New Hybrid Method for Isogeny-based Cryptosystems using Edwards Curves

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

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

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

[3] Security Algorithm Lab, Samsung SDS, Inc., Seoul, Republic of Korea
jihoon.kwon@samsung.com

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

Abstract. Along with the resistance against quantum computers, isogeny-based cryptography offers attractive cryptosystems due to small key sizes and compatibility with the current elliptic curve primitives. While the state-of-the-art implementation uses Montgomery curves, which facilitates efficient elliptic curve arithmetic and isogeny computations, other forms of elliptic curves can be used to produce an efficient result.

In this paper, we present the new hybrid method for isogeny-based cryptosystem using Edwards curves. Unlike the previous hybrid methods, we exploit Edwards curves for recovering the curve coefficients and Montgomery curves for other operations. To this end, we first carefully examine and compare the computational cost of Montgomery and Edwards isogenies. Then, we fine-tune and tailor Edwards isogenies in order to blend with Montgomery isogenies efficiently. Additionally, we present the implementation results of Supersingular Isogeny Diffie–Hellman (SIDH) key exchange using the proposed method. We demonstrate that our method outperforms the previously proposed hybrid method, and is as fast as Montgomery-only implementation. Our results show that proper use of Edwards curves for isogeny-based cryptosystem can be quite practical.

Keywords: Isogeny, Post-quantum cryptography, Montgomery curves, Edwards curves, SIDH

1 Introduction

The implementation of Shor's algorithm in a quantum computer at a large enough scale would break the intractability assumptions of integer factorization or discrete logarithm problems. Therefore, cryptographic construction based on RSA, DH (Diffie-Hellman key exchange), DSA (Digital Signature Algorithm) or

ECC (Elliptic Curve Cryptography) will no longer be secure. These recent concerns have accelerated the field of post-quantum cryptography. Post-quantum cryptography (PQC) refers to classical cryptosystems which can be run on a classical computer and remain secure even in the presence of a quantum adversary. The main categories of PQC include cryptosystems based on multivariate quadratic equations, lattices, error correcting codes, hash functions, and isogenies. The cryptosystems based on these mathematical problems are expected to be quantum-secure since there is no known quantum algorithm that can solve underlying problems efficiently. Although isogeny-based cryptography is the newest in this field, due to its small key sizes compared to other cryptosystems in PQC, isogeny-based cryptosystems provide a strong candidate for post-quantum key establishment.

The isogeny-based cryptosystem became increasingly popular after the introduction of SIDH key exchange by De Feo and Jao in 2011 [13]. Although a cryptosystem based on isogenies on ordinary curves was first proposed by Couveignes [12] and rediscovered by Stolbunov [21], their algorithm suffered from the quantum sub-exponential attack proposed by Childs et al. [8]. The attack proposed in [8] exploits the commutative property of the endomorphism rings of ordinary curves. As the endomorphism rings of supersingular curves have noncommutative property, SIDH resists against the attack proposed in [8]. To date, the best known classical and quantum attacks against the underlying problem are both exponential so that SIDH has positioned itself as a promising candidate for PQC. In 2017, Supersingular Isogeny Key Encapsulation (SIKE), which is based on SIDH, was submitted as one of the candidates to the NIST standardization project [1].

The potential of an isogeny-based cryptosystem is that it provides significantly smaller key sizes than other PQC primitives while providing the same level of security. However, its state-of-the-art implementation is slower than any other candidates of the NIST standardization project. Therefore, numerous implementations of isogeny-based cryptosystems have been proposed to increase their viability as a PQC candidate. In 2016, Azarderakhsh et al. implemented the SIDH key exchange protocol on ARM-NEON and FPGA devices [2, 16]. Costello et al. proposed faster computation methods and a library for supersingular isogeny key exchange and their implementation remains state-of-the-art [11]. In 2018, Seo et al. proposed a faster modular multiplication for SIDH and SIKE [20]. Their implementation has resulted in additional speed improvements of SIDH and SIKE on ARM processors.

Regarding the implementation, it is important to choose a model of the elliptic curves that provides efficient elliptic curve arithmetic as well as isogeny computation. Owing to the group structure of elliptic curves used in an isogeny-based cryptosystem, either the curve or its twist has a point of order four, which is isomorphic to Montgomery curves or Edwards curves. The state-of-the-art implementation works entirely on Montgomery curves since it provides fast point operations and efficient isogeny computation. However, whether other forms of elliptic curves are as efficient as Montgomery curves is still unclear.

In [9], Costello et al. remarked that there could exist savings to be gained in a twisted Edwards version of SIDH, or some hybrid method that exploits the simple relationship between Montgomery and twisted Edwards curves. Meyer et al. proposed a hybrid SIDH scheme that exploits the fact that the arithmetic in twisted Edwards curves is efficient in some instances [17]. Their method uses twisted Edwards curves for elliptic curve arithmetic and Montgomery curves for isogeny computation. Bos et al. investigated the result of [17] and concluded that using twisted Edwards curves does not result in faster elliptic curve arithmetic in the setting of SIDH [4]. However, Kim et al. recently proposed isogeny formulas on Edwards curves for an isogeny-based cryptosystem and concluded that isogenies on Edwards curves are as efficient as on Montgomery curves [15]. Their work suggests that using Edwards curves instead of twisted Edwards curves for a hybrid method could result in better performance.

The aim of this work is to demonstrate the optimal combination of the usage of Montgomery curves and Edwards curves. This is done by first analyzing the computational costs of the building blocks of SIDH when Edwards curves are used. The following list details the main contributions of this work.

- We further optimized the 4-isogeny formula on Edwards curves proposed in [15]. Our optimization of the 4-isogeny formula on Edwards curves requires $6\mathbf{M}+6\mathbf{S}$, where $\mathbf{M}$ (resp. $\mathbf{S}$) refers to a field multiplication (resp. a field squaring). Additionally, we analyzed the computational cost of the doubling and tripling formulas on Edwards curves used in the isogeny-based cryptosystem. We conclude that except for doubling and differential addition, the computational costs on Edwards curves are as fast as on Montgomery curves.
- We present an optimized 3- and 4- isogeny formulas for SIDH/SIKE. Through a careful examination of the computational costs of isogeny formulas on Edwards and Montgomery curves, we demonstrate that Edwards curves have an advantage over Montgomery curves when recovering the curve coefficients. To exploit Edwards curves, we tailor the Montgomery isogeny formula for efficient computation. Additionally, the use of Edwards curves breaks the dependency between the curve coefficient computation and the evaluation of an isogeny on Montgomery curves. Although only point evaluations are required in some cases, a function that recovers the curve coefficients is called because it precomputes the values needed for the point evaluations. However, by reconstructing the isogeny function on Montgomery curves through isogenies on Edwards curves, we are able to break this dependency. The computational costs of our functions are detailed in Section 4.
- We present the implementation of SIDH using the proposed formula. Previous works on such hybrid methods use isogenies on Montgomery curves and elliptic curve arithmetic on twisted Edwards curves [4, 17]. However, we demonstrate that exploiting Edwards curves when computing isogenies on Montgomery curves leads to better results. Compared with the current state-of-the-art implementation, our hybrid method is faster by 1.09% for the 192-bit security level [10]. Although the result may seem insignificant,

because the isogeny-based cryptosystem is slower than any other candidates in PQC, small speed improvements are meaningful in this field. The results of our experiments are presented in Section 5.

The remainder of this paper is organized as follows: In Section 2, we review some special forms of elliptic curves that are used throughout the paper. Also, the description of the SIDH protocol is presented. In Section 3, we analyze the computational cost of the lower-level functions used in SIDH. The proposed method of exploiting Edwards curves and the implementation results are presented in Section 4. We draw conclusions and discuss future work in Section 5.

2 Preliminaries

In this section, we provide the required background that will be used throughout the paper. First, we introduce the characteristic of Montgomery and Edwards curves and their relations. Then, we introduce the SIDH protocol used in the implementation.

2.1 Montgomery curves and Edwards curves

Montgomery curves Let K be a field with the characteristic not equal to 2 or 3. The Montgomery elliptic curves over K are given by the equation of the form

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

where $b(a^2-4) \neq 0$. The j-invariant of Montgomery curves is defined as $j(M_{a,b}) = 256(a^2-3)^3/(a^2-4)$. When evaluating the isogenous curve coefficients using Vélu's formula, it is efficient to work with both projective coordinates and projective curve coefficients to avoid field inversions [11]. Let $(A : B : C) \in \mathbb{P}^2(K)$ with $C \in \bar{K}^{\times}$, such that $a = A/C$ and $b = B/C$. Then $M_{a,b}$ can be expressed as

$$M_{A:B:C} : By^2 = Cx^3 + Ax^2 + Cx.$$

Arithmetic on Montgomery Curves Montgomery curves are known for fast elliptic curve arithmetic. In [18], Montgomery simplified the computations by dropping the y-coordinate. For example, let $P = (x, y)$ be a point on Montgomery curve $M_{a,b}$, where $x = X/Z$ and $y = Y/Z$. The doubling $[2]P$ can be obtained using only XZ-coordinates as described below.

$$X' = (X - Z)^2(X + Z)^2$$
$$Z' = ((X + Z)^2 - (X - Z)^2) \cdot \left((X + Z)^2 + \frac{a-2}{4}((X + Z)^2 - (X - Z)^2)\right)$$

Additionally, the Montgomery ladder is a method of computing scalar multiples of points on various forms of elliptic curves. This method is only efficient when used on a Montgomery curve [5].

Edwards curves Edwards elliptic curves over K are defined by the following equation:

$$E_d : x^2 + y^2 = 1 + dx^2y^2, \tag{2}$$

where $d \neq 0, 1$. In Edwards curves, the point $(0, 1)$ is the identity element, and the point $(0, -1)$ has order two. The points $(1, 0)$ and $(-1, 0)$ have order four. The condition that E_d always has a rational point of order four restricts the use of elliptic curves in the Edwards model. Twisted Edwards curves are a generalization of Edwards curves, which were proposed by Bernstein et al. [3], to overcome such deficiency. Twisted Edwards curves are defined by the equation,

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2, \tag{3}$$

for distinct nonzero elements $a, d \in K$ [3]. Clearly, $E_{a,d}$ is isomorphic to an Edwards curve over $K(\sqrt{a})$. The j-invariant of Edwards curves is defined as $j(E_d) = 16(1 + 14d + d^2)^3/d(1 - d)^4$. For the same reason as with Montgomery curves, we use projective curve coefficients on Edwards curves to avoid inversion. Let $(C, D) \in \mathbb{P}^2(K)$ where $C \in \bar{K}^\times$ such that $d = D/C$. Then, E_d can be expressed as

$$E_{C:D} : Cx^2 + Cy^2 = C + Dx^2y^2.$$

Arithmetic on Edwards Curves For points (x_1, y_1) and (x_2, y_2) on Edwards curves E_d, the addition of two points is defined as below, and doubling can be performed with the same formula.

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2} \right).$$

In general, projective coordinates $(X : Y : Z) \in \mathbb{P}^2$ where $x = X/Z$ and $y = Y/Z$ are used for the corresponding affine point (x, y) on E_d to avoid inversion during elliptic curve arithmetic. There are many coordinate systems for Edwards curves and, similar to XZ-only Montgomery arithmetic, Castryck et al. proposed YZ-only doubling formulas on Edwards curves [7]. Let $P = (x, y)$ be a point on an Edwards curve E_d. The doubling of P on Edwards curves is given by,

$$[2]P = \left(\frac{2xy}{1 + dx^2y^2}, \frac{y^2 - x^2}{1 - dx^2y^2} \right).$$

Substituting $x^2 = (1-y^2)/(1-dy^2)$ into the y-coordinate of $[2]P$ using the curve equation, we have,

$$\frac{y^2 - x^2}{1 - dx^2y^2} = \frac{y^2(1 - dy^2) - (1 - y^2)}{(1 - dy^2) - dy^2(1 - y^2)} = \frac{-dy^4 + 2y^2 - 1}{dy^4 - 2dy^2 + 1}.$$

Therefore, the second coordinate of $[2]P$ is expressed using only the y-coordinate. Now, let $P = (X : Y : Z)$ be the projective representation of P, such that $x = X/Z$ and $y = Y/Z$. Then, $[2]P = (Y' : Z')$ is given by

$$\begin{aligned} Y' &= -dY^4 + 2Y^2Z^2 - Z^4 \\ Z' &= dY^4 - 2dY^2Z^2 + Z^4, \end{aligned}$$

which is expressed using only the YZ-coordinates. Tripling on Edwards curves can also be expressed in YZ-coordinates. In our implementation of SIDH, we use the YZ-coordinate system on Edwards curves for computational efficiency and compatibility with the XZ-coordinate on Montgomery curves.

Relation between Montgomery Curves and Edwards Curves Generally, every twisted Edwards curve over K is birationally equivalent over K to a Montgomery curve [3]. In [3], Bernstein et al. demonstrated that for a field K with $\#K \equiv 3 \pmod 4$, every Montgomery curve over K is birationally equivalent over K to an Edwards curve. Therefore, to exploit the birationality of Montgomery and Edwards curves, we shall define K with $\#K \equiv 3 \bmod 4$ in the remainder of this paper, unless otherwise specified.

Let d be a nonzero element in K. Then, every Edwards curve E_d is birationally equivalent to a Montgomery form $M_{A,B}$ via

$$(x, y) \to (u, v) = \left(\frac{1+y}{1-y}, \frac{1+y}{(1-y)x}\right), \tag{4}$$

where $A = 2(1+d)/(1-d)$ and $B = 4/(1-d)$. The inverse of the map from $M_{A,B}$ to E_d, is defined as

$$(u, v) \to (x, y) = \left(\frac{u}{v\sqrt{a}}, \frac{u-1}{u+1}\right), \tag{5}$$

where $a = (A+2)/B$ and $d = (A-2)/(A+2)$. The first coordinate in map (4) is computed by using only the y-coordinate and the second coordinate in map (5) uses only the u-coordinate. In projective coordinates, this map becomes remarkably simple [7]. A point $(X_M : Z_M)$ on a Montgomery curve can be transformed to the corresponding Edwards YZ-coordinates $(Y_E : Z_E)$ and vice versa:

$$\begin{aligned}(X_M : Z_M) \to (Y_E : Z_E) &= (X_M - Z_M : X_M + Z_M),\\ (Y_E : Z_E) \to (X_M : Z_M) &= (Y_E + Z_E : Z_E - Y_E).\end{aligned}$$

Therefore, the point conversion between these two curves costs only two additions.

2.2 Supersingular Isogeny Diffie-Hellman

We recall the SIDH key exchange protocol proposed by De Feo and Jao [13]. For more information, please refer to [13]. The notations used in this section will continue to be used throughout the paper.

SIDH protocol Fix two small prime numbers ℓ_A and ℓ_B. Let p be a prime of the form $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$ for some integer cofactor f, and e_A and e_B are positive integers such that $\ell_A^{e_A} \approx \ell_B^{e_B}$. Then we can easily construct a supersingular elliptic curve E over $\mathbb{F}_{p^2}$ of order $(\ell_A^{e_A}\ell_B^{e_B}f)^2$ [6]. We have the full ℓ^e-torsion subgroup on E over $\mathbb{F}_{p^2}$ for $\ell \in \{\ell_A, \ell_B\}$ and $e \in \{e_A, e_B\}$. Choose basis $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ for the $\ell_A^{e_A}$- and $\ell_B^{e_B}$-torsion subgroups, respectively.

Suppose Alice and Bob want to exchange a shared secret key. Let $\{P_A, Q_A\}$ be the basis for Alice and $\{P_B, Q_B\}$ be the basis for Bob. For the key generation, Alice chooses random elements $m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, which are not both divisible by ℓ_A, and computes the subgroup $\langle R_A\rangle = \langle [m_A]P_A + [n_A]Q_A\rangle$. Then using Vélu's formula, Alice computes a curve $E_A = E/\langle R_A\rangle$ and an isogeny $\phi_A : E \rightarrow E_A$ of degree $\ell_A^{e_A}$, where $ker\phi_A = \langle R_A\rangle$. Alice computes and sends $(E_A, \phi_A(P_B), \phi_A(Q_B))$ to Bob. Bob repeats the same operation as Alice, such that Alice receives $(E_B, \phi_B(P_A), \phi_B(Q_A))$.

For the key establishment, Alice computes the subgroup $\langle R'_A\rangle = \langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A)\rangle$. By using Vélu's formula, Alice computes a curve $E_{AB} = E_B/\langle R'_A\rangle$. Bob repeats the same operation as Alice and computes a curve $E_{BA} = E_A/\langle R'_B\rangle$. The shared secret between Alice and Bob is the j-invariant of E_{AB}, i.e., $j(E_{AB}) = j(E_{BA})$.

Computing large degree isogenies In an isogeny-based cryptosystem, one has to compute an ℓ^e-degree isogeny ϕ. The complexity of Vélu's formula scales linearly with the size of the kernel subgroup. Therefore, we decompose an isogeny of degree ℓ^e into e isogenies of degree ℓ for computational efficiency.

Given a cyclic subgroup $\langle R\rangle \in E[\ell^e]$ of order ℓ^e, let ϕ be the isogeny from E to $E/\langle R\rangle$, with $ker\phi = \langle R\rangle \in E[\ell^e]$. The isogeny ϕ is computed as a composition of e isogenies of degree ℓ by Vélu's formula [22]. Starting by setting $E = E_0$ and $R = R_0$, compute $E_{i+1} = E_i/\langle \ell^{e-i-1}R_i\rangle$ for $0 \leq i < e$ [13]. For each iteration, compute an ℓ-isogeny $\phi_i : E_i \rightarrow E_{i+1}$, with $ker\phi_i = \langle \ell^{e-i-1}R_i\rangle$ of order ℓ, and set $R_{i+1} = \phi_i(R_i)$. The point R_i is an ℓ^{e-i} torsion point, such that $[\ell^{e-i-1}]R_i$ has order ℓ. Therefore, by combining $\phi = \phi_{e-1} \circ \cdots \circ \phi_0$ we can obtain the isogeny ϕ of degree ℓ^e with $\langle R\rangle$ as a kernel.

3 Formulas for Constructing Isogeny-based Cryptosystems

In this section, we present the computational costs of lower-level functions that are used as building blocks of SIDH. To avoid inversions during the computation, not only projective coordinates but also projective curve coefficients are used [11]. The formulas presented in this section are the result of considering such circumstances. Additionally, since $\ell_A = 2$ and $\ell_B = 3$ are the common choice for implementing isogeny-based systems, we consider formulas focusing on the doubling (resp. tripling) and 4-isogeny (resp. 3-isogeny). For the projective 4-isogeny formula on Edwards curves, we optimized the formula presented in [15].

3.1 Elliptic Curves Arithmetic in SIDH

The elliptic curves arithmetic on Edwards curves using projective curve coefficients is similar to the case of using twisted Edwards curves. Unlike the currently used ECC, the elliptic curves used in an isogeny-based cryptosystem are not fixed but change as moving around an isogeny class. Therefore, the formula used in SIDH cannot be optimized for specific curve coefficients.

Doubling Let $P = (x, y)$ be a point on an Edwards curve E_d defined as in equation (2). Let $d = D/C$, $x = X/Z$, and $y = Y/Z$. For $P = (Y : Z)$ in projective coordinates, the doubling of P gives $[2]P = (Y' : Z')$, where Y' and Z' are defined as

$$\begin{aligned} Y' &= -DY^4 + 2CY^2Z^2 - CZ^4 \\ Z' &= DY^4 - 2DY^2Z^2 + CZ^4. \end{aligned}$$

The cost of doubling is 5**M**+2**S**.

Tripling For $P = (Y : Z)$ on an Edwards curve E_d represented in projective coordinates, the tripling of P gives $[3]P = (Y' : Z')$, where Y' and Z' are defined as

$$\begin{aligned} Y' &= Y(D^2Y^8 - 6CDY^4Z^4 + 4C^2Y^2Z^6 + 4CDY^2Z^6 - 3C^2Z^8) \\ Z' &= Z(C^2Z^8 - 6CDY^4Z^4 + 4D^2Y^6Z^2 + 4CDY^6Z^2 - 3D^2Y^8). \end{aligned}$$

Let $F = Y' + Z'$ and $G = Y' - Z'$. Then F and G can be written as,

$$\begin{aligned} F = Y' + Z' &= (DY^2(Y^2 - 2YZ) + CZ^2(2YZ - Z^2))^2(Y + Z) \\ G = Y' - Z' &= (DY^2(Y^2 + 2YZ) - CZ^2(2YZ + Z^2))^2(Y - Z). \end{aligned}$$

After computing F and G, the results Y' and Z' can be obtained by computing $F + G$ and $F - G$. The cost of tripling is 7**M**+5**S**.

Differential addition SIDH starts by computing $R = [m]P + [n]Q$ for chosen basis P and Q and a secret key (m, n). For SIDH or SIKE, we may assume that m is invertible, and compute $R = P + [m^{-1}n]Q$. This can be done by using the Montgomery ladder which requires computing differential additions as a subroutine. In the proof of [14], Edwards curves have a similar formula, and we briefly introduce it here.

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two different points on the Edwards curve E_d. Let $P_1 + P_2 = (x_3, y_3)$ and $P_1 - P_2 = (x_4, y_4)$. The addition formula on Edwards curves gives

$$\begin{aligned} y_3(1 - dx_1x_2y_1y_2) &= y_1y_2 - x_1x_2, \\ y_4(1 + dx_1x_2y_1y_2) &= y_1y_2 + x_1x_2. \end{aligned}$$

By multiplying the two equations above, we obtain

$$y_3y_4(1 - d^2x_1^2x_2^2y_1^2y_2^2) = y_1^2y_2^2 - x_1^2x_2^2. \tag{6}$$

Substitute $x_1^2 = \frac{1-y_1^2}{1-dy_1^2}$ and $x_2^2 = \frac{1-y_2^2}{1-dy_2^2}$, and let $y_i = Y_i/Z_i$ for $i = 1, 2, 3, 4$. Then, equation (6) can be rewritten as,

$$\frac{Y_3}{Z_3} = -\frac{(dY_1^2Y_2^2 - Y_2^2Z_1^2 - Y_1^2Z_2^2 + Z_1^2Z_2^2)Z_4}{(dY_1^2Y_2^2 - dY_2^2Z_1^2 - dY_1^2Z_2^2 + Z_1^2Z_2^2)Y_4}. \tag{7}$$

Using the equation (7), the doubling-and-addition for ladder computation on an Edwards curve costs 8**M**+4**S**.

The *j*-invariant In SIDH, the j-invariant of the image curve E_{AB} is used as a shared secret between two parties. The j-invariant of an Edwards curve E_d is defined as

$$j(E_d) = \frac{16(C^2 + 14CD + D^2)^3}{CD(C-D)^4}$$

where $d = D/C$. The computational cost of the j-invariant is 3**M**+4**S**+1**I**, where **I** represents field inversion.

3.2 Isogenies on Edwards Curves

Projective 3-isogenies The formulas for odd-degree isogenies on Edwards curves were first proposed by Moody and Shumow in [19]. They proposed a "multiplicative" isogeny formula on Edwards curves that resulted in better algebraic complexity than the "additive" form of Vélu's formula. Let $P = (\alpha, \beta)$ be a 3-torsion point on an Edwards curve E_d. Then ϕ is a 3-isogeny from E_d to E_d', given by the following equation:

$$\psi(x, y) = \left(\frac{x}{\beta^2} \frac{\beta^2x^3 - \alpha^2y^2}{1 - d^2\alpha^2\beta^2x^2y^2}, \frac{y}{\beta^2} \frac{\beta^2y^2 - \alpha^2x^2}{1 - d^2\alpha^2\beta^2x^2y^2} \right),$$

where $d' = \beta^8 d^3$ and with $\langle P \rangle$ as a kernel. Later, Kim et al. optimized the isogeny formula presented in [19] using projective coordinates, projective curve coefficients, and division polynomials, for use in isogeny-based cryptosystems [15]. Let $P = (\alpha, \beta)$ be a 3-torsion point on Edwards curve E_d, where $\beta = Y_3/Z_3$. Let $\phi : E_d \to E_{d'}$ be a 3-isogeny generated by a kernel $\langle P \rangle$, such that $E_{d'} = E_d/\langle P \rangle$. Let $(Y : Z)$ be the additional input and $(Y' : Z')$ be its corresponding image such that $\phi(Y : Z) = (Y' : Z')$. The projective version of the 3-isogeny on Edwards curves is given as

$$\begin{aligned}(Y' : Z') =& (Y(Z^2Y_3^2 + 2Z^2Y_3Z_3 + Y^2Z_3^2) \\ & : Z(Z^2Y_3^2 + 2Y^2Y_3Z_3 + Y^2Z_3^2)).\end{aligned} \tag{8}$$

The curve coefficients of the isogenous curve $E_{d'}$ are,

$$D' = (Z_3 + 2Y_3)^3 Z_3, \qquad C' = (2Z_3 + Y_3)^3 Y_3, \tag{9}$$

where $d' = D'/C'$. The computational cost for evaluating the 3-isogeny and curve coefficients is 6**M**+5**S** [15].

Projective 4-isogenies In [15], Kim et al. proposed a 4-isogeny formula on Edwards curves by exploiting the efficiency of transforming Edwards curves and Montgomery curves. They combined the transformation between the two curves and the 4-isogeny on Montgomery curves.

Let $P = (\alpha, \beta)$ be a 4-torsion point on an Edwards curve E_d, where $\beta = Y_4/Z_4$. Let $\phi : E_d \to E_{d'}$ be a 4-isogeny generated by a kernel $\langle P \rangle$, such that $E_{d'} = E_d/\langle P \rangle$. Let $(Y : Z)$ be the additional input and $(Y' : Z')$ be its corresponding image such that $\phi(Y : Z) = (Y' : Z')$. The projective version of the 4-isogeny on Edwards curves is given as

$$\begin{aligned} Y' &= (Z^2Y_4^2 + Y^2Z_4^2)YZ(Y_4 + Z_4)^2, \\ Z' &= (Z^2Y_4^2 + Y^2Z_4^2)^2 + 2Y^2Z^2Y_4Z_4(Y_4^2 + Z_4^2). \end{aligned} \tag{10}$$

The curve coefficients of the isogenous curve $E_{d'}$ are,

$$\begin{aligned} D' &= 8Y_4Z_4(Y_4^2 + Z_4^2), \\ C' &= (Y_4 + Z_4)^4. \end{aligned} \tag{11}$$

where $d' = D'/C'$. For the computational cost, we further optimized the result presented in [15]. The computational cost for evaluating the 4-isogeny and curve coefficients is 6**M**+6**S**.

3.3 Summary of the Lower-level Functions

Table 1 summarizes the computational costs of point and isogeny operations on Montgomery and Edwards curves. The `get_4_isog` and `get_3_isog` are functions that compute the coefficients of the isogenous curve. The `eval_4_isog` and `eval_3_isog` are functions that evaluate the isogeny on a given input point. As shown in Table 1, except for the doubling and differential addition for computing the Montgomery ladder, operations on Edwards curves are as efficient as on Montgomery curves, especially when recovering the coefficients of the image curve.

Table 1: Computational costs of lower-level functions on Montgomery and Edwards curves

	Montgomery Curves	Edwards Curves
Differential Addition	6**M**+4**S**	8**M**+4**S**
Doubling	4**M**+2**S**	5**M**+2**S**
`get_4_isog`	4**S**+4**a**+1**s**	4**S**+2**a**+2**s**
`eval_4_isog`	6**M**+2**S**+3**a**+3**s**	6**M**+2**S**+4**a**+3**s**
Tripling	7**M**+5**S**+3**a**+7**s**	7**M**+5**S**+2**a**+7**s**
`get_3_isog`	2**M**+3**S**+12**a**+3**s**	2**M**+3**S**+7**a**+4**s**
`eval_3_isog`	4**M**+2**S**+2**a**+2**s**	4**M**+2**S**+3**a**+3**s**
j-invariant	3**M**+4**S**+1**I**	3**M**+4**S**+1**I**

4 New Hybrid Method for SIDH

In this section, we present a new hybrid method for SIDH. As Edwards curves have benefits when computing isogenies, an optimal combination with Montgomery curves results in better performance than Montgomery-only-SIDH. The proposed idea is to compute the curve coefficient of an image curve using the Edwards formula and modify the evaluation of an isogeny on Montgomery curves with Edwards isogenies. To conclude, we first show the computational cost of the conversion between Montgomery and Edwards curves. Then, we propose efficient 3- and 4- isogeny formulas for SIDH. Lastly, we present our implementation results.

4.1 Switching between Montgomery and Edwards curves

In this subsection, we analyze the additional cost required during the transformation process. Let A_M, B_M, and C_M be the projective curve coefficients of the Montgomery curve $M_{A_M:B_M:C_M}$ and D_E and C_E be the projective curve coefficients of the corresponding Edwards curve $E_{C_E:D_E}$. Fortunately, the arithmetic on the Montgomery curve only uses the curve coefficients A_M and C_M, which correspond to the Edwards curve coefficients C_E and D_E. Instead of storing A_M and C_M, the implementation in $[1, 10]$ stores $A_M + 2C_M$ and $4C_M$ for doubling and $A_M + 2C_M$ and $A_M - 2C_M$ for tripling, for computational efficiency.

Montgomery to Edwards The conversion of a Montgomery curve to an Edwards curve occurs after elliptic curve point operations in order to use Edwards isogenies. Moreover, as the coefficients of an elliptic curve are not used for computing `get_`ℓ`_isog` and `eval_`ℓ`_isog`, where $\ell \in \{3, 4\}$, we may omit the conversion of the curve coefficients. Let $(X_M : Z_M)$ be the projective point on a Montgomery curve $M_{A_M:B_M:C_M}$ and $(Y_E : Z_E)$ be the projective point on an Edwards curve $E_{C_E:D_E}$. The transformation from a Montgomery curve to an Edwards curve on Alice's side is the same as on Bob's side, and is as follows:

$$(X_M : Z_M) \rightarrow (Y_E : Z_E) = (X_M - Z_M : X_M + Z_M)$$

Edwards to Montgomery The conversion of points on an Edwards curve to points on a Montgomery curve occurs after computing `eval_`ℓ`_isog` and the conversion of the curve coefficients occurs after computing `get_`ℓ`_isog`. Let $(X_M : Z_M)$ be the projective point on a Montgomery curve $M_{A_M:B_M:C_M}$ and $(Y_E : Z_E)$ be the projective point on an Edwards curve $E_{C_E:D_E}$. The transformation from an Edwards curve to a Montgomery curve on Alice's side is as follows:

$$\begin{aligned} &(Y_E : Z_E) \rightarrow (X_M : Z_M) = (Y_E + Z_E : Z_E - Y_E) \\ &(C_E : D_E) \rightarrow (A' : C') = (4C_E : 4(C_E - D_E)) = (C_E : (C_E - D_E)) \end{aligned} \tag{12}$$

where $A' = A_M + 2C_M$ and $C' = 4C_M$.

The transformation from an Edwards curve to a Montgomery curve on Bob's side is as follows:

$$(Y_E : Z_E) \to (X_M : Z_M) = (Y_E + Z_E : Z_E - Y_E)$$
$$(C_E : D_E) \to (A' : C') = (4C_E : 4D_E) = (C_E : D_E)$$

where $A' = A_M + 2C_M$ and $C' = A_M - 2C_M$.

As shown in the above equations, there is no additional cost in the conversion of the curve coefficients on Bob's side. Now, we present a method to combine Edwards isogenies and point conversions between two curves efficiently.

4.2 Proposed Method

Efficient isogeny formulas for hybrid SIDH The main idea is to use Edwards curves for recovering the curve coefficients. As denoted in Table 1, Edwards isogenies have fewer field operations than Montgomery isogenies. Moreover, when combined with the transformation between Montgomery and Edwards curves, the computation cost can be reduced even further. For example, instead of computing C_E and D_E, our modified `get_4_isog` on Edwards curves now computes C_E and $C_E - D_E$ as in equation (12). This allows computing $C_E = (Y_4 + Z_4)^4$ and $C_E - D_E = (Y_4 - Z_4)^4$ for a 4-torsion point $(Y_4 : Z_4)$ on an Edwards curve, which requires fewer field operations than standard Edwards isogenies.

However, direct substitution of Montgomery isogenies by Edwards isogenies would increase computational costs due to additional field additions and subtractions during point conversions. As isogenies are computed multiple times during key exchange, such an increase in the number of field additions and subtractions would result in performance degradation. Therefore, because points on a Montgomery curve are converted to points on an Edwards curve to recover the curve coefficients, we tailored the isogeny evaluation function to reuse the converted points. Additionally, we noticed that tripling on an Edwards curve requires fewer field additions than on a Montgomery curve. Hence, we used a similar technique to obtain an Edwards-like Montgomery tripling function. While the input of the tripling function is a point on a Montgomery curve, the tripling is computed using the Edwards tripling formula. The output of the function is a point on a Montgomery curve.

The outline of the modified isogeny evaluation function is summarized in the equation below, and tripling on a Montgomery curve is modified by a similar process. As shown in the equation, the modified evaluation function is merely a composite function, where ϕ denotes an isogeny on an Edwards curve, ι denotes conversion from Montgomery to Edwards curves, and ι^{-1} denotes conversion from Edwards to Montgomery curves.

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

Combining the functions ι, ϕ, and ι^{-1} we obtain an efficient isogeny evaluation and tripling formula on Montgomery curves. Table 4 briefly summarizes the conversion process and form of the curve mainly used for implementing SIDH.

Table 2: Result of the conversion processes and isogeny computation for the proposed method.

	Main curve	Input	Output	Conversion
Tripling	Montgomery	Point on a Montgomery curve	Point on a Montgomery curve	Mont. → Ed. → Mont.
`get_`ℓ`_isog`	Edwards	Points on an Edwards curve	Curve coefficient of a Montgomery curve	Ed. → Mont.
`eval_`ℓ`_isog`	Montgomery	Converted Edwards points Points on a Montgomery curve	Points on a Montgomery curve	Mont. → Ed. → Mont.

Algorithms 1 – 4 illustrate the proposed isogeny formulas. Algorithm 1 describes ways to compute the curve coefficients of the 3-isogenous image curve, given 3-torsion points on an Edwards curve. Let $P = (Y_3 : Z_3)$ be a 3-torsion point on an Edwards curve E_d, which is birationally equivalent to a Montgomery curve $M_{a,b}$. Let $\phi : E_d \to E_{d'}$, where $ker\phi = \langle P \rangle$. Algorithm 1 outputs the curve coefficients of a Montgomery curve $M_{a',b'}$, where $M_{a',b'}$ is birationally equivalent to $E_{d'}$. For an additional curve point $Q = (X : Z)$ on a Montgomery curve $M_{a,b}$, Algorithm 2 outputs $Q' = (X' : Z')$ on a Montgomery curve $M_{a',b'}$, using a 3-torsion point on an Edwards curve. Similarly, Algorithm 3 describes ways to compute curve coefficients of the 4-isogenous image curve, given 4-torsion points on an Edwards curve. Let $P = (Y_4 : Z_4)$ be a 4-torsion point on an Edwards curve E_d, which is birationally equivalent to a Montgomery curve $M_{a,b}$. Let $\phi : E_d \to E_{d'}$, where $ker\phi = \langle P \rangle$. Algorithm 3 outputs the curve coefficients of a Montgomery curve $M_{a',b'}$, where $M_{a',b'}$ is birationally equivalent to $E_{d'}$. For an additional curve point $Q = (X : Z)$ on a Montgomery curve $M_{a,b}$, Algorithm 4 outputs $Q' = (X' : Z')$ on a Montgomery curve $M_{a',b'}$ using 4-torsion points on an Edwards curve.

Proposed hybrid SIDH Combining the results of the above subsections, we describe the proposed hybrid SIDH on Alice's side. The proposed method computes the kernel $\langle R = \langle P + [m_A]Q \rangle$ on a Montgomery curve. The points $R' = [\ell^{e-i-1}]R$ are computed on Montgomery curves. Then R' is converted to a point R'_E on an Edwards curve to obtain the Montgomery curve coefficient of the image curve using the Edwards isogeny formula. The evaluation of the points $(P_B, Q_B, P_B - Q_B)$ on a Montgomery curve is performed by using R'_E and R' on a Montgomery curve. When computing the shared secret key, the points are no longer evaluated and only the curve coefficients are obtained. Without the need to convert back to a Montgomery curve, we use the j-invariant formula of an Edwards curve to reduce the field additions and subtractions. Bob repeats the same operation as Alice in the SIDH protocol, except for the kernel computation. The computation of $R' = [\ell^{e-i-1}]R$ is then obtained on Montgomery curves by using Edwards-like tripling formula. Table 3 compares the computational cost of Montgomery-only SIDH and the proposed hybrid SIDH.

Table 3: Computational costs of key generation stage in Montgomery-only SIDH and proposed hybrid SIDH.

		[10]	Ours
`Alice`	Kernel	6**M**+4**S**	
	Doubling	4**M**+2**S**	
	Point conversion	-	1**a**+1**s**
	`get_4_isog`	4**S**+4**a**+1**s** Mont.	4**S**+2**a** Ed.
	`eval_4_isog`	6**M**+2**S**+3**a**+3**s**	6**M**+2**S**+3**a**+3**s**
`Bob`	Kernel	6**M**+4**S**	
	Tripling	7**M**+5**S**+3a+7**s**	7**M**+5**S**+2a+7**s**
	Point conversion	-	1**a**+1**s**
	`get_3_isog`	2**M**+3**S**+12**a**+3**s** Mont.	2**M**+3**S**+7**a**+2**s** Ed.
	`eval_3_isog`	4**M**+2**S**+2**a**+2**s**	4**M**+2**S**+2**a**+2**s**

To conclude, using Edwards curves for computing the coefficient of the image curve has two benefits. First, as shown in Table 3, the number of field additions and subtractions is reduced. Last but not least, the dependency between `get_`ℓ`_isog` and `eval_`ℓ`_isog` is reduced. Note that for the parameter $p = \ell_A^{e_A}\ell_B^{e_B} \pm 1$, Alice computes the ℓ_A-isogeny e_A times, and Bob computes the ℓ_B-isogeny e_B times. For $0 \leq i \leq e_A - 2$, the curve coefficients of the isogenous curve must be recovered to obtain the kernel for the $i+1$-th isogeny computation. However, at the very last step of the key generation stage, only the evaluation of an isogeny is required, because the evaluated points of the opponent's public parameters — $(\phi_A(P_B), \phi_A(Q_B), \phi_A(P_B - Q_B))$ or $(\phi_B(P_A), \phi_B(Q_A), \phi_B(P_A - Q_A))$ — are exchanged. However, in the implementation of SIDH in [1], `get_`ℓ`_isog` is called at the last step of the key generation stage, because `get_`ℓ`_isog` computes the precomputation values used for `eval_`ℓ`_isog`. On the contrary, when using Edwards isogenies this dependency is reduced, so that `get_`ℓ`_isog` is omitted at the last step of public key generation.

4.3 Implementation Results

In this subsection, we present the implementation results of the proposed SIDH method. We used the finite field $\mathbb{F}_{p^2}$, where p is prime, and $\mathbb{F}_{p^2} = \mathbb{F}_p[X]/(X^2+1)$. For the prime p, we used the 503-bit prime $p_{503} = 2^{250} \cdot 3^{159} - 1$ and the 751-bit prime $p_{751} = 2^{372} \cdot 3^{239} - 1$, presented in [1, 10], which aim at the 128-bit and 192-bit security levels, respectively.

To evaluate the performance, the algorithms are implemented in the C language. We used the SIDH library version 3.1 for SIDH on Montgomery curves [10]. To make an exact comparison, the field operations implemented in the SIDH library were used for both curves which are written in x64 assembly. As a result, the difference in performance lies purely in the choice of the elliptic curve model. All the cycle counts were obtained on one core of an Intel Core i7-8700K at 3.70 GHz, running Ubuntu 16.04 LTS. For the compilation, we used GNU GCC version 5.4.0 with an optimization level -O3.

We first measured the field operations over $\mathbb{F}_{p^2}$ to examine the ratio between each operation. To this end, each field operation was repeated 10^8 times for each prime field. Table 4 summarizes the average cycle counts of field operations over $\mathbb{F}_{p^2}$.

Table 4: Cycle counts of the field operations over $\mathbb{F}_{p^2}$.

Field size	Addition	Subtraction	Multiplication	Squaring	Inversion
p_{503}	41	34	429	322	86,497
p_{751}	61	52	784	591	242,671

As in Table 4, 1**S** equals approximately 0.8**M**, for both the 503-bit prime and 751-bit prime. Combining the results from Table 3 and Table 4, Table 5 compares the performance of SIDH between the Montgomery-only implementation and the proposed hybrid SIDH. For each implementation, we report the average cycles of 10^7 times. As shown in Table 5, the proposed method is 1.03% and 1.09% faster than the Montgomery-only implementation on p_{503} and p_{751}, respectively. The reason for this is that the computational cost of recovering the coefficient of the isogenous curve is reduced when an Edwards isogeny is used. Also, the use of the Edwards-like tripling formula for Montgomery curves has contributed to the overall speed improvement. The implementation is available at https://github.com/CIST-CAL/HybridSIDH.

Table 5: Performance results of SIDH implementation. The results were rounded to the nearest 10^3 clock cycles.

	Montgomery-only [10]		This Work	
	p_{503}	p_{751}	p_{503}	p_{751}
Alice's Keygen	6,348	18,256	6,332	18.009
Bob's Keygen	7,034	20,483	6,985	20,324
Alice's Shared Key	5,180	14,950	5,154	14,828
Bob's Shared Key	6,057	17,589	5,892	17,335
Total	24,619	71,278	24,363	70,496

5 Conclusion and Future Work

In this paper, we proposed a new hybrid method for an SIDH implementation. Although using Edwards curves does not result in better SIDH performance, we noticed that Edwards curves have an advantage in isogeny computations and examined the optimal combination of using Montgomery and Edwards curves. We demonstrated that using Edwards curves for recovering the coefficient of the image curve and exploiting Edwards isogenies for evaluation on a Montgomery curve is faster than the currently proposed hybrid method.

The proposed method reduces the computational costs of isogenies on Montgomery curves in terms of field additions and subtractions. Using the proposed method, our Montgomery-Edwards hybrid SIDH is faster than the standard SIDH by 1.03% and 1.09% for the 128-bit and 192-bit security level, respectively. The hybrid method proposed in this paper is meaningful in two ways: i) it is as fast as the current state-of-the-art implementation and ii) it is faster than the previously proposed Montgomery–twisted Edwards version of hybrid SIDH. We emphasize the fact that using Edwards curves on isogeny-based cryptosystems can be quite practical. Additionally, because recently proposed isogeny-based cryptosystems have the same structure as in SIDH, our implementation results can also be applied.

References

1. Azarderakhsh, R., Campagna, M., Costello, C., Feo, L., Hess, B., Jalali, A., Jao, D., Koziel, B., LaMacchia, B., Longa, P., et al.: Supersingular isogeny key encapsulation. Submission to the NIST Post-Quantum Standardization project (2017)
2. Azarderakhsh, R., Koziel, B., Langroudi, S.H.F., Kermani, M.M.: Fpga-sidh: High-performance implementation of supersingular isogeny diffie-hellman key-exchange protocol on fpga. IACR Cryptology ePrint Archive 2016, 672 (2016)
3. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards curves. In: International Conference on Cryptology in Africa. pp. 389–405. Springer (2008)
4. Bos, J., Friedberger, S.: Arithmetic considerations for isogeny based cryptography. IEEE Transactions on Computers (2018)

5. Bos, J.W., Costello, C., Longa, P., Naehrig, M.: Selecting elliptic curves for cryptography: An efficiency and security analysis. Journal of Cryptographic Engineering 6(4), 259–286 (2016)
6. Bröker, R.: Constructing supersingular elliptic curves. J. Comb. Number Theory 1(3), 269–273 (2009)
7. Castryck, W., Galbraith, S.D., Farashahi, R.R.: Efficient arithmetic on elliptic curves using a mixed Edwards-Montgomery representation. IACR Cryptology ePrint Archive 2008, 218 (2008)
8. Childs, A., Jao, D., Soukharev, V.: Constructing elliptic curve isogenies in quantum subexponential time. Journal of Mathematical Cryptology 8(1), 1–29 (2014)
9. Costello, C., Hisil, H.: A simple and compact algorithm for SIDH with arbitrary degree isogenies. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 303–329. Springer (2017)
10. Costello, C., Longa, P., Naehrig, M.: SIDH library (2016-2018). https://github.com/Microsoft/PQCrypto-SIDH
11. Costello, C., Longa, P., Naehrig, M.: Efficient algorithms for supersingular isogeny diffie-hellman. In: Annual Cryptology Conference. pp. 572–601. Springer (2016)
12. Couveignes, J.M.: Hard homogeneous spaces. IACR Cryptology ePrint Archive 2006, 291 (2006)
13. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: International Workshop on Post-Quantum Cryptography. pp. 19–34. Springer (2011)
14. Justus, B., Loebenberger, D.: Differential addition in generalized Edwards coordinates. In: International Workshop on Security. pp. 316–325. Springer (2010)
15. Kim, S., Yoon, K., Kwon, J., Hong, S., Park, Y.H.: Efficient isogeny computations on twisted Edwards curves. Security and Communication Networks 2018 (2018)
16. Koziel, B., Jalali, A., Azarderakhsh, R., Jao, D., Mozaffari-Kermani, M.: NEON-SIDH: Efficient implementation of supersingular isogeny Diffie-Hellman key exchange protocol on arm. In: International Conference on Cryptology and Network Security. pp. 88–103. Springer (2016)
17. Meyer, M., Reith, S., Campos, F.: On hybrid SIDH schemes using edwards and montgomery curve arithmetic. IACR Cryptology ePrint Archive 2017, 1213 (2017)
18. Montgomery, P.L.: Speeding the pollard and elliptic curve methods of factorization. Mathematics of computation 48(177), 243–264 (1987)
19. Moody, D., Shumow, D.: Analogues of Vélu's formulas for isogenies on alternate models of elliptic curves. Mathematics of Computation 85(300), 1929–1951 (2016)
20. Seo, H., Liu, Z., Longa, P., Hu, Z.: SIDH on ARM: faster modular multiplications for faster post-quantum supersingular isogeny key exchange. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 1–20 (2018)
21. Stolbunov, A.: Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. Adv. in Math. of Comm. 4(2), 215–235 (2010)
22. Vélu, J.: Isogénies entre courbes elliptiques. CR Acad. Sc. Paris. 273, 238–241 (1971)

Algorithm 1 Computing 3-isogeny on Edwards curves

Require: 3-torsion point $P = (Y_3 : Z_3)$ on an Edwards curve E_d

Ensure: The 3-isogenous Montgomery curve with projective curve coefficients C_M/D_M where $C_M = A' + 2C'$ and $D_M = A' - 2C'$.

```
$t_0 \leftarrow Y_3 + Z_3$                  // $t_0 = Y_3 + Z_3$
$t_0 \leftarrow t_0^2$                      // $t_0 = (Y_3 + Z_3)^2$
$t_1 \leftarrow Y_3^2$                      // $t_1 = Y_3^2$
$t_2 \leftarrow Z_3^2$                      // $t_2 = Z_3^2$
$t_3 \leftarrow t_0 - t_1$                  // $t_3 = Z_3^2 + 2Y_3Z_3$
$t_4 \leftarrow t_0 - t_2$                  // $t_4 = Y_3^2 + 2Y_3Z_3$
$C_M \leftarrow t_0 + t_3$                  // $C_M = Y_3^2 + 4Y_3Z_3 + 2Z_3^2$
$t_2 \leftarrow t_2 + t_2$                  // $t_2 = 2Z_3^2$
$C_M \leftarrow C_M + t_2$                  // $C_M = (Y_3 + 2Z_3)^2$
$D_M \leftarrow t_0 + t_4$                  // $D_M = Z_3^2 + 4Y_3Z_3 + 2Y_3^2$
$t_1 \leftarrow t_1 + t_1$                  // $t_1 = 2Y_3^2$
$D_M \leftarrow D_M + t_1$                  // $D_M = (Z_3 + 2Y_3)^2$
$C_M \leftarrow C_M \cdot t4$               // $C_M = (Y_3 + 2Z_3)^3 Y_3$
$D_M \leftarrow D_M \cdot t3$               // $D_M = (Z_3 + 2Y_3)^3 Z_3$
return $C_M, D_M$
```

Algorithm 2 Evaluating 3-isogeny on Montgomery curves

Require: 3-torsion point $P = (Y_3 : Z_3)$ on E_d and a curve point $Q_m = (X_m : Z_m)$ on $M_{a,b}$ corresponding to a point $Q_e = (Y : Z)$ on an Edwards curve E_d.

Ensure: Image point $Q' = (X' : Z')$ on the image curve $M_{a,b}$ birationally equivalent to $\phi(E_d)$

```
$y \leftarrow X - Z$                        // $y$ : convert to point on an Edwards curve
$z \leftarrow X + Z$                        // $z$ : convert to point on an Edwards curve
$t_0 \leftarrow z \cdot Y_3$                // $t_0 = z \cdot Y_3$
$t_1 \leftarrow y \cdot Z_3$                // $t_1 = y \cdot Z_3$
$t_2 \leftarrow t_0 + t_1$                  // $t_2 = (z \cdot Y_3 + y \cdot Z_3)$
$t_2 \leftarrow t_2^2$                      // $t_2 = (z \cdot Y_3 + y \cdot Z_3)^2$
$X' \leftarrow t_2 \cdot X$                 // $X' = (1/2) \cdot (y + z) \cdot (z \cdot Y_3 + y \cdot Z_3)^2$
$t_0 \leftarrow t_0 - t_1$                  // $t_0 = (z \cdot Y_3 - y \cdot Z_3)$
$t_0 \leftarrow t_0^2$                      // $t_0 = (z \cdot Y_3 - y \cdot Z_3)^2$
$Z' \leftarrow t_0 \cdot Z$                 // $Z' = (1/2) \cdot (z - y) \cdot (ZY_3 - YZ_3)^2$
return $X', Z'$
```

Algorithm 3 Computing 4-isogeny on Edwards curves

Require: 4-torsion point $P_m = (X_m : Z_m)$ on a Montgomery curve $M_{a,b}$ corresponding to a 4-torsion point $P_e = (Y_4 : Z_4)$ on an Edwards curve E_d

Ensure: The 4-isogenous Montgomery curve with projective curve coefficients C_M/D_M where $C_M = A' + 2C'$ and $D_M = 4C'$ with coefficient c_0 that are used to evaluate the 4-isogeny

```
$C_M \leftarrow X_m^2$                 // $C_M = (1/4) \cdot (Y_4 + Z_4)^2$
$t_0 \leftarrow Z_m^2$                 // $t_0 = (1/4) \cdot (Y_4 - Z_4)^2$
$c_0 \leftarrow C_M + C_M$             // $c_0 = (1/2) \cdot (Y_4 + Z_4)^2$
$c_0 \leftarrow c_0 + c_0$             // $t_0 = (Y_4 - Z_4)^2$
$C_M \leftarrow C_M^2$                 // $C_M = (1/16) \cdot (Y_4 + Z_4)^4$
$D_M \leftarrow t_0^2$                 // $D_M = (1/16) \cdot (Y_4 - Z_4)^4$
return $C_M, D_M, c_0$
```

Algorithm 4 Evaluating 4-isogeny on Montgomery curves

Require: 4-torsion point $P = (Y_4 : Z_4)$, a curve point $Q = (X_m : Z_m)$ on $M_{a,b}$ corresponding to a point $Q_e = (Y : Z)$ on an Edwards curve E_d, and c_0 computed from Algorithm 3.

Ensure: Image point $Q' = (X' : Z')$ on the image curve $\phi(M_{a,b})$ birationally equivalent to $\phi(E_d)$

```
$z \leftarrow X + Y$                   // $z$ : convert to point on an Edwards curve
$y \leftarrow X - Y$                   // $y$ : convert to point on an Edwards curve
$t_0 \leftarrow y \cdot z$             // $t_0 = YZ$
$t_1 \leftarrow c_0 \cdot t_0$         // $t_2 = YZ(Y_4 + Z_4)^2$
$t_0 \leftarrow Y \cdot Z_4$           // $t_0 = YZ_4$
$t_2 \leftarrow Z \cdot Y_4$           // $t_1 = ZY_4$
$t_3 \leftarrow t_0 + t_2$             // $t_3 = YZ_4 + ZY_4$
$t_4 \leftarrow t_0 - t_2$             // $t_4 = YZ_4 - ZY_4$
$t_3 \leftarrow t_3^2$                 // $t_3 = (YZ_4 + ZY_4)^2$
$t_4 \leftarrow t_4^2$                 // $t_4 = (YZ_4 - ZY_4)^2$
$X' \leftarrow t_1 + t_4$              // $X' = YZ(Y_4 + Z_4)^2 + (YZ_4 - ZY_4)^2$
$Z' \leftarrow t_3 - t_1$              // $Z' = (YZ_4 + ZY_4)^2 - YZ(Y_4 + Z_4)^2$
$X' \leftarrow X' \cdot t_3$           // $X' = (YZ(Y_4 + Z_4)^2 + (YZ_4 - ZY_4)^2)(YZ_4 + ZY_4)^2$
$Z' \leftarrow Z' \cdot t_4$           // $Z' = ((YZ_4 + ZY_4)^2 - YZ(Y_4 + Z_4)^2)(YZ_4 - ZY_4)^2$
return $X', Z'$
```