On the Performance Analysis for CSIDH-Based Cryptosystems

Donghoe Heo 1, Suhri Kim 1, Young-Ho Park 2 and Seokhie Hong 1,*

1 Graduate School of Information Security, Institute of Cyber Security and Privacy (ICSP), Korea University, Seoul 02841, Korea; dong5641@korea.ac.kr (D.H.); suhrikim@gmail.com (S.K.)
2 Department of Information Security, Graduate School of Information Security, Sejong Cyber University, Seoul 05000, Korea; youngho@sjcu.ac.kr
* Correspondence: shhong@korea.ac.kr; Tel.: +82-10-6201-6348

Received: 26 August 2020; Accepted: 29 September 2020; Published: 2 October 2020

Abstract: In this paper, we present the performance and security analysis for various commutative SIDH (CSIDH)-based algorithms. As CSIDH offers a smaller key size than SIDH and provides a relatively efficient signature scheme, numerous CSIDH-based key exchange algorithms have been proposed to optimize the CSIDH. In CSIDH, the private key is an ideal class in a class group, which can be represented by an integer vector. As the number of ideal classes represented by these vectors determines the security level of CSIDH, it is important to analyze whether the different vectors induce the same public key. In this regard, we generalize the existence of a collision for a base prime $p \equiv 7 \mod 8$. Based on our result, we present a new interval for the private key to have a similar security level for the various CSIDH-based algorithms for a fair comparison of the performance. Deduced from the implementation result, we conclude that for a prime $p \equiv 7 \mod 8$, CSIDH on the surface using the Montgomery curves is the most likely to be efficient. For a prime $p \equiv 3 \mod 8$, CSIDH on the floor using the hybrid method with Onuki’s collision-free method is the most likely to be efficient and secure.

Keywords: post-quantum cryptography; isogeny; Montgomery curves; CSIDH; CSURF

1. Introduction

Isogeny-based cryptography was first proposed by Couveignes in 1997 [1] and is constructed using the isogeny classes of ordinary elliptic curves defined over a finite field $\mathbb{F}_p$. The scheme proposed by Couveignes was later rediscovered by Rostovtsev and Stolbunov, which we now typically call as the CRS scheme. While the CRS scheme is attractive for having a small key size, the scheme was extremely inefficient and even suffered from the quantum sub-exponential algorithm proposed by Childs et al. [2]. The isogeny-based cryptosystem began to gain attention after the introduction of the SIDH key exchange by Jao, De Feo, and Plût in 2011 [3]. As SIDH is constructed using the isogenies between supersingular elliptic curves, the cryptosystem resists against the attack proposed in [2], as the endomorphism ring of supersingular curves is non-commutative while the attack in [2] exploits the commutativity of the endomorphism ring of an ordinary curve. Until now, the best known classical and quantum attacks against the underlying problem are both exponential. The Supersingular Isogeny Key Encapsulation (SIKE), based on SIDH, was submitted as one of the candidates to the NIST post-quantum cryptography standardization project [4]. Currently, SIKE is an alternative candidate in Round 3 of the NIST standardization project. However, one of the drawbacks of isogeny-based cryptography is that not only the algorithm is slower than any other post-quantum cryptography algorithms, but it is also hard to design various cryptographic primitives.
The CRS scheme was pointed out again by De Feo, Kieffer, and Smith in [5], and independently by Castryck et al. in [6]. As the CRS scheme offers efficient and safe public key validation, this makes it possible to reuse a key without the need for confirmation. More importantly, CSIDH provides a practical isogeny-based signature scheme called CSI-FiSh [7]. Note that SIDH and CSIDH provide relatively efficient key-exchange schemes. However, constructing an isogeny-based digital signature scheme is much harder to achieve. The first SIDH-based digital signature scheme was by Yoo et al. in [8]. Not only was their scheme inefficient, but the size of the signature is larger than other post-quantum signature schemes. SeaSign, a CSIDH-based digital signature scheme proposed by De Feo and Galbraith, alleviated this problem, although several minutes are still required to sign a message [9]. Later, by computing the class group of an imaginary quadratic field having 154-digit discriminant, CSI-FiSh [7] offers a practical digital signature scheme which requires 390 ms to sign a message. For isogeny-based cryptography, this is a remarkable result, which shed a light that various cryptographic primitives can be constructed through elliptic curve isogenies. As speed is the main drawback of the practical use of CSIDH-based algorithms, many studies focus on developing an efficient algorithm or modifying an algorithm to increase the performance.

In CSIDH, as higher degree isogenies are used, computing the curve coefficient of the image curve is painstaking compared to the 3- and 4- isogenies used in SIDH-based systems. In [10], by exploiting the fact that conversion between Montgomery curves and Edwards curves are efficient, Meyer and Reith proposed a hybrid version of implementation for CSIDH. In [11], they proposed a method to use a 2-torsion point for recovering the coefficient of the image curve for Montgomery curves. In [12], they proposed optimized odd-degree isogeny formula by using the \( w \)-coordinate on Edwards curves. By adapting the formula in [12], faster Edwards-only CSIDH can be implemented. Additionally, there are studies that focus on both security and performance enhancement. In [13], Onuki and Takagi examined that there exist collisions related to an ideal class of order 3 in CSIDH. CSIDH uses the exponents \( (e_1, e_2, \ldots, e_n) \) as a private key. The work of [13] means that \( (e_1, e_2, \ldots, e_n) \) and \( (e_1 + 3, e_2 + 3, \ldots, e_n + 3) \) represent the same ideal class (same private key) so that the size of the private key space is reduced. To eliminate this collision, they present a new ideal representation that exploits an isogeny of degree 4. In [14], they proposed CSIDH on the surface, called CSURF, using supersingular elliptic curve with endomorphism ring \( \mathbb{Z}[1 + \sqrt{-p}] / 2 \). The CSURF algorithm allows the use of 2-isogeny in CSIDH by choosing the prime \( p \) such that \( p \equiv 7 \mod 8 \), unlike the original CSIDH execute on the floor, i.e., using endomorphism ring \( \mathbb{Z}[\sqrt{-p}] \) with the base prime \( p \equiv 3 \mod 8 \). As the computation of the large degree isogenies corresponds to the performance degradation of CSIDH, CSURF proposed a way to use 2-isogenies more and use a fewer number of large degree isogenies at the same security level. To summarize, all of this work focuses on optimizing the performance of CSIDH while enhancing the security level.

In this paper, we analyze the performance and security of the various CSIDH-based algorithms in order to find out what sort of prime \( p \) and which method is most efficient. The following list details the main contribution of this work.
We implement the CSIDH-based algorithms in the same environment for the exact performance comparison. More explicitly, we implement CSURF [13,14] in projective coordinates. The algorithm in [13] has not been previously implemented. For CSURF, the authors presented only Magma-based implementation. We implemented both of the algorithms in C for an exact comparison with CSIDH [6]. The projectivized formula for the building blocks for both of the algorithms is presented in Section 2.

We generalize the existence of a collision for a base prime \( p \) such that \( p \equiv 7 \mod 8 \). As CSIDH-based algorithms use ideal classes expressed by an integer vector as a private key, the number of ideal classes represented by these vectors determines the security level of CSIDH. Hence, analyzing whether different private key results in the same public key is important. The collisions for CSIDH and CSURF were examined in [13,15], respectively. We generalize this idea to the prime \( p \equiv 7 \mod 8 \). Details of our proof are presented in Section 3.4.

We analyze the performance and the security of the three algorithms—CSIDH, CSURF, and Onuki’s CSIDH over the prime \( p \) with \( p \equiv 3 \mod 8 \) and \( p \equiv 7 \mod 8 \). Additionally, we present a new interval for the private key to have a similar security level for the various CSIDH-based algorithms. The details of our implementation are presented in Section 4. From the implementation result, we conclude that for a prime \( p \equiv 7 \mod 8 \), CSIDH on the surface using the Montgomery curves is the most likely to be efficient. For a prime \( p \equiv 3 \mod 8 \), CSIDH on the floor using the hybrid method with Onuki’s collision-free method is the most likely to be efficient and secure.

This paper is organized as follows. In Section 2, we introduce two types of elliptic curves, which will be used for the implementation. We also present the computational cost of the lower-level functions to construct CSIDH-based algorithms over these curves. In Section 3, we review the CSIDH algorithms and two of its variants. The implementation results are presented in Section 4, and we draw our conclusions and future work in Section 5.

2. Montgomery Curve and Tweaked Montgomery Curve

This section introduces two types of Montgomery elliptic curves, which will be used throughout the paper. Then, we analyze the computational cost of elliptic curve arithmetic and isogeny computation on both curves, which are the main building blocks for implementing CSIDH-based algorithms.

Let \( K \) be a field with the characteristic not equal to 2 or 3. The Montgomery curves over \( K \) are denoted by

\[
M_{a,b} : by^2 = x^3 + ax^2 + x,
\]

where \( b(a^2 - 4) \neq 0 \). We shall write \( M_b \) when \( b = 1 \) throughout the paper. Moreover, the tweaked Montgomery curves over \( K \) are denoted by

\[
M_{a,b}^t : by^2 = x^3 + ax^2 - x,
\]

where \( b(a^2 + 4) \neq 0 \). We shall write \( M_b^t \) when \( b = 1 \) in this paper. Similar to the arithmetic on \( M_a \), the elliptic curve arithmetic on \( M_a^t \) can also be constructed using only \( x \)-coordinate.

For the remainder of this section, we introduce the elliptic curve arithmetic and isogeny formulas and analyze the computational cost for each operation on both curves. As the projective curve coefficient and projective coordinate are used for implementing isogeny-based cryptography, we shall evaluate the computational cost on both curves in these circumstances. For the elliptic curve arithmetic, we mainly focus on differential addition and doubling formula. For isogeny computation, we consider odd-degree isogenies.
2.1. Elliptic Curve Arithmetic on $M_a$ and $M'_a$

Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be a point on a Montgomery curve $M_a$ such that $x_P \neq x_Q$. Let $P - Q = (x_{P-Q}, y_{P-Q})$ be given. Then the $x$ coordinates of their sum $P + Q$ and the doubling of $P$, $x_{[2]P}$ can be computed as follows:

\[
\begin{align*}
x_{P+Q} &= (x Px_Q - 1)^2/(x_{P-Q}(x_P - x_Q)) \\
x_{[2]P} &= (x_P^2 - 1)^2/(4x_P(x_P^2 + ax_P + 1))
\end{align*}
\]

For a tweaked Montgomery curve $M'_a$, let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be a point on $M'_a$ such that $x_P \neq x_Q$. Let $P - Q = (x_{P-Q}, y_{P-Q})$ be given. Then the $x$ coordinates of their sum $P + Q$ and the doubling of $P$, $x_{[2]P}$ can be computed as follows [14]:

\[
\begin{align*}
x_{P+Q} &= (x Px_Q + 1)^2/(x_{P-Q}(x_P - x_Q)) \\
x_{[2]P} &= (x_P^2 + 1)^2/(4x_P(x_P^2 + ax_P - 1))
\end{align*}
\]

At a glance, the computational costs of the differential addition and doubling on both curves are the same. However, when projective $x$-coordinate (XZ-coordinate) and projective curve coefficients are used, the computational costs are slightly different for both curves.

Now let $P = (X_P : Z_P)$ and $Q = (X_Q : Z_Q)$ be a point on a Montgomery curve $M_a$ such that $x_P = X_P/Z_P$ and $x_Q = X_Q/Z_Q$ for $x_P \neq x_Q$. Let $P - Q = (X_{P-Q} : Z_{P-Q})$ be the given difference of $P$ and $Q$ in projective coordinates such that $x_{P-Q} = X_{P-Q}/Z_{P-Q}$. Then the addition formula in projective coordinates can be decomposed as follows [16]:

\[
\begin{align*}
X_{P+Q} &= Z_{P-Q}(X_P X_Q - Z_P Z_Q)^2 = Z_{P-Q}((X_P - Z_P)(X_Q + Z_Q) + (X_P + Z_P)(X_Q - Z_Q))^2, \\
Z_{P+Q} &= X_{P-Q}((X_P - Z_P)(X_Q + Z_Q) - (X_P + Z_P)(X_Q - Z_Q))^2.
\end{align*}
\]

The computational cost is $4M+2S$, where the $M$ and $S$ refers to a field multiplication and squaring, respectively. The doubling of $P$ gives $[2]P = (X_{[2]P} : Z_{[2]P})$, where $X_{[2]P}$ and $Z_{[2]P}$ are defined as:

\[
\begin{align*}
X_{[2]P} &= C(X_P^2 - Z_P)^2, \\
Z_{[2]P} &= 4X_P Z_P(CX_P^2 + CZ_P^2 + AXPZ_P),
\end{align*}
\]

where $a = A/C$. The computational cost is $4M+2S$.

On the other hand, for a tweaked Montgomery curve, let $P = (X_P : Z_P)$ and $Q = (X_Q, Z_Q)$ be a point on $M'_a$ such that $x_P = X_P/Z_P$ and $x_Q = X_Q/Z_Q$ for $x_P \neq x_Q$. Let $P - Q = (X_{P-Q} : Z_{P-Q})$ be the given difference of $P$ and $Q$ in projective coordinates such that $x_{P-Q} = X_{P-Q}/Z_{P-Q}$. Then the sum $P + Q$ in projective coordinates can be computed as follows:

\[
\begin{align*}
X_{P+Q} &= Z_{P-Q}(X_P X_Q + Z_P Z_Q)^2, \\
Z_{P+Q} &= X_{P-Q}(X_P Z_Q - Z_P X_Q)^2,
\end{align*}
\]

and the concrete computation process is presented as below:

\[
\begin{align*}
t_0 &= X_P \cdot X_Q, & t_1 &= Z_P \cdot Z_Q, & t_2 &= t_0 + t_1, & t_3 &= t_2^2, & X_{P+Q} &= t_3 \cdot Z_{P-Q}, \\
t_0 &= X_P \cdot Z_Q, & t_1 &= Z_P \cdot X_Q, & t_2 &= t_0 - t_1, & t_3 &= t_2^2, & Z_{P+Q} &= t_3 \cdot X_{P-Q}.
\end{align*}
\]

In this case, the techniques used to compute the differential addition on a curve $M_a$ cannot be used, so that the computational cost of the addition formula in the tweaked Montgomery curves is $6M+2S$. The doubling of $P$ gives $[2]P = (X_{[2]P} : Z_{[2]P})$, where $X_{[2]P}$ and $Z_{[2]P}$ are defined as:
\[ X_{[2]}p = C(X_p^2 + Z_p^2)^2, \]
\[ Z_{[2]}p = 4X_pZ_p(CX_p^2 - CZ_p^2 + AX_pZ_p), \]
where \( a = A/C \). Moreover, the concrete computation process is presented as below:

\[ t_0 = X_p^2, \quad t_1 = Z_p^2, \quad t_2 = t_0 + t_1, \quad t_3 = t_2^2, \quad X_{[2]}p = C \cdot t_3, \quad t_2 = t_0 - t_1, \quad t_3 = X_p \cdot Z_p, \]
\[ t_0 = C \cdot t_2, \quad t_1 = A \cdot t_3, \quad t_2 = t_0 + t_1, \quad t_0 = t_2 \cdot t_3, \quad t_1 = t_0 + t_0, \quad Z_{[2]}p = t_1 + t_1. \]

The computational cost of the doubling formula in the tweaked Montgomery curves is \( 5M + 3S \).

2.2. Odd-Degree Isogeny Formulas on \( M_a \) and \( M_a' \)

In [16], Costello and Hisil proposed a formula for computing an arbitrary odd-degree isogenies on the Montgomery curves. Let \( P = (x_1, y_1) \) be a point on a Montgomery curve \( M_a \), having order \( \ell = 2d + 1 \) and let \( (x_i, y_i) = [i]P \). Then \( \ell \)-isogeny \( \phi \) from \( M_a \) to \( M_a' = M_a/(P) \) is given by:

\[ \phi : (x, y) \rightarrow (f(x), y \cdot f'(x)) \]

where

\[ f(x) = x \cdot \prod_{i=1}^{d} \left( \frac{x x_i - 1}{x - x_i} \right)^2 \]

In the above equation, \( f'(x) \) is derivative of \( f(x) \) and \( a' = (6\sigma - 6\sigma + a)\pi^2 \) for \( \sigma = \sum_{i=1}^{d} x_i \), \( \bar{\sigma} = \sum_{i=1}^{d} 1/x_i \), and \( \pi = \prod_{i=1}^{d} x_i \).

Now, in projective \( XZ \)-coordinate, let \( [i]P = P_i = (X_i : Z_i) \), where \( x_i = X_i/Z_i \) and \( P_1 = P \). Then the evaluation of an isogeny refers to the computation of the image point of \( \phi \). Let \( Q = (X : Z) \) be another point on \( M_a \) and \( \phi(Q) = (X' : Z') \). Then \( X' \) and \( Z' \) are as follows:

\[ X' = X \left( \prod_{i=1}^{d} \left[ (X - Z)(X_i + Z_i) + (X + Z)(X_i - Z_i) \right] \right)^2, \tag{3} \]
\[ Z' = Z \left( \prod_{i=1}^{d} \left[ (X - Z)(X_i + Z_i) - (X + Z)(X_i - Z_i) \right] \right)^2. \tag{4} \]

The computational cost of this formula is \( (4d)M + 2S \). For the coefficient of the image curve \( a' \), Castryck et al. present a formula in projective coordinate in [6], which is as follows:

\[ a' = \pi(a - 3\sigma), \]

where \( \pi = \prod_{i=1}^{\ell-1} \frac{X_i}{Z_i} \) and \( \sigma = \sum_{i=1}^{\ell-1} \left( \frac{X_i}{Z_i} - \frac{Z_i}{X_i} \right) \). Which can be computed in \( (6d - 2)M + 3S \).

For the isogeny evaluation and computing the curve coefficient of the image curve of a tweaked Montgomery curve \( M_a' \) presented in [14], \( f(x) \) is now defined as:

\[ f(x) = x \cdot \prod_{i=1}^{d} \left( \frac{x x_i + 1}{x - x_i} \right)^2 \]
Let $P$ be a point on $M_d^\ell$, having order $\ell = 2d + 1$. In projective coordinate, let $[i]P = P_i = (X_i : Z_i)$, where $x_i = X_i/Z_i$ and $P_1 = P$. Let $\phi$ be an $\ell$-isogeny from $M_d^\ell$ to $M_d^\ell = M_d^\ell/(\ell P)$. Let $Q = (X : Z)$ be another point on $M_d^\ell$ and let $\phi(Q) = (X' : Z')$. Then $X'$ and $Z'$ are as follows:

$$X' = X\left(\prod_{i=1}^{d}(XX_i + ZZ_i)\right)^2,$$

$$Z' = Z\left(\prod_{i=1}^{d}(XZ_i - ZX_i)\right)^2.$$ (5)

Similar to the case for computing the differential addition, note that for (5) and (6), the optimized computation methods like (3) and (4) do not exist. So, the computational cost of odd-degree isogeny point evaluation on the tweaked Montgomery curve is $(6d)M + 2S$. Formula in [14] for computing the coefficient of the image curve $a'$, is similar to the formula for Montgomery curve which is

$$a' = \pi(a - 3\sigma),$$

where $\pi = \prod_{i=1}^{\ell-1} X_i/Z_i$ and $\sigma = \sum_{i=1}^{\ell-1} (X_i/Z_i + Z_i/X_i)$. This can be computed in $(6d - 2)M + 3S$.

Summarizing the section, Table 1 presents the computational cost of the elliptic curve arithmetic and isogeny operations on Montgomery and tweaked Montgomery curves. In Table 1, DBLADD refers to the differential addition with doubling, and DBL refers to the doubling. $\ell$-isogeny eval. denotes the evaluation of an $\ell$-isogeny, and $\ell$-isogeny coeff. denotes the computation of the coefficient of the image curve for an $\ell$-isogeny.

|                     | Montgomery Curves | Tweaked Montgomery Curves |
|---------------------|-------------------|---------------------------|
| DBLADD              | 8M+4S             | 11M+5S                    |
| DBL                 | 4M+2S             | 5M+3S                     |
| $\ell$-isogeny eval.| $(4d)M + 2S$      | $(6d)M + 2S$              |
| $\ell$-isogeny coeff.| $(6d - 2)M + 3S$  | $(6d - 2)M + 3S$          |

**Remark 1.** In [10], Meyer and Reith proposed a hybrid version of CSIDH, which exploits Edwards curves for recovering the coefficient of the image curve. By using the efficiency of the birational equivalence between Montgomery and Edwards curves, the coefficient of the image curve is obtained using the Edwards isogeny formula. The obtained Edwards curve coefficient is then transformed into the Montgomery coefficient. The computational cost is $(2d)M + 6S + 2w(\ell)$, where $w(\ell)$ is the cost of the $\ell$-th power on $\mathbb{F}_p$ [13].

### 3. CSIDH-Based Schemes

In this section, we introduce the CSIDH key exchange and two main CSIDH-based algorithms—CSURF [14] and collision-free CSIDH proposed by Onuki and Takagi [13]—to compare the performance and security. As CSIDH made a noticeable improvement by exploiting supersingular elliptic curves to instantiate the CRS scheme, various methods began to propose in order to optimize the performance and improve the security. The former is the CSURF, which proposes a way to exploit efficient horizontal 2-isogenies for a speed-up, and the latter is the method by Onuki and Takagi, where they analyzed the existence of a collision in the private keyspace and provided a method to eliminate such collisions. Before going into the details of the algorithms, we present three primes, $p_1$, $p_2$, and $p_3$, which will be used throughout the paper.
First, we use the primes $p_1$ and $p_2$ presented as below, in order to match the size of the base field for a fair comparison.

$$p_1 = 4 \cdot (3 \cdot \ldots \cdot 373) \cdot 587 - 1 \approx 2^{510.668},$$
73 first odd primes

$$p_2 = 4 \cdot 2^2 \cdot 3^2 \cdot 11 \cdot (3 \cdot \ldots \cdot 373) - 1 \approx 2^{510.100}.$$ 73 first odd primes

The first prime $p_1 \equiv 3 \mod 8$, presented in [6], will be used to compare the original CSIDH and Onuki’s CSIDH. The second prime $p_2 \equiv 7 \mod 8$, presented in [11], will be used to compare the original CSIDH and CSURF.

On the other hand, in [14], Castryck and Decru used the prime $p_3$ defined as below for CSURF. In this paper, we use $p_3$ to explain the CSURF algorithm, but $p_3$ will not be used for the implementation as the size of the prime is larger than $p_1$.

$$p_3 = 4 \cdot 2^3 \cdot (3 \cdot \ldots \cdot 389) - 1 \approx 2^{512.880}.$$ 74 consecutive primes, skip 347 and 359

### 3.1. CSIDH

CSIDH is an isogeny-based Diffie-Hellman protocol proposed by Castryck et al. [6] using supersingular curves defined over $\mathbb{F}_p$ and commutative group action. The prime $p$ of the base field is of the form $p = 4 \prod_{i=1}^n \ell_i - 1$, where $\ell_i$’s are odd primes. For an order $O = \text{End}_{\mathbb{F}_p}(E)$, it is well-known that the class group $\text{cl}(O)$ acts freely and transitively on $\mathcal{E}E_{\mathbb{F}_p}(O)$, where $\mathcal{E}E_{\mathbb{F}_p}(O)$ is the set of elliptic curves $E$ defined over $\mathbb{F}_p$ with $\text{End}_{\mathbb{F}_p}(E) = O$. This group action is represented by $[a]E$, where $E \in \mathcal{E}E_{\mathbb{F}_p}(O)$ and an ideal class $[a] \in \text{cl}(O)$. Since $E$ is a supersingular curve with $#E(\mathbb{F}_p) = p + 1 = 4 \cdot \ell_1 \cdots \ell_n$, for each $i$, there is $\mathbb{F}_p$-rational subgroup of order $\ell_i$. Moreover, let $\pi = \sqrt{−p}$ be the $\mathbb{F}_p$-Frobenius endomorphism of $E$. Then, since $p = −1 \mod \ell_i$, for a prime $\ell_i$, it is well-known that $\ell_iO$ splits into two prime ideals $I_i = (\ell_i, \pi - 1)$ and $I_i^{-1} = (\ell_i, \pi + 1)$. Using Velu’s formula, we compute $[b]E$ through the isogeny $\phi_{\ell_i}$ with the kernel generated by a point of order $\ell_i$, which lies in the kernel of $\pi - 1$ and compute $[I_i^{-1}]E$ through the isogeny $\phi_{\ell_i^{-1}}$ with the kernel generated by a point of order $\ell_i$, which lies in the kernel of $\pi + 1$.

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

As an element of the ideal-class group $\text{cl}(O)$ is expected to be of the form $\prod_{i=1}^n I_i^{e_i} = (\ell_i, \pi - 1)$ by Cohen–Lenstra heuristics, the private key $[a]$ and $[b]$ is represented as the integer vectors $(e_1, e_2, \ldots, e_n) \in \mathbb{Z}^n$, each of $e_i$ sampled randomly from a range $[-m, m]$. Thus a group action $[a]E$ can be computed by applying $\ell_i$-isogeny operation $e_i$ times for $a = \prod_{i=1}^n I_i^{e_i} \in \text{cl}(O)$. If $e_i > 0$, $\ell_i$-isogeny is applied with the kernel generated by a point in $E(\mathbb{F}_p)$ of order $\ell_i$. If $e_i < 0$, $\ell_i$-isogeny is applied with the kernel generated by a point in $E(\mathbb{F}_p) \backslash E(\mathbb{F}_p)$ of order $\ell_i$. The CSIDH-512 offers an 128-bit classical security and uses the prime $p_1$. The secret exponent of CSIDH-512 is of the form $(e_1, e_2, \ldots, e_{256})$, where $e_i \in [-5, 5]$. Thus, they expect that there are $11^{256} \approx 2^{285.998}$ distinct exponents.

### 3.2. Onuki’s CSIDH

In [13], Onuki and Takagi proposed a new interval of the secret exponent and a new method for computing the coefficient of the image curve using 4-torsion points for CSIDH protocol. In CSIDH, the ideal classes, which are used as a private key, are represented by vectors with integer coefficients. As the number of ideal classes represented by these vectors determines the security level of CSIDH, it is
important to examine the correspondence between the ideal classes and the vectors. They proved that the vector \((1, \ldots, 1)\) corresponds to an ideal class of order 3. This means that a secret exponent \((e_1, e_2, \ldots, e_n)\) and \((e_1 + 3, e_2 + 3, \ldots, e_n + 3)\) represents the same ideal class. Since CSIDH-512 selects a secret exponent \((e_1, e_2, \ldots, e_n)\) from a range \([-5, 5]\), there exists the collision of the form \((e_1 + 3, e_2 + 3, \ldots, e_n + 3)\). Thus, Onuki and Takagi used the ideal \(l_0 = (4, \pi - 1)\) instead of using the ideal \(l_0 = (\ell_n, \pi - 1)\). Therefore, a secret exponent proposed in [13] is of the form \((e_0, e_1, \ldots, e_{n-1})\) to compute class group action \([a]E := [e_0 e_1^{\pi} \cdots e_{n-1}^{\pi^{n-1}}]E\), where \(e_0 \in [-1, 1]\) and \(e_i \in [-m, m]\) for \(1 \leq i \leq n - 1\). They also proposed a new formula for computing the actions of the ideal classes represented by \((1, \ldots, 1)\) and \((-1, \ldots, -1)\) by using degree 4 isogenies. Let \(P_-\) and \(P_+\) be a point of \(M_a\) of \(x\)-coordinate \(-1\) and \(1\), respectively. Then, for 4-isogenies \(\phi : M_a \to M_{a'}\) used in this algorithm with \(\ker \phi = \langle P_- \rangle\) (resp. \(\ker \phi = \langle P_+ \rangle\)), \(a'\) is computed as

\[
a' = \frac{2a - 12}{a + 2} \quad \text{(resp. } a' = \frac{-2a - 12}{a - 2} \text{).} \tag{7}
\]

The former case is computed if \(e_0 = 1\), and the latter case is computed when \(e_0 = -1\). Applicability of these 4-isogenies is argued in below proposition, where \(\text{End}_p(M_a) \cong \mathbb{Z}[(1 + \sqrt{-p})/2]\).

**Proposition 1.** Let \(p\) be a prime of the form \(p \equiv 7 \mod 8\) and \(M_a : y^2 = x^3 + ax^2 + x\) be a Montgomery curve with the endomorphism ring \(\text{End}_p(M_a) \cong \mathcal{O}_K\). Then, the 4-isogenies \(\phi : M_a \to M_{a'}\) presented in (7)

\[
\begin{align*}
(i) & \quad a' = \frac{2a - 12}{a + 2} \\
(ii) & \quad a' = \frac{-2a - 12}{a - 2}
\end{align*}
\]

do not preserve the same \((\ell, \pi - 1)\) action class.

**Proof.** First, the set \(S_{p,\mathcal{O}_K} = \{a \mid \text{End}_p(M_a) \cong \mathcal{O}_K\} \) splits into two partitions, denoted \(S_{p,1}^+\) and \(S_{p,2}^+\), respectively. These sets are defined as below,

\[
\begin{align*}
S_{p,1}^+ &= \{a \in S_{p,\mathcal{O}_K}^+ \mid (0, 0) \not\in 2M_a(\mathbb{F}_p)\}, \\
S_{p,2}^+ &= \{a \in S_{p,\mathcal{O}_K}^+ \mid (0, 0) \in 2M_a(\mathbb{F}_p)\}.
\end{align*}
\]

Note that \(a \pm 2\) are both square in \(\mathbb{F}_p\) if \(a \in S_{p,2}^+\), and \(a \pm 2\) are not both square in \(\mathbb{F}_p\) if \(a \in S_{p,1}^+\) by Theorem 2 in [11]. Assume that we apply above former 4-isogeny. Since \(a' = (2a - 12)/(a + 2)\),

\[
a' + 2 = \frac{4(a - 2)}{a + 2} \quad \text{and} \quad a' - 2 = \frac{-16}{a + 2}.
\]

If \(a \in S_{p,1}^+\), then \(a' + 2\) are both square in \(\mathbb{F}_p\), so that \(a' \in S_{p,2}^+\). If \(a \in S_{p,2}^+\), then \(a' + 2\) is square and \(a' - 2\) is not square in \(\mathbb{F}_p\), so that \(a' \in S_{p,\mathcal{O}}\). That is, the image curve \(M_{a'}\) is on the floor. Similarly, the case of later 4-isogeny does not preserve the same \((\ell, \pi - 1)\) action class. \(\square\)

We summarize the result of Proposition 1 in Table 2.

| Table 2. The results of 4-isogenies in [13] on the surface. |
|---------------------------------|-----------------|-----------------|-----------------|
| \(S_{p,1}^+\)                  | \(S_{p,2}^+\)   | \(S_{p,\mathcal{O}}\) | \(S_{p,1}^+\)   |
|---------------------------------|-----------------|-----------------|-----------------|
| \((i)\)                        | \((ii)\)        | \((i)\)        | \((ii)\)        |
|---------------------------------|-----------------|-----------------|-----------------|
|\(a' + 2\)                      |\(a' - 2\)       |\(a' + 2\)       |\(a' - 2\)       |
|\(a' + 2\)                      |\(a' - 2\)       |\(a' + 2\)       |\(a' - 2\)       |
|\(a' + 2\)                      |\(a' - 2\)       |\(a' + 2\)       |\(a' - 2\)       |
|\(a' + 2\)                      |\(a' - 2\)       |\(a' + 2\)       |\(a' - 2\)       |
Moreover, Onuki and Takagi presented a new formula for computing the image coefficient of the image curve using 4-torsion points. For a \((2d+1)\)-isogeny \(\phi : M_a \to M_{a'}\) with \(\ker \phi = \langle P \rangle\), \(a'\) is computed as below,

\[
C' = C\left(\prod_{i=1}^{d}(X_i + Z_i) \prod_{i=1}^{d} Z_i\right)^2,
\]

\[
A' = (A - 2C) \left(\prod_{i=1}^{d}(X_i + Z_i) + 2 \sum_{i=1}^{d}(X_i - Z_i) \prod_{j \neq i}(X_i + Z_i) \prod_{i=1}^{d} X_i\right)^2 + 2C',
\]

where \([i]P = (X_i : Z_i), a = A/C, \) and \(a' = A'/C'\). The computational cost of this formula is \((5d - 1)M + 2S\).

This collision-free CSIDH proposed by Onuki and Takagi offers little extra security to the original CSIDH. For the implementation of Onuki’s algorithm, we use the prime \(p_1\) with \(n = 74\) and \(m = 5\), as the parameters are not explicitly described in [13]. This setting gives \(3 \cdot 11^{73} \approx 2^{254.125}\) distinct exponents.

### 3.3. CSURF

Since CSIDH protocol used a prime of the form \(p \equiv 3 \mod 8\), the Montgomery curves \(M_a(F_p)\) has no \(F_p\)-rational 2-torsion point except for \((0,0)\). Using only odd-degree isogenies without 2-isogenies resulted in the inefficiency of computing the class group action. To overcome this problem, Castryck and Decru presented a new hard homogeneous space using tweaked Montgomery curves in [14]. The CSURF protocol uses a prime of the form \(p \equiv 7 \mod 8\) and the tweaked Montgomery curves \(M'_a/F_p\). Thus, the \(F_p\)-endomorphism ring of \(M'_a/F_p\) is isomorphic to \(Z_l(1 + \sqrt{-p})/2\) and every curve in this setting has three \(F_p\)-rational 2-torsion points. Hence, CSURF can now exploit horizontal 2-isogenies with the ideal \(l_0 = (2, \sqrt{-7} - 1)\) to help compute the class group action. For a supersingular Montgomery curve \(M_a\), the Montgomery coefficient \(a\) and \(F_p\)-isomorphism class of \(M_a\) are one-to-one correspondence when the base prime \(p\) is of the form \(p \equiv 3 \mod 8\). Likewise, for a supersingular tweaked Montgomery curve \(M'_a\), the tweaked Montgomery coefficient \(a\) and \(F_p\)-isomorphism class of \(M'_a\) are also one-to-one correspondence when the base prime \(p\) is of the form \(p \equiv 7 \mod 8\). This is summarized in Table 3. Thus, Castryck and Decru can construct well-defined free and transitive group action. Finally, they used a secret exponent \((e_0, e_1, \ldots, e_n)\) to compute class group action \([a]M'_a := [l_0^{e_0} t_1^{e_1} \cdots t_n^{e_n}]M'_a\).

| \(p \equiv 3 \mod 8\) | \(|\mathcal{S}^+_{p,|O|} : |\mathcal{C}Ell_p(|O|)|\) | \(|\mathcal{S}^+_{p,\mathcal{O}} : |\mathcal{C}Ell_p(|O|)|\) |
|---|---|---|
| \(O = Z_l[1 + \sqrt{-p}]\) | 0 | (3:1) |
| \(O' = Z_l[\sqrt{-p}]\) | (1:1) | 0 |

In [14], Castryck and Decru used the prime \(p_3\), which have 74 odd primes. For a secret exponent, they used \((e_0, e_1, \ldots, e_74) \in [-137, 137]^3 \times [-4, 4]^13 \times [-5, 5]^46 \times [-4, 4]^{25}\) with the class group action \([l_0^{e_0} t_1^{e_1} \cdots t_74^{e_74}]\), where \(l_0 = (2, \sqrt{-p_3 - 1})\). This leads to about \(275 \cdot 9^{28} \cdot 11^{46} \approx 2^{255.995}\) distinct exponents.

### 3.4. Collisions for CSIDH-Based Algorithms

In this subsection, we examine the correspondence between the ideal classes and the vectors for the CSIDH-based algorithms. As denoted in [6, 13, 14], the private keys in CSIDH-based algorithms are ideal classes in the class group \(\text{cl}(|O|)\). Due to the design choices, this ideals can be expected to have the
form $\prod_{i=1}^{n} \ell_i^e$, for small $e_i$. Hence, selecting an ideal classes corresponds to selecting an integer vector $(e_1, \ldots, e_n)$. Therefore, for an exact security evaluation, analyzing whether two different integer vectors $(e_1, \ldots, e_n)$ and $(e_1', \ldots, e_n')$ represent the same ideal class is important. In CSIDH-based schemes, there are different collisions depending on the prime of the base field and the $F_p$-endomorphism ring $End_p(E)$ of the elliptic curve $E$. As we use two different types of prime $p_1$ and $p_2$ for our implementation, we examine the collision in this prime field. We first state the main theorem in [13] and the following corollary.

**Theorem 1** ([13]). Let $p \equiv 3 \mod 8$ and $End_p(E) = \mathcal{O}$. Then, the ideal class $\{\ell_1^{e_1} \cdots \ell_n^{e_n}\}$ has order 3 in $\text{cl}(\mathcal{O})$.

**Proof.** See ([13], Theorem 3). $\square$

**Corollary 1** ([13]). In the CSIDH protocol, the secret exponents

$$(e_1, e_2, \ldots, e_n) \quad \text{and} \quad (e_1 + 3r_1, e_2 + 3r_2, \ldots, e_n + 3r_n)$$

represent the same ideal class.

CSIDH-512 use $p_1$ as the base prime, where $p_1 \equiv 3 \mod 8$. Hence, the collisions pointed out in [13] exists for the original parameters of CSIDH-512. Similarly, in [15] Fan et al. proved that there also exist collisions of the form $(e_0 + 1, e_1 + 2, e_2 + 1, \ldots, e_{74} + 1)$ for the CSURF prime $p_3$.

Now, we generalize the idea of [15] to a prime $p \equiv 7 \mod 8$ of a certain form.

Let $p$ be a prime of base field of the form $p = 4 \cdot 2^n \ell_1^{e_1} \cdots \ell_n^{e_n} - 1$, with $p \equiv 7 \mod 8$. Let $\pi = \sqrt{-\frac{p}{2}}$, $K = \mathbb{Q}(\pi)$, $\mathcal{O} = \mathbb{Z}[\pi]$, and $\mathcal{O}_K = \mathbb{Z}[\frac{1 + \pi}{2}]$. Theorem 2 generalizes the theorem in [15] to the prime $p$, where $p \equiv 7 \mod 8$.

**Theorem 2.** Let $p \equiv 7 \mod 8$ and $End_p(E) = \mathcal{O}_K$. Then the secret exponents

$$(e_0, e_1, \ldots, e_n) \quad \text{and} \quad (e_0 + r_0, e_1 + r_1, \ldots, e_n + r_n)$$

represent the same ideal class, where $l_0 = (2, \frac{\pi - 1}{2})$.

**Proof.** Since $l_0 = (\ell_0, \pi - 1) = (\ell_0, \frac{\pi - 1}{2})$ in $\text{cl}(\mathcal{O}_K)$, the followings are hold.

$$\ell_0^{e_0} \ell_1^{e_1} \cdots \ell_n^{e_n} = (2, \frac{\pi - 1}{2}) r_0 \cdot (\ell_1, \frac{\pi - 1}{2}) r_1 \cdots (\ell_n, \frac{\pi - 1}{2}) r_n$$

$$= (2^n \ell_1^{e_1} \cdots \ell_n^{e_n}, \frac{\pi - 1}{2})$$

$$= (p + 1, \pi - 1) = (\frac{\pi - 1}{2})$$

Thus, $[\ell_0^{e_0} \ell_1^{e_1} \cdots \ell_n^{e_n}]E = [l_0^{r_0 + e_0} l_1^{r_1 + e_1} \cdots l_n^{r_n + e_n}]E$. $\square$

**Corollary 2.** When the prime $p_2$ is used on the surface, the secret exponents

$$(e_0, e_1, \ldots, e_{73}) \quad \text{and} \quad (e_0 + 2, e_1 + 3, e_2 + 1, e_3 + 1, e_4 + 2, e_5 + 1, \ldots, e_{73} + 1)$$

represent the same ideal class.

Since collisions in a secret exponent reduce the size of the private keyspace, we must either avoid the collisions or endure the risk for collisions by counting the number of possible public keys.
Theorem 3 deals with the number of ideal classes that a secret exponent can represent, assuming that a collision exists.

**Theorem 3.** Assume that a secret exponent \((e_0, e_1, \ldots, e_n)\) has the collision so that it represent the same ideal class as \((e_0 + c_0, e_1 + c_1, \ldots, e_n + c_n)\), where \(e_i \in [-m_i, m_i]\), and \(c_i \in \mathbb{Z}\). Then, there are \(\prod_{i=0}^{n}(2m_i + 1 - c_i)\) collisions. Therefore the order of the private key space \(\left\{ [c_0^{e_0}c_1^{e_1} \cdots c_n^{e_n}]E \mid e_i \in [-m_i, m_i]\right\}\) is \(\prod_{i=0}^{n}(2m_i + 1) - \prod_{i=0}^{n}(2m_i + 1 - c_i)\).

**Proof.** If every exponent \(e_i\) is equal to or greater than \(c_i - m_i\), then the secret exponent \((e_0, e_1, \ldots, e_n)\) and \((e_0 - c_0, e_1 - c_1, \ldots, e_n - c_n)\) represent the same ideal class. So, there are \(\prod_{i=0}^{n}(m_i - (c_i - m_i) + 1) = \prod_{i=0}^{n}(2m_i + 1 - c_i)\) collisions and we must contain \(e_i\) smaller than \(c_i - m_i\) for at least one \(i \in [0, n]\). Thus, \(\left| \left\{ [c_0^{e_0}c_1^{e_1} \cdots c_n^{e_n}]E \mid e_i \in [-m_i, m_i]\right\}\right| = \prod_{i=0}^{n}(2m_i + 1) - \prod_{i=0}^{n}(2m_i + 1 - c_i)\). \(\square\)

To avoid this type of collision, we can consider two options – dropping some degree of an isogeny or adding supplementary factors to the prime of the base field. The former method is used in [13,15] and has the advantage of avoiding computation of some large odd-degree isogenies. However, the interval adjustment of a secret exponent is inevitable to guarantee the security of the protocol. The latter method is to let exponent \(r_i\) of \(\ell_i\) in \(p\) for at least one \(i\) to be bigger than \(2m_i\). In this case, we must choose the prime having those factors. The advantage of this method is that we can expect that CSIDH protocols to have a certain level of the resistance for subexponential quantum attack [2,17,18] by expanding the size of the base field.

### 4. Implementation and Security Analysis

In this section, we provide the implementation results and security analysis for the algorithms presented in previous chapters. First, we measure the performance of each algorithm using the initial parameters. However, for implementing the algorithms on the surface, we choose \(p_2\) as the prime of the base field to match the cost of the field arithmetic with \(p_1\), as much as possible. Then, we present the performance result by modifying the interval of the private key of each algorithm in order to match the security level.

All of the algorithms in this paper are implemented in C language to evaluate the performance of each algorithm. To this end, we use the field arithmetic implemented in [6]. Moreover, wall-clock times and clock cycles are obtained on one core of an Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, running Ubuntu 18.04.1 LTS. For compilation, we used GNU GCC version 7.5.0 with compile option -O3 using the benchmark provided by [6]. All results are averaged over 500,000 rounds.

Note that the prime \(p_1\) and the initial curve \(y^2 = x^3 + x\) are used for implementing the original CSIDH and Onuki’s CSIDH, and the prime \(p_2\) and the initial curve \(y^2 = x^3 - x\) are used for implementing CSURF. When implementing CSIDH over \(p_2\), as a rational 2-torsion point exist, the 2-torsion method in [11,16] is used for the implementation. Therefore, for CSIDH over \(p_2\), the curve \(y^2 = x^3 + ax^2 + x\) is used as the initial curve, where \(a\) of the initial curve is presented in [11]. We also implement Meyer’s hybrid method for a fair comparison. The prime and the base curve for implementing Meyer’s hybrid method follows the setting of CSIDH.

Table 4 shows the implementation results for each scheme, using the intervals provided in the original papers. For Onuki’s CSIDH, as [13] does not specified the intervals of the secret exponents, we arbitrarily set the intervals according the the security level. The security in Table 4 is the result of considering the collisions mentioned in Section 3.4. For CSIDH over \(p_1\) and \(p_2\), this equals to \(11^{74} - 8^{74} \approx 2^{255.998}\) and \(11^{73} - 8^{73} \approx 2^{252.536}\), respectively. For CSURF over \(p_2\), this equals to \(275 \cdot 9^{28} \cdot 11^{45} - 273 \cdot 6 \cdot 8^{27} \cdot 9 \cdot 10^{44} \approx 2^{252.535}\).
Table 4. Wall-clock time and clock cycles of group action of CSIDH-based algorithms (original).

|       | CSIDH [6] | Onuki’s [13] | Hybrid [10] | CSURF [14] |
|-------|-----------|---------------|-------------|------------|
| $p_1$ | 32.39 ms  | 30.77 ms      | 28.68 ms    | -          |
|       | 110,401,470 cc | 104,866,008 cc | 97,752,709 cc | -          |
| $p_2$ | 29.62 ms  | -             | 28.01 ms    | 39.38 ms   |
|       | 100,945,178 cc | -         | 95,480,448 cc | 134,216,582 cc |
| Interval | $[-5,5]^{74}/[-5,5]^{73}$ | $[-1,1] \times [-5,5]^{73}$ | $[-5,5]^{74}/[-5,5]^{73}$ | $[-137,137] \times [-4,4]^{13} \times [-5,5]^{45} \times [-4,4]^{25}$ |
| Security | 255.998/252.536 | 254.123      | 255.998/252.536 | 252.535 |

As CSURF is an algorithm only applicable on the surface of supersingular curves, the result using $p_1$ does not exist. Similarly, Onuki’s method cannot be applied directly on the surface using $p_2$, so that the result using this prime does not exist. This is because 4-isogenies presented in [13] do not preserve the same $(\ell, \pi - 1)$ action class of the Montgomery curves used on the surface, as proved in the Proposition 1.

Lastly, we provide the performance of CSIDH-based algorithms by modifying the intervals of the secret exponents for a similar security level. As in [6,13,14], we heuristically expect that these exponents represent the elements of the class group quasi-uniformly. Note that the intervals are modified in a way so that the first three 3-, 5-, and 7-isogenies are performed up to four times, as in line with the idea in [14]. We manage to select the exponent of the first three primes small since probability of selecting a random small torsion point is lower than selecting a random large torsion point.

**Remark 2.** We do not apply other technical optimization methods like SIMBA [19], new addition chains for a scalar multiplication [20], and Velusqrt algorithm [21]. This is because we intend to present the comparison results of primitive algorithms as possible. Except for applying Velusqrt to the original CSIDH and the Onuki’s method, those techniques are applicable for all of the algorithms in this paper.

As denote in Table 4 and 5, CSIDH using $p_2$ is faster than CSIDH using $p_1$. While this speed gap is meaningless because the hybrid method surpasses both algorithms, we conclude that the potential derived from the applicability of 2-isogenies makes CSIDH on the surface more attractive as computing 2-isogeny in CSURF [14] does not require sampling of a 2-torsion point. On the other hand, CSURF is slower than other algorithms, since the tweaked Montgomery curves have inefficient elliptic curve arithmetic – DBLADD and isogeny evaluation – compared to the Montgomery curves in projective coordinates. Thus, deducing from the implementation of CSIDH and CSURF, instead of using tweaked Montgomery curves, CSIDH on the surface can be executed more efficiently by using a prime of the form $p \equiv 7 \mod 8$ and the Montgomery curves. When prime $p \equiv 3 \mod 8$ is used, then we can implement CSIDH efficiently on the floor by exploiting the hybrid method proposed in [10]. Moreover, it is recommended to use Onuki’s collision-free method, since an attack on the collision can potentially exist.
Table 5. Wall-clock time, clock cycles, and security considering collisions with modifying intervals.

|       | CSIDH [6] | Onuki’s [13] | Hybrid [10] | CSURF [14] |
|-------|-----------|--------------|-------------|------------|
| $p_1$ | 31.79 ms  | 30.43 ms     | 28.38 ms    | -          |
|       | 108,355,188 cc | 103,712,678 cc | 96,735,839 cc | -          |
| $p_2$ | 30.57 ms  | -            | 28.74 ms    | 40.81 ms   |
|       | 104,177,392 cc | -            | 97,963,451 cc | 139,102,772 cc |
| Interval | $[-4, 4]^3 \times [-6, 6]^{11} \times [-5, 5]^{12} \times [-4, 4]^{11}$ | $[-1, 1] \times [-4, 4]^3 \times [-6, 6]^{38} \times [-5, 5]^{12} \times [-4, 4]^{22}$ | $[-4, 4]^3 \times [-6, 6]^{14} \times [-5, 5]^{10} \times [-4, 4]^{19} \times [-6, 6]^{29} \times [-5, 5]^{15}$ | $[-4, 4]^{25}$ |
| Security | 256.051/256.112 | 256.044 | 256.051/256.112 | 256.065 |

5. Conclusions

In this paper, we provide the performance and security analysis for the various CSIDH-based algorithms. First, we implement the CSIDH-based algorithms presented in [6,13,14] in C for a fair comparison between those algorithms. By projectivizing the arithmetic formula in the tweaked Montgomery curve, we conclude that using this curve is inefficient compared to using the Montgomery curves, as of now.

Moreover, we analyze the security against brute-force attack on the private key by generalizing the possible collisions in CSIDH executed on the surface. In this regard, we present a new interval for the private key to have a similar security level for those algorithms. Thus, we can compare fairly the performances of three algorithms and offer optimization scenarios for using each parameter.

From the implementation result, we conclude that for a prime $p \equiv 7 \mod 8$, then CSIDH on the surface using the Montgomery curves is the most likely to be efficient. For a prime $p \equiv 3 \mod 8$ CSIDH on the floor, using the hybrid method Onuki’s collision-free method is the most likely to be efficient and secure.

For future work, we plan to study a potential attack against CSIDH-based algorithms with the collisions presented in this paper. Additionally, we plan to implement an optimized algorithm for each form of base primes and to provide more obvious standards for parameter selection by applying the various optimization methods as in [19–21].

Author Contributions: Conceptualization, D.H. and S.K.; Data curation, D.H.; Formal analysis, D.H.; Methodology, D.H. and S.K.; Resources, D.H.; Software, D.H.; Supervision, S.K., Y.-H.P and S.H.; Validation, S.K., Y.-H.P. and S.H.; Writing—original draft, D.H. and S.K.; Writing—review and editing, D.H. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Military Crypto Research Center funded by the Defense Acquisition Program Administration (DAPA) and the Agency for Defense Development (ADD) under Grant UD170109ED.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Couveignes, J.-M. Hard Homogeneous Spaces. 2006. Available online: https://eprint.iacr.org/2006/291 (accessed on 6 June 2020).
2. Childs, A.; Jao, D.; Soukharev, V. Constructing elliptic curve isogenies in quantum subexponential time. J. Math. Cryptol. 2014, 8, 1–29.
3. Feo, L.D.; Jao, D.; Plüt, J. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Math. Cryptol. 2014, 8, 209–247.
4. Koziel, B.; Ackie, A.B.; El Khatib, R.; Azarderakhsh, R.; Kermani, M.M. Supersingular Isogeny Key Encapsulation. Submission to the NIST Post-Quantum Standardization Project. 2017. Available online: https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions (accessed on 6 June 2020).

5. Feo, L.D.; Kieffer, J.; Smith, B. Towards practical key exchange from ordinary isogeny graphs. In Proceedings of the 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia, 2–6 December 2018; pp. 365–394.

6. Castryck, W.; Lange, T.; Martindale, C.; Panny, L.; Renes, J. CSIDH: An efficient post-quantum commutative group action. In Proceedings of the 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia, 2–6 December 2018; pp. 395–427.

7. Beullens, W.; Kleinjung, T.; Vercauteren, F. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Proceedings of the 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, 8–12 December 2019; pp. 227–247.

8. Yoo, Y.; Azarderakhsh, R.; Jalali, A.; Jao, D.; Soukharev, V. A post-quantum digital signature scheme based on supersingular isogenies. In Proceedings of the International Conference on Financial Cryptography and Data Security, Kota Kinabalu, Malaysia, 10–14 February 2017; pp. 163–181.

9. De Feo, L.; Galbraith, S.D. SeaSign: Compact isogeny signatures from class group actions. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, 19–23 May 2019; pp. 759–798.

10. Meyer, M.; Reith, S. A faster way to the CSIDH. In Proceedings of the 19th International Conference Cryptology in India, New Delhi, India, 9–12 December 2018; pp. 137–152.

11. Heo, D.; Kim, S.; Yoon, K.; Park, Y.-H.; Hong, S. Optimized CSIDH Implementation Using a 2-Torsion Point. Cryptography 2020, 4, 20.

12. Kim, S.; Yoon, K.; Park, Y.H.; Hong, S. Optimized method for computing odd-degree isogenies on Edwards curves. In Proceedings of the 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, 8–12 December 2019; pp. 273–292.

13. Onuki, H.; Takagi, T. On Collisions Related to an Ideal Class of Order 3 in CSIDH. 2019. Available online: https://eprint.iacr.org/2019/1209 (accessed on 20 Aug 2020).

14. Castryck, W.; Decru, T. CSIDH on the surface. In Proceedings of the 11th International Conference PQCrypto 2020, Paris, France, 15–17 April 2020; pp. 111–129.

15. Fan, X.; Tian, S.; Li, B.; Xiu, X. CSIDH on Other Form of Elliptic Curves. 2019. Available online: https://eprint.iacr.org/2019/1417 (accessed on 20 Aug 2020).

16. Costello, C.; Hisil, H. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In Proceedings of the 23rd International Conference on the Theory and Application of Cryptology and Information Security, Hong Kong, China, 3–7 December 2017; pp. 303–329.

17. Peikert, C. He Gives C-Sieves on the CSIDH. In Advances in Cryptology—EUROCRYPT 2020; Canteaut, A., Ishai, Y., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12106.

18. Bonnetain, X.; Schrötenholer, A. Quantum Security Analysis of CSIDH. In Advances in Cryptology—EUROCRYPT 2020; Canteaut, A., Ishai, Y., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12106.

19. Meyer, M.; Campos, F.; Reith, S. On Lions and Elligators: An Efficient Constant-Time Implementation of CSIDH. 2018. Available online: https://eprint.iacr.org/2018/1198 (accessed on 23 Sep 2020).

20. Cervantes-Vázquez, D.; Chenu, M.; Chi-Domínguez, J.J.; De Feo, L.; Rodríguez-Henríquez, F.; Smith, B. Stronger and Faster Side-Channel Protections for CSIDH. 2019. Available online: https://eprint.iacr.org/2019/837 (accessed on 23 Sep 2020).

21. Bernstein, D.; De Feo, L.; Leroux, A.; Smith, B. Faster Computation of Isogenies of Large Prime Degree. 2020. Available online: https://eprint.iacr.org/2020/341 (accessed on 23 Sep 2020).