IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 April 2023, accepted 27 April 2023, date of publication 2 May 2023, date of current version 10 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3272549

RESEARCH ARTICLE

Practical Usage of Radical Isogenies for CSIDH

DONGHOE HEO [1], **SUHRI KIM** [2], **AND SEOKHIE HONG** [1]
[1]School of Cybersecurity, Korea University, Seoul 02841, South Korea
[2]School of Mathematics, Statistics and Data Science, Sungshin Women's University, Seoul 02844, South Korea

Corresponding author: Suhri Kim (suhrikim@sungshin.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant NRF-2020R1A2C1011769.

ABSTRACT Recently, a radical isogeny was proposed to boost commutative supersingular isogeny Diffie–Hellman (CSIDH) implementation. Radical isogenies reduce the generation of a kernel of a small prime order when implementing CSIDH. However, when the size of the base field increases, field exponentiation, a core component of computing radical isogenies, becomes more computationally intensive. As the size of the field inevitably grows to resist a quantum attack, so it is necessary to discuss the practical utilization of the radical CSIDH. This paper presents an optimized implementation of radical isogenies and analyzes its ideal use in CSIDH-based cryptography with a review of quantum analysis. We tailored the formula for transforming Montgomery curves into the Tate normal form and further optimized the radical 2-isogeny formula and projective versions of the radical 5- and 7-isogenies. Except for CSIDH-512, using only the radical 2-isogeny for all parameters improves performance by 6% to 10%.

INDEX TERMS CSIDH, isogeny, post-quantum cryptography, radical isogeny.

I. INTRODUCTION

Isogeny-based cryptography was first proposed by Couveignes in [15]. Couveignes presented a non-interactive key exchange using ordinary elliptic curves defined over $\mathbb{F}_q$, whose endomorphism ring is equivalent to a given order $\mathcal{O}$ in an imaginary quadratic field. A Diffie–Hellman-like key exchange protocol can be constructed from the commutativity of $Cl(\mathcal{O})$. This work was later rediscovered independently by Rostovtsev and Stolbunov [13], which is now called the CRS scheme. However, the quantum-subexponential attack exists for the scheme [14], and the scheme is inefficient for practical use.

The isogeny-based cryptography regained attention after the introduction of supersingular isogeny Diffie–Hellman (SIDH) by Jao and De Feo [12]. As SIDH uses supersingular elliptic curves, the endomorphism ring is non-commutative, so it resists the attack proposed in [14]. The security of SIDH is based on the difficulty of finding an isogeny between two given isogenous elliptic curves over a finite field, known to be quantum-exponential. The supersingular isogeny key encapsulation (SIKE), a key encapsulation mechanism based on SIDH, was selected as an alternative candidate for NIST PQC standardization round three. However, due to a polynomial-time key recovery attack by Castryck and Decru, SIDH-based cryptosystems are no longer safe [11]. Although various masking methods are presented in [8], [10], and [9], masked variants of SIDH are not yet attractive in terms of performance and key size. Thus, commutative SIDH (CSIDH), described later, could be a more attractive choice.

The CRS scheme was revisited by De Feo, Kieffer, and Smith in [17] and independently by Castryck et al. in [16]. The advantage of the CRS scheme is that it offers efficient and safe public key validation, making it suitable for constructing a noninteractive key exchange [17]. In [17], they modernized the CRS construction by offering a more efficient method to compute the group action and select algorithm parameters. The CRS scheme was further improved by Castryck et al. in [16] by proposing CSIDH, which solves the parameter selection problem of the CRS schemes using supersingular elliptic curves defined over $\mathbb{F}_p$. As SIDH-based cryptosystems become inefficient, CSIDH has attracted more researcher interest because various cryptographic primitives can be constructed [24], [25]. The average performance of

The associate editor coordinating the review of this manuscript and approving it for publication was S. K. Hafizul Islam.

one group action of CSIDH is around tens of milliseconds, which is faster than other CRS-based protocols.

The advantage of CSIDH-based cryptography is that its key size is smaller than that of any other PQC primitives. However, unlike other PQC primitives, which use simple matrix-vector multiplication as building blocks, isogeny-based cryptography uses complicated elliptic curve arithmetic over a finite field larger than 500 bits. The disadvantage of isogeny-based cryptography is that it is much slower than other PQC primitives. Hence, numerous studies have been proposed to optimize the performance of isogeny-based cryptography. One line of work is to optimize isogeny computation, which can be performed using another form of elliptic curve or by optimizing the isogeny formula. In [27], [28], and [26], hybrid methods employing the birational equivalence between Montgomery and twisted Edwards curves have been proposed for faster implementation. To optimize isogeny computation, Bernstein et al. recently proposed a new method of computing an ℓ-isogeny, reducing the computational cost from $\tilde{O}(\ell)$ to $\tilde{O}(\sqrt{\ell})$ field operations [23]. Another line of work is to tweak the current schemes for faster implementation. In [29], Costello proposed a new type of SIDH called B-SIDH. In this scheme, Alice computes isogenies from a $(p+1)$-torsion supersingular curve subgroup, while Bob computes on the $(p-1)$-torsion subgroup of the quadratic twist of the curve. In addition, B-SIDH can be viewed as a tweak to SIDH, allowing faster computation on Alice's side with a more reduction-friendly prime field.

For CSIDH, CSURF was proposed in [22], exploiting the horizontal 2-isogenies using the supersingular elliptic curves defined on the surface. Further, CSURF uses supersingular elliptic curves with the endomorphism ring $\mathbb{Z}[(1+\sqrt{-p})/2]$ for $p \equiv 7 \mod 8$. They demonstrated that these elliptic curves could be identified with tweaked Montgomery curves (Montgomery$^-$ curves), which have elliptic curve arithmetic and isogeny formulae similar to Montgomery curves (Montgomery$^+$ curves). Over this prime field, the prime number 2 splits in $\mathbb{Q}(\sqrt{-p})$, allowing for the use of horizontal 2-isogenies. As a 2-isogeny merely consists of a single exponentiation over $\mathbb{F}_p$, adjusting the private key exponent can lead to better performance, and the desired security level can be tailored more precisely. The CSURF method is slower than CSIDH, as the elliptic curve arithmetic and isogeny formula using projective coordinates are slower on Montgomery$^-$ curves than on Montgomery$^+$ curves.

However, the idea of exploiting the 2-isogeny has extended to the introduction of the *radical isogeny* in [21]. The CSIDH-based algorithms require isogeny computations of various degrees, and for this operation, a point on an elliptic curve of a specific order must be created to generate a kernel of an isogeny. A random point Q is selected in $\mathbb{F}_p$ to generate a kernel of a given order, which costs approximately $1.5 \log p$ field multiplications, and is multiplied by some cofactor k, which costs approximately $11 \log p$ field multiplications in CSIDH-based settings. If $P = [k]Q$ equals the identity, another random point is selected to repeat the process. Hence, generating a kernel is a painstaking process, especially for small torsion points where the failure probability is $1/\ell$ [18], [21].

Hence, in [21], a novel approach called *radical isogeny* is introduced that computes chains of n-isogenies. This approach requires sampling at most one n-torsion point. Similar to CSURF, the maximum value of the private key exponent corresponding to primes using radical isogeny can be enlarged, and the maximum value of the private key exponent corresponding to primes not using radical isogeny can be reduced to minimize the number of kernel point generations.

A. OUR CONTRIBUTIONS

This work analyzes the optimal usage of radical isogenies for implementing CSIDH. The following list details the main contributions of this work.

- In this paper, we optimize the radical isogeny formulae in affine and projective versions proposed in earlier studies [21], [30]. We can implement it more efficiently in C by rationalizing the denominator and tailoring the conversion between various curves. In addition, we analyze the radical 3- and 4-isogeny formula in [7] from an implementation perspective. Through these studies, we present the optimized C implementation results of CSIDH with the N-isogeny ($N \in \{2, 3, 5, 7\}$).
- We review the quantum complexity of CSIDH and derive CSIDH parameters that satisfy NIST security Level 1 according to the power of the quantum adversary. For the first time, we provide the C implementation result of CSIDH with the sliding window method, improving the cost of field exponentiation, a core component of computing radical isogenies. Through several experiments, we conclude that using only the radical 2-isogeny is better with a larger prime field. Except for CSIDH-512, using only the radical 2-isogeny for all parameters improves performance by 6% to 10%. The results of the implementation are presented in Section IV.

B. ORGANIZATION

This paper is organized as follows. Section II introduces the required background. Next, Section III briefly details the radical isogeny and presents the optimization results for degrees of 2, 3, 4, 5, and 7 for implementation. The implementation results are presented in Section IV, and we draw conclusions in Section V.

II. PRELIMINARY

This section introduces two types of Montgomery elliptic curves. Then, CSIDH and the idea of radical isogeny are presented.

A. MONTGOMERY CURVE AND TWEAKED MONTGOMERY CURVE

We let K be a field with characteristics not equal to 2 or 3. The Montgomery curves over K are defined by the following equation:

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

where $b(a^2 - 4) \neq 0$. Throughout the paper, an elliptic curve in the above form is called the Montgomery$^+$ curve. When $b = 1$, we express it as M_a. The tweaked Montgomery curves over K are denoted by

$$M^t_{a,b} : by^2 = x^3 + ax^2 - x,$$

where $b(a^2 + 4) \neq 0$. Throughout the paper, an elliptic curve in the above form is called the Montgomery$^-$ curve. When $b = 1$, we express it as M^-_a.

It is well known that point arithmetic on M_a can be efficiently performed using only the x-coordinates. We let $P = (x_p, y_p)$ and $Q = (x_q, y_q)$ be points on M_a such that $x_p \neq x_q$, and $P - Q = (x_{p-q}, y_{p-q})$. Then, the x coordinates of their sum $P+Q$, denoted as x_{p+q}, and the doubling of $[2]P$, denoted as $x_{[2]P}$, can be computed as follows:

$$x_{p+q} = (x_p x_q - 1)^2/(x_{p-q}(x_p - x_q)^2)$$
$$x_{[2]P} = (x_p^2 - 1)^2/(4x_p(x_p^2 + ax_p + 1)).$$

We can induce a similar formula for a Montgomery$^-$ curve, M^-_a [22]. We let $P = (x_p, y_p)$ and $Q = (x_q, y_q)$ be points on M^-_a such that $x_p \neq x_q$, and $P - Q = (x_{p-q}, y_{p-q}$. Then, the x coordinates of their sum $P + Q$, denoted as x_{p+q}, and the doubling of $[2]P$, denoted as $x_{[2]P}$, can be computed as follows:

$$x_{p+q} = (x_p x_q + 1)^2/(x_{p-q}(x_p - x_q)^2)$$
$$x_{[2]P} = (x_p^2 + 1)^2/(4x_p(x_p^2 + ax_p - 1)).$$

As defined in the above equations, the elliptic curve arithmetic formula on M^-_a is similar to the case of M_a, except for some sign flips in the numerator. However, these sign flips cause changes in the computational costs when using projective coordinates and projective curve coefficients for implementation. In addition, as the isogeny formula is induced using the differential addition formula, the elliptic curve arithmetic and isogeny on M^-_a are slower than on M_a.

B. CSIDH PROTOCOL AND SECURITY

1) CSIDH PROTOCOL

The CSIDH is an isogeny-based Diffie–Hellman-like key exchange protocol proposed by Castryck et al. [16] and uses commutative group action on supersingular elliptic curves defined over a finite field $\mathbb{F}_p$. We let $\mathcal{O}$ be an imaginary quadratic order and $\mathcal{E}\ell\ell_p(\mathcal{O})$ denote the set of elliptic curves defined over $\mathbb{F}_p$ with the endomorphism ring $\mathcal{O}$.

It is well known that the class group $Cl(\mathcal{O})$ acts freely and transitively on $\mathcal{E}\ell\ell_p(\mathcal{O})$. This group action is represented by $[\mathfrak{a}]E$, where $E \in \mathcal{E}\ell\ell_p(\mathcal{O})$ and an ideal class $[\mathfrak{a}] \in Cl(\mathcal{O})$. We let $p = f \cdot \prod_{i=1}^n \ell_i - 1$, where ℓ_i values are small, distinct odd primes. We let E be a supersingular elliptic curve over $\mathbb{F}_p$ such that $\text{End}_p(E) = \mathbb{Z}[\pi]$, where $\text{End}_p(E)$ is the endomorphism ring of E over $\mathbb{F}_p$ and $\pi = \sqrt{-p}$. Note that $\text{End}_p(E)$ is a commutative subring of the quaternion maximal order $\text{End}(E)$. Then, the trace of Frobenius is zero; hence, $\#E(\mathbb{F}_p) = p + 1$.

As $\pi^2 - 1 = 0 \mod \ell_i$, the ideal $\ell_i\mathcal{O}$ splits as $\ell_i\mathcal{O} = \mathfrak{l}_i\bar{\mathfrak{l}}_i$, where $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\bar{\mathfrak{l}}_i = (\ell_i, \pi + 1)$. The group action $[\mathfrak{l}_i]E$ (resp. $[\bar{\mathfrak{l}}_i]E$) is computed via the isogeny $\phi_{\mathfrak{l}_i}$ (resp. $\phi_{\bar{\mathfrak{l}}_i}$) over $\mathbb{F}_p$ (resp. $\mathbb{F}_{p^2}$) using Vélu's formulas.

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

2) QUANTUM SECURITY OF CSIDH

In [5], the quantum security of CSIDH was thoroughly investigated. They revealed that the quantum security of CSIDH depends on the size of the prime field, not on the size of the private key exponent. Hence, to achieve a 128-bit quantum security level, the authors recommended using a prime field of at least 4096 bits. In [5], a 4096-bit prime is presented using 417 small primes. Using all 417 primes for a group action degrades the performance and exceeds the target classical security level.

The meet-in-the-middle type of attack is the best-known classical attack; thus, based on the complexity of this attack, the number of primes to be used varies according to the maximum value of the private key exponent. For example, for a constant-time CSIDH using the method in [19], if the maximum value of the private key exponent is 5, then we can use the 64 smallest primes. If the maximum value of the private key exponent is 1, then we can use the 139 smallest primes. The group action of CSIDH-4096 using the method in [19] takes approximately 23 gigacycles. For details on the quantum analysis, please refer to [5].

C. RADICAL ISOGENIES

Castryck et al. proposed an efficient method to compute small-degree isogenies in [21]. Computing an ℓ-isogeny from an elliptic curve $E(\mathbb{F}_p)$ consists of two steps in CSIDH. First, a point P over $\mathbb{F}_p$ of order ℓ is generated. Second, an isogenous curve $E(\mathbb{F}_p)/\langle P \rangle$ is generated.

To generate a kernel of a given order, a random point Q is selected in $\mathbb{F}_p$, which costs approximately $1.5 \log p$ field multiplications, and is multiplied by the cofactor $k = \#E(\mathbb{F}_p)/\ell$, which costs approximately $11 \log p$ field multiplications.

If $P = [k]Q$ equals the identity, another random point is selected to repeat the process. Hence, generating a kernel is a painstaking process, especially for small torsion points where the failure probability is $1/\ell$ [18], [21].

Thus, when computing ℓ_i-isogenies for $1 \leq i \leq n$, it is more efficient to sample a $\prod_{i=1}^{n} \ell_i$-torsion point and push it through the isogeny to create a chain of isogenies of degrees $\ell_1, \ldots, \ell_n$ than to generate ℓ_i-torsion points for each ℓ_i-isogeny. Nevertheless, the probability of failure is higher when creating a small torsion point; therefore, more random points are selected than are needed.

In [21], they proposed an innovative approach to construct a formula to compute chains of n-isogenies for small n. For an elliptic curve E, we let $\phi : E \rightarrow E'$ be an n-isogeny, where $\ker(\phi) = \langle P \rangle$ for a n-torsion point P on E. The aim is to express the n-torsion point P' on E' in terms of the coefficients of E and the coordinates of P. Then, by composing an isogeny $E' \rightarrow E'/\langle P \rangle$ with ϕ, we obtain an isogeny of degree n^2. More explicitly, they applied the fact that an elliptic curve E over a field K with a K-rational point P of order $n \geq 4$ can be represented by the Tate normal form:

$$E : y^2 + (1-c)xy - by = x^3 - bx^2, \quad P = (0, 0),$$

for $b, c \in K$. Then, using Vélu's formula, we can compute the isogenous curve $E' = E/\langle P \rangle$. The n-torsion point P' on E' can be expressed in terms of the coefficients of E and coordinates of P through the corresponding dual isogeny $E' \rightarrow E$. Then, the composition $E \rightarrow E' \rightarrow E'/\langle P' \rangle$ is an isogeny of order n^2. This method allows for computing chains of n-isogenies of arbitrary length and requires only one n-torsion point for the first step.

In the next section, we specifically state the formula for radical isogeny of degrees 2, 3, 4, 5, and 7, which we use to implement CRADS$_n$. For general formula details, please refer to [21].

III. INTEGRATION AND OPTIMIZATION OF RADICAL ISOGENIES

This section presents the optimization techniques for implementing radical isogenies. To exploit radical isogenies for applications in CSIDH-based algorithms, we chose radical isogenies of degrees 2, 3, 4, 5, and 7 for the following reasons. Other than radical 2- or 4-isogenies, to compute an n^e-isogeny, we must have one n-torsion point to start the process. Hence, using m different degrees of radical isogeny requires processing m torsion point generations over a finite field, which is costly. Although this can be minimized using the Elligator method in [20], the advantage of the radical isogeny is that it can minimize the number of randomly generated points with a certain order. However, the radical isogeny formula itself is costly because it requires n-th root computation (exponentiation in this setting). Additionally, as the radical isogeny formula becomes more complicated as the degree increases, we infer that 7 is the upper bound for CSIDH and implementation in C.

In [30], it was noted that using projective curve coefficients for computing radical isogenies is more efficient because it can reduce inversions during the computation of a chain of isogenies. As this applies to radical isogeny of degrees 4, 5, and 7, we apply the optimized version of the following formulas. Moreover, we tailor the transformation between forms of elliptic curves for further optimization. In [7], Onuki and Moriya proposed new representations of the radical isogeny with degrees 3 and 4. Including these formulae, we analyze the radical isogeny formulae comprehensively from the perspective of implementation.

The notation **M** and **E** refer to field multiplication and exponentiation, respectively, and we assume $1M \approx 1S$. We consider the field inversion and n-th root computation to be field exponentiation.

Remark 1: Recently, further optimization of the radical isogeny formulae was proposed by Castryck et al. [6]. According to the paper, computing the radical N-isogeny was optimized or newly proposed for $N \in \{2, 3, \ldots, 17\} \cup \{19\}$. However, we do not discuss the formulae of [6] because they have not resulted in noticeable improvement, at least in this paper.

A. RADICAL 2^e-ISOGENY

1) RADICAL 2^E-ISOGENY USING THE MONTGOMERY$^-$ CURVE

For the radical 2-isogeny, we briefly define the formula for supersingular elliptic curves E defined over a finite field $\mathbb{F}_p$ with $p \equiv 7 \mod 8$, which is the main field used in CSIDH-based algorithms. Over this prime field, curves (E) can be divided into two groups: those located on the *floor* with the endomorphism ring $\mathbb{Z}[\sqrt{-p}]$ and a unique $\mathbb{F}_p$-rational point of order two or those located on the *surface* with the endomorphism ring $\mathbb{Z}[(1+\sqrt{-p})/2]$ and three distinguished $\mathbb{F}_p$-rational points of order two. These three points of order two are categorized as follows:

- P^-: whose halves have x-coordinates not defined over $\mathbb{F}_p$;
- P_1^+: whose halves are not defined over $\mathbb{F}_p$, but their x-coordinates are; and
- P_2^+: whose halves are defined over $\mathbb{F}_p$.

As denoted in Lemma 9 in [21], using points P_1^+ or P_2^+ allows us to compute the chain of 2-isogenies. Additionally, as stated in Proposition 4 of [22], supersingular elliptic curves with the endomorphism ring $\mathbb{Z}[(1+\sqrt{-p})/2]$ are $\mathbb{F}_p$-isomorphic to the curve M_a^-.

Hence, we optimized the 2^e-isogeny formula in [22]. In [22], an algorithm that computes a chain of 2-isogenies is presented by composing the 2-isogeny formula on Montgomery$^+$ curves and transformations between a Montgomery$^-$ curve and Montgomery$^+$ curve. Step 4 in Algorithm 1 can be rewritten as follows:

$$a \leftarrow 2(a\sqrt{a^2+4} - (a^2+3))$$

Algorithm 1 Computing 2^e-Isogeny on M_a^- Over $\mathbb{F}_p$, With $p \equiv 7 \mod 8$ [22]

```
if $e = 0$ then
    return $a$
else
    $a \leftarrow sign(e) \cdot a$
    $a \leftarrow 2\frac{a-3\sqrt{a^2+4}}{a+\sqrt{a^2+4}}$
    For $i$ from 2 to $e$ do
        $a \leftarrow 2(3 + a(\sqrt{a^2-4} - a))$
    $a \leftarrow \frac{a+3\sqrt{a^2-4}}{\sqrt{2\sqrt{a^2-4}(a+\sqrt{a^2-4})}}$
    return $sign(e) \cdot a$
end if
```

Compared to the direct implementation of Step 4, the above equation saves one inversion. The cost for computing 2^e-isogeny is 5**M**+3**E**+ $(e-1)\cdot$(2**M**+1**E**), where **E** refers to a field exponentiation. Over a finite field $\mathbb{F}_p$, where $p \equiv 3 \mod 4$, for $a \in \mathbb{F}_p$, the square root of a is computed as $a^{(p+1)/4}$, the inverse of a is computed as $a^{(p-2)}$, and the square root inverse is computed as $a^{(p+1)(p-2)/4 \mod p-1}$. As exponentiation dominates the performance of the 2-isogeny, we used the sliding window method.

2) RADICAL 2^E-ISOGENY USING THE TATE NORMAL FORM

When computing the 2^e-isogeny, it is sometimes better to work with a chain of 4-isogenies and compute the $4^{e/2}$-isogeny. This approach is applied to implement SIDH-based algorithms, and we describe the corresponding process as stated in [21]. To use a chain of 4-isogenies, we transformed the Montgomery$^-$ curve into the Tate normal form:

$$E_b : y^2 + xy - by = x^3 - bx^2,$$

where $P = (0, 0)$ is a 4-torsion point on E_b and $b \in \mathbb{F}_p$. We let r be the x-coordinate of the 4-torsion point on the Montgomery$^-$ curve. Then, r is expressed as follows:

$$r = 1/2 \cdot \left(\sqrt{2(a^2+4-a\sqrt{a^2+4})} + \sqrt{a^2+4} - a\right).$$

In addition, b, expressed in terms of r, is

$$b = \frac{(\gamma^2(3\gamma^2+8a\gamma-24)-16)^3}{(\gamma(4\gamma^2+8a\gamma-16))^4},$$

where $\gamma = 2r$. Applying Vélu's formula results in

$$E' : y^2 + xy - by = x^3 - bx^2 + (-5b^2+5b)x \\ + (-3b^3 - 12b^2 + b),$$

where $E' = E/\langle P\rangle$. To compute consecutive 4-isogenies, we transformed a 4-torsion point P' on E' to $(0, 0)$. Then, E' is isomorphic to the following elliptic curve:

$$E' : y^2 + xy - b'y = x^3 - b'x^2,$$

where $b' = -\alpha(4\alpha^2+1)/(2\alpha+1)^4$ for $\alpha = \sqrt[4]{-b}$. If $p \equiv 7 \mod 16$, then $\alpha = -b^\mu$. If $p \equiv 15 \mod 16$, then $\alpha = b^\mu$, where $4\mu \equiv 1 \mod (p-1)/2$. After computing a chain of 4-isogenies, we transformed E' back to the corresponding Montgomery$^-$ curve,

$$a = \frac{3\sqrt{-16b'+1} + 8b' - 1}{\sqrt{-2(\sqrt{-16b'+1} + 8b' - 1)\sqrt{-16b'+1}}}.$$

The computational cost for transforming a Montgomery$^-$ curve to a Tate normal form (i.e., computing b) is 9**M**+3**E**. The cost for computing one 4-isogeny in affine curve coefficients is 3**M**+2**E**. Using projective curve coefficients for computing radical isogenies is more efficient [30], as it can reduce inversions during the computation of a chain of isogenies. We let $\alpha = A/C$ for $A, C \in \mathbb{F}_p$. Then, $-b' = X/Z^4 \in \mathbb{F}_p$, where

$$X = (4A^2 + C^2)AC$$
$$Z = (2A + C).$$

Now, in the next round of computing the 4-isogeny, we must calculate $\sqrt[4]{-b'} = \sqrt[4]{X}/\sqrt[4]{Z^4}$. In projective coordinates, this is equivalent to $(\sqrt[4]{X} : \sqrt[4]{Z^4}) = (\sqrt[4]{X} : Z)$. However, $gcd(4, p-1) = 2$ for the chosen prime field; thus, $\sqrt[4]{Z^4}$ is not unique. Hence, applying the fact that $(X : Z^4) = (XZ^4 : Z^8)$, computing the fourth root results in $(\sqrt[4]{XZ^4} : Z^2)$. Hence, if we map it to $(XZ^4 : Z)$, then $\sqrt[4]{-b'}$ can be computed as $(\sqrt[4]{XZ^4} : Z^2)$, which saves one inversion.

3) RADICAL 2^E-ISOGENY USING THE MONTGOMERY$^+$ CURVE

Onuki and Moriya proposed an optimized representation of the radical 4-isogeny in [7]. If M_a is a Montgomery curve with coefficient $a \in \mathbb{F}_p$, and β is a fourth root of $4(a+2)$, then the 4-isogeny $M_a \to M_{a'}$ can be computed by

$$a' = \frac{(\beta+2)^2}{4\beta(\beta^2+4)} - 2.$$

To compute the 4-isogeny chain, we calculated the intermediate value corresponding to $4(a'+2)$ and computed a' at the end of the radical $4^{e/2}$-isogeny. Thus, the cost of computing the radical 4-isogeny once is 3**M**+2**E**. The computational cost of transforming the modified Montgomery coefficient of the form $4(a+2)$ into the Montgomery coefficient is only $1M + 1A$ through precomputed constants. If e is an odd number, the last 2-isogeny must be computed at the end of the 4-isogeny chains. This 2-isogeny $M_{a'} \to M_{a''}$ is as follows:

$$4(a''+2) = \frac{(\beta+4)^2}{\beta},$$

where β is a square root of $4(a'+2)$. This 2-isogeny can be computed by $1M + 2E$. There is an advantage of not transforming the curve form; therefore, we apply this method in the implementation.

TABLE 1. Computational cost of 2^e-isogeny. `Others` **include transforming curves, etc.**

	2-isogeny	4-isogeny	Others
Sec III-A1	2**M**+1**E**	-	5**M**+3**E**
Sec III-A2 (Affine)	-	3**M**+2**E**	9**M**+3**E**
Sec III-A2 (Projective)	-	8**M**+1**E**	13**M**+4**E**
Sec III-A3	1**M**+2**E**	3**M**+2**E**	1**M**

B. RADICAL 3-ISOGENY

1) RADICAL 3-ISOGENY USING THE WEIERSTRASS CURVE

For a given 3-torsion point Q on M_a, we can transform M_a into an isomorphic curve of the form:

$$E : y^2 + a_1xy + a_3y = x^3,$$

for some $a_1, a_3 \in \mathbb{F}_p$, where Q on M_a is mapped to a point $P = (0, 0)$ on E. We let r be the x-coordinate of Q. Then, a_1 and a_3, expressed in terms of r, are as follows:

$$a_1 = \frac{2ar + 3r^2 + 1}{\sqrt{r(r^2 + ar + 1)}}$$
$$a_3 = 2\sqrt{r(r^2 + ar + 1)}.$$

Applying Vélu's formula to E by letting $\langle P \rangle$ be the kernel results in the 3-isogenous curve $E' = E/\langle P \rangle$. A translation that maps the 3-torsion point Q' on E' to $(0, 0)$ is required to construct a formula for computing the chain of 3-isogenies. Hence, the final curve, obtained by translating Q' to $(0, 0)$, is of the form:

$$E' : y^2 + a'_1xy + a'_3y = x^3,$$

where $a'_1 = -6\alpha + a_1$ and $a'_3 = 3a_1\alpha^2 - a_1^2\alpha + 9a_3$, for $\alpha = \sqrt[3]{-a_3}$. After computing the chain of 3-isogenies, we transform E' back into the corresponding Montgomery curve. The formula for transforming a Weierstrass curve to a Montgomery curve is presented in Magma code in [21]. Like the Weierstrass coefficient $a_2 = a_4 = a_6 = 0$ in the case of a radical 3-isogeny, we specifically optimized for these circumstances. The computational cost for transforming the Montgomery curve to a (close) Tate curve (i.e., computing a_1 and a_3) is 4**M**+2**E**. The cost for computing one 3-isogeny is 2**M**+1**E**, and the cost for transforming a Weierstrass curve back into the Montgomery form is 16**M**+4**E**.

2) RADICAL 3-ISOGENY USING THE MONTGOMERY CURVE

Onuki and Moriya proposed an optimized representation of the radical 3-isogeny in [7]. We let M_a be a Montgomery curve with coefficient $a \in \mathbb{F}_p$ and let α be a cube root of $t(t^2 - 1)$, where t is the x-coordinate of the 3-torsion point of M_a. Then, the 3-isogeny $M_a \to M_{a'}$ and t', the x-coordinate of the 3-torsion point of $M_{a'}$, can be computed by

$$t' = 3t\alpha^2 + (3t^2 - 1)\alpha + 3t^3 - 2t,$$
$$a' = \frac{-3(t')^4 - 6(t')^2 + 1}{4(t')^3}.$$

TABLE 2. Computational cost of 3-isogeny. `Others` **include transforming curves, etc.**

	3-isogeny	Others
Sec III-B1	2**M**+1**E**	20**M**+6**E**
Sec III-B2	4**M**+1**E**	4**M**+1**E**

The computational cost of a 3-torsion point on the image curve is 4**M**+1**E**, slightly more expensive than the Weierstrass version. However, recovering the Montgomery coefficient is much cheaper at 4**M**+1**E**. Thus, for the Weierstrass version to be better than this method, the radical 3-isogeny must be performed at least 2.5 k + 8 times, with $k \approx 1.5 \log_2 p$. Therefore, we implement this radical 3-isogeny.

C. RADICAL 5-ISOGENY

The computation of the radical 5-isogeny follows a process similar to that of the radical 3-isogeny. For a given 5-torsion point Q on M_a, we transformed M_a into an isomorphic curve of the following form:

$$E : y^2 + (1 - b)xy - by = x^3 - bx^2,$$

where Q on M_a is mapped to a point $P = (0, 0)$ on E. If r is the x-coordinate of Q, b is computed as follows:

$$b = -\frac{(4ar^3 + 3r^4 + 6r^2 - 1)^3}{(4r(r^2 + ar + 1))^4}.$$

Applying Vélu's formula to E by letting $\langle P \rangle$ be the kernel results in a 5-isogenous curve $E' = E/\langle P \rangle$. Again, to construct a formula for computing a chain of 5-isogenies, a translation that maps the 5-torsion point Q' on E' to $(0, 0)$ is necessary. Hence, the final curve, obtained by translating Q' to $(0, 0)$, is of the following form:

$$E' : y^2 + (1 - b')xy - b'y = x^3 - b'x^2,$$

where

$$b' = \alpha \frac{\alpha^4 + 3\alpha^3 + 4\alpha^2 + 2\alpha + 1}{\alpha^4 - 2\alpha^3 + 4\alpha^2 - 3\alpha + 1},$$

for $\alpha = \sqrt[5]{b}$. After computing a chain of 5-isogenies, we transformed E' back into a corresponding Montgomery curve. The computational cost for transforming a Montgomery curve into a Tate curve (i.e., computing b) is 8**M**+1**E**. The cost for computing one 5-isogeny is 5**M**+2**E**, and the cost for transforming a Weierstrass curve back into the Montgomery form is 18**M**+4**E**. Similar to the 4-isogeny, using the projective curve coefficient as in [30] can save one exponentiation. If $\alpha = X/Z$, for $X, Z \in \mathbb{F}_p$, then b' can be expressed as $b' = X'/Z'$, where

$$X' = X(X^4 + 3X^3Z + 4X^2Z^2 + 2XZ^3 + Z^4)$$
$$Z' = Z(X^4 - 2X^3Z + 4X^2Z^2 - 3XZ^3 + Z^4).$$

Because $(X' : Z') = (X' : Z'^5)$, only one fifth-root computation is required, which can save an inversion.

TABLE 3. Comparison of the computational cost of radical isogenies.

	[30]			This Work		
	`Mont_to_E`	isogeny	`E_to_Mont`	`Mont_to_E`	isogeny	`E_to_Mont`
2-isogeny	$2\mathbf{M}+2\mathbf{E}$	$2\mathbf{M}+1\mathbf{E}$	$4\mathbf{M}+3\mathbf{E}$	$2\mathbf{M}+1\mathbf{E}$	$2\mathbf{M}+1\mathbf{E}$	$3\mathbf{M}+2\mathbf{E}$
3-isogeny	$4\mathbf{M}+2\mathbf{E}$	$2\mathbf{M}+1\mathbf{E}$	$27\mathbf{M}+5\mathbf{E}$	$4\mathbf{M}+2\mathbf{E}$	$2\mathbf{M}+1\mathbf{E}$	$16\mathbf{M}+4\mathbf{E}$
4-isogeny	$15\mathbf{M}+7\mathbf{E}$	$8\mathbf{M}+1\mathbf{E}$	$8\mathbf{M}+5\mathbf{E}$	$11\mathbf{M}+4\mathbf{E}$	$8\mathbf{M}+1\mathbf{E}$	$7\mathbf{M}+4\mathbf{E}$
`extra_2_isog`	-	$8\mathbf{M}+5\mathbf{E}$	-	-	$2\mathbf{M}+2\mathbf{E}$	-
5-isogeny	$10\mathbf{M}+2\mathbf{E}$	$14\mathbf{M}+1\mathbf{E}$	$31\mathbf{M}+6\mathbf{E}$	$8\mathbf{M}+1\mathbf{E}$	$12\mathbf{M}+1\mathbf{E}$	$22\mathbf{M}+5\mathbf{E}$
7-isogeny	$10\mathbf{M}+2\mathbf{E}$	$26\mathbf{M}+1\mathbf{E}$	$30\mathbf{M}+6\mathbf{E}$	$10\mathbf{M}+1\mathbf{E}$	$18\mathbf{M}+1\mathbf{E}$	$21\mathbf{M}+5\mathbf{E}$

TABLE 4. Computational cost of radical isogenies using affine curve coefficient.

degree	`Mont_to_E`	`ℓ-isogeny`	`E_to_Mont`
2	$2\mathbf{M}+1\mathbf{E}$	$2\mathbf{M}+1\mathbf{E}$	$3\mathbf{M}+2\mathbf{E}$
3	$4\mathbf{M}+2\mathbf{E}$	$2\mathbf{M}+1\mathbf{E}$	$16\mathbf{M}+4\mathbf{E}$
4	$9\mathbf{M}+3\mathbf{E}$	$3\mathbf{M}+2\mathbf{E}$	$3\mathbf{M}+3\mathbf{E}$
5	$8\mathbf{M}+1\mathbf{E}$	$5\mathbf{M}+2\mathbf{E}$	$18\mathbf{M}+4\mathbf{E}$
7	$10\mathbf{M}+1\mathbf{E}$	$10\mathbf{M}+2\mathbf{E}$	$20\mathbf{M}+4\mathbf{E}$

D. RADICAL 7-ISOGENY

For a given 7-torsion point Q on M_a, whose x-coordinate is r, we transformed M_a into an isomorphic curve of the form:

$$\begin{aligned} E: y^2+(-N^2+N+1)xy+(-N^3+N^2)y \\ = x^3+(-N^3+N^2)x^2, \end{aligned}$$

where

$$N = \frac{(r^2(3r^2+4ar+6)-1)^3}{(2(r^2(r^2+2ar+6)+2ar+1))} \cdot \frac{1}{(4r^2+4ar+4)^2(r^4-r^2)}.$$

where N' can be expressed in terms of α for $\alpha = \sqrt[7]{N^5-N^4}$. As N' is too large, we do not explicitly state it in this paper. After computing the chain of 7-isogenies, we transformed E' back into the corresponding Montgomery curve. The computational cost for transforming a Montgomery curve into a Tate curve (i.e., computing N) is $10\mathbf{M}+1\mathbf{E}$. The cost for computing one 7-isogeny is $10\mathbf{M}+2\mathbf{E}$, and the cost for transforming a Weierstrass curve back into a Montgomery form is $20\mathbf{M}+4\mathbf{E}$.

Table 4 lists the computational cost of a radical isogeny of various degrees when the affine curve coefficient is used. In Table 4, `Mont_to_E` refers to the transformation from the Montgomery curve to the Tate normal form for odd-degree isogenies and refers to the transformation from the Montgomery$^+$ curves to the Montgomery$^-$ curves for 2- and 4-isogenies. The `ℓ-isogeny` refers to the computation of the one ℓ-isogeny, where $\ell \in \{2, 3, 4, 5, 7\}$, and `E_to_Mont` refers to the transformation from the Weierstrass or Tate curve to the Montgomery curve for odd-degree isogenies and refers to the transformation from the Montgomery$^-$ curves to the Montgomery$^+$ curves for 2- and 4-isogenies.

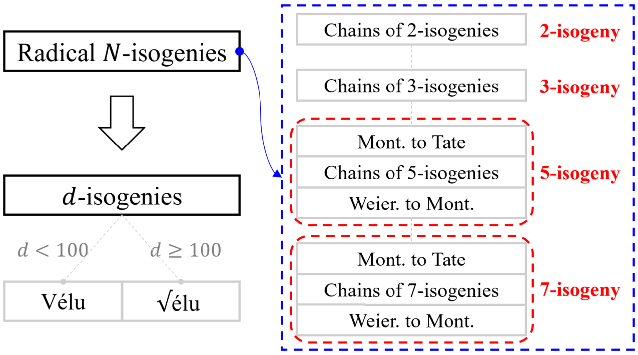

FIGURE 1. Strategy for computing a group action in our implementation.

Table 3 compares the computational cost of radical isogenies in [30] and in this work. In Table 3, the 2- and 3-isogenies refer to the computational cost of a radical isogeny using the affine curve coefficients. Hence, affine curve coefficients are used to implement 2- and 3-isogenies, whereas projective curve coefficients are used to implement 4-, 5-, and 7-isogenies.

The computational cost of `E_to_Mont` combines the transformation from the projective curve coefficients to the affine curve coefficients and the transformation between curves. Last, `extra_2_isog` refers to the additional computation of the 2-isogeny when the 4-isogeny formula is used to compute the 2^e-isogeny for an odd integer e.

Remark 2: Table 3 excludes the multiplication by a small constant as a multiplication count in [30] because multiplication by a constant in radical isogenies can be substituted with addition.

As denoted in Table 4, computing a 4-isogeny once saves only $1\mathbf{M}$ compared to computing a 2-isogeny twice. Moreover, the transformation from the Montgomery$^-$ curve to a certain form of an elliptic curve is more costly in the 4-isogeny case than in the 2-isogeny case; therefore, using the radical 4-isogeny in affine coordinates does not reduce computational cost. As denoted in Table 3, computing a 4-isogeny once saves $1\mathbf{E}$, which can offset the increased computation necessary to change the curve compared to a 2-isogeny.

IV. IMPLEMENTATION RESULTS

This section discusses parameter selection for CSIDH against quantum attacks and presents the implementation results of

TABLE 5. Performance results of the sliding window method with various sizes (in thousands).

Size		-	4	5	6	7	8	9	10
p_{512}	Exp.	157.5	126.4	**124.9**	125.1	129.8	141.3	165.9	217.2
	Savings	-	19.77%	**20.70%**	20.56%	17.57%	10.27%	-5.35%	-37.88%
p_{1024}	Exp.	1,205	948	**928**	937	946	978	1,065	1,249
	Savings	-	21.33%	**22.95%**	22.19%	21.44%	18.81%	11.58%	-3.65%
p_{1792}	Exp.	6,631	5,308	5,159	5,089	**5,082**	5,179	5,445	6,030
	Savings	-	19.95%	22.21%	23.25%	**23.36%**	21.91%	17.89%	9.07%
p_{2048}	Exp.	9,830	7,896	7,736	7,594	**7,579**	7,694	8,034	8,814
	Savings	-	19.67%	21.30%	22.74%	**22.90%**	21.73%	18.27%	10.34%
p_{3072}	Exp.	33,875	27,113	26,424	26,006	**25,817**	25,951	26,668	28,314
	Savings	-	19.96%	21.99%	23.23%	**23.79%**	23.39%	21.27%	16.41%
p_{4096}	Exp.	81,139	64,840	63,069	62,039	**61,462**	61,504	62,598	65,460
	Savings	-	20.09%	22.27%	23.54%	**24.25%**	24.20%	22.85%	19.32%

TABLE 6. The *DW*-cost of solving AES according to **MAXDEPTH** ($\log_2$ scale).

	MAXDEPTH		
	2^{40}	2^{64}	2^{96}
AES-128	121	97	87
AES-192	185	161	130
AES-256	249	225	194

TABLE 7. The *DW*-cost of CSIDH ($\log_2$ scale).

Prime Length	Minimum Depth	*DW*-cost
p_{512}	40	83
p_{1024}	40	92
p_{1792}	40	103
p_{2048}	40	107
p_{3072}	41	123
p_{4096}	74	153

TABLE 8. Primes of the form: $p = 8(\prod_{i=1}^{n} \ell_i) \cdot \ell_{lf} - 1$, where $\ell_1, \ldots, \ell_n$ are the first n odd prime and ℓ_{lf} refers to last factor of $p + 1$. k is the number of used odd primes and m is derived from [5] for classical security level 1.

p	$\log_2 p$	n	ℓ_{lf}	k	$[-m, m]$
p_{512}	511	72	373	69	$[-4, 4]$
p_{1024}	1022	129	2017	94	$[-2, 2]$
p_{1792}	1791	206	13729	138	$[-1, 1]$
p_{2048}	2047	230	730819	138	$[-1, 1]$
p_{3072}	3070	325	36433	138	$[-1, 1]$
p_{4096}	4095	416	4603	138	$[-1, 1]$

CSIDH using various radical isogenies. From this section onward, CSIDH that uses a radical isogeny up to prime degree n is denoted as CRADS$_n$. We measured the performance of CRADS$_n$ and CSIDH with various parameters. For CSIDH-k, an approximately k-bit prime was used in the implementation. These primes are in the form $p \equiv 7 \mod 8$ to use the radical 2-isogeny.

To estimate the implementation results, we executed CSIDH and CRADS$_n$ with the maximum exponent private key. In addition, to implement large odd-degree isogenies for CSIDH and CRADS$_n$, we used the square root Vélu formula in [23]. In this paper, the square root Vélu formula was applied for isogenies of degrees greater than 100. The strategy for computing a group action in our implementation is summarized in Fig. 1.

Other optimization methods, such as the action strategy, were not applied in the results presented in Tables 9 and 11 to understand the pure influence and availability of radical isogenies. We measured only the group action without validation and averaged over 10 000 rounds. All cycle counts were obtained on a one-core Intel(R) Xeon(R) Gold 6230R CPU at 2.10 GHz, running Ubuntu 22.04 LTS. For the compilation, we used GCC version 11.3.0 with the optimization level -O3.

A. PARAMETER SELECTION FOR CSIDH

As mentioned in Section II-B2, a quantum-subexponential attack occurs against CSIDH. Earlier studies are presented to estimate the quantum complexity of CSIDH [3], [4], [5]. In this paper, we chose parameters with the potential to satisfy NIST security Level 1. With the *DW* cost, which is the product of the quantum circuit depth and width, we estimated the quantum security of each parameter and compared it with AES-128. We used the c-sieve estimator provided by [5], including the CSIDH oracle cost. We refer to the quantum security of AES algorithms according to **MAXDEPTH** from [1] and [2].

Table 6 reveals that decreasing the limit of the quantum depth results in an increase in the *DW* cost of solving AES because the depth limitation does not guarantee enough Grover iterations, resulting in higher costs. Ironically, AES-128 and CSIDH-512/1024 have similar security levels for a stronger quantum adversary. However, in **MAXDEPTH** 2^{40} (a more realistic assumption), CSIDH-3072 is similar to AES-128, indicating that the quantum security level must be comprehensively reviewed by analyzing the development status of quantum computers and various factors in real time.

TABLE 9. Performance results of a group action of CSIDH and CRADS_n with naive interval of exponents (clock cycles).

	CSIDH	CRADS_2	CRADS_3	CRADS_5	CRADS_7
p_{512}	147,836,333	**143,863,814**	145,565,480	148,058,907	149,594,815
p_{1024}	529,089,244	**494,044,783**	509,911,470	528,797,392	545,878,253
p_{1792}	1,809,444,472	**1,700,112,090**	1,775,243,407	1,877,015,531	1,980,599,292
p_{2048}	2,404,674,938	**2,238,218,919**	2,347,104,965	2,505,286,938	2,650,683,612
p_{3072}	5,824,117,095	**5,319,278,183**	5,700,391,598	6,249,727,712	6,747,811,246
p_{4096}	10,993,637,042	**9,916,102,370**	10,861,605,668	12,149,836,002	13,377,742,800

TABLE 10. Modified interval of exponents for CSIDH and CRADS_n.

		CSIDH	CRADS_2	CRADS_3	CRADS_5	CRADS_7
p_{512}	RAD	-	(32, 0, 0, 0)	(32, 32, 0, 0)	(32, 32, 16, 0)	(32, 32, 16, 16)
	ODD	(4 : 70)	(4 : 56, 3 : 13)	(4 : 47, 3 : 21)	(4 : 40, 3 : 27)	(4 : 33, 3 : 33)
p_{1024}	RAD	-	(12, 0, 0, 0)	(12, 12, 0, 0)	(12, 12, 12, 0)	(12, 12, 12, 12)
	ODD	(2 : 95)	(2 : 91, 1 : 3)	(2 : 87, 1 : 6)	(2 : 83, 1 : 9)	(2 : 79, 1 : 12)
p_{1792}	RAD	-	(4, 0, 0, 0)	(4, 4, 0, 0)	(4, 4, 4, 0)	(4, 4, 4, 4)
	ODD	(1 : 139)	(1 : 137)	(1 : 136)	(1 : 135)	(1 : 134)
p_{2048}	RAD	-	(4, 0, 0, 0)	(4, 4, 0, 0)	(4, 4, 4, 0)	(4, 4, 4, 4)
	ODD	(1 : 139)	(1 : 137)	(1 : 136)	(1 : 135)	(1 : 134)
p_{3072}	RAD	-	(4, 0, 0, 0)	(4, 4, 0, 0)	(4, 4, 4, 0)	(4, 4, 4, 4)
	ODD	(1 : 139)	(1 : 137)	(1 : 136)	(1 : 135)	(1 : 134)
p_{4096}	RAD	-	(4, 0, 0, 0)	(4, 4, 0, 0)	(4, 4, 4, 0)	(4, 4, 4, 4)
	ODD	(1 : 139)	(1 : 137)	(1 : 136)	(1 : 135)	(1 : 134)

Thus, we considered p_{512} to p_{4096} for candidates for quantum security Level 1 and experimented.

B. PERFORMANCE OF CSIDH AND CRADS_n

In this implementation, all primes are of the form $p = 8(\prod_{i=1}^{n} \ell_i) \cdot \ell_{lf} - 1$. We set the basic parameters to satisfy classical security Level 1, as listed in Table 8. In this field, we chose a supersingular Montgomery$^+$ curve, $M_0^+ : y^2 = x^3 + x$, as a base curve for CSIDH and a Montgomery$^-$ curve, $M_0^- : y^2 = x^3 - x$, as a base curve for CRADS_n. Field exponentiation is the core operation of radical isogeny; thus, we used the sliding window method for effective exponentiation. The window sizes of this implementation are 5 and 7, obtained from the results in Table 5.

As a result of Section III, we used the radical 2-/4- and 3-isogenies of Onuki's method with some adjustments and used the projective radical 5- and 7-isogenies in this paper. The performance of the group action and comparison results of CSIDH and CRADS_n are presented in Table 9. The parameter settings are based on Table 8, and radical isogenies are used only for a fixed m.

As indicated in Table 9, even considering that we do not adjust the interval of radical isogenies, it seems inefficient to use odd-degree radical isogenies. Moreover, as the size of the prime field increases, the inefficiency of odd-degree radical isogenies also increases because odd radical isogenies still require the point sampling process and have various operations that require considerable field exponentiation. We executed further experiments with modified intervals to determine the optimal exponents.

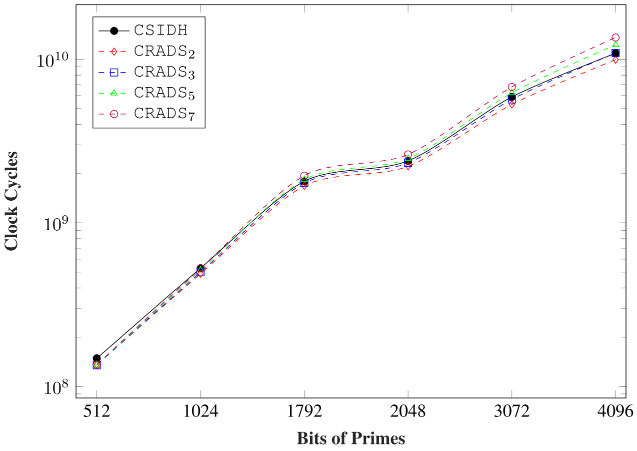

FIGURE 2. Performance results of a group action of CSIDH and CRADS_n (a logarithmic chart of Table 11).

In Table 10, (e_2, e_3, e_5, e_7) in the RAD row indicate that we ran CSIDH/CRADS_n using radical 2^{e_2}-, $3^{e_3}-$, $5^{e_5}-$, 7^{e_7}-isogenies. Further, $(m : k)$ in the ODD row means that each k odd-degree isogeny is iterated at most m times, respectively. For example, a group action of CRADS_7 with prime p_{512} is computed over the following interval:

$$[-32, 32]^2 \times [-16, 16]^2 \times [-4, 4]^{33} \times [-3, 3]^{33}.$$

As listed in Table 11, CRADS_2 using the window method leads to a 6% to 10% performance improvement in all prime fields. In p_{512}, where the cost of exponentiation is relatively small, CRADS_n $(n \in \{3, 5, 7\})$ outperforms CRADS_2.

TABLE 11. Performance results of a group action of CSIDH and CRADS$_n$ with Table 10 (clock cycles).

		CSIDH	CRADS$_2$	CRADS$_3$	CRADS$_5$	CRADS$_7$
p_{512}	Action	148,530,151	136,240,058	**134,781,850**	134,868,137	135,761,351
	Savings	-	8.27%	**9.26%**	9.20%	8.60%
p_{1024}	Action	528,347,977	**488,714,895**	498,910,269	513,950,599	526,513,227
	Savings	-	**7.50%**	5.57%	2.72%	0.35%
p_{1792}	Action	1,796,307,805	**1,684,476,018**	1,752,154,453	1,840,709,316	1,944,917,931
	Savings	-	**6.23%**	2.46%	-2.47%	-8.27%
p_{2048}	Action	2,396,311,058	**2,218,584,796**	2,326,043,704	2,471,083,333	2,621,629,215
	Savings	-	**7.42%**	2.93%	-3.12%	-9.40%
p_{3072}	Action	5,892,554,015	**5,299,667,399**	5,689,704,537	6,239,020,198	6,783,498,775
	Savings	-	**10.06%**	3.44%	-5.88%	-15.12%
p_{4096}	Action	10,892,071,309	**9,950,062,342**	10,930,654,241	12,268,263,256	13,607,725,933
	Savings	-	**8.65%**	-0.35%	-12.63%	-24.93%

However, as the size of the prime field increases, using only a radical 2-isogeny makes CSIDH more efficient.

V. CONCLUSION

There is a risk that a quantum-subexponential attack can change CSIDH parameters at any time. Thus, studying various optimization techniques is valuable. This paper analyzed the optimal use of radical isogenies to implement CSIDH over various prime fields. In this regard, we further optimized the radical isogeny formulae in [21] and [30] and compared them with the results of previous studies.

However, according to the results, odd-degree radical isogenies appear impractical for implementing large CSIDH due to the inefficiency of exponentiation in a large finite field. Nevertheless, the experiments demonstrate that a radical 2-isogeny is still valuable, leading to a 6% to 10% improvement, although other optimizing methods are not considered.

Finally, as the size of the finite field must be increased to resist quantum attacks, more research and optimization are required to use radical isogenies effectively. In particular, a more effective approach to utilizing radical isogeny will be achieved by combining it with previously studied techniques, such as those discussed in [31], to derive the optimal group action strategies. We expect our study to form the basis for extension to more practical use of a radical isogeny in the future.

REFERENCES

[1] J. H. Davenport and B. Pring, "Improvements to quantum search techniques for block-ciphers, with applications to AES," in *Proc. 27th Int. Conf. Sel. Areas Cryptograph.* Halifax, NS, Canada: Springer, 2021, pp. 360–384.

[2] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing Grover oracles for quantum key search on AES and LowMC," in *Pro. 39th Annu. Int. Conf. Theory EUROCRYPT*. Zagreb, Croatia: Springer, 2020, pp. 280–310.

[3] X. Bonnetain and A. Schrottenloher, "Quantum security analysis of CSIDH," in *Pro. 39th Annu. Int. Conf. Theory EUROCRYPT*. Zagreb, Croatia: Springer, 2020, pp. 493–522.

[4] C. Peikert, "He gives C-sieves on the CSIDH," in *Pro. 39th Annu. Int. Conf. Theory EUROCRYPT*, Zagreb, Croatia: Springer, 2020, pp. 463–492.

[5] J. Chávez-Saab, J.-J. Chi-Domínguez, S. Jaques, and F. Rodríguez-Henríquez, "The SQALE of CSIDH: Sublinear Vélu quantum-resistant isogeny action with low exponents," *J. Cryptograph. Eng.*, vol. 12, no. 3, pp. 349–368, Sep. 2022.

[6] W. Castryck, T. Decru, M. Houben, and F. Vercauteren, "Horizontal racewalking using radical isogenies," in *Proc. 28th int. Conf. Theory Appl. Cryptol. Inf. Sec.* Taipei, Taiwan: Springer, 2022, pp. 67–96.

[7] H. Onuki and T. Moriya, "Radical isogenies on Montgomery curves," in *Pro. 25th IACR Int. Conf. Prac. Theory Pub. Key Cryptograph.* Virtual Event: Springer, 2022, pp. 473–497.

[8] T. B. Fouotsa, "SIDH with masked torsion point images," in *Cryptol. ePrint Arch.*, 2022.

[9] T. Moriya, "Masked-degree SIDH," in *Cryptol. ePrint Arch.*, 2022, pp. 1–11.

[10] T. B. Fouotsa, T. Moriya, and C. Petit, "M-SIDH and MD-SIDH: Countering SIDH attacks by masking information," in *Cryptol. ePrint Arch.*, 2023, pp. 1–26.

[11] W. Castryck and T. Decru, "An efficient key recovery attack on SIDH (preliminary version)," in *Cryptol. ePrint Arch.*, 2022, pp. 1–15.

[12] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *Proc. 4th Int. Workshop PQCrypto*. Taipei, Taiwan: Springer, 2011, pp. 19–34.

[13] A. Stolbunov, "Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves," *Adv. Math. Commun.*, vol. 4, no. 2, pp. 215–235, 2010.

[14] A. Childs, D. Jao, and V. Soukharev, "Constructing elliptic curve isogenies in quantum subexponential time," *J. Math. Cryptol.*, vol. 8, no. 1, pp. 1–29, 2014.

[15] J. M. Couveignes, "Hard homogeneous spaces," in *Cryptol. ePrint Arch.*, 2006, pp. 1–11.

[16] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes, "CSIDH: An efficient post-quantum commutative group action," in *Proc. 24th Int. Conf. Theory Appl. Cryptol. Inf. Sec. ASIACRYPT*, 2018, pp. 395–427.

[17] L. De Feo, J. Kieffer, and B. Smith, "Towards practical key exchange from ordinary isogeny graphs," in *Proc. 24th Int. Conf. Theory Appl. Cryptol. Inf. Sec. ASIACRYPT*, 2018, pp. 365–394.

[18] A. Jalali, R. Azarderakhsh, M. M. Kermani, and D. Jao, "Towards optimized and Constant-time CSIDH on embedded devices," in *Proc. 10th Int. Workshop COSADE*. Darmstadt, Germany: Springer, 2019, pp. 215–231.

[19] H. Onuki, Y. Aikawa, T. Yamazaki, and T. Takagi, "A constant-time algorithm of CSIDH keeping two points," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 103, no. 10, pp. 1174–1182, 2020.

[20] D. J. Bernstein, M. Hamburg, A. Krasnova, and T. Lange, "Elligator: Elliptic-curve points indistinguishable from uniform random strings," in *Proc. ACM SIGSAC Conf. Comput. Commun. Sec.* Berlin, Germany: ACM, 2013, pp. 967–980.

[21] W. Castryck, T. Decru, and F. Vercauteren, "Radical Isogenies," in *Proc. 26th Int. Conf. Theory Appl. Cryptol. Inf. Sec. ASIACRYPT*. Daejeon, South Korea: Springer, 2020, pp. 493–519.

[22] W. Castryck and T. Decru, "CSIDH on the surface," in *Proc. 11th Int. Conf. PQCrypto*. Paris, France: Springer, 2020, pp. 111–129.

[23] D. J. Bernstein, L. De Feo, A. Leroux, and B. Smith, "Faster computation of isogenies of large prime degree," *Open Book*, vol. 4, no. 1, pp. 39–55, Dec. 2020.

[24] W. Beullens, T. Kleinjung, and F. Vercauteren, "CSI-FiSh: Efficient isogeny based signatures through class group computations," in *Proc. 25th Int. Conf. Theory Appl. Cryptol. Inf. Sec. ASIACRYPT*. Kobe, Japan: Springer, 2019, pp. 227–247.

[25] T. Kawashima, K. Takashima, Y. Aikawa, and T. Takagi, "An efficient authenticated key exchange from random self-reducibility on CSIDH," in *Proc. 23rd Int. Conf. ICISC*, Seoul, South Korea: Springer, 2020, pp. 55–84.

[26] M. Meyer and S. Reith, "A faster way to the CSIDH," in *Proc. 19th Int. Conf. Cryptol. INDOCRYPT*. New Delhi, India: Springer, 2020, pp. 137–152.

[27] M. Meyer, S. Reith, and F. Campos, "On hybrid SIDH schemes using Edwards and Montgomery curve arithmetic," in *Cryptol. ePrint Arch.*, 2017, pp. 1–12.

[28] S. Kim, K. Yoon, J. Kwon, Y.-H. Park, and S. Hong, "New hybrid method for isogeny-based cryptosystems using Edwards curves," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1934–1943, Mar. 2020.

[29] C. Costello, "B-SIDH: Supersingular isogeny Diffie–Hellman using twisted torsion," in *Proc. 26th Int. Conf. Theory Appl. Cryptol. Inf. Sec. ASIACRYPT*. Daejeon, South Korea: Springer, 2020, pp. 440–463.

[30] J. J. Chi-Domínguez and K. Reijnders, "Fully projective radical isogenies in constant-time," in *Proc. Cryptograph. Track RSA Conf.*, 2022, pp. 73–95.

[31] J.-J. Chi-Domínguez and F. Rodríguez-Henríquez, "Optimal strategies for CSIDH," *Adv. Math. Commun.*, vol. 16, no. 2, p. 383, 2020.

DONGHOE HEO received the B.A. degree in mathematics from Hanyang University, in 2019. He is currently pursuing the Ph.D. degree with the Graduate School of Information Security, Korea University. His Ph.D. research focuses on post-quantum cryptography, especially isogeny-based cryptosystems and quantum algorithms.

SUHRI KIM received the B.A. degree in mathematics and the M.A. degree in information security from Korea University, in 2014 and 2016, respectively, and the Ph.D. degree from the Graduate School of Information Security, Korea University, in 2020. She is currently an Associate Professor with the School of Mathematics, Statistics and Data Science, Sungshin Women's University. Her research interests include post-quantum cryptography and efficient computations for isogeny-based cryptosystems.

SEOKHIE HONG received the M.S. and Ph.D. degrees in mathematics from Korea University, in 1997 and 2001, respectively. From 2000 to 2004, he was with Security Technologies Inc. Subsequently, he conducted Postdoctoral Research with COSIC at KU Leuven, Belgium, from 2004 to 2005, after which he joined the Graduate School of Cyber Security, Korea University. His research interests include cryptography, public and symmetric cryptosystems, hash functions, and MACs.