Optimized Method for Computing Odd-Degree Isogenies on Edwards Curves

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
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. In this paper, we present an efficient method to compute arbitrary odd-degree isogenies on Edwards curves. By using the w-coordinate, we optimized the isogeny formula on Edwards curves by Moody *et al.* The state-of-the-art implementation of isogeny-based cryptosystems works entirely with Montgomery curves since they provide efficient isogeny computation and elliptic curve arithmetic. However, we demonstrated that the same computational costs of elliptic curve arithmetic and isogeny evaluation could be achieved by using the w-coordinate on Edwards curves, with additional benefit when computing isogenous curves. For ℓ-degree isogeny where $\ell = 2s + 1$, our isogeny formula on Edwards curves outperforms Montgomery curves when $s \geq 2$. The result of our work opens the door for the usage of Edwards curves in isogeny-based cryptography, especially in CSIDH which requires higher degree isogenies.

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

1 Introduction

Cryptosystems based on isogenies using supersingular elliptic curves were first proposed by De Feo and Jao [16]. They proposed a Diffie-Hellman type key exchange protocol named Supersingular Isogeny Diffie-Hellman (SIDH). Instead of relying on the discrete logarithm problems where intractability assumption of the problem is broken by Shor's algorithm, the security relies on the problem of finding an isogeny between two given isogenous elliptic curves over a finite field. Moreover, since the key sizes are small compared to other post-quantum cryptography (PQC) categories, isogeny-based cryptography has positioned itself as a promising candidate for PQC. Later, SIDH led to the development of the key encapsulation mechanism called Supersingular Isogeny Key Encapsulation (SIKE), which is a Round 2 candidate in the NIST PQC standardization project [1].

Recently, De Feo *et al.* proposed the improvements to the CRS scheme in [12] and [22]. The CRS scheme was the first cryptosystem based on isogenies between ordinary curves. However, the scheme was highly inefficient and the use of ordinary curves makes the algorithm suffer from the subexponential attack proposed by [8]. The scheme proposed in [13] optimized the CRS scheme, although several minutes are still required for a single key exchange. Independent from [13], Castryck *et al.* proposed CSIDH (Commutative SIDH), which also adapted the CRS scheme, but applied it to supersingular elliptic curves [7]. Instead of working with supersingular elliptic curves over $\mathbb{F}_{p^2}$ as in SIDH/SIKE, CSIDH works over $\mathbb{F}_p$. CSIDH is a non-interactive key exchange protocol having smaller key sizes than SIDH/SIKE.

Considering the implementation, isogeny-based cryptosystems involve complicating isogeny operations in addition to the standard elliptic curve arithmetic over a finite field. Regarding the isogeny operations, the degree of an isogeny used in the cryptosystem depends on the prime chosen for the scheme. For SIDH or SIKE, p is of the form $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$, where ℓ_A and ℓ_B are coprime to each other. The ℓ_A and ℓ_B can be considered as the degree of isogenies used in the scheme. Since the complexity of computing isogenies increases as the degree increases, isogenies of degree 3- and 4- were mostly considered for implementing SIDH or SIKE. CSIDH exploits p of the form $p = 4\ell_1\ell_2\cdots\ell_n - 1$, where ℓ_i are odd-primes. Similarly, ℓ_i are degrees of isogenies used in the scheme, so that demands for odd-degree isogeny formulas have increased after the proposal of CSIDH. Regarding the elliptic curve arithmetic, it is important to select the form of elliptic curves that can provide efficient curve operations. Until recently, only Montgomery curves were used, as they offer fast computations on both components – *i.e.* isogeny computation and curve arithmetic. The state-of-the-art implementation proposed in [11] is also based on Montgomery curves.

Meanwhile, researches have extended to finding new forms of elliptic curves that yield efficient arithmetic or isogeny computation. In [9], it was mentioned that due to the birationality between twisted Edwards curves and Montgomery curves, there might exist savings to be gained when twisted Edwards curves are used for SIDH/SIKE. The utilization of elliptic curve arithmetic on twisted Edwards curves was first proposed by Meyer *et al.* [20]. Their method uses twisted Edwards curves for elliptic curve arithmetic and Montgomery curves for isogeny computation. For isogenies on Edwards curves, optimized 3- and 4- isogeny formulas were first proposed in [17], in order to apply Edwards curves in isogeny-based cryptosystems. In [19], they implemented CSIDH by using Montgomery curves for isogenies and twisted Edwards curves for recovering the coefficient of the image curve.

Currently, using Edwards curves for isogeny-based cryptosystems is not so promising. As Bos *et al.* have demonstrated, working with twisted Edwards curves does not result in faster elliptic curve arithmetic in the setting of SIDH or SIKE [5]. The implementation results in [2] and [18] also show that Edwards curves do not result in faster performance. In short, Edwards curves for implementing SIDH or SIKE have one critical disadvantage – elliptic curve arithmetic

are slower on Edwards curves than on Montgomery curves in SIDH or SIKE settings. When it comes to CSIDH, the most painstaking part is constructing odd-degree isogenies. Although the motivation for the work in [9] is slightly different, the proposed odd-degree isogeny formula can naturally be applied in CSIDH when using Montgomery curves. The only generalized odd-degree isogeny formula on Edwards curves is the formula proposed by Moody *et al.* in [21]. Though, as stated in [19], the coordinate map of the formula is not as simple to compute as in [9].

However, there are still some aspects to optimize the odd-degree isogeny formula on Edwards curves. Until now, the optimization of isogenies on Edwards curves was only done for small degree isogenies. In [17] and [18], the 3- and 4- isogeny formula on Edwards curves were optimized by substituting the x-coordinate and curve coefficients of Moody *et al.*'s formula to y-coordinates using division polynomials and curve equations. As the degree goes higher, optimizing Moody *et al.*'s formula by using the method presented in [17] and [18] is cumbersome. Additional improvements can be achieved on a higher degree isogenies if different approaches are applied for the optimization.

The aim of this work is to construct efficient and generalized odd-degree isogenies on Edwards curves to be suitable for isogeny-based cryptosystems. The following list details the main contributions of this work.

- We exploit the w-coordinate proposed in [14] on Edwards curves. As mentioned above, the main disadvantage of using Edwards curves is that the elliptic curve arithmetic is faster on Montgomery curves in SIDH or SIKE settings. However, the costs of doubling, tripling, and differential addition using projective w-coordinate are the same as on Montgomery curves, which motivates us to use the w-coordinate system on Edwards curves.
- We present the formula for computing odd-degree isogenies using the w-coordinate. By optimizing the isogeny formula proposed by Moody *et al.*, the computational cost of evaluating ℓ-isogeny is the same as on Montgomery curves. We also optimized the formula for obtaining the curve coefficient of the image curve. Our formula for computing the curve coefficient does not require additional points and has benefits over Montgomery curves when the degree is higher than 5. Derivations of our isogeny formula and computational cost are presented in Section 3, and analysis of our isogeny formula is presented in Section 4.
- We present the clock cycles for computing odd-degree isogenies on Edwards curves and give an insight for the usage of Edwards curves for isogeny-based cryptosystems. In this regard, we also compared the cost of lower-level functions used in isogeny-based cryptosystems between Montgomery and Edwards curves. We conclude that the Edwards curves are a suitable choice when implementing CSIDH and Montgomery curves are a suitable choice for SIDH/SIKE. The analyzed results are presented in Section 4.

This paper is organized as follows: In Section 2, we review on Edwards curves and their arithmetic using w-coordinates. Also, the description of the SIDH and

CSIDH protocol are presented. In Section 3, we present our optimization of a generalized odd-degree isogeny formula on Edwards curves. The demonstration of the computational cost of the lower-level functions and isogenies is presented in Section 4. We draw our conclusions and future work in Section 5.

2 Preliminaries

In this section, we provide the required background that will be used throughout the paper. First, we review the Edwards curves and their arithmetic using the w-coordinate. Then, we introduce the SIDH and CSIDH protocol to illustrate the required degree of an isogeny for each protocol.

2.1 Edwards curves and their arithmetic

Edwards Curves Edwards elliptic curves over K are defined by the equation,

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

where $d \neq 0, 1$. The E_d has singular points $(1 : 0 : 0)$ and $(0 : 1 : 0)$ at infinity. 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. Since 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 proposed by Bernstein *et al.* in [3], to overcome such deficiency. Twisted Edwards curves are defined by the equation,

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

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 in [11], we use projective curve coefficients on Edwards curves to avoid inversions when recovering the coefficient of the image curves. 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 exactly the same formula.

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

Generally, 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 inversions during elliptic curve arithmetic. There are many coordinate systems relating to Edwards curves such as inverted coordinates $(X : Y : Z)$ which represents the point $(Z/X, Z/Y)$ on an Edwards curve or extended coordinates which uses $(X : Y : Z : T)$ with $XY = ZT$, for an efficient computation [4, 15].

2.2 w-coordinate on Edwards curve

To evaluate the point addition efficiently, Farashai and Hosseini proposed w-coordinate system on Edwards curves, and we briefly introduce here [14]. In [14], they proposed the rational map w as $w(x,y) = dx^2y^2$ or $w(x,y) = x^2/y^2$ for points (x,y) on an Edwards curve. Since either the map induces the identical result, we shall use the map $w(x,y) = dx^2y^2$ for the explanation.

Define the rational function w by $w(x,y) = dx^2y^2$. This function is well defined for all affine points on an Edwards curve. For $P = (x,y)$ on an Edwards curve E_d, $-P = (-x,y)$ so that $w(P) = w(-P)$. Also, $w(O) = 0$. Let $P_1 = (x_1,y_1)$ and $P_2 = (x_2,y_2)$ be the points on E_d. Let $w_0 = w(2P_1)$, $w_3 = (P_1+P_2)$, and $w_4 = (P_1 - P_2)$. The addition formula on Edwards curves gives

$$\begin{aligned}
x_3(1 + dx_1x_2y_1y_2) &= x_1y_2 + x_2y_1,\\
x_4(1 - dx_1x_2y_1y_2) &= x_1y_2 - x_2y_1,\\
y_3(1 - dx_1x_2y_1y_2) &= y_1y_2 - x_1x_2,\\
y_4(1 + dx_1x_2y_1y_2) &= y_1y_2 + x_1x_2.
\end{aligned}$$

By multiplying the above equations and squaring both sides we have,

$$x_3^2y_3^2x_4^2y_4^2 = \frac{(x_1^2y_2^2 - x_2^2y_1^2)^2(y_1^2y_2^2 - x_1^2x_2^2)}{(1 - d^2x_1^2x_2^2y_1^2y_2^2)^4}$$

Multiplying both sides by d^2 of the above equation, we obtained the differential addition formula as presented in [14]. In [14], the doubling and differential addition formulas are defined as,

$$w_0 = \frac{4w_1((w_1+1)^2 - ew_1)}{(w_1^2-1)^2}, \qquad w_3w_4 = \frac{(w_1-w_2)^2}{(w_1w_2-1)^2}.$$

where $e = 4/d$. For the rest of the subsection, we analyze the computational cost of doubling, tripling, and differential additions in the setting of isogeny-based cryptosystems, using projective w-coordinates. In the remainder of this paper, we shall consider WZ-coordinate as projective w-coordinates. As mentioned above, although we define $w(x,y)$ as $w(x,y) = dx^2y^2$, computational costs are identical when $w(x,y)$ is defined as $w(x,y) = x^2/y^2$. Note that these elliptic curve arithmetic form the building block when implementing isogeny-based cryptosystems.

Doubling Let $P = (x,y)$ be a point on an Edwards curve E_d defined as in equation (1). Let $d = D/C$, $w = dx^2y^2$, and $w = W/Z$. For $P = (W : Z)$ in projective w-coordinates, the doubling of P gives $[2]P = (W' : Z')$, where W' and Z' are defined as

$$\begin{aligned}
W' &= 4WZ(D(W+Z)^2 - 4CWZ)\\
Z' &= D(W+Z)^2(W-Z)^2
\end{aligned}$$

The above equation can be computed as,

$$t_0 = (W+Z)^2, \quad t_1 = (W-Z)^2, \quad t_2 = D \cdot t_0,$$
$$Z' = t_2 \cdot t_1, \quad t_0 = t_0 - t_1, \quad t_1 = C \cdot t_0,$$
$$W' = t_2 - t_1, \quad W' = W' \cdot t_0$$

The computational cost is 4**M**+2**S**.

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

$$W' = W(D(W^2 - Z^2)^2 - Z^2(4D(W+Z)^2 - 16CWZ))^2$$
$$Z' = Z(-D(W^2 - Z^2)^2 + W^2(4D(W+Z)^2 - 16CWZ))^2$$

The computational cost is 7**M**+5**S**.

Differential addition The differential addition is needed when computing the kernel for SIDH or CSIDH. For example, SIDH starts by computing $R = [m]P + [n]Q$ for chosen basis P and Q and a secret key (m, n). Without loss of generality, 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.

Let $P_1 = (W_1 : Z_1)$ and $P_2 = (W_2 : Z_2)$ be the points on E_d. Let $w_0 = (P_1 - P_2)$ and $w_3 = (P_1 + P_2)$, so that $w_0 = W_0/Z_0$, and $w_3 = W_3/Z_3$.

Then,

$$W_3 = Z_0(W_1Z_2 - W_2Z_1)^2$$
$$Z_3 = W_0(W_1W_2 - Z_1Z_2)^2$$

The computational cost of differential addition and doubling on Edwards curves is 6**M**+4**S**.

2.3 Isogeny-based Cryptosystems

We recall the SIDH and CSIDH key exchange protocol proposed in [16] and [7]. For more information, please refer to [16] and [7] for SIDH and CSIDH, respectively. The notations used in this section will continue to be used throughout the paper.

SIDH protocol Fix two coprime 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 be 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 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 secret key. Let $\{P_A, Q_A\}$ be the basis for Alice and $\{P_B, Q_B\}$ be the basis for Bob. For key generation, Alice chooses random elements $m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, 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 Velu's formula, Alice computes a curve $E_A = E/\langle R_A\rangle$ and an isogeny $\phi_A : E \to E_A$ 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 so 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 Velu'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. $j(E_{AB}) = j(E_{BA})$.

CSIDH protocol CSIDH uses commutative group action on supersingular elliptic curves defined over a finite field $\mathbb{F}_p$. Let $\mathcal{O}$ be an imaginary quadratic order. Let $\mathcal{E}\ell\ell_p(\mathcal{O})$ denote the set of elliptic curves defined over $\mathbb{F}_p$ with the endomorphism ring $\mathcal{O}$. It is well-known that the class group $Cl(\mathcal{O})$ acts freely and transitively on $\mathcal{E}\ell\ell_p(\mathcal{O})$. We call the group action as CM-action and denote the action of an ideal class $[\mathfrak{a}] \in Cl(\mathcal{O})$ on an elliptic curve $E \in \mathcal{E}\ell\ell_p(\mathcal{O})$ by $[\mathfrak{a}]E$.

Let $p = 4\ell_1\ell_2\cdots\ell_n - 1$ be a prime where $\ell_1, \cdots, \ell_n$ are small distinct odd primes. Let E be a supersingular elliptic curve over $\mathbb{F}_p$ such that $\mathrm{End}_p(E) = \mathbb{Z}[\pi]$, where $\mathrm{End}_p(E)$ is the endomorphism ring of E over $\mathbb{F}_p$. Note that $\mathrm{End}_p(E)$ is a commutative subring of the quaternion order $\mathrm{End}(E)$. Then the trace of Frobenius is zero, hence $E(\mathbb{F}_p) = p + 1$. Since $\pi^2 - 1 = 0 \mod \ell_i$, the ideal $\ell_i\mathcal{O}$ splits as $\ell_i\mathcal{O} = \mathfrak{l}_i\bar{\mathfrak{l}_i}$, where $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\bar{\mathfrak{l}_i} = (\ell_i, \pi + 1)$. The group action $[\mathfrak{l}_i]E$ (resp. $[\bar{\mathfrak{l}_i}]E$) is computed via isogeny $\phi_{\mathfrak{l}_i}$ (resp. $\phi_{\bar{\mathfrak{l}_i}}$) over $\mathbb{F}_p$ (resp. $\mathbb{F}_{p^2}$) using Velu's formulas.

Suppose Alice and Bob want to exchange a secret key. Alice chooses a vector $(e_1, \cdots, e_n) \in \mathbb{Z}^n$, where $e_i \in [-m, m]$, for a positive integer m. The vector represents an isogeny associated to the group action by the ideal class $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1}\cdots\mathfrak{l}_n^{e_n}]$, where $\mathfrak{l}_i = (\ell_i, \pi - 1)$. Alice computes the public key $E_A := [\mathfrak{a}]E$ and sends E_A to Bob. Bob repeats the similar operation with his secret ideal $\mathfrak{b}$ and sends the public key $E_B := [\mathfrak{b}]E$ to Alice. Upon receiving opponents' public key, Alice computes $[\mathfrak{a}]E_B$ and Bob computes $[\mathfrak{b}]E_A$. Due to the commutativity, $[\mathfrak{a}]E_B$ and $[\mathfrak{b}]E_A$ are isomorphic to each other so that they can derive a shared secret value from the elliptic curves.

3 Optimized odd-degree isogenies on Edwards curve

In this section, we present the optimized method for computing odd-degree isogenies on Edwards curves. We used Moody *et al.*'s result as a base formula and optimized it by using w-coordinates. We conclude that the structure of

odd-degree isogenies on Edwards curves is similar to the coordinate map on Montgomery curves presented in [9].

3.1 Motivation

After the proposal of CSIDH, demands on the general formula for computing odd-degree isogenies have aroused. The prime p in CSIDH is of the form $p = 4\ell_1\ell_2\cdots\ell_n - 1$, where ℓ_i are small distinct odd primes. To implement CSIDH, isogeny of degree ℓ_i is required for all i, $1 \leq i \leq n$. The parameter CSIDH-512 presented in [7] uses $n = 74$, meaning that $\ell_1, \ldots, \ell_{73}$ are smallest 73 odd primes, and ℓ_{74} is a smallest prime distinct from other primes that makes p a prime. Therefore, isogeny formulas of degrees up to at least 587 ($=\ell_{74}$) are required. Although the motivation of the work in [9] is independent of CSIDH scheme, they presented an efficient and generalized odd-degree isogeny formula on Montgomery curves so that the formula can naturally be used for CSIDH. For Edwards curves, optimization of the Moody *et al.*'s formula must be performed for the use in CSIDH and other isogeny-based cryptosystems.

Let G be a subgroup of the Edwards curve E_d with odd order $\ell = 2s + 1$, and points $G = \{(0,1), (\pm\alpha_1, \beta_1), ..., (\pm\alpha_s, \beta_s)\}$. Let ϕ be an ℓ-isogeny from E_d with kernel G. The ϕ proposed by Moody *et al.* is given as follows [21].

$$\phi(x,y) = \left(\frac{x}{B^2} \prod_{i=1}^{s} \frac{\beta_i^2 x^2 - \alpha_i^2 y^2}{1 - d^2\alpha_i^2\beta_i^2 x^2 y^2}, \frac{y}{B^2} \prod_{i=1}^{s} \frac{\beta_i^2 y^2 - \alpha_i^2 x^2}{1 - d^2\alpha_i^2\beta_i^2 x^2 y^2} \right) \tag{3}$$

For optimizing 3-isogeny formula on Edwards curves, Kim *et al.* used the curve equation and division polynomial to represent the x-coordinate and the curve coefficient in equation (3), in terms of y-coordinate [17]. However, for higher degree isogenies, this optimization method is burdensome. On the other hand, the computational costs of elliptic curve arithmetic are the same for both curves when WZ-coordinate and XZ-coordinate are used for Edwards curves and Montgomery curves, respectively. This motivates us to optimize the odd-degree isogeny on Edwards curves using the w-coordinate. For the rest of the section, we present an odd-degree isogeny formula on Edwards curves expressed in w-coordinate.

3.2 Proposed odd-degree isogeny formula

We first present the isogeny formula using the w-coordinate, where the rational function w is defined as $w(x,y) = dx^2y^2$ for points (x,y) on E_d.

Theorem 1. *Let P be a point on the Edwards curve E_d of odd order $\ell = 2s+1$. Let $\langle P\rangle = \{(0,1), (\pm\alpha_1, \beta_1), \cdots, (\pm\alpha_s, \beta_s)\}$, where $P = (\alpha_1, \beta_1)$. Let $w_i = d\alpha_i^2\beta_i^2$ for $1 \leq i \leq s$, and $w = w(Q)$, where $Q = (x,y) \in E_d$. Then for ℓ-isogeny ϕ from E_d to $E_{d'} = E_d/\langle P\rangle$ the evaluation of w, $\phi(w)$, is given by,*

$$\phi(w) = w \prod_{i=1}^{s} \frac{(w - w_i)^2}{(1 - ww_i)^2} \tag{4}$$

Proof. The proof of Theorem 1 is as follows. From the formula proposed by Moody *et al.*, ϕ is as in equation (3), where $d' = B^8 d^\ell$ and $B = \prod_{i=1}^{s} \beta_i$ [21]. In order to use the w-coordinate, we need to express the input and output of an isogeny function in terms of the w-coordinate. The points $(x, y) \in E_d$ and $(\alpha_i, \beta_i) \in E_d$ where $1 \leq i \leq s$, are expressed as $w = dx^2y^2$ and $w_i = d\alpha_i^2\beta_i^2$, in w-coordinates, respectively. Let $\phi(x, y) = (X, Y)$ be the image point. Then $w(\phi(x, y)) = d'X^2Y^2$ so that,

$$d'X^2Y^2 = B^8 d^\ell \cdot \left(\frac{x}{B^2} \prod_{i=1}^{s} \frac{\beta_i^2 x^2 - \alpha_i^2 y^2}{1 - d^2\alpha_i^2\beta_i^2 x^2 y^2} \right)^2 \left(\frac{y}{B^2} \prod_{i=1}^{s} \frac{\beta_i^2 y^2 - \alpha_i^2 x^2}{1 - d^2\alpha_i^2\beta_i^2 x^2 y^2} \right)^2$$

The above equation can be simplified as follows.

$$\begin{aligned} d'X^2Y^2 &= B^8 d^\ell \cdot \frac{x^2}{B^4} \frac{y^2}{B^4} \left(\prod_{i=1}^{s} \frac{\beta_i^2 x^2 - \alpha_i^2 y^2}{1 - d^2\alpha_i^2\beta_i^2 x^2 y^2} \cdot \frac{\beta_i^2 y^2 - \alpha_i^2 x^2}{1 - d^2\alpha_i^2\beta_i^2 x^2 y^2} \right)^2 \\ &= dx^2y^2 \prod_{i=1}^{s} \left(\frac{d(\beta_i^2 x^2 - \alpha_i^2 y^2)(\beta_i^2 y^2 - \alpha_i^2 x^2)}{(1 - d^2\alpha_i^2\beta_i^2 x^2 y^2)^2} \right)^2 \end{aligned}$$

Since $w_i = d\alpha_i^2\beta_i^2$ and $w = dx^2y^2$, the denominator on the inside of the product in the above equation can be simplified as $(1 - ww_i)^4$, which gives,

$$d'X^2Y^2 = w \prod_{i=1}^{s} \frac{(d(\beta_i^2 x^2 - \alpha_i^2 y^2)(\beta_i^2 y^2 - \alpha_i^2 x^2))^2}{(1 - ww_i)^4} \tag{5}$$

Now, the numerator on the inside of the product of equation (5) can be simplified as follows.

$$\begin{aligned} (d(\beta_i^2 x^2 - \alpha_i^2 y^2)(\beta_i^2 y^2 - \alpha_i^2 x^2))^2 &= (d(x^2y^2\beta_i^4 - \alpha_i^2\beta_i^2 x^4 - \alpha_i^2\beta_i^2 y^4 - x^2y^2\alpha_i^4))^2 \\ &= (w(\alpha_i^4 + \beta_i^4) - w_i(x^4 + y^4))^2 \end{aligned} \tag{6}$$

For further simplification of equation (6) we use the curve equation. Note that (α_i, β_i) and (x, y) are on the Edwards curve E_d. Then, $\alpha_i^2 + \beta_i^2 = 1 + w_i$ so that

$$\begin{aligned} \alpha_i^4 + \beta_i^4 &= (1 + w_i)^2 - 2\alpha_i^2\beta_i^2 \\ &= (1 + w_i)^2 - 2w_i/d \end{aligned}$$

Similarly for the point (x, y), we have $x^4 + y^4 = (1 + w)^2 - 2w/d$. Substituting the result to equation (6), we have,

$$\begin{aligned} (d(\beta_i^2 x^2 - \alpha_i^2 y^2)(\beta_i^2 y^2 - \alpha_i^2 x^2))^2 &= \left(w \left((1 + w_i)^2 - \frac{2w_i}{d} \right) - w_i \left((1 + w)^2 - \frac{2w}{d} \right) \right)^2 \\ &= ((w - w_i)(1 - ww_i))^2 \end{aligned}$$

Now if we substitute the above equation to equation (5), we have

$$\begin{aligned} d'X^2Y^2 &= w\prod_{i=1}^{s}\frac{((w-w_i)(1-ww_i))^2}{(1-ww_i)^4} \\ &= w\prod_{i=1}^{s}\frac{(w-w_i)^2}{(1-ww_i)^2} \end{aligned}$$

, which gives the desired result. □

Theorem 1 shows that the evaluation of an isogeny on Edwards curves can be expressed in w-coordinate. Now, it remains to express the coefficient of the image curve in w-coordinates. From the formula proposed by Moody *et al.*, the curve coefficient d' of the image curve $E_{d'}$ is $d' = d^\ell B^8$ where $B = \prod_{i=1}^{s}\beta_i$. Since (α_i, β_i) satisfies the curve equation, $\alpha_i = (1-\beta_i^2)/(1-d\beta_i^2)$ so that

$$\begin{aligned} w_i &= d\alpha_i^2\beta_i^2 \\ &= d\left(\frac{1-\beta_i^2}{1-d\beta_i^2}\right)\beta_i^2 \end{aligned}$$

Solving the above equation for β_i^2, we can express the curve coefficient of the image curve in w-coordinate. However, direct change of d' to w-coordinate is computationally inefficient due to the square root computation. To solve this problem, we refer to the following theorem. Let $P_i = (\alpha_i, \beta_i) \in \langle P\rangle$ for $1 \le i \le s$, where $-P_i = (-\alpha_i, \beta_i)$. We exploit the fact that the set of y-coordinates of $[2]P_i$ where $1 \le i \le s$, is equal to the set of y-coordinates of P_j, where $1 \le j \le s$, up to a permutation.

Theorem 2. *The curve coefficient d' of the image curve $E_{d'}$ in Theorem 1 is equal to*

$$d' = d^\ell \prod_{i=1}^{s}\frac{(w_i+1)^8}{4^4} \tag{7}$$

Proof. The proof of the Theorem 2 is as follows. From the formula proposed by Moody *et al.*, $d' = d^\ell B^8$ where $B = \prod_{i=1}^{s}\beta_i$. In order to use w-coordinate system for isogeny computations, we also need to express d' in w-coordinate. As denoted above, converting β_i directly to w-coordinate is cumbersome. The idea is that doubling the kernel points also generates the same subgroup since we are only dealing with odd-degree isogenies.

Let $P_i = (\alpha_i, \beta_i)$. Instead of computing the square of the y-coordinate (or x-coordinate) of P_i, we shall compute the square of the y-coordinate (or x-coordinate) of $[2]P_i$. Note that since P is an ℓ-torsion point where $\ell = 2s+1$, $[2]P_i = \pm P_j$ for some $i, j \in \{1, ..., s\}$. Then from the addition formula on Edwards curves, we have

$$[2]P_i = \left(\frac{2\alpha_i\beta_i}{1+d\alpha_i^2\beta_i^2}, \frac{\beta_i^2-\alpha_i^2}{1-d\alpha_i^2\beta_i^2}\right)$$

Squaring the x-coordinate of $[2]P_i$, we have

$$\left(\frac{2\alpha_i\beta_i}{1+d\alpha_i^2\beta_i^2}\right)^2 = \frac{4\alpha_i^2\beta_i^2}{(1+w_i)^2} = \frac{4w_i/d}{(1+w_i)^2}$$

Since $w_i = d\alpha_i^2\beta_i^2$, $\beta_i^2 = w_i/d\alpha_i^2$. Hence, by substituting the results, we have

$$d' = d^\ell \prod_{i=1}^{s} \beta_i^8 = d^\ell \prod_{i=1}^{s} \frac{(w_i+1)^8}{4^4}$$

which gives the desired result. □

3.3 Alternate odd-degree isogeny formula

In this section, we present the isogeny formula by defining the rational function w as $w(x,y) = x^2/y^2$ for a point (x,y) on E_d. As shown below, the cost of evaluating isogenies is the same as the case when $w(x,y) = dx^2y^2$. Formulas for computing the coefficient of the image curve are similar in both cases.

Theorem 3. *Let P be a point on the Edwards curve E_d of odd order $\ell = 2s+1$. Let $\langle P\rangle = \{(0,1), (\pm\alpha_1, \beta_1), \cdots, (\pm\alpha_s, \beta_s)\}$, where $P = (\alpha_1, \beta_1)$. Let $w_i = \alpha_i^2/\beta_i^2$ for $1 \le i \le s$. and $w = w(Q)$, where $Q = (x,y) \in E_d$. Then for ℓ-isogeny ϕ from E_d to $E_{d'} = E_d/\langle P\rangle$ the evaluation of w, $\phi(w)$, is given by,*

$$\phi(w) = w \prod_{i=1}^{s} \frac{(w-w_i)^2}{(1-ww_i)^2} \tag{8}$$

Proof. The proof of Theorem 3 is similar to the proof of Theorem 1. From the formula proposed by Moody *et al.*, ϕ is given by equation (3). The points $(x,y) \in E_d$ and $(\alpha_i, \beta_i) \in E_d$, where $1 \le i \le s$, are expressed as $w = x^2/y^2$ and $w_i = \alpha_i^2/\beta_i^2$ in w-coordinates, respectively. Let $\phi(x,y) = (X,Y)$ be the image point. Then $\phi(x,y)$ can be expressed in w-coordinate as,

$$\phi(w) = \frac{X^2}{Y^2} = \frac{x^2}{y^2} \prod_{i=1}^{s} \frac{(\beta_i^2x^2 - \alpha_i^2y^2)^2}{(\beta_i^2y^2 - \alpha_i^2x^2)^2}$$

Simplifying the equation and express in w-coordinate, we obtain $\phi(w)$ as in equation (8). □

To obtain the coefficient of the image curve, we refer to the following theorem.

Theorem 4. *The curve coefficient d' of the image curve $E_{d'}$ in Theorem 1 is equal to*

$$d' = d^{\ell} \prod_{i=1}^{s} \frac{4^4}{(w_i+1)^8} \tag{9}$$

Proof. Let $P_i = (\alpha_i, \beta_i)$ be the point of the kernel. Similar to the proof of the Theorem 2, the Theorem 4 exploits the square of the x-coordinate of $[2]P_i$. From the addition formula on Edwards curves, we have

$$[2]P_i = \left(\frac{2\alpha_i\beta_i}{1+d\alpha_i^2\beta_i^2}, \frac{\beta_i^2-\alpha_i^2}{1-d\alpha_i^2\beta_i^2} \right)$$

Squaring the x-coordinate of $[2]P_i$ and dividing both the denominator and numerator by β_i^4, we have,

$$\begin{aligned} \frac{4\alpha_i^2\beta_i^2}{(1+d\alpha_i^2\beta_i^2)^2} &= \frac{4\alpha_i^2\beta_i^2}{(\alpha_i^2+\beta_i^2)^2} \\ &= \frac{4w_i}{(1+w_i)^2} \end{aligned}$$

Now, since $w_i = \alpha_i^2/\beta_i^2$, $\beta_i^2 = \alpha_i^2/w_i$ so that

$$\begin{aligned} d' &= d^{\ell} \prod_{i=1}^{s} \beta_i^8 \\ &= d^{\ell} \prod_{i=1}^{s} \left(\frac{\alpha_i^2}{w_i} \right)^4 \\ &= d^{\ell} \prod_{i=1}^{s} \frac{4^4}{(w_i+1)^8} \end{aligned}$$

which gives the desired result. □

4 Implementation

In this section, we provide the implementation results of our odd-degree isogeny formulas. We first compare the computational costs between Montgomery and Edwards curves. We then show the cycle counts of ℓ-isogeny for $\ell \in \{3, 5, 7, 9\}$.

4.1 Computational costs

To evaluate the computational costs of the proposed formula, we first projectivize the function into $\mathbb{P}^1$ to avoid inversions. Since both rational maps induce the similar formula, we shall explain this section by defining the rational map as $w(x, y) = x^2/y^2$ for points (x, y) on Edwards curves. Thus, for $(\alpha_i, \beta_i) \in E_d$,

$(W_i : Z_i) = (w_i : 1)$ for $i = 1, ..., s$ where $w_i = \alpha_i^2/\beta_i^2$. Let ϕ be a degree ℓ isogeny from E_d to $E_{d'}$. For additional input point $(W : Z)$ on the curve E_d, the output is expressed as $(W' : Z')$ where $(W' : Z') = \phi(W : Z)$. Then,

$$W' = W \cdot \prod_{i=1}^{s} (WZ_i - ZW_i)^2$$
$$Z' = Z \cdot \prod_{i=1}^{s} (WW_i - ZZ_i)^2$$

Let $F = (W - Z)(W_i + Z_i)$ and $G = (W + Z)(W_i - Z_i)$. Then the above equation can be rewritten as,

$$W' = W \cdot \prod_{i=1}^{s} (F - G)^2$$
$$Z' = Z \cdot \prod_{i=1}^{s} (F + G)^2$$

Therefore, computation of $(WZ_i - ZW_i)$ and $(WW_i - ZZ_i)$ cost $2\mathbf{M}+6\mathbf{a}$, where $\mathbf{M}$ and $\mathbf{a}$ refers to a field multiplication and addition, respectively. For $\ell = 2s + 1$-isogeny, evaluation of an isogeny costs $(4s)\mathbf{M}+2\mathbf{S}$, where $\mathbf{s}$ refers to a field squaring. To compute the curve coefficients, let $d = D/C$. Then we have,

$$D' = D^\ell \cdot \prod_{i=1}^{s} (2Z_i)^8$$
$$C' = C^\ell \cdot \prod_{i=1}^{s} (W_i + Z_i)^8$$

where $d' = D'/C'$. Concluding the section, Table 1 presents the computational costs of evaluation of an isogeny as well as curve coefficient for degree $\ell \in \{3, 5, 7, 9\}$.

As shown in Table 1, the computational costs of evaluating isogenies are identical on both curves. In Table 1, we used the 2-torsion method for Montgomery curves to analyze the computational costs of computing the coefficients. In [9], instead of directly computing the curve coefficients, they exploit the fact that pushing 2-torsion points through an odd-degree isogeny preserves their order on the image curve. When the image of the 2-torsion point is obtained, the curve coefficient of the image curve can be recovered in $2\mathbf{S}+5\mathbf{a}$. For the details of the method, please refer to [10].

Table 1: Computational costs of isogenies of degree 3, 5, 7, and 9 on Montgomery cures and Edwards curves. For computing the curve coefficients on Montgomery curve, 2-torsion method is used, and the table presents the combined computational cost of evaluating image of the 2-torsion point ($(4s)\mathbf{M}+2\mathbf{S}$) and recovering curve coefficient ($2\mathbf{S}$).

	Evaluation		Curve coefficient	
	Montgomery	Edwards (This Work)	Montgomery	Edwards (This Work)
3	$4\mathbf{M}+2\mathbf{S}$	$4\mathbf{M}+2\mathbf{S}$	$2\mathbf{M}+3\mathbf{S}$	$4\mathbf{M}+6\mathbf{S}$
5	$8\mathbf{M}+2\mathbf{S}$	$8\mathbf{M}+2\mathbf{S}$	$8\mathbf{M}+4\mathbf{S}$	$6\mathbf{M}+6\mathbf{S}$
7	$12\mathbf{M}+2\mathbf{S}$	$12\mathbf{M}+2\mathbf{S}$	$12\mathbf{M}+4\mathbf{S}$	$8\mathbf{M}+6\mathbf{S}$
9	$16\mathbf{M}+2\mathbf{S}$	$16\mathbf{M}+2\mathbf{S}$	$16\mathbf{M}+4\mathbf{S}$	$10\mathbf{M}+6\mathbf{S}$

Since an additional 2-torsion point is evaluated, the computational cost of recovering the curve coefficient of the image curve is equal to $(4s)\mathbf{M}+4\mathbf{S}$, where $(4s)\mathbf{M}+2\mathbf{S}$ is for isogeny evaluation and $2\mathbf{S}$ is for recovering from image points. One drawback of the 2-torsion method is that the additional 2-torsion point must be evaluated to recover the curve coefficient. Therefore, the computational cost of obtaining the curve coefficient of the image curve increases as the degree of isogeny increases. Although this is also the case on Edwards curves, an additional 2-torsion point is not required for Edwards curves.

For Montgomery curves, curve coefficients can also be recovered using the x-coordinates of points and the x-coordinate of their differences – *i.e.* x-coordinates of the points P, Q, and $Q-P$ on a Montgomery curve [9]. We shall call this method as `get_a_from_diff` method. Recovering the curve coefficient using this method costs $8\mathbf{M}+5\mathbf{S}+11\mathbf{a}$ and the cost does not increase even if the degree of isogeny increases. In SIDH/SIKE settings, the points P, Q, and $Q-P$ can be seen as a public key $(P_A, Q_A, P_A - Q_A)$ (or $(P_B, Q_B, P_B - Q_B)$ on Bob's side) and are evaluated for each iteration for efficient ladder computations. Therefore, `get_a_from_diff` method are more efficient in SIDH than the 2-torsion method.

Figure 1 depicts the difference in the computational cost of recovering the curve coefficient between Montgomery curves and Edwards curves. The horizontal axis represents the degree of an isogeny and vertical axis represents the number of multiplication used for the computation. The blue line indicates the computational cost on Montgomery curves and the orange line indicates the computational cost on Edwards curves. We considered 1S as 0.8M. Note that when WZ-coordinate is used for Edwards curves and XZ-coordinate is used for Montgomery curves, the difference in the performance purely lies on the cost of recovering the coefficients of the image curve, because the costs of all the remaining operations are the same. As shown in Figure 1.(a), when the 2-torsion method is used on Montgomery curves, Edwards curves become more efficient as the degree of isogeny increases. On the other hand, as shown in Figure 1.(b), when `get_a_from_diff` method is used for Montgomery curves, Montgomery

curves become more efficient as the degree of isogeny increases. More concretely, Montgomery curves are preferred in SIDH/SIKE settings and are more efficient then Edwards curves for $s \geq 3$. In CSIDH setting, the points P, Q, and $Q - P$ are not evaluated so that the 2-torsion method is used for Montgomery curves. Hence Edwards curves are preferred and are more efficient than Montgomery curves in CSIDH for $s \geq 2$.

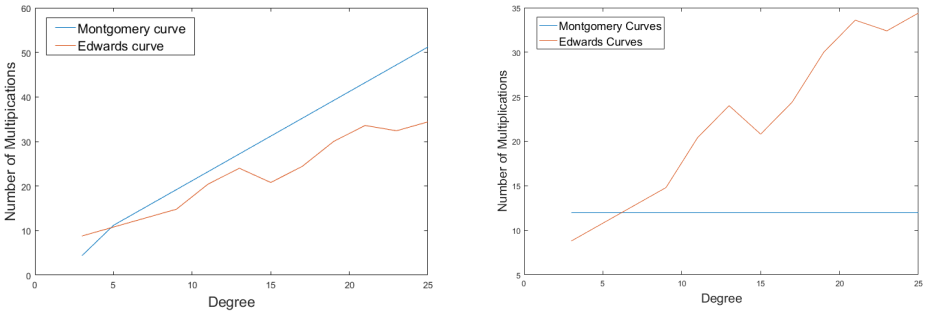

Fig. 1: (a) Computational costs of recovering the curve coefficient of the image curve when the 2-torsion method is used for Montgomery curves. (b) Computational costs of recovering the curve coefficient of the image curve when `get_a_from_diff` method is used for Montgomery curves.

4.2 Odd-degree isogenies on Edwards curves

To evaluate the performance, the algorithms are implemented in C language. We use the field $\mathbb{F}_{p^2}$, where p is prime and $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$ for $i^2 = -1$. For the prime p, we used the 751-bit prime $p_{751} = 2^{372} \cdot 3^{279} - 1$, presented in [2,10]. The field arithmetic implemented in SIDH library was used for the implementation [10]. All cycle counts were obtained on one core of an Intel Core i7-6700 (Skylake) at 3.40 GHz, running Ubuntu 16.04 LTS. For compilation, we used GNU GCC version 5.4.0.

The Table 2 presents the computational costs and cycle counts of the elliptic curve arithmetic and isogeny computations. The `get_`ℓ`_isog` functions compute the curve coefficient of the image curve of ℓ-isogeny. The `eval_`ℓ`_isog` functions represents the evaluation of the ℓ-isogeny. For the `get_`ℓ`_isog` functions on Montgomery curves, 2-torsion method presented in [9] is used.

As shown in Table 2, the cost of computing the elliptic curve arithmetic and `eval_`ℓ`_isog` are identical on both curves. Edwards curves has a benefit over Montgomery curves on `get_`ℓ`_isog` functions for $s \geq 2$ when the 2-torsion method is used for Montgomery curves. When `get_a_from_diff` method is used for Montgomery curves, Montgomery curves are efficient for odd-degrees higher than $s \geq 3$. We can conclude that when implementing isogeny-based cryptosystems, Montgomery curves are efficient for SIDH or SIKE and Edwards curves are efficient for CSIDH.

Table 2: Computational costs and cycle counts of lower-level functions on Montgomery and Edwards curves

	Montgomery	Edwards (This Work)
Differential addition	6**M**+4**S**	
Doubling	4**M**+2**S**	
	5,539	5,531
Tripling	7**M**+5**S**	
	10,912	10,918
`get_3_isog`	2**M**+3**S**	4**M**+6**S**
	5,146	8,578
`eval_3_isog`	4**M**+2**S**	
	5,467	5,548
`get_5_isog`	8**M**+2**S**	6**M**+6**S**
	11,346	10,228
`eval_5_isog`	8**M**+2**S**	
	9,572	9,597
`get_7_isog`	12**M**+4**S**	8**M**+6**S**
	15,348	12,126
`eval_7_isog`	12**M**+2**S**	
	13,644	13,676

5 Conclusion

In this paper, we proposed the optimized method for computing odd-degree isogenies on Edwards curves. By using the w-coordinates, we optimized the isogeny formula proposed by Moody *et al.* The use of the w-coordinate makes the costs of elliptic curve arithmetic and evaluation of an isogeny identical to that of on Montgomery curves, having efficiency when computing the coefficient of the image curve. For ℓ-degree isogeny where $\ell = 2s + 1$, the proposed formula has benefit over Montgomery curves when $s \geq 2$. We conclude that if degree $\ell = 2s + 1$ where $s \geq 3$ is used for implementing isogeny-based cryptosystems, Montgomery curves are efficient for SIDH or SIKE. For $s \geq 2$, Edwards curves are an efficient choice for CSIDH. For the future work, we plan to implement CSIDH using w-coordinate on Edwards curves.

References

1. Azarderakhsh, R., Campagna, M., Costello, C., De 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., Lang, E.B., Jao, D., Koziel, B.: Edsidh: Supersingular isogeny diffie-hellman key exchange on edwards curves. In: International Conference on Security, Privacy, and Applied Cryptography Engineering. pp. 125–141. Springer (2018)
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. Bernstein, D.J., Lange, T.: Inverted edwards coordinates. In: International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes. pp. 20–27. Springer (2007)
5. Bos, J., Friedberger, S.: Arithmetic considerations for isogeny based cryptography. IEEE Transactions on Computers (2018)
6. Bröker, R.: Constructing supersingular elliptic curves. J. Comb. Number Theory 1(3), 269–273 (2009)
7. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: Csidh: An efficient post-quantum commutative group action. Tech. rep., Cryptology ePrint Archive, Report 2018/383 (2018)
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)
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. De Feo, L., Kieffer, J., Smith, B.: Towards practical key exchange from ordinary isogeny graphs. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 365–394. Springer (2018)
14. Farashahi, R.R., Hosseini, S.G.: Differential addition on twisted edwards curves. In: Pieprzyk, J., Suriadi, S. (eds.) Information Security and Privacy. pp. 366–378. Springer International Publishing, Cham (2017)
15. Hisil, H., Wong, K.K.H., Carter, G., Dawson, E.: Twisted edwards curves revisited. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 326–343. Springer (2008)
16. 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)
17. Kim, S., Yoon, K., Kwon, J., Hong, S., Park, Y.H.: Efficient isogeny computations on twisted edwards curves. Security and Communication Networks 2018 (2018)
18. Kim, S., Yoon, K., Kwon, J., Park, Y.H., Hong, S.: New hybrid method for isogeny-based cryptosystems using edwards curves
19. Meyer, M., Reith, S.: A faster way to the csidh. In: International Conference in Cryptology in India. pp. 137–152. Springer (2018)
20. Meyer, M., Reith, S., Campos, F.: On hybrid sidh schemes using edwards and montgomery curve arithmetic
21. 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)

22. 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)