group structure. This computation is needed for the decryption process of SiGamal. Therefore, it is not trivial to construct a SiGamal-based PKE scheme from a LIT diagram.

To construct LIT-SiGamal, we use the technique used in FESTA and IS-CUBE, which is to add auxiliary points masked by a matrix to the diagram. To be more precise, Alice and Bob perform the following steps:

1. Let N be a sufficiently large integer. Alice takes points P_A, Q_A that form a basis of $E_s[N]$ and a matrix $\mathbf{A}$ in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$.
2. Alice publishes P_A, Q_A and ${}^t(P_1, Q_1) := \mathbf{A} \cdot {}^t(\phi_1(P_A), \phi_1(Q_A))$ in addition to the top of the LIT diagram.
3. Bob computes $G = \langle P_B + sQ_B \rangle$ and $\phi_1(G) = \langle \phi_1(P_B) + s\phi_1(Q_B) \rangle$, and the vertical isogenies ϕ_2 and ϕ_2'.
4. Bob takes a matrix $\mathbf{B}$ in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ and computes

$$ {}^t(P_2, Q_2) := \mathbf{B} \cdot {}^t(\phi_2(P_A), \phi_2(Q_A)), \quad {}^t(P_3, Q_3) := \mathbf{B} \cdot {}^t(\phi_2'(P_1), \phi_2'(Q_1)). $$

   He publishes these points with E_s/G and $E_1/\phi_1(G)$.
5. Suppose that $\mathbf{AB} = \mathbf{BA}$. Note that it holds that

$$ {}^t(P_3, Q_3) = \mathbf{A} \cdot {}^t(\phi_1'(P_2), \phi_1'(Q_2)). $$

   Alice obtains the images of P_2, Q_2 under ϕ_1' by using $\mathbf{A}^{-1}$. Finally, by using the SIDH attacks, Alice computes ϕ_1'.

From the above steps, Alice can compute ϕ_1' without knowing the vertical isogenies ϕ_2 and ϕ_2'.

We use the SIDH attacks in the final step of the above computations. We use isogenies of abelian varieties of dimension 4 or 8 in general cases [24]; however, it is inefficient to compute such high-dimensional isogenies in practice. To use

isogenies of dimension 2 in the decryption process of LIT-SiGamal, we use the following technique. Let n be a small positive integer, and let ϕ_1 be an isogeny of degree N^2-n^2, where N is the order of P_A and Q_A. Then, we have the following isogeny diamond:

$$\begin{array}{ccc} E_s/G & \xrightarrow{\phi_1'} & E_1/\phi_1(G) \\ {\scriptstyle [n]}\downarrow & & \downarrow{\scriptstyle [n]} \\ E_s/G & \xrightarrow{\phi_1'} & E_1/\phi_1(G) \end{array}$$

We can use the same trick appearing in [24] for the above diagram via isogenies of dimension 2. To be precise, we construct two isogenies Ψ_0 and Ψ_1 of dimension 2 mapping from $E_s/G \times E_1/\phi_1(G)$ to an abelian variety V with

$$\ker \Psi_0 = \langle (nP_2, P_3'), (nQ_2, Q_3') \rangle, \quad \text{and} \quad \ker \Psi_1 = \langle (nP_2, -P_3'), (nQ_2, -Q_3') \rangle,$$

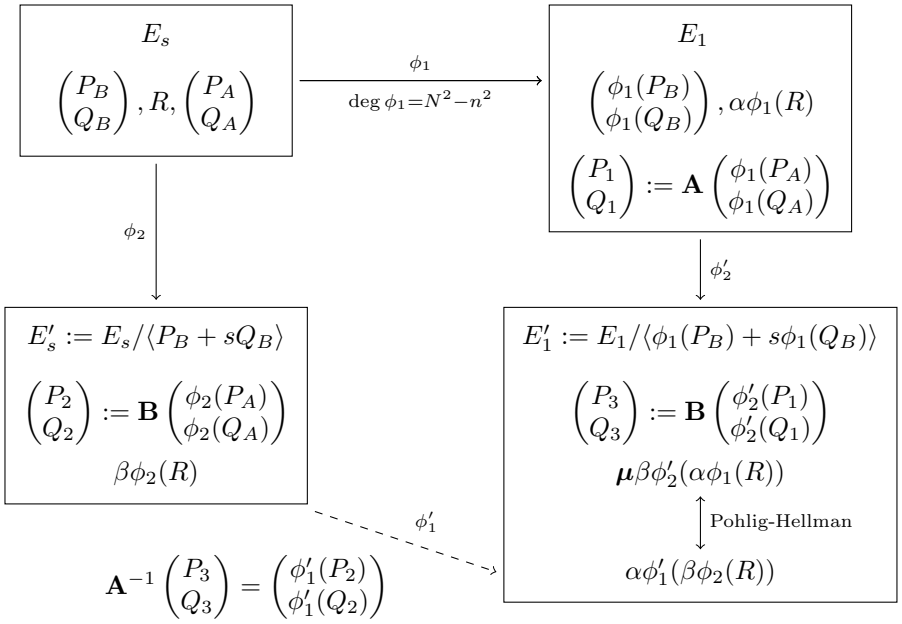
where ${}^t(P_3', Q_3') = \mathbf{A}^{-1} \cdot {}^t(P_3, Q_3)$. It holds that

$$\hat{\Psi}_1 \circ \Psi_0 = \begin{pmatrix} [n] & \hat{\phi}_1' \\ -\phi_1' & [n] \end{pmatrix};$$

therefore, Alice can compute ϕ_1' by using Ψ_0 and Ψ_1.

In summary, the following figure shows the outline of LIT-SiGamal:

Public key: $(E_s, P_A, Q_A, P_B, Q_B, R), (E_1, P_1, Q_1, \phi_1(P_B), \phi_1(Q_B), \alpha\phi_1(R))$
Secret key: $(\mathbf{A}, \alpha)$
Plaintext: μ
Ciphertext: $(E_s', P_2, Q_2, \beta\phi_2(R)), (E_1', P_3, Q_3, \mu\beta\phi_2'(\alpha\phi_1(R)))$



4.2 Scheme of LIT-SiGamal

In this subsection, we explain the PKE scheme of LIT-SiGamal. We assume that Bob tries to send a secret message μ to Alice.

Public parameters: Let p be a prime defined by $p = \ell_A^a \ell_B^b \ell_C^c \cdot f - 1$, where ℓ_A, ℓ_B, ℓ_C are pairwise coprime integers and f is a small integer. Denote by $\mathcal{M}_a$ the subgroup consisting of diagonal matrices in the general linear group of degree 2 over $\mathbb{Z}/\ell_A^a\mathbb{Z}$.[1] Let the plaintext space $\mathcal{P}$ be the unit group of $\mathbb{Z}/\ell_C^c\mathbb{Z}$.

Public key / Secret key: Alice first constructs a pair of supersingular elliptic curves over $\mathbb{F}_{p^2}$ denoted by (E_s, E_1) such that there is a cyclic isogeny $\phi_1 \colon E_s \to E_1$ of degree $\ell_A^{2a} - n^2$ for some integer n. Let $\{P_A, Q_A\}$ be a basis of $E_s[\ell_A^a]$, and let $\{P_B, Q_B\}$ be a basis of $E_s[\ell_B^b]$. Let R be a random point in E_s of order ℓ_C^c. Take a random matrix $\mathbf{A} \in \mathcal{M}_a$ and a random element $\alpha \in (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times$. Compute ${}^t(P_1, Q_1) := \mathbf{A} \cdot {}^t(\phi_1(P_A), \phi_1(Q_A))$ and $R_1 := \alpha\phi(R)$. Alice publishes

$$(E_s, P_A, Q_A, P_B, Q_B, R), (E_1, P_1, Q_1, \phi(P_B), \phi(Q_B), R_1)$$

as her public key. Alice keeps $(\mathbf{A}, \alpha)$ as her secret key.

Encryption: Bob takes the plaintext μ from the plaintext space $(\mathbb{Z}/\ell_C^c\mathbb{Z})^\times$. Bob takes random elements $s \in (\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$, $\mathbf{B} \in \mathcal{M}_a$, and $\beta \in (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times$. He next computes two ℓ_B-isogenies:

$$\begin{aligned} \phi_2 \colon E_s &\longrightarrow E_s' := E_s/\langle P_B + sQ_B \rangle, \\ \phi_2' \colon E_1 &\longrightarrow E_1' := E_1/\langle \phi(P_B) + s\phi(Q_B) \rangle \end{aligned}$$

and points

$$\begin{gathered} {}^t(P_2, Q_2) := \mathbf{B} \cdot {}^t(\phi_2(P_A), \phi_2(Q_A)), \quad {}^t(P_3, Q_3) := \mathbf{B} \cdot {}^t(\phi_2'(P_1), \phi_2'(Q_1)), \\ R' := \beta\phi_2(R), \quad R_1' := \mu\beta\phi_2'(R_1). \end{gathered}$$

Bob sends to Alice

$$(E_s', P_2, Q_2, R'), (E_1', P_3, Q_3, R_1')$$

as a ciphertext.

Decryption: Alice computes ${}^t(P_2', Q_2') := \mathbf{A} \cdot {}^t(P_2, Q_2)$ and $R'' := \alpha R'$. She next computes two isogenies

$$\begin{aligned} \Psi_0 \colon E_s' \times E_1' &\longrightarrow V \quad \text{with} \quad \ker \Psi_0 = \langle (nP_2', P_3), (nQ_2', Q_3) \rangle, \\ \Psi_1 \colon E_s' \times E_1' &\longrightarrow V \quad \text{with} \quad \ker \Psi_1 = \langle (nP_2', -P_3), (nQ_2', -Q_3)) \rangle, \end{aligned}$$

where V is an abelian variety of dimension 2. Put $R_1'' = \mathsf{pr}_2 \circ \hat{\Psi}_1 \circ \Psi_0((-R'', 0))$, where pr_2 is the projection $E_s' \times E_1' \to E_1'$. By solving the Discrete Logarithm Problem for R_1' and R_1'', Alice obtains the value μ' such that $R_1' = \mu' R_1''$. Output μ' as the plaintext.

[1] We can also use circulant matrices for the construction of LIT-SiGamal; however, we choose to use diagonal matrices to simplify the implementation and its security analysis.

Theorem 1. *LIT-SiGamal is correct.*

Proof. Let ϕ'_1 be an isogeny of degree $\ell_A^{2a} - n^2$ from E'_s to E'_1 satisfying $\phi'_1 \circ \phi_2 = \phi'_2 \circ \phi_1$. The kernel of an isogeny $\Phi\colon E'_s \times E'_1 \to E'_s \times E'_1$ represented by

$$\Phi = \begin{pmatrix} n & \hat{\phi'_1} \\ -\phi'_1 & n \end{pmatrix}$$

is $\langle (nP, \phi'_1(P)) \mid P \in E'_s[\ell_A^{2a}] \rangle$. Since the kernel of $\hat{\Phi}$ is

$$\Phi(\langle (nP, 0) \mid P \in E'_s[\ell_A^{2a}] \rangle) = \langle (nP, -\phi'_1(P)) \mid P \in E'_s[\ell_A^{2a}] \rangle,$$

we have $\Phi = \hat{\Psi_1} \circ \Psi_0$. Therefore, the point R''_1 is an image of $R'' = \alpha\beta\phi_0(R)$ under ϕ'. Since it holds that $R'_1 = \mu\alpha\beta\phi_1 \circ \phi(R)$, it is clear that $\mu = \mu'$. □

Remark 1. To reduce the computational cost of the scheme, we often represent points in elliptic curves by their x-coordinates. In this case, we cannot distinguish R'_1 and $-R'_1$ because we forget their y-coordinates. Therefore, at the end of the decryption process of LIT-SiGamal, Alice may obtain $-\mu$ instead of μ. We do not care about this error in practice since this error is easily corrected.

4.3 Construction of the public key of LIT-SiGamal

In general, it is not easy to construct the public key of LIT-SiGamal. It is because we need to compute an isogeny of degree $\ell_A^{2a} - n^2$ that is generally not smooth. This subsection introduces the method to generate the public key of LIT-SiGamal. The above problem also occurs in generating the public parameters of IS-CUBE; therefore, we can use similar techniques to that for generating the public parameters of IS-CUBE appearing in [18, Section 3.2 and 3.3]. I.e., we compute a desired isogeny by using the structure of the endomorphism ring of the curve of j-invariant 1728 over $\mathbb{F}_{p^2}$.

From the discussion in Section 5.2, we have $p \approx 2^{6\lambda}$, $\ell_A^a \approx 2^{3\lambda}$, and $\ell_B^b \approx 2^{2\lambda}$; therefore, we can assume that $p < (\ell_A^{2a} - n^2)\ell_B^b$. Then, we can compute four integers x, y, z, w satisfying

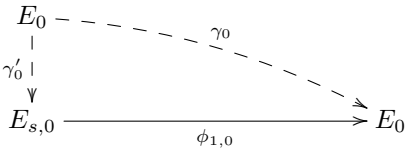
$$x^2 + y^2 + p(z^2 + w^2) = (\ell_A^{2a} - n^2)\ell_B^b$$

by using the Cornacchia algorithm (see [11, Algorithm 1] for more details). Let E_0 be the curve of j-invariant 1728, let π_p be the p-Frobenius map of E_0, and let ι be an endomorphism of E_0 such that $\iota^2 = [-1]$. Define $\gamma_0\colon E_0 \to E_0$ by

$$\gamma_0 := [x] + [y]\iota + \pi_p([z] + [w]\iota).$$

Then, it holds that $\deg \gamma_0 = (\ell_A^{2a} - n^2)\ell_B^b$. Let γ'_0 be a separable isogeny mapping from E_0 with $\ker \gamma'_0 = \ker \gamma_0 \cap E_0[\ell_B^b]$, and let $E_{s,0}$ be the codomain of γ'_0. There

is an isogeny $\phi_{1,0}\colon E_{s,0} \to E_0$ of degree $\ell_A^{2a} - n^2$ such that $\gamma_0 = \phi_{1,0} \circ \gamma_0'$ as the following diagram shows.



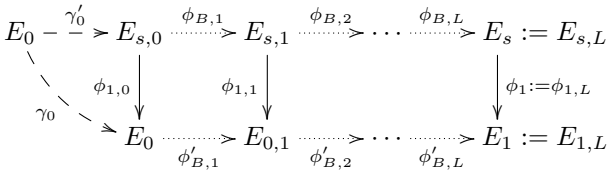
We need to compute the images of points of order ℓ_A^a, ℓ_B^b, and ℓ_C^c under $\phi_{1,0}$ for the public key of LIT-SiGamal. Let P be a point in $E_{s,0}$ of order coprime to ℓ_B. In this case, we can compute $\phi_{1,0}(P)$ easily because it holds that $\gamma_0 \circ \hat{\gamma_0'} = [\ell_B^b] \circ \phi_{1,0}$. In particular, we can compute the images of points of order ℓ_A^a and ℓ_C^c. If P is of order divided by ℓ_B, we can compute $\phi_{1,0}(P)$ by using the same method as the decryption. Note that we can compute $\{\phi_{1,0}(P_A), \phi_{1,0}(Q_A)\}$, where $\{P_A, Q_A\}$ is a basis of $E_{s,0}[\ell_A^a]$. Therefore, we can compute the image point under $\phi_{1,0}$ of a point of order divided by ℓ_B from $\{P_A, Q_A\}$ and $\{\phi_{1,0}(P_A), \phi_{1,0}(Q_A)\}$ and the SIDH attacks.

From the above method, we can construct the public key of LIT-SiGamal; however, this construction has security concerns. One of the concerns is that we use the curve of j-invariant 1728. Castryck and Vercauteren showed in [8] that FESTA using the curve of j-invariant 1728 is broken if the public points satisfy the special property. Another security concern is the distribution of $E_{s,0}$. The number of γ_0 is less than $2^{2\lambda}$ because we have

$$\begin{aligned} &\#\{(x,y,z,w) \in \mathbb{Z}^4 \mid x^2 + y^2 + p(z^2 + w^2) = \ell_B^b(\ell_A^{2a} - n^2)\} \\ &\leq \#\{(z,w) \in \mathbb{Z}^2 \mid z^2 + w^2 \leq \ell_B^b(\ell_A^{2a} - n^2)/p\} \\ &< \left(2\sqrt{\frac{\ell_B^b(\ell_A^{2a} - n^2)}{p}}\right)^2 \approx 2^{2\lambda}. \end{aligned}$$

Since the number of isogenies of degree $\ell_A^{2a} - n^2$ from E_0 is about $2^{6\lambda}$, we take $E_{s,0}$ from a tiny subset of the set of possible elliptic curves. Therefore, it is recommended to use random elliptic curves for the public key.

We now explain the method to construct the public key with random elliptic curves. The idea is the same as in [18, Section 3.3] and shown in the following diagram.



Precisely, we perform the following procedure:

Step 0: By computing γ_0 and γ_0', construct $E_{s,0}$, $\phi_{1,0}$, and the auxiliary points $P_{A,0}, Q_{A,0}$, $P_{B,0}, Q_{B,0}$, $R_{C,0}$, $\phi_{1,0}(P_{A,0}), \phi_{1,0}(Q_{A,0})$, $\phi_{1,0}(P_{B,0}), \phi_{1,0}(Q_{B,0})$, and $\phi_{1,0}(R_{C,0})$, where $\{P_{A,0}, Q_{A,0}\}$ is a basis of $E_{s,0}[\ell_A^a]$, $\{P_{B,0}, Q_{B,0}\}$ is a basis of $E_{s,0}[\ell_B^b]$, and $R_{C,0}$ is of order ℓ_C^c. Let $E_{0,0}$ be E_0 (the curve of j-invariant 1728).

Step $3i-2$ ($1 \leq i \leq L$): Compute two ℓ_B^b-isogenies $\phi_{B,i}$ and $\phi'_{B,i}$ from $E_{s,i}$ and $E_{0,i}$ satisfying $\ker \phi'_{B,i} = \phi_{1,i}(\ker \phi_{B,i})$ at random without backtracking. Denote the codomain of $\phi_{B,i}$ (resp. $\phi'_{B,i}$) by $E_{s,i}$ (resp. by $E_{0,i}$). Notice that there is an isogeny $\phi_{1,i}$ of degree $\ell_A^{2a} - n^2$ between $E_{s,i}$ and $E_{0,i}$.

Step $3i-1$ ($1 \leq i \leq L$): Compute the images of $P_{A,i-1}, Q_{A,i-1}, R_{C,i-1}$ under $\phi_{B,i}$. Put

$$P_{A,i} := \phi_{B,i}(P_{A,i-1}), \quad Q_{A,i} := \phi_{B,i}(Q_{A,i-1}), \quad R_{C,i} := \phi_{B,i}(R_{C,i-1}).$$

Compute $\phi_{1,i}(P_{A,i})$, $\phi_{1,i}(Q_{A,i})$, and $\phi_{1,i}(R_{C,i})$ by computing the images of $\phi_{1,i-1}(P_{A,i-1})$, $\phi_{1,i-1}(Q_{A,i-1})$, and $\phi_{1,i-1}(R_{C,i-1})$ under $\phi'_{B,i}$.

Step $3i$ ($1 \leq i \leq L$): Generate a random basis $\{P_{B,i}, Q_{B,i}\}$ of $E_{s,i}[\ell_B^b]$ and compute the images of $P_{B,i}, Q_{B,i}$ under $\phi_{B,i}$ as the decryption process of LIT-SiGamal using $\{P_{A,i}, Q_{A,i}\}$ and $\{\phi_{1,i}(P_{A,i}), \phi_{1,i}(Q_{A,i})\}$.

Step $3L+1$: Mask the auxiliary points appropriately and output $E_{s,L}$, $E_{0,L}$ and the masked auxiliary points as the public key.

Here, the integer L is the number of the iteration.

Remark 2. It seems that we cannot take two random elliptic curves using the above method because it takes one random elliptic curve and another curve depending on the first curve. However, [2, Theorem 7.3] and [3, Theorem 3] showed that the supersingular ℓ-isogeny graph with a level-$(\ell_A^{2a} - n^2)$ Borel structure is connected and satisfies a Ramanujan property. Therefore, we can obtain a random supersingular elliptic curve and its random cyclic subgroup of order $(\ell_A^{2a} - n^2)$ by a random walk on the graph. That is, if L is sufficiently large, then we can construct a random pair of $(\ell_A^{2a} - n^2)$-isogenous supersingular elliptic curves by the above method. The mixing rate of the graph is $1/\sqrt{\ell_B}$ from [1, Theorem 1.1]. I.e., it holds that

$$\frac{1}{\sqrt{\ell_B}} = \limsup_{l \to \infty} \max_{u,v \in G} \left| P_{u,v}^{(l)} - \frac{1}{\#G} \right|^{\frac{1}{l}},$$

where G is the set of the vertices of the graph, and $P_{u,v}^{(l)}$ is a probability that an l-length non-backtracking random walk in the graph from u reaches v. Therefore, we suggest the number L to satisfy $(1/\sqrt{\ell_B})^{bL} \leq 1/2^\lambda$ and $\ell_B^{bL} \geq \#G \approx 2^{12\lambda}$. That is, we set $L \approx (\log_{\ell_B} 2^{12\lambda})/b = 12\lambda / \log_2 \ell_B^b \approx 6$.

4.4 Compressed LIT-SiGamal

In this subsection, we introduce the compressed version of LIT-SiGamal. As the same as other isogeny-based schemes, the main idea of the compression is based

on the compression of SIDH [9]. That is, we define a canonical basis of an N-torsion subgroup of E and represent a basis of $E[N]$ by a matrix in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ using the Pohlig-Hellman algorithm.

We explain the detail of the compression. Recall that the public key of LIT-SiGamal is

$$(E_s, P_A, Q_A, P_B, Q_B, R), (E_1, P_1, Q_1, \phi(P_B), \phi(Q_B), R_1),$$

which is represented as an element in $\mathbb{F}_{p^2}^{12}$. We assume that there is a deterministic algorithm $\mathcal{A}$ that outputs a basis of $E[N]$ from an elliptic curve E and an integer N. We define

$$\{P_A, Q_A\} := \mathcal{A}(E_s, \ell_A^a), \quad \{P_B, Q_B\} := \mathcal{A}(E_s, \ell_B^b).$$

Then, we do not need to include $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ in the public key of LIT-SiGamal. Let $\{P_C, Q_C\}$ be a basis of $E_s[\ell_C^c]$ that is output of $\mathcal{A}(E_s, \ell_C^c)$. Then, we can define $R := P_C + rQ_C$, and R can be represented by $r \in \mathbb{Z}/\ell_C^c\mathbb{Z}$. We define

$$\{P_{A,1}, Q_{A,1}\} := \mathcal{A}(E_1, \ell_A^a), \quad \{P_{B,1}, Q_{B,1}\} := \mathcal{A}(E_1, \ell_B^b).$$

Then, we can represent $\{P_1, Q_1\}$ and $\{\phi(P_B), \phi(Q_B)\}$ by matrices as

$$\begin{pmatrix} P_1 \\ Q_1 \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \begin{pmatrix} P_{A,1} \\ Q_{A,1} \end{pmatrix}, \quad \begin{pmatrix} \phi(P_B) \\ \phi(Q_B) \end{pmatrix} = \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix} \begin{pmatrix} P_{B,1} \\ Q_{B,1} \end{pmatrix}.$$

Note that we can ignore the constant factors of the representations. For example, we can represent $\{P_1, Q_1\}$ by $(1, a_{01}/a_{00}, a_{10}/a_{00}, a_{11}/a_{00})$ if $a_{00} \not\equiv 0 \pmod{\ell_A}$ and by $(a_{00}/a_{10}, 1, a_{10}/a_{01}, a_{11}/a_{01})$ if $a_{00} \equiv 0 \pmod{\ell_A}$. Therefore, we can represent $\{P_1, Q_1\}$ by an element in $\{0,1\} \times (\mathbb{Z}/\ell_A^a\mathbb{Z})^3$ and $\{\phi(P_B), \phi(Q_B)\}$ by that in $\{0,1\} \times (\mathbb{Z}/\ell_B^b\mathbb{Z})^3$. We define

$$\{P_{C,1}, Q_{C,1}\} := \mathcal{A}(E_1, \ell_C^c).$$

Then, we can represent R_1 by $(r_1, r_1') \in (\mathbb{Z}/\ell_C^c\mathbb{Z})^2$ with $R_1 = r_1 P_{C,1} + r_1' Q_{C,1}$. Because we can also ignore the constant factor in this case, the point R_1 can be represented by an element in $\{0,1\} \times \mathbb{Z}/\ell_C^c\mathbb{Z}$ as $\{P_1, Q_1\}$. Therefore, the public key of LIT-SiGamal can be represented by an element in

$$\mathbb{F}_{p^2}^2 \times (\mathbb{Z}/\ell_A^a\mathbb{Z})^3 \times (\mathbb{Z}/\ell_B^b\mathbb{Z})^3 \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^2 \times \{0,1\}^3.$$

The ciphertext of LIT-SiGamal can also be compressed in a similar way. Recall that the ciphertext of LIT-SiGamal is

$$(E_s', P_2, Q_2, R'), (E_1', P_3, Q_3, R_1'),$$

which is represented by an element in $\mathbb{F}_{p^2}^8$. The basis $\{P_2, Q_2\}$ can be represented by an element in $\{0,1\} \times (\mathbb{Z}/\ell_A^a\mathbb{Z})^3$. However, it is not clear that $\{P_3, Q_3\}$ can be represented as the same as $\{P_2, Q_2\}$ because the constant factors of $\{P_2, Q_2\}$

	Original	Compressed
Public key	$\mathbb{F}_{p^2}^{12}$	$\mathbb{F}_{p^2}^{2} \times (\mathbb{Z}/\ell_A^a\mathbb{Z})^3 \times (\mathbb{Z}/\ell_B^b\mathbb{Z})^3 \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^2 \times \{0,1\}^3$
Cipertext	$\mathbb{F}_{p^2}^{8}$	$\mathbb{F}_{p^2}^{2} \times (\mathbb{Z}/\ell_A^a\mathbb{Z})^6 \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^3 \times \{0,1\}^3$

Table 1. Original and compressed LIT-SiGamal

and of $\{P_3, Q_3\}$ should be compatible. In other words, it is because it should hold that $\mathbf{A}^{-1} \cdot {}^t(P_3, Q_3) = {}^t(\phi_1'(P_2), \phi_1'(Q_2))$. Actually, we can solve this issue. If (P_3, Q_3) is multiplied by a constant, we can detect this constant by using the Weil pairing and $\deg \phi_1' = \ell_A^{2a} - n^2$. Consequently, both $\{P_2, Q_2\}$ and $\{P_3, Q_3\}$ can be represented by elements in $\{0,1\} \times (\mathbb{Z}/\ell_A^a\mathbb{Z})^3$. On the contrary, the points R' and R_1' cannot be represented by elements in $\{0,1\} \times \mathbb{Z}/\ell_C^c\mathbb{Z}$ at the same time because we cannot obtain information from the Weil pairing. Therefore, the points (R', R_1') cannot be represented by an element in $\{0,1\}^2 \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^2$ but in $\{0,1\} \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^3$. Consequently, the ciphertext of LIT-SiGamal can be represented by an element in

$$\mathbb{F}_{p^2}^{2} \times (\mathbb{Z}/\ell_A^a\mathbb{Z})^6 \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^3 \times \{0,1\}^3.$$

Table 1 summarizes the representation sets of the public key and ciphertext of the original and compressed LIT-SiGamal.

5 Security analysis of LIT-SiGamal

We provide security analysis of LIT-SiGamal in this section.

5.1 Security assumptions

In this subsection, we provide some security assumptions associated with LIT-SiGamal and prove that LIT-SiGamal is IND-CPA secure. In particular, we show that if the LIT-DDH assumption, which is analogous to the DDH assumption in a LIT diagram, holds, then LIT-SiGamal is IND-CPA secure.

Let p be a prime of the form $p = \ell_A^a \ell_B^b \ell_C^c f - 1$, where ℓ_A, ℓ_B, and ℓ_C are distinct small primes, and f is a small integer. Let n be a small integer. Let $\mathcal{M}_a$ be the subgroup of diagonal matrices in $\mathrm{GL}_2(\mathbb{Z}/\ell_A^a\mathbb{Z})$, and let $\mathcal{M}_{a,c}$ be the subgroup of diagonal matrices in $\mathrm{GL}_2(\mathbb{Z}/\ell_A^a\ell_C^c\mathbb{Z})$.

We first introduce the Computational and Decisional LIT-SiGamal assumptions (Definition 5 and 6), which are basic assumptions for considering the security of LIT-SiGamal.

Definition 5 (Computational LIT-SiGamal assumption)**.** Let E_s and E_1 be supersingular elliptic curves with an isogeny $\phi_1 \colon E_s \to E_1$ of degree $\ell_A^{2a} - n^2$. Let $\{P_A, Q_A\}$ be a basis of $E_s[\ell_A^a]$, let $\{P_B, Q_B\}$ be a basis of $E_s[\ell_B^b]$, and let R be a point of E_s of order ℓ_C^c. Define

$$\begin{pmatrix} P_{1,A} \\ Q_{1,A} \end{pmatrix} := \mathbf{A} \begin{pmatrix} \phi_1(P_A) \\ \phi_1(Q_A) \end{pmatrix}, \quad \begin{pmatrix} P_{1,B} \\ Q_{1,B} \end{pmatrix} := \begin{pmatrix} \phi_1(P_B) \\ \phi_1(Q_B) \end{pmatrix}, \quad R_1 := \alpha\phi_1(R),$$

where $\mathbf{A}$ is a matrix in $\mathcal{M}_a$ and α is an element in $(\mathbb{Z}/\ell_C^c\mathbb{Z})^\times$. Let s be an element in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$, let $\phi_2 \colon E_s \to E'_s$ be a separable isogeny with $\ker \phi_2 = \langle P_B + sQ_B \rangle$, and let $\phi'_2 \colon E_1 \to E'_1$ be a separable isogeny with $\ker \phi'_2 = \langle P_{1,B} + sQ_{1,B} \rangle$. Define

$$\begin{pmatrix} P'_A \\ Q'_A \end{pmatrix} := \mathbf{B} \begin{pmatrix} \phi_2(P_A) \\ \phi_2(Q_A) \end{pmatrix}, \quad \begin{pmatrix} P'_{1,A} \\ Q'_{1,A} \end{pmatrix} := \mathbf{B} \begin{pmatrix} \phi'_2(P_{1,A}) \\ \phi'_2(Q_{1,A}) \end{pmatrix}, \quad R' := \beta\phi_2(R), \quad R'_1 := \beta\phi'_2(R_1),$$

where $\mathbf{B}$ is a matrix in $\mathcal{M}_a$ and β is an element in $(\mathbb{Z}/\ell_C^c\mathbb{Z})^\times$. We denote the tuple

$$(E_s, P_A, Q_A, P_B, Q_B, R, E_1, P_{1,A}, Q_{1,A}, P_{B,1}, Q_{1,B}, R_1, E'_s, P'_A, Q'_A, R', E'_1, P'_{1,A}, Q'_{1,A}, R'_1)$$

by $T_{\text{LIT-SiGamal}}$ and denote the set of the tuples by $\mathcal{S}_{\text{LIT-SiGamal}}$. We also denote by $T_{\text{LIT-SiGamal}}(\mu)$ the tuple that is defined by replacing R'_1 in $T_{\text{LIT-SiGamal}}$ with $\mu R'_1$ for $\mu \in \mathbb{Z}/\ell_C^c\mathbb{Z}$.

We call the following assumption the *Computational LIT-SiGamal assumption*:

Any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ satisfies

$$\Pr\left[\mu = \mu^* \;\middle|\; \begin{array}{l} \mu \xleftarrow{\$} (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times,\ T_{\text{LIT-SiGamal}} \xleftarrow{\$} \mathcal{S}_{\text{LIT-SiGamal}}, \\ \mu^* \leftarrow \mathcal{A}\left(T_{\text{LIT-SiGamal}}(\mu)\right) \end{array} \right] < \text{negl}(\lambda),$$

where the notation $X \xleftarrow{\$} Y$ means that X is sampled uniformly at random from Y and $\text{negl}(\cdot)$ *is a negligible function.*

We have the following proposition immediately from the construction of LIT-SiGamal.

Proposition 2. *LIT-SiGamal is OW-CPA secure if the Computational LIT-SiGamal assumption holds.*

Definition 6 (Decisional LIT-SiGamal assumption)**.** We call the following assumption the *Decisional LIT-SiGamal assumption*:

Any PPT algorithm $\mathcal{A}$ satisfies

$$\left| \Pr\left[i = i^* \;\middle|\; \begin{array}{l} i \xleftarrow{\$} \{0,1\},\ \mu_0 = 1,\ \mu_1 \xleftarrow{\$} (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times, \\ T_{\text{LIT-SiGamal}} \xleftarrow{\$} \mathcal{S}_{\text{LIT-SiGamal}}, \\ i^* \leftarrow \mathcal{A}\left(T_{\text{LIT-SiGamal}}(\mu_i)\right) \end{array} \right] - \frac{1}{2} \right| < \text{negl}(\lambda).$$

We also have the following proposition.

Proposition 3. *LIT-SiGamal is IND-CPA secure if the Decisional LIT-SiGamal assumption holds.*

Proof. Suppose that LIT-SiGamal is not IND-CPA secure. That is, we suppose that there is a PPT algorithm $\mathcal{A}'$ satisfying the value

$$\varepsilon := \left| \Pr\left[i = i^* \;\middle|\; \begin{array}{l} T_{\text{LIT-SiGamal}} \xleftarrow{\$} \mathcal{S}_{\text{LIT-SiGamal}}, \\ i \xleftarrow{\$} \{0,1\},\ \mu'_0, \mu'_1 \leftarrow \mathcal{A}'(\mathbf{pk}), \\ i^* \leftarrow \mathcal{A}'(T_{\text{LIT-SiGamal}}(\mu'_i)) \end{array} \right] - \frac{1}{2} \right|$$

is not negligible, where $\mathbf{pk}$ is the tuple

$$(E_s, P_A, Q_A, P_B, Q_B, R, E_1, P_{1,A}, Q_{1,A}, P_{B,1}, Q_{1,B}, R_1).$$

We define a PPT algorithm $\mathcal{A}$ for an input $T_{\text{LIT-SiGamal}}(\mu)$ by the following procedure:

1. Define $\mathbf{pk}$ appropriately and compute $\mu'_0, \mu'_1 \leftarrow \mathcal{A}'(\mathbf{pk})$.
2. Take i from $\{0,1\}$ uniformly at random.
3. Compute $T_{\text{LIT-SiGamal}}(\mu'_i\mu)$.
4. Compute $i^* \leftarrow \mathcal{A}'(T_{\text{LIT-SiGamal}}(\mu'_i\mu))$.
5. If $i = i^*$, output 0, and if $i \neq i^*$ output 1.

If μ is a random element in $\mathbb{Z}/\ell_C^c\mathbb{Z}$, we cannot distinguish $T_{\text{LIT-SiGamal}}(\mu'_0\mu)$ and $T_{\text{LIT-SiGamal}}(\mu'_1\mu)$; therefore, we have, for a random $\mu \in \mathbb{Z}/\ell_C^c\mathbb{Z}$,

$$\Pr[1 \leftarrow \mathcal{A}(T_{\text{LIT-SiGamal}}(\mu))] = \frac{1}{2}.$$

Therefore, it holds that

$$\begin{aligned} &\Pr\left[i = i^* \;\middle|\; \begin{array}{l} i \xleftarrow{\$} \{0,1\},\ \mu_0 = 1,\ \mu_1 \xleftarrow{\$} (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times, \\ T_{\text{LIT-SiGamal}} \xleftarrow{\$} \mathcal{S}_{\text{LIT-SiGamal}}, \\ i^* \leftarrow \mathcal{A}(T_{\text{LIT-SiGamal}}(\mu_i)) \end{array} \right] \\ &= \frac{1}{2}\Pr[0 \leftarrow \mathcal{A}(T_{\text{LIT-SiGamal}})] + \frac{1}{2}\Pr[1 \leftarrow \mathcal{A}(T_{\text{LIT-SiGamal}}(\mu_1))] \\ &= \frac{1}{2}\left(\frac{1}{2}\Pr[0 \leftarrow \mathcal{A}'(T_{\text{LIT-SiGamal}}(\mu'_0))] + \frac{1}{2}\Pr[1 \leftarrow \mathcal{A}'(T_{\text{LIT-SiGamal}}(\mu'_1))]\right) + \frac{1}{4} \\ &= \frac{1}{2}\left(\frac{1}{2} \pm \varepsilon\right) + \frac{1}{4} = \frac{1}{2} \pm \frac{\varepsilon}{2}. \end{aligned}$$

Hence, the algorithm $\mathcal{A}$ distinguishes $T_{\text{LIT-SiGamal}}(\mu_0)$ and $T_{\text{LIT-SiGamal}}(\mu_1)$, and the Decisional LIT-SiGamal assumption does not hold. This completes the proof of Proposition 3. □

From Proposition 3, it is sufficient to analyze the correctness of the Decisional LIT-SiGamal assumption for analysis of the security of LIT-SiGamal. To analyze this assumption, we first define the LIT-DDH assumption, which is analogous to the DDH assumption in a LIT diagram.

Definition 7 (LIT-DDH assumption)**.** Let E_s and E_1 be random supersingular elliptic curves with an isogeny $\phi_1 \colon E_s \to E_1$ of degree $\ell_A^{2a} - n^2$. Let $\{P_B, Q_B\}$ be a random basis of $E_s[\ell_B^b]$, and let $\{P_{A,C}, Q_{A,C}\}$ be a random basis of $E_s[\ell_A^a \ell_C^c]$. Let $\mathbf{A}$ and $\mathbf{B}$ be random matrices in $\mathcal{M}_{a,c}$. Define

$$\begin{pmatrix} P_{1,A,C} \\ Q_{1,A,C} \end{pmatrix} := \mathbf{A} \begin{pmatrix} \phi_1(P_{A,C}) \\ \phi_1(Q_{A,C}) \end{pmatrix}, \quad \begin{pmatrix} P_{1,B} \\ Q_{1,B} \end{pmatrix} := \begin{pmatrix} \phi_1(P_B) \\ \phi_1(Q_B) \end{pmatrix}.$$

Let s be a random element in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$, let $\phi_2 \colon E_s \to E'_s$ be a separable isogeny with $\ker \phi_2 = \langle P_B + sQ_B \rangle$, and let $\phi'_2 \colon E_1 \to E'_1$ be a separable isogeny with $\ker \phi'_2 = \langle P_{1,B} + sQ_{1,B} \rangle$. Define

$$\begin{pmatrix} P'_{A,C} \\ Q'_{A,C} \end{pmatrix} := \mathbf{B} \begin{pmatrix} \phi_2(P_{A,C}) \\ \phi_2(Q_{A,C}) \end{pmatrix}, \quad \begin{pmatrix} P'_{1,A,C} \\ Q'_{1,A,C} \end{pmatrix} := \mathbf{B} \begin{pmatrix} \phi'_2(P_{1,A,C}) \\ \phi'_2(Q_{1,A,C}) \end{pmatrix}.$$

We denote by $T_{\text{LIT-DDH},0}$ the tuple

$$(E_s, P_{A,C}, Q_{A,C}, P_B, Q_B, E_1, P_{1,A,C}, Q_{1,A,C}, P_{1,B}, Q_{1,B}, E'_s, P'_{A,C}, Q'_{A,C}, E'_1, P'_{1,A,C}, Q'_{1,A,C}),$$

and denote by $\mathcal{S}_{\text{LIT-DDH}}$ the set of the above tuples. Let E_r be a random supersingular elliptic curve, and let $\{P_r, Q_r\}$ be a random basis of $E_r[\ell_A^a \ell_C^c]$ satisfying $e(P,Q)^{\ell_B^b} = e(P_{1,A,C}, Q_{1,A,C})$. We denote by $T_{\text{LIT-DDH},1}$ the tuple

$$(E_s, P_{A,C}, Q_{A,C}, P_B, Q_B, E_1, P_{1,A,C}, Q_{1,A,C}, P_{1,B}, Q_{1,B}, E'_s, P'_{A,C}, Q'_{A,C}, E_r, P_r, Q_r),$$

and denote by $\mathcal{S}_{\text{LIT-DDH}'}$ the set of the above tuples.

We call the following assumption the *LIT-DDH assumption*:

Any PPT algorithm $\mathcal{A}$ satisfies

$$\left| \Pr \left[i = i^* \;\middle|\; \begin{array}{l} T_{\text{LIT-DDH},0} \xleftarrow{\$} \mathcal{S}_{\text{LIT-DDH}}, \\ T_{\text{LIT-DDH},1} \xleftarrow{\$} \mathcal{S}_{\text{LIT-DDH}'}, \\ i \xleftarrow{\$} \{0,1\},\ i^* \leftarrow \mathcal{A}(T_{\text{LIT-DDH},i}) \end{array} \right] - \frac{1}{2} \right| < \text{negl}(\lambda).$$

Then, we have the following theorem.

Theorem 2. *If the LIT-DDH assumption holds, then the Decisional LIT-SiGamal assumption also holds.*

Proof. Suppose that the Decisional LIT-SiGamal assumption does not hold. That is, there is a PPT algorithm $\mathcal{A}'$ satisfying the value

$$\varepsilon := \left| \Pr \left[i = i^* \;\middle|\; \begin{array}{l} i \xleftarrow{\$} \{0,1\},\ \mu_0 = 1,\ \mu_1 \xleftarrow{\$} (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times, \\ T_{\text{LIT-SiGamal}} \xleftarrow{\$} \mathcal{S}_{\text{LIT-SiGamal}}, \\ i^* \leftarrow \mathcal{A}'\left(T_{\text{LIT-SiGamal}}(\mu_i)\right) \end{array} \right] - \frac{1}{2} \right|$$

is not negligible.

Let the following tuple T be a tuple in $\mathcal{S}_{\text{LIT-DDH}} \cup \mathcal{S}_{\text{LIT-DDH}'}$:

$$(E_s, P_{A,C}, Q_{A,C}, P_B, Q_B, E_1, P_{1,A,C}, Q_{1,A,C}, P_{1,B}, Q_{1,B}, E'_s, P'_{A,C}, Q'_{A,C}, E_2, P_2, Q_2).$$

From the Chinese remainder theorem, we have an isomorphism between $E[\ell_A^a \ell_C^c]$ and $E[\ell_A^a] \times E[\ell_C^c]$ for any supersingular elliptic curve E. By using these isomorphisms, a basis of $E[\ell_A^a \ell_C^c]$ is mapped to bases of $E[\ell_A^a]$ and $E[\ell_C^c]$. Additionally, by removing one point of order ℓ_C^c, a basis of $E[\ell_A^a \ell_C^c]$ is mapped to a basis of $E[\ell_A^a]$ and a point of order ℓ_C^c. Therefore, we have a transformation from the tuple T to a tuple

$$(E_s, P_A, Q_A, P_B, Q_B, R, E_1, P_{1,A}, Q_{1,A}, P_{B,1}, Q_{1,B}, R_1, E'_s, P'_A, Q'_A, R', E_2, P_{2,A}, Q_{2,A}, R_2),$$

where $\{P_A, Q_A\}$ (resp. $\{P_{1,A}, Q_{1,A}\}$, $\{P'_A, Q'_A\}$, $\{P_{2,A}, Q_{2,A}\}$) is a basis of $E_s[\ell_A^a]$ (resp. $E_1[\ell_A^a]$, $E'_s[\ell_A^a]$, $E_2[\ell_A^a]$), and R (resp. R_1, R', R_2) is a point in E_s (resp. E_1, E'_s, E_2) of order ℓ_C^c derived from $P_{A,C}$ (resp. $P_{1,A,C}$, $P'_{A,C}$, P_2). We denote this map by $F_{\text{DDH}\to\text{SiGamal}}$.

We define a PPT algorithm $\mathcal{A}$ with an input T by the following procedure:

1. Take i from $\{0,1\}$ uniformly at random.
2. Set $\mu_0 = 1$ and take μ_1 from $(\mathbb{Z}/\ell_C^c\mathbb{Z})^\times$ uniformly at random.
3. Compute $T' = F_{\text{DDH}\to\text{SiGamal}}(T)$.
4. Compute $i^* \leftarrow \mathcal{A}'(T'(\mu_i))$.
5. if $i = i^*$, output 0, and if $i \neq i^*$, output 1.

Note that if $T \in \mathcal{S}_{\text{LIT-DDH}}$, then $T' \in \mathcal{S}_{\text{LIT-SiGamal}}$. Moreover, if $T \in \mathcal{S}_{\text{LIT-DDH}'}$, then

$$\Pr[1 \leftarrow \mathcal{A}(T)] = \frac{1}{2}$$

because we cannot distinguish $T'(\mu_0)$ and $T'(\mu_1)$. Therefore, it holds that

$$\begin{aligned}
&\Pr\left[i = i^* \;\middle|\; \begin{array}{l} T_{\text{LIT-DDH},0} \xleftarrow{\$} \mathcal{S}_{\text{LIT-DDH}}, \\ T_{\text{LIT-DDH},1} \xleftarrow{\$} \mathcal{S}_{\text{LIT-DDH}'}, \\ i \xleftarrow{\$} \{0,1\},\ i^* \leftarrow \mathcal{A}(T_{\text{LIT-DDH},i}) \end{array} \right] \\
&= \frac{1}{2}\Pr[0 \leftarrow \mathcal{A}(T_{\text{LIT-DDH},0})] + \frac{1}{2}\Pr[1 \leftarrow \mathcal{A}(T_{\text{LIT-DDH},1})] \\
&= \frac{1}{2}\left(\frac{1}{2}\Pr[0 \leftarrow \mathcal{A}'(T_{\text{LIT-SiGamal}})] + \frac{1}{2}\Pr[1 \leftarrow \mathcal{A}'(T_{\text{LIT-SiGamal}}(\mu_1))]\right) + \frac{1}{4} \\
&= \frac{1}{2}\left(\frac{1}{2} \pm \varepsilon\right) + \frac{1}{4} = \frac{1}{2} + \frac{\varepsilon}{2}.
\end{aligned}$$

Hence, the algorithm $\mathcal{A}$ can distinguish $T_{\text{LIT-DDH},0}$ and $T_{\text{LIT-DDH},1}$, and the LIT-DDH assumption does not hold. This completes the proof of Theorem 2. □

From Proposition 3 and Theorem 2, we immediately have Corollary 1.

Corollary 1. *If the LIT-DDH assumption holds, LIT-SiGamal is IND-CPA secure.*

Remark 3. LIT-SiGamal is not IND-CCA secure for the same reason as the original SiGamal (see [20, Section 3.3]). Namely, it is because, from the ciphertext

$$\mathbf{ct} = (E'_s, P_2, Q_2, R', E'_1, P_3, Q_3, R'_1),$$

we can compute a different ciphertext

$$(E'_s, P_2, Q_2, R', E'_1, P_3, Q_3, \mu' R'_1)$$

whose plaintext is μ' times the plaintext of $\mathbf{ct}$. Therefore, an appropriate transformation is required to use LIT-SiGamal in practice (*e.g.,* the Fujisaki-Okamoto transform [14]).

5.2 Secure prime of LIT-SiGamal

In this subsection, we discuss the appropriate size of the prime p. Note that p is represented by $p = \ell_A^a \cdot \ell_B^b \cdot \ell_C^c \cdot f - 1$. Since f is a small integer, it is sufficient to estimate the size of ℓ_A^a, ℓ_B^b, and ℓ_C^c. We denote the security parameter by λ.

From Corollary 1, the security of LIT-SiGamal is based on the LIT-DDH assumption. Therefore, for the security of LIT-SiGamal, it is necessary not to be able to solve the LIT and CIST problems (Definition 4 and 3) in polynomial time. Additionally, we need a sufficiently large plaintext space. We determine the size of ℓ_A^a, ℓ_B^b, and ℓ_C^c from these requirements.

The size of $\boldsymbol{\ell_C^c}$. We first estimate the appropriate size of ℓ_C^c. The size of ℓ_C^c corresponds to that of a plaintext space. From the definition of the plaintext space $\mathcal{P}$, we have $\#\mathcal{P} \approx \ell_C^c$. Since the size of $\mathcal{P}$ is desired to be 2^λ, we have $\ell_C^c \approx 2^\lambda$.

The size of $\boldsymbol{\ell_B^b}$. We now discuss the size of ℓ_B^b. Bob computes ℓ_B^b-isogenies and needs to prevent revealing these isogenies.

The hardness of revealing Bob's isogenies is associated with the hardness of the CIST problem. By the statement of the CIST problem, information on torsion points in the CIST problem is masked by appropriate matrices; therefore, we expect any adversary cannot get valid torsion points information for revealing isogenies. Hence, the size of ℓ_B^b is determined by the hardness of the Isogeny problem of degree ℓ_B^b, and we set $\ell_B^b \approx 2^{2\lambda}$.

The size of $\boldsymbol{\ell_A^a}$. We finally estimate an appropriate size of ℓ_A^a. Alice computes an isogeny of degree $\ell_A^{2a} - n^2$ and needs to hide this isogeny.

The hardness of revealing Alice's isogeny is associated with the hardness of the LIT problem. Alice reveals the exact image of a basis of the ℓ_B^b-torsion subgroup under her isogeny. From the discussion in [18, Section 4.2], we are suggested to take ℓ_A^a such that $(\ell_A^{2a} - n^2) > \ell_B^{2b} \cdot 2^{2\lambda}$. Since n is a small integer, it suffices to take ℓ_A^a satisfying $\ell_A^{2a} \approx \ell_B^{2b} \cdot 2^{2\lambda}$. Note that we set $\ell_B^b \approx 2^{2\lambda}$ from the previous discussion. Therefore, we set $\ell_A^a \approx \ell_B^b \cdot 2^\lambda \approx 2^{3\lambda}$.

Consequently, we suggest to use a prime p about $2^{\lambda} \cdot 2^{2\lambda} \cdot 2^{3\lambda} = 2^{6\lambda}$ for LIT-SiGamal. We provide explicit primes for security parameter $\lambda = 128, 192, 256$ in Section 6.1.

5.3 Adaptive attack

We claim that there is an adaptive attack for LIT-SiGamal in this subsection. It is because LIT-SiGamal is based on the CIST problem and we can adapt a similar attack for IS-CUBE to LIT-SiGamal.

Suppose that Bob tries to reveal Alice's secret matrix $\mathbf{A}$ by sending incorrect ciphertext. If Bob obtains $\mathbf{A}$, he can decrypt any ciphertexts for Alice. Let

$$(E'_s, P', Q', R', E'_1, P'_1, Q'_1, R'_1)$$

be a ciphertext of Bob. If Bob is honest, he uses a common matrix in $\mathcal{M}_a$ to compute two pairs of torsion points (P', Q') and (P'_1, Q'_1). We assume that Bob uses different matrices $\mathbf{B}'$ and $\mathbf{B}''$ not necessarily in $\mathcal{M}_a$ to mask these pairs respectively. If $\mathbf{AB}' = \mathbf{B}''\mathbf{A}$, it holds that, from Kani's theorem,

$$(E'_s \times E'_1)/\langle (nP', P'_1), (nQ', Q'_1)\rangle \cong (E'_s \times E'_1)/\langle (nP', -P'_1), (nQ', -Q'_1)\rangle;$$

therefore, Alice succeeds in decrypting the incorrect ciphertext. Otherwise, the above equation does not hold without a negligible probability, and Alice fails the decryption. Hence, we have the following oracle:

$$O(\mathbf{B}', \mathbf{B}'') = \begin{cases} 1 & (\text{if } \mathbf{AB}' = \mathbf{B}''\mathbf{A}) \\ 0 & (\text{if } \mathbf{AB}' \neq \mathbf{B}''\mathbf{A}) \end{cases}.$$

This oracle is the same as in the adaptive attack proposed in [19]. Therefore, we have an adaptive attack for LIT-SiGamal based on the attack in [19].

FESTA can prevent this attack because, in the decryption process, Alice obtains the matrix $\mathbf{B}'$ and $\mathbf{B}''$. Conversely, Alice cannot obtain these matrices in the LIT-SiGamal setting. Therefore, to prevent this attack, we need an appropriate transform for LIT-SiGamal (see also Remark 3).

6 PoC implementation

In this section, we provide the results related to our proof-of-concept implementation of LIT-SiGamal. We implemented LIT-SiGamal via sagemath [12], and our code is available at `https://tomoriya.work/code.html`.

6.1 Parameter selection and the size of the scheme

In this subsection, we suggest the primes for LIT-SiGamal for the security parameters 128, 192, and 256 based on Section 5.2. Moreover, we provide the sizes

	Public key		Ciphertext	
λ	Original	Compressed	Original	Compressed
128	2,334	664	1,556	728
192	3,489	992	2,326	1,088
256	4,653	1,322	3,102	1,451

Table 2. The sizes of LIT-SiGamal (byte)

	λ	LIT-SiGamal	FESTA [5]	QFESTA [21]	terSIDH$^{\text{hyb}}$ [4]
Public key	128	664	561	247	701
	192	991	864	367	1,089
	256	1,322	1,246	487	1,479
Ciphertext	128	728	1,122	494	459
	192	1,088	1,728	734	706
	256	1,451	2,492	974	956

Table 3. Comparison of the sizes of isogeny-based PKEs (byte)

of the public keys and ciphertexts of LIT-SiGamal when using these primes, and compare them with other isogeny-based PKE schemes.

We denote by p_λ the prime for the security parameter λ. We define the primes as follows:

$$p_{128} = 2^{128\cdot 3+2} \cdot 3^{162} \cdot 5^{56} \cdot 30 - 1,$$
$$p_{192} = 2^{192\cdot 3+2} \cdot 3^{243} \cdot 5^{83} \cdot 118 - 1,$$
$$p_{256} = 2^{256\cdot 3+2} \cdot 3^{324} \cdot 5^{111} \cdot 436 - 1.$$

The "+2"s appearing in the exponents are used for the computation of isogenies between abelian varieties of dimension 2 via theta coordinates (see [10]). The bit length of p_{128} is 778, that of p_{192} is 1163, and that of p_{256} is 1551.

Table 2 summarizes the sizes of the public keys and ciphertexts of LIT-SiGamal using the above primes. Here, the compressed LIT-SiGamal is the variation of LIT-SiGamal provided in Section 4.4. Moreover, we summarize in Table 1 the sizes of the public keys and ciphertexts of the compressed LIT-SiGamal and some other isogeny-based PKE schemes. Here, we assume that terSIDH$^{\text{hyb}}$ is transformed into a PKE scheme by using the XOR operator and a hash function. As shown in Table 1, the public key of the compressed LIT-SiGamal is larger than that of FESTA, and its ciphertext is smaller than that of FESTA.

6.2 Computational time

We provide the experimental result of our PoC implementation. We measured the computational time of LIT-SiGamal and compared it with those of other isogeny-based PKE schemes.

Table 4 shows the computational times of the original and compressed LIT-SiGamal. We used p_{128}, p_{192}, and p_{256} provided in Section 6.1 for the primes.

	Public key gen.		Encryption		Decryption	
λ	Original	Compressed	Original	Compressed	Original	Compressed
128	15.76	18.98	0.31	3.68	1.30	2.53
192	31.45	38.68	0.58	7.51	2.70	5.12
256	56.48	70.77	0.97	12.93	4.90	9.17

Table 4. Computational time of LIT-SiGamal (sec.)

	λ	LIT-SiGamal	FESTA [5]	QFESTA [21]	terSIDH$^{\text{hyb}}$ [4]
Public key gen.	128	18.98	8.11	2.86	6.33
	192	38.68	164.75	5.81	23.66
	256	70.77	499.62	10.41	55.96
Encryption	128	3.68	5.63	3.98	0.83
	192	7.51	34.32	8.49	2.11
	256	12.93	100.28	15.57	4.15
Decryption	128	2.53	17.83*	10.01	4.62
	192	5.12	43.07*	24.33	17.08
	256	9.17	104.52*	48.95	39.83

Table 5. Comparison of the computational times of isogeny-based PKEs (sec.)

We measured the averages of 100 run times of each algorithm in LIT-SiGamal. The version of sagemath that we used was 10.1, and the computer for measuring the computational times was a MacBook Air with an Apple M1 CPU (3.2 GHz).

We also measured the averages of 100 run times of PoC implementations of FESTA [5], QFESTA [21], and terSIDH$^{\text{hyb}}$ [4] under the same environment. We summarized comparing these computational times in Table 5. As shown in this table, the encryption of LIT-SiGamal is as efficient as that of QFESTA, and the decryption of LIT-SiGamal is about 5x faster than that of QFESTA. In particular, the total of the encryption and decryption of LIT-SiGamal is the most efficient for $\lambda = 192$ and 256 in Table 5. On the other hand, the public key generation of LIT-SiGamal is about 7x slower than that of QFESTA.

Remark 4. The decryption process of FESTA can be improved by using algorithms provided in [10]; however, the PoC implementation of FESTA does not use these algorithms, and it is less efficient than the theoretical state-of-the-art implementation. Therefore, the times of the decryption process of FESTA shown in Table 5 are just for reference.

Remark 5. terSIDH$^{\text{hyb}}$ is a key exchange scheme and not a PKE scheme; therefore, we considered a PKE scheme to which terSIDH$^{\text{hyb}}$ is transformed by a hash function and the XOR operator. Because terSIDH$^{\text{hyb}}$ is an asymmetric scheme (*i.e.,* the difference of the times of two users is quite large), there are two ways to construct a PKE scheme from terSIDH$^{\text{hyb}}$. In Table 5, we selected the scheme with the lowest total computational cost for encryption and decryption.

7 Conclusion

We proposed a novel isogeny-based PKE scheme named LIT-SiGamal. LIT-SiGamal is a SiGamal-based PKE scheme that uses a LIT diagram instead of a CSIDH diagram. We also proposed a compressed version of LIT-SiGamal based on the compression technique of SIDH.

We proved that LIT-SiGamal has IND-CPA security under the LIT-DDH assumption, which is analogous to the DDH assumption in a LIT diagram. Moreover, we showed the existence of an adaptive attack for LIT-SiGamal.

We implemented LIT-SiGamal by using `sagemath` as a proof-of-concept. In our experimentation, LIT-SiGamal realizes efficient encryption and decryption. Compared with QFESTA, the encryption scheme of LIT-SiGamal is as efficient as QFESTA, and the decryption scheme is about 5x faster than QFESTA. Additionally, the total time of the encryption and decryption of LIT-SiGamal is the shortest among isogeny-based PKE schemes used in our experimentation for the security parameters 192 and 256.

Acknowledgements.

This work was supported by EPSRC through grant EP/V011324/1.

References

1. Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics*, 9(04):585–603, 2007.
2. Sarah Arpin. Adding level structure to supersingular elliptic curve isogeny graphs, 2022. `https://arxiv.org/abs/2203.03531`.
3. Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In *Advances in Cryptology – EUROCRYPT 2023*, pages 405–437. Springer, 2023.
4. Andrea Basso and Tako Boris Fouotsa. New SIDH countermeasures for a more efficient key exchange. In *Advances in Cryptology – ASIACRYPT 2023*, pages 208–233. Springer, 2023.
5. Andrea Basso, Luciano Maino, and Giacomo Pope. FESTA: Fast encryption from supersingular torsion attacks. In *Advances in Cryptology – ASIACRYPT 2023*, pages 98–126. Springer, 2023.
6. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In *Advances in Cryptology – EUROCRYPT 2023*, pages 423–447. Springer, 2023.
7. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in Cryptology – ASIACRYPT 2018*, pages 395–427. Springer, 2018.
8. Wouter Castryck and Frederik Vercauteren. A polynomial time attack on instances of M-SIDH and FESTA. In *Advances in Cryptology – ASIACRYPT 2023*, pages 127–156. Springer, 2023.

9. Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of SIDH public keys. In *Advances in Cryptology – EUROCRYPT 2017*, pages 679–706. Springer, 2017.
10. Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert. An algorithmic approach to $(2, 2)$-isogenies in the theta model and applications to isogeny-based cryptography, 2023. `https://eprint.iacr.org/2023/1747`.
11. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In *Advances in Cryptology – ASIACRYPT 2020*, pages 64–93. Springer, 2020.
12. The Sage Developers. Sagemath, version 10.0, 2023. `http://www.sagemath.org`.
13. Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: Countering SIDH attacks by masking information. In *Advances in Cryptology – EUROCRYPT 2023*, pages 282–309. Springer, 2023.
14. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – CRYPTO'99*, pages 537–554. Springer, 1999.
15. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Post-Quantum Cryptography – PQCrypto 2011*, pages 19–34. Springer, 2011.
16. Ernst Kani. The number of curves of genus two with elliptic differentials. 1997.
17. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In *Advances in Cryptology – EUROCRYPT 2023*, pages 448–471. Springer, 2023.
18. Tomoki Moriya. IS-CUBE: An isogeny-based compact KEM using a boxed SIDH diagram, 2023. `https://eprint.iacr.org/2023/1516`.
19. Tomoki Moriya and Hiroshi Onuki. The wrong use of FESTA trapdoor functions leads to an adaptive attack. `https://eprint.iacr.org/2023/1092`.
20. Tomoki Moriya, Hiroshi Onuki, and Tsuyoshi Takagi. SiGamal: A supersingular isogeny-based pke and its application to a prf. In *Advances in Cryptology – ASIACRYPT 2020*, pages 551–580. Springer, 2020.
21. Kohei Nakagawa and Hiroshi Onuki. QFESTA: Efficient algorithms and parameters for festa using quaternion algebras, 2023. `https://eprint.iacr.org/2023/1468`.
22. Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Transactions on information Theory*, 24(1):106–110, 1978.
23. Damien Robert. Efficient algorithms for abelian varieties and their moduli spaces, 2021. Université de Bordeaux (UB). `https://hal.science/tel-03498268`.
24. Damien Robert. Breaking SIDH in polynomial time. In *Advances in Cryptology – EUROCRYPT 2023*, pages 472–503. Springer, 2023.
25. Joseph Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.
26. Benjamin Smith. *Explicit endomorphisms and correspondences.* PhD thesis, University of Sydney, 2005.
27. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris Sér. A*, 273(5):238–241, 1971.

Appendix A Computation of $(2,2)$-isogenies via theta coordinates

In the process of LIT-SiGamal, we need to compute isogenies of high-dimensional varieties. In particular, we need to calculate $(2,2)$-isogenies. There are two well-known methods for the computation of $(2,2)$-isogenies: the method of Richelot isogenies and that of theta coordinates of level 2 structure. We choose the method of theta coordinates for computing these isogenies.

Dartois, Maino, Pope and Robert provided a concrete method to compute $(2,2)$-isogenies by using theta coordinates in [10]. Their method is used in general cases, and not optimized for LIT-SiGamal. In this section, we modify some of their algorithms and introduce novel propositions to compute $(2,2)$-isogenies for LIT-SiGamal.

A.1 Introduction of theta coordinates

In this subsection, we introduce theta coordinates and their properties for computing isogenies. See [23] and [10] for more details.

Let k be a field, let A be an abelian variety over k of dimension g, and let $\mathcal{L}$ be a totally symmetric ample invertible sheaf of separable type over A. I.e., $\mathcal{L}$ is an ample invertible sheaf over A such that there is an isomorphism $\varphi\colon \mathcal{L} \xrightarrow{\sim} [-1]^*\mathcal{L}$ satisfying $\varphi_P = \mathrm{id}_{\mathcal{L}_P}$ for any $P \in A$ of order 2, and the dimension of the space of the global sections $\Gamma(A,\mathcal{L})$ is coprime to the characteristic of k. We define $H(\mathcal{L})$ as a subgroup $\{P \in A \mid T_P^*\mathcal{L} \cong \mathcal{L}\}$, where T_P is the translation-by-P map over A. Then, there is a sequence $D = (d_1, \ldots, d_g)$ with $d_{i+1} | d_i$ $(i = 1, \ldots, g-1)$ and $d_1 > 1$ such that

$$H(\mathcal{L}) \cong \bigoplus_{i=1}^{g} \mathbb{Z}/d_i\mathbb{Z} \oplus \mathrm{Hom}\left(\bigoplus_{i=1}^{g} \mathbb{Z}/d_i\mathbb{Z},\ \overline{k}^{\times}\right).$$

We say $\mathcal{L}$ is of type D. We denote $\oplus_{i=1}^{g}\mathbb{Z}/d_i\mathbb{Z}$ by $K(D)$ and $\mathrm{Hom}(K(D), \overline{k}^{\times})$ by $\hat{K}(D)$. We also define $\mathcal{G}(\mathcal{L})$ as

$$\mathcal{G}(\mathcal{L}) := \{(P, \phi_P) \mid x \in H(\mathcal{L}),\ \phi_P\colon T_P^*\mathcal{L} \xrightarrow{\sim} \mathcal{L}\}.$$

Then, the $\mathcal{G}(\mathcal{L})$ also has a group structure, and there is an exact sequence:

$$0 \longrightarrow \overline{k}^{\times} \longrightarrow \mathcal{G}(\mathcal{L}) \longrightarrow H(\mathcal{L}) \longrightarrow 0.$$

Define $\mathcal{G}(D) := \overline{k}^{\times} \oplus K(D) \oplus \hat{K}(D)$. Then, the above sequence is isomorphic to the following sequence:

$$0 \longrightarrow \overline{k}^{\times} \longrightarrow \mathcal{G}(D) \longrightarrow K(D) \oplus \hat{K}(D) \longrightarrow 0.$$

We call the isomorphism $\Theta_{\mathcal{L}}\colon \mathcal{G}(D) \xrightarrow{\sim} \mathcal{G}(\mathcal{L})$ a *theta structure of type* D. Let $V(D)$ be the space $\mathrm{Hom}(K(D), \overline{k})$. Then, the $V(D)$ is the unique irreducible

representation of $\mathcal{G}(D)$. Similarly, the space of global sections $\Gamma(A, \mathcal{L})$ is the irreducible representation of $\mathcal{G}(\mathcal{L})$. Therefore, there is an isomorphism between $V(D)$ and $\Gamma(A, \mathcal{L})$ compatible with those group actions. The basis of $\Gamma(A, \mathcal{L})$ derived from the canonical basis of $V(D)$ (*i.e.,* the basis consisting of the Kronecher delta functions on $K(D)$) by the isomorphism is called *theta coordinates*. If $d_1 = \ldots = d_g = n$, we call them theta coordinates of *level* n.

Let $(\theta_{A,i})_{i=0,\ldots,2^g-1}$ be theta coordinates of level 2 over A. Let $S_1, \ldots, S_g$ be points of $A[2]$ derived from the canonical basis of $K(2, \ldots, 2)$, and let $T_1, \ldots, T_g$ be those from $\hat{K}(2, \ldots, 2)$. Theta coordinates $(\theta_{A,i})_i$ can be determined by a symplectic basis $(S'_1, \ldots, S'_g, T'_1, \ldots, T'_g)$ of $A[4]$ with $2S'_i = S_i$ and $2T'_i = T_i$ for $i = 0, \ldots, 2^g - 1$. By using theta coordinates, we can represent points in A by elements in $\mathbb{P}^{2^g-1}$ as follows:

$$P \longmapsto (\theta_{A,0}(P) : \theta_{A,1}(P) : \ldots : \theta_{A,2^g-1}(P)) \in \mathbb{P}^{2^g-1}.$$

This map is an embedding of a Kummer variety $A/\{\pm 1\}$ to $\mathbb{P}^{2^g-1}$. We call $(\theta_{A,i}(0))_i$ the *theta-null point*, which represents A. Let $\mathcal{H}$ be the Hadamard matrix of order 2^g. We call $\mathcal{H}((\theta_{A,i})_i)$ the *dual theta coordinates* of $(\theta_{A,i})_i$ and denote them by $(\tilde{\theta}_{A,i})_i$. The dual theta coordinates of $(\theta_{A,i})_i$ are also theta coordinates of A. It is easy to see that $(\tilde{\tilde{\theta}}_{A,i})_i = (\theta_{A,i})_i$. There is a symplectic basis $(\tilde{S}'_1, \ldots, \tilde{S}'_g, \tilde{T}'_1, \ldots, \tilde{T}'_g)$ associated to $(\tilde{\theta}_{A,i})_i$ such that

$$(\tilde{\theta}_{A,i}(\tilde{T}'_j))_i = \mathcal{H}((\theta_{A,i}(S'_j))_i), \quad (\tilde{\theta}_{A,i}(\tilde{S}'_j))_i = \mathcal{H}((\theta_{A,i}(T'_j))_i).$$

Define $K_1 := \langle S_1, \ldots, S_g \rangle$ and $K_2 := \langle T_1, \ldots, T_g \rangle$. Let f be a separable isogeny $f\colon A \to B$ with $\ker f = K_2$. The following properties of theta coordinates are important to construct algorithms for computing $(2,2)$-isogenies:

- $(\theta_{A,i}(P))_i = (\theta_{A,i}(-P))_i$,
- $(\theta_{A,i}(P + T_j))_i = ((-1)^{\langle i|j \rangle}\theta_{A,i}(P))_i$,
- $(\theta_{A,i}(P + S_j))_i = (\theta_{A,i+j}(P))_i$,
- There are theta coordinates of B such that

$$(\theta_{A,i}(P+Q) \cdot \theta_{A,i}(P-Q))_i = \mathcal{H}((\tilde{\theta}_{B,i}(f(P)) \cdot \tilde{\theta}_{B,i}(f(Q)))_i),$$

  points derived from the canonical basis of $K(2, \ldots, 2)$ by the theta structure associated to $(\theta_{B,i})_i$ are $f(S_1), \ldots, f(S_g)$, and those from $\hat{K}(2, \ldots, 2)$ are $f(T'_1), \ldots, f(T'_g)$ (the duplication formula),
- $\theta_{A,i}(T'_j) = 0$ if $\langle i|j \rangle = 1$,
- $\theta_{A,i}(S'_j) = \theta_{A,i+j}(S'_j)$.

Here, we assume that the indices i and j belong to $K(2, \ldots, 2)$ by the bijection map

$$\begin{array}{ccc} K(2, \ldots, 2) & \longrightarrow & \{0, \ldots, 2^g - 1\} \\ (a_1, \ldots, a_g) & \longmapsto & \sum_{l=1}^{g} a_l 2^{l-1} \end{array},$$

and $\langle \cdot \mid \cdot \rangle\colon K(2, \ldots, 2)^2 \to \mathbb{Z}/2\mathbb{Z}$ is the inner product over $K(2, \ldots, 2)$. By using the above properties, we can construct algorithms to compute $(2,2)$-isogenies (see [10]).

Algorithm 1 Change of a basis

Require: Montgomery curves E, E', a basis $\{P, Q\}$ of $E[4]$, and a basis $\{P', Q'\}$ of $E'[4]$ such that $x(P) = x(P') = 1$
Ensure: The basis-change matrix N
$H_1 \leftarrow \mathsf{action\_by\_translation}(\mathrm{Q})$ ([10, Algorithm 1])
$H_2 \leftarrow \mathsf{action\_by\_translation}(\mathrm{Q'})$
$n_{00} \leftarrow h_{00|1} \cdot h_{00|2} + h_{10|1} \cdot h_{10|2} + 1$
$n_{01} \leftarrow h_{00|1} \cdot h_{10|2} + h_{10|1} \cdot h_{00|2}$
$n_{02} \leftarrow h_{10|1} \cdot h_{00|2} + h_{00|1} \cdot h_{10|2}$
$n_{03} \leftarrow n_{00}$
$n_{10} \leftarrow h_{00|2} \cdot n_{00} + h_{01|2} \cdot n_{01}$
$n_{11} \leftarrow h_{10|2} \cdot n_{00} + h_{11|2} \cdot n_{01}$
$n_{12} \leftarrow h_{00|2} \cdot n_{02} + h_{01|2} \cdot n_{03}$
$n_{13} \leftarrow h_{10|2} \cdot n_{02} + h_{11|2} \cdot n_{03}$
$\begin{pmatrix} n_{20} & n_{21} & n_{22} & n_{23} \end{pmatrix} \leftarrow \begin{pmatrix} n_{02} & n_{03} & n_{00} & n_{01} \end{pmatrix}$
$\begin{pmatrix} n_{30} & n_{31} & n_{32} & n_{33} \end{pmatrix} \leftarrow \begin{pmatrix} n_{12} & n_{13} & n_{10} & n_{11} \end{pmatrix}$
return N

A.2 Computing theta coordinates from a product of Montgomery curves

In the LIT-SiGamal setting, we compute an isogeny from a product of two Montgomery curves to an abelian variety of dimension 2. Therefore, to use the formulas of theta coordinates, we first determine a theta structure of a product of Montgomery curves. We find that [10, Algorithm 2] determines theta coordinates of a product of two Montgomery curves $E \times E'$ determined by $T_1' = (P, P')$, $T_2' = (Q, Q')$, $S_1' = (0, Q')$, and $S_2' = (P, 0)$, where $\{P, Q\}$ is a basis of $E[4]$ and $\{P', Q'\}$ is that of $E'[4]$. Precisely speaking, [10, Algorithm 2] outputs a basis-change matrix and we can easily compute the theta structure by using this matrix. We can assume that $x(P) = x(P') = 1$ in the LIT-SiGamal setting; therefore, we can simplify [10, Algorithm 2] for LIT-SiGamal. Algorithm 1 shows our new algorithm to obtain a basis-change matrix N from a product of two Montgomery curves.

A.3 Property of special codomain

Let $(\theta_{E\times E',i})_i$ be theta coordinates of $E \times E'$ defined by the above method, let K_2 be a subgroup of $E \times E'$ generated by T_1 and T_2, and let $f\colon E \times E' \to B$ be a $(2,2)$-isogeny with $\ker f = K_2$. Denote theta coordinates of B by $(\theta_{B,i})_i$. Then, it has been known that one of $\tilde{\theta}_{B,i}(0)$s may be zero. We prove in Proposition 4 that $\tilde{\theta}_{B,3}(0)$ is always 0 in our construction of the theta coordinates of $E \times E'$.

Proposition 4. *Let E and E' be elliptic curves, let $\{P, Q\}$ be a basis of $E[4]$, and let $\{P', Q'\}$ be a basis of $E'[4]$. Assume that $(\theta_{E\times E',i})_i$ are theta coordinates of $E \times E'$ determined by $T_1' = (P, P')$, $T_2' = (Q, Q')$, $S_1' = (0, Q')$, and $S_2' = (P, 0)$. Denote a subgroup $\langle 2T_1', 2T_2' \rangle$ by K_2, and $(E \times E')/K_2$ by B. Let*

$(\theta_{B,i})_{i=0,1,2,3}$ be theta coordinates of B satisfying the duplication formula. Then, we have $\tilde{\theta}_{B,3}(0) = 0$.

Proof. Note that it follows from the construction of $(\theta_{E\times E',i})_i$ that

$$\theta_{E\times E',i}((R,R')) = \theta_{E\times E',i}((-R,R')) = \theta_{E\times E',i}((R,-R'))$$

for any $(R,R') \in E \times E'$. Since we have $P = -3P$, it holds that

$$(\theta_{E\times E',i}(T_1'))_i = (\theta_{E\times E',i}(T_1' + 2S_2'))_i.$$

Recall that it holds that

$$\begin{aligned}(\theta_{E\times E',i}(P + 2S_2'))_i &= (\theta_{E\times E',i+2}(P))_i,\\ (\theta_{E\times E',i}(T_1'))_i &= (* : 0 : * : 0).\end{aligned}$$

Therefore, we have

$$(\theta_{E\times E',i}(T_1'))_i = (1 : 0 : \pm 1 : 0).$$

It also holds that

$$\begin{aligned}(\theta_{E\times E',i}(T_2'))_i &= (\theta_{E\times E',i}(T_2' + 2S_1'))_i,\\ \theta_{E\times E',i}(T_2') &= (* : * : 0 : 0).\end{aligned}$$

Therefore, by the same discussion as above, we have

$$(\theta_{E\times E',i}(T_2'))_i = (1 : \pm 1 : 0 : 0).$$

Denote by f the $(2,2)$-isogeny $E \times E' \to B$. It follows from the duplication formula that

$$\begin{aligned}(\tilde{\theta}_{B,i}(f(T_1')) \cdot \tilde{\theta}_{B,i}(0))_i &= \mathcal{H}((\theta_{E\times E',i}(T_1')^2)_i) = (1 : \pm 1 : 0 : 0),\\ (\tilde{\theta}_{B,i}(f(T_2')) \cdot \tilde{\theta}_{B,i}(0))_i &= \mathcal{H}((\theta_{E\times E',i}(T_2')^2)_i) = (1 : 0 : \pm 1 : 0).\end{aligned}$$

Denote $(\tilde{\theta}_{B,i}(0))_i$ by $(\alpha : \beta : \gamma : \delta)$. Note that $f(T_1')$ and $f(T_2')$ are points derived from the canonical basis of $K(2,2)$ by the theta structure related to $(\tilde{\theta}_{B,i})_i$. Therefore, it holds that

$$\begin{aligned}(\tilde{\theta}_{B,i}(f(T_1')))_i &= (\beta : \alpha : \delta : \gamma),\\ (\tilde{\theta}_{B,i}(f(T_2')))_i &= (\gamma : \delta : \alpha : \beta),\end{aligned}$$

and we have $\beta\delta = 0$ and $\alpha\beta \neq 0$. Hence, we conclude $\delta = 0$. □

A.4 Computation of (2, 2)-isogenies using the projective inversion

In [10], the authors proposed the algorithms using the batched inversion method for computing inversions of multiple elements in $\mathbb{F}_{p^2}$. Since theta coordinates embed points to the projective space $\mathbb{P}^3$, we do not need to compute the exact

Algorithm 2 Projective inversion (proj_inv)

Require: $(a_0 : a_1 : \ldots : a_m) \in \mathbb{P}^m$
Ensure: $(a_0^{-1} : a_1^{-1} : \ldots : a_m^{-1}) \in \mathbb{P}^m$
1: **if** $m = 0$ **then**
2: **return** (1)
3: **else if** $m = 1$ **then**
4: **return** $(a_1 : a_0)$
5: **else**
6: $(b_0 : \ldots : b_{\lfloor m/2 \rfloor}) \leftarrow \mathsf{proj\_inv}((a_0 : \ldots : a_{\lfloor m/2 \rfloor}))$
7: $(b_{\lfloor m/2 \rfloor + 1} : \ldots : b_m) \leftarrow \mathsf{proj\_inv}((a_{\lfloor m/2 \rfloor + 1} : \ldots : a_m))$
8: $a_{\mathrm{left}} \leftarrow b_0 \cdot a_0$
9: $a_{\mathrm{right}} \leftarrow b_m \cdot a_m$
10: **for** $i = 0, \ldots, \lfloor m/2 \rfloor$ **do**
11: $b_i \leftarrow a_{\mathrm{right}} \cdot b_i$
12: **end for**
13: **for** $i = \lfloor m/2 \rfloor + 1, \ldots, m$ **do**
14: $b_i \leftarrow a_{\mathrm{left}} \cdot b_i$
15: **end for**
16: **return** $(b_0 : b_1 : \ldots : b_m)$
17: **end if**

inversions of the given elements in some cases. Therefore, we can reduce the computational cost of some of the algorithms for theta coordinates by using the projective inversion (Algorithm 2) instead of the batched inversion.

Algorithms 3 and 4 are algorithms to compute the codomain variety of a $(2, 2)$-isogeny using the projective inversion. These are respectively more efficient than algorithms proposed in [10] (*i.e.,* [10, Algorithm 6] and [10, Algorithm 7]).

Remark 6. The outputs of [10, Algorithms 6 and 7] are supposed to have $\alpha = 1$, while those of our novel algorithms are not. Therefore, there is a possibility that the total computational cost to compute $(2, 2)$-isogenies using Algorithms 3 and 4 is less efficient than the original algorithm in [10] because extra computations using α occur in the total computation. In particular, m extra multiplications occur in evaluating m points.

In our implementation, the total algorithm becomes more efficient by replacing all [10, Algorithms 6 and 7] to Algorithms 3 and 4. However, for the above reason, it is thought that the most efficient algorithm for computing $(2, 2)$-isogenies uses both [10, Algorithms 6 and 7] and Algorithms 3 and 4. The optimization is our future work.

A.5 Computation of an isogenies between products of two elliptic curves

Let E, E', E_1, E_1' be elliptic curves, and let $\Phi\colon E \times E' \to E_1 \times E_1'$ be a $(2^a, 2^a)$-isogeny with $\ker \Phi = \langle (P, P'), (Q, Q') \rangle$, where $\{P, Q\}$ is a basis of $E[2^a]$ and $\{P', Q'\}$ is that of $E'[2^a]$. In the LIT-SiGamal setting, we compute this isogeny

Algorithm 3 Codomain

Require: Theta coordinates $(x_{T_1''} : y_{T_1''} : z_{T_1''} : w_{T_1''})$ and $(x_{T_2''} : y_{T_2''} : z_{T_2''} : w_{T_2''})$ of eight torsion points T_1'' and T_2'' such that $K_2 = \langle 2T_1'', 2T_2'' \rangle$
Ensure: The dual theta-null point $(\alpha : \beta : \gamma : \delta)$ of A/K_2 and $(\alpha^{-1} : \beta^{-1} : \gamma^{-1} : \delta^{-1})$ and the theta-null point $(a : b : c : d)$ of A/K_2
1: $(x_1 : y_1 : z_1 : w_1) \leftarrow \mathcal{H} \circ \mathcal{S}(x_{T_1''} : y_{T_1''} : z_{T_1''} : w_{T_1''})$
2: $(x_2 : y_2 : z_2 : w_2) \leftarrow \mathcal{H} \circ \mathcal{S}(x_{T_2''} : y_{T_2''} : z_{T_2''} : w_{T_2''})$
3: $(\alpha_0 : \gamma_0 : \delta_0) \leftarrow \mathsf{proj\_inv}((x_1 : x_2 : y_2))$
4: $\alpha \leftarrow x_1 \cdot \alpha_0$
5: $\beta \leftarrow y_1 \cdot \alpha_0$
6: $\gamma \leftarrow z_2 \cdot \gamma_0 \cdot \alpha$
7: $\delta \leftarrow w_2 \cdot \delta_0 \cdot \beta$
8: $\beta \leftarrow \beta \cdot \alpha$
9: $\alpha \leftarrow \alpha^2$
10: $(a : b : c : d) \leftarrow \mathcal{H}(\alpha : \beta : \gamma : \delta)$
11: **return** $(\alpha : \beta : \gamma : \delta)$, $\mathsf{proj\_inv}((\alpha : \beta : \gamma : \delta))$, $(a : b : c : d)$

Algorithm 4 Codomain if $\delta = 0$

Require: Theta coordinates $(x_{T_1''} : y_{T_1''} : z_{T_1''} : w_{T_1''})$ and $(x_{T_2''} : y_{T_2''} : z_{T_2''} : w_{T_2''})$ of eight torsion points T_1'' and T_2'' such that $K_2 = \langle 2T_1'', 2T_2'' \rangle$
Ensure: The dual theta-null point $(\alpha : \beta : \gamma : 0)$ of A/K_2 and $(\alpha^{-1} : \beta^{-1} : \gamma^{-1} : 0)$ and the theta-null point $(a : b : c : d)$ of A/K_2
1: $(x_1 : y_1 : z_1 : w_1) \leftarrow \mathcal{H} \circ \mathcal{S}(x_{T_1''} : y_{T_1''} : z_{T_1''} : w_{T_1''})$
2: $(x_2 : y_2 : z_2 : w_2) \leftarrow \mathcal{H} \circ \mathcal{S}(x_{T_2''} : y_{T_2''} : z_{T_2''} : w_{T_2''})$
3: $(\alpha_0 : \gamma_0) \leftarrow \mathsf{proj\_inv}((x_1 : x_2))$
4: $\alpha \leftarrow x_1 \cdot \alpha_0$
5: $\beta \leftarrow y_1 \cdot \alpha_0$
6: $\gamma \leftarrow z_2 \cdot \gamma_0$
7: $(a : b : c : d) \leftarrow \mathcal{H}(\alpha : \beta : \gamma : 0)$
8: **return** $(\alpha : \beta : \gamma : 0)$, $(\mathsf{proj\_inv}((\alpha : \beta : \gamma)) : 0)$, $(a : b : c : d)$

by computing two isogenies $\Phi_0 \colon E \times E' \to V$ and $\Phi_1 \colon E_1 \times E_1' \to V$ such that $\Phi = \hat{\Phi}_1 \circ \Phi_0$. However, theta coordinates of V are not unique, and the representation of the codomain of Φ_0 and that of Φ_1 are not the same if we use the method proposed in [10]. Therefore, to compute Φ, we need to detect the isomorphism between two representations.

We prove in Proposition 5 that the following matrix gives the isomorphism between two representations in our construction:

$$\mathcal{H} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

By using this transformation, we can evaluate image points under $\hat{\Phi}_1 \circ \Phi_0$.

Proposition 5. *Let E, E', E_1, E_1' be elliptic curves. Assume that there is an isogeny diamond as follows:*

$$\begin{array}{ccc} E_1 & \xrightarrow{\phi} & E \\ {\scriptstyle\phi_1}\downarrow & & \downarrow{\scriptstyle\phi_1'} \\ E' & \xrightarrow[\phi']{} & E_1' \end{array}$$

Here, we also assume that $\deg\phi + \deg\phi_1 = 2^a$ *and* $\deg\phi_1 \equiv -1 \pmod 4$ *for an integer* $a \geq 2$. *Let* $\Phi\colon E \times E' \to E_1 \times E_1'$ *be a* $(2^a, 2^a)$*-isogeny represented by*

$$\Phi = \begin{pmatrix} \hat{\phi} & \hat{\phi_1} \\ -\phi_1' & \phi' \end{pmatrix}.$$

Let a_1, a_2 *be integers with* $a = a_1 + a_2$, *let* $\{P, Q\}$ *be a basis of* $E_1[2^a]$, *let* Φ_0 *be a separable isogeny from* $E \times E'$ *with*

$$\ker\Phi_0 = \langle (2^{a_2}\phi(P), 2^{a_2}\phi_1(P)), (2^{a_2}\phi(Q), 2^{a_2}\phi_1(Q))\rangle,$$

and let Φ_1 *be a separable isogeny from* $E_1 \times E_1'$ *with*

$$\ker\Phi_1 = \langle (2^{a_1}(\deg\phi_1)P, 2^{a_1}(\phi_1' \circ \phi)(P)), (2^{a_1}(\deg\phi_1)Q, 2^{a_1}(\phi_1' \circ \phi)(Q))\rangle.$$

Note that $\Phi = \hat{\Phi}_1 \circ \Phi_0$. *Denote by* V *the codomain abelian variety of* Φ_0 *and* Φ_1.

Let $(\theta_{V,i})_i$ *be theta coordinates of* V *determined by the symplectic basis*

$$\begin{aligned}&(\Phi_0((0, 2^{a-2}\phi_1(Q))),\ \Phi_0((2^{a-2}\phi(P), 0)),\\ &\quad \Phi_0((2^{a_2-2}\phi(P), 2^{a_2-2}\phi_1(P))),\ \Phi_0((2^{a_2-2}\phi(Q), 2^{a_2-2}\phi_1(Q)))),\end{aligned}$$

and let $(\theta'_{V,i})_i$ *be theta coordinates of* V *determined by the symplectic basis*

$$\begin{aligned}&(\Phi_1((0, 2^{a-2}(\phi_1' \circ \phi)(Q))),\ \Phi_1((2^{a-2}P, 0)),\\ &\quad \Phi_1((2^{a_2-2}P, -2^{a_2-2}(\phi_1' \circ \phi)(P))),\ \Phi_1((-2^{a_2-2}Q, 2^{a_2-2}(\phi_1' \circ \phi)(Q)))).\end{aligned}$$

Then, for any $R \in E \times E'$, *it holds that*

$${}^t(\theta'_{V,i}(\Phi_0(R)))_i = \mathcal{H} \cdot \begin{pmatrix} 1&0&0&0\\ 0&0&1&0\\ 0&1&0&0\\ 0&0&0&1 \end{pmatrix} \cdot {}^t(\theta_{V,i}(\Phi_0(R)))_i.$$

Proof. It suffices to find the isomorphism mapping the symplectic base associated to $(\theta_{V,i})_i$ to that associated to $(\theta'_{V,i})_i$. We denote the basis associated to $(\theta_{V,i})_i$ by

$$(S_1, S_2, T_1, T_2)$$

and that associated to $(\theta'_{V,i})_i$ by

$$(S_1', S_2', T_1', T_2').$$

It is clear that $S_1 = 2^{a-2}\Phi_0((0,\phi_1(Q)))$, and it follows from $\deg\phi_1 \equiv -1 \pmod 4$ that

$$T_2' = 2^{a_1-2}(\Phi_1 \circ \Phi)((0,\phi_1(Q))) = 2^{a-2}\Phi_0((0,\phi_1(Q))).$$

Therefore, it holds that $S_1 = T_2'$. For a similar reason, we have $S_2 = T_1'$. Note that $\hat{\Phi}$ can be represented by

$$\hat{\Phi} = \begin{pmatrix} \phi & -\hat{\phi}_1' \\ \phi_1 & \hat{\phi}' \end{pmatrix}.$$

We have

$$T_2 = 2^{a_2-2}(\Phi_0 \circ \hat{\Phi})((0,(\phi_1' \circ \phi)(Q))) = 2^{a-2}\Phi_1((0,(\phi_1' \circ \phi)(Q))) = S_1'.$$

For a similar reason, we also have $T_1 = S_2'$. Therefore, the isomorphism mapping (S_1, S_2, T_1, T_2) to (S_1', S_2', T_1', T_2') is given by the composition of the following two maps:

- The isomorphism mapping (S_1, S_2, T_1, T_2) to (S_2, S_1, T_2, T_1).
- The isomorphism mapping (S_1, S_2, T_1, T_2) to (T_1, T_2, S_1, S_2),

The first isomorphism is represented by the matrix

$$\begin{pmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{pmatrix},$$

and the second one is represented by $\mathcal{H}$. This completes the proof of the proposition. □

A.6 Computation of a product of Montgomery curves from theta coordinates

At the end of the computation of a $(2^a, 2^a)$-isogeny, we transform a point given by theta coordinates to a point in a product of two Montgomery curves. The basic method to obtain the representation of a point in Montgomery curves from theta coordinates is found in [10, Section 4.1], and we can simplify this method by using Proposition 4. However, this method may not be directly adapted to LIT-SiGamal. It is because, in the LIT-SiGamal setting, we want to obtain a representation in a fixed product of Montgomery curves $E \times E'$, while the basic method may provide a representation in $E_1 \times E_1'$ that is isomorphic to but does not equal to $E \times E'$. Therefore, we need to modify the method for LIT-SiGamal.

Let E and E' be Montgomery curves, and $(\theta_{E\times E',i})_i$ be theta coordinates determined by

$$((0,Q'),(P,0),(P,P'),(Q,Q')),$$

where $\{P, Q\}$ and $\{P', Q'\}$ are bases of $E[4]$ and $E'[4]$ respectively such that $x(P) = x(P') = 1$. The aim is to find the method to obtain $x(R)$ and $x(R')$ from $(\theta_{E\times E',i}((R, R')))_i$. Denote $\mathcal{H}((\theta_{E\times E',i}((R, R')))_i)$ by $(a : b : c : d)$. From the method in [10, Section 4.1] and Proposition 4, we can represent (R, R') by $((a : \sqrt{-1}b), (a : \sqrt{-1}c))$. Here, $((a : \sqrt{-1}b), (a : \sqrt{-1}c))$ is the pair of $(\theta_{E,i}(R))_i$ and $(\theta_{E',i}(R'))_i$, where $(\theta_{E,i})_i$ (resp. $(\theta_{E',i})_i$) are theta coordinates of E (resp. E') defined by an appropriate theta structure. We now discuss the way to compute $x(R)$ and $x(R')$ from $(a : \sqrt{-1}b)$ and $(a : \sqrt{-1}c)$. We first focus on $(2P, 0)$ and $(P, 0)$. Denote the dual theta-null point of $E \times E'$ by $(\alpha : \beta : \gamma : \delta)$. We have

$$\begin{aligned}(\alpha : \beta : -\gamma : -\delta) &= \mathcal{H}((\theta_{E\times E',i}((2P, 0)))_i),\\ (* : * : 0 : 0) &= \mathcal{H}((\theta_{E\times E',i}((P, 0)))_i),\\ (\alpha : \beta : \gamma : \delta) &= \mathcal{H}((\theta_{E\times E',i}((0, 0)))_i).\end{aligned}$$

Therefore, it holds that

$$\begin{aligned}\{(\theta_{E,i}(2P))_i, (\theta_{E',i}(0))_i\} &= \{(\alpha : \sqrt{-1}\beta), (\alpha : -\sqrt{-1}\gamma)\},\\ \{(\theta_{E,i}(0))_i, (\theta_{E',i}(0))_i\} &= \{(\alpha : \sqrt{-1}\beta), (\alpha : \sqrt{-1}\gamma)\},\\ \{(\theta_{E,i}(P))_i, (\theta_{E',i}(0))_i\} &\supset \{(1 : 0)\}.\end{aligned}$$

Note that it follows from $\pm 2P \neq 0$ that $(\theta_{E,i}(2P))_i \neq (\theta_{E,i}(0))_i$. We conclude that

$$\begin{gathered}(\theta_{E',i}(0))_i = (\alpha : \sqrt{-1}\beta), \quad (\theta_{E,i}(2P))_i = (\alpha : -\sqrt{-1}\gamma),\\ (\theta_{E,i}(P))_i = (1 : 0), \quad (\theta_{E,i}(0))_i = (\alpha : \sqrt{-1}\gamma).\end{gathered}$$

From the above observation, we also have

$$(\theta_{E',i}(R'))_i = (a : \sqrt{-1}b), \quad (\theta_{E,i}(R))_i = (a : \sqrt{-1}c).$$

Recall that $x(P) = 1$ and $x(2P) = 0$. The transformation from $(\theta_{E,i}(R))_i$ to $x(R)$ is given a 2×2-matrix M_T. This matrix needs to satisfy

$$\begin{pmatrix} * \\ 0 \end{pmatrix} = M_T \begin{pmatrix} \alpha \\ \sqrt{-1}\gamma \end{pmatrix}, \quad \begin{pmatrix} 0 \\ * \end{pmatrix} = M_T \begin{pmatrix} \alpha \\ -\sqrt{-1}\gamma \end{pmatrix}, \quad * \begin{pmatrix} 1 \\ 1 \end{pmatrix} = M_T \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Therefore, we have

$$M_T = \begin{pmatrix} \sqrt{-1}\gamma & \alpha \\ \sqrt{-1}\gamma & -\alpha \end{pmatrix}.$$

We next focus on $(2P, 2P')$ and (P, P'). From a similar discussion as above, we have

$$(\theta_{E',i}(2P'))_i = (\beta : \sqrt{-1}\alpha), \quad (\theta_{E',i}(P'))_i = (1 : \sqrt{-1}).$$

Therefore, from $x(2P') = 0$ and $x(P') = 1$, the representation matrix M'_T of the transformation from $(\theta_{E',i}(R'))_i$ to $x(R')$ satisfies

$$\begin{pmatrix} * \\ 0 \end{pmatrix} = M'_T \begin{pmatrix} \alpha \\ \sqrt{-1}\beta \end{pmatrix}, \quad \begin{pmatrix} 0 \\ * \end{pmatrix} = M'_T \begin{pmatrix} \beta \\ \sqrt{-1}\alpha \end{pmatrix}, \quad * \begin{pmatrix} 1 \\ 1 \end{pmatrix} = M'_T \begin{pmatrix} 1 \\ \sqrt{-1} \end{pmatrix}.$$

Algorithm 5 Theta coordinates to Montgomery curves

Require: Theta coordinates $(\theta_{E\times E',i}((R,R')))_i$ and theta-null point $(\theta_{E\times E',i}((0,0)))_i$ of $E \times E'$
Ensure: $x(R)$ and $x(R')$
1: $(a : b : c : d) \leftarrow \mathcal{H}((\theta_{E\times E',i}((R,R')))_i)$
2: $(X_R : Z_R) \leftarrow (a : \sqrt{-1}c)$
3: $(X_{R'} : Z_{R'}) \leftarrow (a : \sqrt{-1}b)$
4: $(\alpha : \beta : \gamma : \delta) \leftarrow \mathcal{H}((\theta_{E\times E',i}((0,0)))_i)$
5: $M_T \leftarrow \begin{pmatrix} \sqrt{-1}\gamma & \alpha \\ \sqrt{-1}\gamma & -\alpha \end{pmatrix}$
6: $\begin{pmatrix} X_R \\ Z_R \end{pmatrix} \leftarrow M_T \cdot \begin{pmatrix} X_R \\ Z_R \end{pmatrix}$
7: $M'_T \leftarrow \begin{pmatrix} -\sqrt{-1}\alpha & \beta \\ \sqrt{-1}\beta & -\alpha \end{pmatrix}$
8: $\begin{pmatrix} X_{R'} \\ Z_{R'} \end{pmatrix} \leftarrow M'_T \cdot \begin{pmatrix} X_{R'} \\ Z_{R'} \end{pmatrix}$
9: **return** $(X_R : Z_R)$ and $(X_{R'} : Z_{R'})$

We have

$$M'_T = \begin{pmatrix} -\sqrt{-1}\alpha & \beta \\ \sqrt{-1}\beta & -\alpha \end{pmatrix}.$$

From the above discussion, we can construct an algorithm to compute $x(R)$ and $x(R')$ from $(\theta_{E\times E',i}((R,R')))_i$ (Algorithm 5).