An efficient collision attack on Castryck-Decru-Smith's hash function

Ryo Ohashi[1] and Hiroshi Onuki[2]

[1] The University of Tokyo, Japan. ryo-ohashi@g.ecc.u-tokyo.ac.jp
[2] The University of Tokyo, Japan. hiroshi-onuki@g.ecc.u-tokyo.ac.jp

Abstract. In 2020, Castryck-Decru-Smith constructed a hash function, using the $(2,2)$-isogeny graph of superspecial principally polarized abelian surfaces. In their construction, the initial surface was chosen from vertices very "close" to the square of a supersingular elliptic curve with a known endomorphism ring. In this paper, we introduce an algorithm for detecting a collision on their hash function. Under some heuristic assumptions, the time complexity and space complexity of our algorithm are estimated to be $\widetilde{O}(p^{3/10})$ which are smaller than the complexity $\widetilde{O}(p^{3/2})$ the authors claimed to be necessary to detect a collision, where p is the characteristic of the base field. In particular case where p has a special form, then both the time and space complexities of our algorithm are polynomial in $\log p$. We implemented our algorithm in Magma, and succeeded in detecting a collision in 17 hours (using 64 parallel computations) under a parameter setting which the authors had claimed to be 384-bit secure.

Keywords: Hash function · Isogeny-based cryptography · Superspecial abelian surface · Kani's lemma

1 Introduction

Many authors have been investigated *isogeny-based cryptosystems*, as one of the candidates for post-quantum cryptography. In recent years, some isogeny-based public key encryption and signature schemes utilizing not only isogenies between elliptic curves but also ones between abelian surfaces have been proposed such as (Q)FESTA [3,33], SCALLOP-HD [9], and SQIsign2D [2,34,14]. Specifically, these schemes use *Kani's lemma* [26] to compute non-smooth degree isogenies between supersingular elliptic curves.

On the other hand, there are several studies on cryptographic hash functions using isogenies. One example is the hash function proposed by Charles, Lauter, and Goren [8], which is called *CGL hash function*. In its construction, they used the ℓ-isogeny graphs of supersingular elliptic curves over $\overline{\mathbb{F}}_p$ with a prime $\ell \neq p$. The choice of the initial curve E_{start} in CGL hash function is very important: if the endomorphism ring of E_{start} is known, a collision can be found in polynomial time in $\log p$ by [15]. We refer to [1] for how to generate the secure initial curve. In the case where we appropriately choose E_{start}, the collision resistance of CGL hash function is considered to have the same security as its pre-image resistance, solved in time complexity $\widetilde{O}(p^{1/2})$ and polynomial space complexity [16, §6.2].

Takashima [36] first proposed a 2-dimensional version of CGL hash function, using the entire $(2,2)$-isogeny graph of superspecial principally polarized abelian surfaces defined over $\overline{\mathbb{F}}_p$. However, it was later pointed out by Flynn-Ti [19] that this graph has many short cycles, which unfortunately indicates that Takashima's hash function is not collision-resistant. Castryck, Decru, and Smith [7] improved it by imposing a restriction on the edges, and they proposed a new hash function (we will call it *CDS hash function* hereafter).

The initial surface A_0 in CDS hash function is determined via a deterministic "short" path from the square of the supersingular elliptic curve E_0 with a known endomorphism ring (see Section 4 for its specific construction) unlike CGL hash function. Nevertheless, breaking the security of CDS hash function is considered computationally hard. To be specific, the authors claimed in [7, Section 7.4] that the collision resistance of CDS hash function also has the same levels of security as its pre-image resistance, which are solved in complexity $\widetilde{O}(p^{3/2})$. We note that a message causing an error can be found in time complexity $\widetilde{O}(p)$ and polynomial space complexity, as pointed out by Costello-Smith (see Remark 3.2 for details).

In this paper, we give a collision attack for CDS hash function, which is more efficient than the existing attacks:

Theorem 1.1. *Under either the heuristics in [20, Lemma 3] or the Generalized Riemann Hypothesis (GRH) together with Assumptions 5.1 and 5.2, there exists an algorithm for detecting a collision on CDS hash function with time and space complexities $\widetilde{O}(p^{3/10})$, where p is the characteristic of the base field.*

The key point of our algorithm is generating a $(2^e, 2^e)$-isogeny $A_0 \to E_1 \times E_2$ for an integer $2^e \approx p \log p$ where E_1 and E_2 are elliptic curves. For the purpose, we use Kani's lemma (Theorem 2.2). All the vertices adjacent to the product of two elliptic curves induce a multiple edge, and hence such an isogeny $A_0 \to E_1 \times E_2$ causes a desired collision as shown in Corollary 2.1.

The complexity of our algorithm depends on the size of the field of definition of the 2^e-torsion points in the supersingular elliptic curve $E_0 : v^2 = u^3 + 1$ whose j-invariant is 0. In particular, for the characteristic p such that $E_0[2^e] \subset E_0(\mathbb{F}_{p^k})$ with $k = O(1)$, we show that our collision attack is a polynomial-time and space algorithm in Corollary 5.1. In fact for such a 128-bit (resp. 256-bit) prime p, we succeeded in detecting a collision on CDS hash function within 3.02 hours (resp. 16.74 hours) through 64 parallel computational experiments. For details of these experimental results, see Section 6.

The rest of this paper is structured as follows: we begin by reviewing several mathematical knowledge in Section 2, and the specific construction of CDS hash function in Section 3. In Section 4, we propose an algorithm to detect a collision on CDS hash function. In Section 5, we estimate the time and space complexities of our collision attack, and give a countermeasure against this. We describe the computational results found by implementing and executing our collision attack with Magma Algebra System [5] in Section 6. Finally, we summarize our results and we briefly state future works in Section 7.

2 Preliminaries

In this section, we summarize some mathematical background which will be used in the later sections. All curves and abelian varieties in this section are assumed to be defined over a field of characteristic $p > 3$.

2.1 Deuring correspondence

An *elliptic curve* is a (projective) curve E of genus 1 with the base point $O \in E$. As is well known, the set of points on an elliptic curve E forms an abelian group with its base point O as the identity. An *isogeny* of elliptic curves is a surjective homomorphism between elliptic curves whose kernel is finite. The set $\mathrm{End}(E)$ of all the isogenies from an elliptic curve E into itself together with the trivial map is a ring, which is called the *endomorphism ring* of E. The structure of $\mathrm{End}(E)$ varies significantly depending on whether E is supersingular or not:

Definition 2.1. *An elliptic curve E is said to be* ***supersingular*** *if E does not have any point of order p.*

In the following, we consider the case where $p \equiv 5 \pmod 6$. Then, the endomorphism ring of a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ is isomorphic to a maximal order of the *quaternion algebra*

$$\mathcal{B}_{p,\infty} := \mathbb{Q} + \mathbb{Q}\mathbf{i} + \mathbb{Q}\mathbf{j} + \mathbb{Q}\mathbf{k}, \ \text{ where } \ \mathbf{i}^2 = -3, \ \mathbf{j}^2 = -p, \ \mathbf{k} = \mathbf{ij} = -\mathbf{ji}$$

ramified exactly at p and ∞. For $\alpha = x + y\mathbf{i} + z\mathbf{j} + w\mathbf{k} \in \mathcal{B}_{p,\infty}$, its *reduced norm* equals to $n(\alpha) := \alpha\overline{\alpha} = x^2 + 3y^2 + p(z^2 + 3w^2) \in \mathbb{Q}$.

Example 2.1. It is well-known (cf. [35, Example V.4.4]) that the elliptic curve

$$E_0 : v^2 = u^3 + 1, \quad \text{with } j\text{-invariant} = 0$$

is supersingular if and only if $p \equiv 5 \pmod 6$. The endomorphism ring $\mathrm{End}(E_0)$ is isomorphic to a maximal order

$$\mathcal{O}_0 := \mathbb{Z} + \mathbb{Z}\frac{1+\mathbf{i}}{2} + \mathbb{Z}\frac{\mathbf{j}+\mathbf{k}}{2} + \mathbb{Z}\frac{\mathbf{i}+\mathbf{k}}{3}$$

of the quaternion algebra $\mathcal{B}_{p,\infty}$.

Deuring [13] gave a correspondence between a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and a maximal order $\mathcal{O}$ of the quaternion algebra $\mathcal{B}_{p,\infty}$. Under this correspondence, a left $\mathcal{O}$-ideal I corresponds to the isogeny φ_I with kernel

$$\bigcap_{\alpha \in I} \ker \alpha = \{P \in E \mid \alpha(P) = 0 \text{ for all } \alpha \in I\}$$

whose degree equals to $n(I) := \gcd(\{n(\alpha) \mid \alpha \in I\})$.

Supersingular elliptic curves	Quaternions
an endomophism $\in \mathrm{End}(E)$	a quaternion $\in \mathcal{O}$
an isogeny $\varphi_I : E \to E'$	a left $\mathcal{O}$-ideal and right $\mathcal{O}'$-ideal I
the degree of $\varphi_I : E \to E'$	the (reduced) norm of I

Table 1. Deuring correspondence

Algorithm 1 RepresentInteger

```
Input: An integer $N > p$
Output: A random endomorphism $\alpha \in \mathbb{Z} + \mathbb{Z}\mathbf{i} + \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$ whose degree equals to $N$
1: Sample random integers $z, w$ such that $p(z^2 + 3w^2) \leq N$
2: Let $m \leftarrow N - p(z^2 + 3w^2)$ and find its 100-smooth part $m'$ using trial division
3: if $m/m'$ is not prime or $\mathsf{Cornacchia}(3, m) = \perp$ then
4:     Go back to Step 1
5: end if
6: Let $(x, y) \leftarrow \mathsf{Cornacchia}(3, m)$
7: return the endomorphism $\alpha = x + y\mathbf{i} + z\mathbf{j} + w\mathbf{k}$ with the notations in Example 2.1
```

Table 1 shows a summary of this correspondence. In addition, the codomains of the isogenies corresponding to left $\mathcal{O}$-ideals I and J are isomorphic to each other if and only if there exists $\alpha \in (\mathcal{B}_{p,\infty})^{\times}$ such that $I = J\alpha$. Such ideals I and J are called *equivalent*, and denoted by $I \sim J$.

2.2 Existing algorithms on quaternions

In this subsection, we review the existing two algorithms on quaternions, which will be used in later sections. We continue to suppose $p \equiv 5 \pmod{6}$, and fix an isomorphism $\mathrm{End}(E_0) \cong \mathcal{O}_0$ as described in Example 2.1.

Firstly, we recall RepresentInteger proposed by Kohel-Lauter-Petit-Tignol [27]. It takes an integer $N > p$ as input, and outputs an endomorphism of the form

$$\alpha = x + y\mathbf{i} + z\mathbf{j} + w\mathbf{k} \in \mathrm{End}(E_0), \quad \text{with } x, y, z, w \in \mathbb{Z} \tag{2.1}$$

whose degree is equal to N. For the reader's convenience, we give a pseudocode of RepresentInteger in Algorithm 1. We denote by $\mathsf{Cornacchia}(d, m)$ an algorithm which returns a pair (x, y) of integers satisfying $x^2 + dy^2 = m$ if it exists, or $\perp$ otherwise (cf. [10, Algorithm 1.5.2]) provided we know the factorization of m.

Remark 2.1. In general, the factorization of an integer requires sub-exponential time in the size of the input. Hence, we run $\mathsf{Cornacchia}(3, m)$ only for integers m which can be written as the product of a smooth integer $m' > 0$ and a prime in line 3 of Algorithm 1. This makes Algorithm 1 a polynomial-time algorithm, but as a trade-off, some endomorphisms may fail to be output. By the prime number theory (e.g. [12, Theorem 1.1.4]), one can assume that at least $1/\log N$ of all the endomorphisms in $\mathbb{Z} + \mathbb{Z}\mathbf{i} + \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$ whose degree $= N$ could be the outputs of Algorithm 1, as in the discussion in [33, §2.3].

Next, we consider algorithms for computing the codomain of an isogeny corresponding to a given left $\mathcal{O}_0$-ideal I. For this, we find a left $\mathcal{O}_0$-ideal $J \sim I$ such that the reduced norm of J is power-smooth (i.e. written as ${\ell_1}^{e_1}{\ell_2}^{e_2} \cdots {\ell_r}^{e_r}$ with small primes ℓ_i and integers $e_i > 0$). The isogeny φ_J corresponding to J can be decomposed into

$$\varphi_J = \varphi_r \circ \cdots \circ \varphi_2 \circ \varphi_1, \quad \text{with } \deg \varphi_i = \ell_i^{e_i},$$

and then the desired output is obtained by computing its codomain.

By assuming some heuristics or the Generalized Riemann Hypothesis (GRH), the complexity of the above algorithm can be estimated as follows:

Lemma 2.1. *Let I be a left $\mathcal{O}_0$-ideal such that $\log(n(I)) = O(\log p)$. Then, there exists an algorithm for computing the isogeny φ_I corresponding to I, in expected polynomial time in $\log p$ under the heuristics in [20, Lemma 3] or the GRH.*

Proof. First, we assume the heuristics in [20, Lemma 3]. This implies that there exists the algorithm which takes I as input, outputs a left $\mathcal{O}_0$-ideal J equivalent to I of $(\frac{7}{2} + o(1)) \log p$-powersmooth norm, and runs in polynomial time in $\log p$. Second, we assume the GRH instead. Then [38, Theorem 6.4] implies that there exists the algorithm which takes I as input, outputs a left $\mathcal{O}_0$-ideal J equivalent to I of $(\log p)^c$-powersmooth norm where $c \in \mathbb{Z}$ is independent of p, and runs in polynomial rime in $\log p$.

In both cases, we obtain a left $\mathcal{O}_0$-ideal J equivalent to I whose reduced norm is bounded by polynomial in $\log p$. Then, it follows from [20, Lemma 4] that the isogeny φ_J can be computed in expected polynomial time in $\log p$. □

We will refer to the above algorithms in Lemma 2.1 as IdealToIsogeny(I) without distinguishing between them (implicitly choosing one or the other)[*].

2.3 Abelian varieties and their isogenies

An *abelian variety* is a projective algebraic variety which is also a group variety. The *dimension* of an abelian variety is its dimension as an algebraic variety. For two abelian varieties A_1 and A_2 of dimensions g_1 and g_2, their product $A_1 \times A_2$ is an abelian variety of dimension $g_1 + g_2$. Another example of abelian varieties is the Jacobian variety of a curve:

Definition 2.2. *Let g be a positive integer. The divisor classes of degree 0 over a genus-g curve C form an abelian variety of dimension g. This abelian variety is called the **Jacobian variety** of C, and denoted by $\mathrm{Jac}(C)$.*

Torelli's theorem (cf. [32, Corollary 12.2]) tells us that if the Jacobian varieties of two genus-g curves are isomorphic to each other (as principally polarized abelian varieties), then so are the underlying curves.

For an abelian variety A of dimension g and an integer $\ell > 0$, we define

$$A[\ell] \coloneqq \{P \in A \mid \ell P = 0_A\},$$

called the *ℓ-torsion subgroup* of A.

Definition 2.3. *An **$(\ell, \dots, \ell)$-subgroup** of $A[\ell]$ is a maximal subgroup of $A[\ell]$ on which the ℓ-Weil pairing is trivial.*

The number of $(\ell, \dots, \ell)$-subgroups of $A[\ell]$ is known (cf. [11, Lemma 2]) to be

$$N_g(\ell) \coloneqq \prod_{k=1}^{g} (\ell^k + 1). \tag{2.2}$$

[*] Recently, it is reported by [30, Theorem 6.2] that we can compute φ_I corresponding to I in expected polynomial time in $\log p$, assuming access to a factoring oracle.

As in the case of elliptic curves, a surjective homomorphism between abelian varieties which has a finite kernel is said to be an *isogeny* of abelian varieties. In particular, we will consider the following type of isogenies:

Definition 2.4. *An isogeny $A \to B$ of abelian varieties is called an* $\boldsymbol{(\ell,\dots,\ell)}$***-isogeny*** *if its kernel is an $(\ell,\dots,\ell)$-subgroup of $A[\ell]$.*

Abelian varieties A and B are called *isogenous* if there exists an isogeny $A \to B$. All the dimensions of isogenous abelian varieties are the same. Isogenies $A \to B$ with the same kernel are equivalent up to an automorphism of B, and therefore they are identified (in particular, the codomains of isogenies with the same kernel are isomorphic to each other). Then, we will denote the codomain of an isogeny with a domain A and a kernel G as A/G.

Definition 2.5. *Let $\phi_1 : A_0 \to A_1$ and $\phi_2 : A_1 \to A_2$ be two $(\ell,\dots,\ell)$-isogenies of abelian varieties.*

- *If $\ker\phi_2 = \phi_1(A_0[\ell])$ holds, then ϕ_2 is called the* ***dual extension*** *of ϕ_1.*
- *If $\ker\phi_2 \cap \phi_1(A_0[\ell]) = 0$ holds, then ϕ_2 is called a* ***good extension*** *of ϕ_1.*
- *Otherwise ϕ_2 is called a* ***bad extension*** *of ϕ_1.*

Given an $(\ell,\dots,\ell)$-isogeny $\phi_1 : A_0 \to A_1$ of abelian varieties of dimension g, the number of its good extensions is known (e.g. [6, Lemma 2]) to be $\ell^{g(g+1)/2}$. For an integer $n > 0$, any $(\ell^n,\dots,\ell^n)$-isogeny $\phi : A \to B$ can be decomposed into

$$A = A_0 \xrightarrow{\phi_0} A_1 \xrightarrow{\phi_1} A_2 \longrightarrow \cdots \longrightarrow A_{n-1} \xrightarrow{\phi_{n-1}} A_n = B,$$

where ϕ_i is a good extension of ϕ_{i-1} for all $i \in \{1,\dots,n\}$.

We are interested in superspecial abelian varieties, which are generalizations of supersingular elliptic curves:

Definition 2.6. *An abelian variety A is said to be* ***superspecial*** *if A is isomorphic to the product of supersingular elliptic curves (without polarizations).*

The number of isomorphism classes of superspecial principally polarized abelian varieties over $\overline{\mathbb{F}}_p$ of dimension g is known (cf. [17, p. 159]) to be $O(p^{g(g+1)/2})$. In addition, a curve C is also said to be *superspecial* if the Jacobian variety of C is a superspecial abelian variety.

Definition 2.7. *For an integer g and a prime $\ell \neq p$, the* ***superspecial isogeny graph*** *$\mathcal{G}_g(\ell,p)$ is defined as*

- *The vertices are all isomorphism classes of superspecial principally polarized abelian varieties of dimension g.*
- *The edges are all $(\ell,\dots,\ell)$-isogenies between two vertices, by identifying two isogenies with the same kernel as the same edge.*

One can see that $\mathcal{G}_g(\ell,p)$ is an $N_g(\ell)$-regular multigraph, where $N_g(\ell)$ is defined in (2.2). In addition, the following important fact is known:

Theorem 2.1. *The graph $\mathcal{G}_g(\ell,p)$ is connected for all $g \geq 1$ and $\ell \neq p$.*

Proof. See [31, p. 223] for $g = 1$ and [25, Theorem 34] for $g \geq 2$. □

2.4 Kani's lemma

In this paper, we will mainly focus on *abelian surfaces*, that is, abelian varieties of dimension 2. Any principally polarized abelian surface is isomorphic to either

(i) the product of two elliptic curves, or
(ii) the Jacobian variety of a (hyperelliptic) genus-2 curve.

In particular, the latter one is said to be a *simple* abelian surface. It is well-known (cf. [35, Theorem V.4.1]) that the number of isomorphism classes of supersingular elliptic curves is $p/12 + O(1)$, and then there exist about $(p/12)^2 \times 1/2 = p^2/288$ non-simple superspecial abelian surfaces up to isomorphism. On the other hand, Ibukiyama-Katsura-Oort [22, Theorem 3.3] shows that there exist

$$\frac{p^3 + 24p^2 + 141p}{2880} + O(1)$$

simple superspecial abelian surfaces, up to isomorphism.

From the above discussion, the number of vertices in $\mathcal{G}_2(\ell, p)$ is about $p^3/2880$, and among them, approximately $10/p$ of the total is decomposed into a product of elliptic curves. This implies that it is "rare" to reach such a vertex when doing a random walk on $\mathcal{G}_2(\ell, p)$ for large p. Nevertheless, we can generate a path from the product vertex of two supersingular elliptic curves to another product vertex in the graph $\mathcal{G}_2(\ell, p)$ by using *Kani's lemma*: let E_0, E_1, E_2, E_3 be elliptic curves connected by the following commutative diagram

$$\begin{tikzcd}
E_0 \arrow[r, "\varphi_2"] \arrow[d, "\varphi_1"'] \arrow[dr, "f"] & E_2 \arrow[d, "\psi_1"] \\
E_1 \arrow[r, "\psi_2"'] & E_3
\end{tikzcd}$$

where φ_1, ψ_1 are N_1-isogenies and φ_2, ψ_2 are N_2-isogenies with $\gcd(N_1, N_2) = 1$. In this setting, the following result (based on Kani's paper [26]) is known.

Theorem 2.2 ([29, Theorem 1])**.** *With the notations above, the isogeny*

$$\Phi = \begin{pmatrix} \varphi_1 & -\hat{\psi}_2 \\ \varphi_2 & \hat{\psi}_1 \end{pmatrix} : E_0 \times E_3 \to E_1 \times E_2$$

is an $(N_1 + N_2, N_1 + N_2)$*-isogeny with the set* $\{([N_1]P, f(P)) \mid P \in E_0[N_1 + N_2]\}$ *as its kernel.*

If we are given an isogeny $f : E_0 \to E_3$ of degree $= N_1N_2$ such that $N_1 + N_2 = \ell^n$ and $\gcd(N_1, N_2) = 1$, then the codomain of the isogeny Φ starting from $E_0 \times E_3$ which has a kernel

$$\{([N_1]P, f(P)) \mid P \in E_0[\ell^n]\}$$

decomposes into the product of two elliptic curves by Theorem 2.2. In a particular case where both E_0 and E_3 are supersingular, the isogeny Φ constructed in this way provides a path from $E_0 \times E_3$ to another product in the graph $\mathcal{G}_2(\ell, p)$, as desired.

2.5 Vertices around the product of two elliptic curves

Florit-Smith [18] studied the structure of the superspecial isogeny graph $\mathcal{G}_2(2,p)$. In this subsection, let us briefly review some of their results.

Definition 2.8. *The* ***reduced automorphism group*** *of an abelian surface A is defined to be*

$$\mathrm{RA}(A) \coloneqq \mathrm{Aut}(A)/\langle[-1]\rangle$$

where $[-1]$ denotes the multiplication-by-(-1) map.

We use the following three "types" notations for the classes of vertices in $\mathcal{G}_2(2,p)$ borrowed from [4, §2] and [18, §3].

- Type-A: a genus-2 Jacobian with trivial reduced automorphism group.
- Type-I: a genus-2 Jacobian with reduced automorphism group $\cong \mathbb{Z}/2\mathbb{Z}$.
- Type-Π: an elliptic product with reduced automorphism group $\cong \mathbb{Z}/2\mathbb{Z}$.

Remark 2.2. In other words, a Type-Π vertex is written as $E \times E'$ with $E \ncong E'$, where the j-invariants of E and E' are neither 0 nor 1728.

Recall from [18, Figure 5] that the neighbourhoods of Type-Π and Type-I vertices are illustrated as follows:

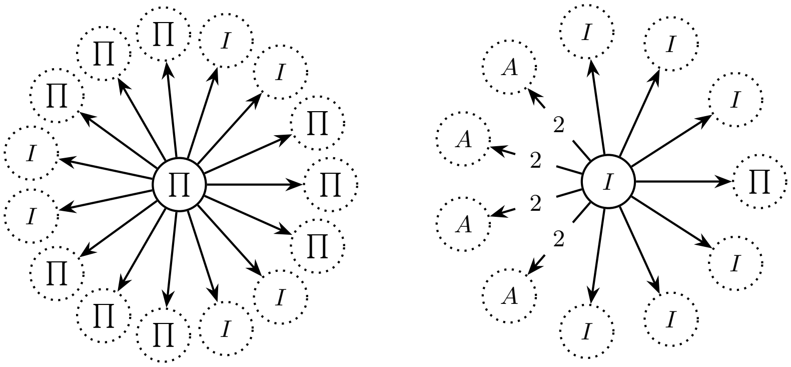


Fig. 1. Neighbourhoods of the general Type-Π and Type-I vertices

In Fig. 1, the solid vertices have definite types, while the dotted vertices have an indicative type. In short, the reduced automorphism group of each dotted vertex could be larger (but this occurs with a *negligible* probability for large p).

Remark 2.3. It follows from [22, Theorem 3.3] or [18, Table 3] that

$$(\text{the number of all the Type-}A\text{ vertices}) = \frac{p^3 - 36p^2 + 381p}{2880} + O(1),$$

$$(\text{the number of all the Type-}I\text{ vertices}) = \frac{p^2 - 18p}{48} + O(1).$$

On the other hand, the number of all the vertices whose reduced automorphism group $\supsetneq \mathbb{Z}/2\mathbb{Z}$ is given as $7p/24 + O(1)$.

Each Type-I vertex is isomorphic to the Jacobian variety of a genus-2 curve

$$C_I : v^2 = (u^2 - 1)(u^2 - \lambda)(u^2 - \mu)$$

with $\lambda, \mu \neq 0, 1$ and $\lambda \neq \mu$. This curve has an automorphism $\sigma : (u, v) \to (-u, v)$ of order 2, and the induced automorphism of $J_I := \mathrm{Jac}(C_I)$ will be also denoted by σ (though this is an abuse of the notation). For a $(2, 2)$-subgroup G of $J_I[2]$, we have the commutative diagram

$$\begin{tikzcd} J_I \arrow[r] \arrow[d, "\sigma"'] & J_I/G \arrow[d, "\cong"] \\ J_I \arrow[r] & J_I/\sigma(G) \end{tikzcd}$$

since $\sigma(G)$ is also a $(2, 2)$-subgroup of $J_I[2]$. In particular case where $G = \sigma(G)$, one can see that σ induces an order-2 automorphism of J_I/G distinct from $[-1]$. This shows us that J_I/G is a Type-A vertex only if $G \neq \sigma(G)$. For our purpose, we show the following statements:

Proposition 2.1. *Under the notations above, let G be a $(2, 2)$-subgroup of $J_I[2]$ such that $G \neq \sigma(G)$. Then, we have that $G \cap \sigma(G) = \{0\}$.*

Proof. All the 2-torsion points of J_I is written as $D_{i,j} := [P_i] - [P_j]$ with $i < j$, where we write

$$\begin{aligned} &P_1 = (1, 0), \quad &&P_2 = (\sqrt{\lambda}, 0), \quad &&P_3 = (\sqrt{\mu}, 0) \\ &P_4 = (-1, 0), \quad &&P_5 = (-\sqrt{\lambda}, 0), \quad &&P_6 = (-\sqrt{\mu}, 0). \end{aligned}$$

It follows from (2.2) that there exist 15 distinct $(2, 2)$-subgroups G of $J_I[2]$, but only 8 of them satisfy $G \neq \sigma(G)$ as follows:

$$\begin{aligned} K_8 &:= \{0, D_{1,5}, D_{2,6}, D_{3,4}\}, \quad & K_9 &:= \{0, D_{1,6}, D_{2,4}, D_{3,5}\}, \\ K_{10} &:= \{0, D_{1,3}, D_{2,4}, D_{5,6}\}, \quad & K_{11} &:= \{0, D_{1,5}, D_{2,3}, D_{4,6}\}, \\ K_{12} &:= \{0, D_{1,6}, D_{2,3}, D_{4,5}\}, \quad & K_{13} &:= \{0, D_{1,2}, D_{3,4}, D_{5,6}\}, \\ K_{14} &:= \{0, D_{1,3}, D_{2,6}, D_{4,5}\}, \quad & K_{15} &:= \{0, D_{1,2}, D_{3,5}, D_{4,6}\}. \end{aligned}$$

We remark that $\sigma(K_8) = K_9$, $\sigma(K_{10}) = K_{11}$, $\sigma(K_{12}) = K_{13}$, and $\sigma(K_{14}) = K_{15}$ (the notations are consistent with [18, §4.5]). Then, for each $G \in \{K_8, \ldots, K_{15}\}$, one can check that $G \cap \sigma(G) = \{0\}$ by a straightforward computation. □

Corollary 2.1. *Let $\gamma : J_A \to J_I$ be an edge in $\mathcal{G}_2(2, p)$ from a Type-A vertex to a Type-I vertex. Then, there exists a good extension $\gamma' : J_I \to J_A$ of γ.*

Proof. Let $\hat{\gamma} : J_I \to J_A$ be the dual extension of γ, and we write its kernel as G. By the definition of the dual extension, we obtain that $G = \gamma(J_A[2])$. Moreover, since $J_I/G = J_A$ is a Type-A vertex, the condition $G \neq \sigma(G)$ holds. Hence, there exists the other $(2, 2)$-isogeny $\gamma' : J_I \to J_A$ with setting its kernel to $\sigma(G)$. Then, it follows from Proposition 2.1 that

$$\ker \gamma' \cap \gamma(J_A[2]) = \sigma(G) \cap G = \{0\},$$

which shows that γ' is a good extension of γ, as desired. □

3 Castryck-Decru-Smith's hash function

In this section, we review the construction of a cryptographic hash function which was proposed by Castryck-Decru-Smith [7] (see Algorithm 2 for its pseudocode). We also discuss the original security of their hash function in Section 3.3.

3.1 Construction

In isogeny-based cryptography, the most famous hash function would be the one by Charles, Lauter, and Goren [8]. Their hash function, often referred to as "CGL hash function", is constructed by using the graph $\mathcal{G}_1(\ell, p)$. Castryck, Decru, and Smith [7] have proposed a variant of CGL hash function using the graph $\mathcal{G}_2(2, p)$ instead. Let us review the construction of their function (we call this "CDS hash function" for simplicity).

Set-up We fix an initial $(2, 2)$-isogeny $\phi_0 : A_{-1} \to A_0$ between two superspecial principally polarized abelian surfaces (we explain later how to choose this). Express a message $m \in \{0, 1\}^*$ as $m_n \cdots m_2 m_1$ in octal with $m_i \in \{0, \ldots, 7\}$.

Walking There are 8 good extensions of ϕ_0, and we label them ${\phi_0}^{(0)}, \ldots, {\phi_0}^{(7)}$ in a deterministic order. Then, define the next isogeny $\phi_1 = {\phi_0}^{(m_1)} : A_0 \to A_1$ and let ${\phi_1}^{(0)}, \ldots, {\phi_1}^{(7)}$ be its 8 good extensions. Next, by setting $\phi_2 = {\phi_1}^{(m_2)}$, repeat this procedure until the last isogeny $\phi_n : A_{n-1} \to A_n$ is obtained.

Output The last vertex A_n must be isomorphic to the Jacobian variety of some genus-2 curve C_n with a very high probability. Then, we output the absolute Igusa invariants (j_1, j_2, j_3) of C_n (see [23] for invariants). We note that j_1, j_2, and j_3 all belong to $\mathbb{F}_{p^2}$, since superspecial C_n has a model defined over $\mathbb{F}_{p^2}$.

In the construction of CDS hash function, let us explain why we must choose next isogenies from the 8 good extensions rather than all the 15 edges in $\mathcal{G}_2(2, p)$. Let $\phi_i : A_{i-1} \to A_i$ be a $(2, 2)$-isogeny between superspecial principally polarized abelian surfaces. If the dual extension of ϕ_i were chosen as the next isogeny ϕ_{i+1}, then we have the diagram in the left of Fig. 2 which leads to a cycle of length 2. In addition, if a bad extensions of ϕ_i were chosen as the next isogeny ϕ_{i+1}, then we have the diagram in the right of Fig. 2 which leads to a cycle of length 4.

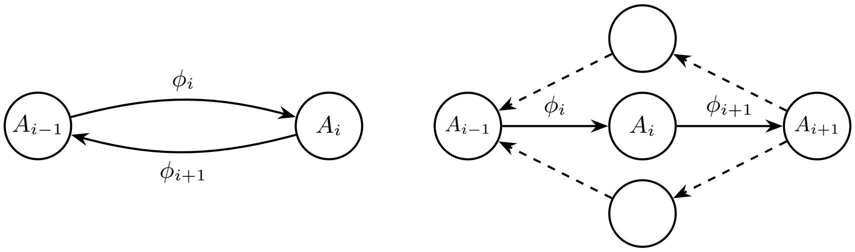


Fig. 2. Dual extension and bad extensions

The existence of such a trivial cycle can easily break the collision resistance of a hash function, and thus this is why we are only allowed choose a good extension of ϕ_i as the next isogeny.

Algorithm 2 CDS_d

```
Input: A message $m \in \{0,1\}^*$
Output: The hash value of $m$, or $\perp$ (failure)
Let $\phi_{-d} : E_0 \times E_0 \to A_{-d}$ be the $(2,2)$-isogeny defined as in Section 3.2
Add $3d$ zeros from the right to $m$, and write it as $m_n \cdots m_1 m_0 \cdots m_{-d+1}$ in octal
for $i = -d+1, \ldots, 0, 1, \ldots, n$ do
    Let $\phi_i : A_{i-1} \to A_i$ be the $m_i$-th good extension of $\phi_{i-1}$ in the graph $\mathcal{G}_2(2,p)$
    if the codomain $A_i$ is isomorphic to a product of elliptic curves then
        return $\perp$
    end if
end for
return the absolute Igusa invariants of the underlying genus-2 curve of $A_n$
```

3.2 Choice of initial isogeny

How to choose the initial isogeny $\phi_0 : A_{-1} \to A_0$ is an important problem, just as with CGL hash function. Let us review the choice of ϕ_0 in the original paper: we suppose that $p \equiv 5 \pmod 6$, then $E_0 : v^2 = u^3 + 1$ is a supersingular elliptic curve over $\mathbb{F}_{p^2}$ as stated in Example 2.1. One can check that

$$\begin{aligned} &G_0 \coloneqq \{(O_0, O_0), (P_0, P_0), (Q_0, R_0), (R_0, Q_0)\}, \\ &\text{with } P_0 \coloneqq (-1, 0),\ Q_0 \coloneqq (1+\omega, 0),\ R_0 \coloneqq (-\omega, 0) \in E_0 \end{aligned}$$

is one of the $(2,2)$-subgroups of $(E_0 \times E_0)[2]$ where ω is a primitive cube root of unity. The codomain of the $(2,2)$-isogeny ϕ_{start} with the kernel G_0 is isomorphic to the Jacobian variety of the superspecial genus-2 curve

$$v^2 = u(u-1)(u+1)(u-2)(u-1/2).$$

Although we would like to set ϕ_{start} as the initial isogeny, it is known [18, §4.13] that among the 8 good extensions of ϕ_{start}, they have only 3 distinct codomains up to isomorphism, which leads to a trivial collision.

To fix this defect, Castryck-Decru-Smith claimed in [7, §7.3] that all we have to do is pad the input $m \in \{0,1\}^*$ with 30 zeros from the right. In the following, we generalize their setting by padding the input m with $3d$ zeros for fixed $d \in \mathbb{N}$. In other words, we consider the deterministic chain of good extensions

$$E_0 \times E_0 \xrightarrow{\phi_{-d}} A_{-d} \xrightarrow{\phi_{-d+1}} A_{-d+1} \longrightarrow \cdots \longrightarrow A_{-1} \xrightarrow{\phi_0} A_0, \qquad (3.1)$$

with setting $\phi_{-d} \coloneqq \phi_{\text{start}}$, $\phi_i \coloneqq {\phi_{i-1}}^{(0)}$ for each $i \in \{-d+1, \ldots, 0\}$, and finally define $\phi_0 : A_{-1} \to A_0$ to be the initial isogeny. For the reader's convenience, we give a pseudocode of (generalized) CDS hash functions in Algorithm 2. Remark that CDS_{10} is the original CDS hash function itself.

Remark 3.1. In Castryck-Decru-Smith's original settings, their function returns an error if we pass through a vertex corresponding to the product of two elliptic curves while walking on the graph (the probability of this happening is negligible for sufficiently large p, as mentioned in the middle of Section 2.4). The authors also presented two alternatives against this in [7, §7.2].

3.3 Original estimate of security

With the same notation in Section 3.1, the pre-image resistance and the collision resistance of CDS hash function are each equivalent to the following mathematical problems, respectively:

Problem 3.1 (Pre-image problem). Given a superspecial genus-2 curve C, find a path of good extensions

$$\begin{tikzcd} A_{-1} \arrow[r, "\phi_0"] & A_0 \arrow[r] & A_1 \arrow[r] & \cdots \arrow[r] & A_{n-1} \arrow[r] & \mathrm{Jac}(C) \end{tikzcd}$$

in the graph $\mathcal{G}_2(2,p)$, where all A_i are simple abelian surfaces.

Problem 3.2 (Collision problem). Find a superspecial genus-2 curve C and two distinct paths of good extensions

$$\begin{tikzcd} A_{-1} \arrow[r, "\phi_0"] & A_0 \arrow[r] & A_1 \arrow[r] & \cdots \arrow[r] & A_{n-1} \arrow[r] & \mathrm{Jac}(C) \end{tikzcd}$$

and

$$\begin{tikzcd} A_{-1} \arrow[r, "\phi_0"] & A_0 \arrow[r] & B_1 \arrow[r] & \cdots \arrow[r] & B_{N-1} \arrow[r] & \mathrm{Jac}(C) \end{tikzcd}$$

in the graph $\mathcal{G}_2(2,p)$, where all A_i and B_i are simple abelian surfaces.

In the original paper [7, §7.4], the authors claimed that the time complexity and space complexity needed for solving the above problems are $\widetilde{O}(p^{3/2})$. Indeed, there are 8^n codomains of all the $(2^n,2^n)$-isogenies starting from A_0 and the size of $\mathcal{G}_2(2,p)$ is asymptotically $O(p^3)$, and therefore we can expect that there exists a $(2^n,2^n)$-isogeny $A_0 \to \mathrm{Jac}(C)$ for n such that $8^n \approx p^3$. Such an isogeny can be computed in time and space complexities $\widetilde{O}(p^{3/2})$, using the meet-in-the-middle algorithm since we need to compute and store $O(8^{n/2}) = O(p^{3/2})$ vertices.

Remark 3.2. If paths passing through the product of elliptic curves are allowed in the above two problems, then these can be solved in the time complexity $\widetilde{O}(p)$ and polynomial space complexity, by using Costello-Smith's method [11, §6]. As noted in Remark 3.1, CDS hash function returns an error when passing through any such vertex, hence their attack is not directly available.

With a quantum computer, we can solve the above two problems using Grover's algorithm [21] with time and space complexities $\widetilde{O}(p^{3/2})$. It is known that Tani's algorithm [37] find a solution to these problems in the time complexity $\widetilde{O}(p)$, but Jaques-Schanck [24] reported that Tani's algorithm requires much more quantum memory taking data structures into account. For the reason, the time and space complexities required for solving Problem 3.1 and Problem 3.2 are estimated in the original paper [7, §7.4] to be $\widetilde{O}(p^{3/2})$, even when using quantum algorithms.

4 Our collision attack

In this section, we will explain the algorithm of our collision attack on CDS hash function. In other words, our aim is finding two distinct paths of good extensions in Problem 3.2. We will first outline how to compute these paths in Section 4.1, followed by its specific algorithm in Algorithm 3.

4.1 Overview

The key to our attack lies in generating a path of good extensions

$$E_0 \times E_0 \xrightarrow{\phi_{-d}} \cdots \xrightarrow{\phi_{-1}} A_{-1} \xrightarrow{\phi_0} A_0 \longrightarrow \cdots \longrightarrow E_1 \times E_2, \tag{4.1}$$

such that each A_i is a simple abelian surface and E_1, E_2 are elliptic curves. This can be computed in the following procedure: we let e be the smallest (or slightly larger) positive integer satisfying

$$\frac{N_1 N_2}{\log N_1 N_2} > \frac{15 \cdot 2^{3d-1}}{\pi^2} p \quad \text{with } N_1 := 2^{e-1} + 1,\ N_2 := 2^{e-1} - 1. \tag{4.2}$$

We will explain why we take e in this way in Section 5.1. Using RepresentInteger algorithm, one can find an endomorphism α of E_0 whose degree equals to $N_1 N_2$. Since $N_1 + N_2 = 2^e$ and $\gcd(N_1, N_2) = 1$, the codomain of the $(2^e, 2^e)$-isogeny $\varPhi$ with the kernel $\{([N_1]P, \alpha(P)) \mid P \in E_0[2^e]\}$ is isomorphic to an elliptic product, as mentioned in Section 2.4. Repeating Algorithm 1 a sufficient number of times, one should obtain a path of good extensions from $E_0 \times E_0$ to $E_1 \times E_2$ such that its first $d+1$ steps are given (3.1), for some elliptic curves E_1 and E_2.

Remark 4.1. The probability that any A_i decomposes into the product of elliptic curves is negligible. Even if that were to happen, one could replace simply (4.1) with the chain up to A_i.

In this way, we obtain a desired chain (4.1) of length $= e$. Here, we write the vertices and edges after the initial isogeny $\phi_0 : A_{-1} \to A_0$ as follows:

$$A_{-1} \xrightarrow{\phi_0} A_0 \xrightarrow{\phi_1} \cdots \xrightarrow{\phi_{e-d-3}} A_{e-d-3} \xrightarrow{\phi_{e-d-2}} A_{e-d-2} \xrightarrow{\phi_{e-d-1}} E_1 \times E_2.$$

Generically, the two elliptic curves E_1, E_2 above are not isomorphic to each other and do not have a special automorphism (this means $E_1 \times E_2$ is a Type-Π vertex). Then, the left in Fig. 1 tells us that A_{e-d-2} is generically a Type-I vertex. Also, since $\hat{\phi}_{e-d-2} \circ \hat{\phi}_{e-d-1}$ is a $(4,4)$-isogeny, it follows from [18, §4.5] that A_{e-d-3} is generically a Type-A vertex. To sum up the discussions, the isogeny ϕ_{e-d-2} is an edge from a Type-A vertex to a Type-I vertex, with a very high probability.

Remark 4.2. It follows from Remark 2.3 that the probability that ϕ_{e-d-2} is not an isogeny from a Type-A vertex to a Type-I vertex is $O(1/p)$, which is negligible. Even if that were to happen, one could generate another chain (one may replace an integer e if necessary).

By applying Corollary 2.1 to the edge ϕ_{e-d-2}, we see that there exists a good extension $\phi'_{e-d-1} : A_{e-d-2} \to A_{e-d-3}$ of it. Then, we have a path

$$\begin{tikzcd}
A_{-1} \arrow[r, "\phi_0"] & A_0 \arrow[r, "\phi_1"] & \cdots \arrow[r, "\phi_{e-d-3}"] & A_{e-d-3} \arrow[r, "\phi_{e-d-2}"] & A_{e-d-2} \arrow[l, bend left, "\phi'_{e-d-1}"] \arrow[r, dotted] & E_1 \times E_2
\end{tikzcd}$$

of good extensions, where all A_i are simple abelian surfaces. Therefore, we finally obtain two distinct paths from A_0 to A_{e-d-2} of good extensions

$$\phi_{e-d-3} \circ \cdots \circ \phi_2 \circ \phi_1 \quad \text{and} \quad \phi'_{e-d-1} \circ \phi_{e-d-2} \circ \phi_{e-d-3} \circ \cdots \circ \phi_2 \circ \phi_1$$

in Problem 3.2. This causes a collision of CDS hash function, as desired.

4.2 How to generate the desired chain

In Section 4.1, we generated a $(2^e, 2^e)$-isogeny $\varPhi : E_0 \times E_0 \to E_1 \times E_2$ which can be decomposed as in (4.1) with $\phi_i = {\phi_{i-1}}^{(0)}$ for all $i \in \{-d+1, \dots, 0\}$. However, in practice, it is not necessary to compute itself thanks to the following lemma:

Lemma 4.1. *Let $\phi, \psi : A \to A'$ be multiple edges in $\mathcal{G}_2(2, p)$. If $\phi' : A' \to A''$ is a good extension of ϕ, then there exists a good extension $\psi' : A' \to A''$ of ψ.*

Proof. It is known [18, §3.1] that multiple edges ϕ, ψ are induced by a non-trivial automorphism ρ of the domain A such that $\ker \psi = \rho(\ker \phi)$. Then, there exists an automorphism $\rho' : A' \to A'$ which makes the following diagram commute:

$$\begin{tikzcd}
A \arrow[r, "\phi"] \arrow[d, "\rho"] & A' \arrow[d, "\rho'"] \\
A \arrow[r, "\psi"] & A'
\end{tikzcd}$$

For any good extension $\phi' : A' \to A''$ of ϕ, we define $\psi' := \phi' \circ \rho'^{-1} : A' \to A''$. Then, one can see that

$$\begin{aligned}
\ker \psi' \cap \psi(A[2]) &= \rho'(\ker \phi') \cap (\rho' \circ \phi \circ \rho^{-1})(A[2]) \\
&= \rho'(\ker \phi' \cap \phi(A[2])) = \rho'(\{0\}) = \{0\},
\end{aligned}$$

which means that ψ' is a good extension of ψ. □

Lemma 4.1 tells us that, for multiple edges $\phi_i, \psi_i : A_{i-1} \to A_i$, the codomains of their good extensions are common. In short even if $\phi_i = {\phi_{i-1}}^{(m_i)} : A_{i-1} \to A_i$ is obtained where $m_i \neq 0$, one can consider it as $\phi_i = {\phi_{i-1}}^{(0)}$ instead.

From the above discussion, our task is to find an endomorphism $\alpha \in \mathrm{End}(E_0)$ of degree $= N_1 N_2$ such that the $(2^{d+1}, 2^{d+1})$-isogeny with the kernel

$$\langle (N_1 P_0, \alpha(P_0)), (N_1 Q_0, \alpha(Q_0)) \rangle, \text{ where } (P_0, Q_0) \text{ is a basis of } E_0[2^{d+1}] \quad (4.3)$$

passes through $A_{-d}, \dots, A_{-1}, A_0$ in order. For the purpose, we sample endomorphisms $\alpha \leftarrow \mathsf{RepresentInteger}(N_1 N_2)$ of E_0 again and again, but it is not efficient to compute the induced isogeny by each α. Instead, we check whether α induces the desired $(2^{d+1}, 2^{d+1})$-isogeny using the following procedure:

- *Step 1:* Collect $(2^{d+1}, 2^{d+1})$-isogenies from $E_0 \times E_0$ which passes through all the vertices $A_{-d}, \dots, A_{-1}, A_0$ and denote the list of them by $\mathcal{L}$.
- *Step 2:* For each $\varphi \in \mathcal{L}$, find $c_{11}, c_{12}, c_{21}, c_{22} \in \mathbb{Z}/2^{d+1}\mathbb{Z}$ such that
$$\ker \varphi = \langle (N_1 P_0, c_{11} P_0 + c_{12} Q_0), (N_1 Q_0, c_{21} P_0 + c_{22} Q_0) \rangle$$
and define $\gamma_\varphi := (c_{ij})_{i,j}$.
- *Step 3:* Sample $\alpha \leftarrow \mathsf{RepresentInteger}(N_1 N_2)$, and let M_α be the matrix of α with respect to the basis (P_0, Q_0), that is,
$$M_\alpha \begin{pmatrix} P_0 \\ Q_0 \end{pmatrix} = \begin{pmatrix} \alpha(P_0) \\ \alpha(Q_0) \end{pmatrix}, \quad \text{where each entry in } M_\alpha \text{ belongs to } \mathbb{Z}/2^{d+1}\mathbb{Z}.$$
If M_α belongs to $\{\gamma_\varphi \mid \varphi \in \mathcal{L}\}$, output α; otherwise, go back to *Step 3*.

In the case where $M_\alpha = \gamma_\varphi$ holds, then this implies that

$$\langle (N_1 P_0,\, \alpha(P_0)),\, (N_1 Q_0,\, \alpha(Q_0)) \rangle = \ker \varphi,$$

and therefore α induces the desired $(2^{d+1}, 2^{d+1})$-isogeny φ. We denote the above algorithm by $\mathsf{FindEndomorphism}_d(N_1, N_2, P_0, Q_0)$.

4.3 How to compute the desired chain

Once the desired endomorphism α is obtained as in Section 4.2, we then need to compute the entire $(2^e, 2^e)$-isogeny $\varPhi$ from $E_0 \times E_0$ to an elliptic product which has the kernel $\{([N_1]P,\, \alpha(P)) \mid P \in E_0[2^e]\}$. The computational cost depends on the size of the field of definition of $E_0[2^e]$ and is dominant in our collision attack. In this subsection, we describe a method to reduce the time and space complexity for this (a detailed complexity analysis will be provided in Section 5.2).

First of all, let us restore the codomain of $\varPhi$ without computing $\varPhi$. Since we are given an endomorphism $\alpha \in \mathrm{End}(E_0) =: \mathcal{O}_0$ of degree $N_1 N_2$, there exist two elliptic curves E_1, E_2 which make the following diagram commute:

$$\begin{tikzcd}
E_0 \arrow[r, "\varphi_2"] \arrow[d, "\varphi_1"'] \arrow[dr, "\alpha"] & E_2 \arrow[d, "\psi_1"] \\
E_1 \arrow[r, "\psi_2"'] & E_0
\end{tikzcd}$$

with $\deg \varphi_i = \deg \psi_i = N_i$ for each $i \in \{1, 2\}$. It follows from [2, Lemma 6] that the left $\mathcal{O}_0$-ideal $I_i := \mathcal{O}_0 \alpha + N_i \mathcal{O}_0$ corresponds to φ_i, and then we can compute these two elliptic curves by $E_i \leftarrow \mathsf{IdealToIsogeny}(I_i)$. Moreover, by Theorem 2.2, we conclude that we have obtained $E_1 \times E_2$, which is the codomain of $\varPhi$.

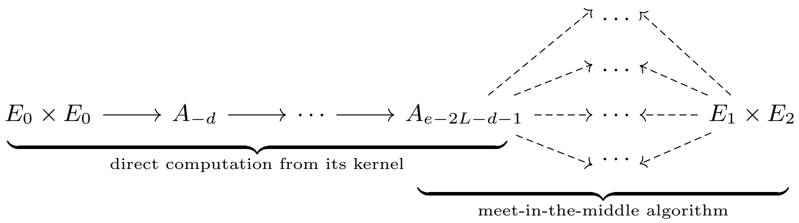
Then, let $L \geq 0$ be the smallest integer such that

$$\text{there is a model of } E_0 \text{ whose } 2^{e-2L}\text{-torsion subgroup is } \mathbb{F}_{p^r}\text{-rational} \qquad (4.4)$$

with $r := 2^{3L+1}$, and we also denote by E_0 this model although it is an abuse of notation. We compute the induced chain by α using the following procedure:

- *Former step:* We compute the $(2^{e-2L}, 2^{e-2L})$-isogeny $E_0 \times E_0 \to A_{e-2L-d-1}$ with the kernel $\{([N_1]P,\, \alpha(P)) \mid P \in E_0[2^{e-2L}]\}$.
- *Latter step:* We find a path of good extensions from $A_{e-2L-d-1}$ to $E_1 \times E_2$, using the meet-in-the-middle algorithm. In this step, we do not use information of the kernel, thus we can compute it over $\mathbb{F}_{p^2}$.

Their composition gives the desired chain.



Algorithm 3 An collision attack for CDS hash function

```
Input: A positive integer $d$
Output: Messages $m \neq m' \in \{0,1\}^*$ such that $\mathsf{CDS}_d(m) = \mathsf{CDS}_d(m')$
 1: Let $e > 0$ be the smallest integer satisfying (4.2) and $N_1 \leftarrow 2^{e-1}+1$, $N_2 \leftarrow 2^{e-1}-1$
 2: Let $L \geq 0$ be the integer as in (4.4)
 3: Let $(P, Q)$ be a basis of $E_0[2^{e-2L}]$ and $P_0 \leftarrow 2^{e-2L-d-1}P$, $Q_0 \leftarrow 2^{e-2L-d-1}Q$
 4: Let $\alpha \leftarrow \mathsf{FindEndomorphism}_d(N_1, N_2, P_0, Q_0)$
 5: Let $I_1, I_2$ be the left $\mathcal{O}_0$-ideals such that $I_1 = \mathcal{O}_0\alpha + N_1\mathcal{O}_0$ and $I_2 = \mathcal{O}_0\alpha + N_2\mathcal{O}_0$
 6: Let $E_1 \leftarrow \mathsf{IdealToIsogeny}(I_1)$, $E_2 \leftarrow \mathsf{IdealToIsogeny}(I_2)$
 7: Let $D_P \leftarrow ([N_1]P,\ \alpha(P))$, $D_Q \leftarrow ([N_1]Q,\ \alpha(Q)) \in E_0 \times E_0$
 8: for $i \in \{1, \dots, e-2L\}$ do
 9:     Compute $\phi_{i-d-1}$ be the $(2,2)$-isogeny with the kernel $\langle 2^{e-2L-i}D_P, 2^{e-2L-i}D_Q \rangle$,
        and let $A_{i-d-1}$ be the codomain of $\phi_{i-d-1}$
10:     Let $D_P \leftarrow \phi_{i-d-1}(D_P)$, $D_Q \leftarrow \phi_{i-d-1}(D_Q) \in A_{i-d-1}$
11: end for
12: Let $\phi_n \circ \phi_{n-1} \circ \cdots \circ \phi_{e-2L-d} \leftarrow \mathsf{MeetInTheMiddle}(\phi_{e-2L-d-1}, E_1 \times E_2, L)$
13: for $i = 1, \dots, n-1$ do
14:     Choose $m_i \in \{0, \dots, 7\}$ such that the codomain of ${\phi_{i-1}}^{(m_i)}$ is isomorphic to $A_i$
15: end for
16: Choose $m_n \in \{0, \dots, 7\}$ such that the codomain of ${\phi_{n-1}}^{(m_n)}$ is isomorphic to $A_{n-2}$
17: Let $m \leftarrow m_{n-2} \cdots m_2 m_1$ and $m' \leftarrow m_n m_{n-1} m_{n-2} \cdots m_2 m_1$
18: return the binary representations of $m$ and $m'$
```

Summarizing the above discussions, a pseudocode of our collision attack can be described in Algorithm 3. For an edge $\phi : A \to B$ and a vertex B' in $\mathcal{G}_2(2, p)$, we denote by $\mathsf{MeetInTheMiddle}(\phi, B', n)$ the meet-in-the-middle algorithm which returns a path $B \to B'$ of good extensions of ϕ of length $\leq 2n$ as follows:

- *Step 1:* We compute all the paths of good extensions of ϕ of length $\leq n$ and store them together with their codomains in the table T.
- *Step 2:* We compute $(2^\bullet, 2^\bullet)$-isogenies starting from B' of length $\leq n$, then check whether their codomains exist in the table T or not. Repeat this procedure until two paths $\psi' : B \to B''$, $\psi : B' \to B''$ with isomorphic codomains are obtained.
- *Step 3:* For ψ, ψ' obtained in *Step 2*, return $\hat{\psi}' \circ \psi : B \to B'' \to B'$ if it is a $(2^\bullet, 2^\bullet)$-isogeny. Otherwise, go back to *Step 2*.

Since there is a $(2^{2L}, 2^{2L})$-isogeny $A_{e-2L-d-1} \to E_1 \times E_2$ which is a good extension of $\phi_{e-2L-d-1}$ by the choice of α, line 12 of Algorithm 3 terminates.

Remark 4.3. To determine whether two abelian surfaces are isomorphic to each other, it suffices to verify whether their absolute Igusa invariants of the underlying genus-2 curves are the same.

The computational cost of Algorithm 3 heavily depends on the computation of a $(2^e, 2^e)$-isogeny $E_0 \times E_0 \to E_1 \times E_2$ in lines 8–12. We have defined $L \geq 0$ as the smallest integer satisfying (4.4) in order to balance the computational costs of the former step (lines 8–11) and the latter step (line 12). See Section 5.2 for a detailed complexity analysis of Algorithm 3.

5 Complexity analysis

In this section, we discuss the termination of our collision attack (Algorithm 3), and then we estimate its computational complexity. Also, a simple countermeasure for this attack will also be proposed in Section 5.3.

5.1 Termination

In Algorithm 3, the termination is non-trivial only in line 4, and in this section, we will illustrate the reason why we believe that $\mathsf{FindEndomorphism}_d$ terminates in a finite number of iterations. First, we make the following heuristic assumption about the $\mathsf{RepresentInteger}$ algorithm (Algorithm 1).

Assumption 5.1. For an integer $N > p$, the $\mathsf{RepresentInteger}(N)$ outputs at least

$$\frac{\pi^2}{3}\frac{N}{p\log N}$$

distinct endomorphisms $\alpha \in \mathbb{Z} + \mathbb{Z}\mathbf{i} + \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$ whose degree $= N$.

We explain the reasoning behind why we assume that Assumption 5.1 holds. To sample $\alpha \leftarrow \mathsf{RepresentInteger}(N)$, recall from the first half of Section 2.2 that we find integers $x, y, z, w \geq 0$ satisfying

$$x^2 + 3y^2 + p(z^2 + 3w^2) = N. \tag{5.1}$$

The equation (5.1) implies that $p(z^2 + 3w^2) \leq N$, and one can estimate that the number of (z, w) satisfying this inequality is approximately

$$\frac{\pi}{\sqrt{3}}\frac{N}{p},$$

by considering the area of the region $p(z^2+3w^2) \leq N$. For each pair (z, w), since the number of (x, y) satisfying (5.1) is on average $\pi/\sqrt{3}$, one can estimate that the number of (x, y, z, w) satisfying (5.1) is approximately

$$\frac{\pi^2}{3}\frac{N}{p},$$

which is an estimate of the number of all endomorphisms $\alpha \in \mathbb{Z} + \mathbb{Z}\mathbf{i} + \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$ whose degree $= N$. However, as stated in Remark 2.1, some endomorphisms are not output by $\mathsf{RepresentInteger}(N)$ due to the failure of the integer factorization. Since we can assume that at least $1/\log N$ of them could be the outputs, we can generate at least

$$\frac{\pi^2}{3}\frac{N}{p\log N}$$

distinct endomorphisms $\alpha \in \mathbb{Z} + \mathbb{Z}\mathbf{i} + \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$ whose norm $= N$. This is why we expect that Assuption 5.1 holds.

Remark 5.1. If we take $z = w = 0$ in (5.1), we obtain an endomorphism α of E_0 corresponding to a quaternion $\in \mathbb{Z}[\sqrt{-3}]$. It can be shown (see Proposition A.1) that paths induced by such α via Kani's lemma do not pass through any simple abelian surface. Therefore, we can exclude the case where $z = w = 0$, in fact.

Next, let e be the smallest (or slightly larger) positive integer satisfying (4.2) and define $N_1 \coloneqq 2^{e-1}+1,\ N_2 \coloneqq 2^{e-1}-1$. Sample $\alpha \leftarrow \mathsf{RepresentInteger}(N_1N_2)$ and consider the $(2^{d+1}, 2^{d+1})$-isogeny φ_α such that

$$\ker \varphi_\alpha = \langle (N_1P_0, \alpha(P_0)), (N_1Q_0, \alpha(Q_0)) \rangle \text{ where } E_0[2^{d+1}] = \langle P_0, Q_0 \rangle.$$

In this situation, we make the following heuristic assumption:

Assumption 5.2. The φ_α defined as above is uniformly distributed among all of the $(2^{d+1}, 2^{d+1})$-isogenies starting from $E_0 \times E_0$.

Under the two heuristic assumptions above, we show that $\mathsf{FindEndomorphism}_d$ terminates in a finite number of iterations:

Lemma 5.1. *For two integers N_1, N_2 as above and a basis (P_0, Q_0) of $E_0[2^{d+1}]$, the $\mathsf{FindEndomorphism}(N_1, N_2, P_0, Q_0)$ outputs a solution in expected*

$$\approx (5 \cdot 2^{3d-1}) \log N_1N_2$$

iterations, under Assumptions 5.1 and 5.2.

Proof. For a sampled endomorphism $\alpha \leftarrow \mathsf{RepresentInteger}(N_1N_2)$, we estimate the probability that φ_α passes through all $A_{-d}, \dots, A_{-1}, A_0$ in order:

- *A first step*: there exist 15 edges ϕ_{-d} from $E_0 \times E_0$ in the graph $\mathcal{G}_2(2,p)$, of which 3 have A_{-d} as their codomains (the remaining 12 have the product of two elliptic curves as their codomain, see [18, §4.13] for details). Hence, the probability that ϕ_{-d} passes through A_{-d} equals to 1/5.
- *A second step*: there exist 8 good extensions ϕ_{-d+1} of ϕ_{-d} among which the edges to a $\mathsf{Type}\text{-}I$ vertex have multiplicity 6, and the other ($\mathsf{Type}\text{-}IV$) vertex have multiplicity = 2 from [18, §4.13]. Which of the two is the target A_{-d+1} depends on how the pre-computed edge $\phi_{-d,0}$ was chosen, but in any case, the probability that ϕ_{-d+1} passes through A_{-d+1} is at least 1/4.
- *After a third step*: for $i \in \{-d+2, \dots, 0\}$, there exist 8 good extensions ϕ_i of $\phi_{i-1} : A_{i-2} \to A_{i-1}$, and hence the probability that ϕ_i passes through A_i is at least 1/8.

Therefore, the probability that a random $(2^{d+1}, 2^{d+1})$-isogeny φ_α passes through all $A_{-d}, \dots, A_{-1}, A_0$ is estimated to be $\geq 1/5 \cdot 1/4 \cdot (1/8)^{d-1} = 1/(5 \cdot 2^{3d-1})$.

Now, recall from Assumption 5.1 and (4.2) that $\mathsf{RepresentInteger}(N_1N_2)$ can generate at least

$$\frac{\pi^2}{3} \frac{N_1N_2}{p \log N_1N_2} > 5 \times 2^{3d-1}$$

random endomorphisms, thus we can expect that one of them will be output by $\mathsf{FindEndomorphism}_d(N_1, N_2, P_0, Q_0)$. Taking into account the failure of factorization, the number of iterations required is $\approx (5 \cdot 2^{3d-1}) \log N_1N_2$, as desired. □

Remark 5.2. Although there are several methods to generate the desired α other than the one described in Section 4.2, they are expected to induce much longer isogenies, see Appendix B for details. As will be discussed in Section 5.2 below, this worsens the computational complexity of our collision attack.

5.2 Efficiency

In the following, we estimate the computational complexity of Algorithm 3 and give the proof of our main theorem (Theorem 1.1). Here, the soft O-notation $\widetilde{O}$ means ignoring logarithmic factors of p.

Theorem 5.1. *Let $d > 0$ be an integer in $O(\log p)$. Then, Algorithm 3 can solve Problem 3.2 for CDS_d with*

- *the time complexity $\widetilde{O}(2^{3d})$ and $\widetilde{O}(2^{9d/10}p^{3/10})$,*
- *the space complexity $\widetilde{O}(2^{9d/10}p^{3/10})$*

under either the heuristics in [20, Lemma 3] or the GRH together with Assumptions 5.1 and 5.2.

Proof. For e, N_1, N_2 chosen in line 1 of Algorithm 3, we have

$$2^{2e-2} - 1 = N_1N_2 \approx \frac{15 \cdot 2^{3d-1}}{\pi^2} p \log\left(\frac{15 \cdot 2^{3d-1}}{\pi^2} p\right),$$

which means $2^{2e} = \widetilde{O}(2^{3d}p)$. Now, let us define $f : \mathbb{N} \to \mathbb{N}$ as

$$n \mapsto (\text{the smallest integer } k > 0 \text{ such that there is a model of } E_0 \\ \text{whose } 2^n\text{-torsion subgroup is } \mathbb{F}_{p^k}\text{-rational}).$$

It is known that there is a model of E_0 whose 4-torsion subgroup is $\mathbb{F}_{p^2}$-rational, then we have that $f(n) \leq 2^{n-1}$ for all $n \in \mathbb{N}$. This fact tells us that L defined in line 2 satisfies the inequality $3L + 1 \leq e - 2L$, which implies $L \leq e/5$. Based on these parameters, we evaluate the computational complexity of each part of Algorithm 3 in the following:

- In line 4, we get an endomorphism $\alpha \leftarrow \mathsf{FindEndomorphism}_d(N_1, N_2, P_0, Q_0)$ in approximately
$$(5 \cdot 2^{3d-1}) \log N_1N_2 = \widetilde{O}(2^{3d})$$
iterations (requires only integer arithmetics), as shown in Lemma 5.1.
- In line 6, we can recover elliptic curves E_1, E_2 in expected polynomial time in $\log p$, thanks to Lemma 2.1 together with that $\log N_1, \log N_2 = O(\log p)$.
- In lines 8–11, it requires exactly $e - 2L = O(\log p)$ times of the $(2,2)$-isogeny computations over $\mathbb{F}_{p^r}$ with $r := 2^{3L+1}$. This can be done in complexity
$$\widetilde{O}(2^{3L+1}) = \widetilde{O}(2^{3e/5}) = \widetilde{O}(2^{9d/10}p^{3/10})$$
from the fact (cf. [12, §9.6]) that operations on $\mathbb{F}_{p^r}$ can be done in $\widetilde{O}(r)$.
- In line 12, recalling that the number of good extensions of a $(2,2)$-isogeny is equal to $2^3 = 8$, we need to compute and store a total of $O(8^L)$ edges. This can be done in time and space complexities
$$O(8^L) = O(8^{e/5}) = \widetilde{O}(2^{9d/10}p^{3/10})$$
since all the $(2,2)$-isogeny computations are performed in $\mathbb{F}_{p^2}$.
- In lines 13–16, at most $8e = O(\log p)$ times of the $(2,2)$-isogeny computations over $\mathbb{F}_{p^2}$ are required, which can be done in polynomial time in $\log p$.

From the above discussions, we obtain the statement of this theorem. □

Finally, we provide the proof of Theorem 1.1.

Proof (Proof of Theorem 1.1). Recalling that we set $d \coloneqq 10$ in original settings of CDS hash functions, then it follows immediately from Theorem 5.1 that both the time and space complexity of Algorithm 3 are $\widetilde{O}(p^{3/10})$. This completes the proof of Theorem 1.1. □

In addition, we provide a condition under which our collision attack on CDS hash function is a polynomial-time and space algorithm.

Corollary 5.1. *If the 2^e-torsion subgroup of E_0 is $\mathbb{F}_{p^k}$-rational with $k = O(1)$, then collisions on CDS hash function can be found in polynomial-time and space with respect to $\log p$.*

Proof. In Algorithm 3 with $d = 10$, all lines except for lines 8–11 and line 12 run in polynomial-time and space. Moreover, we can take $L = O(1)$ by the assumption, and thus all the $(2, 2)$-isogeny computations in lines 8–12 can be performed in polynomial time and space with respect to $\log p$. □

Remark 5.3. From the same perspective as Castryck-Decru-Smith [7, §7.4], it is considered that the quantum computational complexity of our collision attack is not different from its classical computational complexity. This is based on the belief by Castryck-Decru-Smith that (currently) the best claw-finding algorithm for a graph of size $= N$ has a quantum computational complexity $\widetilde{O}(N^{1/2})$. We do not have any idea to optimize the meet-in-the-middle part by using quantum algorithms, and thus we will not explore this further.

5.3 Countermeasures

The main reason our collision attack for CDS hash function is applicable is that there were too few zeros appended to the messages $m \in \{0, 1\}^*$. In other words, by setting the initial isogeny $\phi_0 : A_{-1} \to A_0$ further "away" from the product of elliptic curves, one can increase the computational complexity of our attack. In particular, if ϕ_0 is put sufficiently far away, then our polynomial-time algorithm will no longer be applicable.

However, it seems inevitable that the computational complexity required for solving Problem 3.2 reduce to $\widetilde{O}(p)$ no matter how $\phi_0 : A_{-1} \to A_0$ is set. Indeed, we should adopt the following strategy like Costello-Smith's method:

- *Random walk:* Perform a random walk on the graph $\mathcal{G}_2(2, p)$ such that ϕ_i is a good extension of ϕ_{i-1} for all $i \geq 1$ until the codomain of ϕ_i is isomorphic to the product of elliptic curves.
- *Collision finding:* For the path $A_{-1} \to E_1 \times E_2$ obtained as described above, collisions can be found using a method similar to that in Section 4.1.

As stated in Remark 3.2, such a path $A_{-1} \to E_1 \times E_2$ is obtained with the time complexity $\widetilde{O}(p)$ and polynomial memory**.

** Using a quantum computer, such a path can be found with time complexity $\widetilde{O}(p^{1/2})$ and a polynomial memory, according to [11, Theorem 2].

From these reason, we propose to put the initial surface A_0 from the product of elliptic curves by at least $d > (1/3)\log_2 p$. If we set $d \approx (1/3)\log_2 p$, then our collision attack requires time complexity $\widetilde{O}(p)$ and space complexity $\widetilde{O}(p^{3/5})$ by Theorem 5.1. Therefore, such a modification does not allow us to claim that our collision attack is more efficient than the Costello-Smith's method.

If one wants to avoid such an attack, it might be valuable to consider, as suggested by [28], the $(2,2)$-isogeny graph of supersingular *non-superspecial* abelian surfaces over $\mathbb{F}_{p^4}$ instead of $\mathcal{G}_2(2,p)$. Since their alternative graph does not have vertices corresponding to the products of elliptic curves, Costello-Smith's method and our collision attack would no longer applicable by making this modification.

6 Computational results

In this section, we will give some computational results found by executing our algorithm (Algorithm 3). We implemented this algorithm with Magma V2.28-4, and ran it on a machine with an AMD EPYC 7742 CPU and 2TB of RAM.

In the following experiments, we chose two parameters in which our collision attack works in polynomial time (to be more precise, we have chosen p such that all the $(2,2)$-isogenies appeared through the computations are defined over $\mathbb{F}_{p^2}$). Under these parameters, the dominant step is the $\mathsf{FindEndomorphism}_{10}$ part, and performed 64 parallel computations only for this part by changing the seed value. Our implementation is available at

`https://github.com/Ryo-Ohashi/CDScollision`.

Result for a 128-bit prime. We choose a 128-bit prime $p = 2^{86} \cdot 3 \cdot f - 1$ with a cofactor $f = 6397 \cdot 229172347$ as the characteristic of the base field (the security of CDS hash function was originally considered to be 192-bit, in this parameter). We set $e := 86$ and found a collision

$m_1 =$ `5720C642EC34DB2C62639590384D92E3124A6F032AEE11650D4BFE0`,
$m_2 =$ `AD720C642EC34DB2C62639590384D92E3124A6F032AEE11650D4BFE0`.

It took 10861.3 seconds in total to find the collision as a single parallel instance, of which 10819.6 seconds were spent on the $\mathsf{FindEndomorphism}_{10}$ part, requiring the 4253856 iterations of trying random endomorphisms.

Result for a 256-bit prime. We choose a 256-bit prime $p = 2^{150} \cdot 3 \cdot f - 1$ with a cofactor $f = 5 \cdot 19 \cdot 199 \cdot 45153169 \cdot 639964709 \cdot 49503663809$ as the characteristic of the base field (the security of CDS hash function was originally considered to be 394-bit, in this parameter). We set $e := 150$ and found a collision

$m_1 =$ `3ABE8C61C4CAAD5E4C127D76278AB541C8A27F627793A1BF7DD9452E`
`C73CCC987DBBA0EA4AB98521C459F31B6FBFFF42C89C11C`,

$m_2 =$ `143ABE8C61C4CAAD5E4C127D76278AB541C8A27F627793A1BF7DD9452E`
`C73CCC987DBBA0EA4AB98521C459F31B6FBFFF42C89C11C`.

It took 60239.1 seconds in total to find the collision as a single parallel instance, of which 60198.4 seconds were spent on the $\mathsf{FindEndomorphism}_{10}$ part, requiring the 9315093 iterations of trying random endomorphisms.

7 Concluding remarks

In this paper, we have proposed an explicit algorithm (Algorithm 3) for finding a collision on CDS hash function. Our collision attack utilizes the property that the initial surface A_0 is too "close" to the square of a supersingular elliptic curve with a known endomorphism ring. Indeed, if we put the initial surface A_0 sufficiently far away, then it is considered that CDS hash function remains secure.

This situation is significantly different from CGL hash function: if we know a path from a vertex with known endomorphism ring to the initial curve E_{start}, then the hash function is no longer safe. Similarly, we suspect that it is not safe to set A_0 as the initial surface when the path $E_0 \times E_0 \to A_0$ is known, where E_0 is a supersingular elliptic curve with known endomorphism ring. Further analysis of the security of CDS hash functions is a future work.

Acknowledgements. The authors thank Prof. Katsuyuki Takashima for his helpful comments at the JSIAM Annual Meeting 2024. This research was conducted under a contract of "Research and development on new generation cryptography for secure wireless communication services" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan.

References

1. A. Basso, G. Codogni, D. Connolly, L. D. Feo, T. B. Fouotsa, G. M. Lido, T. Morrison, L. Panny, S. Patranabis, B. Wesolowski: *Supersingular curves you can trust*, Eurocrypt 2023, LNCS **14005**, 405–437.
2. A. Basso, L. D. Feo, P. Dartois, A. Leroux, L. Maino, G. Pope, D. Robert, B. Wesolowski: *SQIsign2D-West: The fast, the small, and the safer*, to appear in Asiacrypt 2024.
3. A. Basso, L. Maino, G. Pope: *FESTA: Fast encryption from supersingular torsion attacks*, Asiacrypt 2023, LNCS **14438**, 98–126.
4. O. Bolza: *On binary sextics with linear transformations into themselves*, American J. **10**(1), 47–70, 1887.
5. W. Bosma, J. Cannon, C. Playoust: *The Magma algebra system I: The user language*, J. Symb. Comput. **24**, 235–265, 1997.
6. W. Castryck and T. Decru: *Multiradical isogenies*, Contemp. Math. **779**, 57–89, 2022.
7. W. Castryck, T. Decru, B. Smith: *Hash functions from superspecial genus-2 curves using Richelot isogenies*, J. Math. Cryptol. **14**, 268–292, 2020.
8. D. Charles, K. E. Lauter, E. Z. Goren: *Cryptographic hash functions from expander graphs*, J. Cryptology **22**(1), 93–113, 2009.
9. M. Chen, A. Leroux, L. Panny: *SCALLOP-HD: Group action from 2-dimensional isogenies*, PKC 2024, LNCS **14603**, 190–216.
10. H. Cohen: *A Course in Computational Algebraic Number Theory*, GTM **138**, 1993.
11. C. Costello and B. Smith: *The supersingular isogeny problem in genus 2 and beyond*, PQCrypto 2020, LNCS **12100**, 151–168.
12. R. Crandall and C. Pomerance: *Prime Numbers: A Computational Perpective*, Springer, Second Edition, 2005.

13. M. Deuring: *Die typen der multiplikatorenringe elliptischer funktionenkörper*, Abh. Math. Semin. Univ. Hamb. **14**, 197–272, 1941.
14. M. Duparc and T. B. Fouotsa: *SQIPrime: A dimension 2 variant of SQIsignHD with non-smooth challenge isogenies*, to appear in Asiacrypt 2024.
15. K. Eisenträger, S. Hallgren, K. Lauter, T. Morrison, C. Petit: *Supersingular isogeny graphs and endomorphism rings: reductions and solutions*, Eurocrypt 2018, LNCS **10822**, 329–368.
16. K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison, J. Park: *Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs*, ANTS-XIV **4**, 215–232, 2022.
17. T. Ekedahl: *On supersingular curves and abelian varieties*, Math. Scand. **60**, 151–178, 1987.
18. E. Florit and B. Smith: *An atlas of the Richelot isogeny graph*, RIMS Kôkyûroku Bessatsu **90**, 195–219, 2022.
19. E.V. Flynn and Y. B. Ti: *Genus two isogeny cryptography*, PQCrypto 2019, LNCS **11505**, 286–306.
20. S. D. Galbraith, C. Petit, J. Silva: *Identification protocols and signature schemes based on supersingular isogeny problems*, Asiacrypt 2017, LNCS **10624**, 3–33.
21. L. K. Grover: *A fast quantum mechanical algorithm for database search*, In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 212–219, 1996.
22. T. Ibukiyama, T. Katsura, F. Oort: *Supersingular curves of genus two and class numbers*, Compos. Math. **57**, 127–152, 1986.
23. J. Igusa: *Arithmetic variety of moduli for genus two*, Ann. Math. **72**(2), 612–649, 1960.
24. S. Jaques and J. M. Schanck: *Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE*, LNCS **11692**, 32–61, 2019.
25. B. W. Jordan and Y. Zaytman: *Isogeny graphs of superspecial abelian varieties and Brandt matrices*, preprint, arXiv: 2005.09031.
26. E. Kani: *The number of curves of genus two with elliptic differentials*, J. Reine Angew. Math. **485**, 93–121, 1997.
27. D. Kohel, K. Lauter, C. Petit, J. -P. Tignol: *On the quaternion ℓ-isogeny path problem*, LMS J. Comput. Math. **17A**, 418–432, 2014.
28. J. LeGrow, Y. B. Ti, L. Zobernig: *Supersingular non-superspecial abelian surfaces in cryptography*, Mathematical Cryptology **3**(2), 11–23, 2023.
29. L. Maino, C. Martindale, L. Panny, G. Pope, B. Wesolowski: *A direct key recovery attack on SIDH*, Eurocrypt 2023, LNCS **14008**, 448–471.
30. M. Mamah: *The supersingular isogeny path and endomorphism ring problems: unconditional reductions*, preprint, `https://ia.cr/2024/1569`.
31. J. -F. Mestre: *La méthode des graphes. Exemples et applications*, In: Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata 1986), 217–242, 1986.
32. J. S. Milne: *Jacobian varieties*, In: Arithmetic geometry (Storrs/Conn. 1984), 167–212, 1986.
33. K. Nakagawa and H. Onuki: *QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras*, Crypto 2024, LNCS **14924**, 75–106.
34. K. Nakagawa and H. Onuki: *SQIsign2D-East: A new signature scheme using 2-dimensional isogenies*, to appear in Asiacrypt 2024.
35. J. H. Silverman: *The Arithmetic of Elliptic Curve*, GTM **106**, 1986.
36. K. Takashima: *Efficient algorithms for isogeny sequences and their cryptographic applications*, Mathematics for Industry **29**, 97–114, 2018.

37. S. Tani: *Claw finding algorithm using quantum walk*, Theor. Comput. Sci. **410**(50), 5285–5297, 2009.
38. B. Wesolowski: *The supersingular isogeny path and endomorphism ring problems are equivalent*, In: FOCS 2021–62nd Annual IEEE Symposium on Foundations of Computer Science, 1100–1111, 2022.

A In the case where $\boldsymbol{\alpha}$ belongs to $\mathbb{Z}[\boldsymbol{\omega}]$

Let $p \equiv 5 \pmod 6$ be a prime and consider the supersingular elliptic curve

$$E_0 : v^2 = u^3 + 1, \quad \text{with } j\text{-invariant} = 0$$

defined over $\mathbb{F}_{p^2}$. The curve E_0 has an automorphism $(u, v) \mapsto (\omega u, v)$ of order 3 where ω is a primitive cube root of unity, and we denote by ω the automorphism even though it may be considered an abuse of notation.

Lemma A.1. *If $\alpha \in \mathbb{Z}[\omega]$, then the codomain of the $(2,2)$-isogeny whose kernel is $G_0 \coloneqq \{(P_0, \alpha(P_0)) \mid P_0 \in E_0[2]\}$ is isomorphic to $E_0 \times E_0$.*

Proof. We can write $\alpha = x + y\omega \in \mathbb{Z}[\omega]$ with $x, y \in \mathbb{Z}$ since $\omega^2 = -(1+\omega)$.

- If x and y are even, then $\alpha(P_0) = O$ for all $P_0 \in E_0[2]$. In this case G_0 is not maximal 2-isotropic, which contradicts our assumption.
- If x is odd and y is even, then $\alpha(P_0) = P_0$ for all $P_0 \in E_0[2]$. In this case
$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} : E_0 \times E_0 \to E_0 \times E_0$$
gives the $(2,2)$-isogeny with the kernel G_0.
- If x is even and y is odd, then $\alpha(P_0) = \omega(P_0)$ for all $P_0 \in E_0[2]$. In this case
$$\begin{pmatrix} 1 & -\omega \\ \omega^2 & 1 \end{pmatrix} : E_0 \times E_0 \to E_0 \times E_0$$
gives the $(2,2)$-isogeny with the kernel G_0.
- If x and y are odd, then $\alpha(P_0) = P_0 + \omega(P_0)$ for all $P \in E_0[2]$. In this case
$$\begin{pmatrix} 1 & -\omega^2 \\ \omega & 1 \end{pmatrix} : E_0 \times E_0 \to E_0 \times E_0$$
gives the $(2,2)$-isogeny with the kernel G_0.

The codomains for the last three cases are isomorphic to $E_0 \times E_0$, as desired. □

Proposition A.1. *If $\alpha \in \mathbb{Z}[\omega]$, then the codomain of the $(2^e, 2^e)$-isogeny whose kernel is $G \coloneqq \{(P, \alpha(P)) \mid P \in E_0[2^e]\}$ is isomorphic to $E_0 \times E_0$ for all $e \geq 1$.*

Proof. We give a proof by induction on e. The case where $e = 1$ is no other than Lemma A.1, and hence true. Next, we suppose that the assertion holds for $e - 1$ and let Φ be the $(2^e, 2^e)$-isogeny with $\ker \Phi = G$. Decomposing this as $\Phi = \psi \circ \phi$ where ϕ is a $(2^{e-1}, 2^{e-1})$-isogeny and ψ is a $(2,2)$-isogeny, we can write

$$\phi = \begin{pmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{pmatrix} : E_0 \times E_0 \to E_0 \times E_0$$

with $\beta_{11}, \beta_{12}, \beta_{21}, \beta_{22} \in \mathbb{Z}[\omega]$ by the induction hypothesis. Moreover, we have

$$\begin{aligned} \ker \psi &= \{(\beta_{11}P + \beta_{12}\alpha(P), \beta_{21}P + \beta_{21}\alpha(P)) \mid P \in E_0[2^e]\} \\ &= \{(\beta_1(P_0), \beta_2(P_0)) \mid P_0 \in E_0[2]\}, \quad \text{with } \beta_1, \beta_2 \in \{1, \omega, \omega^2\}. \end{aligned}$$

This can be rewritten as $\ker \psi = \{(P_0, \beta(P_0)) \mid P_0 \in E_0[2]\}$ with $\beta \in \mathbb{Z}[\omega]$, since any $\beta_1 \in \{1, \omega, \omega^2\}$ is a unit in $\mathbb{Z}[\omega]$. Applying Lemma A.1 again, the codomain of ψ is isomorphic to $E_0 \times E_0$, as desired. □

B Alternative methods for computing α

In Algorithm 3, we first fix a degree of an endomorphism α of E_0 and then search for α randomly until its restriction on $E_0[2^{d+1}]$ satisfies the "desired" conditions. In this appendix, we consider alternative methods for computing α which take a converse approach to the original method. In other words, we firstly compute an endomorphism α_0 of E_0 such that its restriction on $E_0[2^{d+1}]$ satisfies the desired conditions. Then, we find an endomorphism α whose degree satisfies the desired conditions and $\alpha \equiv m\alpha_0 \pmod{2^{d+1}}$ for some integer m (we will discuss how to choose m later). Unfortunately, we cannot find an alternative method finding α whose degree is as small as in the original method. However, some techniques on quaternion could be applied to reduce the degree of α, and therefore we present alternative methods for future research.

We use the same notation as in Algorithm 3. Our task is to find an endomorphism α of E_0 and positive integers e, N_1, N_2 satisfying

- the $(2^{d+1}, 2^{d+1})$-isogeny with kernel (4.3) passes through $A_{-d}, \dots, A_{-1}, A_0$,
- the degree of α equals to $N_1 N_2$ with $N_1 + N_2 = 2^e$ and $\gcd(N_1, N_2) = 1$.

First, let us compute an endomorphism $\alpha_0 \in \mathrm{End}(E_0)$ such that the $(2^{d+1}, 2^{d+1})$-isogeny with kernel $\langle (P_0, \alpha_0(P_0)), (Q_0, \alpha_0(Q_0)) \rangle$ passes through $A_{-d}, \dots, A_{-1}, A_0$ in order. This can be done as follows:

- *Step 1:* Let $(\beta_1, \beta_2, \beta_3, \beta_4)$ be a Minkowski reduced basis of $\mathcal{O}_0 \coloneqq \mathrm{End}(E_0)$, and $M_1, M_2, M_3, M_4 \in \mathbb{Z}/2^{d+1}\mathbb{Z}$ be the matrices of $\beta_1, \beta_2, \beta_3, \beta_4$ with respect to the basis (P_0, Q_0).
- *Step 2:* Find $c_{11}, c_{12}, c_{21}, c_{22} \in \mathbb{Z}$ such that the $(2^{d+1}, 2^{d+1})$-isogeny φ with
$$\ker \varphi = \langle (P_0, c_{11}P_0 + c_{12}Q_0), (Q_0, c_{21}P_0 + c_{22}Q_0) \rangle$$
passes through all the vertices $A_{-d}, \dots, A_{-1}, A_0$ in order.
- *Step 3:* Solve the linear equation
$$x_1 M_1 + x_2 M_2 + x_3 M_3 + x_4 M_4 \equiv \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \pmod{2^{d+1}}$$
and define $\alpha_0 \coloneqq x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + x_4\beta_4$.

Since x_1, x_2, x_3, x_4 are approximately 2^{d+1}, we have $\deg \alpha_0 \approx 2^{2d+2}p$.

The remaining task is to find positive integers e, N_1, N_2 such that

$$\begin{cases} \deg \alpha = N_1 N_2, \\ N_1 + N_2 = 2^e, \\ \gcd(N_1, N_2) = 1, \\ \alpha \equiv N_1 \alpha_0 \pmod{2^{d+1}}. \end{cases}$$

Then, we have that

$$\langle (N_1 P_0, \alpha(P_0)), (N_1 Q_0, \alpha(Q_0)) \rangle = \langle (P_0, \alpha_0(P_0)), (Q_0, \alpha_0(Q_0)) \rangle,$$

which implies that α is the desired endomorphism.

We can mitigated the congruence condition to $\alpha \equiv m\alpha_0 \pmod{2^{d+1}}$ for some integer m. The reason is as follows: assume α satisfies the condition with $m \in \mathbb{Z}$. Since $\langle (P_0, \alpha_0(P_0)), (Q_0, \alpha_0(Q_0)) \rangle$ is isotropic with respect to the 2-Weil pairing, we have $\deg \alpha_0 \equiv -1 \pmod{2^{d+1}}$. Therefore, we have $\deg \alpha \equiv -m^2 \pmod{2^{d+1}}$. On the other hand, we have

$$\deg \alpha = N_1 N_2 \equiv -N_1^2 \pmod{2^{d+1}},$$

which means $m \equiv N_1 u \pmod{2^{d+1}}$ for a square root u of 1 modulo 2^{d+1}. In the case $d \geq 2$, there are 4 possible values of u. If $u \equiv 1 \pmod{2^{d+1}}$, then we use α. If $u \equiv -1 \pmod{2^{d+1}}$, then we use $-\alpha$. Otherwise, we may rerun the algorithm for finding α.

There are two possible approaches to find e, N_1, N_2 and α. The first is to find endomorphisms α from $\mathbb{Z}\alpha_0 + 2^{d+1}\mathcal{O}_0$. However, as far as our knowledge, there is no efficient method to take an endomorphism $\alpha \in \mathbb{Z}\alpha_0 + 2^{d+1}\mathcal{O}_0$ of a desired degree. The second is to find $\alpha_1 \in \mathbb{Z} + 2^{d+1}\mathcal{O}_0$ such that $\alpha \coloneqq \alpha_1\alpha_0$ satisfies the desired conditions.

The second approach cannot find α whose degree is as small as in the original method. We explain the reason as follows: since we require that $\deg \alpha$ is smaller than or equal to approximately $2^{3d+3}p$ which is the degree of endomorphisms in the original method, the degree of α_1 should be $\approx 2^{d+1}$. In our collision attack, we assume that $2^{d+1} < p$, otherwise our attack is less efficient than Costello-Smith's method in Section 5.3. Therefore, we obtain $\alpha_0 \in \mathbb{Z}[\omega] \subset \mathcal{O}_0$. The number of all endomorphisms $\in \mathbb{Z}[\omega]$ whose degree is less than or equal to is at most $4 \cdot 2^{d+1}$. For a fixed positive integer N, there are at most $\log_2 N$ divisors of N. Therefore, for a fixed α_1, we have at most $2^{d+1}\log_2(2^{3d+3}p)$ chances to test whether $\deg \alpha$ decomposes into the product of N_1 and N_2 such that $N_1 + N_2$ is a power of two. On the other hand, the ratio of the number of positive integers which is a power of two and is less than or equal to $2^{3d+3}p$ is $\lfloor \log_2(2^{3d+3}p) \rfloor / (2^{3d+3}p)$. Therefore, the probability that we find N_1 and N_2 is at most

$$2^{d+1}\log_2(2^{3d+3}p) \cdot \frac{\lfloor \log_2(2^{3d+3}p) \rfloor}{2^{3d+3}p} \leq \frac{(\log_2(2^{3d+3}p))^2}{2^{2d+2}p}.$$

This probability is too small to expect to exist α_1 as we want.