How to Construct CSIDH on Edwards Curves

Tomoki Moriya[1], Hiroshi Onuki[1], and Tsuyoshi Takagi[1]

Department of Mathematical Informatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{tomoki_moriya,onuki,takagi}@mist.i.u-tokyo.ac.jp

Abstract. CSIDH is an isogeny-based key exchange protocol proposed by Castryck *et al.* in 2018. It is based on the ideal class group action on $\mathbb{F}_p$-isomorphism classes of Montgomery curves. The original CSIDH algorithm requires a calculation over $\mathbb{F}_p$ by representing points as x-coordinate over Montgomery curves. There is a special coordinate on Edwards curves (the w-coordinate) to calculate group operations and isogenies. If we try to calculate the class group action on Edwards curves by using the w-coordinate in a similar way on Montgomery curves, we have to consider points defined over $\mathbb{F}_{p^4}$. Therefore, it is not a trivial task to calculate the class group action on Edwards curves with w-coordinates over only $\mathbb{F}_p$.
In this paper, we prove some theorems about the properties of Edwards curves. By these theorems, we construct the new CSIDH algorithm on Edwards curves with w-coordinates over $\mathbb{F}_p$. This algorithm is as fast as (or a little bit faster than) the algorithm proposed by Meyer and Reith. This paper is an extend version of [25]. We added the construction of a technique similar to Elligator on Edwards curves. This technique contributes the efficiency of the constant-time CSIDH algorithm. We also added the construction of new formulas to compute isogenies in $\tilde{O}(\sqrt{\ell})$ times on Edwards curves. It is based on formulas on Montgomery curves proposed by Bernstein *et al.* ($\sqrt{}$élu's formulas). In our analysis, these formulas on Edwards curves is a little bit faster than those on Montgomery curves.

Keywords: Isogeny-based cryptography · Montgomery curves · Edwards curves · CSIDH · Post-quantum cryptography.

1 Introduction

This paper is an extend version of [25]. The first additional content is the construction of Elligator [6] on Edwards curves. Using Elligator makes the constant-time CSIDH algorithm faster. The second additional content is the construction of $\sqrt{}$élu's formulas on Edwards curves. In both results, our proposal is as fast as (or a little faster than) those on Montgomery curves.

Currently, there are two popular public-key cryptosystems: RSA [28], whose security is based on the computational complexity of the Prime Factorization Problem, and Elliptic Curve Cryptography [22, 19], whose security is based on

Table 1. Comparing CSIDH algorithms on Montgomery curves and Edwards curves

	group operations	calculation of isogenies	kernel points
Montgomery	✓	✓	✓
Edwards (y-coordinate)	✓	✓	✓
Edwards (w-coordinate)	✓	✓	not trivial

the computational complexity of the Discrete Logarithm Problem. However, Shor pointed out in 1994 that both the Prime Factorization Problem and the Discrete Logarithm Problem can be solved in polynomial time by using a quantum computer [29, 30]. This means we should develop new cryptosystems which cannot be broken by quantum computers. We call such cryptosytems post quantum cryptography (PQC).

Isogeny-based cryptography is a branch of public-key cryptography based on the computational complexity of the Isogeny Problem, which is a problem arising when we calculate isogenies between given two elliptic curves. It is considered to be a candidate of PQC. Jao and De Feo proposed a Diffie-Hellman type isogeny-based key exchange protocol, called SIDH (Supersingular Isogeny Diffie-Hellman), in 2011 [17]. SIKE (Supersingular Isogeny Key Encapsulation) [2], which is derived from SIDH, is a round 3 alternate candidate in the NIST PQC standardization competition [26]. The SIDH calculation uses supersingular elliptic curves over $\mathbb{F}_{p^2}$. Castryck, Lange, Martindale, Panny, and Renes proposed another Diffie-Hellman type of isogeny-based key exchange protocol, called CSIDH (Commutative Supersingular Isogeny Diffie-Hellman), in 2018 [8]. Its calculation uses supersingular elliptic curves over $\mathbb{F}_p$.

CSIDH is based on a commutative group action on $\mathbb{F}_p$-isomorphism classes of supersingular Montgomery curves defined over $\mathbb{F}_p$. In order to calculate this group action, we need to generate a point in $\ker(\pi_p - 1)$ or in $\ker(\pi_p + 1)$ and determine which set the point belongs to, where π_p is the p-Frobenius map. Castryck, Lange, Martindale, Panny, and Renes showed that if we take a random element from $\mathbb{F}_p$ as an x-coordinate of a point in a Montgomery curve and determine whether y-coordinate of the point belongs to $\mathbb{F}_p$ or not, then we can get a point in $\ker(\pi_p - 1)$ or in $\ker(\pi_p + 1)$ and determine which set the point belongs to [8]. They also showed that a Montgomery coefficient is unique up to $\mathbb{F}_p$-isomorphism [8]. Since it is known that a group operation of a Montgomery curve can be calculated using only the x-coordinates of the points [23] and that isogenies between Montgomery curves can be also calculated by using only the x-coordinates of the points of the kernel [10, 21], we can compute a CSIDH group action using only $\mathbb{F}_p$-arithmetic.

Meyer and Reith proposed a faster CSIDH algorithm in 2018 [21]. This algorithm calculates isogenies over Edwards curves instead of Montgomery curves, by using a birational map between a Montgomery curve and an Edwards curve. In this algorithm, the method for generating a point in $\ker(\pi_p - 1)$ or in $\ker(\pi_p + 1)$ and determining which set the point belongs to is the same as in the original

CSIDH algorithm [8]. Hence, a question arises: How do we calculate the CSIDH algorithm on *purely* Edwards curves over $\mathbb{F}_p$?

There are two special coordinates (the y-coordinate and the w-coordinate) on Edwards curves for efficiently calculating the group operation [9, 14] and isogenies [24, 9, 18] respectively. For a point P in an Edwards curve, if the y-coordinate of P is in $\mathbb{F}_p$, then P always belongs to $\ker(\pi_p - 1)$ or $\ker(\pi_p + 1)$. Therefore, it is not difficult to construct the CSIDH algorithm on Edwards curves with y-coordinates. We detail this algorithm in Appendix C. However, if we take a random element from $\mathbb{F}_p$ as the w-coordinate of a point on an Edwards curve, the point is sometimes defined outside of $\mathbb{F}_{p^2}$ (defined over $\mathbb{F}_{p^4}$). Since the points in $\ker(\pi_p - 1)$ and those in $\ker(\pi_p + 1)$ are defined over $\mathbb{F}_{p^2}$, it is not a trivial task to run the CSIDH algorithm using only Edwards curves over $\mathbb{F}_p$ with w-coordinates. We summarize the above discussion in Table 1.

The computational costs of the CSIDH group action depend on its secret key. Therefore, CSIDH is vulnerable to side-channel attacks. There are some proposals for constant-time CSIDH algorithms [20, 27, 9]. They use a special map named *Elligator* [6]. Elligator makes these algorithms more efficient. Elligator can be used for x-coordinates of Montgomery curves; however, there are no techniques similar to Elligator for w-coordinates of Edwards curves.

In 2020, Bernstein, De Feo, Leroux, and Smith proposed the new formulas for computing ℓ-isogenies in $\tilde{O}(\sqrt{\ell})$ times [5]. Moreover, Adj, Chi-Domínguez, and Rodríguez-Henríquez improved these formulas in [1]. We call these formulas $\sqrt{\text{é}}$lu's formulas. Bernstein et al. showed that by using these formulas, the CSIDH algorithm gets more efficient. These formulas are constructed by using the x-coordinates of Montgomery curves. There is no result about $\sqrt{\text{é}}$lu's formulas on Edwards curves.

1.1 Our results

In this paper, we prove four important theorems about the w-coordinate on Edwards curves and use them to construct a new implementation of the CSIDH key exchange. First, we show that if we take a random element from the set of square elements in $\mathbb{F}_p$ as the w-coordinate of a point P and determine whether the w-coordinate of $2P$ is square in $\mathbb{F}_p$ or not, then we can generate a point in $\ker(\pi_p - 1)$ or in $\ker(\pi_p + 1)$ and determine which set the point belongs to. Specifically, if the w-coordinate of $2P$ is square, then this coordinate represents a point in $\ker(\pi_p + 1)$, and if the w-coordinate of $2P$ is not square, then the reciprocal of this coordinate represents a point in $\ker(\pi_p - 1)$. Second, we show that there is no difference between the probability of generating a point in $\ker(\pi_p - 1)$ and the probability of generating a point in $\ker(\pi_p + 1)$ in the previous way. Third, we prove the probability that we get a point of order ℓ_i is $1 - 1/\ell_i$, like Montgomery curves. Finally, we show that an Edwards coefficient is unique up to an $\mathbb{F}_p$-isomorphism, like a Montgomery coefficient.

From these theorems, we construct a non-trivial new implementation of the CSIDH key exchange that uses w-coordinates on Edwards curves non-trivially (Algorithm 3). We show that our algorithm is as fast as (or a little bit faster than) the algorithm proposed by Meyer and Reith [21]. This fact can be also confirmed from data obtained by the implementation (Table 2).

Moreover, we realize the technique similar to Elligator on Edwards curves with using w-coordinates. This technique is more efficient than that on Montgomery curves. Therefore, the constant-time CSIDH algorithm on Edwards curves is as fast as (or a little bit faster than) that on Montgomery curves.

Furthermore, we propose the new $\sqrt{\text{é}}$lu's formulas on Edwards curves. These formulas are constructed by the w-coordinates on Edwards curves. In our analysis, those on Edwards curves are a little bit faster than those on Montgomery curves.

2 Preliminaries

2.1 Basic mathematical concepts

Here, we explain basic mathematical concepts behind isogeny-based cryptography.

Let $\mathbb{L}$ be a field, and $\mathbb{L}'$ be an algebraic extension field of $\mathbb{L}$. An elliptic curve E defined over $\mathbb{L}$ is a non-singular algebraic curve defined over $\mathbb{L}$ of genus one. Denote by $E(\mathbb{L}')$ the $\mathbb{L}'$-rational points of the elliptic curve E. $E(\mathbb{L}')$ is an abelian group [31, Ⅲ. 2]. A supersingular elliptic curve E over a finite field $\mathbb{L}$ of characteristic p is defined as an elliptic curve which satisfies $\#E(\mathbb{L}) \equiv 1 \pmod p$, where $\#E(\mathbb{L})$ is the cardinality of $E(\mathbb{L})$.

Let E, E' be elliptic curves defined over $\mathbb{L}$. Define an isogeny $\phi\colon E \to E'$ over $\mathbb{L}'$ to be a rational map over $\mathbb{L}'$ which is a non-zero group homomorphism from $E(\overline{\mathbb{L}})$ to $E'(\overline{\mathbb{L}})$, where $\overline{\mathbb{L}}$ is the algebraic closure of $\mathbb{L}$. A separable isogeny with $\#\ker\phi = \ell$ is called an ℓ-isogeny. Denote by $\mathrm{End}_{\mathbb{L}'}(E)$ the endomorphism ring of E over $\mathbb{L}'$. It is represented as $\mathrm{End}_p(E)$ when $\mathbb{L}'$ is a prime field $\mathbb{F}_p$. An isogeny $\phi\colon E \to E'$ defined over $\mathbb{L}'$ is called an isomorphism over $\mathbb{L}'$ if ϕ has an inverse isogeny over $\mathbb{L}'$.

If G is a finite subgroup of $E(\overline{\mathbb{L}})$, then there exists an isogeny $\phi\colon E \to E'$ whose kernel is G, and E' is unique up to an $\overline{\mathbb{L}}$-isomorphism [31, Proposition Ⅱ I.4.12]. This isogeny can be efficiently calculated by using Vélu's formulas [32]. We denote a representative of E' by E/G.

$E[k]$ ($k \in \mathbb{Z}_{>0}$) is defined as the k-torsion subgroup of $E(\overline{\mathbb{L}})$. For an endomorphism ϕ of E, we sometimes denote $\ker\phi$ by $E[\phi]$.

Let $\mathbb{L}$ be a number field, and $\mathcal{O}$ be an order in $\mathbb{L}$. A fractional ideal $\mathfrak{a}$ of $\mathcal{O}$ is a finitely generated $\mathcal{O}$-submodule of $\mathbb{L}$ which satisfies $\alpha\mathfrak{a} \subset \mathcal{O}$ for some $\alpha \in \mathcal{O} \setminus \{0\}$. An invertible fractional ideal $\mathfrak{a}$ of $\mathcal{O}$ is defined as a fractional ideal of $\mathcal{O}$ which satisfies $\mathfrak{a}\mathfrak{b} = \mathcal{O}$ for some fractional ideal $\mathfrak{b}$ of $\mathcal{O}$. The fractional ideal $\mathfrak{b}$ is represented as $\mathfrak{a}^{-1}$. If a fractional ideal $\mathfrak{a}$ is contained in $\mathcal{O}$, then $\mathfrak{a}$ is called an integral ideal of $\mathcal{O}$.

Let $I(\mathcal{O})$ be a set of invertible fractional ideals of $\mathcal{O}$. $I(\mathcal{O})$ is an abelian group derived from multiplication of ideals with the identity $\mathcal{O}$. Let $P(\mathcal{O})$ be a subgroup of $I(\mathcal{O})$ defined by $P(\mathcal{O}) = \{\mathfrak{a} \mid \mathfrak{a} = \alpha\mathcal{O} \text{ (for some } \alpha \in \mathbb{L}^{\times})\}$. We call the abelian group $\mathrm{cl}(\mathcal{O})$ defined by $I(\mathcal{O})/P(\mathcal{O})$ the ideal class group of $\mathcal{O}$.

The $\mathbb{F}_p$-endomorphism ring $\mathrm{End}_p(E)$ of a supersingular elliptic curve E defined over $\mathbb{F}_p$ is isomorphic to an order in an imaginary quadratic field [12]. Denote by $\mathcal{E}\ell\ell_p(\mathcal{O})$ the set of $\mathbb{F}_p$-isomorphism classes of elliptic curves E whose $\mathbb{F}_p$-endomorphism ring $\mathrm{End}_p(E)$ is isomorphic to $\mathcal{O}$.

2.2 Montgomery curves

Let $\mathbb{L}$ be a field whose characteristic is odd. An elliptic curve E defined by the following equation is called a Montgomery curve:

$$E\colon bY^2Z = X^3 + aX^2Z + XZ^2 \quad (a, b \in \mathbb{L} \text{ and } b(a^2-4) \neq 0).$$

In this paper, we denote the Montgomery curve $Y^2Z = X^3 + aX^2Z + XZ^2$ by $E_{\mathcal{M},a}$. The identity of E is $(0 : 1 : 0)$, and the inverse of $(X : Y : Z)$ is $(X : -Y : Z)$.

Montgomery showed that the group operations on Montgomery curves can be efficiently computed by using x-coordinates [23]. Define a function x as

$$x(X : Y : Z) = \frac{X}{Z}.$$

The function x is not defined at the point $(0 : 1 : 0)$. If P and Q satisfy $x(P) = x(Q)$, then $P = Q$ or $P = -Q$. Next define a function $\mathbf{x}$ as $\mathbf{x}(X : Y : Z) = (X : Z)$. We call $\mathbf{x}(P)$ the projective x-coordinates of P.

Let P be a point on E. Let $A/C = a$ and $B/C = b$. Let $(X : Z) = \mathbf{x}(P)$. The projective x-coordinates $(X' : Z')$ of $2P$ are calculated as follows [23]:

$$X' = 4C(X+Z)^2(X-Z)^2, \quad Z' = 4XZ(4C(X-Z)^2 + (A+2C)4XZ). \quad (1)$$

The computational cost is $4\mathbf{M} + 2\mathbf{S} + 4\mathbf{a}$. If $Z = 1$, the computational cost is $4\mathbf{M} + 1\mathbf{S} + 5\mathbf{a}$. (We denote field multiplications by $\mathbf{M}$, field squarings by $\mathbf{S}$, and field additions, subtractions, or doublings by $\mathbf{a}$.)

Let P_1 and P_2 be points on E, and $(X_1 : Z_1) = \mathbf{x}(P_1)$, $(X_2 : Z_2) = \mathbf{x}(P_2)$. Let $(X_0 : Z_0) = \mathbf{x}(P_1 - P_2)$. The projective x-coordinates $(X_3 : Z_3)$ of $P_1 + P_2$ are calculated as follows [23]:

$$X_3 = Z_0(X_1X_2 - Z_1Z_2)^2, \quad Z_3 = X_0(X_1Z_2 - X_2Z_1)^2. \quad (2)$$

The computational cost is $4\mathbf{M} + 2\mathbf{S} + 6\mathbf{a}$. If $Z_0 = 1$, the computational cost is $3\mathbf{M} + 2\mathbf{S} + 6\mathbf{a}$.

Costello and Hisil proposed efficient calculations for odd-degree isogenies by using x-coordinates [10], and Meyer and Reith improved them [21]. Let ℓ be an odd integer and s be the integer which satisfies that $\ell = 2s + 1$. Let P be a point on E, and $(X : Z) = \mathbf{x}(P)$. Let Q be an order-ℓ point on E, and $(X_1 : Z_1) = \mathbf{x}(Q)$. Let $(X_k : Z_k) = \mathbf{x}(kQ)$. Let $E' = E/\langle Q\rangle$ and ϕ be an isogeny $\phi\colon E \to E'$ with $\ker\phi = \langle Q\rangle$. The projective x-coordinates $(X' : Z')$ of $\phi(P)$ are calculated as follows [10]:

$$X' = X \cdot \prod_{i=1}^{s}(XX_i - ZZ_i)^2, \quad Z' = Z \cdot \prod_{i=1}^{s}(XZ_i - ZX_i)^2. \quad (3)$$

The computational cost is $(4s)\mathbf{M} + 2\mathbf{S} + (4s+2)\mathbf{a}$. Let $A/C = a$. The curve coefficient $a' = A'/C'$ of E' is calculated as follows [21]:

$$\begin{aligned} \tilde{a} = A + 2C, \quad \tilde{d} = A - 2C, \quad \tilde{a}' = \tilde{a}^{\ell} \cdot \prod_{i=1}^{s} (X_i + Z_i)^8, \\ \tilde{d}' = \tilde{d}^{\ell} \cdot \prod_{i=1}^{s} (X_i - Z_i)^8, \quad A' = 2(\tilde{a}' + \tilde{d}'), \quad C' = \tilde{a}' - \tilde{d}'. \end{aligned} \tag{4}$$

The computational cost is $(2s+2)\mathbf{M} + 6\mathbf{S} + (2s+6)\mathbf{a}$ and that of the two s-th powers. Since $X_i + Z_i$ and $X_i - Z_i$ are also used for calculating $\phi(P)$, the computational cost of calculating $\phi(P)$ and E' is $(6s+2)\mathbf{M} + 8\mathbf{S} + (4s+8)\mathbf{a}$ and that of the two s-th powers. Appendix A. 1 describes why the computational costs are as above.

Furthermore, for a high-degree isogeny ϕ, there are more efficient methods to compute $\phi(P)$ and (A', C') on Montgomery curves [5, 1]. These formulas can be computed in $\tilde{O}(\sqrt{\ell})$ times. We call them $\sqrt{\text{é}}$lu's formulas. The equations (3) and (4) can be rewritten as follows:

$$X' = X \cdot (h_S(Z, X))^2, \quad Z' = Z \cdot (h_S(X, Z))^2. \tag{5}$$

$$\begin{aligned} \tilde{a} = A + 2C, \quad \tilde{d} = A - 2C, \quad \tilde{a}' = \tilde{a}^{\ell} \cdot (h_S(1,1))^8, \\ \tilde{d}' = \tilde{d}^{\ell} \cdot (h_S(-1,1))^8, \quad A' = 2(\tilde{a}' + \tilde{d}'), \quad C' = \tilde{a}' - \tilde{d}'. \end{aligned} \tag{6}$$

Here, h_S is a polynomial in $\mathbb{F}_p[T_1, T_2]$ defined as $h_S(T_1, T_2) := \prod_{i \in S}(Z_i T_1 - X_i T_2)$, and S is a set $\{1, 3, \ldots, \ell - 2\}$. By using resultants, we can compute $h_S(\alpha, \beta)$ for some α, β. Appendix A. 3 describes more details about the above method.

2.3 Edwards curves

In 2007, Edwards introduced a new form of an elliptic curve [13]. Bernstein and Lange extended these curves to another form in 2007, called Edwards curves [7]. For representing points at infinity, Hisil, Wong, Carter, and Dawson proposed projective closures of Edwards curves in $\mathbb{P}^3$ in 2018 [16].

Let $\mathbb{L}$ be a field. If an elliptic curve E is defined by the following equations, E is called an Edwards curve [16]:

$$E\colon X^2 + Y^2 = Z^2 + dT^2, \quad XY = ZT \quad (d \in \mathbb{L} \text{ and } d \neq 0, 1).$$

In this paper, we denote the Edwards curve $X^2 + Y^2 = Z^2 + dT^2,\ XY = ZT$ by E_d. The identity of E_d is $(0:1:1:0)$, which we will denote by 0_d for simplicity, while the inverse of $(X:Y:Z:T)$ is $(-X:Y:Z:-T)$. We obtain the group addition formulas as follows [16]:

$$\begin{aligned} &(X_1 : Y_1 : Z_1 : T_1) + (X_2 : Y_2 : Z_2 : T_2) \\ &= ((X_1Y_2 + Y_1X_2)(Z_1Z_2 - dT_1T_2) : (Y_1Y_2 - X_1X_2)(Z_1Z_2 + dT_1T_2) \\ &\quad : (Z_1Z_2 - dT_1T_2)(Z_1Z_2 + dT_1T_2) : (Y_1Y_2 - X_1X_2)(X_1Y_2 + Y_1X_2)). \end{aligned} \tag{7}$$

For simplicity, we will sometimes consider an Edwards curve to be an affine curve defined by the following equation:

$$E\colon x^2 + y^2 = 1 + dx^2y^2 \quad (d \in \mathbb{L} \text{ and } d \neq 0, 1),$$

where $x = X/Z$ and $y = Y/Z$. In this equation, only $(\pm\sqrt{d} : 0 : 0 : 1)$ and $(0 : \pm\sqrt{d} : 0 : 1)$ are points at infinity. $(\pm\sqrt{d} : 0 : 0 : 1)$ are points of order 2, and $(0 : \pm\sqrt{d} : 0 : 1)$ are points of order 4. Hence, if the order of a point P on E_d is neither 2 nor 4, P can be represented in affine coordinates (x, y).

In [24, 9] it was showed that the group calculations of Edwards curves can be efficiently performed by using the y-coordinate. Define a function y as

$$y(X : Y : Z : T) = \begin{cases} \frac{Y}{Z} & (\text{if } Z \neq 0) \\ \infty & (\text{if } Z = 0 \text{ (points at infinity)}) \end{cases}.$$

We call $y(P)$ the y-coordinate of P. If P and Q satisfy that $y(P) = y(Q)$, then $P = Q$ or $P = -Q$. Define a function $\mathbf{y}$ as $\mathbf{y}(X : Y : Z : T) = (Y : Z)$. We call $\mathbf{y}(P)$ the projective y-coordinates of P.

Let P be a point on E_d, and $(Y : Z) = \mathbf{y}(P)$. Let $D/C = d$. The projective y-coordinates $(Y' : Z')$ of $2P$ are calculated as follows [9]:

$$\begin{aligned} Y' &= (C - D)Y^2Z^2 - (Z^2 - Y^2) \cdot ((C - D)Y^2 + C(Z^2 - Y^2)), \\ Z' &= (C - D)Y^2Z^2 + (Z^2 - Y^2) \cdot ((C - D)Y^2 + C(Z^2 - Y^2)). \end{aligned} \tag{8}$$

The computational cost is $4\mathbf{M} + 2\mathbf{S} + 5\mathbf{a}$. If $Z = 1$, the computational cost is $3\mathbf{M} + 1\mathbf{S} + 5\mathbf{a}$.

Let P_1 and P_2 be points on E_d, and $(Y_1 : Z_1) = \mathbf{y}(P_1)$, $(Y_2 : Z_2) = \mathbf{y}(P_2)$. Let $(Y_0 : Z_0) = \mathbf{y}(P_1 - P_2)$. The projective y-coordinates $(Y_3 : Z_3)$ of $P_1 + P_2$ are calculated as follows [9]:

$$\begin{aligned} Y_3 &= (Z_0 - Y_0)(Y_1Z_2 + Y_2Z_1)^2 - (Z_0 + Y_0)(Y_1Z_2 - Y_2Z_1)^2, \\ Z_3 &= (Z_0 - Y_0)(Y_1Z_2 + Y_2Z_1)^2 + (Z_0 + Y_0)(Y_1Z_2 - Y_2Z_1)^2. \end{aligned} \tag{9}$$

The computational cost is $4\mathbf{M} + 2\mathbf{S} + 6\mathbf{a}$. In the case that $Z_0 = 1$, the computational cost is also $4\mathbf{M} + 2\mathbf{S} + 6\mathbf{a}$.

In [9] efficient calculations were proposed for odd-degree isogenies by using projective y-coordinates. Let ℓ be an odd integer and s be the integer which satisfies $\ell = 2s + 1$. Let P be a point on E_d, and $(Y : Z) = \mathbf{y}(P)$. Let Q be an order-ℓ point on E_d, and $(Y_1 : Z_1) = \mathbf{y}(Q)$. Let $(Y_k : Z_k) = \mathbf{y}(kQ)$. Let $E_{d'} = E_d/\langle Q \rangle$, and ϕ be an isogeny $\phi\colon E_d \to E_{d'}$ with $\ker\phi = \langle Q \rangle$. The projective y-coordinates $(Y' : Z')$ of $\phi(P)$ are calculated as follows [9]:

$$\begin{aligned} Y' =& (Z + Y) \cdot \prod_{i=1}^{s} (ZY_i + Z_iY)^2 - (Z - Y) \cdot \prod_{i=1}^{s} (ZY_i - Z_iY)^2, \\ Z' =& (Z + Y) \cdot \prod_{i=1}^{s} (ZY_i + Z_iY)^2 + (Z - Y) \cdot \prod_{i=1}^{s} (ZY_i - Z_iY)^2. \end{aligned} \tag{10}$$

The computational cost is $(4s)\mathbf{M}+2\mathbf{S}+(2s+4)\mathbf{a}$. The projective curve coefficient $d' = D'/C'$ is calculated as follows [24]:

$$D' = D^{\ell} \cdot \prod_{i=1}^{s} (Y_i)^8, \quad C' = C^{\ell} \cdot \prod_{i=1}^{s} (Z_i)^8. \tag{11}$$

The computational cost is $(2s+2)\mathbf{M}+6\mathbf{S}$ and that of the two s-th powers. The computational cost of calculating $\phi(P)$ and $E_{d'}$ is $(6s+2)\mathbf{M}+8\mathbf{S}+(2s+4)\mathbf{a}$ and that of the two s-th powers.

Farashahi and Hosseini showed that the group calculations of Edwards curves can be efficiently performed by using the w-coordinate [14]. Define a function w as

$$w(X:Y:Z:T) = \begin{cases} \frac{dT^2}{Z^2} & (\text{if } Z \neq 0) \\ \infty & (\text{if } Z = 0 \text{ (points at infinity)}) \end{cases}.$$

In affine coordinates, $w(x,y) = dx^2y^2$. We call $w(P)$ the w-coordinate of P. If P and Q satisfy that $w(P) = w(Q)$, then $P+Q$ or $P-Q$ is an element of

$$\{0_d, (0:-1:1:0), (1:0:1:0), (-1:0:1:0)\}.$$

In this paper, we will denote $\{0_d, (0:-1:1:0), (1:0:1:0), (-1:0:1:0)\}$ by $\mathcal{G}_4$ for simplicity. Note that $\mathcal{G}_4$ is a cyclic group of order 4. Define a function $\mathbf{w}$ as $\mathbf{w}(X:Y:Z:T) = (dT^2 : Z^2)$. We call $\mathbf{w}(P)$ the projective w-coordinates of P.

Let P be a point on E_d, and $(W:Z) = \mathbf{w}(P)$. Let $D/C = d$. The projective w-coordinates $(W':Z')$ of $2P$ are calculated as follows [14]:

$$W' = 4WZ(D(W+Z)^2 - 4CWZ), \quad Z' = D(W+Z)^2(W-Z)^2. \tag{12}$$

The computational cost is $4\mathbf{M}+2\mathbf{S}+4\mathbf{a}$. If $Z = 1$, the computational cost is $4\mathbf{M}+1\mathbf{S}+5\mathbf{a}$.

Let P_1 and P_2 be points on E_d, and $(W_1:Z_1) = \mathbf{w}(P_1)$, $(W_2:Z_2) = \mathbf{w}(P_2)$. Let $(W_0:Z_0) = \mathbf{w}(P_1 - P_2)$. The projective w-coordinates $(W_3:Z_3)$ of P_1+P_2 are calculated as follows [14]:

$$W_3 = Z_0(W_1Z_2 - W_2Z_1)^2, \quad Z_3 = W_0(W_1W_2 - Z_1Z_2)^2. \tag{13}$$

The computational cost is $4\mathbf{M}+2\mathbf{S}+6\mathbf{a}$. If $Z_0 = 1$, the computational cost is $3\mathbf{M}+2\mathbf{S}+6\mathbf{a}$.

Kim, Yoon, Park, and Hong proposed efficient calculations for odd-degree isogenies by using projective w-coordinates [18]. Let ℓ be an odd integer and s be the integer which satisfies $\ell = 2s+1$. Let P be a point on E_d, and $(W:Z) = \mathbf{w}(P)$. Let Q be an order-ℓ point on E_d, and $(W_1:Z_1) = \mathbf{w}(Q)$. Let $(W_k:Z_k) = \mathbf{w}(kQ)$. Let $E_{d'} = E_d/\langle Q\rangle$, and ϕ be an isogeny $\phi\colon E_d \to E_{d'}$ with $\ker\phi = \langle Q\rangle$. The projective w-coordinates $(W':Z')$ of $\phi(P)$ are calculated as follows [18]:

$$W' = W \cdot \prod_{i=1}^{s} (ZW_i - Z_iW)^2, \quad Z' = Z \cdot \prod_{i=1}^{s} (WW_i - ZZ_i)^2. \tag{14}$$

The computational cost is $(4s)\mathbf{M}+2\mathbf{S}+(4s+2)\mathbf{a}$. The projective curve coefficient $d' = D'/C'$ is calculated as follows [18]:

$$D' = D^{\ell} \cdot \prod_{i=1}^{s}(W_i + Z_i)^8, \quad C' = C^{\ell} \cdot \prod_{i=1}^{s}(2Z_i)^8. \tag{15}$$

The computational cost is $(2s+2)\mathbf{M} + 6\mathbf{S} + (s+4)\mathbf{a}$ and that of the two s-th powers. Since $W_i + Z_i$ is also used for calculating $\phi(P)$, the computational cost of calculating $\phi(P)$ and $E_{d'}$ is $(6s+2)\mathbf{M} + 8\mathbf{S} + (4s+6)\mathbf{a}$ and that of the two s-th powers. Appendix A. 2 describes why the computational costs are as above.

Furthermore, we found the method to compute ℓ-isogenies on Edwards curves in $\tilde{O}(\sqrt{\ell})$ times as on Montgomery curves. We explain these formulas in section 9.

An Edwards curve has a following property.

Theorem 1. *Let p be a prime and $p \geq 3$. The Edwards curve E_d defined over $\mathbb{F}_p$ is $\mathbb{F}_p$-isomorphic to the Montgomery curve,*

$$E_{\mathcal{M}} \colon \frac{4}{1-d}Y^2Z = X^3 + \frac{2(1+d)}{1-d}X^2Z + XZ^2.$$

Proof. Bernstein, Birkner, Joye, Lange, and Peters show that there is a birational map between E_d and $E_{\mathcal{M}}$ [4]. This birational map becomes an isomorphism.

The proof of this theorem is given in Appendix Appendix B. □

It is known that there is a birational map between a Montgomery curve and an Edwards curve [4]. However, we need an isomorphism for constructing the CSIDH algorithm using only Edwards curves.

Corollary 1. *Let p be a prime, $p \geq 3$, and $p \equiv 3 \pmod 4$. An Edwards curve E_d defined over $\mathbb{F}_p$ is $\mathbb{F}_p$-isomorphic to the Montgomery curve,*

$$E_{\mathcal{M}} \colon Y^2Z = X^3 + \chi(1-d) \cdot \frac{2(1+d)}{1-d}X^2Z + XZ^2,$$

where the map $\chi \colon \mathbb{F}_p \to \mathbb{F}_p$ is defined as $\chi(a) := a^{(p-1)/2}$.

Corollary 1 is easily proven from Theorem 1.

Corollary 2. *Let p be a prime, $p \geq 3$, and $p \equiv 3 \pmod 8$. Let $E_{\mathcal{M},a}$ be a supersingular Montgomery curve $Y^2Z = X^3 + aX^2Z + XZ^2$ defined over $\mathbb{F}_p$. If $a-2$ is square, then $E_{\mathcal{M},a}$ is $\mathbb{F}_p$-isomorphic to the Edwards curve,*

$$E_{\frac{a+2}{a-2}} \colon X^2 + Y^2 = Z^2 + \frac{a+2}{a-2}T^2, \quad XY = ZT,$$

and if $a-2$ is not square, then $E_{\mathcal{M},a}$ is $\mathbb{F}_p$-isomorphic to the Edwards curve,

$$E_{\frac{a-2}{a+2}} \colon X^2 + Y^2 = Z^2 + \frac{a-2}{a+2}T^2, \quad XY = ZT.$$

Proof. As $E_{\mathcal{M},a}$ is supersingular, $\#E_{\mathcal{M},a}(\mathbb{F}_p) = \#\tilde{E}_{\mathcal{M},a}(\mathbb{F}_p) = p+1 \equiv 4 \pmod 8$, where $\tilde{E}_{\mathcal{M},a}$ is a quadratic twist of $E_{\mathcal{M}}$. From Table 1 of [11], $(a-2)(a+2)$ is not square.

If $a-2$ is square, the Edwards curve $E_{\frac{a+2}{a-2}}$ is $\mathbb{F}_p$-isomorphic to $E_{\mathcal{M},a}$ by Corollary 1. If $a-2$ is not square, since $a+2$ is square, the Edwards curve $E_{\frac{a-2}{a+2}}$ is $\mathbb{F}_p$-isomorphic to $E_{\mathcal{M},a}$ by Corollary 1.

This completes the proof of Corollary 2. □

By using Corollary 1 and Corollary 2, it is easy to convert an Edwards curve into a Montgomery curve and convert a Montgomery curve into an Edwards curve.

3 CSIDH [8]

CSIDH (Commutative Supersingular Isogeny Diffie-Hellman) was proposed by Castryck, Lange, Martindale, Panny, and Renes in 2018 [8].

CSIDH is based on the action of $\mathrm{cl}(\mathbb{Z}[\pi_p])$ on $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi_p])$. Let the prime p be $4 \cdot \ell_1 \cdots \ell_n - 1$, where the $\ell_1, \ldots, \ell_n$ are small distinct odd primes, for Alice and Bob to calculate the action efficiently. Alice and Bob let random elements of $\mathrm{cl}(\mathbb{Z}[\pi_p])$ be secret keys and calculate the actions on $E_{\mathcal{M},0}\colon Y^2Z = X^3 + XZ^2$. They publish the obtained elliptic curves as public keys. Finally, they calculate the actions on the public keys, respectively. The obtained elliptic curves are identical up to $\mathbb{F}_p$-isomorphism by the commutativity of $\mathrm{cl}(\mathbb{Z}[\pi_p])$; therefore, the values of the Montgomery coefficients are the same from Theorem 3. Let their values be $\mathrm{SK}_{\mathrm{shared}}$.

3.1 CSIDH protocol

Before explaining the protocol of CSIDH, we should state the following important theorems.

Theorem 2 ([33, Theorem 4.5]). *Let $\mathcal{O}$ be an order of an imaginary quadratic field and E be an elliptic curve defined over $\mathbb{F}_p$. If $\mathcal{E}\ell\ell_p(\mathcal{O})$ contains the $\mathbb{F}_p$-isomorphism class of supersingular elliptic curves, then the action of the ideal class group $\mathrm{cl}(\mathcal{O})$ on $\mathcal{E}\ell\ell_p(\mathcal{O})$,*

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

is free and transitive, where $\mathfrak{a}$ is an integral ideal of $\mathcal{O}$, and $E[\mathfrak{a}]$ is the intersection of the kernels of elements in the ideal $\mathfrak{a}$.

Denote a representative of $E/E[\mathfrak{a}]$ by $[\mathfrak{a}]E$.

Theorem 3 ([8, Proposition 8]). *Let p be a prime satisfying $p \equiv 3 \pmod 8$. Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$. Then, $\mathrm{End}_p(E) = \mathbb{Z}[\pi_p]$ holds if and only if there uniquely exists $a \in \mathbb{F}_p$ such that E is $\mathbb{F}_p$-isomorphic to a Montgomery curve $E_{\mathcal{M},a}$, where π_p is the p-Frobenius map.*

The exact protocol is as follows. Suppose that Alice and Bob want to share a secret key denoted by $\mathrm{SK}_{\mathrm{shared}}$.

Setup. Let p be a prime which satisfies $p = 4 \cdot \ell_1 \cdots \ell_n - 1$, where $\ell_1, \ldots, \ell_n$ are small distinct odd primes. Let the public parameters be p and $E_{\mathcal{M},0}$.

Key generation. One randomly chooses a integer vector $(e_1, \ldots, e_n)$ from $\{-m, \ldots, m\}^n$. Define $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}] \in \mathrm{cl}(\mathbb{Z}[\pi_p])$, where $\mathfrak{l}_i = (\ell_i, \pi_p - 1)$, $\mathfrak{l}_i^{-1} = (\ell_i, \pi_p + 1)$, and m is the smallest integer which satisfies $2m + 1 \geq \sqrt[n]{\#\mathrm{cl}(\mathbb{Z}[\pi_p])} \approx p^{1/2n}$. One calculates the action of $[\mathfrak{a}]$ on $E_{\mathcal{M},0}$ and the Montgomery coefficient $a \in \mathbb{F}_p$ of $[\mathfrak{a}]E_{\mathcal{M},0} \colon Y^2Z = X^3 + aX^2Z + XZ^2$.
Let the integer vector $(e_1, \ldots, e_n)$ be the secret key, and $a \in \mathbb{F}_p$ be the public key.

Key exchange. Alice and Bob have pairs of keys, $([\mathfrak{a}], a)$ and $([\mathfrak{b}], b)$, respectively. Alice calculates the action $[\mathfrak{a}]E_{\mathcal{M},b} = [\mathfrak{a}][\mathfrak{b}]E_{\mathcal{M},0}$. Bob calculates the action $[\mathfrak{b}]E_{\mathcal{M},a} = [\mathfrak{b}][\mathfrak{a}]E_{\mathcal{M},0}$. Denote the Montgomery coefficient of $[\mathfrak{a}][\mathfrak{b}]E_{\mathcal{M},0}$ by SK_{Alice} and the Montgomery coefficient of $[\mathfrak{b}][\mathfrak{a}]E_{\mathcal{M},0}$ by SK_{Bob}.

From the commutativity of $\mathrm{cl}(\mathbb{Z}[\pi_p])$ and Theorem 3, $\mathrm{SK}_{Alice} = \mathrm{SK}_{Bob}$ holds. Let these values be the shared key $\mathrm{SK}_{\mathrm{shared}}$.

3.2 Evaluating the class group action on Montgomery curves

In this subsection, we explain how to evaluate the class group action on Montgomery curves [8]. Algorithm 1 is an algorithm for evaluating the class group action.

Let p be a prime satisfying $p = 4 \cdot \ell_1 \cdots \ell_n - 1$, where $\ell_1, \ldots, \ell_n$ are small distinct odd primes. The inputs of the algorithm are a Montgomery coefficient $a \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$. The output is a Montgomery coefficient $a' \in \mathbb{F}_p$ that satisfies $E_{\mathcal{M},a'} = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_{\mathcal{M},a}$.

We calculate a' by repeating the calculations of the actions of $[\mathfrak{l}_i]$ or $[\mathfrak{l}_i]^{-1}$ (i.e., repeating the calculations of ℓ_i-isogenies).

Sampling points (line 2-8 in Algorithm 1) For calculating the class group action, we first sample a point which belongs to $\ker(\pi_p - 1)$ or $\ker(\pi_p + 1)$. We take a uniformly random element of $\mathbb{F}_p$. Let the element be x, and P be a point in $E_{\mathcal{M},a}$ such that $x(P) = x$. We calculate $x^3 + ax^2 + x$, which is a square of $y(P)$, where $y(P)$ is the y-coordinate of P. If $x^3 + ax^2 + x$ is square in $\mathbb{F}_p$, then $P \in \ker(\pi_p - 1)$, and if $x^3 + ax^2 + x$ is not square in $\mathbb{F}_p$, then $P \in \ker(\pi_p + 1)$. If $x^3 + ax^2 + x$ is square, we define S to be a set of i such that the sign of e_i is $+1$, and if $x^3 + ax^2 + x$ is not square, we define S to be a set of i such that the sign of e_i is -1. If $S = \emptyset$, we repeat this procedure with another sample point.

Scalar multiplication (line 9 in Algorithm 1) Next, we calculate $P_1 = \frac{p+1}{k}P$, where $k = \prod_{i \in S} \ell_i$. The calculation uses the Montgomery ladder algorithm [23].

Calculation of isogenies (line 10-16 in Algorithm 1) We calculate $P_2 = \frac{k}{\ell_i}P_1$. The order of P_2 is 1 or ℓ_i. The probability that P_2 is not the identity is $1 - \frac{1}{\ell_i}$ [8]. Therefore, with highly probability, we get a point of order ℓ_i. Then, we calculate an ℓ_i-isogeny,

$$\phi\colon E_{\mathcal{M},a} \longrightarrow E_{\mathcal{M},a}/\langle P_2\rangle,$$

by using the formulas in [10, 21]. Denote the Montgomery coefficient of $E_{\mathcal{M},a}/\langle P_2\rangle$ by $a' \in \mathbb{F}_p$. From Theorem 3, a' is unique. We redefine e_i as $e_i - 1$ (if $e_i > 0$) or $e_i + 1$ (if $e_i < 0$), k as k/ℓ_i, P_1 as $\phi(P_1)$, and a as a'.

We repeat this calculation for all $i \in S$. After that, if the list of integers $(e_1, \ldots, e_n)$ is not the zero vector, we return to the ***Sampling points*** part.

Output (line 18 in Algorithm 1) If the list of integers $(e_1, \ldots, e_n)$ is the zero vector, we output the Montgomery coefficient $a' \in \mathbb{F}_p$.

Algorithm 1 Evaluating the class group action on Montgomery curves [8]

Input: $a \in \mathbb{F}_p$ such that $E_{\mathcal{M},a}$ is supersingular and a list of integers $(e_1, \ldots, e_n)$
Output: a' such that $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_{\mathcal{M},a} = E_{\mathcal{M},a'}$
1: **while** some $e_i \neq 0$ **do**
2: Sample a random $x \in \mathbb{F}_p$
3: $\mathbf{x}(P) \leftarrow (x : 1)$
4: Set $s \leftarrow +1$ if $x^3 + ax^2 + x$ is a square in $\mathbb{F}_p$, else $s \leftarrow -1$
5: Let $S = \{i \mid \mathrm{sign}(e_i) = s\}$
6: **if** $S = \emptyset$ **then**
7: Go to line 2
8: **end if**
9: $k \leftarrow \prod_{i \in S} \ell_i$, $\mathbf{x}(P) \leftarrow \mathbf{x}(((p+1)/k)P)$
10: **for all** $i \in S$ **do**
11: $\mathbf{x}(Q) \leftarrow \mathbf{x}((k/\ell_i)P)$
12: **if** $Q \neq (0 : 1 : 0)$ **then**
13: Compute an ℓ_i-isogeny $\phi\colon E_{\mathcal{M},a} \to E_{\mathcal{M},a'}$ with $\ker\phi = \langle Q\rangle$
14: $a \leftarrow a'$, $\mathbf{x}(P) \leftarrow \mathbf{x}(\phi(P))$, $k \leftarrow k/\ell_i$, $e_i \leftarrow e_i - s$
15: **end if**
16: **end for**
17: **end while**
18: **return** a

3.3 Elligator on Montgomery curves

In this subsection, we explain Elligator in detail. Elligator (specifically Elligator 2 in [6]) is used as a technique mapping some points in $\ker(\pi_p \pm 1)$ to points in $\ker(\pi_p \mp 1)$ over Montgomery curves. Meyer, Campos, and Reith used this technique for implementations of constant-time CSIDH algorithms for efficiency

[20]. By using Elligator, we can sample a pair of points in $\ker(\pi_p - 1)$ and points in $\ker(\pi_p + 1)$ efficiently. Elligator reduces the number of computing Legendre symbol in the constant-time CSIDH algorithm, and makes the algorithm more efficient.

First, we take a random value u from $\{2, 3, \dots, (p-1)/2\}$. We compute $v := a/(u^2 - 1)$, and output $(v, -v-a)$. If $v^3 + av^2 + v$ is square, then v is the x-coordinate of the point in $\ker(\pi_p - 1)$, and $-v-a$ is the x-coordinate of the point in $\ker(\pi_p + 1)$. If not square, then v is the x-coordinate of the point in $\ker(\pi_p + 1)$, and $-v-a$ is the x-coordinate of the point in $\ker(\pi_p - 1)$. These facts can be easily checked.

Moreover, Cervantes-Vázquez *et al.* proposed the constant-time projective Elligator for the constant-time CSIDH algorithm [9]. We show this algorithm in Algorithm 2.

Algorithm 2 Constant-time projective Elligator on Montgomery curves [9]

Input: $A, C \in \mathbb{F}_p$ such that $E_{\mathcal{M},A/C}$ is supersingular and an random element u from $\{2, 3, \dots, (p-1)/2\}$

Output: The projective x-coordinate of $P \in \ker(\pi_p - 1)$ and the projective x-coordinate of $Q \in \ker(\pi_p + 1)$

$t \leftarrow A((u^2-1)u^2A^2C + ((u^2-1)C)^3)$

$\epsilon \leftarrow \mathsf{isequal}(t, 0)$

$\alpha, \beta \leftarrow u, 0$

$\mathsf{cswap}(\alpha, \beta, \epsilon)$

$t' \leftarrow t + \alpha(u^2 + 1)$

$\zeta \leftarrow \mathsf{Legendre\_symbol}(t', p)$

$\epsilon' \leftarrow \mathsf{isequal}(\zeta, -1)$

$\mathsf{cswap}((X : Z), (X' : Z'), \epsilon')$

return $(X : Z), (X' : Z')$

4 Main theorems used for our algorithm

Here, we state and prove four theorems needed to construct the algorithm for evaluating the class group action based on Edwards curves.

First, we prove important lemmas in order to prove four main theorems.

Let E_d be a supersingular Edwards curve defined over $\mathbb{F}_p$, and p be a prime.

Lemma 1. *Let $p \equiv 3 \pmod 8$. If E_d satisfies $\mathrm{End}_p(E_d) \cong \mathbb{Z}[\pi_p]$, then d is not square.*

Proof. There exists a Montgomery curve $E_{\mathcal{M}}$ which is $\mathbb{F}_p$-isomorphic to E_d, by Corollary 1. If $E_{\mathcal{M}}[2] \subset E_{\mathcal{M}}(\mathbb{F}_p)$, Table 1 of [11] shows that the order of $E_{\mathcal{M}}$ or its quadratic twist can be divided by 8; however, both orders are $p+1 \equiv 4 \pmod 8$. $E_{\mathcal{M}}$ has the only one point of order 2 over $\mathbb{F}_p$. Therefore, E_d also has only one point of order 2 over $\mathbb{F}_p$.

Points of order 2 in E_d are $(0 : -1 : 1 : 0)$ and $(\pm\sqrt{d} : 0 : 0 : 1)$. Since $(0 : -1 : 1 : 0)$ is a $\mathbb{F}_p$-rational point, d is not square. □

Lemma 2. *Let $p \equiv 3 \pmod 8$. If E_d satisfies $\mathrm{End}_p(E_d) \cong \mathbb{Z}[\pi_p]$, then $1-d$ is not square.*

Proof. As $p \equiv 3 \pmod 8$, $\#E_d(\mathbb{F}_p) = p+1 \equiv 4 \pmod 8$.

By Lemma 1, there are no points at infinity on $E_d(\mathbb{F}_p)$. Hence, in this proof, we consider E_d to be an affine curve.

If a point (x, y) belongs to $E_d(\mathbb{F}_p)$, the points,

$$(-x, y), (x, -y), (-x, -y), (y, x), (-y, x), (y, -x), (-y, -x),$$

also belong to $E_d(\mathbb{F}_p)$. If $x \neq 0$, $y \neq 0$, $x \neq y$, and $x \neq -y$ hold, these eight points are different. If $x = 0$ or $y = 0$, the four points,

$$(0, 1), (0, -1), (1, 0), (-1, 0),$$

are different. If $x = y$ or $x = -y$, x is a root of the equation,

$$2x^2 = 1 + dx^4.$$

Therefore,

$$x^2 = \frac{1 \pm \sqrt{1-d}}{d}.$$

Assume that $1-d$ is square. Note that

$$\frac{1+\sqrt{1-d}}{d} \cdot \frac{1-\sqrt{1-d}}{d} = \frac{1-(1-d)}{d^2} = \frac{1}{d}.$$

By Lemma 1, d is not square. Hence, one of $\frac{1+\sqrt{1-d}}{d}$ or $\frac{1-\sqrt{1-d}}{d}$ is square, and the other one is not square. Therefore, if $x = y$ or $x = -y$, the four points,

$$(x, x), (x, -x), (-x, x), (-x, -x),$$

are different, where x is $\sqrt{\frac{1+\sqrt{1-d}}{d}}$ or $\sqrt{\frac{1-\sqrt{1-d}}{d}}$.

From the above, $\#E_d(\mathbb{F}_p) \equiv 4+4 \equiv 0 \pmod 8$ holds. This is a contradiction. Therefore, $1-d$ is not square. □

Lemma 3. *If P is a point of E_d such that $w(P) \in \mathbb{F}_p$, then $(\pi_p + 1)(P) \in \mathcal{G}_4$ or $(\pi_p - 1)(P) \in \mathcal{G}_4$.*

Proof. Since $\pi_p(w(P)) = w(\pi_p(P))$, $w(\pi_p(P)) = w(P)$. Therefore, $(\pi_p+1)(P) \in \mathcal{G}_4$ or $(\pi_p - 1)(P) \in \mathcal{G}_4$. □

Lemma 4 and Lemma 5 describe the relationship between points in $E_d[\pi_p \pm 1]$ and their w-coordinates.

Lemma 4. *Let* $p \equiv 3 \pmod 8$*. Let* P *be a point of* E_d*, not a point at infinity, and* $w(P) \neq 0$*. If* $P \in E_d[\pi_p + 1]$*, then* $w(P) \in \mathbb{F}_p$ *and is square in* $\mathbb{F}_p$*, and if* $P \in E_d[\pi_p - 1]$*, then* $w(P) \in \mathbb{F}_p$ *and is not square in* $\mathbb{F}_p$*.*

Proof. Denote the coordinates of P by (x, y) (affine coordinates). As $w(P) \neq 0$, $x \neq 0$ and $y \neq 0$. If $P \in E_d[\pi_p + 1]$, then $(x^p, y^p) = (-x, y)$. Therefore, $x^p = -x$ and $y \in \mathbb{F}_p$. As $(x^2)^p = x^2$ and $x \notin \mathbb{F}_p$, $x^2y^2 \in \mathbb{F}_p$ and x^2y^2 is not square. If $P \in E_d[\pi_p - 1]$, then $(x^p, y^p) = (x, y)$. Therefore, $x, y \in \mathbb{F}_p$. Thus, $x^2y^2 \in \mathbb{F}_p$ and x^2y^2 is square. Since d is not square by Lemma 1, Lemma 4 holds. □

Lemma 5. *Let* $p \equiv 3 \pmod 8$*. Let* $P \in E_d[\pi_p - 1]$ *or* $E_d[\pi_p + 1]$*, not a point at infinity, and* $w(P) \neq 0$*. If* $w(P)$ *is square in* $\mathbb{F}_p$*, then* $P \in E_d[\pi_p + 1]$*, and if* $w(P)$ *is not square in* $\mathbb{F}_p$*, then* $P \in E_d[\pi_p - 1]$*.*

Proof. This lemma obviously holds by Lemma 4. □

Lemma 6. *Let* P *be a point of* E_d*. Then, points* P_{odd} *and* P_{2power} *uniquely exist such that* $P = P_{odd} + P_{2power}$*, the order of* P_{odd} *is odd, and the order of* P_{2power} *is a power of* 2*.*

Proof. Note that $P \in E_d(\mathbb{F}_q)$, where q is a power of p. Therefore, P has finite order. By the fundamental theorem of finite abelian groups, there exist points P_{odd} and P_{2power} such that $P = P_{odd} + P_{2power}$, the order of P_{odd} is odd, and the order of P_{2power} is a power of 2.

Assume that $P_{odd} + P_{2power} = P'_{odd} + P'_{2power}$, where the orders of P_{odd} and P'_{odd} are odd, and the orders of P_{2power} and P'_{2power} are powers of 2. As $P_{odd} - P'_{odd} = -P_{2power} + P'_{2power}$,

$$P_{odd} - P'_{odd} = 0_d \text{ and } P_{2power} - P'_{2power} = 0_d.$$

Therefore, uniqueness holds. □

Lemma 7 states the property of P_{odd} and P_{2power} in Lemma 6. In particular, it is argued that P_{odd} belongs to $E_d[\pi_p \pm 1]$.

Lemma 7. *Let* P *be a point of* E_d *such that* $w(P) \in \mathbb{F}_p$*. Let* P_{odd} *and* P_{2power} *be points of* E_d *such that* $P = P_{odd} + P_{2power}$*, the order of* P_{odd} *is odd, and the order of* P_{2power} *is a power of* 2*. Then, one of the following holds.*

- $P_{odd} \in E_d[\pi_p - 1]$ *and* $(\pi_p - 1)(P_{2power}) \in \mathcal{G}_4$*.*
- $P_{odd} \in E_d[\pi_p + 1]$ *and* $(\pi_p + 1)(P_{2power}) \in \mathcal{G}_4$*.*

Proof. By Lemma 3, $(\pi_p \pm 1)(P) \in \mathcal{G}_4$. In the case that $(\pi_p - 1)(P) \in \mathcal{G}_4$, $(\pi_p - 1)(P_{odd}) = 0_d$, since the order of P_{odd} is odd and $\mathcal{G}_4$ is a cyclic group of order 4. Then, $(\pi_p - 1)(P_{2power}) = (\pi_p - 1)(P) \in \mathcal{G}_4$.

Similarly, in the case that $(\pi_p + 1)(P) \in \mathcal{G}_4$, $P_{odd} \in E_d[\pi_p + 1]$ and $(\pi_p + 1)(P_{2power}) \in \mathcal{G}_4$ hold. □

Lemma 8. *Let* P *be a point in* E_d *whose order is not a power of* 2*. Then, the number of points* Q *which satisfies* $w(Q) = w(P)$ *is* 8*.*

Proof. Assume that the number of points Q which satisfies $w(Q) = w(P)$ is not 8. Since a set $\pm P + \mathcal{G}_4$ does not have 8 elements, there are points $G_1, G_2 \in \mathcal{G}_4$ which satisfy $P + G_1 = -P + G_2$. However, the order of $-G_1 + G_2$ is a power of 2, and the order of $2P$ is not a power of 2. This is a contradiction.

This completes the proof of Lemma 8. □

Lemma 9. *Let $p \equiv 3 \pmod 8$. There exists a bijection,*

$$f\colon E_d[\pi_p + 1] \cap E_d[(p+1)/4] \longrightarrow E_d[\pi_p - 1] \cap E_d[(p+1)/4],$$

such that $f(0_d) = 0_d$.

Proof. We will prove that the cardinality of $E_d[\pi_p + 1] \cap E_d[(p+1)/4]$ and the cardinality of $E_d[\pi_p - 1] \cap E_d[(p+1)/4]$ are finite and equal and that 0_d belongs to both sets.

Since E_d is supersingular and $\pi_p - 1$ and $\pi_p^2 - 1$ are separable, $\deg(\pi_p^2 - 1) = \#E_d(\mathbb{F}_{p^2}) = (p+1)^2$ and $\deg(\pi_p - 1) = \#E_d(\mathbb{F}_p) = p+1$. Therefore, $\deg(\pi_p + 1) = p+1$. As $\pi_p - 1$ and $\pi_p + 1$ are separable, $\#E_d[\pi_p - 1] = p+1$ and $\#E_d[\pi_p + 1] = p+1$. As the set $E_d[\pi_p - 1] \cap E_d[(p+1)/4]$ is the set of all points of order odd in $E_d[\pi_p - 1]$,

$$\#(E_d[\pi_p - 1] \cap E_d[(p+1)/4]) = \frac{p+1}{4}.$$

Similarly,

$$\#(E_d[\pi_p + 1] \cap E_d[(p+1)/4]) = \frac{p+1}{4}.$$

We have proven that $\#(E_d[\pi_p+1] \cap E_d[(p+1)/4])$ and $\#(E_d[\pi_p-1] \cap E_d[(p+1)/4])$ are finite and equal.

It is obvious that 0_d belongs to $E_d[\pi_p + 1] \cap E_d[(p+1)/4]$ and $E_d[\pi_p - 1] \cap E_d[(p+1)/4]$.

This completes the proof of Lemma 9. □

We now prove four main theorems.

Roughly speaking, Theorem 4 claims that by examining a value $w(2P)$, we can get the w-coordinate of a point in $E_d[\pi_p \pm 1]$. This theorem leads to a sampling method.

Theorem 4. *Let $p \equiv 3 \pmod 8$. Let P be a point on an Edwards curve E_d such that the w-coordinate $w(P) \in \mathbb{F}_p$, the order of P is not a power of 2, and $w(P)$ is square. If $w(2P)$ is square, there exists P' such that $P' \in E_d[\pi_p + 1]$, $w(2P) = w(P')$, and $\frac{p+1}{4}P' = 0_d$. If $w(2P)$ is not square, there exists P' such that $P' \in E_d[\pi_p - 1]$, $1/w(2P) = w(P')$, and $\frac{p+1}{4}P' = 0_d$.*

Proof. Let (x, y) be the coordinates of P. Let P_{odd} and P_{2power} be points of E_d such that $P = P_{odd} + P_{2power}$, the order of P_{odd} is odd, and the order of P_{2power} is a power of 2. The existence of P_{odd} and P_{2power} are guaranteed by Lemma 6. By Lemma 7, one of the following holds.

– $(\pi_p - 1)(P_{2power}) \in \mathcal{G}_4$ and $P_{odd} \in E[\pi_p - 1]$.

– $(\pi_p + 1)(P_{2power}) \in \mathcal{G}_4$ and $P_{odd} \in E[\pi_p + 1]$.

It is easy to check that $(\pi_p + 1)\mathcal{G}_4 = \{0_d, (0, -1)\}$ and $(\pi_p - 1)\mathcal{G}_4 = \{0_d\}$. Therefore,

$$(\pi_p^2 - 1)(P_{2power}) = \begin{cases} 0_d & (\text{if } P_{odd} \in E[\pi_p + 1]), \\ 0_d \text{ or } (-1, 0) & (\text{if } P_{odd} \in E[\pi_p - 1]). \end{cases}$$

As $\pi_p^2 + p = 0$, $\pi_p^2 - 1 = -p - 1$. Since P_{2power} is a point whose order is a power of 2,

$$4P_{2power} = \begin{cases} 0_d & (\text{if } P_{odd} \in E[\pi_p + 1]), \\ 0_d \text{ or } (-1, 0) & (\text{if } P_{odd} \in E[\pi_p - 1]). \end{cases}$$

Hence, if $P_{odd} \in E[\pi_p + 1]$, then

$$2P_{2power} = 0_d, (0, -1), (\pm\sqrt{d} : 0 : 0 : 1),$$

and if $P_{odd} \in E[\pi_p - 1]$, then

$$2P_{2power} = 0_d, (0, -1), (\pm\sqrt{d} : 0 : 0 : 1), (1, 0), (-1, 0), (0 : \pm\sqrt{d} : 0 : 1).$$

It is easy to check that if $w(2P_{2power}) = 0$, then $w(2P) = w(2P_{odd})$, and if $w(2P_{2power}) = \infty$, then $w(2P) = 1/w(2P_{odd})$. Therefore, if $w(2P)$ is square, then $w(2P_{odd})$ is square, and if $w(2P)$ is not square, then $w(2P_{odd})$ is not square. By Lemma 5, if $w(2P)$ is square, then $2P_{odd} \in E_d[\pi_p + 1]$, and if $w(2P)$ is not square, then $2P_{odd} \in E_d[\pi_p - 1]$.

Denote $w(P)$ by w. By the Edwards addition formula (7), we have

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

Since w is square, if $w(2P)$ is square, then $y^2 - x^2 \in \mathbb{F}_p$, and if $w(2P)$ is not square, then $y^2 - x^2 \notin \mathbb{F}_p$. As

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

if $w(2P)$ is square, then the y-coordinate of $2P$ is an element of $\mathbb{F}_p$, and if $w(2P)$ is not square, then the y-coordinate of $2P$ is not an element of $\mathbb{F}_p$.

In the case that $w(2P)$ is square, $y(2P) \in \mathbb{F}_p$ and $2P_{odd} \in E_d[\pi_p + 1]$. Therefore, $y(2P_{odd}) \in \mathbb{F}_p$. Assume that $2P_{2power} = (\sqrt{d} : 0 : 0 : 1)$ or $(-\sqrt{d} : 0 : 0 : 1)$. It is easy to check that

$$y(2P) = \pm \frac{1}{\sqrt{d} \cdot y(2P_{odd})}.$$

As $y(2P_{odd}) \in \mathbb{F}_p$, $y(2P) \notin \mathbb{F}_p$ by Lemma 1. This is a contradiction. We conclude that $2P_{2power}$ is 0_d or $(0, -1)$. Therefore, $w(2P) = w(2P_{odd})$. As $(\pi_p^2 - 1)(2P_{odd}) = 0_d$,

$$\frac{p + 1}{4}(2P_{odd}) = 0_d.$$

In the case that $w(2P)$ is not square, $y(2P) \notin \mathbb{F}_p$ and $2P_{odd} \in E_d[\pi_p - 1]$. Therefore, $y(2P_{odd}) \in \mathbb{F}_p$. Assume that

$$2P_{2power} = 0_d, (0, -1), (1, 0), (-1, 0).$$

It is easy to check that $y(2P) = \pm y(2P_{odd})$. As $y(2P_{odd}) \in \mathbb{F}_p$, $y(2P) \in \mathbb{F}_p$. This is a contradiction. We conclude that $2P_{2power}$ is $(\pm\sqrt{d} : 0 : 0 : 1)$ or $(0 : \pm\sqrt{d} : 0 : 1)$. Therefore, it is easy to check that $w(2P) = 1/w(2P_{odd})$. As $(\pi_p^2 - 1)(2P_{odd}) = 0_d$,

$$\frac{p+1}{4}(2P_{odd}) = 0_d.$$

Let P' be $2P_{odd}$. This completes the proof of Theorem 4. □

Theorem 5 shows that there is no bias in the points generated by the sampling method derived from Theorem 4.

Theorem 5. *Let $p \equiv 3 \pmod 8$. Let P be a point on an Edwards curve E_d such that the w-coordinate $w(P) \in \mathbb{F}_p$, the order of P is not a power of 2, and $w(P)$ is square. The number of $w(P)$ such that $w(2P)$ is square is the same as the number of $w(P)$ such that $w(2P)$ is not square.*

Proof. Let the coordinates of P be (x, y). Let P_{odd} and P_{2power} be points of E_d such that $P = P_{odd} + P_{2power}$, the order of P_{odd} is odd, and the order of P_{2power} is a power of 2. The existence of P_{odd} and P_{2power} are guaranteed by Lemma 6. As shown in the proof of Theorem 4, we have

$$2P_{2power} = 0_d, (0, -1), (\pm\sqrt{d} : 0 : 0 : 1), (0 : \pm\sqrt{d} : 0 : 1).$$

If $2P_{2power}$ is 0_d or $(0, -1)$, $w(P_{2power})$ is 0 or ∞, since it is easy to check that

$$P_{2power} = 0_d, (0, -1), (\pm 1, 0), (\pm\sqrt{d} : 0 : 0 : 1), (0 : \pm\sqrt{d} : 0 : 1).$$

If $2P_{2power}$ is $(\pm\sqrt{d} : 0 : 0 : 1)$ or $(0 : \pm\sqrt{d} : 0 : 1)$, $w(P_{2power})$ is ± 1 since

$$w(2P_{2power}) = \frac{4w(P_{2power})((1 + w(P_{2power}))^2 - 4w(P_{2power})/d)}{(1 - w(P_{2power}))^2(1 + w(P_{2power}))^2}.$$

Assume that $w(P_{2power})$ is -1. $w(2P_{2power}) = \infty$. As shown in the proof of Theorem 4, $(\pi_p - 1)(P_{odd}) = 0_d$. Let the coordinates of P_{odd} be (x_o, y_o). It is easy to check that

$$P_{2power} = \left(\sqrt{\sqrt{\frac{1}{d}}}, \sqrt{-\sqrt{\frac{1}{d}}}\right) + Q',$$

where Q' is a point of E_d such that $w(Q') = 0$ or $w(Q') = \infty$. From the addition formula of Edward curves,

$$P = P_{odd} + P_{2power} = \left(\frac{x_o\sqrt{-\sqrt{\frac{1}{d}}} + y_o\sqrt{\sqrt{\frac{1}{d}}}}{1 + dx_oy_o\sqrt{\frac{-1}{d}}}, \frac{y_o\sqrt{-\sqrt{\frac{1}{d}}} - x_o\sqrt{\sqrt{\frac{1}{d}}}}{1 - dx_oy_o\sqrt{\frac{-1}{d}}}\right) + Q'.$$

Therefore,

$$w(P) = \frac{(2x_oy_o + (y_o^2 - x_o^2)\sqrt{-1})^2}{(1 + dx_o^2y_o^2)^2} \text{ or } \frac{(1 + dx_o^2y_o^2)^2}{(2x_oy_o + (y_o^2 - x_o^2)\sqrt{-1})^2}.$$

As $p \equiv 4 \pmod 3$, -1 is not square. Since P_{odd} is not 0_d, $x_o \neq 0$ and $y_o \neq 0$. If we assume that $x_o^2 = y_o^2$, then it is easy to check that $2x_o^2 = 1 + dx_o^4$, and

$$x_o^2 = \frac{1 \pm \sqrt{1-d}}{d} \notin \mathbb{F}_p \quad \text{(by Lemma 2)}.$$

Since $x_o^2 \in \mathbb{F}_p$, $x_o^2 \neq y_o^2$. Therefore, $(2x_oy_o + (y_o^2 - x_o^2)\sqrt{-1})^2$ does not belong to $\mathbb{F}_p$. Hence, $w(P) \notin \mathbb{F}_p$. This is a contradiction. We conclude $w(P_{2power})$ is 0 or ∞ or 1.

If $w(2P)$ is square, as shown in the proof of Theorem 4, $w(P_{odd})$ is square and $2P_{2power} = 0_d$ or $(0, -1)$. Therefore, $w(P_{2power})$ is 0 or ∞. If $w(2P)$ is not square, as shown in the proof of Theorem 4, $w(P_{odd})$ is not square and $2P_{2power} = (\pm\sqrt{d} : 0 : 0 : 1)$ or $(0 : \pm\sqrt{d} : 0 : 1)$. Therefore, $w(P_{2power})$ is 1.

We prove that if $P_{odd} \in E_d[\pi_p - 1]$, then $w(P_{odd} + Q)$ is square for all points Q at which $w(Q)$ is 1. It is easy to check that

$$Q = \left(\sqrt{1 + \sqrt{-1}r}, \sqrt{1 - \sqrt{-1}r}\right) + Q',$$

where $r = \sqrt{\frac{1-d}{d}}$, and Q' is a point such that $w(Q') = 0$ or $w(Q') = \infty$. By Lemma 1 and Lemma 2, $r \in \mathbb{F}_p$. Let the coordinates of P_{odd} be (x_o, y_o). Denote $\left(\sqrt{1 + \sqrt{-1}r}, \sqrt{1 - \sqrt{-1}r}\right)$ by R. Note that

$$P_{odd} + R = \left(\frac{x_o\sqrt{1 - \sqrt{-1}r} + y_o\sqrt{1 + \sqrt{-1}r}}{1 + \sqrt{d}x_oy_o}, \frac{y_o\sqrt{1 - \sqrt{-1}r} - x_o\sqrt{1 + \sqrt{-1}r}}{1 - \sqrt{d}x_oy_o}\right).$$

Therefore,

$$\begin{aligned} w\left(P_{odd} + R\right) =& \frac{d(-2x_oy_o\sqrt{-1}r + (y_o^2 - x_o^2)\sqrt{1 + r^2})^2}{(1 - dx_o^2y_o^2)^2} \\ =& \frac{(-2x_oy_o\sqrt{-d}r + (y_o^2 - x_o^2))^2}{(1 - dx_o^2y_o^2)^2}. \end{aligned}$$

By Lemma 1, $\sqrt{-d} \in \mathbb{F}_p$. As $P_{odd} \in E_d[\pi_p - 1]$, $x_o, y_o \in \mathbb{F}_p$. Therefore, $w\left(P_{odd} + R\right)$ belongs to $\mathbb{F}_p$ and is square. Since $w(P_{odd} + Q) = w(P_{odd} + R)$ or $1/w(P_{odd} + R)$, $w(P_{odd} + Q)$ belongs to $\mathbb{F}_p$ and is square.

Let S_+ be the set of points P of E_d such that both $w(P)$ and $w(2P)$ are square and the order of P is not a power of 2, and let S_- be the set of points P of E_d such that $w(P)$ is square, $w(2P)$ is not square, and the order of P is not a power of 2. From Lemma 8, it suffices to prove that there is a bijection $\phi\colon S_+ \to S_-$. Define $\phi\colon S_+ \to S_-$ as follows.

$$\phi(P) := f(P_{odd}) + P_{2power} + R,$$

where P_{odd} and P_{2power} are points of E_d such that $P = P_{odd} + P_{2power}$, the order of P_{odd} is odd, the order of P_{2power} is a power of 2, R is defined as above, and f is the bijection in Lemma 9. As has already been shown, if $P \in S_+$, then $w(P_{2power})$ is 0 or ∞. As $f(P_{odd}) \in E_d[\pi_p - 1]$ and $w(P_{2power} + R) = 1$, $w(\phi(P))$ is square. Since $w(2\phi(P)) = 1/w(2f(P_{odd}))$ and $2f(P_{odd}) \in E_d[\pi_p - 1]$, $w(2\phi(P))$ is not square. As $f(P_{odd})$ is not 0_d, the order of $\phi(P)$ is not a power of 2. From Lemma 6 and the above, ϕ is well-defined. Define $\psi\colon S_- \to S_+$ as follows.

$$\psi(P) := f^{-1}(P_{odd}) + P_{2power} - R,$$

where P_{odd} and P_{2power} are points of E_d such that $P = P_{odd} + P_{2power}$, the order of P_{odd} is odd, and the order of P_{2power} is a power of 2. As has already een shown, if $P \in S_-$, then $w(P_{2power}) = 1$. As $w(P_{2power} - R)$ is 0 or ∞, $w(\psi(P)) = w(f^{-1}(P_{odd}))$ or $1/w(f^{-1}(P_{odd}))$. Since $f^{-1}(P_{odd}) \in E_d[\pi_p + 1]$, $w(f^{-1}(P_{odd}))$ is square by Lemma 4. Hence, $w(\psi(P))$ and $w(2\psi(P))$ are square. As $f^{-1}(P_{odd})$ is not 0_d, the order of $\psi(P)$ is not a power of 2. From Lemma 6 and the above, ψ is well-defined. It is easy to check that $\psi = \phi^{-1}$.

This completes the proof of Theorem 5. □

Theorem 6 claims that the probability of success of the sampling method derived from Thorem 4 is sufficiently large (same probability as that on Montgomery curves).

Theorem 6. *Let p be $4 \cdot \ell_1 \cdots \ell_n - 1$, where the $\ell_1, \ldots, \ell_n$ are small distinct odd primes. Let P be a point on an Edwards curve E_d such that the w-coordinate $w(P) \in \mathbb{F}_p$, the order of P is not a power of 2, and $w(P)$ is square. The probability that $\frac{p+1}{4\ell_i}P'$ is a point of order ℓ_i is $\frac{(\ell_i - 1)\frac{N}{\ell_i}}{N-1} \approx 1 - \frac{1}{\ell_i}$, where P' is a point in Theorem 4, and $N = \ell_1 \cdot \ell_2 \cdots \ell_n$.*

Proof. Let P_{odd} and P_{2power} be points of E_d such that $P = P_{odd} + P_{2power}$, the order of P_{odd} is odd, and the order of P_{2power} is a power of 2. As shown in the proof of Theorem 4, $P' = 2P_{odd}$. As shown in the proof of Theorem 5, for each point $Q \neq 0_d$ in $E_d[\pi_p + 1] \cap E_d[(p+1)/4]$ or $E_d[\pi_p - 1] \cap E_d[(p+1)/4]$, there is a point $\tilde{Q}$ that satisfies $w(\tilde{Q}) \in \mathbb{F}_p$, $w(\tilde{Q})$ is square, and $2\tilde{Q}_{odd} = Q$. It is easy to check that if $Q_1 \neq Q_2$, then $w(\tilde{Q_1}) \neq w(\tilde{Q_2})$. Therefore, if we uniform randomly take P that satisfies $w(P)$ is square, then P' is a uniformly random point of $E_d[\pi_p + 1] \cap E_d[(p+1)/4] \setminus \{0_d\}$ or $E_d[\pi_p - 1] \cap E_d[(p+1)/4] \setminus \{0_d\}$. Since

$$E_d[\pi_p + 1] \cap E_d[(p+1)/4] \cong \mathbb{Z}/((p+1)/4)\mathbb{Z} \cong \mathbb{Z}/\ell_1\mathbb{Z} \times \cdots \times \mathbb{Z}/\ell_n\mathbb{Z},$$

$$E_d[\pi_p - 1] \cap E_d[(p+1)/4] \cong \mathbb{Z}/((p+1)/4)\mathbb{Z} \cong \mathbb{Z}/\ell_1\mathbb{Z} \times \cdots \times \mathbb{Z}/\ell_n\mathbb{Z},$$

Theorem 6 holds. □

Theorem 7 shows that an Edwards coefficient d is unique. Therefore, we can use these coefficients as shared keys.

Theorem 7. *Let $p \equiv 3 \pmod 8$ and E be a supersingular elliptic curve defined over $\mathbb{F}_p$. Then $\mathrm{End}_p(E) \cong \mathbb{Z}[\pi_p]$ holds if and only if there exists $d \in \mathbb{F}_p$ such that E is $\mathbb{F}_p$-isomorphic to an Edwards curve E_d. Moreover, if such a d exists, then it is unique.*

Proof. The first half of this theorem follows from Corollary 1, Corollary 2, and Theorem 3.

Let us prove the uniqueness of d. Let $d_1, d_2 \in \mathbb{F}_p$ such that E_{d_1} and E_{d_2} are supersingular Edwards curves, $\mathrm{End}_p(E_{d_1}) \cong \mathbb{Z}[\pi_p]$, $\mathrm{End}_p(E_{d_2}) \cong \mathbb{Z}[\pi_p]$, and $E_{d_1} \cong E_{d_2}$ over $\mathbb{F}_p$.

As $1 - d_1$ and $1 - d_2$ are not square by Lemma 2,

$$E_{d_i} \cong Y^2Z = X^3 - \frac{2(1+d_i)}{1-d_i}X^2Z + XZ^2 \quad (i = 1, 2)$$

holds by Corollary 1. Therefore,

$$\frac{2(1+d_1)}{1-d_1} = \frac{2(1+d_2)}{1-d_2}$$

holds by the uniqueness of coefficients in Theorem 3. This equation reduces to $d_1 = d_2$.

This completes the proof of Theorem 7. □

Now we proved all main theorems. Though the following lemma is not important essentially, we use it in order to construct the CSIDH algorithm. We use Lemma 10 for rejecting points whose order is a power of 2 in the ***Sampling points*** calculation of Algorithm 3.

Lemma 10. *Let $p \equiv 3 \pmod 8$. Let P be a point on E_d such that $w(P) \in \mathbb{F}_p$ and the order of P is a power of 2. Then, $w(P)$ is 0 or ± 1.*

Proof. By Lemma 7, $(\pi_p - 1)(P) \in \mathcal{G}_4$ or $(\pi_p + 1)(P) \in \mathcal{G}_4$. As $(\pi_p - 1)\mathcal{G}_4 = \{0_d\}$, $(\pi_p + 1)\mathcal{G}_4 = \{0_d, (0, -1)\}$, and $\pi_p^2 - 1 = -p - 1$, we have

$$4P = 0_d, (0, -1).$$

Therefore, it is easy to check that

$$2P = 0_d, (0, -1), (\pm 1, 0), (\pm\sqrt{d} : 0 : 0 : 1), (0 : \pm\sqrt{d} : 0 : 1).$$

Hence, $w(2P) = 0$ or $w(2P) = \infty$. Since

$$w(2P) = \frac{4w(P)((1 + w(P))^2 - 4w(P)/d)}{(1 - w(P))^2(1 + w(P))^2},$$

$w(P) = 0, \frac{d-2\pm 2\sqrt{1-d}}{d}, 1, -1$. From Lemma 2, $1 - d$ is not square. Therefore, $w(P) = 0, \pm 1$. □

Algorithm 3 Evaluating the class group action on Edwards curves

Input: $d \in \mathbb{F}_p$ such that Edwards curve E_d is supersingular and a list of integers $(e_1, \dots, e_n)$
Output: d' such that $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_d = E_{d'}$
1: **while** some $e_i \neq 0$ **do**
2: $w \leftarrow 0$
3: **while** $w = 0$ or $w = 1$ or $w = -1$ **do**
4: Sample a random $w \in \mathbb{F}_p$
5: **end while**
6: $w \leftarrow w^2$ (**Theorem 4, 5**)
7: $\mathbf{w}(P) \leftarrow (w : 1)$
8: Compute $\mathbf{w}(2P)$ (**Theorem 4**)
9: $(W : Z) \leftarrow \mathbf{w}(2P)$
10: Set $s \leftarrow +1$ if W is a square in $\mathbb{F}_p$, else $s \leftarrow -1$
11: Let $S = \{i \mid \mathrm{sign}(e_i) = s\}$
12: **if** $S = \emptyset$ **then**
13: Go to line 2
14: **end if**
15: $\mathbf{w}(P) \leftarrow (W : Z)$, $k \leftarrow \prod_{i \in S} \ell_i$
16: $\mathbf{w}(P) = (W : Z) \leftarrow \mathbf{w}(((p+1)/4k)P)$ (**Theorem 4, 6**)
17: **if** $s = 1$ **then**
18: $\mathbf{w}(P) \leftarrow (Z : W)$ (**Theorem 4**)
19: **end if**
20: **for all** $i \in S$ **do**
21: $\mathbf{w}(Q) \leftarrow \mathbf{w}((k/\ell_i)P)$
22: **if** $Q \neq 0_d$ **then**
23: Compute an ℓ_i-isogeny $\phi \colon E_d \to E_{d'}$ with $\ker \phi = \langle Q \rangle$
24: $d \leftarrow d'$, $\mathbf{w}(P) \leftarrow \mathbf{w}(\phi(P))$, $k \leftarrow k/\ell_i$, $e_i \leftarrow e_i - s$
25: **end if**
26: **end for**
27: **end while**
28: **return** d (**Theorem 7**)

5 Evaluating the class group action on Edwards curves

In this section, we propose the method for evaluating the class group action based on Edwards curves. The theorems proved in the previous section will be used to construct the method. The algorithm is described in Algorithm 3. All of its calculations are done over $\mathbb{F}_p$.

The inputs of the algorithm are an Edwards coefficient $d \in \mathbb{F}_p$ and a list of integers $(e_1, \dots, e_n)$. The output of this algorithm is an Edwards coefficient $d' \in \mathbb{F}_p$ such that $E_{d'} = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_d$.

Sampling points (line 2-14 in Algorithm 3) To sample a point that belongs to $E_d[\pi_p - 1]$ or $E_d[\pi_p + 1]$, we take a uniformly random element of $\mathbb{F}_p$. Denote this element by w. If w is 0 or ± 1, we take a random element again. (We reject any point whose order is a power of 2 by Lemma 10.) Then, we calculate w^2.

Let P be a point in E_d such that $w(P) = w^2$. By Theorem 4, if $w(2P)$ is square in $\mathbb{F}_p$, then there exists a point P' such that $w(P') = w(2P)$, $\frac{p+1}{4}P' = 0_d$, and $P' \in E_d[\pi_p + 1]$. If $w(2P)$ is not square in $\mathbb{F}_p$, then there exists a point P' such that $w(P') = 1/w(2P)$, $\frac{p+1}{4}P' = 0_d$, and $P' \in E_d[\pi_p - 1]$. Thus, we calculate $w(2P)$ by using the doubling formulas on Edwards curves and determine whether $w(2P)$ is square or not. If $w(2P)$ is square, we can use $w(2P)$ as an element of $E_d[\pi_p+1]$. If $w(2P)$ is not square, we can use $1/w(2P)$ as an element of $E_d[\pi_p-1]$. If $w(2P)$ is square, we define S as a set of i such that the sign of e_i is -1. If $w(2P)$ is not square, we define S as a set of i such that the sign of e_i is $+1$. If $S = \emptyset$, we go back to the ***Sampling points*** calculation.

From Theorem 5, the probability of getting points in $E_d[\pi_p - 1]$ is equal to the probability of getting points in $E_d[\pi_p + 1]$.

Scalar multiplication (line 15-19 in Algorithm 3) From Theorem 4, it suffices to calculate $w(\frac{p+1}{4k}(P'))$ instead of $w(\frac{p+1}{k}(P))$, where $k = \prod_{i\in S} \ell_i$. To calculate $w(\frac{p+1}{4k}(P'))$ efficiently, we use Algorithm 4.

Algorithm 4 The Edwards ladder using P and $2P$

Input: E_d, $k = \sum_{i=0}^{\ell-1} k_i 2^i$ with $k_{\ell-1} = 1$, $(W_0 : 1) = \mathbf{w}(P)$, and $(W : Z) = \mathbf{w}(2P)$ s.t. $P \in E_d$
Output: $(W' : Z') = \mathbf{w}(kP)$
$(W_1 : Z_1) \leftarrow (W_0 : 1)$ and $(W_2 : Z_2) \leftarrow (W : Z)$
for $i = \ell - 2$ **down to** 0 **do**
  if $k_i = 0$ **then**
    $(W_1 : Z_1) \leftarrow 2(W_1 : Z_1)$ (doubling on E_d)
    $(W_2 : Z_2) \leftarrow (W_1 : Z_1) + (W_2 : Z_2)$ (addition on E_d with $Z_0 = 1$)
  else
    $(W_2 : Z_2) \leftarrow 2(W_1 : Z_1)$ (doubling on E_d)
    $(W_1 : Z_1) \leftarrow (W_1 : Z_1) + (W_2 : Z_2)$ (addition on E_d with $Z_0 = 1$)
  end if
end for
return $(W_1 : Z_1)$

If $w(2P)$ is not square, the proof of Theorem 4 indicates that $P' = 2P + Q$, where Q is a point at infinity. Since $\frac{p+1}{4k}$ is odd and an odd multiple of Q is also a point at infinity, $w(\frac{p+1}{4k}(P')) = 1/w(\frac{p+1}{4k}(2P))$.

Calculation of isogenies (line 20-26 in Algorithm 3) By Theorem 6 and 7, we can calculate isogenies by using the same strategy as the original CSIDH algorithm. To do so, we can use the formulas on Edwards curves [18].

Output (line 28 in Algorithm 3) If the list of integers $(e_1, \ldots, e_n)$ is the zero vector, we output the Edwards coefficient $d' \in \mathbb{F}_p$.

Table 2. Computational costs on each CSIDH algorithm

	Montgomery [21]	Edwards (y-coordinate)	Edwards (w-coordinate)
M	328,195	332,707	328,055
S	116,915	116,893	116,857
a	332,822	355,533	331,844
total	438,368	443,999	438,133

Remark 1. To determine whether $w(2P)$ is square or not, we only need to consider W, where $(W : Z) = \mathbf{w}(2P)$. We explain the reason below.

Recall the isogenies formulas on Edwards curves:

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

As ℓ is odd, if D is not square, then D' is also not square. At the beginning of the algorithm, we let $(D : C) = (d : 1)$. Hence, we can assume that D is not square. Let the projective w-coordinates of P be $(W' : Z')$, the projective w-coordinates of $2P$ be $(W : Z)$, and the projective coordinates of d be $(D : C)$. Z is not square, since

$$\mathbf{w}(2P) = (4W'Z'(D(W' + Z')^2 - 4CW'Z') : D(W' + Z')^2(W' - Z')^2).$$

Therefore, if W is square, then $w(2P)$ is not square. Moreover, if W is not square, then $w(2P)$ is square.

6 Computational costs

In this section, we compare computational costs of our proposed CSIDH algorithm and that of the algorithm proposed by Meyer and Reith [21], theoretically. Moreover, we show our result of implementation on three different CSIDH algorithms: the algorithm on Montgomery curves proposed by Meyer and Reith [21] (Algorithm 1), that on Edwards curves with y-coordinates (Algorithm 6 in Appendix Appendix C), and that on Edwards curves with w-coordinates (Algorithm 3). The results are summarized in Table 2. In this table, "total" means total numbers of multiplications on $\mathbb{F}_p$, where we assume $1\mathbf{S} = 0.8\mathbf{M}$, and $1\mathbf{a} = 0.05\mathbf{M}$.

6.1 Comparing computational costs theoretically

Our proposed CSIDH algorithm using only w-coordinates on Edwards curves is as fast as (or a little bit faster than) the algorithm proposed by Meyer and Reith [21]. In this subsection, we explain computational savings of our algorithm relative to the algorithm of Meyer and Reith.

On Edwards curves, the ***Sampling points*** calculation costs $1\mathbf{S}$ for taking a uniformly random element of $(\mathbb{F}_p)^2$ and requires one doubling on Edwards curves with $Z = 1$ (the cost of $4\mathbf{M} + 1\mathbf{S} + 5\mathbf{a}$) for determining the set which the point belongs to. On the other hand, on Montgomery curves, ***Sampling points*** calculation entails calculating $Cx^3 + Ax^2 + Cx$ (the cost of $3\mathbf{M} + 1\mathbf{S} + 2\mathbf{a}$) for determining the set which the point belongs to, where $(A : C)$ is a projective coordinates of a. Therefore, our algorithm saves a cost of $-\mathbf{M} - \mathbf{S} - 3\mathbf{a}$ per ***Sampling points*** calculation.

The ***Scalar multiplication*** part entails multiplication by $\frac{p+1}{4k}$ on Edwards curves and multiplication by $\frac{p+1}{k}$ on Montgomery curves. Therefore, per ***Scalar multiplication***, the proposed algorithm saves the cost of a doubling on Edwards curves with $Z = 1$ and the cost of doubling on Edwards curves with $Z \neq 1$ (i.e., $8\mathbf{M} + 3\mathbf{S} + 9\mathbf{a}$).

The probability that $S = \emptyset$ after performing the ***Sampling points*** calculation is at most $\frac{1}{2}$, by Theorem 5. Hence, we expect the proposed algorithm to save at least

$$\frac{1}{2}(-\mathbf{M} - \mathbf{S} - 3\mathbf{a}) + \frac{1}{2}(8\mathbf{M} + 3\mathbf{S} + 9\mathbf{a} - \mathbf{M} - \mathbf{S} - 3\mathbf{a}) = 3\mathbf{M} + \frac{1}{2}\mathbf{S} + \frac{3}{2}\mathbf{a},$$

per ***Sampling points*** and ***Scalar multiplication*** calculation.

The difference between ***Calculation of isogenies*** on Edwards curves and on Montgomery curves is only in calculating the isogenies. The computational cost of calculating $(2s + 1)$-degree isogenies on Edwards curves is $(6s + 2)\mathbf{M} + 8\mathbf{S} + (4s + 6)\mathbf{a}$ and that of the two s-th powers, while the computational cost on Montgomery curves is $(6s + 2)\mathbf{M} + 8\mathbf{S} + (4s + 8)\mathbf{a}$ and that of the two s-th powers. Therefore, the proposed algorithm saves $2\mathbf{a}$ per isogeny calculation.

From the above, we conclude that our proposed CSIDH algorithm using only Edwards curves is as fast as or a little bit faster than the algorithm proposed by Meyer and Reith [21].

6.2 Implementations

We measured average of computational costs of 50000 times, respectively. The results are summarized in Table 2. Here, p was chosen as $4 \cdot \ell_1 \cdots \ell_{74} - 1$, where ℓ_1 through ℓ_{73} were the smallest 73 odd primes and $\ell_{74} = 587$, and m was chosen as 5. These are parameters proposed in [8]. Secret keys were randomly taken for 50000 times. In our implementation, we did not use formulas to compute isogenies in $\tilde{O}(\sqrt{\ell})$.

As shown in Table 2, there is no big difference of computational costs among the three different algorithms. The algorithm on Edwards curves with w-coordinates is the little bit fastest one in our implementation.

Remark 2. Our implementation of the algorithm on Montgomery curves is based on the algorithm proposed by Meyer and Reith [21]. There are some techniques to make the CSIDH algorithm faster [20, 9]. We did not implement these techniques. However, as far as we know, these techniques affect only a little or can be also adapted to the our proposed algorithms. Therefore, even if we consider these techniques, we can conclude that there is no big difference of computational costs among the above three different algorithms.

7 Elligator like technique on Edwards curves

In this section, we propose an Elligator like technique on Edwards curves with using w-coordinates. As far as we know, currently proposed constant-time algorithms can be migrated to those on Edwards curves except for the part of Elligator.

7.1 Construction

We introduce the following theorem.

Theorem 8. *Let $p \equiv 3 \pmod 8$. Let P be a point on an Edwards curve E_d such that the w-coordinate $w(P) \in \mathbb{F}_p$, the order of P is not a power of 2. If $w(2P)$ is square, there exists P' such that $P' \in E_d[\pi_p + 1]$, $w(P') = w(4P)$, and $\frac{p+1}{4}P' = 0_d$. If $w(2P)$ is not square, there exists P' such that $P' \in E_d[\pi_p - 1]$, $w(P') = w(4P)$, and $\frac{p+1}{4}P' = 0_d$.*

Proof. From Lemma 7, we have $P_{odd} \in E_d[\pi_p \pm 1]$. From the proof of Theorem 4, we have $w(2P) = w(2P_{\text{odd}})^{\pm 1}$. Hence, from Lemma 5, if $w(2P)$ is square, then $2P_{odd}$ and $4P_{odd}$ belong to $E_d[\pi_p + 1]$, and if $w(2P)$ is not square, then $2P_{odd}$ and $4P_{odd}$ belong to $E_d[\pi_p - 1]$. From the proof of Theorem 4, we have $4P_{2power} = 0_d, (-1, 0)$. Therefore, it holds that $w(4P) = w(4P_{odd})$.

This completes the proof of Theorem 8. □

From this theorem, it is sufficient to output a point Q such that $\chi(w(2Q)) = -\chi(w(2P))$ from an input P, where the map $\chi\colon \mathbb{F}_p \to \mathbb{F}_p$ is defined as $\chi(a) := a^{(p-1)/2}$.

We recall the doubling formulas on an Edwards curve E_d:

$$w(2P) = \frac{4w(P)(d(w(P)+1)^2 - 4w(P))}{d(w(P)-1)^2(w(P)+1)^2} = \frac{4w(P)\left(w(P)^2 + \frac{2d-4}{d}w(P) + 1\right)}{(w(P)-1)^2(w(P)+1)^2}.$$

We see the polynomial $w\left(w^2 + \frac{2d-4}{d}w + 1\right)$ is similar to the right-hand side of the definitional equation of a Montgomery curve. Therefore, we get the required map by considering Elligator on $y^2 = x^3 + \frac{2d-4}{d}x^2 + x$.

The outline of the construction is as follows. First, we take a random element u from $\{2, 3, \ldots, (p-1)/2\}$. Take the point P such that $w(P) = (2d-4)/(d(u^2 - 1))$. Compute $w(2P)$, and determine whether $w(2P)$ is square or not. Let Q be a point such that $w(Q) = -w(P) - (2d-4)/d$. Note that it holds that $\chi(w(2Q)) = -\chi(w(2P))$. Finally, we output $\{w(4P), w(4Q)\}$.

In [9], the constant-time projective Elligator was proposed. Our proposed technique can also be changed to the constant-time projective one using the same way in [9]. The result is shown in Algorithm 5. Although this technique is not Elligator, we often call this technique "Elligator on Edwards curves" for simplicity.

Algorithm 5 Constant-time projective Elligator on Edwards curves

Input: $D, C \in \mathbb{F}_p$ such that $E_{D/C}$ is supersingular and an random element u from $\{2, 3, \ldots, (p-1)/2\}$

Output: The projective w-coordinate of $P \in E_{D/C}[\pi_p - 1]$ and the projective w-coordinate of $Q \in E_{D/C}[\pi_p + 1]$

1: $t \leftarrow (2D-4C)((u^2-1)u^2(2D-4C)^2 D + ((u^2-1)D)^3)$
2: $\epsilon \leftarrow \mathsf{isequal}(t, 0)$
3: $\alpha, \beta, \gamma, \delta \leftarrow u, 0, 1, 0$
4: $\mathsf{cswap}(\alpha, \beta, \epsilon)$
5: $\mathsf{cswap}(\gamma, \delta, \epsilon)$
6: $t' \leftarrow t + \alpha D^4(u^2+1)$
7: $s \leftarrow ((2D-4C) + D(u^2-1))^2((2D-4C) - D(u^2-1))^2 + \gamma D^4(u^2-1)^2$
8: $(W_1 : Z_1) \leftarrow ((2D-4C) + \alpha D(u^2-1) : D(u^2-1))$
9: $(W_1' : Z_1') \leftarrow (-(2D-4C)u^2 - \alpha D(u^2-1) : D(u^2-1))$
10: $(W_2 : Z_2) \leftarrow (4t' : s)$
11: $(W_3 : Z_3) \leftarrow 2(W_2 : Z_2)$ (doubling on $E_{D/C}$)
12: $(W_2' : Z_2') \leftarrow ((-u^2)t' : (W_1' - Z_1')^2(W_1' + Z_1')^2)$
13: $(W_3' : Z_3') \leftarrow 2(W_2' : Z_2')$ (doubling on $E_{D/C}$)
14: $\zeta \leftarrow \mathsf{Legendre\_symbol}(t', p)$
15: $\epsilon' \leftarrow \mathsf{isequal}(\zeta, -1)$
16: $\mathsf{cswap}((W_3 : Z_3), (W_3' : Z_3'), \epsilon')$
17: **return** $(W_3 : Z_3), (W_3' : Z_3')$

7.2 Computational costs of Elligator on Edwards curves

In this subsection, we discuss the difference of the computational costs of our new proposed technique (Algorithm 5) and Elligator on Montgomery curves (Algorithm 2). Here, we consider Elligator used in constant-time CSIDH algorithms that use the two torsion method [27, 9].

The first difference between these algorithms appears when computing t and t'. Our proposal requires $1\mathbf{M} + 2\mathbf{S} + 3\mathbf{a}$ more cost than that on Montgomery curves. Moreover, we needs to compute an extra variable s in the line 7 in the algorithm of Elligator on Edwards curves. Its cost is $2\mathbf{M} + 2\mathbf{S} + 3\mathbf{a}$.

The second difference appears when computing multiplications of P and Q by 4. Algorithm 2 outputs $\mathbf{x}(P)$ and $\mathbf{x}(Q)$; however, when we use these points in the CSIDH algorithm, we needs to compute multiplications of these points by 4. Therefore, we compare the proposed Elligator algorithm with the algorithm on Montgomery curves including computations of multiplications of P and Q by 4. Note that the both cost of doubling formulas on Edwards curves and on Montgomery curves are $4\mathbf{M} + 2\mathbf{S} + 4\mathbf{a}$. The cost of our proposed algorithm is $10\mathbf{M} + 5\mathbf{S} + 13\mathbf{a}$. On the other hand, the cost of that on Montgomery curves is $16\mathbf{M} + 8\mathbf{S} + 16\mathbf{a}$.

Consequently, our proposed algorithm saves the cost of $3\mathbf{M} - 1\mathbf{S} - 3\mathbf{a}$. Therefore, our proposal is more efficient than Elligator on Montgomery curves. Since the impact of Elligator on the whole CSIDH algorithm is small, the constant-time CSIDH algorithm on Edwards curves is as fast as (or a little bit faster than) that on Montgomery curves.

8 √élu's formulas on Edwards curves

In this section, we give the √élu's formulas on Edwards curves. The rough computing process of these formulas is in Appendix A. 4. This method is similar to that on Montgomery curves (in Appendix A. 3).

In our analysis, we can use lower degree polynomials for computing √élu's formulas on Edwards curves than those on Montgomery curves. Hence, the computational cost of computing those on Edwards curves is a little bit smaller than those on Montgomery curves.

8.1 Formulas

Let P be a point of E_d, and let $(W : Z) = \mathbf{w}(P)$. Let $D/C = d$. Let Q be an order-ℓ point of E_d, and $(W_1 : Z_1) = \mathbf{w}(Q)$. Let $(W_k : Z_k) = \mathbf{w}(kQ)$. Let $E_{d'} = E_d/\langle Q\rangle$, and let ϕ be an isogeny $\phi\colon E_d \to E_{d'}$ with $\ker\phi = \langle Q\rangle$. If we let $(W' : Z')$ be the projective w-coordinate of $\phi(P)$, and let $D'/C' = d'$, then the following equations hold (equations (14) and (15)).

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

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

Define the polynomial $h_S \in \mathbb{F}_p[T_1, T_2]$ as $h_S(T_1, T_2) := \prod_{i\in S}(Z_iT_1 - W_iT_2)$. Then, these equations can be rewritten as follows:

$$W' = W \cdot (h_S(W, Z))^2, \quad Z' = Z \cdot (h_S(Z, W))^2. \tag{16}$$

$$D' = D^\ell \cdot (h_S(-1, 1))^8, \quad C' = C^\ell \cdot \left(2^s h_S(1, 0)\right)^8. \tag{17}$$

Here, S is a set $\{1, 3, \ldots, \ell - 2\}$. By using resultants, we can compute $h_S(\alpha, \beta)$ for some α, β in $\tilde{O}(\sqrt{\ell})$ times.

Now, we explain the method to compute h_S on Edwards curves using resultants.

As in [5], let $I = \{2b(2i + 1) \mid 0 \le i < b'\}$, let $J = \{1, 3, \ldots, 2b - 1\}$, and let $K = S \setminus (I \pm J)$, where $b = \lfloor\sqrt{\ell - 1}/2\rfloor$, and $b' = \lfloor(\ell - 1)/4b\rfloor$ (for $b > 0$). Define polynomials F_0, F_1, and F_2 in $\mathbb{F}_p[T_1, T_2, T_3, T_4]$ such that

$$(T - w(P + Q))(T - w(P - Q)) = T^2 + \frac{F_1(\mathbf{w}(P), \mathbf{w}(Q))}{F_0(\mathbf{w}(P), \mathbf{w}(Q))}T + \frac{F_2(\mathbf{w}(P), \mathbf{w}(Q))}{F_0(\mathbf{w}(P), \mathbf{w}(Q))}.$$

In other words,

$$\begin{aligned}
F_0(T_1,T_2,T_3,T_4) &= D(T_1T_3 - T_2T_4)^2,\\
F_1(T_1,T_2,T_3,T_4) &= -2(D(T_1T_3 + T_2T_4)(T_1T_4 + T_2T_3) + (4D - 8C)T_1T_2T_3T_4),\\
F_2(T_1,T_2,T_3,T_4) &= D(T_1T_4 - T_2T_3)^2.
\end{aligned}$$

Then, it holds that,

$$h_S(\alpha,\beta) = \left(\prod_{i\in(I\pm J)} Z_i\right) \cdot \frac{h_K(\alpha,\beta)}{\Delta_{I,J}} \cdot \mathrm{Res}_T(h_I(T,1), E_J(\alpha,\beta,T)),$$

where $\mathrm{Res}_T(\cdot,\cdot)$ is the resultant of two polynomials in T,

$$\Delta_{I,J} = \mathrm{Res}_T(h_I(T,1), \prod_{j\in J} F_0(T,1,W_j,Z_j)),$$

and

$$\begin{aligned}
&E_J(T_1,T_2,T)\\
:=& \prod_{j\in J}(F_0(T,1,W_j,Z_j)T_1^2 + F_1(T,1,W_j,Z_j)T_1T_2 + F_2(T,1,W_j,Z_j)T_2^2)\\
=& \prod_{j\in J}(F_0(T_1,T_2,W_j,Z_j)T^2 + F_1(T_1,T_2,W_j,Z_j)T + F_2(T_1,T_2,W_j,Z_j)).
\end{aligned}$$

Therefore, by using resultants, we can compute the equations (16) and (17). Denote $h_K(\alpha,\beta)\cdot\mathrm{Res}_T(h_I(T,1),E_J(\alpha,\beta,T))$ by $\tilde{h}_S(\alpha,\beta)$. Since $\left(\prod_{i\in(I\pm J)} Z_i\right)$ and $\Delta_{I,J}$ do not depend on α and β, it is enough to consider $\tilde{h}_S(\alpha,\beta)$ instead of $h_S(\alpha,\beta)$ to compute these equations. Furthermore, it holds that

$$\begin{aligned}
\tilde{h}_S(1,0) &= h_K(1,0)\cdot\mathrm{Res}_T(h_I(T,1), \prod_{j\in J} F_0(T,1,W_j,Z_j))\\
&= h_K(1,0)\cdot\mathrm{Res}_T(h_I(T,1), \prod_{j\in J} D(W_jT - Z_j)^2)\\
&= h_K(1,0)\cdot D^{\#I\#J}\cdot\left(\mathrm{Res}_T(h_I(T,1),h_J(1,T))\right)^2.
\end{aligned}$$

Denote $h_K(1,0)\cdot\left(\mathrm{Res}_T(h_I(T,1),h_J(1,T))\right)^2$ by $\tilde{\tilde{h}}_S(1,0)$.

From above discussions, we get the new formulas for computing isogenies on Edwards curves as follows:

$$\begin{aligned}
W' &= W\cdot(h_K(W,Z)\cdot\mathrm{Res}_T(h_I(T,1),E_J(W,Z,T))^2,\\
Z' &= Z\cdot(h_K(Z,W)\cdot\mathrm{Res}_T(h_I(T,1),E_J(Z,W,T))^2,
\end{aligned}$$

$$\begin{aligned}
D' &= D^{2\#K+1}\cdot(h_K(-1,1)\cdot\mathrm{Res}_T(h_I(T,1),E_J(-1,1,T))^8,\\
C' &= C^{\ell}\cdot D^{4\#I\#J}\cdot(2^s h_K(1,0)\cdot(\mathrm{Res}_T(h_I(T,1),h_J(1,T))^2)^8.
\end{aligned}$$

Remark 3. There are some methods to compute resultants. Since h_I is not a monic polynomial, some methods give the value of a resultant multiplied by a constant value determined by the degree of $E_J(\alpha, \beta, T)$ (*e.g.,* the remainder-tree algorithm [15, §2. Method C] with using pseudo division, the scaled remainder-tree algorithm [3] with using pseudo reciprocal). The following problems may occur when using such a method. Although we compute projective coordinates, the constant value affects the final computational result of Edwards coefficients. It is because the degree of $h_J(1, T)$ is different from that of $E_J(-1, 1, T)$, and the constant value multiplied by D' is not same as the constant value multiplied by C'.

If we know these constant values, this problem is easily solved. This is the case if we use the scaled remainder-tree algorithm. If we do not know, we can avoid this situation by doing the following. First, we divide $E_J(-1, 1, T)$ into degree $2\lfloor \#J/2 \rfloor$ and degree $2\lceil \#J/2 \rceil$ polynomials. Next, by adding terms with zero coefficients to these polynomials and $h_J(1, T)$, we set degrees of these two polynomials to $2\lceil \#J/2 \rceil$. Finally, we compute resultants, respectively. In this way, we can cancel out the effect of the constant values.

8.2 Analysis of the formulas

In this subsection, we explain the difference of two √élu's formulas and analyze the efficiency of our proposed formulas. Here, we use the techniques proposed in [1] for Montgomery curves.

In [5], they use the scaled remainder-tree algorithm [3] to compute resultants. It is the improved version of the remainder-tree algorithm [15]. Therefore, we choose the scaled remainder-tree algorithm in our analysis.

The outline of these formulas are same; however, there are small differences that affect their efficiency. In particular, the following difference is important in efficiency.

The main significant difference is whether we compute $\tilde{\tilde{h}}_S(1, 0)$ (on Edwards curves) or $\tilde{h}_S(1, 1)$ (on Montgomery curves). In order to compute $\tilde{h}_S(1, 1)$, it is need to compute the product of $\#J$ polynomials of degree 2, and use the scaled remainder-tree algorithm for the result polynomial of degree $2\#J$. On the other hand, to compute $\tilde{\tilde{h}}_S(1, 0)$, it is need to compute the product of $\#J$ polynomials of degree 1, and use the scaled remainder-tree algorithm for the result polynomial of degree almost $\#J$. Therefore, for computing $\tilde{\tilde{h}}_S(1, 0)$, we use lower degree polynomials than those for computing $\tilde{h}_S(1, 1)$. It shows that the computational cost of computing $\tilde{\tilde{h}}_S(1, 0)$ is a little bit smaller than that of computing $\tilde{h}_S(1, 1)$.

Moreover, as we use the scaled remainder-tree algorithm, we need to care the problem in Remark 3. We denote $\prod_{i \in I} Z_i$ by $\tilde{Z}$. First, we compute the Laurent series of $1/h_I$ in the variable T^{-1}. In the natural method, we need to compute a division of $\tilde{Z}$. To avoid this division, we consider a pseudo reciprocal. By this computation, we get Ω/h_I instead of $1/h_I$, where Ω is a constant value determined by h_I and the degree of the other polynomial of input. The value Ω can be easily computed by considering the Laurent series of Ω/h_I and $\tilde{Z}$. Next, we compute $\#I$ values. By multiplying all these $\#I$ values together, we get (pseudo) resultants. Here, the i-th value is the value multiplied by the conventional value and $1/Z_i$ (and Ω). Therefore, the constant value in Remark 3 is $\Omega^{\#I}/\tilde{Z}$. Thus, we get constant values that come up when computing $\mathrm{Res}_T(h_I(T,1), E_J(-1,1,T))$ and $\mathrm{Res}_T(h_I(T,1), h_J(1,T))$, respectively. Multiplying these values properly yields the correct result. This calculation does not occur in the case of Montgomery curves; however, the impact of this computation is smaller than that of the calculations in the above paragraphs.

Since the other differences have a small impact for their efficiency, we conclude $\sqrt{\text{é}}$lu's formulas on Edwards curves are more efficient than those on Montgomery curves.

9 Conclusion and future work

9.1 Conclusion

We proved four important theorems (Theorem 4, Theorem 5, Theorem 6, and Theorem 7) on Edwards curves and used them to construct a CSIDH algorithm on Edwards curves with w-coordinates. Theorem 4 shows that if $w(P)$ and $w(2P)$ are square, then $w(2P)$ can be treated as a point in $E_d[\pi_p + 1]$, and if $w(P)$ is square and $w(2P)$ is not square, then $1/w(2P)$ can be treated as a point in $E_d[\pi_p - 1]$. Theorem 5 claims that the number of $w(P)$ such that $w(P)$ and $w(2P)$ are square is equal to the number of $w(P)$ such that $w(P)$ is square and $w(2P)$ is not square. Theorem 6 shows the probability that $w\left(\frac{p+1}{4\ell_i}2P\right)$ represents a point of order ℓ_i is almost $1 - \frac{1}{\ell_i}$. Theorem 7 proves that an Edwards coefficient d is unique up to $\mathbb{F}_p$-isomorphism. From these four theorems, we extended the CSIDH algorithm to that on Edwards curves with w-coordinates over $\mathbb{F}_p$.

We compared complexities of the our proposed algorithm and the algorithm proposed by Meyer and Reith. We showed that our proposed algorithm is as fast as (or a little bit faster than) the one of Meyer and Reith. Moreover, we implemented three different CSIDH algorithms (the algorithm on Montgomery curves [21], that on Edwards curves with y-coordinates, and that on Edwards curves with w-coordinates), and compared computational costs of them. There was no big difference of computational costs among the three different algorithms. The algorithm on Edwards curves with w-coordinates was the little bit fastest one in our implementation.

Moreover, we construct Elligator on Edwards curves, which contributes the efficiency of the constant-time CSIDH algorithm on Edwards curves. Theoretically, our proposed constant-time CSIDH algorithm is as fast as (or a little bit faster than) that on Montgomery curves.

Furthermore, we proposed the new $\sqrt{\text{é}}$lu's formulas on Edwards curves. Those on Edwards curves were a little bit faster than those on Montgomery curves.

Acknowlegements. This work was supported by JST CREST Grant Number JPMJCR14D6, Japan.

Appendix Appendix A How to compute the calculations and isogenies

Here, we explain how to compute the calculations and isogenies on Montgomery curves and Edwards curves.

A. 1 Montgomery curves

The doublings formula (1) can be computed as

$$t_1 \leftarrow X+Z,\quad t_2 \leftarrow X-Z,\quad t_1 \leftarrow t_1^2,\quad t_2 \leftarrow t_2^2,\quad s \leftarrow t_1 - t_2,\quad t_2 \leftarrow t_2 \cdot (4C),$$

$$X' \leftarrow t_1 \cdot t_2,\quad t_1 \leftarrow (A+2C)\cdot s,\quad t_1 \leftarrow t_1 + t_2,\quad Z' \leftarrow s \cdot t_1.$$

If $Z = 1$, the doublings formula (1) can be computed as

$$t_1 \leftarrow X+1,\quad t_1 \leftarrow t_1^2,\quad s \leftarrow 2\cdot X,\quad s \leftarrow 2\cdot s,\quad t_2 \leftarrow t_1 - s,\quad t_2 \leftarrow t_2 \cdot (4C),$$

$$X' \leftarrow t_1 \cdot t_2,\quad t_1 \leftarrow (A+2C)\cdot s,\quad t_1 \leftarrow t_1 + t_2,\quad Z' \leftarrow s \cdot t_1.$$

The addition formula (2) can be computed as

$$t_1 \leftarrow X_1 + Z_1,\quad s_1 \leftarrow X_2 + Z_2,\quad t_2 \leftarrow X_1 - Z_1,\quad s_2 \leftarrow X_2 - Z_2,\quad t \leftarrow t_1 \cdot s_2,$$

$$s \leftarrow t_2 \cdot s_1,\quad X_3 \leftarrow t+s,\quad Z_3 \leftarrow t-s,\quad X_3 \leftarrow X_3^2 \cdot Z_0,\quad Z_3 \leftarrow Z_3^2 \cdot X_0.$$

The formula for calculating $\phi(P)$ (3) can be computed as

$$t_i \leftarrow X_i + Z_i,\quad s_i \leftarrow X_i - Z_i,\quad t_i \leftarrow t_i \cdot (X-Z),\quad s_i \leftarrow s_i \cdot (X+Z),$$

$$X' \leftarrow \prod_{i=1}^{s}(t_i - s_i),\quad Z' \leftarrow \prod_{i=1}^{s}(t_i + s_i),\quad X' \leftarrow X \cdot (X')^2,\quad Z' \leftarrow Z \cdot (Z')^2.$$

The formula for calculating E' (4) can be computed as

$$c \leftarrow 2\cdot C,\quad a \leftarrow A + c,\quad d \leftarrow A - c,\quad a' \leftarrow \prod_{i=1}^{s}(X_i + Z_i),$$

$$d' \leftarrow \prod_{i=1}^{s}(X_i - Z_i),\quad a' \leftarrow (a')^4,\quad d' \leftarrow (d')^4,\quad a' \leftarrow a^s \cdot a',\quad d' \leftarrow d^s \cdot d',$$

$$a' \leftarrow a \cdot (a')^2,\quad d' \leftarrow d \cdot (d')^2,\quad A' \leftarrow 2\cdot(a'+d'),\quad C' \leftarrow a' - d'.$$

A. 2 Edwards curves

The doublings formula (8) can be computed as

$$t_1 \leftarrow Y^2, \quad t_2 \leftarrow Z^2, \quad t_3 \leftarrow C - D, \quad t_4 \leftarrow t_2 - t_1, \quad t_1 \leftarrow t_3 \cdot t_1, \quad t_5 \leftarrow C \cdot t_4,$$

$$t_6 \leftarrow t_1 + t_5, \quad t_6 \leftarrow t_4 \cdot t_6, \quad t_1 \leftarrow t_1 \cdot t_2, \quad Y' \leftarrow t_1 - t_6, \quad Z' \leftarrow t_1 + t_6.$$

If $Z = 1$, the doublings formula (8) can be computed as

$$t_1 \leftarrow Y^2, \quad t_3 \leftarrow C - D, \quad t_4 \leftarrow 1 - t_1, \quad t_1 \leftarrow t_3 \cdot t_1, \quad t_5 \leftarrow C \cdot t_4,$$

$$t_6 \leftarrow t_1 + t_5, \quad t_6 \leftarrow t_4 \cdot t_6, \quad Y' \leftarrow t_1 - t_6, \quad Z' \leftarrow t_1 + t_6.$$

The addition formula (9) can be computed as

$$t_1 \leftarrow Y_1 \cdot Z_2, \quad t_2 \leftarrow Y_2 \cdot Z_1, \quad s_1 \leftarrow t_1 + t_2, \quad s_2 \leftarrow t_1 - t_2, \quad s_1 \leftarrow s_1^2, \quad s_2 \leftarrow s_2^2,$$

$$s_1 \leftarrow (Z_0 - Y_0) \cdot s_1, \quad s_2 \leftarrow (Z_0 + Y_0) \cdot s_2, \quad Y_3 \leftarrow s_1 - s_2, \quad Z_3 \leftarrow s_1 + s_2.$$

The formula for calculating $\phi(P)$ (10) can be computed as

$$t_i \leftarrow Z \cdot Y_i, \quad t_i' \leftarrow Z_i \cdot Y, \quad s_1 \leftarrow \prod_{i=1}^{s} (t_i + t_i'), \quad s_2 \leftarrow \prod_{i=1}^{s} (t_i - t_i'), \quad s_1 \leftarrow s_1^2,$$

$$s_2 \leftarrow s_2^2, \quad s_1 \leftarrow (Z + Y) \cdot s_1, \quad s_2 \leftarrow (Z - Y) \cdot s_2, \quad Y' \leftarrow s_1 - s_2, \quad Z' \leftarrow s_1 + s_2.$$

The formula for calculating E' (11) can be computed as

$$D' \leftarrow \prod_{i=1}^{s} Y_i, \quad C' \leftarrow \prod_{i=1}^{s} Z_i, \quad D' \leftarrow (D')^4, \quad C' \leftarrow (C')^4,$$

$$D' \leftarrow D^s \cdot D', \quad C' \leftarrow C^s \cdot C', \quad D' \leftarrow D \cdot (D')^2, \quad C' \leftarrow C \cdot (C')^2.$$

The doublings formula (12), addition formula (13), and formula for calculating $\phi(P)$ (14) can be computed similarly as the formulas on Montgomery curves.

The formula for calculating E' (15) can be computed as

$$D' \leftarrow \prod_{i=1}^{s} (W_i + Z_i), \quad C' \leftarrow \prod_{i=1}^{s} Z_i, \quad D' \leftarrow (D')^4, \quad C' \leftarrow (C')^4,$$

$$D' \leftarrow D^s \cdot D', \quad C' \leftarrow (2 \cdot 2 \cdot 2 \cdot 2 \cdot C)^s \cdot C', \quad D' \leftarrow D \cdot (D')^2, \quad C' \leftarrow C \cdot (C')^2.$$

A. 3 Calculations of $\sqrt{\text{é}}$lu's formulas on Montgomery curves

In this subsection, we explain the method to compute ℓ-isogenies proposed in [5]. Although in [5], they wrote down the formulas using affine coordinates, we consider the formulas using projective coordinates to estimate their computational costs.

Let E be a Montgomery curve $y^2 = x^3 + ax^2 + x$, and let $A/C = a$. Let $G = \{P_i \mid i = 1, \ldots, \ell\}$ be a finite subgroup of E, and let $\phi\colon E \to E/\langle G \rangle$ be an isogeny satisfying $\ker \phi = G$. Denote $\mathbf{x}(P_i)$ by (X_i, Z_i). Let $h_{\mathrm{Set}}(T_1, T_2)$ be a polynomial defined by $h_{\mathrm{Set}}(T_1, T_2) := \prod_{i \in \mathrm{Set}} (Z_i T_1 - X_i T_2)$. Let $S = \{1, 3, \ldots, \ell - 2\}$, let $I = \{2b(2i + 1) \mid 0 \le i < b'\}$, let $J = \{1, 3, \ldots, 2b - 1\}$, and let $K = S \setminus (I \pm J)$, where $b = \lfloor \sqrt{\ell - 1}/2 \rfloor$, and $b' = \lfloor (\ell - 1)/4b \rfloor$ (for $b > 0$). Define polynomials F_0, F_1, and F_2 in $\mathbb{F}_p[T_1, T_2, T_3, T_4]$ such that

$$(T - x(P + Q))(T - x(P - Q)) = T^2 + \frac{F_1(\mathbf{x}(P), \mathbf{x}(Q))}{F_0(\mathbf{x}(P), \mathbf{x}(Q))} T + \frac{F_2(\mathbf{x}(P), \mathbf{x}(Q))}{F_0(\mathbf{x}(P), \mathbf{x}(Q))}.$$

In other words,

$$\begin{aligned}
F_0(T_1, T_2, T_3, T_4) &= C(T_1 T_4 - T_2 T_3)^2, \\
F_1(T_1, T_2, T_3, T_4) &= -2(C(T_1 T_3 + T_2 T_4)(T_1 T_4 + T_2 T_3) + 2A T_1 T_2 T_3 T_4), \\
F_2(T_1, T_2, T_3, T_4) &= C(T_1 T_3 - T_2 T_4)^2.
\end{aligned}$$

Note that $\#S = 2\#I\#J + \#K$. From [5, Theorem 4.11], it holds that,

$$h_S(\alpha, \beta) = \left(\prod_{i \in (I \pm J)} Z_i \right) \cdot \frac{h_K(\alpha, \beta)}{\Delta_{I,J}} \cdot \mathrm{Res}_T(h_I(T, 1), E_J(\alpha, \beta, T)),$$

where $\mathrm{Res}_T(f, g)$ is the resultant of polynomials f and g in $\mathbb{F}_p[T]$, $\Delta_{I,J}$ is $\mathrm{Res}_T(h_I(T, 1), \prod_{j \in J} F_0(T, 1, X_j, Z_j))$, and

$$\begin{aligned}
&E_J(T_1, T_2, T) \\
&:= \prod_{j \in J} (F_0(T, 1, X_j, Z_j) T_1^2 + F_1(T, 1, X_j, Z_j) T_1 T_2 + F_2(T, 1, X_j, Z_j) T_2^2) \\
&= \prod_{j \in J} (F_0(T_1, T_2, X_j, Z_j) T^2 + F_1(T_1, T_2, X_j, Z_j) T + F_2(T_1, T_2, X_j, Z_j)).
\end{aligned}$$

Therefore, by using resultants, we can compute the equations (5) and (6). Denote $h_K(\alpha, \beta) \cdot \mathrm{Res}_T(h_I(T, 1), E_J(\alpha, \beta, T))$ by $\tilde{h}_S(\alpha, \beta)$. Since $\left(\prod_{i \in (I \pm J)} Z_i \right)$ and $\Delta_{I,J}$ do not depend on α and β, it is enough to consider $\tilde{h}_S(\alpha, \beta)$ instead of $h_S(\alpha, \beta)$ to compute these formulas.

We use the scaled remainder-tree algorithm to compute resultants. First, we generate a product tree of polynomials $\{Z_i T - X_i \mid i \in I\}$. Next, by using the scaled remainder-tree algorithm for $E_J(\alpha, \beta, T)$, we compute $\{E_J(\alpha, \beta, X_i/Z_i) \mid i \in I\}$. Finally, we multiply all these values together to get the result.

The formula for calculating $\phi(P)$ (5) can be computed as

$$t_{1,j} \leftarrow X_j X, \quad t_{2,j} \leftarrow X_j Z, \quad t_{3,j} \leftarrow Z_j X, \quad t_{4,j} \leftarrow Z_j Z,$$
$$t_{5,j} \leftarrow 2At_{1,j}t_{4,j}, \quad t_{6,j} \leftarrow (t_{1,j} + t_{4,j})(t_{2,j} + t_{3,j}),$$
$$h_{1,j} \leftarrow (-2)(t_{6,j} \cdot C + t_{5,j}), \quad h_{0,j} \leftarrow C(t_{1,j} - t_{4,j})^2, \quad h_{2,j} \leftarrow C(t_{2,j} - t_{3,j})^2,$$
$$E_J(X, Z, T) \leftarrow \prod_{j \in J}(h_{2,j}T^2 + h_{1,j}T + h_{0,j}),$$
$$\mathrm{Set}_{(X,Z)} \leftarrow \mathsf{Resultant}_T(E_J(X, Z, T), \{Z_i T - X_i \mid i \in I\}),$$
$$\tilde{h}_S(X, Z) \leftarrow \left(\prod_{v \in \mathrm{Set}_{(X,Z)}} v\right) \cdot \left(\prod_{k \in K}(Z_k X - X_k Z)\right),$$
$$E_J(Z, X, T) \leftarrow \prod_{j \in J}(h_{0,j}T^2 + h_{1,j}T + h_{2,j}) = \mathsf{Reverse}(E_J(X, Z, T)),$$
$$\mathrm{Set}_{(Z,X)} \leftarrow \mathsf{Resultant}_T(E_J(Z, X, T), \{Z_i T - X_i \mid i \in I\}),$$
$$\tilde{h}_S(Z, X) \leftarrow \left(\prod_{v \in \mathrm{Set}_{(Z,X)}} v\right) \cdot \left(\prod_{k \in K}(Z_k Z - X_k X)\right),$$
$$X' \leftarrow \tilde{h}_S(Z, X), \quad Z' \leftarrow \tilde{h}_S(X, Z), \quad X' \leftarrow X \cdot (X')^2, \quad Z' \leftarrow Z \cdot (Z')^2.$$

The formula for calculating E' (6) can be computed as

$$t_{1,j} \leftarrow (X_j + Z_j)^2, \quad t_{2,j} \leftarrow (X_j - Z_j)^2, \quad t_{3,j} \leftarrow C \cdot t_{1,j}, \quad t_{4,j} \leftarrow C \cdot t_{2,j},$$
$$t_{5,j} \leftarrow A(t_{2,j} - t_{1,j}), \quad h_{+,j} \leftarrow t_{4,j}, \quad h_{-,j} \leftarrow t_{3,j},$$
$$h_{+,1,j} \leftarrow t_{5,j} - 2h_{-,j}, \quad h_{-,1,j} \leftarrow 2h_{+,j} - t_{5,j},$$
$$E_J(1, 1, T) \leftarrow \prod_{j \in J}(h_{+,j}T^2 + h_{+,1,j}T + h_{+,j}),$$
$$E_J(-1, 1, T) \leftarrow \prod_{j \in J}(h_{-,j}T^2 + h_{-,1,j}T + h_{-,j}),$$
$$\mathrm{Set}_+ \leftarrow \mathsf{Resultant}_T(E_J(1, 1, T), \{Z_i T - X_i \mid i \in I\}),$$
$$\mathrm{Set}_- \leftarrow \mathsf{Resultant}_T(E_J(-1, 1, T), \{Z_i T - X_i \mid i \in I\}),$$
$$\tilde{h}_S(1, 1) \leftarrow \left(\prod_{v \in \mathrm{Set}_+} v\right) \cdot \left(\prod_{k \in K}(Z_k - X_k)\right),$$
$$\tilde{h}_S(-1, 1) \leftarrow \left(\prod_{v \in \mathrm{Set}_-} v\right) \cdot \left(\prod_{k \in K}(-Z_k - X_k)\right),$$
$$c \leftarrow 2 \cdot C, \quad a \leftarrow A + c, \quad d \leftarrow A - c, \quad a' \leftarrow \tilde{h}_S(-1, 1)^2, \quad d' \leftarrow \tilde{h}_S(1, 1)^2,$$
$$a' \leftarrow a^{\#I\#J} \cdot a', \quad d' \leftarrow d^{\#I\#J} \cdot d', \quad a' \leftarrow (a')^2, \quad d' \leftarrow (d')^2,$$
$$a' \leftarrow a^{\#K} \cdot a', \quad d' \leftarrow d^{\#K} \cdot d', \quad a' \leftarrow a \cdot (a')^2, \quad d' \leftarrow d \cdot (d')^2,$$
$$A' \leftarrow 2 \cdot (a' + d'), \quad C' \leftarrow a' - d'.$$

A. 4 Calculations of √élu's formulas on Edwards curves

The formula for calculating $\phi(P)$ (16) can be computed as

$$t_{DC} \leftarrow 2(D-2C), \quad t_{1,j} \leftarrow W_jW, \quad t_{2,j} \leftarrow W_jZ, \quad t_{3,j} \leftarrow Z_jW,$$
$$t_{4,j} \leftarrow Z_jZ, \quad t_{5,j} \leftarrow 2t_{DC}t_{1,j}t_{4,j}, \quad t_{6,j} \leftarrow (t_{1,j}+t_{4,j})(t_{2,j}+t_{3,j}),$$
$$h_{1,j} \leftarrow (-2)(t_{6,j}\cdot D + t_{5,j}), \quad h_{0,j} \leftarrow D(t_{2,j}-t_{3,j})^2, \quad h_{2,j} \leftarrow D(t_{1,j}-t_{4,j})^2,$$
$$E_J(W,Z,T) \leftarrow \prod_{j\in J}(h_{2,j}T^2 + h_{1,j}T + h_{0,j}),$$
$$\mathrm{Set}_{(W,Z)} \leftarrow \mathsf{Resultant}_T(E_J(W,Z,T), \{Z_iT - W_i \mid i \in I\}),$$
$$\tilde{h}_S(W,Z) \leftarrow \left(\prod_{v\in \mathrm{Set}_{(W,Z)}} v\right)\cdot\left(\prod_{k\in K}(Z_kW - W_kZ)\right),$$
$$E_J(Z,W,T) \leftarrow \prod_{j\in J}(h_{0,j}T^2 + h_{1,j}T + h_{2,j}) = \mathsf{Reverse}(E_J(W,Z,T)),$$
$$\mathrm{Set}_{(Z,W)} \leftarrow \mathsf{Resultant}_T(E_J(Z,W,T), \{Z_iT - W_i \mid i \in I\}),$$
$$\tilde{h}_S(Z,W) \leftarrow \left(\prod_{v\in \mathrm{Set}_{(Z,W)}} v\right)\cdot\left(\prod_{k\in K}(Z_kZ - W_kW)\right),$$
$$W' \leftarrow \tilde{h}_S(W,Z), \quad Z' \leftarrow \tilde{h}_S(Z,W), \quad W' \leftarrow W\cdot(W')^2, \quad Z' \leftarrow Z\cdot(Z')^2.$$

The formula for calculating E' (17) can be computed as

$$t_{1,j} \leftarrow (W_j+Z_j)^2, \quad t_{2,j} \leftarrow (W_j-Z_j)^2, \quad t_{3,j} \leftarrow D\cdot t_{1,j}, \quad t_{4,j} \leftarrow D\cdot t_{2,j},$$
$$t_{5,j} \leftarrow t_{1,j}-t_{2,j}, \quad h_{0,j} \leftarrow t_{3,j}, \quad h_{1,j} \leftarrow t_{DC}\cdot t_{5,j} + 2t_{4,j},$$
$$E_J(-1,1,T) \leftarrow \prod_{j\in J}(h_{0,j}T^2 + h_{1,j}T + h_{0,j}),$$
$$\mathrm{Set}_{(-1,1)} \leftarrow \mathsf{Resultant}_T(E_J(-1,1,T), \{Z_iT - W_i \mid i \in I\}),$$
$$\tilde{h}_S(-1,1) \leftarrow \left(\prod_{v\in \mathrm{Set}_{(-1,1)}} v\right)\cdot\left(\prod_{k\in K}(-Z_k - W_k)\right),$$
$$\mathrm{Set}_{(1,0)} \leftarrow \mathsf{Resultant}_T\left(\prod_{j\in J}(W_jT - Z_j), \{Z_iT - W_i \mid i \in I\}\right),$$
$$\tilde{\tilde{h}}_S(1,0) \leftarrow \left(\prod_{v\in \mathrm{Set}_{(1,0)}} v\right)^2\cdot\left(\prod_{k\in K} Z_k\right),$$
$$D' \leftarrow \tilde{h}_S(-1,1), \quad C' \leftarrow \tilde{\tilde{h}}_S(1,0), \quad D' \leftarrow (D')^4, \quad C' \leftarrow (C')^2,$$
$$C'' \leftarrow 2\cdot 2\cdot 2\cdot 2\cdot C, \quad C' \leftarrow (D\cdot C'')^{\#I\#J}\cdot C', \quad C' \leftarrow (C')^2,$$
$$D' \leftarrow D^{\#K}\cdot D', \quad C' \leftarrow (C'')^{\#K}\cdot C', \quad D' \leftarrow D\cdot(D')^2, \quad C' \leftarrow C\cdot(C')^2.$$

Remark 4. The above formulas do not care about the problem explained in Remark 3. If we want to do the actual calculation, we need to do some additional calculations about the constant values in Remark 3.

Appendix Appendix B Proof of Theorem 1

In this section, we prove Theorem 1.

Theorem 1. *Let p be a prime and $p \geq 3$. The Edwards curve E_d defined over $\mathbb{F}_p$ is $\mathbb{F}_p$-isomorphic to the Montgomery curve,*

$$E_{\mathcal{M}}\colon \frac{4}{1-d}Y^2Z = X^3 + \frac{2(1+d)}{1-d}X^2Z + XZ^2.$$

Proof. Define a rational map $\psi\colon E_d \to E_{\mathcal{M}}$,

$$(X:Y:Z:T) \mapsto (X+T : Y+Z : X-T).$$

ψ is a morphism, because all points except for $(0:-1:1:0)$ are simply regular, and from the following equation, $(0:-1:1:0)$ is also regular.

$$\frac{(Z-Y)Z}{X}(X+T, Y+Z, X-T) = (Z^2-Y^2, XZ-dYT, (Z-Y)^2).$$

$\psi(0_d) = (0:1:0)$ holds, so ψ is an isogeny defined over $\mathbb{F}_p$.

Define a rational map $\phi\colon E_{\mathcal{M}} \to E_d$,

$$(X:Y:Z:T) \mapsto (X(X+Z) : Y(X-Z) : Y(X+Z) : X(X-Z)).$$

ϕ is a morphism, because all points except for $(0:1:0)$ and $(0:0:1)$ are simply regular, and from the following two equations, $(0:1:0)$ and $(0:0:1)$ are also regular.

$$\begin{aligned} &\frac{1}{X}(X(X+Z), Y(X-Z), Y(X+Z), X(X-Z)) \\ &= \left(X+Z, \frac{bY^2-X^2-aXZ-Z^2}{bY}, \frac{bY^2+X^2+aXZ+Z^2}{bY}, X-Z\right), \end{aligned}$$

and

$$\begin{aligned} &\frac{1}{Y}(X(X+Z), Y(X-Z), Y(X+Z), X(X-Z)) \\ &= \left(\frac{bYZ(X+Z)}{X^2+aXZ+Z^2}, X-Z, X+Z, \frac{bYZ(X-Z)}{X^2+aXZ+Z^2}\right), \end{aligned}$$

where $b = \dfrac{4}{1-d}$ and $a = \dfrac{2(1+d)}{1-d}$.

$\phi(0:1:0) = 0_d$ holds, so ϕ is an isogeny defined over $\mathbb{F}_p$.

As $\psi \circ \phi = \mathrm{id}_{E_{\mathcal{M}}}$ and $\phi \circ \psi = \mathrm{id}_{E_d}$ hold, E_d is $\mathbb{F}_p$-isomorphic to $E_{\mathcal{M}}$. □

Appendix Appendix C CSIDH on Edwards curves with y-coordinates

In this section, we explain the CSIDH algorithm on Edwards curves with y-coordinates. There is no difference essentially between this algorithm and the original CSIDH algorithm [8]. The precise algorithm is as follows.

Algorithm 6 Evaluating the class group action on Edwards curves with y-coordinates

Input: $d \in \mathbb{F}_p$ such that E_d is supersingular and a list of integers $(e_1, \dots, e_n)$
Output: d' such that $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_d = E_{d'}$
1: **while** some $e_i \neq 0$ **do**
2: Sample a random $y \in \mathbb{F}_p$
3: $\mathbf{y}(P) \leftarrow (y : 1)$
4: Set $s \leftarrow +1$ if $(1-y^2)(1-dy^2)$ is a square in $\mathbb{F}_p$, else $s \leftarrow -1$
5: Let $S = \{i \mid \mathrm{sign}(e_i) = s\}$
6: **if** $S = \emptyset$ **then**
7: Go to line 2
8: **end if**
9: $k \leftarrow \prod_{i \in S} \ell_i$, $\mathbf{y}(P) \leftarrow \mathbf{y}(((p+1)/k)P)$
10: **for all** $i \in S$ **do**
11: $\mathbf{y}(Q) \leftarrow \mathbf{y}((k/\ell_i)P)$
12: **if** $Q \neq 0_d$ $(\mathbf{y}(Q) \neq (1:1))$ **then**
13: Compute an ℓ_i-isogeny $\phi\colon E_d \to E_{d'}$ with $\ker\phi = \langle Q \rangle$
14: $d \leftarrow d'$, $\mathbf{y}(P) \leftarrow \mathbf{y}(\phi(P))$, $k \leftarrow k/\ell_i$, $e_i \leftarrow e_i - s$
15: **end if**
16: **end for**
17: **end while**
18: **return** d (**Theorem 7**)

Sampling points (line 2-8 in Algorithm 6) We take a uniformly random element of $\mathbb{F}_p$. Let the element be y, and P be a point in E_d such that $y(P) = y$. We calculate $(1-y^2)(1-dy^2)$. Here, $\frac{1-y^2}{1-dy^2}$ is a square of $x(P)$, where $x(P)$ is the x-coordinate of P. If $(1-y^2)(1-dy^2)$ is square in $\mathbb{F}_p$, then $P \in \ker(\pi_p - 1)$, and if $(1-y^2)(1-dy^2)$ is not square in $\mathbb{F}_p$, then $P \in \ker(\pi_p + 1)$.

Scalar multiplication (line 9 in Algorithm 6) Next, we calculate $P_1 = \frac{p+1}{k}(P)$, where $k = \prod_{i \in S} \ell_i$. The calculation uses the ladder algorithm which is constructed in the same way as Montgomery curves [23].

Calculation of isogenies (line 10-16 in Algorithm 6) We calculate $P_2 = \frac{k}{\ell_i} P_1$. The order of P_2 is 1 or ℓ_i. The probability that P_2 is not the identity is almost $1 - \frac{1}{\ell_i}$. This fact can be proven in the similar way in [8]. Therefore, with highly probability, we get a point of order ℓ_i. Then, by Theorem 7, we can calculate isogenies by using the same strategy as the original CSIDH algorithm. To do so, we can use the formulas on Edwards curves [24, 9].

Output (line 18 in Algorithm 6) If the list of integers $(e_1, \ldots, e_n)$ is the zero vector, we output the Edwards coefficient $d' \in \mathbb{F}_p$.

References

1. Gora Adj, Jesús-Javier Chi-Domínguez, and Francisco Rodríguez-Henríquez. On new Vélu's formulae and their applications to CSIDH and B-SIDH constant-time implementations. *IACR Cryptology ePrint Archive*, 2020:1109, 2020. https://ia.cr/2020/1109.
2. Reza Azarderakhsh, Matthew Campagna, Craig Costello, LD Feo, Basil Hess, A Jalali, D Jao, B Koziel, B LaMacchia, P Longa, et al. Supersingular isogeny key encapsulation. *Submission to the NIST Post-Quantum Standardization project*, 2017.
3. Daniel J Bernstein. Scaled remainder trees, August 2004. https://cr.yp.to/papers.html#scaledmod.
4. Daniel J Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. In *International Conference on Cryptology in Africa–AFRICACRYPT 2008*, pages 389–405. Springer, 2008.
5. Daniel J Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. In *Proceedings of the Fourteenth Algorithmic Number Theory Symposium–ANTS 2020*, volume 4, pages 39–55. Mathematical Sciences Publishers, 2020.
6. Daniel J Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: elliptic-curve points indistinguishable from uniform random strings. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 967–980, 2013.
7. Daniel J Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *International Conference on the Theory and Application of Cryptology and Information Security–ASIACRYPT 2007*, pages 29–50. Springer, 2007.
8. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security–ASIACRYPT 2018*, pages 395–427. Springer, 2018.
9. Daniel Cervantes-Vázquez, Mathilde Chenu, Jesús-Javier Chi-Domínguez, Luca De Feo, Francisco Rodríguez-Henríquez, and Benjamin Smith. Stronger and faster side-channel protections for CSIDH. In *International Conference on Cryptology and Information Security in Latin America–LATINCRYPT 2019*, pages 173–193. Springer, 2019.
10. Craig Costello and Huseyin Hisil. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security–ASIACRYPT 2017*, pages 303–329. Springer, 2017.
11. Craig Costello and Benjamin Smith. Montgomery curves and their arithmetic: The case of large characteristic fields. *IACR Cryptology ePrint Archive*, 2017:212, 2017. https://ia.cr/2017/212.

12. Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, pages 425–440, 2016.
13. Harold Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, pages 393–422, 2007.
14. Reza Rezaeian Farashahi and Seyed Gholamhossein Hosseini. Differential Addition on Twisted Edwards Curves. In *Australasian Conference on Information Security and Privacy*, pages 366–378. Springer, 2017.
15. Charles M Fiduccia. Polynomial evaluation via the division algorithm the fast fourier transform revisited. In *Proceedings of the fourth annual ACM symposium on Theory of computing*, pages 88–93, 1972.
16. Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. Twisted Edwards curves revisited. In *International Conference on the Theory and Application of Cryptology and Information Security–ASIACRYPT 2008*, pages 326–343. Springer, 2008.
17. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography–PQCrypto 2011*, pages 19–34. Springer, 2011.
18. Suhri Kim, Kisoon Yoon, Young-Ho Park, and Seokhie Hong. Optimized method for computing odd-degree isogenies on Edwards curves. In *Advances in Cryptology–ASIACRYPT 2019*, pages 273–292. Springer, 2019.
19. Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, pages 203–209, 1987.
20. Michael Meyer, Fabio Campos, and Steffen Reith. On Lions and Elligators: An efficient constant-time implementation of CSIDH. In *International Conference on Post–Quantum Cryptography–PQCrypto 2018*, pages 307–325. Springer, 2019.
21. Michael Meyer and Steffen Reith. A faster way to the CSIDH. In *International Conference on Cryptology in India–INDOCRYPT 2018*, pages 137–152. Springer, 2018.
22. Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques–CRYPTO 1985*, pages 417–426. Springer, 1985.
23. Peter L Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of computation*, pages 243–264, 1987.
24. Dustin Moody and Daniel Shumow. Analogues of Vélu's formulas for isogenies on alternate models of elliptic curves. *Mathematics of Computation*, pages 1929–1951, 2016.
25. Tomoki Moriya, Hiroshi Onuki, and Tsuyoshi Takagi. How to construct CSIDH on Edwards curves. In *Cryptographers' Track at the RSA Conference–CT-RSA 2020*, pages 512–537. Springer, 2020.
26. National Institute of Standards and Technology. Post–quantum cryptography standardization, July 2020. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization`.
27. Hiroshi Onuki, Yusuke Aikawa, Tsutomu Yamazaki, and Tsuyoshi Takagi. A faster constant-time algorithm of CSIDH keeping two points. In *Advances in Information and Computer Security–IWSEC 2019*, pages 23–33. Springer, 2019.
28. Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, pages 120–126, 1978.

29. Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.
30. Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, pages 303–332, 1999.
31. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.
32. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris, Séries A*, pages 305–347, 1971.
33. William C Waterhouse. Abelian varieties over finite fields. In *Annales scientifiques de l'École Normale Supérieure*, pages 521–560, 1969.