Optimized CSIDH Implementation Using a 2-torsion Point

Donghoe Heo[1], Suhri Kim[1], Kisoon Yoon[2], Young-Ho Park[3], and Seokhie Hong[1]

[1] Center for Information Security Technologies (CIST), Korea University, Seoul, Republic of Korea
dong5641@korea.ac.kr, suhrikim@gmail.com, shhong@korea.ac.kr

[2] NSHC Inc., Uiwang, Republic of Korea
kisoon.yoon@gmail.com

[3] Sejong Cyber University, Seoul, Republic of Korea
youngho@sjcu.ac.kr

Abstract. The implementation of isogeny-based cryptography mainly use Montgomery curves as they offer fast elliptic curve arithmetic and isogeny compuation. However, although Montgomery curves have efficient 3- and 4-isogenies, it becomes inefficient when recovering the coefficient of the image curve for large degree isogenies. This is the main bottleneck of using a Montgomery curve for CSIDH as it requires odd-degree isogenies up to at least 587 [4]. In this paper, we present a new optimization method for faster CSIDH protocols entirely on Montgomery curves. To this end, we present a new parameter for CSIDH in which the rational 2-torsion points are defined over $\mathbb{F}_p$. By using the proposed parameters the CSIDH moves around the surface. The curve coefficient of the image curve can be recovered by a 2-torsion point. We also proved that the CSIDH using the proposed parameter guarantees a free and transitive group action. Additionally, we present the implementation result using our method. We demonstrated that our method is 8.6% faster than the original CSIDH. Our works show that quite higher performance of CSIDH is achieved using only Montgomery curves.

Keywords: Post-quantum cryptography, Isogeny, Montgomery curves, 2-torsion points, CSIDH.

1 Introduction

With the evolution of a quantum computing environment, currently used public key cryptosystems based on factorization and discrete logarithm problems, such as RSA and ECC, will not be able to guarantee their security in the near future. This has led to the need for post-quantum cryptography (PQC) that is secure even in quantum computing environments. The National Institute of Standards and Technology (NIST) opened the PQC standardization project, which is now in Round 2. Among the PQC categories, isogeny-based cryptography interests

many researchers as it offers smaller key sizes than any other PQC candidates. The isogeny-based cryptography is based on the difficulty of finding a specific isogeny between two elliptic curves defined on the same finite field and having the same order. Despite having a fairly small key size, isogeny-based cryptography has the disadvantage of being considerably slower than most of the PQC candidates.

The isogeny-based cryptography was first proposed by Couveignes in 2006 [8]. This is a non-interactive key exchange protocol which uses a set of $\mathbb{F}_q$-isomorphism classes of ordinary elliptic curves defined on $\mathbb{F}_q$. The endomorphism ring between these curves is given by the order $\mathcal{O}$ in an imaginary quadratic field. Then, the ideal class group $\mathrm{cl}(\mathcal{O})$ acts freely and transitively on this endomorphism ring through an isogeny operation. Couveignes designed the Diffie-Hellman style key exchange protocol using the commutativity of $\mathrm{cl}(\mathcal{O})$. This method was rediscovered by Rostovtsev and Stolbunov and called CRS-scheme. On the other hand, the underlying problem of CRS-scheme can be classified as an abelian hidden-shift problem. It is shown by Childs et al. that there is a subexponential quantum attack algorithm with time complexity of $L_q[1/2]$ [6]. Considering that RSA is widely used even in subexponential complexity in classical computers, this was not considered as a big problem. However, prolonged execution time was pointed out as the biggest problem, as it took several minutes for a single key exchange.

The isogeny-based cryptography was noticed again with a rapid speed improvement by De Feo et al. [11]. They proposed a new key exchange protocol called SIDH using a supersingular curve. As Childs-Jao-Soukharev's attack exploits the commutativity of $\mathrm{cl}(\mathcal{O})$ of an ordinary curve, their attack cannot be applied to SIDH since it uses supersingular curves, which has non-commutative full endomorphism ring. Until now, SIDH is known to have exponential time complexity, even in quantum computing environments.

SIKE (Supersingular Isogeny Key Encapsulation), which is based on SIDH, is currently on the NIST PQC standardization Round 2 [1]. On the other hand, in the case of SIDH-based scheme, the key validation problem could not be solved efficiently. To solve this problem, SIKE applied a transformation similar to the Fujisaki-Okamoto transformation proposed in [10].

In CRS-scheme, efficient key validation is possible, so that CCA-secure encryption can be achieved only by the basic algorithm itself, without the need of applying FO-transformation. This allows a non-interactive key exchange, where several of the previously proposed PQC algorithms do not provide this property efficiently. With this in mind, De Feo et al. proposed a method to efficiently perform CRS-schemes on ordinary curves in [9]. However, there was still a problem that it was difficult to select parameters satisfying a certain condition because of the characteristics of ordinary curves. Independently, Castryck et al. proposed CSIDH (Commutative Supersingular Isogeny Diffie-Hellman), an algorithm that increases efficiency over conventional techniques by using the supersingular curve defined over a prime field $\mathbb{F}_p$ in CRS-scheme [4]. By using supersingular curves,

CSIDH solved the parameter selection problem of ordinary curves in the algorithm proposed by De Feo et al.

CSIDH uses a subring consisting of $\mathbb{F}_p$-rational endomorphisms instead of using a full endomorphism ring, and uses the commutativity of $\mathrm{cl}(\mathcal{O})$ and has the same protocol as CRS-scheme. The CSIDH-512 provides a key size of 64 bytes, which is smaller than SIKE for the same security level. Even considering the subexponential time attack, the key size is expected to be relatively smaller than SIKE. Recently, various papers related to CSIDH have been submitted to PQCrypto 2019 and Eurocrypt 2019, and various researches such as digital signature, efficient implementation techniques, various attack techniques, and side-channel resistant implementations have been conducted [2, 5, 13, 14].

However, one disadvantage of CSIDH is that it has a slower execution speed than the state-of-the-art implementation of SIKE. On the other hand, since the key validation can be performed efficiently, a non-interactive key exchange can be provided, and a smaller key size and a simpler algorithm can be designed. In addition, considering more efficient digital signature scheme than SIDH can be derived, it is possible to say that CSIDH has more potential for developing various cryptographic applications. Hence, various studies are being actively conducted to improve the speed of CSIDH [13, 14].

The original implementation of CSIDH in [4] uses Montgomery curves, as they were known to provide efficient isogeny computation. However, one drawback of using Montgomery curves is that the computational cost for recovering the coefficient of the image curve is higher than Edwards curves for large degree isogenies. Since CSIDH protocol uses large odd-degree isogenies, this can be an obstacle for CSIDH to implement entirely on Montgomery curves.

In this paper, we apply an optimization technique proposed by Costello and Hisil in CSIDH to obtain image curve coefficients during isogeny computations [7]. The followings are the main contributions of this work.

- We present a new initial curve and a new prime of the form $8k + 7$, which enable to use the 2-torsion method by Costello and Hisil [7]. In the parameter presented in the original CSIDH, $\mathbb{F}_p$-rational 2-torsion points do not exist except for $(0, 0)$, so that this method cannot be used for recovering the coefficient of the image curve in CSIDH. Compared to the Meyer's method [14], computing the coefficient of the image curve is the main bottleneck for implementing faster CSIDH entirely on Montgomery curves. By using our prime, $\mathbb{F}_p$-rational 2-torsion points exist so that the coefficient can be computed efficiently.
- We also prove that our algorithm assures one-to-one correspondence between image curves and elliptic curve isomorphism classes. Given a Montgomery curve $M_A : y^2 = x^3 + Ax^2 + x$ on the surface with curve coefficient A and base field prime p, we prove that the ideal-class group $\mathrm{cl}(\mathcal{O})$ acts freely and transitively on the set $S^{+}_{p,\mathbb{Z}[(1+\sqrt{-p})/2],i}$ in [3]. Details of our proof are denoted in Section 4.
- We present the implementation results of our proposed method. The group action of our implementation is about 13.9% faster than the original CSIDH.

The entire key exchange is about 8.6% faster than the original CSIDH. Although the proposed CSIDH implementation is slower than [14], we stress the fact that we provide the fastest performance using only Montgomery curves. Details of our implementation and results are denoted in Sections 5.

This paper is organized as follows. In Section 2, we review on background of elliptic curves and CSIDH key exchange. In Section 3, we introduce a various way of odd-degree isogeny computations. In Section 4, we present a new parameter that makes the use of the 2-torsion point. Section 5 describes the specific implementation process and the result of comparing the costs and speed. We draw our conclusions and future work in Section 6.

2 Preliminary

In this section, we describe the background knowledge needed to develop this paper. First, we review some properties of elliptic curves. Then, we introduce the CSIDH protocol and odd-degree isogeny formula on Montgomery curves.

2.1 Elliptic curves and isogenies

Montgomery curves Let K be a field with the characteristic not equal to 2 or 3. The Montgomery elliptic curves over K are expressed by the following equation:

$$M_{a,b} : by^2 = x^3 + ax^2 + x, \tag{1}$$

where $b(a^2 - 4) \neq 0$. We shall write M_a when $b = 1$ throughout the paper. For efficient implementation of isogeny operation, we use the projective coordinate and projective curve coefficient to avoid inversions. Since Montgomery curve arithmetic can be constructed only with the x-coordinate, XZ-coordinate system is mainly used for implementing isogeny-based cryptography. Now, we write a point $P = (x, y)$ on $M_{a,b}$ and coefficient a as $P = (X : Z)$ and $a = (A : C)$, respectively, where $x = X/Z$ and $a = A/C$.

Isogeny Let O_E be a point at infinity of an elliptic curve E. Given two elliptic curves E and E', we define an isogeny ϕ between E and E' by $\phi : E \to E'$ satisfying $\phi(O_E) = O_{E'}$, where ϕ is a morphism. Since ϕ is group homomorphism between E and E', $\ker(\phi)$ is a subgroup of E. Given any finite subgroup K of E, we use Velu's formula to compute an isogeny $\phi : E \to E'$. Then we obtain an isogeny $\phi : E \to E'$ satisfying $\ker(\phi) = K$ and denote $\deg(\phi) = |K|$.

Supersingularity Given a prime p, let E be an elliptic curve defined over $\mathbb{F}_p$. Then E is a supersingular curve if and only if

$$\#E(\mathbb{F}_p) = p + 1$$

Otherwise, E is an ordinary curve. Let $\mathrm{End}(E)$ be a full endomorphism ring of E and $\mathrm{End}_{\mathbb{F}_p}(E)$ be an $\mathbb{F}_p$-rational endomorphism ring defined over $\mathbb{F}_p$. A full endomorphism ring of an oridinary curve is an order in an imaginary quadratic field. On the other hand, A full endomorphism ring $\mathrm{End}(E)$ of supersingular curve E is an order in a quaternion algebra. Also, $\mathbb{F}_p$-rational endomorphism ring $\mathrm{End}_{\mathbb{F}_p}(E)$ of supersingular curve E is an order in an imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$. Now, denote an order $\mathcal{O}$ for $\mathrm{End}_{\mathbb{F}_p}(E)$.

Ideal Class Group Given an order $\mathcal{O}$, the ideal class group of $\mathcal{O}$ is defined by a quotient group

$$\mathrm{cl}(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O})$$

Note that $I(\mathcal{O})$ is the set of invertible fractional ideals and $P(\mathcal{O})$ is the set of principal fractional ideals.
Let $\pi \in \mathcal{O}$ be the $\mathbb{F}_p$-Frobenius endomorphism of E and $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ be the set of elliptic curves E defined over $\mathbb{F}_p$ satisfying $\mathcal{O} = \mathrm{End}_{\mathbb{F}_p}(E)$. Then, the ideal-class group $\mathrm{cl}(\mathcal{O})$ acts freely and transitively on $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ by

$$\begin{aligned} \mathrm{cl}(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}, \pi) &\longrightarrow \mathcal{E}\ell\ell_p(\mathcal{O}, \pi) \\ ([\mathfrak{a}], E) &\longrightarrow E/\mathfrak{a} \end{aligned}$$

2.2 CSIDH

CSIDH Protocol CSIDH is an isogeny-based Diffie-Hellman protocol proposed by Castryck et al. [4] using supersingular curves defined over $\mathbb{F}_p$ and commutative group action. The prime p of the base field is of the form $p = 4\prod_{i=1}^{n} \ell_i - 1$, where ℓ_i's are odd primes. For an order $\mathcal{O} = \mathrm{End}_{\mathbb{F}_p}(E)$, it is well-known that the class group $\mathrm{cl}(\mathcal{O})$ acts freely and transitively on $\mathcal{E}\ell\ell_p(\mathcal{O})$. This group action is represented by $[\mathfrak{a}]E$, where $E \in \mathcal{E}\ell\ell_p(\mathcal{O})$ and an ideal class $[\mathfrak{a}] \in \mathrm{cl}(\mathcal{O})$. Since E is a supersingular curve with $\#E(\mathbb{F}_p) = p + 1 = 4 \cdot \ell_1 \cdots \ell_n$, for each i, there is $\mathbb{F}_p$-rational subgroup of order ℓ_i. Also, let $\pi = \sqrt{-p}$ be the $\mathbb{F}_p$-Frobenius endomorphism of E. Then, since $p = -1 \mod \ell_i$, for a prime ℓ_i, it is well-known that $\ell_i\mathcal{O}$ splits into two prime ideals $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\mathfrak{l}_i^{-1} = (\ell_i, \pi + 1)$. Using Velu's formula, we compute $[\mathfrak{l}_i]E$ through the isogeny $\phi_{\mathfrak{l}_i}$ defined over $\mathbb{F}_p$ and compute $[\mathfrak{l}_i^{-1}]E$ through the isogeny $\phi_{\mathfrak{l}_i^{-1}}$ defined over $\mathbb{F}_{p^2}$.

Assume that Alice and Bob execute a key exchange. Alice and Bob randomly select each secet key $[\mathfrak{a}]$ and $[\mathfrak{b}]$ in $\mathrm{cl}(\mathcal{O})$, respectively. Next, Alice sends $E_A = [\mathfrak{a}]E$ to Bob, Bob sends $E_B = [\mathfrak{b}]E$ to Alice. Upon the receipt of E_B from Bob, Alice computes $[\mathfrak{a}]E_B$ and obtains $E_{AB} = [\mathfrak{a}]E_B$. Similarly, Bob obtains $E_{BA} = [\mathfrak{b}]E_A$. The $E_{AB} = E_{BA}$ is the shared secret between Alice and Bob.

CSIDH group action An element of the ideal-class group $\mathrm{cl}(\mathcal{O})$ is of the form $\prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ $(\mathfrak{l}_i = (\ell_i, \pi - 1))$ for small $e_i \in [-m, m]$. So, in CSIDH protocol, Alice and Bob randomly select a vector $(e_1, e_2, \cdots, e_n) \in \mathbb{Z}^n$ and consider it as a secret

key. Thus a group action $[\mathfrak{a}]E$ can be computed by applying ℓ_i-isogeny operation e_i times for $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i} \in \mathrm{cl}(\mathcal{O})$.

If $e_i > 0$, ℓ_i-isogeny is applied with the kernel generated by a point in $E(\mathbb{F}_p)$ of order ℓ_i. If $e_i < 0$, ℓ_i-isogeny is applied with the kernel generated by a point in $E(\mathbb{F}_{p^2} \setminus \mathbb{F}_p)$ of order ℓ_i. As ℓ_is are all primes, this means that efficient odd-degree isogeny formula at least up to 587 for CSIDH-512 is required for implementation. For Montgomery curves, Costello and Hisil proposed an efficient method for computing odd-degree isogenies [7]. For twisted Edwards curves, Moody and Shumow proposed generalized odd-degree isogeny formula [15]. In [12], they optimized Moody and Shumow formula by using the w-coordinate on Edwards curves.

3 Odd-degree isogenies

Generally, an isogeny operation is divided into two parts – evaluation of an isogeny and coefficients computation of an image curve. In this section, we shall briefly introduce the formula in [7] for point evaluations. For coefficient computations, we introduce various methods that can be used to implement CSIDH. From this section, the **M**, **S**, and **a** refers to a field multiplication, squaring, and addition, respectively.

3.1 Point evaluation

In [7], Costello and Hisil proposed a simple formula for computing arbitrary degree isogenies on Montgomery curves. Their formula can be summarized as follows.

Theorem 1. *For a field K, whose characteristic is not 2, let P be a point of order $\ell = 2d + 1$ on the Montgomery curve $M_{a,b}/K : by^2 = x^3 + ax^2 + x$. Writing $\sigma = \sum_{i=1}^{d} x_{[i]P}$, $\tilde{\sigma} = \sum_{i=1}^{d} 1/x_{[i]P}$ and $\pi = \prod_{i=1}^{d} x_{[i]P}$, let ℓ-isogeny $\phi : M_{a,b} \to M_{a',b'}$ with $\ker(\phi) = \langle P \rangle$, where $M_{a',b'}/K : b'y^2 = x^3 + a'x^2 + x$. Then,*

$$a' = (6\tilde{\sigma} - 6\sigma + a) \cdot \pi^2 \quad \text{and} \quad b' = b \cdot \pi^2 \tag{2}$$

$$\phi : (x, y) \mapsto (f(x), yf'(x)), \tag{3}$$

where $f(x) = x \prod_{i=1}^{d} (\frac{x \cdot x_{[i]P} - 1}{x - x_{[i]P}})^2$ and $f'(x)$ is its derivative.

As mentioned earlier, since Montgomery curve arithmetic can be constructed only with the x-coordinate, the function $f(x)$ is of our main interest in the equation (3).

Let P be a point on a Montgomery curve having order $\ell = 2d + 1$. Then in projective XZ-coordinate we express P as $P = (X : Z)$, where $x = X/Z$. Let ϕ be an isogeny ℓ-isogeny, where $\ker \phi = \langle P \rangle$. From the formula proposed in [7],

$P' = \phi(P) = (X' : Z')$ is computed as

$$X' = X \cdot \left(\prod_{i=1}^{d} (X \cdot X_i - Z_i \cdot Z) \right)^2,$$
$$Z' = Z \cdot \left(\prod_{i=1}^{d} (X \cdot Z_i - X_i \cdot Z) \right)^2$$

where $x_i = X_i/Z_i$ is x-coordinate of $[i]P$ for $1 \leq i \leq d$. The computational cost of $(X \cdot X_i - Z_i \cdot Z)$ and $(X \cdot Z_i - X_i \cdot Z)$ is $2\mathbf{M} + 6\mathbf{a}$ by rewriting the equation as below.

$$X' = X \cdot \left(\prod_{i=1}^{d} [(X - Z)(X_i + Z_i) + (X + Z)(X_i - Z_i)] \right)^2, \tag{4}$$
$$Z' = Z \cdot \left(\prod_{i=1}^{d} [(X - Z)(X_i + Z_i) - (X + Z)(X_i - Z_i)] \right)^2 \tag{5}$$

For ℓ-isogeny evaluaton, the computational cost is $(4d)\mathbf{M} + 2\mathbf{S} + (6d)\mathbf{a}$.

As denoted in the equation (3), the computation of the image curve using Theorem 1 in [7] is somewhat complicated. Therefore, an alternate way to recover the coefficient of the image curve is presented in [7]. The first method is to use a 2-torsion point of a Montgomery curve, and another is to use two points and its differential of a Montgomery curve. We shall call the former method as a 2-torsion method and the later as a differential method. As the 2-torsion method is of our primary interest in this paper, we shall only describe the details of the 2-torsion method in this paper. Additionally, we provide two other ways to compute the coefficient of the image curve presented in [4, 14], in the following subsection.

3.2 Coefficients computations

The 2-torsion method A point P in an elliptic curve is called a k-torsion point if $[k]P = O$, where O is a point at infinity of an elliptic curve. In [7], the main idea is to use 2-torsion points for coefficient computation, as pushing a 2-torsion point through an odd-degree isogeny preserves their order on the image curve.

For a Montgomery curve, it is well-known that the 2-torsion point has the following form

$$(0, 0),\ (\alpha, 0),\ (\alpha^{-1}, 0) \text{ where } \alpha \in \bar{\mathbb{F}}_p$$

If we know α of the 2-torsion point on a Montgomery curve, then we can recover the coefficient of a Montgomery curve. For a given elliptic curve M_a, since $\alpha^3 + a\alpha^2 + \alpha = 0$, we can calculate the coefficient a of M_a by

$$a = -(\alpha^2 + 1)/\alpha \tag{6}$$

Let $\phi : M_a \to M_{a'}$ be an isogeny of odd-degree $\ell = 2d+1$, and $P = (\alpha, 0)$ be a 2-torsion point on M_a. Then it is clear that $\phi(P)$ is 2-torsion point on $M_{a'}$. Using this, we can recover the coefficient of the image curve by first, evaluating $\phi(P)$ and obtain the coefficient by the equation (6). More precisely, assume that $\phi(P) = (\alpha', 0)$. Then we obtain $a' = -((\alpha')^2 + 1)/\alpha'$. In projective coordinate, let $P = (X_\alpha, Z_\alpha)$, where $\alpha = X_\alpha/Z_\alpha$. Then projective curve coefficient of the image curve droven by the equation (6)

$$a' = (A' : C') = (X_{\alpha'}^2 + Z_{\alpha'}^2 : -X_{\alpha'}Z_{\alpha'}),$$

where $\phi(P) = (X_{\alpha'} : Z_{\alpha'})$ and $a' = A'/C'$. This computation cost is $2\mathbf{S} + 5\mathbf{a}$. Using the 2-torsion method, the cost of calculating a coefficient of $\ell = 2d+1$-isogeny image curve is $(4d)\mathbf{M} + 4\mathbf{S} + (6d+5)\mathbf{a}$.

Remark 1. Recently, in [3], Castryck and Decru proposed CSURF algorithm using tweaked Montgomery curve $M_a^t : y^2 = x^3 + ax^2 - x$ and it is about 5.68% faster than the original CSIDH. CSURF can also use the 2-torsion method because three 2-torsion points are on $M_a^t(\mathbb{F}_p)$. If $(\alpha, 0)$ is a 2-torsion point on a tweaked Montgomery curve M_a^t for $\alpha \neq 0$, then since $\alpha^2 + a\alpha - 1 = 0$, we can reconstruct tweaked Montgomery coefficient a by $a = (A : C) = -(\alpha^2 - 1)/\alpha = (Z - X)(Z + X)/XZ$, where $\alpha = X/Z$. So, we can compute an image curve coefficient by one additional point evaluation and $2\mathbf{M} + 2\mathbf{a}$. Using the 2-torsion method, CSURF will be more efficient in computing odd-degree isogeny parts.

Optimization by Castryck et al. [4] In [4] they optimize the equation (2) to compute the coefficient of the image curve, as $\mathbb{F}_p$-rational 2-torsion point does not exist for the original parameters of CSIDH.

For a point P of order ℓ on E and $k \in \{1, \cdots, \ell - 1\}$, let $(X_k : Z_k)$ be the projective x-coordinate of $[k]P$. Define $c_i \in \mathbb{F}_p$ such that

$$\prod_{i=1}^{\ell-1}(Z_i w + X_i) = \sum_{i=0}^{\ell-1} c_i w^i$$

as polynomials in w, and define τ, σ by

$$\tau = \prod_{i=1}^{\ell-1} \frac{X_i}{Z_i},\ \sigma = \sum_{i=1}^{\ell-1}\left(\frac{X_i}{Z_i} - \frac{Z_i}{X_i}\right)$$

Then coefficient $(a' : 1)$ of image curve of ℓ-isogeny with the kernel $\langle P \rangle$ is computed by

$$\begin{aligned}(a' : 1) &= (\tau(a - 3\sigma) : 1) \\ &= (ac_0c_{\ell-1} - 3(c_0c_{\ell-2} - c_1c_{\ell-1}) : c_{\ell-1}^2) \end{aligned} \tag{7}$$

Using this method, the cost of calculating curve coefficient is $(6d-2)\mathbf{M} + 3\mathbf{S} + 4\mathbf{a}$ in implementation.

Exploiting twisted Edwards curves In [14], Meyer proposed Montgomery-Edwards hybrid method for implementing CSIDH. They exploited the fact that recovering the coefficient of the image curve is more efficient on twisted Edwards curves than Montgomery curves. By using the efficiency of the birational map between Montgomery curves and twisted Edwards curves, they used Montgomery curves for scalar multiplication and isogeny evaluation and used twisted Edwards curves for recovering the coefficient of the image curve.

The outline of the process is summarized in the equation below. In the equation, ϕ denotes an isogeny on a twisted Edwards curve, ι denotes conversion from Montgomery to twisted Edwards curves, and ι^{-1} denotes conversion from twisted Edwards to Montgomery curves.

$$M \xrightarrow{\iota} E \xrightarrow{\psi} E' \xrightarrow{\iota^{-1}} M'$$

By composing the functions $\phi = \iota^{-1} \circ \psi \circ \iota$, one can obtain the coefficient of a Montgomery curve. Using this method, the computational cost of recovering the curve coefficient is $(2d)\mathbf{M} + 6\mathbf{S} + 6\mathbf{a} + 2c(\ell)$, where $c(\ell)$ is the cost for computing r^ℓ for a constant $r \in \mathbb{F}_p$. Details of this method can be found in [14].

degree	**Montgomery [4]**	**Hybrid method [14]**	**2-torsion method [7]**
3	6.4 **M**	8.8 **M**	7.2 **M**
5	12.4 **M**	10.8 **M**	11.2 **M**
7	18.4 **M**	12.8 **M**	15.2 **M**
11	30.4 **M**	16.8 **M**	23.2 **M**
13	36.4 **M**	18.8 **M**	27.2 **M**

Table 1: Computation costs of the coefficient of image curve in original CSIDH, using Edwards curve, using 2-torsion

Remark 2. Given three points P, Q, and $P - Q$ on a Montgomery curve, we can alternatively compute the image curve coefficient with the cost $8\mathbf{M} + 5\mathbf{S} + 11\mathbf{a}$ using differential method [7]. However, unlike SIDH, as CSIDH does not require such three points, additional point evaluation are required to use this method. Thus when differential method is used, CSIDH will have inefficient speed and large key size compared to original method. Therefore, we exclude the use of differential method in this paper.

4 Proposed method

In this section, we present the optimized algorithms for CSIDH group action. First, we briefly state our motivation for this paper. The idea is to use the 2-torsion method to recover the coefficient of the image curve. To use the 2-torsion method in [7], we adjust the prime so that the rational 2-torsion points exist on $\mathbb{F}_p$. The CSIDH using the proposed parameter is performed on the surface. We provide two versions of our modified CSIDH, where one exchanges the 2-torsion points, and the other calculates the 2-torsion point for a given elliptic curve.

4.1 Motivation

As denoted in Section 2, although there is an efficient way for computing 3- and 4- isogenies on Montgomery curves, the original formula in [7] for computing the coefficient of the image curve is inefficient for large degree isogenies. Therefore, Costello and Hisil proposed alternate methods for computing the curve coefficient of the image curve. However, these methods unfit in the CSIDH protocol, as there is no rational 2-torsion points, nor they use the difference of two points as in SIDH. Hence, Castryck et al. compute the coefficient of the image curve using the equation (7).

On the other hand, Meyer et al. exploit twisted Edwards curve for computing the coefficients of the image curve, as there is a simple formula for recovering the coefficient proposed by Moody and Shomow in [15]. Combining Montgomery and twisted Edwards curves, Meyer's method led to speed up of CSIDH protocol. In [12], using Edwards w-coordinate, Kim et al. proposed optimized isogeny formula on Edwards curves, which can be used to implement CSIDH fully on Edwards curves.

To summarize, unlike SIDH, using only Montgomery curves might be an inefficient choice for implementing CSIDH protocol. However, associated in Table 1, if the application of the 2-torsion method is possible, then we can implement CSIDH entirely on Montgomery curves efficiently. Therefore, we provide the way to use the 2-torsion method for computing the coefficients in CSIDH by tailoring the primes used in the base field. The proposed parameter executes CSIDH on the surface. We prove that our method also provides free and transitive group action.

4.2 Proposed Method

In order to use the 2-torsion method, we define a new prime and a new base curve in order to have rational 2-torsion point other than $(0, 0)$. Since 2-isogeny is available in our chosen parameter, we can construct more efficient Montgomery-only CSIDH as in [3].

New parameters Let M_a be a Montgomery curve defined over finite field $\mathbb{F}_p$ where $p \equiv 3 \mod 4$. If E has a 2-torsion point on $\mathbb{F}_p$ except for $(0, 0)$, then the 2-torsion subgroup $M_a(\mathbb{F}_p)[2]$ satisfy $|M_a(\mathbb{F}_p)[2]| = 4$. In this situation, supersingular elliptic curve $M_a/\mathbb{F}_p$ is on the surface satisfying $\mathrm{End}_{\mathbb{F}_p}(M_a) = \mathbb{Z}\left[(1 + \sqrt{-p})/2\right]$ [3]. Note that the original CSIDH uses $p \equiv 3 \mod 8$, so that the supersingular curve $M_a/\mathbb{F}_p$ exists on the floor satisfying $\mathrm{End}_{\mathbb{F}_p}(M_a) = \mathbb{Z}[\sqrt{-p}]$. Thus, in order to have 2-torsion points on $\mathbb{F}_p$, we must use a prime of the form $p \equiv 7 \mod 8$. Following the notation in [3], we define the set $S_p^+ = \{a \in \mathbb{F}_p \mid y^2 = x^3 + ax^2 + x \text{ is supersingular}\}$ and the set of an elliptic curves satisfying $\mathrm{End}_{\mathbb{F}_p}(M_a) = \mathbb{Z}\left[(1 + \sqrt{-p})/2\right]$ is defined by $S^+_{p,\mathbb{Z}\left[(1+\sqrt{-p})/2\right]} = \{A \in S_p^+ \mid \mathrm{End}_{\mathbb{F}_p}(M_a) = \mathbb{Z}\left[(1 + \sqrt{-p})/2\right]\}$. This set splits into two partitions as fol-

lows.

$$S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2],1} = \{a \in S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2]} \mid (0,0) \notin 2M_a(\mathbb{F}_p)\},$$
$$S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2],2} = \{a \in S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2]} \mid (0,0) \in 2M_a(\mathbb{F}_p)\},$$

Since $S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2]}$ consists of two orbits, the group action

$$\mathrm{cl}(\mathcal{O}) \times S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2]} \to S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2]}$$

is free and *not* transitive group action on $S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2]}$. In order to have transitive group action, we refer to the following lemma.

Lemma 1. *Let $p \equiv 7 \bmod 8$ and supersingular Montgomery curve $M_a : y^2 = x^3 + ax^2 + x$ be on the surface. Then there exists $P = (x, y) \in M_a(\mathbb{F}_p)$ such that $[2]P = (0, 0)$ if and only if $a \pm 2$ are both square in $\mathbb{F}_p$.*

Proof. Since M_a is on the surface, there exists a 2-torsion point $(\alpha, 0) \neq (0, 0)$ in $M_a(\mathbb{F}_p)$. So, $A^2 - 4$ must be square in $\mathbb{F}_p$.

Then $a \pm 2$ are both square or both not square in $\mathbb{F}_p$. From $[2]P = ((X + Z)^2(X - Z)^2, -)$ where $x = X/Z$, $[2]P = (0, 0)$ if and only if $X = \pm Z$. i.e., $P = (\pm 1, -)$.

Since P is on the curve M_a, at least one of $1^3 + a \cdot 1^2 + 1 = a + 2$ and $(-1)^3 + a \cdot (-1)^2 + (-1) = a - 2$ must be square in $\mathbb{F}_p$. Therefore, $a \pm 2$ are both square in $\mathbb{F}_p$.

Using this lemma, we can prove the following theorem.

Theorem 2. *Let ϕ be an odd isogeny from M_a to $M_{a'}$ where $a, a' \in S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2]}$. Then*

$$a, a' \in S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2],1} \quad \text{or} \quad a, a' \in S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2],2}$$

Proof. Let $P = (X : Z)$ be a 2-torsion point in $M_a(\mathbb{F}_p)$. Then $P' = (X' : Z') = \phi(X : Z)$ is a 2-torsion point in $M_{a'}$. Since 2-torsion point of Montgomery curve is of the form $(\alpha, 0)$, $a = -(X^2 + Z^2)/XZ$, where $\alpha = X/Z$. Hence, $a \pm 2 = (X \mp Z)^2/(-XZ)$.

Similarly, $a' \pm 2 = (X' \mp Z')^2/(-X'Z')$. Squareness of $a \pm 2$ (resp. $a' \pm 2$) and $-XZ$ (resp. $-X'Z'$) is the same. Also, by the equations (4) and (5), squareness of $-XZ$ and $-X'Z'$ is the same.

Following the proof of Lemma 1, $a \pm 2$ and $a' \pm 2$ are all squares in $\mathbb{F}_p$ or not squares in $\mathbb{F}_p$. Therefore, Theorem 1 holds by Lemma 1.

By Theorem 1, we consider free and transitive group action

$$\mathrm{cl}(\mathcal{O}) \times S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2],i} \to S^+_{p,\mathbb{Z}[(1+\sqrt{-p})/2],i} \tag{8}$$

A 2-torsion point P on a Montgomery curve is always of the form $(\alpha, 0)$. Since $\alpha^2 + a\alpha + 1 = 0$, $\alpha \in \mathbb{F}_p$ or $\alpha \in \mathbb{F}_{p^2}$. The initial curve of the original CSIDH is $y^2 = x^3 + x$, whose x-coordinate of the 2-torsion point is on $\mathbb{F}_{p^2}$, extension field of $\mathbb{F}_p$. So, we need new parameters that offer 2-torsion points in $M_a(\mathbb{F}_p)$ except for $(0, 0)$. The followings are those parameters.

$$p = 2^4 \cdot 3^3 \cdot 5 \cdot \ldots \cdot 373 - 1 \approx 2^{510.1} \tag{9}$$

$$\begin{aligned} a = {} & \text{0x2C36E679F542D63441367BC57EFA26639} \\ & \text{FA0EE9EA65967F55F9D9BAAE672F82BFB} \\ & \text{429BD324D738568EF225AAA1E9F32F8056} \\ & \text{B55B9833D048EE2D99131D655918} \end{aligned} \tag{10}$$

We use the prime $p \equiv 7 \mod 8$ and the Montgomery curve M_a satisfying $|M_a(\mathbb{F}_p)[2]| = 4$. So, we can apply free and transitive group action in the equation (8). Note that using the above 73 consecutive odd primes starting at 3, this parameter provides less security level than the parameters of CSIDH-512. Note that the proposed parameter in this paper is just an example parameter to apply 2-torsion method on CSIDH.

Remark 3. Since $((a\pm2)/p) = -1$, this parameters correspond to $S^{+}_{p,\mathbb{Z}\left[(1+\sqrt{-p})/2\right],1}$

First method : Exchanging the 2-torsion The first method is to exchange 2-torsion points when exchanging a curve. Alice and Bob calculate curve coefficients of image curves using a 2-torsion point when computing the group action and pass it along with the image curve to each other.

Alice computes her secret isogeny $\phi_A : E \to E_A$ with her secret key $[\mathfrak{a}]$, and compute the coefficient of E_A through $\phi_A(T)$. Upon receiving the Bob's public key E_B, Alice also receives $\phi_B(T)$ in order to compute the proceeding phase. Likewise, Bob must also receive Alice's public key E_A and $\phi_A(T)$. As they need to send the image of 2-torsion point as well as the curve, the key size will be $2 \cdot b_p$ bits, where b_p is the number of bits in P.

Summing up the whole process, a class group action by exchanging the 2-torsion is presented in Algorithm 1

Second method : Computing the 2-torsion Note that when using the first method, the key size is twice as much as b_p bits, where b_p bits is the key size of the original CSIDH protocol. This is a huge loss compared to a little increase in speed.

Since a 2-torsion point on a Montgomery curve is of the form $(\alpha, 0)$, we can calculate α through solving a quadratic equation modulo p. Also, as $T_A = \phi_A(T)$ is a 2-torsion point in $E_A(\mathbb{F}_p)$ and $T_B = \phi_B(T)$ is a 2-torsion point in $E_B(\mathbb{F}_p)$, Alice and Bob can directly calculate the 2-torsion point upon the receipt of the image curve computed through each other's secret isogeny.

For $p \equiv 3 \mod 4$, if a is a quadratic residue modulo p, then the square root of a modulo p is computed by $x = a^{(p+1)/4} \mod p$. Using this equation, finding

Algorithm 1 Evaluating the class group action using the first method – Exchanging the 2-torsion

Require: $a \in \mathbb{F}_p$ such that $M_a : y^2 = x^3 + ax^2 + x$ is supersingular curve over $\mathbb{F}_p$ and an integer vector $(e_1, e_2, \cdots, e_n)$ for $e_i \in [-m, m]$, a 2-torsion point T in $M_a(\mathbb{F}_p)$
Ensure: a' such that $M_{a'} : y^2 = x^3 + a'x^2 + x$ where $M_{a'} = [\mathfrak{l}_1^{e_1}\mathfrak{l}_2^{e_2}\cdots\mathfrak{l}_n^{e_n}]M_a$, 2-torsion point $\phi(T)$ in $M_{a'}(\mathbb{F}_p)$ where ϕ is an isogeny from M_a to $M_{a'}$

while some $e_i \neq 0$ **do**
  Sample a random point $P = (x : 1)$ where $x \in \mathbb{F}_p$
  Set $s \leftarrow +1$ if there exist $y \in \mathbb{F}_p$ satisfying $y^2 = x^3 + ax^2 + x$
  Otherwise, $s \leftarrow -1$
  Let $S = \{i \mid e_i \neq 0, \text{sign}(e_i) = s\}$
  if $S = \emptyset$ **then**
    go to line 2
  else
    $k \leftarrow \prod_{i \in S} \ell_i$
    $Q \leftarrow [(p+1)/k]P$
    for $i \in S$ **do**
      $R \leftarrow [k/\ell_i]Q$
      if $R \neq \infty$ **then**
        Compute an isogeny $\phi : M_a \to M_{a'}$ with $\ker \phi = R$
        $a \leftarrow a'$, $T \leftarrow \phi(T)$, $Q \leftarrow \phi(Q)$, $k \leftarrow k/\ell_i$, $e_i \leftarrow e_i - s$
      end if
    end for
  end if
end while
return a', T

a 2-torsion point for a given elliptic curve E is presented in Algorithm 2. By precomputing $2^{-1} \bmod p$, we can get a 2-torsion point with less computation. Note that the cost of Step 4 in Algorithm 2 is very small compared to the total CSIDH algorithm. Also, Algorithm 2 is used only 2 times throughout the total protocol – i.e., Alice computes $E_A = [\mathfrak{a}]E$ using Algorithm 1 with precomputed 2-torsion point. Upon receiving E_B, Alice compute $[\mathfrak{a}]E_B$ using Algorithm 3.

Algorithm 2 Compute a 2-torsion point in $E(\mathbb{F}_p)$

Require: $a \in \mathbb{F}_p$ such th at $M_a : y^2 = x^3 + ax^2 + x$ is supersingular curve
Ensure: A 2-torsion point $T = (\alpha, 0)$ in $M_a(\mathbb{F}_p)$

1: $t \leftarrow a/2$ // 1**M** (with precomputed $2^{-1} \bmod p$)
2: $\alpha \leftarrow t^2$ // 1**S**
3: $\alpha \leftarrow \alpha - 1$
4: $\alpha \leftarrow \alpha^{(p+1)/4}$
5: $\alpha \leftarrow \alpha - t$ // $\alpha = -a/2 + (a^2/4 - 1)^{(p+1)/4}$
6: **return** $T = (\alpha, 0)$

Algorithm 3 Evaluating the class group action using the second method – Computing the 2-torsion

Require: $a \in \mathbb{F}_p$ such that $M_a : y^2 = x^3 + ax^2 + x$ is supersingular curve over $\mathbb{F}_p$ and an integer vector $(e_1, e_2, \cdots, e_n)$ for $e_i \in [-m, m]$
Ensure: a' such that $M_{a'} : y^2 = x^3 + a'x^2 + x$ where $M_{a'} = [\mathfrak{l}_1^{e_1}\mathfrak{l}_2^{e_2}\cdots\mathfrak{l}_n^{e_n}]M_a$
1: Compute a 2-torsion point T in $M_a(\mathbb{F}_p)$ // Algorithm 2
2: **while** some $e_i \neq 0$ **do**
3: Sample a random point $P = (x : 1)$ where $x \in \mathbb{F}_p$
4: Set $s \leftarrow +1$ if there exist $y \in \mathbb{F}_p$ satisfying $y^2 = x^3 + ax^2 + x$
5: Otherwise, $s \leftarrow -1$
6: Let $S = \{i \mid e_i \neq 0, \text{sign}(e_i) = s\}$
7: **if** $S = \emptyset$ **then**
8: go to line 2
9: **else**
10: $k \leftarrow \prod_{i \in S} \ell_i$
11: $Q \leftarrow [(p+1)/k]P$
12: **for** $i \in S$ **do**
13: $R \leftarrow [k/\ell_i]Q$
14: **if** $R \neq \infty$ **then**
15: Compute an isogeny $\phi : M_a \to M_{a'}$ with $\ker \phi = R$
16: $a \leftarrow a'$, $T \leftarrow \phi(T)$, $Q \leftarrow \phi(Q)$, $k \leftarrow k/\ell_i$, $e_i \leftarrow e_i - s$
17: **end if**
18: **end for**
19: **end if**
20: **end while**
21: **return** a'

When Algorithm 2 is used, the key size decreases to b_p bits again, so we can preserve the key size and improve speed. Summing up the whole process, a class group action by computing the 2-torsion point is presented in Algorithm 3. The public key validation can also be performed as in [4] for both methods.

5 Implementation

In this section, we provide the implementation results and analysis. First, we count the number of multiplications and squarings for group action of each algorithm and analyze the results. Then, we compare the performance of the original CSIDH and ours. For clear expression, we shall denote the first method as `Ours_Exchange` and the second method as `Ours_Compute`.

5.1 Parameter and Implementation setup

Parameter setting For implementation, we used the finite field $\mathbb{F}_p$, where p is the prime in the equation (9), and we used the Montgomery coefficient of the initial curve in the equation (10) for both CSIDH and our methods. To make

an exact comparison, we use the field operations implemented in [4] for both CSIDH and our methods. Also, we changed the base prime in [4] to our prime. Note that we assume that $1\mathbf{M} \approx 1\mathbf{S}$, since squaring in [4] is directly implemented by using multiplication.

Further modification Let M_a be a Montgomery curve. In [7], the coefficient of the Montgomery curve is presented as $(\hat{A} : \hat{C}) = (a + 2 : 4)$ instead of $(A : C) = (a : 1)$ for accelerating the doubling (DBL) and differential addition (DBL&ADD) computation. The cost of DBL&ADD decreases from $8\mathbf{M} + 4\mathbf{S} + 11\mathbf{a}$ to $8\mathbf{M} + 4\mathbf{S} + 8\mathbf{a}$ and the cost of DBL decreases from $4\mathbf{M} + 2\mathbf{S} + 7\mathbf{a}$ to $4\mathbf{M} + 2\mathbf{S} + 4\mathbf{a}$, when we used the transformed coefficient. Also, the cost of recovering the coefficient from a 2-torsion point decreases from $2\mathbf{S}+5\mathbf{a}$ to $2\mathbf{S}+3\mathbf{a}$.

The original CSIDH implementation in [4] does not use this transformed coefficient. Although there is an additional cost for converting the form of the coefficients, we can save the cost of scalar multiplication in all ℓ_i-isogeny operation. As this optimization also holds in our proposed method, we applied this technique for both CSIDH and our method. The transformations $(A : C) \leftrightarrow (\hat{A} : \hat{C})$ occurs before and after the group action, where elliptic curve arithmetic are used.

Additionally, we noticed that the optimized point evaluation in the equations (4) and (5) are not used in the implementation of the original CSIDH. For a reasonable comparison, we apply the equations (4) and (5) to the original CSIDH. To summarize, by using the transformed curve coefficient and additional optimization of the point evaluation in CSIDH, the difference in the performance lies purely in the computation of recovering the curve coefficient.

Implementation setup To evaluate the performance of each algorithms, the algorithms are implemented in C language. All cycle counts were obtained on one core of an Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz, running Ubuntu 18.04.3 LTS. For compilation, we used GNU GCC version 7.5.0 with compile option -O3 using the benchmark provided by [4].

5.2 Computational cost

To examine the effect of our proposed method, we first compare the computational cost. Unlike SIDH, since the number of isogeny computation depends on a secret key in CSIDH protocol, we compared the computational cost of the protocol by averaging the number of isogenies used in the protocol. In CSIDH-512, the ℓ_i-isogeny operation occurs on average 2.5 times for each prime ℓ_i, where $e_i \in [-5, 5]$. We set $e_i = 1$ because our goal is to calculate the ratio.

Now, suppose $e_i = 1$ for all i. Then, the number of $\mathbb{F}_p$-multiplication of isogeny operation (xISOG) for the original implementation in [4] is $98690\mathbf{M} + 12341\mathbf{S} \approx 111031\mathbf{M}$, where the number of multiplication reduce to $86354\mathbf{M} + 12341\mathbf{S} \approx 98695\mathbf{M}$, when the equations (4) and (5) are used.

If we use `Ours_Exchange`, then the number of $\mathbb{F}_p$-multiplication cost of isogeny operation with $e_i = 1$ decreases from $98695\mathbf{M}$ to $74016\mathbf{M} + 12342\mathbf{S} \approx 86358\mathbf{M}$, and it can be seen that there is approximately 14.3% cost reduction. This motivates us to implement our methods and the implementation results are in the next section.

5.3 Implementation result

The running time and clock cycles of the group action (resp. the entire key exchange) performed by original CSIDH, `Ours_Exchange`, and `Ours_Compute` are as in Table 2 (resp. Table 3).

	Wall-clock time	**Clock cycles**	**Stack memory**
CSIDH [4]	42.57 **ms**	$89.2 \cdot 10^6$ **cc**	3184 **bytes**
`Ours_Exchange`	34.45 **ms**	$72.2 \cdot 10^6$ **cc**	2784 **bytes**
`Ours_Compute`	37.38 **ms**	$78.3 \cdot 10^6$ **cc**	4368 **bytes**

Table 2: Wall-clock time and clock cycles of group action

	Wall-clock time	**Clock cycles**	**Key size**
CSIDH [4]	193.74 **ms**	$405.9 \cdot 10^6$ **cc**	64 **bytes**
`Ours_Exchange`	152.66 **ms**	$319.9 \cdot 10^6$ **cc**	128 **bytes**
`Ours_Compute`	178.32 **ms**	$373.6 \cdot 10^6$ **cc**	64 **bytes**

Table 3: Wall-clock time and clock cycles of full key exchange, public key size

Since each algorithm is implemented with a non-constant time, we report the average of 1 million runs. As shown in Table 2 and 3, the group action using `Ours_Compute` is about 13.9% faster than the original algorithm, and the entire key exchange is about 8.6% faster than the original CSIDH. In the actual implementation environment, since algorithms include computing kernel points and many point multiplications, the running time of the group action and key exchange is less efficient, compared to that the theoretical cost of point evaluation and curve coefficient.

Meanwhile, optimized CSIDH using twisted Edwards curves is proposed in [12,14], and using the Edwards curve is more efficient than using the 2-torsion method to computing the coefficient of the image curve for higher odd-degree isogenies. However, by using the 2-torsion method, we can simplify the implementation as transformations between Montgomery curves and Edwards curves

are not required. Moreover, by using our method, we provide the fastest performance among the CSIDH implementation, using only Montgomery curves.

6 Conclusion

In this paper, we proposed the optimized method for improving the performance of CSIDH and provided a new parameter to use our method. We set the parameters so that the three 2-torsion points on Montgomery curve are all in $E(\mathbf{F}_p)$. Therefore, by using a 2-torsion point, we optimized the cost of computing the coefficient of the image curve of odd-degree isogeny required in the group action. When our algorithm is used, the group action is about 13.9% faster than the original CSIDH and the entire key exchange is about 8.6% faster than the original CSIDH.

To apply this method, the prime of the base field and the initial elliptic curve must be well-selected for a target security level. If we choose the parameter which enables applying the 2-torsion method, then CSIDH will be optimized further by studying the application of 2-isogeny as in [3].

References

1. Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, David Jao, Brian Koziel, Brian LaMacchia, Patrick Longa, et al. Supersingular isogeny key encapsulation. submission to the NIST post-quantum standardization project, 2017.
2. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-Fish: Efficient isogeny based signatures through class group computations. Technical report, Cryptology ePrint Archive, Report 2019/498, 2019. https://eprint. iacr. org . . . , 2019.
3. Wouter Castryck and Thomas Decru. CSIDH on the surface. Cryptology ePrint Archive, Report 2019/1404, 2019. https://eprint.iacr.org/2019/1404.
4. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.
5. Daniel Cervantes-Vázquez, Mathilde Chenu, Jesús-Javier Chi-Domínguez, Luca De Feo, Francisco Rodríguez-Henríquez, and Benjamin Smith. Stronger and faster side-channel protections for CSIDH. In *International Conference on Cryptology and Information Security in Latin America*, pages 173–193. Springer, 2019.
6. Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.
7. Craig Costello and Huseyin Hisil. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 303–329. Springer, 2017.
8. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291.

9. Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 365–394. Springer, 2018.
10. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *Theory of Cryptography Conference*, pages 341–371. Springer, 2017.
11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.
12. Suhri Kim, Kisoon Yoon, Young-Ho Park, and Seokhie Hong. Optimized method for computing odd-degree isogenies on Edwards curves. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 273–292. Springer, 2019.
13. Michael Meyer, Fabio Campos, and Steffen Reith. On lions and elligators: An efficient constant-time implementation of CSIDH. In *International Conference on Post-Quantum Cryptography*, pages 307–325. Springer, 2019.
14. Michael Meyer and Steffen Reith. A faster way to the CSIDH. In *International Conference on Cryptology in India*, pages 137–152. Springer, 2018.
15. Dustin Moody and Daniel Shumow. Analogues of vélu's formulas for isogenies on alternate models of elliptic curves. *Mathematics of Computation*, 85(300):1929–1951, 2016.