IS-CUBE: An isogeny-based compact KEM using a boxed SIDH diagram

Tomoki Moriya

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

Abstract. Isogeny-based cryptography is one of the candidates for post-quantum cryptography. One of the benefits of using isogeny-based cryptography is its compactness. In particular, a key exchange scheme SIDH forgave us to use a 4λ-bit prime for the security parameter λ.
Unfortunately, SIDH was broken in 2022 by some studies. After that, some isogeny-based key exchange and public key encryption schemes have been proposed; however, most of these schemes use primes whose sizes are not guaranteed as linearly related to the security parameter λ. As far as we know, the rest schemes have not been implemented due to the computation of isogenies of high dimensional abelian varieties, or they need to use a "weak" curve (*i.e.*, a curve whose endomorphism ring is known) as the starting curve.
In this study, we propose a novel compact isogeny-based key encapsulation mechanism named IS-CUBE via Kani's theorem and a 3-dimensional SIDH diagram. A prime used in IS-CUBE is of the size of about 8λ bits, and its starting curve is a random supersingular elliptic curve. The core idea of IS-CUBE comes from the hardness of some already known computational problems and the novel computational problem (the Long Isogeny with Torsion (LIT) problem), which is the problem to compute a hidden isogeny from given two supersingular elliptic curves and information of torsion points of relatively small order. From our PoC implementation of IS-CUBE via sagemath, it takes about 4.34 sec for the public key generation, 0.61 sec for the encapsulation, and 17.13 sec for the decapsulation if $\lambda = 128$.

Keywords: isogeny-based cryptography; Kani's theorem; SIDH; KEM

1 Introduction

Isogeny-based cryptography is considered one of the candidates for post-quantum cryptography. Some cryptographers are interested in isogeny-based cryptography because the key lengths of isogeny-based cryptosystems are small, and these cryptosystems have mathematical structures that those of the other post-quantum candidates do not have. In particular, SIDH key exchange [21] was known as a compute key exchange scheme, and SIKE [1], a key encapsulation mechanism based on SIDH remained in the 4th round of the NIST PQC standardization process as an alternative candidate [32]. Though SIDH was unfortunately broken in 2022 by the attacks based on Kani's theorem [5,25,37], an isogeny-based digital signature scheme SQISign[16] was submitted to the NIST PQC standardization process.

Though SIDH and SIKE were broken, some other isogeny-based key exchange, public key encryption, or key encapsulation mechanism schemes have been proposed. CSIDH key exchange [7] was proposed in 2018 by Castryck, Lange, Martindale, Panny, and Renes. In 2022, a key exchange scheme M-SIDH [18] was proposed as a countermeasure scheme of the SIDH attacks. As one direction of improvements of M-SIDH, Basso and Fouotsa proposed binSIDH and terSIDH [3]. FESTA [4] is an isogeny-based public key encryption scheme proposed by Basso, Maino, and Pope that uses SIDH attacks for the trapdoor. Moreover, QFESTA [31] was proposed by Nakagawa and Onuki very recently as an improvement of FESTA.

The size of the primes used in most of the above key exchange and public key encryption schemes are not guaranteed as linearly related to the security parameter λ, while the prime used in SIDH was guaranteed as about a 4λ-bit prime. One exception is FESTA with the computation of high-dimensional isogenies. It uses the prime of the size of about 7λ bits; however, it has no implementation due to the computation of high-dimensional isogenies. The other exception is QFESTA. It uses the prime of the size of about 2λ bits; however, the starting curve of QFESTA is an elliptic curve such that the structure of its endomorphism ring is well-known. There are some attacks for schemes using such elliptic curves (*e.g.,* [18, §4.2] and [8]). Though these attacks do not seem to be adaptable for QFESTA so far, we have the following research question:

> *Can we construct a (key exchange/public key encryption/key encapsulation mechanism) scheme that uses a prime of the size linearly related to the security parameter λ and uses a random elliptic curve as the starting curve?*

1.1 Contributions

We propose a novel key encapsulation mechanism named IS-CUBE. IS-CUBE is one answer to the above research question. In other words, IS-CUBE uses a prime of the size of about 8λ bits for the security parameter λ and uses a random elliptic curve as the starting curve.

IS-CUBE is constructed by a 3-dimensional SIDH diagram and Kani's theorem. The rough outline of IS-CUBE is as follows:

1. Alice constructs a SIDH diagram as a top surface.
2. Bob computes vertical isogenies from the top surface and obtains the bottom surface.
3. Bob publishes two elliptic curves of the bottom surface.
4. Alice recovers the bottom surface from the two elliptic curves by using Kani's theorem.

To realize the above construction, we use some techniques proposed in FESTA and introduce a novel computational problem, the LIT (Computational long isogeny with torsion) problem. The LIT problem is the problem to compute a hidden (long) isogeny from given two elliptic curves and images of torsion points whose orders are relatively small.

From our proof-of-concept implementation of IS-CUBE via sagemath, it takes about 4.34 sec for the public key generation, 0.61 sec for the encapsulation, and 17.13 sec for the decapsulation if $\lambda = 128$. Table 4 summarizes the more detailed computational time of IS-CUBE under our PoC implementation.

Organization.

In Section 2, we introduce some mathematical and cryptographical concepts as preliminaries. In Section 3.1, we introduce techniques to construct SIDH diagrams with masking torsion point information. In Section 3.2, we provide the outline of the construction of IS-CUBE. In Section 3.3, 3.4, and 3.5, we discuss some methods to realize IS-CUBE. Section 3.6 provides the precise scheme of IS-CUBE. We give security analyses of IS-CUBE in Section 4. In particular, we give the size of primes used in IS-CUBE in Section 4.2. Section 5 discusses some variations of IS-CUBE. In Section 6, we explain our PoC implementation of IS-CUBE. In Section 6.1, we give the primes used in IS-CUBE. We explain techniques used in our implementation in Section 6.2, and we show the computational time of IS-CUBE in Section 6.3 Finally, we conclude this paper in Section 7.

2 Preliminaries

We introduce some basic knowledge related to our study.

2.1 Principally polarized abelian varieties and isogenies

In this subsection, we introduce some mathematical concepts and facts about principally polarized abelian varieties and isogenies related to isogeny-based cryptography. Refer to [27], [28], and [38] for more details about these topics.

Let k be a field. We denote the characteristic of k by $\mathrm{ch}(k)$. An *abelian variety over* k is a projective algebraic variety over k that has a group structure whose law is defined by regular functions. Let A be an abelian variety, and let $\mathrm{Pic}^0(A)$ be the Picard variety of A. For a divisor D of A, there is a map $\Phi_D \colon A \to \mathrm{Pic}^0(A)$ defined by $x \mapsto t_x^* D - D$, where t_x is a translation map $t_x \colon A \to A; P \mapsto P + x$. A *pricipally polarized abelian variety over* k is a set of an abelian variety A over k and its divisor D such that $\Phi_D \colon A \to \mathrm{Pic}^0(A)$ is an isomorphism. An *elliptic curve over* k is a principally polarized abelian variety over k of dimension 1. In this paper, we often represent a principally polarized abelian variety without its divisor.

Let C be a smooth curve of genus g. The *Jacobian variety of* C is a principally polarized abelian variety that is a set of the abelian variety $\mathrm{Pic}^0(C)$ and its divisor $((g-1)\alpha_{D_0}(C))$, where D_0 is a divisor of C of $\deg D_0 = 1$ and α_{D_0} is an embedding $\alpha_{D_0}\colon C \hookrightarrow \mathrm{Pic}^0(C);\ P \mapsto (P) - D_0$. The moduli space of curves is embedded into that of principal polarized abelian varieties by considering Jacobian varieties. Moreover, a principally polarized abelian variety of dimension 2 over $\overline{\mathbb{F}_p}$ is isomorphic to a Jacobian variety of a curve of genus 2 or a product of two elliptic curves.

An *isogeny* is a surjective morphism between abelian varieties whose kernel is a finite group. The *degree of an isogeny* ϕ is the degree of ϕ as a morphism of algebraic varieties and is denoted by $\deg\phi$. The *dual isogeny of* $\phi\colon A \to B$ is an isogeny $\hat{\phi}\colon B \to A$ such that $\hat{\phi}\circ\phi\colon A \to A$ and $\phi\circ\hat{\phi}\colon B \to B$ are the multiplication-by-$(\deg\phi)$ maps. We say that an isogeny is *separable* if the isogeny is separable as a morphism of algebraic varieties. If $\deg\phi$ is coprime to $\mathrm{ch}(k)$ or $\mathrm{ch}(k) = 0$ for an isogeny ϕ over k, the isogeny ϕ is separable. If an isogeny ϕ is separable, we have $\deg\phi = \#\ker\phi$. Let G be a finite subgroup of an abelian variety A. There is a separable isogeny $\phi\colon A \to B$ such that $\ker\phi = G$. The codomain B is unique up to isomorphism and is denoted by A/G.

Let A be an abelian variety of dimension g over k. The *n-torsion subgroup of* A is the subgroup of A defined by $A[n] = \{P \mid [n]P = 0\}$, where $[n]$ is the multiplication-by-n map of A. If n is coprime to $\mathrm{ch}(k)$ or $\mathrm{ch}(k) = 0$, it holds that $A[n] \cong (\mathbb{Z}/n\mathbb{Z})^{2g}$. In particular, for an elliptic curve E, it holds that $E[n] \cong (\mathbb{Z}/n\mathbb{Z})^2$. We denote the Weil pairing on $A[n]$ by $e_N\colon A[n]\times A[n] \to \mu_n$, where μ_n is the group of nth roots. Refer [27, §13] for the definition of the Weil pairing. It holds that $e_n(\phi(P), \phi(Q)) = e_n(P, Q)^{\deg\phi}$ for an isogeny $\phi\colon A \to B$. An *isotropic subgroup of* $A[n]$ is a subgroup of $A[n]$ on which the Weil pairing e_n is trivial. A separable isogeny from A whose kernel is a maximal isotropic subgroup of $A[n]$ is called an *$(n, \dots, n)$-isogeny.* Let (A, D) be a principally polarized abelian variety (A, D), let G be a maximal isotropic subgroup G of $A[n]$, and let ϕ be an $(n, \dots, n)$-isogeny with $\ker\phi = G$. Then, there is a unique principally polarized abelian variety $(A/G, D')$ such that $\phi^* D' = nD$ up to isomorphism.

From an abelian variety A and its finite subgroup G of smooth order, we can compute a separable isogeny $\phi\colon A \to A/G$ with $\ker\phi = G$. If A is an elliptic curve, we can use Vélu's formulas to compute the isogeny [41]. If A is a principally polarized abelian variety of dimension 2 (*i.e.,* a Jacobian variety of a curve of genus 2 or a product of two elliptic curves) and G is an isotropic group of order a power of 2, we can use the algorithm to compute the Richelot isogenies to compute the isogeny [39].

The *j-invariant* is an invariant of isomorphism classes of elliptic curves. An elliptic curve E over k is isomorphic to E' over the algebraic closure of k if and only if $j(E) = j(E')$. Let p be a prime and let k be a field of characteristic p. We say that an elliptic curve E is *supersingular* if $E[p] = \{0\}$. If E is supersingular, then E is isomorphic to a curve defined over $\mathbb{F}_{p^2}$. If E is a supersingular elliptic curve over $\mathbb{F}_p$, it holds that $\#E(\mathbb{F}_{p^2}) = (p+1)^2$. A *superspecial curve* is a curve whose Jacobian variety is isomorphic to a product of supersingular elliptic curves as an abelian variety.

2.2 Kani's theorem

In this subsection, we introduce Kani's theorem [23].

Definition 1 (Isogeny diamond (SIDH diagram)). *Let k be a field, let E be an elliptic curve, and let G_1 and G_2 be cyclic finite subgroups of E such that* $\gcd(\#G_1, \#G_2) = 1$*. Then, there is a diagram of isogenies as follows:*

$$\begin{tikzcd} E \arrow[r, "\phi_1"] \arrow[d, "\phi_2"'] & E/G_1 \arrow[d, "\phi_2'"] \\ E/G_2 \arrow[r, "\phi_1'"'] & E/(G_1+G_2) \end{tikzcd}$$

Here, we have $\ker\phi_1 = G_1$, $\ker\phi_2 = G_2$, $\ker\phi_1' = \phi_2(G_1)$*, and* $\ker\phi_2' = \phi_1(G_2)$*.*

We call the above diagram an isogeny diamond or a SIDH diagram.

Theorem 1 (Kani's theorem [23]). *Suppose that there is an isogeny diamond as follows:*

$$\begin{tikzcd} E \arrow[r, "\phi_1"] \arrow[d, "\phi_2"'] & E_1 \arrow[d, "\phi_2'"] \\ E_2 \arrow[r, "\phi_1'"'] & E' \end{tikzcd}$$

Then, there is an isogeny $\Psi\colon E_1 \times E_2 \to E \times E'$ *defined by*

$$\Psi = \begin{pmatrix} \hat{\phi_1} & \hat{\phi_2} \\ -\phi_2' & \phi_1' \end{pmatrix}.$$

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

2.3 Key encapsulation mechanism

In this subsection, we define a key encapsulation mechanism and its security models.

Definition 2 (Key encapsulation mechanism (KEM)). *An algorithm* $\mathsf{KEM}(\lambda)$ *is called a key encapsulation mechanism (KEM) if it consists of the following three probabilistic polynomial time (PPT) algorithms:*

KeyGen: *It outputs a public parameter* **params***, a public key* **pk***, a secret key* **sk***, and a key space* $\mathcal{K}$ *from the security parameter* λ*.*

Encap: *It outputs a ciphertext* c *and a shared key* K *from the public key* **pk** *and the public parameter* **params***.*

Decap: *It outputs a shared key K' from the ciphertext c, the secret key* $\mathbf{sk}$*, and the public parameter* $\mathbf{params}$.

If it holds that $K = K'$, we say $\mathsf{KEM}(\lambda)$ *is correct.*

Definition 3 (OW-CPA security). *We say a correct KEM* $\mathsf{KEM}(\lambda)$ *is OW-CPA secure if, for any PPT algorithm $\mathcal{A}$, it holds that*

$$\Pr\left[K = K^* \;\middle|\; \begin{array}{l} (\mathbf{params}, \mathbf{pk}, \mathbf{sk}, \mathcal{K}) \leftarrow \mathsf{KeyGen}(\lambda), \\ (c, K) \leftarrow \mathsf{Encap}(\mathbf{pk}, \mathbf{params}), \\ K^* \leftarrow \mathcal{A}(\mathbf{pk}, c, \mathbf{params}) \end{array} \right] < \mathrm{negl}(\lambda).$$

Definition 4 (IND-CPA security). *We say a correct KEM* $\mathsf{KEM}(\lambda)$ *is IND-CPA secure if, for any PPT algorithm $\mathcal{A}$, it holds that*

$$\left| \Pr\left[i = i^* \;\middle|\; \begin{array}{l} (\mathbf{params}, \mathbf{pk}, \mathbf{sk}, \mathcal{K}) \leftarrow \mathsf{KeyGen}(\lambda), \\ (c, K_0) \leftarrow \mathsf{Encap}(\mathbf{pk}, \mathbf{params}),\; K_1 \xleftarrow{\$} \mathcal{K}, \\ i \xleftarrow{\$} \{0,1\},\; i^* \leftarrow \mathcal{A}(\mathbf{pk}, c, \mathbf{params}, K_i) \end{array} \right] - \frac{1}{2} \right| < \mathrm{negl}(\lambda),$$

where the arrow $\xleftarrow{\$}$ represents a uniformly and randomly sampling.

Definition 5 (IND-CCA security). *We say a correct KEM* $\mathsf{KEM}(\lambda)$ *is IND-CCA secure if, for any PPT algorithm $\mathcal{A}$, it holds that*

$$\left| \Pr\left[i = i^* \;\middle|\; \begin{array}{l} (\mathbf{params}, \mathbf{pk}, \mathbf{sk}, \mathcal{K}) \leftarrow \mathsf{KeyGen}(\lambda), \\ (c, K_0) \leftarrow \mathsf{Encap}(\mathbf{pk}, \mathbf{params}),\; K_1 \xleftarrow{\$} \mathcal{K}, \\ i \xleftarrow{\$} \{0,1\},\; i^* \leftarrow \mathcal{A}^{O(\cdot)}(\mathbf{pk}, c, \mathbf{params}, K_i) \end{array} \right] - \frac{1}{2} \right| < \mathrm{negl}(\lambda),$$

where $O(\cdot)$ is the decapsulation oracle that outputs $\mathsf{Decap}(\mathbf{sk}, \mathbf{params}, c^*)$ *from a ciphertext c^* other than c.*

3 IS-CUBE

In this section, we explain the construction of IS-CUBE.

3.1 Techniques for masking information of torsion points

In this subsection, we introduce some techniques to mask information about torsion points under a secret isogeny.

The breaking news of breaking SIDH showed that the Isogeny Problem with torsion point images of proper order is no longer secure. In other words, from given elliptic curves E, E' and bases $\{P, Q\}$, $\{\phi(P), \phi(Q)\}$, one can compute a hidden isogeny $\phi\colon E \to E'$ in a proper setting. However, these torsion point images play an important role in constructing a SIDH diagram (*i.e.,* an isogeny diamond). Recently, some techniques have been proposed to realize a SIDH diagram despite masking a part of the information on these torsion points. In this subsection, we explain three techniques.

The first technique is provided in [18]. For an isogeny $\phi\colon E \to E'$ and points $P, Q \in E$ and $\phi(P), \phi(Q) \in E'$, we mask the information by multiplying a hidden scalar α to image points $\phi(P), \phi(Q)$ as $\alpha\phi(P), \alpha\phi(Q)$. Here, an order of P and Q need to have a sufficient number of prime factors for security.

The second technique is provided in [4]. For $(\phi, P, Q, \phi(P), \phi(Q))$, we mask the information by multiplying a hidden 2×2-matrix to ${}^t(\phi(P), \phi(Q))$. As the first technique is considered to multiply a matrix αI_2, this technique is considered as a generalization of the first technique. If there is no restriction for matrices used in this technique, then the information on image points is completely hidden. Unfortunately, we need a restriction for these matrices for the construction of cryptographical schemes (typically, we use a matrix in a fixed abelian subgroup of the 2×2-linear group).

The third technique is the original technique of this paper. This technique does not operate image points but sets $\deg\phi \gg \operatorname{ord}(P)$. In Robert's attack and other SIDH attacks, we compute an isogeny $\varPhi$ of abelian varieties of dimension 2, 4, or 8 that has the information of ϕ. In other words, these attacks need sufficient information to compute $\varPhi$. Indeed, Robert's attack requests $\deg\phi \geq \operatorname{ord}(P)^2$. Therefore, if it holds that $\deg\phi \gg \operatorname{ord}(P)$ (*e.g.*, $\deg\phi \approx \operatorname{ord}(P)^3$ or $\deg\phi \approx \operatorname{ord}(P)^{100}$), Robert's attack and other SIDH attacks do not seem to work.

We use the second and third techniques to construct IS-CUBE in the following subsections.

3.2 Core idea

In this subsection, we explain the core idea of IS-CUBE. The precise scheme of IS-CUBE is in Section 3.6. We assume that Bob tries to send a shared key to Alice.

Let p be a prime such that $p = \ell_C^c \ell_A \cdot \ell_B^b f - 1$ and a be an integer, where ℓ_A, ℓ_B, ℓ_C are small distinct primes, $\ell_C^c > \ell_A^a$, $\ell_A^a \gg \ell_B^b$, and f is a small integer. Let $\mathcal{M}_c$ be an abelian subgroup of the 2×2-linear group over $\mathbb{Z}/\ell_C^c\mathbb{Z}$. The following diagram shows the structure of the core idea of IS-CUBE:

$$\begin{tikzcd}
 & E_s \arrow[rr] \arrow[dl] \arrow[dd, dashed] & & E_1 \arrow[dl, dotted] \arrow[dd, dashed] \\
\tilde{E}_s \arrow[rr, dotted] \arrow[dd, dashed] & & E_2 \arrow[dd, dotted] & \\
 & E & & E_1' \arrow[ll, squiggly] \arrow[dl, squiggly] \\
E_s' \arrow[ur, squiggly] \arrow[rr, squiggly] & & E_2' &
\end{tikzcd}$$

As shown in the above diagram, IS-CUBE is constructed by four SIDH diagrams.

The upper diagram (the following diagram) is used for computing the public key of Alice.

$$\begin{tikzcd}
E_s \arrow[r, "\phi_1"] \arrow[d, "\tau"'] & E_1 \arrow[d, dotted] \\
\tilde{E}_s \arrow[r, dotted] & E_2
\end{tikzcd}$$

The elliptic curve E_s is a supersingular elliptic curve. Alice first takes an isogeny $\tau\colon E_s \to \tilde{E}_s$ with $\deg\tau = \ell_C^c - \ell_A^a$ and computes an isogeny $\phi_1\colon \tilde{E}_a \to E_1$ of $\deg\phi_1 = \ell_A^a$ by using the method of the CGL hash function [9]. Then, there is a SIDH diagram as above. Alice does not compute isogenies denoted by dotted arrows. Second, she computes points

$$\phi_1(P_B), \phi_1(Q_B) \text{ and } {}^t(P_1, Q_1) = \mathbf{A} \cdot {}^t(\phi_1(P_C), \phi_1(Q_C)),$$

where $\{P_B, Q_B\}$ is a basis of $E_s[\ell_B^b]$, $\{P_C, Q_C\}$ is a basis of $E_s[\ell_C^c]$, and $\mathbf{A}$ is a matrix in $\mathcal{M}_c$. She outputs

$$(\tau, (P_B, Q_B), (P_C, Q_C), E_1, (\phi_1(P_B), \phi_1(Q_B)), (P_1, Q_1))$$

as her public key. Let $(\phi_1, \mathbf{A})$ be her secret key.

The middle diagrams (the following diagrams) are for Bob's ciphertext and the shared key.

$$\begin{tikzcd}
\tilde{E}_s \arrow[d, dashed, "\phi_{0,B}"] & E_s \arrow[l, "\tau"'] \arrow[r, "\phi_1"] \arrow[d, dashed, "\phi_B"] & E_1 \arrow[d, dashed, "\phi_{1,B}"] \\
E'_s & E & E'_1
\end{tikzcd}$$

Bob takes a random value r in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$ and computes three isogenies

$$\begin{aligned}
\phi_{0,B}\colon \tilde{E}_s &\longrightarrow E'_s = \tilde{E}_s/\langle \tau(P_B) + r\tau(Q_B)\rangle, \\
\phi_{1,B}\colon E_1 &\longrightarrow E'_1 = E_1/\langle \phi_1(P_B) + r\phi_1(Q_B)\rangle, \\
\phi_B\colon E_s &\longrightarrow E = E_s/\langle P_B + rQ_B\rangle,
\end{aligned}$$

where $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$ is the unit group of $(\mathbb{Z}/\ell_B^b\mathbb{Z})$. Bob also computes four points ${}^t(P'_0, Q'_0) = \mathbf{B} \cdot {}^t(\phi_{0,B}(\tau(P_C)), \phi_{0,B}(\tau(Q_C)))$
and ${}^t(P'_1, Q'_1) = \mathbf{B} \cdot {}^t(\phi_{B,1}(P_1), \phi_{B,1}(Q_1))$, where $\mathbf{B}$ is a matrix in $\mathcal{M}_c$. He publishes

$$(E'_s, (P'_0, Q'_0), E'_1, (P'_1, Q'_1))$$

as a ciphertext. The value $j(E)$ is the shared key.

The bottom diagram (the following diagram) is for Alice's shared key.

$$\begin{tikzcd}
E'_s \arrow[r, squiggly] \arrow[d, squiggly] & E'_2 \\
E & E'_1 \arrow[l, squiggly] \arrow[u, squiggly]
\end{tikzcd}$$

The most important idea of this part comes from FESTA [4]. Alice computes ${}^t(P_1'', Q_1'') = \mathbf{A}^{-1} \cdot {}^t(P_1', Q_1')$. Note that

$$\begin{pmatrix} P_1'' \\ Q_1'' \end{pmatrix} = \mathbf{B} \begin{pmatrix} \phi_{1,B}(\phi_1(P_C)) \\ \phi_{1,B}(\phi_1(Q_C)) \end{pmatrix}$$

since $\mathbf{AB} = \mathbf{BA}$. Therefore, if $\deg \tau$ is a proper integer, Alice can compute an isogeny $E_s' \times E_1' \to E \times E_2'$ by using Kani's theorem and $(P_0', Q_0', P_1'', Q_1'')$. Let $j(E)$ be her shared key.

To construct a key encapsulation mechanism based on the above idea, we need to answer the following two questions:

- How do we construct an isogeny τ?
- Is it secure to use the curve with an endomorphism of a small degree like the curve of j-invariant 1728 as the starting curve?

We explain the way to construct τ that has the desired property in Section 3.3. Moreover, we discuss the security of IS-CUBE if we use the curve with an endomorphism of a small degree as the starting curve in Section 3.4, and explain the way to set up the public parameters for IS-CUBE with a random starting curve in Section 3.5.

3.3 Construction of τ

In this subsection, we explain how to construct an isogeny τ. To realize IS-CUBE, the construction of τ is the most non-trivial part. From Kani's theorem, the value $\deg \tau$ needs to be $\ell_C^c - \ell_A^a$ or $(\ell_C^c - \ell_A^a)/\ell$ for some small integer ℓ. However, it is not easy to construct τ of degree $\ell_C^c - \ell_A^a$ for random c and b. We provide two methods to construct τ as follows:

- Use a factorization of $x^{2^*} - y^{2^*}$.
- Compute an endomorphism of the curve of j-invariant 1728.

The first method provides an easy construction of τ. This construction is efficient even if the size of the prime p is too large; however, we need to take a larger p than the smallest prime that makes IS-CUBE secure. The second method uses a part of the KLPT algorithm [24]. This method is more complicated than the first method; however, we can take the smallest (or a near-size) prime p that makes IS-CUBE secure.

Use a factorization of $\boldsymbol{x^{2^*} - y^{2^*}}$. We define $\ell_C = 2$, $\ell_A = 3$, $\ell_B = 7$, and the prime p as

$$p = 2^{2^{a'+1}} 3 \cdot 7^b f - 1,$$

and set $a = 2^{a'}$, where f is a small integer. Set E_s as the curve of j-invariant 1728. In this case, it holds that

$$2^{2^{a'+1}} - 3^{2^{a'}} = (2^2 - 3)(2^2 + 3) \prod_{i=1}^{a'-1} (2^{2^{i+1}} + 3^{2^i}) = 7 \cdot \prod_{i=1}^{a'-1} (2^{2^{i+1}} + 3^{2^i}).$$

Since E_s is the curve of j-invariant 1728, we can construct endomorphisms of degree $2^{2^2}+3^2,\dots,2^{2^{a'}}+3^{2^{a'-1}}$ respectively. Specifically, an endomorphism of E_0 defined by $[3^{2^{i-1}}]+[2^{2^i}]\circ\iota$ is of degree $2^{2^{i+1}}+3^{2^i}$, where ι is an endomorphism satisfying $\iota^2=[-1]$. Therefore, if we define τ as

$$\tau=([3]+[2^2]\circ\iota)\circ\cdots\circ([3^{2^{a'-2}}]+[2^{2^{a'-1}}]\circ\iota),$$

then it holds that $\deg\tau\cdot 7=2^{2^{a'+1}}-3^{2^{a'}}$.

Remark 1. If it holds that $7|\deg\tau$, then the problem occurs because the image of a basis of $E_s[7^b]$ under τ is not a basis of $E_s[7^b]$. This never happens because we have $\alpha^2+\beta^2\not\equiv 0 \pmod q$ for all integers α,β with $q\nmid\alpha,\beta$ if $q\equiv 3 \pmod 4$.

Use an endomorphism of the curve of *j*-invariant 1728. We now explain the second method to construct τ.

We define $\ell_C=2$, $\ell_A=3$, and $\ell_B=5$. In other words, we take p as

$$p=2^c3\cdot 5^bf-1,$$

and set $\deg\phi_1=3^a$, where f is a small integer. The problem is the way to construct an isogeny of degree 2^c-3^a. Let E_0 be the curve of j-invariant 1728 over $\mathbb{F}_p$, π_p be the p-Frobenius map of E_0, and ι be the endomorphism of E_0 such that $\iota^2=[-1]$. There is a well-known algorithm to compute four integers z_1,z_2,z_3,z_4 satisfying

$$z_1^2+z_2^2+p(z_3^2+z_4^2)=N$$

from a given integer $N>p$ (*e.g.*, [16, Algorithm 1]). Therefore, we can compute an endomorphism γ of E_0 of degree N by computing

$$\gamma=[z_1]+[z_2]\iota+\pi_p([z_3]+[z_4]\iota).$$

We take the maximum integer M such that $M|(2^c-3^a)$ and $\gcd(M,5)=1$, and set $N=M\cdot 5^{2b}$. Compute an endomorphism γ of E_0 such that $\deg\gamma=N$ by the above method. The endomorphism γ is separable because $\gcd(p,\deg\gamma)=1$. Therefore, there are separable isogenies ψ_1, ψ_2, and τ such that $\deg\psi_1=\deg\psi_2=5^b$, $\deg\tau=M$, and $\gamma=\psi_2\circ\tau\circ\psi_1$. Since we have

$$\ker\psi_1=\ker\gamma\cap E_0[5^b],\quad \ker\hat{\psi}_2=\ker\hat{\gamma}\cap E_0[5^b],$$

we obtain $\tau\colon E_s\to\tilde{E}_s$ as the following diagram.

$$\begin{tikzcd} & E_0 \arrow[dl, "\psi_1"'] \arrow[dr, "\hat{\psi}_2"] & \\ E_s \arrow[rr, "\tau"'] & & \tilde{E}_s \end{tikzcd}$$

By using ψ_1, ψ_2 and γ, we can compute some images under τ. Technically, we can compute $\tau(P)$ for $P\in E_s$ such that the order of P is coprime to 5 by computing $\frac{1}{5^{2b}}(\hat{\psi}_2 \circ \gamma \circ \hat{\psi}_1)(P)$. However, if the order of P is divisible by 5, then we cannot use this method. This is a problem because we need to compute $\tau(P_B)$ and $\tau(Q_B)$ in the first setting of IS-CUBE, where $\{P_B, Q_B\}$ is a basis of $E_s[5^b]$. We solve this problem via Kani's theorem. We define p as

$$p = 2^c 3 \cdot 5^{b+t} f - 1,$$

and set $a = 2a'$ for an integer a', where t is the maximum integer satisfying $5^t \mid 2^c - 3^{2a'}$. Note that $\deg \tau = (2^c - 3^{2a'})/5^t$. Let $\{P_C, Q_C\}$ be a basis of $E_s[2^c]$. We assume that we already computed $\tau(P_C)$ ad $\tau(Q_C)$. Let ψ' be an isogeny $\psi' \colon \tilde{E}_s \to \tilde{E}'_s$ of degree 5^t. We have the following diagram.

$$\begin{tikzcd}
E_s \arrow[r, "\tau"] \arrow[d, "{[3^{a'}]}"'] & \tilde{E}_s \arrow[r, "\psi'"] & \tilde{E}'_s \arrow[d, "{[3^{a'}]}"] \\
E_s \arrow[rr, "\psi' \circ \tau"'] & & \tilde{E}'_s
\end{tikzcd}$$

Since $\deg\,[3^{a'}] + \deg\,(\psi' \circ \tau) = 2^c$, the kernel of the isogeny $\Psi \colon E_s \times \tilde{E}'_s \to E_s \times \tilde{E}'_s$ defined by

$$\Psi = \begin{pmatrix} [3^{a'}] & \hat{\tau} \circ \hat{\psi}' \\ \psi' \circ \tau & -[3^{a'}] \end{pmatrix}$$

is $\langle (3^{a'} P_C, \psi'(\tau(P_C))), (3^{a'} Q_C, \psi'(\tau(Q_C))) \rangle$ from Kani's theorem. Therefore, we can compute the images of the points in $E_s[5^{b+t}]$ under $\psi' \circ \tau$ by using P_C, Q_C, $\tau(P_C)$, and $\tau(Q_C)$. Let $\{P_B, Q_B\}$ be a basis of $E_s[5^b]$. To compute the images of P_B, Q_B under τ, we have the following process:

1. Take points $P'_B, Q'_B \in E_s[5^{b+t}]$ such that $5^t P'_B = P_B$ and $5^t Q'_B = Q_B$.
2. Compute $\psi' \circ \tau(P'_B)$ and $\psi' \circ \tau(Q'_B)$ by using Ψ.
3. Compute $\hat{\psi}' \circ \psi' \circ \tau(P'_B) = \tau(P_B)$ and $\hat{\psi}' \circ \psi' \circ \tau(Q'_B) = \tau(Q_B)$.

Hence, we can compute τ and desired image points by the above method.

Decapsulation. We now provide the precise method for the decapsulation assuming the above two methods for constructing τ.

Alice first computes a random 7-isogeny (resp. 5^t-isogeny) $E'_s \to E''_s$ and images of (P'_0, Q'_0) under this isogeny if we use the first method (resp. use the second method). Alice can compute an isogeny $E''_s \times E'_1 \to E \times E''_2$ by using Kani's theorem. The following diagram shows the SIDH diagram that Alice considers.

$$\begin{tikzcd}
& E''_s \arrow[rr, rightsquigarrow, "3^a\text{-isogeny}"] \arrow[ddl, rightsquigarrow, "(2^c - 3^a)\text{-isogeny}"] & & E''_2 \\
E'_s \arrow[ur, "7 \text{ or } 5^t\text{-isogeny}"] \arrow[d, "(\deg \tau)\text{-isogeny}"'] & & & \\
E & & E'_1 \arrow[ll, rightsquigarrow] \arrow[uur, rightsquigarrow] &
\end{tikzcd}$$

Thus, Alice can compute her shared key $j(E)$.

Remark 2. Alice has two elliptic curves E and E_2'' in the end. Therefore, it seems for Alice to be hard to determine the correct shared key. There are some solutions to this problem.

One solution is to use the Fujisaki-Okamoto transform [19]. As a result of the FO transform, Alice can obtain Bob's secret key in the decapsulation process; therefore, Alice can obtain the correct shared key E.

We also provide a solution to this problem without using the FO transform. Suppose we use the first method to construct τ or the second method with $t \geq 1$. Alice takes two different 7 or 5^t-isogenies $E_s' \to E_s''$ and $E_s' \to E_s'''$, and computes two $(2^c, 2^c)$-isogenies mapping from $E_s'' \times E_1'$ and from $E_s''' \times E_1'$. Let the images of these two isogenies be $E \times E_2''$ and $E \times E_2'''$. Because E_2'' and E_2''' are different supersingular elliptic curves with high probability, Alice can identify $j(E)$.

If we use the second method with $t = 0$, we take the smallest prime ℓ such that $\ell | (2^c - 3^a)$, and let p be a prime defined by

$$p = 2^c 3 \cdot 5^b \ell \cdot f - 1.$$

By setting $\deg \tau = (2^c - 3^a)/\ell$, Alice can use a similar method to that in the previous paragraph. In other words, Alice can identify $j(E)$ by using two random ℓ-isogenies from E_0'. The KEM explained in Section 3.6 uses this method to identify $j(E)$.

3.4 Vulnerability of using a starting curve with an endomorphism of a small degree

Though a curve with an endomorphism of a small degree (*e.g.,* the curve of j-invariant 1728) provides some useful constructions, it sometimes provides vulnerabilities for cryptographical schemes. For example, the torsion point attack provided by Petit uses an endomorphism of a small degree on the starting curve of SIDH [34].

We use the same notation as in Section 3.2. Assume that the starting curve $E_s/\mathbb{F}_{p^2}$ has an endomorphism θ with $\deg \theta \approx 1$. Let $\{P_C, Q_C\}$ be a basis of $E_s[\ell_C^c]$ and $\phi_1 \colon E_s \to E_1$ be a ℓ_A^a-isogeny. We assume that Alice publishes

$$(E_s, {}^t(P_s, Q_s) := \mathbf{A} \cdot {}^t(\phi_1(P_C)), \phi_1(Q_C)),$$

where $\mathbf{A} \in \mathcal{M}_c$. Then, we have the following theorem.

Theorem 2. *Let* $\mathbf{C}$ *be a* 2×2*-matrix over* $\mathbb{Z}/\ell_C^c\mathbb{Z}$ *representing* θ *with respect to the basis* $\{P_C, Q_C\}$*. I.e., we have* ${}^t(\theta(P_C), \theta(Q_C)) = \mathbf{C} \cdot {}^t(P_C, Q_C)$*. If* $\mathbf{C} \in \mathcal{M}_c$*, then there is a polynomial time algorithm that reveals* $\ker \phi_1$ *when the starting curve* E_s *has an endomorphism* θ *of a small degree.*

Proof. The construction of the algorithm to break IS-CUBE is based on the attack for M-SIDH when the starting curve has an endomorphism of a small degree [18, Section 4.2].

Let $\psi\colon E_1 \to E_1$ be an endomorphism of E_1 defined by $\psi = \phi_1 \circ \theta \circ \hat{\phi}_1$. We have

$$\psi \begin{pmatrix} P_s \\ Q_s \end{pmatrix} = \mathbf{B} \begin{pmatrix} \psi(\phi_1(P_C)) \\ \psi(\phi_1(Q_C)) \end{pmatrix} = \deg \phi_1 \mathbf{AC} \begin{pmatrix} \phi_{B,s}(P_C) \\ \phi_{B,s}(Q_C) \end{pmatrix} = \ell_A^a \mathbf{C} \begin{pmatrix} P_s \\ Q_s \end{pmatrix}$$

since $\mathbf{A}, \mathbf{C} \in \mathcal{M}_c$. Therefore, we can compute the images of (P_s, Q_s) under the endomorphism ψ. Since $\deg \psi \approx \ell_A^{2a}$ and $\ell_A^{2a} \leq \ell_C^{2c}$, Robert's attack [37] reveals $\ker \psi$. We can get $\ker \phi_1$ from $\ker \psi$. $\square$

We can see more details of this type of attack in [8].

By taking a proper basis, we can avoid the attack in Theorem 2. However, it is desired to take a random supersingular elliptic curve with less information about its endomorphism ring as the starting curve to avoid a potential security risk caused by using such "weak" elliptic curves. To construct a random supersingular elliptic curve without the knowledge of its endomorphism ring, we can use a similar method to that proposed in [2] based on the technique provided in the next subsection.

3.5 Public parameters with a random starting curve

From the above subsection, we have a vulnerability if we use a curve with a small-degree endomorphism as the starting curve of IS-CUBE. This subsection provides the method to construct the public parameters of IS-CUBE with a random starting curve. The method proposed in [2] provides a construction of a random supersingular elliptic curve by a multi-party. By combining this method and our computational method in this subsection, we can obtain the trusted public parameters of IS-CUBE.

We obtain the set

$$(E_{s,0}, P_C, Q_C, P_B, Q_B, \tilde{E}_{s,0}, \tau_0(P_C), \tau_0(Q_C), \tau_0(P_B), \tau_0(Q_B))$$

from the method in Subsection 3.3, where $\{P_C, Q_C\}$ is a basis of $E_{s,0}[\ell_C^c]$, $\{P_B, Q_B\}$ is a basis of $E_{s,0}[\ell_B^b]$, and τ_0 is an isogeny with a proper degree. Note that it holds that $\gcd(\deg \tau_0, \ell_B) = 1$ in the both methods to construct τ provided in Subsection 3.3. By using P_B and Q_B, we can construct the following SIDH diagram:

$$\begin{tikzcd}
(E_{s,0}, P_C, Q_C, P_B, Q_B) \arrow[r, "\tau_0"] \arrow[d, "\kappa_0"] & (\tilde{E}_{s,0}, \tau_0(P_C), \tau_0(Q_C), \tau_0(P_B), \tau_0(Q_B)) \arrow[d, "\tilde{\kappa}_0"] \\
(E_{s,1}, \kappa_0(P_C), \kappa_0(Q_C)) \arrow[r, "\tau_1"'] & (\tilde{E}_{s,1}, \tilde{\kappa}_0(\tau_0(P_C)), \tilde{\kappa}_0(\tau_0(Q_C)))
\end{tikzcd}$$

Here, κ_0 is an ℓ_B^b-isogeny mapping from $E_{s,0}$ and $\tilde{\kappa}_0$ is an isogeny mapping from $\tilde{E}_{s,0}$ with $\ker \tilde{\kappa}_0 = \tau_0(\ker \kappa_0)$. Therefore, we obtain random supersingular elliptic curves $E_{s,1}$ and $\tilde{E}_{s,1}$. For the public parameters of IS-CUBE, we additionally need to compute images of $E_{s,1}[\ell_C^c]$ and $E_{s,1}[\ell_B^b]$ under τ_1. The image of $E_{s,1}[\ell_C^c]$ is already computed as four points

$$(\kappa_0(P_C), \kappa_0(Q_C)), (\tilde{\kappa}_0(\tau_0(P_C)), \tilde{\kappa}_0(\tau_0(Q_C))).$$

However, these points reveal the isogeny κ_0 from Robert's attack. Therefore, we need to mask information about them. Take a random 2×2-regular matrix $\mathbf{A}$ over $\mathbb{Z}/\ell_C^c\mathbb{Z}$, and compute ${}^t(P_{C,1}, Q_{C,1}) = \mathbf{A} \cdot {}^t(\kappa_0(P_C), \kappa_0(Q_C))$ and ${}^t(\tilde{P}_{C,1}, \tilde{Q}_{C,1}) = \mathbf{A} \cdot {}^t(\tilde{\kappa}_0(\tau_0(P_C)), \tilde{\kappa}_0(\tau_0(Q_C)))$. Note that $\mathbf{A}$ does not have to belong to $\mathcal{M}_C$. By giving $(\tilde{P}_{C,1}, \tilde{Q}_{C,1})$ and $(\tilde{P}_{C,1}, \tilde{Q}_{C,1})$, we provide the image of $E_{s,1}[\ell_C^c]$ under τ_1 without the information of κ_0. Since a is an even integer for any cases, to compute the image of $E_{s,1}[\ell_B^b]$, we can use the same trick as that appearing in the construction of τ using an endomorphism of the curve of j-invariant 1728. In other words, by using a scalar multiplication $[\ell_A^{a/2}]$, four points $\tilde{P}_{C,1}, \tilde{Q}_{C,1}, \tilde{P}_{C,1}, \tilde{Q}_{C,1}$, and Kani's theorem, we can compute the image of a point in $E_{s,1}[\ell_B^b]$ under τ_1.

By repeating the above process enough times, we can obtain the public parameters of IS-CUBE with a random starting curve.

Remark 3. Suppose that we repeat the above process n times and obtain a random curve $E_{s,n}$. From the discussion in Subsection 4.2, the degree of the isogeny $E_{s,0} \to E_{s,n}$ is desired larger than $((\ell_C)^c)^{3/2}$. Therefore, we set n as an integer such that $(\ell_B)^{bn} \geq (\ell_C)^{3c/2}$.

3.6 Key excansulation mechanism

In this subsection, we explain the precise scheme of IS-CUBE.

Bob tries to send a shared key to Alice.

Public parameters: Take three integers a, b, c and primes ℓ_A, ℓ_B, ℓ_C and set p as a prime such that

$$p = \ell_C^c \ell_A \ell_B^b \ell f - 1,$$

where ℓ is an integer such that $\ell | (\ell_C^c - \ell_A^a)$ and $\gcd((\ell_C^c - \ell_A^a)/\ell, \ell_B) = 1$, and f is a small integer. Take a random supersingular elliptic curve E_s and an isogeny $\tau\colon E_s \to \tilde{E}_s$ of degree $(\ell_C^c - \ell_A^a)/\ell$. Let $\{P_B, Q_B\}$ be a basis of $E_s[\ell_B^b]$ and $\{P_C, Q_C\}$ be a basis of $E_s[\ell_C^c]$. Compute all images of these bases under τ by using methods in Section 3.3 and Section 3.5. Let $\mathcal{M}_c$ be an abelian subgroup of the 2×2-linear group over $\mathbb{Z}/\ell_C^c\mathbb{Z}$. For example, we can set

$$\mathcal{M}_c = \left\{ \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix} \middle| \, \alpha, \beta \in \mathbb{Z}/\ell_C^c\mathbb{Z}, \ \alpha^2 - \beta^2 \in (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times \right\}.$$

The public parameters of IS-CUBE are

$$(p, E_0, \{P_B, Q_B\}, \{\tau(P_B), \tau(Q_B)\}, \{P_C, Q_C\}, \{\tau(P_C), \tau(Q_C)\}, \mathcal{M}_c).$$

Public key (Alice): Alice computes a random isogeny $\phi_1\colon E_s \to E_1$ of degree ℓ_A^a by using the method of the CGL hash function. Moreover, she computes

$$(P_B', Q_B') = (\phi_1(P_B), \phi_1(Q_B)), \quad (\tilde{P}_1, \tilde{Q}_1) = (\phi_1(P_C), \phi_1(Q_C)).$$

She randomly takes a matrix $\mathbf{A}$ from $\mathcal{M}_c$. She finally computes

$$^t(P_1, Q_1) = \mathbf{A} \cdot {}^t(\tilde{P}_1, \tilde{Q}_1).$$

She publishes the following as her public key:

$$(E_1, (P_1, Q_1), (P_B', Q_B')).$$

She keeps $\mathbf{A}$ as her secret key.

Encapsulation (Bob): Bob takes a random value r in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$. He computes two ℓ_B^b-isogenies

$$\phi_{B,0}\colon \tilde{E}_s \longrightarrow E_s' = \tilde{E}_s/\langle \tau(P_B) + r\tau(Q_B)\rangle,$$
$$\phi_{B,1}\colon E_1 \longrightarrow E_1' = E_1/\langle P_B' + rQ_B'\rangle.$$

He also computes points

$$(\tilde{P}_0', \tilde{Q}_0') = (\phi_{B,0}(\tau(P_C)), \phi_{B,0}(\tau(Q_C))), \quad (\tilde{P}_1', \tilde{Q}_1') = (\phi_{B,1}(P_1), \phi_{B,1}(Q_1)).$$

He takes a random matrix $\mathbf{B}$ from $\mathcal{M}_c$. He finally computes

$$^t(P_0', Q_0') = \mathbf{B} \cdot {}^t(\tilde{P}_0', \tilde{Q}_0'), \quad {}^t(P_1', Q_1') = \mathbf{B} \cdot {}^t(\tilde{P}_1', \tilde{Q}_1').$$

He publishes the following as his ciphertext:

$$(E_s', (P_0', Q_0'), E_1', (P_1', Q_1')).$$

Bob also computes a ℓ_B^b-isogeny

$$\phi_B\colon E_s \longrightarrow E = E_s/\langle P_B + rQ_B\rangle.$$

The value $j(E)$ is Bob's shared key.

Decapsulation (Alice): Alice first computes points

$$^t(P_1'', Q_1'') = \mathbf{A}^{-1} \cdot {}^t(P_1', Q_1')$$

She also computes two random ℓ-isogenies $\psi_1\colon E_s' \to E_s''$ and $\psi_2\colon E_s' \to E_s'''$ and points

$$(P_0'', Q_0'') = (\psi_1(P_0'), \psi_1(Q_0')), \quad (P_0''', Q_0''') = (\psi_2(P_0'), \psi_2(Q_0')).$$

She next computes two (ℓ_C^c, ℓ_C^c)-isogenies mapping from $E_s'' \times E_1'$ and $E_s''' \times E_1'$ with kernels

$$\langle (P_0'', P_1'), (Q_0'', Q_1')\rangle \text{ and } \langle (P_0''', P_1'), (Q_0''', Q_1')\rangle$$

respectively. Alice gets four elliptic curves appearing as codomains of the two (ℓ_C^c, ℓ_C^c)-isogenies. If she cannot obtain four elliptic curves (*e.g.,* the case that one of the codomains of the isogenies is not a product of elliptic curves), then she refuses the sharing. She computes all j-invariants of these four elliptic curves. If two of these four values are the same, Alice takes this common value as her shared key, and if not, she refuses the decapsulation.

Theorem 3 (Correctness). *Alice and Bob obtain the same key at the end of IS-CUBE with high probability.*

Proof. From the commutativity of SIDH diagrams, we have there are two isogenies: $\tau' \colon E \to E_s'$ that has the same degree as τ and $\phi_1' \colon E \to E_1'$ of degree ℓ_A^a. Moreover, there are points P, Q in E such that

$$(P_0', Q_0') = (\tau'(P), \tau'(Q)),$$
$$(P_1', Q_1') = \mathbf{A} \cdot {}^t(\phi_1'(P), \phi_1'(Q)),$$

from the constructions of (P_0', Q_0') and (P_1', Q_1'). Therefore, we have

$$(P_1'', Q_1'') = (\phi_1'(P), \phi_1'(Q)).$$

Since we have $\ell \cdot \deg \tau' + \deg \phi_1' = \ell_C^c$, the codomain of the isogeny mapping from $E_s'' \times E_1'$ whose kernel is

$$\langle (\psi_1(P_0'), P_1''), (\psi_1(Q_0'), Q_1'') \rangle$$

is a product of E and E_2'' from Kani's theorem, where E_2'' is a supersingular elliptic curve. For the same reason, we have

$$(E_s''' \times E_1')/\langle (\psi_2(P_0'), P_1''), (\psi_2(Q_0'), Q_1'') \rangle \cong E \times E_2''',$$

where E_2''' is a supersingular elliptic curve. From the commutativity of SIDH diagrams, elliptic curves E_2'' and E_2''' are, respectively, codomains of different ℓ-isogenies mapping from the common domain curve E_2'. Therefore, it holds that $j(E_2'') \neq j(E_2''')$ with high probability. Hence, Alice gets $j(E)$ at the end of IS-CUBE, which is Bob's shared key. $\square$

4 Security analysis

In this section, we provide some discussions about the security of IS-CUBE.

4.1 Computational problems

We introduce some computational problems related to the security of IS-CUBE and prove the security of IS-CUBE. We conjecture that all problems provided in this subsection are computationally infeasible.

Let p be a prime of the form $\ell_C^c \ell_A \ell_B^b \ell f - 1$, let a be an integer with $\ell_C^c > \ell_A^a$, let E_s be a random supersingular elliptic curve over $\mathbb{F}_{p^2}$, let $\{P_C, Q_C\}$ (resp. $\{P_B, Q_B\}$) be a basis of $E_s[\ell_C^c]$ (resp. $E_s[\ell_B^b]$), let τ be an isogeny $\tau\colon E_s \to \tilde{E}_s$ of degree $(\ell_C^c - \ell_A^a)/\ell$, and let $\mathcal{M}_c$ be an abelian subgroup of the general linear group of degree 2 over $\mathbb{Z}/\ell_C^c\mathbb{Z}$. Let

$$(E_s, P_B, Q_B, P_C, Q_C, \tilde{E}_s, \tau(P_B), \tau(Q_B), \tau(P_C), \tau(Q_C))$$

be public parameters.

Problem 1 (CIST problem [4, Problem 7]). Let d be an integer coprime to ℓ_C. Let ϕ be an isogeny $\phi\colon E_s \to E_1$ of degree d. Compute ϕ from (E_s, P_C, Q_C) and $(E_1, \mathbf{A} \cdot {}^t(\phi(P_C), \phi(Q_C)))$, where $\mathbf{A}$ is an element in $\mathcal{M}_c$ taken uniformly at random.

Problem 2 (DIST problem [4, Problem 6]). Let d be an integer coprime to ℓ_C. Let (E_1, P_1, Q_1) be a tuple of a supersingular elliptic curve E_1 and a basis $\{P_1, Q_1\}$ of $E_1[\ell_C^c]$ sampled with probability 1/2 from one of the following distributions:

- (E_1, P_1, Q_1), where E_1 is a random supersingular elliptic curve over $\mathbb{F}_{p^2}$, and $\{P_1, Q_1\}$ is a random basis of $E_1[\ell_C^c]$.
- (E_1, P_1, Q_1), where E_1 is a codomain of an isogeny $\phi\colon E_s \to E_1$ of degree d, and ${}^t(P_1, Q_1) = \mathbf{A} \cdot {}^t(\phi(P_C), \phi(Q_C))$, where $\mathbf{A}$ is an element in $\mathcal{M}_c$ taken uniformly at random.

From tuples (E_1, P_1, Q_1) and (E_s, P_C, Q_C), determine from which distribution the tuple (E_1, P_1, Q_1) is sampled.

Problem 3 (Long isogeny with torsion (LIT) problem). Let d be an integer coprime to ℓ_B such that $d \gg \ell_B^b$. Let ϕ be an isogeny $\phi\colon E_s \to E_1$ of degree d. Compute ϕ from (E_s, P_B, Q_B) and $(E_1, \phi(P_B), \phi(Q_B))$.

Problem 4 (Decisional long isogeny with torsion (DLIT) problem). Let d be an integer coprime to ℓ_B such that $d \gg \ell_B^b$. Let (E_1, P_1, Q_1) be a tuple of a supersingular elliptic curve E_1 and a basis $\{P_1, Q_1\}$ of $E_1[\ell_B^b]$ sampled with probability 1/2 from one of the following distributions:

- (E_1, P_1, Q_1), where E_1 is a random supersingular elliptic curve over $\mathbb{F}_{p^2}$, and $\{P_1, Q_1\}$ is a random basis of $E_1[\ell_B^b]$ such that $e_{\ell_B^b}(P_1, Q_1) = e_{\ell_B^b}(P_B, Q_B)^d$.
- (E_1, P_1, Q_1), where E_1 is a codomain of an isogeny $\phi\colon E_s \to E_1$ of degree d, and $(P_1, Q_1) = (\phi(P_B), \phi(Q_B))$.

From tuples (E_1, P_1, Q_1) and (E_s, P_B, Q_B), determine from which distribution the tuple (E_1, P_1, Q_1) is sampled.

Problem 5 (Supersingular Isogeny Computational Diffie-Hellman with a revealed τ (SSICDH-t) problem). Let r be an element in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$ taken uniformly at random. Let $\phi_{0,B}\colon \tilde{E}_s \to E_s'$ be an ℓ_B^b-isogeny whose kernel is $\langle \tau(P_B) + r\tau(Q_B) \rangle$. From E_s' and public parameters, compute an elliptic curve $E_s/\langle P_B + rQ_B \rangle$.

Problem 6 (Supersingular Isogeny Decisional Diffie-Hellman with a revealed τ (SSIDDH-t) problem). Let r be an element in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$ taken uniformly at random. Let $\phi_{0,B}\colon \tilde{E}_s \to E'_s$ be an ℓ_B^b-isogeny whose kernel is $\langle \tau(P_B)+r\tau(Q_B)\rangle$. An elliptic curve E is sampled with probability 1/2 from one of the following distributions:

- E, where E is a random supersingular elliptic curve over $\mathbb{F}_{p^2}$.
- E, where E is $E_s/\langle P_B + rQ_B\rangle$.

From (E'_s, E) and public parameters, determine from which distribution E is sampled.

The core idea of IS-CUBE comes from the hardness of the above six problems. However, we need to introduce other problems to discuss the security of IS-CUBE due to its complex structure.

Problem 7 (Computational IS-CUBE (CIS-CUBE) problem). Let ϕ_1 be an ℓ_A^a-isogeny $\phi_1\colon E_s \to E_1$. Let r be an element in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$, and let $\mathbf{A}, \mathbf{B}$ be matrices in $\mathcal{M}_c$. Assume that these elements are taken uniformly at random. Define ℓ_B^b-isogenies and points of order ℓ_C^c as follows:

$$\begin{aligned}
&\phi_{0,B}\colon \tilde{E}_s \longrightarrow E'_s := \tilde{E}_s/\langle \tau(P_B) + r\tau(Q_B)\rangle,\\
&\phi_{1,B}\colon E_1 \longrightarrow E'_1 := E_1/\langle \phi_1(P_B) + r\phi_1(Q_B)\rangle,
\end{aligned}$$

$$\begin{aligned}
&{}^t(P_1, Q_1) := \mathbf{A}\cdot {}^t(\phi_1(P_C), \phi_1(Q_C)),\\
&{}^t(P'_0, Q'_0) := \mathbf{B}\cdot {}^t(\phi_{0,B}(\tau(P_C)), \phi_{0,B}(\tau(Q_C))),\\
&{}^t(P'_1, Q'_1) := \mathbf{B}\cdot {}^t(\phi_{1,B}(P_1), \phi_{1,B}(Q_1)).
\end{aligned}$$

From public parameters and

$$((E_1, P_1, Q_1, \phi_1(P_B), \phi_1(Q_B)), (E'_s, P'_0, Q'_0), (E'_1, P'_1, Q'_1)),$$

compute the elliptic curve $E_s/\langle P_B + rQ_B\rangle$.

Problem 8 (Decisional IS-CUBE (DIS-CUBE) problem). Let ϕ_1 be an ℓ_A^a-isogeny $\phi_1\colon E_s \to E_1$. Let r be an element in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$, and let $\mathbf{A}, \mathbf{B}$ be matrices in $\mathcal{M}_c$. Assume that these elements are taken uniformly at random. Define ℓ_B^b-isogenies and points of order ℓ_C^c as follows:

$$\begin{aligned}
&\phi_{0,B}\colon \tilde{E}_s \longrightarrow E'_s := \tilde{E}_s/\langle \tau(P_B) + r\tau(Q_B)\rangle,\\
&\phi_{1,B}\colon E_1 \longrightarrow E'_1 := E_1/\langle \phi_1(P_B) + r\phi_1(Q_B)\rangle,
\end{aligned}$$

$$\begin{aligned}
&{}^t(P_1, Q_1) := \mathbf{A}\cdot {}^t(\phi_1(P_C), \phi_1(Q_C)),\\
&{}^t(P'_0, Q'_0) := \mathbf{B}\cdot {}^t(\phi_{0,B}(\tau(P_C)), \phi_{0,B}(\tau(Q_C))),\\
&{}^t(P'_1, Q'_1) := \mathbf{B}\cdot {}^t(\phi_{1,B}(P_1), \phi_{1,B}(Q_1)).
\end{aligned}$$

An elliptic curve E is sampled with probability 1/2 from one of the following distributions:

– E, where E is a random supersingular elliptic curve over $\mathbb{F}_{p^2}$.
– E, where E is $E_s/\langle P_B + rQ_B \rangle$.

From public parameters and

$$((E_1, P_1, Q_1, \phi_1(P_B), \phi_1(Q_B)), (E'_s, P'_0, Q'_0), (E'_1, P'_1, Q'_1), E),$$

determine from which distribution E is sampled.

There are relationships among the above problems. We summarize some of them in the following proposition.

Proposition 1. *If there is a PPT algorithm that can solve the CIST or LIT or SSICDH-tT problem, then there is a PPT algorithm that can solve the CIS-CUBE problem. Moreover, if there is a PPT algorithm that can solve the SSIDDH-tT problem, then there is a PPT algorithm that can solve the DIS-CUBE problem.*

Finally, we have the following theorem for the security of IS-CUBE.

Theorem 4. *If there is no PPT algorithm to solve the CIS-CUBE problem, then IS-CUBE is OW-CPA secure. If there is no PPT algorithm to solve the DIS-CUBE problem, then IS-CUBE is IND-CPA secure.*

Proof. It is easy to see from the construction of IS-CUBE. □

IS-CUBE is not IND-CCA secure because there is an adaptive attack (see Section 4.3). To construct an IND-CCA KEM based on IS-CUBE, we need some transforms like the Fujisaki-Okamoto transform [19,22].

Further analysis of the security of IS-CUBE is a future work.

4.2 Secure prime for IS-CUBE

This subsection discusses the proper parameter size of p in IS-CUBE under the AES-λ security level (*i.e.,* λ bits security against attacks by classical computers and $\lambda/2$ bits security against attacks by quantum computers). In other words, we discuss the proper parameter under which the CIST problem (Problem 1) and LIT problem (Problem 3) and SSICDH-t problem (Problem 5) are hard to solve. We use the same notation as in Section 3.6 and 4.1. We need to decide the size of ℓ_C^c, ℓ_A^a, and ℓ_B^b.

We first focus on the CIST problem. Let d be the degree of the hidden isogeny ϕ. In the setting of IS-CUBE, we have $d = \ell_A^a$ or $d = \ell_B^b$. A secret matrix in $\mathcal{M}_c$ hides the image points under ϕ; therefore, we can assume that there is no information on image points under the secret isogeny. Hence, we can use the same discussion as that on the parameter size of SIDH. From the meet-in-the-middle approach and the Claw finding algorithm [40], we prefer to let $d \geq 2^{2\lambda}$. Moreover, we consider another approach to computing ϕ. From Robert's attack, it is sufficient to compute image points of order large enough under an isogeny to reveal it. In other words, it is suffice to find $\phi(P), \phi(Q)$ of order N to reveal a secret d-isogeny ϕ, where $\{P, Q\}$ is a basis of $E_s[N]$ and N is an integer such that $N \approx d^{1/2}$ and $\gcd(N, d) = 1$. The number of the bases of $(\mathbb{Z}/N\mathbb{Z})^2$ is $\#\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$. Since we can ignore a constant factor by using the Weil pairing, the number of bases we need to check is the number of the projective linear group $\#\mathrm{PGL}_2(\mathbb{Z}/N\mathbb{Z})$. Since it holds that

$$\#\mathrm{PGL}_2(\mathbb{Z}/N\mathbb{Z}) = N^3 \prod_{q|N \text{ prime}} \frac{1}{q^2}(q^2-1) > N^3 \prod_{q|N \text{ prime}} \frac{1}{q^2} \geq N,$$

we prefer to let $N \approx d^{1/2} \geq 2^{\lambda}$ against attacks by classical computers and $N^{1/2} \approx d^{1/4} \geq 2^{\lambda/2}$ against attacks by quantum computers from Grover's algorithm [20]. Hence, we conclude that we prefer to let $d \geq 2^{2\lambda}$. In other words, we have $\ell_A^a, \ell_B^b \geq 2^{2\lambda}$.

We next focus on the LIT problem. Let d be the degree of the hidden isogeny ϕ. In the setting of IS-CUBE, we have $d = \ell_A^a$. There are two directions to reveal the hidden d-isogeny ϕ based on Robert's attack as follows:

1. We first find a basis $\{P, Q\}$ of $E_s[\ell_B^b N]$ and $\phi(P), \phi(Q)$, where $NP = P_B$ and $NQ = Q_B$, and N is an integer coprime to d such that $(\ell_B^b N)^2 \geq d$. Then, we use Robert's attack to $(\phi, P, Q, \phi(P), \phi(Q))$.
2. By the similar computation of Robert's attack, we first compute two isogenies $E_s^m \to A$ and $E_1^m \to B$ of abelian varieties of dimension m by using $\{P_B, Q_B\}$ and $\{\phi(P_B), \phi(Q_B)\}$ for a proper integer m (*e.g.,* $m = 4, 8$). Using the meet-in-the-middle approach and the Claw finding algorithm, we find an isogeny $\Psi\colon A \to B$ that reveals an isogeny $\phi^m\colon E_s^m \to E_1^m$.

The main part of the first approach is to find two bases. Therefore, we can use the same discussion as in the previous paragraph because it holds that $E_s[\ell_B^b N]/E_s[\ell_B^b] \cong (\mathbb{Z}/N\mathbb{Z})^2$. We prefer to let $N \geq 2^{\lambda}$. Therefore, we prefer to let $d \geq (\ell_B^b \cdot 2^{\lambda})^2$. The second approach's main part is to detect the isogeny $\Psi\colon A \to B$. We have the degree of Ψ is about $(d/\ell_B^{2b})^m$. Put $L = d/\ell_B^{2b}$. The number of $(L^{1/2}, \ldots, L^{1/2})$-isogenies from A depends on m, and it is the minimum when $m = 1$. From the meet-in-the-middle approach and the Claw finding algorithm, we prefer to let $L \geq 2^{2\lambda}$ if $m = 1$; hence, we prefer to let $d \geq \ell_B^{2b} \cdot 2^{2\lambda}$. Since $\ell_B^b \geq 2^{2\lambda}$ from the previous paragraph, we prefer to let $d = \ell_A^a \geq 2^{6\lambda}$.

Finally, we focus on the SSICDH-t problem. Denote the target elliptic curve by E. We have two isogenies relating to E: an $(\ell_C^c - \ell_A^a)$-isogeny from E_s' to E and an ℓ_B^b-isogeny from E_s to E. The number of $(\ell_C^c - \ell_A^a)$-isogenies from E_s' is about $\ell_C^c - \ell_A^a$ and that of ℓ_B^b-isogenies from E_s is about ℓ_B^b. Therefore, we prefer to let $(\ell_C^c - \ell_A^a) \geq 2^{\lambda}$ and $\ell_B^b \geq 2^{\lambda}$. From the above discussions, we have $\ell_A^a \geq 2^{6\lambda}$ and $\ell_B^b \geq 2^{2\lambda}$. Therefore, we have already let $\ell_B^b \geq 2^{\lambda}$. Since $\ell_A^a \geq 2^{6\lambda}$ and $(\ell_C^c - \ell_A^a) \geq 2^{\lambda}$, we have prefer to let $\ell_C^c = (\ell_C^c - \ell_A^a) + \ell_A^a \geq 2^{6\lambda} + 2^{\lambda} \approx 2^{6\lambda}$.

From the above discussions, we have

$$\log_2 p \approx \log_2 \ell_C^c + \log_2 \ell_B^b \approx 6\lambda + 2\lambda = 8\lambda.$$

Remark 4. We provided a very rough estimation of the size of p that makes IS-CUBE secure in the above discussion. There is a possibility that the size can be smaller under a more precise estimation.

4.3 Adaptive attacks

In this subsection, we provide a brief explanation of adaptive attacks for IS-CUBE. IS-CUBE is vulnerable to adaptive attacks provided by an already-known attack for FESTA.

Suppose that Bob tries to obtain Alice's secret key. Recall that Alice computes ${}^t(P_1'', Q_1'') = \mathbf{A}^{-1} \cdot {}^t(P_1', Q_1')$ and an isogeny $E_s' \times E_1' \to E \times E_2'$ by using points $(P_0', Q_0', P_1'', Q_1'')$ to obtain the shared key $j(E)$. Note that (P_0', Q_0') and (P_1', Q_1') are provided by Bob. This situation is close to that of FESTA. Moreover, Alice cannot compute the matrix $\mathbf{B}$ of Bob. This situation satisfies the condition that the adaptive attack proposed in [29] can be used. Therefore, Bob can obtain Alice's secret key by an adaptive attack based on the attack proposed in [29].

For the above reason, IS-CUBE is vulnerable to an adaptive attack. If we want to use IS-CUBE non-interactively, we need a transform like the Fujisaki-Okamoto transform [19].

5 Variations of IS-CUBE

In this section, we provide some variations of IS-CUBE. In Section 5.1, we discuss about the compressed version of IS-CUBE. Moreover, we discuss the use of B-SIDH primes in Section 5.2.

5.1 Compressed version of IS-CUBE

In this subsection, we explain a compressed version of IS-CUBE. There are some studies about compression techniques for SIDH (*e.g.,* [13]). By using these techniques, IS-CUBE can also be compressed.

Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Let $\mathcal{A}$ be a deterministic algorithm that outputs a basis of $E[N]$ from E and N. There are some methods to construct such algorithms (*e.g.,* see [13]). A *canonical basis* of $E[N]$ is a basis of $E[N]$ derived from $\mathcal{A}$. Denote by $\{P_{E,C}, Q_{E,C}\}$ a canonical basis of $E[\ell_C^c]$, and by $\{P_{E,B}, Q_{E,B}\}$ a canonical basis of $E[\ell_B^b]$. Since ℓ_C^c and ℓ_B^b are smooth numbers, we can represent random bases of $E[\ell_C^c]$ and $E[\ell_B^b]$ by 2×2-matrices via the canonical bases and the Pohlig-Hellman algorithm [35]. In other words, a random basis $\{P_C, Q_C\}$ of $E[\ell_C^c]$ (resp. a random basis $\{P_B, Q_B\}$ of $E[\ell_B^b]$) can be represented by a matrix $\mathbf{C}$ in $\mathrm{GL}_2(\mathbb{Z}/\ell_C^c\mathbb{Z})$ such that ${}^t(P_C, Q_C) = \mathbf{C} \cdot {}^t(P_{E,C}, Q_{E,C})$ (resp. by a matrix $\mathbf{B}$ in $\mathrm{GL}_2(\mathbb{Z}/\ell_B^b\mathbb{Z})$ such that ${}^t(P_B, Q_B) = \mathbf{B} \cdot {}^t(P_{E,B}, Q_{E,B})$). Since $\{P_C, Q_C\}$ and $\{P_B, Q_b\}$ are represented by elements in $(\mathbb{F}_{p^2})^2$, we can compress these bases by using the matrix representations.

Moreover, we can compress these matrices additionally by considering the constant factors of these points. Recall that points P_B, Q_B are used to compute a kernel of an ℓ_B^b-isogeny. Since the kernel group is determined regardless of common constant factors of P_B and Q_B, we can use a matrix $\mathbf{B}'$ satisfying $\mathbf{B}' = \beta\mathbf{B}$ for any $\beta \in (\mathbb{Z}/\ell_B^b\mathbb{Z})^\times$ instead of $\mathbf{B}$. Therefore, we can represent

Table 1. Sets in which the public key and ciphertext of IS-CUBE are embedded

	Public key	Ciphertext
Original IS-CUBE	$(\mathbb{F}_{p^2})^5$	$(\mathbb{F}_{p^2})^6$
Compressed IS-CUBE	$\mathbb{F}_{p^2} \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^3 \times (\mathbb{Z}/\ell_B^b\mathbb{Z})^3$	$(\mathbb{F}_{p^2})^2 \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^6$

$\{P_B, Q_B\}$ by a matrix $\mathbf{B}'$ such that at least one element of $\mathbf{B}'$ is 1. In other words, we can represent $\{P_B, Q_B\}$ by an element in $(\mathbb{Z}/\ell_B^b\mathbb{Z})^3$.

We can also represent $\{P_C, Q_C\}$ by an element in $(\mathbb{Z}/\ell_C^c\mathbb{Z})^3$. However, this case is slightly trickier than the case of $\{P_B, Q_B\}$. It is because points P_C and Q_C are used to compute a (ℓ_C, ℓ_C)-isogeny chain in IS-CUBE, and the kernel of this isogeny chain is affected by constant factors of these points. In other words, subgroups $\langle (P_C, P'_C), (Q_C, Q'_C) \rangle$ and $\langle (c_0 P_C, P'_C), (c_0 Q_C, Q'_C) \rangle$ provide different (ℓ_C, ℓ_C)-isogeny chains for $c_0 \in (\mathbb{Z}/\ell_C^c\mathbb{Z})^\times \setminus \{\pm 1\}$. Since the points are multiplied by some constants in our compression method, we need to adjust these constant factors before computing the isogeny chain. Let the target constant be c_0, and let $e_{\ell_C^c}$ be the ℓ_C^c-Weil pairing. In the IS-CUBE setting, the value $e_{\ell_C^c}(P_C, Q_C)$ is defined by P'_C and Q'_C which are the rest points to construct the kernel of the isogeny chain. Precisely, we have

$$e_{\ell_C^c}(P_C, Q_C)^{\ell_A^a} = e_{\ell_C^c}(P'_C, Q'_C)^{\ell_C^c - \ell_A^a} \quad \text{or} \quad e_{\ell_C^c}(P_C, Q_C)^{\ell_C^c - \ell_A^a} = e_{\ell_C^c}(P'_C, Q'_C)^{\ell_A^a}$$

in the IS-CUBE setting. Therefore, we can compute $c_0^2 \pmod{\ell_C^c}$ from the above equation and the property of the Weil pairing. If ℓ_C is an odd integer, we can adjust the constants because we obtain c_0 or $-c_0$ from $c_0^2 \pmod{\ell_C^c}$. If $\ell_C = 2$, we obtain c_0, $-c_0$, $c_0 + 2^{c-1}$, or $-c_0 + 2^{c-1}$ from $c_0^2 \pmod{2^c}$. Unfortunately, we cannot reject the part of 2^{c-1} by the Weil pairing. However, this part only affects the last $(2,2)$-isogeny of the isogeny chain. Therefore, we can compute the correct isogeny chain by computing all possible isogenies in the final step of the computation of the isogeny chain. To summarize, we can compute the correct isogeny chain with small additional calculations even if we represent the basis $\{P_C, Q_C\}$ by an element in $(\mathbb{Z}/\ell_C^c\mathbb{Z})^3$.

Recall that Alice's public key is $(E_1, (P_1, Q_1), (P'_B, Q'_B))$, where E_1 is an elliptic curve, $\{P_1, Q_1\}$ is a basis of $E_1[\ell_C^c]$, and $\{P'_b, Q'_B\}$ is a basis of $E_1[\ell_B^b]$. Therefore, Alice's public key is embedded in $\mathbb{F}_{p^2} \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^3 \times (\mathbb{Z}/\ell_B^b\mathbb{Z})^3$ by using the compression technique. Recall that Bob's ciphertext is $(E'_s, (P'_0, Q'_0), E'_1, (P'_1, Q'_1))$, where E'_s and E'_1 are elliptic curves, $\{P'_0, Q'_0\}$ is a basis of $E'_s[\ell_C^c]$, and $\{P'_1, Q'_1\}$ is a basis of $E'_1[\ell_C^c]$. Therefore, Bob's ciphertext is embedded in $(\mathbb{F}_{p^2})^2 \times (\mathbb{Z}/\ell_C^c\mathbb{Z})^6$ by using the compression technique. We summarize in Table 1 the sets in which the public key and ciphertext of the original and compressed IS-CUBE are embedded.

5.2 Using B-SIDH prime

In this subsection, we discuss IS-CUBE using the prime of the form for B-SIDH.

B-SIDH is one variant of SIDH proposed by Costello [11]. The property of B-SIDH is the use of quadratic twists of elliptic curves. Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ such that $\#E(\mathbb{F}_{p^2}) = (p+1)^2$. Denote the quadratic twist of E by ${}^TE$. I.e., we have $E(\mathbb{F}_{p^4}) \cong {}^TE(\mathbb{F}_{p^4})$ and $E(\mathbb{F}_{p^2}) \ncong {}^TE(\mathbb{F}_{p^2})$. It holds that $\#{}^TE(\mathbb{F}_{p^2}) = (p-1)^2$; therefore, if both $p+1$ and $p-1$ are smooth integers, we can use points of order dividing $p-1$ to compute isogenies in addition to points of order diving $p+1$.

In the IS-CUBE setting, we can consider two different cases to use B-SIDH primes as follows:

- One of the sets $\{P_C, Q_C\}$ and $\{P_B, Q_B\}$ belongs to $E(\mathbb{F}_{p^2})$ and the other belongs to ${}^TE(\mathbb{F}_{p^2})$.
- There are eight points

$$P_{C,0}, Q_{C,0}, P_{B,0}, Q_{B,0} \in E(\mathbb{F}_{p^2}) \quad \text{and} \quad P_{C,1}, Q_{C,1}, P_{B,1}, Q_{B,1} \in {}^TE(\mathbb{F}_{p^2})$$

  such that $P_C = P_{C,0} + \varphi(P_{C,1})$, $Q_C = Q_{C,0} + \varphi(Q_{C,1})$, $P_B = P_{B,0} + \varphi(P_{B,1})$, and $Q_B = Q_{B,0} + \varphi(Q_{B,1})$, where φ is an isomorphism $\varphi\colon {}^TE \to E$.

Here, we suppose the orders of the above points are smooth (not necessarily powers of an integer).

In the first case, we use a prime p of the size of about 6λ bits or more since the order of P_C is about $2^{6\lambda}$. If $P_C, Q_C \in E(\mathbb{F}_{p^2})$ and $P_B, Q_B \in {}^TE(\mathbb{F}_{p^2})$, we compute the ciphertext by using the quadratic twists and compute the two-dimensional isogeny chain by using the original curves.

In the second case, we use a prime p of the size of about 4λ bits or more. However, it seems to be hard to find this type of B-SIDH prime if we construct τ from the factorization of $x^{2^*} - y^{2^*}$. It is because the order of P_C, Q_C is the form of x^{2^*}, and we need to find a rare p that satisfies $x_0^{2^*} | (p+1)$ and $x_1^{2^*} | (p-1)$ for some x_0 and x_1. In the case that we construct τ by using the structure of the endomorphism ring of E_0, we may find a B-SIDH type prime suitable for IS-CUBE. To compute the isogeny chain for the ciphertext, we first compute the isogeny of the kernel generated by $P_{B,0}$ and $Q_{B,0}$, and next compute the isogeny of the kernel generated by $P_{B,1}$ and $Q_{B,1}$ in the world of quadratic twists. We can also compute the two-dimensional isogeny chain for the decapsulation in the same way.

Consequently, we can use the B-SIDH type primes for IS-CUBE. They may provide more compact IS-CUBE implementations. However, the computational costs may become larger because we need to compute high-degree isogenies as B-SIDH.

6 Implementation

We implemented IS-CUBE by using `sagemath` [17] as a proof of concept. The code can be downloaded in `http://tomoriya.work/code.html`.

Table 2. Parameters for IS-CUBE using an endomorphism of E_0 to construct τ

λ	p (in bits)	Public key	Ciphertext	Compressed (P)	Compressed (C)
128	$1,044$	$1,305$ bytes	$1,566$ bytes	649 bytes	$1,104$ bytes
192	$1,558$	$1,948$ bytes	$2,337$ bytes	969 bytes	$1,649$ bytes
256	$2,068$	$2,585$ bytes	$3,102$ bytes	$1,289$ bytes	$2,192$ bytes

Table 3. Comparison of the IS-CUBE primes with primes for other isogeny-based schemes

λ	CSIDH [7,10]		M-SIDH [18]		terSIDH$^{\text{hyb}}$ [3]		FESTA [4]		IS-CUBE	
	$\mathsf{bit}(p)$	$\mathsf{bit}(p)/\lambda$	$\mathsf{bit}(p)$	$\mathsf{bit}(p)/\lambda$	$\mathsf{bit}(p)$	$\mathsf{bit}(p)/\lambda$	$\mathsf{bit}(p)$	$\mathsf{bit}(p)/\lambda$	$\mathsf{bit}(p)$	$\mathsf{bit}(p)/\lambda$
128	$3,072$	24.00	$5,911$	46.18	$1,532$	11.97	$1,292$	10.09	$1,044$	**8.16**
192	$8,192$	42.67	$9,382$	48.86	$2,373$	12.36	$1,966$	10.24	$1,558$	**8.11**
256	-	-	$13,000$	50.78	$3,216$	12.56	$2,772$	10.83	$2,068$	**8.08**

In this section, we provide parameters for IS-CUBE under different security levels and a brief explanation of some techniques used in the IS-CUBE implementation. Moreover, we show the computational time of IS-CUBE under this implementation.

6.1 Parameter selections for IS-CUBE

We first provide the primes for IS-CUBE in three security models: the AES-128, AES-192, and AES-256 security. The primes can be defined as follows. Here, we suppose that τ is constructed by using the structure of the endomorphism ring of E_0 to take primes of the sizes of about 8λ bits. Appendix A provides primes when we construct τ by using the factorization of $x^{2^*} - y^{2^*}$. We summarize the size of these primes and the public information appearing in IS-CUBE in Table 2. The primes for IS-CUBE are as follows:

$$
\begin{aligned}
p_{128} &= 2^{774}3 \cdot 5^{111} \cdot 11 \cdot 122 - 1, \quad \deg \phi_1 = 3^{488}, \\
p_{192} &= 2^{1158}3 \cdot 5^{166} \cdot 5 \cdot 1129 - 1, \quad \deg \phi_1 = 3^{730}, \\
p_{256} &= 2^{1542}3 \cdot 5^{221} \cdot 7 \cdot 263 - 1, \quad \deg \phi_1 = 3^{972},
\end{aligned}
$$

where p_λ is the prime for IS-CUBE in the AES-λ security.

Moreover, we compare the primes for IS-CUBE with the primes for other isogeny-based schemes, CSIDH, M-SIDH, terSIDH$^{\text{hyb}}$, and FESTA. We summarize this comparison in Table 3. Here, the sizes of the primes of CSIDH are provided by [10]. As shown in Table 3, the primes of IS-CUBE are of 8λ-bits, and the growth of them via λ is smaller than those for other schemes.

6.2 Some techniques used in our implementation

In this subsection, we explain some techniques used in our implementation of IS-CUBE.

Arithmetics of Montgomery curves. A Montgomery curve is an elliptic curve that has the form of

$$y^2 = x^3 + \alpha x + x, \quad \alpha^2 \neq 4.$$

The arithmetics of Montgomery curves have been developed along with the development of the Elliptic Curve Cryptography and SIDH. We implemented IS-CUBE by using Montgomery arithmetics. This provides an IS-CUBE implementation without the information of y-coordinates of points in most cases. Refer [14, Table 1] to group arithmetics of Montgomery curves, and refer [36,12,26,30] to isogeny arithmetics of Montgomery curves.

To compute $(2,2)$-isogenies, we need the full information of points belonging to the kernels. Therefore, we need to recover the y-coordinates of the points in generating the shared key of Alice. Though we have the ambiguity of the sign of points, we can take points of the same sign by using the Weil pairing as mentioned in the original paper of FESTA.

Radical isogenies of degree 3 on Montgomery curves. To generate Alice's public key, we need to compute 3-isogenies without backtracking. We use the technique of the Radical isogeny [6] that provides the computation of isogenies without backtracking. In particular, we have the following proposition:

Proposition 2 ([33, Theorem 7]). *Let E_1, E_2 be Montgomery curves, let $\phi\colon E_1 \to E_2$ be a 3-isogeny, and let t be the x-coordinate of a point generating $\ker\phi$. Then, a point whose x-coordinate is $-\frac{1}{3t}$ generates the kernel of $\hat{\phi}$. Moreover, the x-coordinates of the other points of order 3 in E_2 can be represented by*

$$3t\alpha^2 + (3t^2 - 1)\alpha + 3t^3 - 2t,$$

where α is a cube root of $t(t^2-1)$.

By using this proposition, we can take a non-backtracking point of order 3 with a computation of one cube root and a few multiplications and additions. To randomize the direction of the 3-isogeny, we multiply a random cube root of 1 to α.

To compute a cube root of $t(t^2-1)$, we can use the following proposition:

Proposition 3. *Suppose that $(p^2-1)/3$ is an integer coprime to 3. If the equation $x^3 - T = 0$ over $\mathbb{F}_{p^2}$ has at least one root belonging to $\mathbb{F}_{p^2}$, then we can obtain one root x_0 of $x^3 - T = 0$ by*

$$x_0 = T^{\frac{\left(\frac{p^2-1}{3}+\delta\right)}{3}\cdot\delta},$$

where $\delta = 1$ if $(p^2-1)/3 \equiv 2 \pmod 3$ and $\delta = -1$ if $(p^2-1)/3 \equiv 1 \pmod 3$.

Proof. If $T = 0$, the proposition holds.

Suppose that $T \neq 0$. Since $x^3 - T = 0$ has at least one root in $\mathbb{F}_{p^2}$, the element T is in the cyclic group $((\mathbb{F}_{p^2})^\times)^3$. Since the order of $((\mathbb{F}_{p^2})^\times)^3$ is $\frac{p^2-1}{3}$, we have

$$\left(T^{\frac{\left(\frac{p^2-1}{3}+\delta\right)}{3}}\right)^3 = T^{\frac{p^2-1}{3}} \cdot T^\delta = T^\delta.$$

This completes the proof of Proposition 3. □

We may not know whether $x^3 - t(t^2-1) = 0$ has a root in $\mathbb{F}_{p^2}$ or not. However, it is not hard to prove in the setting of IS-CUBE. Since $E_2[3] \subset E_2(\mathbb{F}_{p^2})$ and $E_1[3] \subset E_1(\mathbb{F}_{p^2})$, we have

$$3tx_0^2 + (3t^2-1)x_0 + 3t^3 - 2t \in \mathbb{F}_{p^2}$$

and $t \in \mathbb{F}_{p^2}$, where x_0 is a root of $x^3 - t(t^2-1) = 0$. From [33, §3.1], it holds that $t \neq 0$. Therefore, the element $x_0 \cdot x_0^{p^2}$ belongs to $\mathbb{F}_{p^2}$ from the Vieta's formula. Because $x_0^{p^2}$ is also a root of $x^3 - t(t^2-1) = 0$, there is at least one root belonging to $\mathbb{F}_{p^2}$.

Computation of isogeny chains. In the encapsulation and decapsulation processes, we need to compute ℓ_B and $(2,2)$-isogeny chains. To compute these chains efficiently, the optimal strategy method provided by [15] is useful. For the computation of the ℓ_B-isogeny chains, we use the algorithm in [1]. For the computation of the $(2,2)$-isogeny chains, we use the code provided by the FESTA original paper. Refer [4, 7.2] to the details of the optimization used in their code.

6.3 Computation time

We show the computational time of IS-CUBE based on our PoC implementation. The results are summarized in Table 4. We used primes p_{128}, p_{192}, and p_{256}. We measured the averages of 100 run times of each algorithm of IS-CUBE except for the computational time of the public parameters generation. The time of the key decapsulation was measured in two cases: without a checking process (Remark 2) and with it. The time of the public parameters generation is just for reference. It was measured only once and we did not use the method in Section 3.5. We used a MacBook Air with an Apple M1 CPU (3.2 GHz) to measure the computational time.

7 Conclusion

In this paper, we proposed a novel isogeny-based key encapsulation mechanism named IS-CUBE. The main properties of IS-CUBE are that it uses a prime of the size of about 8λ bits for the security parameter λ and its starting curve is a random supersingular elliptic curve.

Table 4. Computational time of IS-CUBE

Computation \ Security parameter	128	192	256
Public parameters generation*	38.36 sec	112.18 sec	165.75 sec
Public key generation	4.34 sec	13.99 sec	34.43 sec
Key encapsulation	0.61 sec	1.22 sec	2.10 sec
Key decapsulation	17.13 sec	39.06 sec	74.61 sec
Key decapsulation (with a checking process)	31.93 sec	71.59 sec	135.45 sec

IS-CUBE is based on a 3-dimensional SIDH diagram and Kani's theorem. We introduced the LIT problem (Problem 3) to construct the 3-dimensional SIDH diagram appearing in the structure of IS-CUBE. The LIT problem is the problem to compute a hidden isogeny of a large degree from given two supersingular elliptic curves and information of torsion points images of relatively small order.

Our PoC implementation of IS-CUBE via `sagemath` showed that it took about 4.34 sec for the public key generation, 0.61 sec for the encapsulation, and 17.13 sec for the decapsulation in the AES-128 security.

Acknowledgements.

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

References

1. Reza Azarderakhsh, Matthew Campagna, Craig Costello, LD Feo, Basil Hess, A Jalali, D Jao, B Koziel, B LaMacchia, P Longa, et al. Supersingular isogeny key encapsulation. *Submission to the NIST Post-Quantum Standardization project*, 2017.
2. 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.
3. Andrea Basso and Tako Boris Fouotsa. New SIDH countermeasures for a more efficient key exchange, 2023. `https://eprint.iacr.org/2023/791`.
4. Andrea Basso, Luciano Maino, and Giacomo Pope. FESTA: Fast encryption from supersingular torsion attacks, 2023. `https://ia.cr/2023/660` appear at ASIACRYPT 2023.
5. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In *Advances in Cryptology – EUROCRYPT 2023*, pages 423–447. Springer, 2023.
6. Wouter Castryck, Thomas Decru, and Frederik Vercauteren. Radical isogenies. In *Advances in Cryptology – ASIACRYPT 2020*, pages 493–519. Springer, 2020.
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, 2023. `https://eprint.iacr.org/2023/1433` appear at ASIACRYPT 2023.
9. Denis Charles, Kristin Lauter, and Eyal Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, 2009.
10. Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, Samuel Jaques, and Francisco Rodríguez-Henríquez. The SQALE of CSIDH: sublinear vélu quantum-resistant isogeny action with low exponents. *Journal of Cryptographic Engineering*, 12(3):349–368, 2022.
11. Craig Costello. B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. In *Advances in Cryptology – ASIACRYPT 2020*, pages 440–463. Springer, 2020.
12. Craig Costello and Huseyin Hisil. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In *Advances in Cryptology – ASIACRYPT 2017*, pages 303–329. Springer, 2017.
13. 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.
14. Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In *Advances in Cryptology – CRYPTO 2016*, pages 572–601. Springer, 2016.
15. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.
16. 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.
17. The Sage Developers. Sagemath, version 10.0, 2023. `http://www.sagemath.org`.
18. 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.
19. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – CRYPTO'99*, pages 537–554. Springer, 1999.
20. Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
21. 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.
22. Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *Advances in Cryptology – CRYPTO 2018*, pages 96–125. Springer, 2018.
23. Ernst Kani. The number of curves of genus two with elliptic differentials. 1997.
24. David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014.

25. 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.
26. Michael Meyer and Steffen Reith. A faster way to the CSIDH. In *Progress in Cryptology – INDOCRYPT 2018*, pages 137–152. Springer, 2018.
27. James Milne. Abelian varieties. *Arithmetic geometry*, pages 103–150, 1986.
28. James Milne. Jacobian varieties. pages 167–212, 1986.
29. Tomoki Moriya. The wrong use of FESTA trapdoor functions leads to an adaptive attack. `https://eprint.iacr.org/2023/1092`.
30. Tomoki Moriya, Hiroshi Onuki, Yusuke Aikawa, and Tsuyoshi Takagi. The generalized Montgomery coordinate: A new computational tool for isogeny-based cryptography. *Mathematical Cryptology*, 2(1):36–59, 2022.
31. Kohei Nakagawa and Hiroshi Onuki. QFESTA: Efficient algorithms and parameters for festa using quaternion algebras, 2023. `https://eprint.iacr.org/2023/1468`.
32. National Institute of Standards and Technology. Post–quantum cryptography standardization. `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions`.
33. Hiroshi Onuki and Tomoki Moriya. Radical isogenies on Montgomery curves. In *Public-Key Cryptography – PKC 2022*, pages 473–497, 2022.
34. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *Advances in Cryptology – ASIACRYPT 2017*, pages 330–353. Springer, 2017.
35. 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.
36. Joost Renes. Computing isogenies between Montgomery curves using the action of (0, 0). In *Post-Quantum Cryptography – PQCrypto 2018*, pages 229–247. Springer, 2018.
37. Damien Robert. Breaking SIDH in polynomial time. In *Advances in Cryptology – EUROCRYPT 2023*, pages 472–503. Springer, 2023.
38. Joseph Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.
39. Benjamin Smith. *Explicit endomorphisms and correspondences*. PhD thesis, University of Sydney, 2005.
40. Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, 2009.
41. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris Sér. A*, 273(5):238–241, 1971.

Appendix A Primes when we construct τ based on the factorization of $x^{2^*} - y^{2^*}$

We provide the primes when we construct τ via the factorization of $x^{2^*} - y^{2^*}$. Table 5 summarizes the size of these primes and public information of IS-CUBE. Here, the prime p'_λ is the prime for IS-CUBE in the AES-λ security.

$$\begin{aligned}
p'_{128} &= 2^{2^{10}} 3 \cdot 7^{92} \cdot 7 \cdot 123 - 1, \quad \deg \phi_1 = 3^{2^9}, \\
p'_{192} &= 2^{2^{11}} 3 \cdot 7^{137} \cdot 7 \cdot 335 - 1, \quad \deg \phi_1 = 3^{2^{10}}, \\
p'_{256} &= 2^{2^{11}} 3 \cdot 7^{183} \cdot 7 \cdot 190 - 1, \quad \deg \phi_1 = 3^{2^{10}}.
\end{aligned}$$

Table 5. Parameters for IS-CUBE using the factorization of $x^{2^*} - y^{2^*}$ to construct τ

λ	p (in bits)	Public key	Ciphertext	Compressed (P)	Compressed (C)
128	$1,294$	$1,618$ bytes	$1,941$ bytes	805 bytes	$1,415$ bytes
192	$2,446$	$3,058$ bytes	$3,669$ bytes	$1,524$ bytes	$2,759$ bytes
256	$2,574$	$3,218$ bytes	$3,861$ bytes	$1,605$ bytes	$2,823$ bytes