Solving the Constructive Deuring Correspondence via the Kohel-Lauter-Petit-Tignol Algorithm

Yuta Kambe[1, 2], Masaya Yasuda[2, *], Masayuki Noro[2], Kazuhiro Yokoyama[2], Yusuke Aikawa[3], Katsuyuki Takashima[4], Momonari Kudo[5]

[1] Sugakubunka
[2] Rikkyo University
[3] Mitsubishi Electric Corporation
[4] Waseda University
[5] The University of Tokyo

Received: 1st June 2021 | Revised: 1st August 2021 | Accepted: 1st September 2021

Abstract *For an odd prime p, let E_0 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ with $\mathcal{O}_0 = \mathrm{End}(E_0)$. The Deuring correspondence gives a one-to-one correspondence between isogenies $\varphi_I : E_0 \longrightarrow E_I$ and left $\mathcal{O}_0$-ideals I. The constructive Deuring correspondence is equivalent to the problem that computes the j-invariant of the curve E_I corresponding to given I. In this paper, we compute the j-invariant of E_I via the Kohel-Lauter-Petit-Tignol (KLPT) algorithm that seeks an ideal J of smooth reduced norm $\mathrm{Nrd}(J)$ such that $E_J \simeq E_I$. The target j-invariant can be obtained by computing $\varphi_J : E_0 \longrightarrow E_J$. For every prime factor ℓ of $\mathrm{Nrd}(J)$, we use symbolic formulas related with isogenies to compute a basis of the ℓ-torsion group $E_0[\ell]$, a bottleneck part in computing φ_J. We demonstrate the efficacy of our method by showing our implementation results for numerical examples in primes p of up to 25 bits.*

Keywords: Supersingular elliptic curves, Endomorphisms, Quaternion algebras, The Deuring correspondence

2010 Mathematics Subject Classification: 14H52, 14G50

1 INTRODUCTION

Since proposals of the hash function of [7] and the key exchange of [21], isogenies between supersingular elliptic curves have been actively used in building modern cryptosystems. In particular, SIKE [22] was selected in July 2020 as an alternate candidate in the standardization project of post-quantum cryptography by the National Institute of Standards and Technology [29, 31]. Furthermore, a number of new isogeny-based cryptosystems have been recently proposed, such as CSIDH [4] and OSIDH [8] for key exchange, SÉTA [20] and SiGamal [30] for public-key encryption, and SeaSign [11] and SQISign [12] for signature. The security of supersingular isogeny-based cryptography relies on the hardness of finding an isogeny connecting two given supersingular elliptic curves. For every prime p, there exists a one-to-one correspondence, called the *Deuring correspondence* [10], between the j-invariants of supersingular elliptic curves over $\mathbb{F}_{p^2}$ and the maximal orders in a quaternion algebra $B_{p,\infty}$ over $\mathbb{Q}$ ramified at both p and the point at infinity. In [26], Kohel-Lauter-Petit-Tignol provided a probabilistic polynomial-time algorithm solving the quaternion analogue of an isogeny problem under the Deuring correspondence. It is important for both cryptanalyses [14] and cryptographic constructions [12, 18]. (Recently, a generalization of the KLPT algorithm was proposed in [12] to build a compact signature scheme.) In computational number theory, the KLPT algorithm is also a useful tool in [14, 15] for a reduction from the problem of computing the endomorphism ring of a supersingular elliptic curve to the path-finding problem in an isogeny graph.

The *constructive Deuring correspondence* is the problem that computes the j-invariant of a supersingular elliptic curve corresponding to a given maximal order in $B_{p,\infty}$ under the Deuring correspondence. It is related to computational problems for supersingular elliptic curves, their isogeny graphs, and endomorphism rings, which are closely connected to the security of some isogeny-based cryptosystems [14]. A simple approach for the constructive Deuring correspondence is to list all isomorphism classes of supersingular elliptic curves together with information of their maximal order in $B_{p,\infty}$ (see [5, 27], and also [6] for an improvement). This approach has complexity at least exponential in $\log p$ since there are roughly $\left\lfloor \frac{p}{12} \right\rfloor$ isomorphism classes of supersingular elliptic curves over $\mathbb{F}_{p^2}$. Then we consider another approach. Fix a supersingular elliptic curve E_0 over $\mathbb{F}_{p^2}$, and set $\mathcal{O}_0 = \mathrm{End}(E_0)$ that is a maximal order in $B_{p,\infty}$. The Deuring correspondence gives a one-to-one correspondence between isogenies

*Corresponding Author: myasuda@rikkyo.ac.jp

$E_0 \longrightarrow E$ and left O_0-ideals [25, 41]. Then the constructive Deuring correspondence is equivalent to the problem that computes the j-invariant of the supersingular elliptic curve E_I corresponding to a given left O_0-ideal I [14].

In this paper, we aim to solve the equivalent problem of the constructive Deuring correspondence via the KLPT algorithm [26] that seeks an equivalent ideal J of I with smooth norm $\mathrm{Nrd}(J)$. (See [34] for his implementation report of the same approach for small primes p.) Instead of directly computing an isogeny $\varphi_I : E_0 \longrightarrow E_I$, the target j-invariant $j(E_I)$ can be obtained by computing another isogeny $\varphi_J : E_0 \longrightarrow E_J$ since $E_I \simeq E_J$, and the isogeny φ_J can be computed more efficiently for smaller $\mathrm{Nrd}(J)$ since $\deg \varphi_J = \mathrm{Nrd}(J)$. Specifically, we use the modified KLPT algorithm in [23] that performs an exhaustive search in the prime norm algorithm of [26] to find an ideal J with small $\mathrm{Nrd}(J)$. Our main contribution is to improve a basis computation of the ℓ-torsion group $E_0[\ell]$ for every prime factor ℓ of $\mathrm{Nrd}(J)$, a dominant part of computing the isogeny φ_J. In general, the ℓ-th division polynomial $\psi_\ell(x)$ is useful to compute a basis of $E_0[\ell]$, but it is computationally expensive to handle $\psi_\ell(x)$ for large ℓ. To resolve the difficulty, we compute a kernel polynomial [38] (or called an Elkies polynomial) that is a factor of $\psi_\ell(x)$. (In elliptic curve cryptography, kernel polynomials play a central role in the Schoof-Elkies-Atkin (SEA) algorithm for determining the order of an elliptic curve over a finite field. E.g., see [2, Chapter VII].) Specifically, we make use of *symbolic formulas* related with isogenies over $\mathbb{Q}$ in [32], which had been obtained using Gröbner basis computation for algebraic constraints derived from Vélu's formula [40]. Such symbolic formulas enable us to obtain the first coefficient of a kernel polynomial $F(x)$ and then recover the whole polynomial like in the SEA algorithm. An ℓ-torsion point in E_0 can be obtained by factorizing $F(x)$ into irreducible factors over $\mathbb{F}_{p^2}$. The complexity is $O(\ell^3 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$ since $\deg F(x) = \frac{\ell-1}{2}$ while that of factorization of $\psi_\ell(x)$ is $O(\ell^6 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$ since $\deg \psi_\ell(x) = \frac{\ell^2-1}{2}$. In other words, we use pre-computed symbolic formulas to reduce the online running time of a basis computation of $E_0[\ell]$. (Symbolic formulas are available for several primes ℓ in [32] like modular polynomials.) There is a trade-off between the cost of (offline) Gröbner basis computation for symbolic formulas and the cost of online computation of $E_0[\ell]$. To demonstrate the efficacy of our method, we show our implementation results for several numerical examples. While experiments for primes p of up to around 10 bits were conducted in [34], our method enables us to run in practice for larger primes p.

2 MATHEMATICAL PRELIMINARIES

In this section, we review basic definitions and properties of quaternion algebras and elliptic curves over finite fields to introduce the Deuring correspondence over supersingular elliptic curves.

2.1 QUATERNION ALGEBRAS, THEIR ORDERS, AND IDEALS

For a prime p with $p \equiv 3 \pmod 4$, we handle quaternion algebras over $\mathbb{Q}$ ramified at p and the point at infinity. Such any algebra can be written as $B_{p,\infty} := \mathbb{Q}\langle \mathbf{i}, \mathbf{j} \rangle$ with $\mathbf{i}^2 = -1$, $\mathbf{j}^2 = -p$ and $\mathbf{k} := \mathbf{ij} = -\mathbf{ji}$. Every element of $B_{p,\infty}$ can be expressed as $\alpha = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ with $a, b, c, d \in \mathbb{Q}$, and its *conjugation* is defined as $\bar{\alpha} := a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}$. The *reduced trace* and the *reduced norm* of α are respectively defined as $\mathrm{Trd}(\alpha) := \alpha + \bar{\alpha} = 2a$ and $\mathrm{Nrd}(\alpha) := \alpha \cdot \bar{\alpha} = a^2 + b^2 + p(c^2 + d^2)$. The reduced trace and norm are additive and multiplicative, respectively.

A $\mathbb{Z}$-lattice $O \subseteq B_{p,\infty}$ of rank 4 is called an *order* if it forms a subring of $B_{p,\infty}$. In particular, it is said *maximal* if it is not properly contained in any other order. The quaternion algebra $B_{p,\infty}$ includes several maximal orders such as $\left\langle 1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2} \right\rangle_{\mathbb{Z}}$. Fix a maximal order O of $B_{p,\infty}$. An (integral) *left O-ideal* is a $\mathbb{Z}$-lattice $I \subseteq O$ that satisfies $\alpha I \subseteq I$ for every $\alpha \in O$. The reduced norm of I is defined as $\mathrm{Nrd}(I) := \gcd(\{\mathrm{Nrd}(\alpha) : \alpha \in I\})$. Every left O-ideal can be represented as $I = ON + O\alpha$ with $N = \mathrm{Nrd}(I)$ for some $\alpha \in I$. Two non-zero left O-ideals I and J are said *equivalent* if and only if there exists an element q of $B_{p,\infty}$ such that $J = Iq$.

2.2 ELLIPTIC CURVES, THEIR ISOGENIES, AND ENDOMORPHISM RINGS

Every elliptic curve over a finite field $\mathbb{F}_q$ of characteristic $p \geq 5$ is defined by a (short) Weierstrass equation $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{F}_q$. Its discriminant and j-invariant are defined as $\Delta(E) = -16(4a^3 + 27b^2) \neq 0$ and $j(E) = -1728\frac{(4a)^3}{\Delta(E)}$, respectively. Two curves are isomorphic over an algebraic closure $\overline{\mathbb{F}}_q$ of $\mathbb{F}_q$ if and only if they have the same j-invariant. In addition, there exists an elliptic curve E over $\mathbb{F}_q$ with j-invariant $j(E)$ equal to a given element $j \in \mathbb{F}_q$. The set of $\mathbb{F}_q$-rational points on E as $E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q^2 : y^2 = x^3 + ax + b\} \cup \{\infty_E\}$ forms an abelian group, where ∞_E denotes the point at infinity that plays the zero element. The order of $E(\mathbb{F}_q)$ is represented as $\#E(\mathbb{F}_q) = q + 1 - t$, where t denotes the trace of the q^{th}-power Frobenius map. An elliptic curve over $\mathbb{F}_q$ is said *supersingular* if its trace t is divisible by p. Every supersingular curve has its j-invariant defined over $\mathbb{F}_{p^2}$. Let $E[n]$ denote the subgroup of $E(\overline{\mathbb{F}}_q)$ of n-torsion points for every $n \geq 2$.

An *isogeny* is a morphism $\phi : E \longrightarrow E'$ between two elliptic curves E and E' satisfying $\phi(\infty_E) = \infty_{E'}$. A non-zero isogeny $\phi : E \longrightarrow E'$ induces an injection of function fields $\phi^* : \overline{\mathbb{F}}_q(E') \longrightarrow \overline{\mathbb{F}}_q(E)$. The degree of ϕ is defined as $\deg \phi = [\overline{\mathbb{F}}_q(E) : \phi^*\overline{\mathbb{F}}_q(E')]$. In particular, we say that ϕ is *separable* if the extension $\overline{\mathbb{F}}_q(E)/\phi^*\overline{\mathbb{F}}_q(E')$ is separable. A non-zero isogeny ϕ also induces a surjective group homomorphism from $E(\overline{\mathbb{F}}_q)$ to $E'(\overline{\mathbb{F}}_q)$, and its kernel is a finite subgroup of $E(\overline{\mathbb{F}}_q)$, denoted by $E[\phi]$. It holds $\deg \phi = \#E[\phi]$ if ϕ is separable. Conversely, given a finite subgroup C of $E(\overline{\mathbb{F}}_q)$, there are an elliptic curve E' and a separable isogeny $\phi : E \longrightarrow E'$ with $E[\phi] = C$ (see [39, Chapter III]). An isogeny $\phi : E \longrightarrow E$ is called an *endomorphism*. The set of endomorphisms, denoted by $\mathrm{End}(E)$, has a ring structure (see [39]). If E is supersingular, the endomorphism ring of E is a maximal order $\mathcal{O}$ of the quaternion algebra $B_{p,\infty}$.

2.3 THE DEURING CORRESPONDENCE OVER SUPERSINGULAR ELLIPTIC CURVES

It was shown in [10] that for every prime p, the map $E \longmapsto \mathrm{End}(E)$ gives a bijection between the j-invariants of supersingular elliptic curves over $\mathbb{F}_{p^2}$ up to Galois conjugacy, and the maximal orders in the quaternion algebra $B_{p,\infty}$ up to the equivalence relation given by $\mathcal{O} \sim \mathcal{O}'$ if and only if $\mathcal{O} = \alpha^{-1}\mathcal{O}'\alpha$ for some $\alpha \in B_{p,\infty}$. Fixed a supersingular elliptic curve E_0 over $\mathbb{F}_{p^2}$ with $\mathcal{O}_0 = \mathrm{End}(E_0)$, the Deuring correspondence gives an equivalence of categories between supersingular elliptic curves and left $\mathcal{O}_0$-ideals [25, Chapter 5]. In particular, a one-to-one correspondence between isogenies $E_0 \longrightarrow E$ and left $\mathcal{O}_0$-ideals is given as below [41, Chapter 42]; For a left $\mathcal{O}_0$-ideal I with reduced norm $\mathrm{Nrd}(I)$ coprime to p, define its corresponding kernel $E_0[I] \subseteq E_0(\overline{\mathbb{F}}_p)$ to be the set

$$E_0[I] := \left\{P \in E_0(\overline{\mathbb{F}}_p) : \alpha(P) = \infty_{E_0}, \forall \alpha \in I\right\}.$$

Then the isogeny corresponding to I is given by $\varphi_I : E_0 \longrightarrow E_I := E_0/E_0[I]$. We have $\deg \varphi_I = \mathrm{Nrd}(I)$ by [41, Proposition 42.2.16]. Two curves E_I and E_J are isomorphic if their corresponding left $\mathcal{O}_0$-ideals I and J are equivalent [41, Lemma 42.2.13]. Conversely, for an isogeny $\varphi : E_0 \longrightarrow E$, the corresponding ideal is given by

$$I_\varphi := \left\{\alpha \in \mathcal{O}_0 : \alpha(P) = \infty_{E_0}, \forall P \in \ker \varphi\right\}.$$

3 BASIS COMPUTATION OF TORSION GROUPS

In this section, we give a new method to find a basis of each torsion group of an elliptic curve over a finite field. This can solve a bottleneck part of basis computation for the constructive Deuring correspondence (see Step B-2 in Subsection 4.1.2 below). We first recall the simplest method for computing a basis using division polynomials, and then present our method using kernel polynomials. Our method can be considered as an analogue of the SEA algorithm for counting the number of points of an elliptic curve over a finite field.

3.1 DIVISION POLYNOMIALS

Let $E_0 : y^2 = x^3 + ax + b$ be an elliptic curve over a finite field $\mathbb{F}_q$ of characteristic $p \geq 5$. Division polynomials associated with E_0 are recursively defined as

$$\begin{cases} \psi_0 = 0, \quad \psi_1 = 1, \quad \psi_2 = 2y, \quad \psi_3 = 3x^4 + 6ax^2 + 12bx - a^2, \\ \psi_4 = 4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3), \\ \psi_{2m+1} = \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3 \quad (m \geq 2), \\ \psi_{2m} = (\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2)\psi_m/2y \quad (m \geq 3). \end{cases}$$

For every odd integer $n \geq 3$, the n-th division polynomial ψ_n is a polynomial in x over $\mathbb{F}_q$ of degree $\frac{n^2-1}{2}$. Furthermore, the roots of $\psi_n(x)$ are the x-coordinates of n-torsion points in $E_0[n] \setminus \{\infty_{E_0}\}$ (see [39]), that is, $(x, y) \in E_0[n] \iff \psi_n(x) = 0$. Thus division polynomials are useful to compute a basis of a torsion group in E_0. However, it becomes more computationally expensive to compute the n-th division polynomial for larger n.

3.2 VÉLU'S FORMULA AND KERNEL POLYNOMIALS

Let ℓ be an odd prime number with $\ell \neq p$. Let S be a subgroup of $E_0(\overline{\mathbb{F}}_q)$ of order ℓ, and set $S^* = S \setminus \{\infty_{E_0}\}$. Then a separable isogeny $\phi_S : E_0 \longrightarrow \widetilde{E}_0 := E_0/S$ with kernel S can be written as

$$\phi_S(x, y) = \left(\frac{N_S(x)}{D_S(x)},\ y\left(\frac{N_S(x)}{D_S(x)}\right)'\right), \tag{1}$$

where $D_S(x)$ is the polynomial defined by $D_S(x) = \prod_{P \in S^*}(x - x_P) = x^{\ell-1} - s_1x^{\ell-2} + s_2x^{\ell-3} - s_3x^{\ell-4} + \cdots + s_{\ell-1}$ and the polynomial $N_S(x)$ is related to $D_S(x)$ through the formula

$$\frac{N_S(x)}{D_S(x)} = \ell x - s_1 - (3x^2 + a)\frac{D_S'(x)}{D_S(x)} - 2(x^3 + ax + b)\left(\frac{D_S'(x)}{D_S(x)}\right)'. \tag{2}$$

Here $T'(x)$ denotes the derivative of a function $T(x)$ and x_P the x-coordinate of a point $P \in E_0 \backslash \{\infty_{E_0}\}$. This is called Vélu's formula [40]. (Precisely, this is a modified form in [3]. Such form was discovered much earlier in [13].) With the first three coefficients s_1, s_2, s_3 of $D_S(x)$, set $v = a(\ell-1)+3(s_1^2-2s_2)$ and $w = 3as_1+2b(\ell-1)+5(s_1^3-3s_1s_2+3s_3)$. Then the Weierstrass equation for $\widetilde{E}_0$ is given by $y^2 = x^3 + \tilde{a}x + \tilde{b}$ with $\tilde{a} = a - 5v$ and $\tilde{b} = b - 7w$. Now we partition the set S^* into two parts S^+ and S^- such that $S^* = S^+ \cup S^-$ and $S^- = \{-P : P \in S^+\}$. The *kernel polynomial* (or called the *Elkies polynomial*) associated with S is defined as

$$F_S(x) = \prod_{P \in S^+} (x - x_P) = x^k + t_1 x^{k-1} + t_2 x^{k-2} + \cdots + t_k \tag{3}$$

with $k = \frac{\ell-1}{2}$. It is clear that $D_S(x) = F_S(x)^2$ and $s_1 = -2t_1$. Thus, by (2), the coefficients of $D_S(x)$ and $N_S(x)$ are represented as polynomials in $a, b, t_1, \dots, t_k$ and so coefficients of rational functions in the formula (1). An interesting relation among t_i's is given in [38]; To the curve E_0, associate the reduced Weierstrass $\wp$-function by

$$\wp(z) = \frac{1}{z^2} + \sum_{k=1}^{\infty} c_k z^{2k} \quad \text{with} \quad c_1 = -\frac{a}{5}, \quad c_2 = -\frac{b}{7}, \quad c_k = \frac{3}{(k-2)(2k+3)} \sum_{j=1}^{k-2} c_j c_{k-1-j} \quad (k \geq 3).$$

For the curve $\widetilde{E}_0$, consider its isomorphic curve $\widehat{E}_0 : y^2 = x^3 + \hat{a}x + \hat{b}$ with $\hat{a} = \ell^4 \tilde{a}$ and $\hat{b} = \ell^6 \tilde{b}$, and define the function $\hat{\wp}(z)$ and its coefficients $\hat{c}_k$'s for $\widehat{E}_0$ in the same manner. Then the kernel polynomial $F_S(x)$ satisfies

$$z^{\ell-1} F_S(\wp(z)) = \exp\left(-\frac{1}{2} t_1 z^2 - \sum_{k=1}^{\infty} \frac{\hat{c}_k - \ell c_k}{(2k+1)(2k+2)} z^{2k+2} \right). \tag{4}$$

Precisely, this is obtained by reduction from $\mathbb{C}$, and hence a prime p must be large enough for it to hold over a finite field of characteristic p. From this equation, every coefficient t_i for $i \geq 2$ can be represented using t_1, c_k's and $\hat{c}_k$'s. For examples, the first few coefficients of $F_S(x)$ are given as below:

$$\begin{cases} t_2 = \dfrac{t_1^2}{2} - \dfrac{\hat{c}_1 - \ell c_1}{12} - \dfrac{\ell - 1}{2} c_1, \\ t_3 = \dfrac{t_1^3}{6} - \dfrac{\hat{c}_2 - \ell c_2}{30} - \dfrac{\hat{c}_1 - \ell c_1}{12} t_1 - \dfrac{\ell - 1}{2} c_2 - \dfrac{\ell - 3}{2} c_1 t_1, \\ \quad \vdots \end{cases} \tag{5}$$

3.3 SYMBOLIC FORMULAS OF ISOGENIES

For the isogeny $\phi_S : E_0 \longrightarrow \widetilde{E}_0 \simeq \widehat{E}_0$, we apply the formula (1) to the Weierstrass equation of $\widehat{E}_0$ as

$$y^2 \left\{ \left(\frac{N_S(x)}{D_S(x)} \right)' \right\}^2 = \left(\frac{N_S(x)}{D_S(x)} \right)^3 + \hat{a} \left(\frac{N_S(x)}{D_S(x)} \right) + \hat{b}.$$

We expand this equation as polynomials in x by using the relation $y^2 = x^3 + ax + b$ to obtain a system of algebraic equations. When we consider $a, b, \hat{a}, \hat{b}$ and the coefficients t_i's of $F_S(x)$ as variables, the system of algebraic equations is defined over $\mathbb{Q}[a, b, \hat{a}, \hat{b}, t_1, \dots, t_k]$, since the coefficients of $D_S(x)$ and $N_S(x)$ can be rewritten as polynomials in $a, b, t_1, \dots, t_k$ described above. From the system of algebraic equations, several explicit symbolic formulas of isogenies of degree ℓ are shown in [32] for odd primes ℓ. Specifically, Weierstrass coefficients a, b of E_0 are regarded as symbolic variables in [32], and symbolic formulas of isogenies from E_0 are given using symbolic variables a, b over $\mathbb{Q}$. In this setting, all of $t_1, \dots, t_k$, $\hat{a}$ and $\hat{b}$ are shown to be integral over $\mathbb{Q}[a, b]$ [32, Lemma 3.2]. In particular, the minimal polynomial $m_\ell(t_1; a, b)$ of the first coefficient t_1 of the kernel polynomial $F_S(x)$ is calculated over $\mathbb{Q}[a, b]$ in [32] for a subgroup S of E_0 of order ℓ. The actual calculation was performed by using efficient Gröbner basis computation of the ideal associated with the system of algebraic equations. The polynomial $m_\ell(t_1; a, b)$ depends on ℓ (rather on S), and its degree is $\ell + 1$ [32, Lemma 3.5]. For an elliptic curve E_0 over a finite field $\mathbb{F}_q$, we can substitute its Weierstrass coefficients a, b into $m_\ell(t_1; a, b)$ to obtain a polynomial $m_\ell(t_1)$ over $\mathbb{F}_q$ of degree $\ell + 1$. The roots of $m_\ell(t_1)$ correspond to $\ell + 1$ subgroups $S_1, \dots, S_{\ell+1}$ of order ℓ in $E_0[\ell]$ if the characteristic p of $\mathbb{F}_q$ does not divide ℓ. Precisely, the roots of $m_\ell(t_1)$ coincide with the first coefficients of kernel polynomials $F_{S_i}(x)$ for $1 \leq i \leq \ell + 1$. (Recall Equation (3) for the first coefficient t_1 of a kernel polynomial.) Indeed, the ℓ-th division polynomial $\psi_\ell(x)$ can be factored with the kernel polynomials as

$$\psi_\ell(x) = \ell \prod_{i=1}^{\ell+1} F_{S_i}(x). \tag{6}$$

In addition, it follows from [32, Theorem 3.9] that Weierstrass coefficients $\hat{a}$ and $\hat{b}$ of $\widehat{E}_0$ have a rational univariate representation (RUR) with respect to t_1 as

$$\hat{a} = \frac{A(t_1; a, b)}{m'_\ell(t_1; a, b)}, \quad \hat{b} = \frac{B(t_1; a, b)}{m'_\ell(t_1; a, b)} \tag{7}$$

for some elements $A(t_1; a, b)$ and $B(t_1; a, b)$ of $\mathbb{Q}[t_1, a, b]$, where $m'_\ell(t_1; a, b)$ denotes the derivative of $m_\ell(t_1; a, b)$ with respect to t_1 (see [36] for the notion and properties of RUR). In other words, Weierstrass coefficients $\hat{a}$ and $\hat{b}$ of $\widehat{E}_0$ can be recovered from a root of $m_\ell(t_1; a, b)$ and a, b by substituting them for the RUR formula (7). (In contrast, the RUR formula (7) indicates that a multiple root of $m_\ell(t_1; a, b)$ can not determine the values of $\hat{a}$ and $\hat{b}$.) Moreover, from the associated ideal, for each coefficient t_i ($2 \le i \le k$), its polynomial representation in $a, b, t_1, \ldots, t_{i-1}$ can be computed, which corresponds to the formulas (5). Then, $t_2, \ldots, t_k$ can be recovered from a, b and $\hat{a}, \hat{b}$, and hence all coefficients of $F_S(x)$ can also be recovered using Equation (4) like (5).

3.4 APPLICATION OF SYMBOLIC FORMULAS TO FINDING TORSION POINTS

Here we apply symbolic formulas of isogenies in [32] to computing a basis of the ℓ-torsion group $E_0[\ell]$ for a (supersingular) elliptic curve $E_0 : y^2 = x^3 + ax + b$ over $\mathbb{F}_{p^2}$ and an odd prime $\ell \neq p$. A basic procedure is below:

(i) Get the minimal polynomial $m_\ell(t_1; a, b)$ of the first coefficient t_1 of a kernel polynomial over $\mathbb{Q}(a, b)$ from [32] with symbolic variables a, b. Then substitute Weierstrass coefficients $a, b \in \mathbb{F}_{p^2}$ of E_0 to obtain a polynomial $m(t_1)$ over $\mathbb{F}_{p^2}$ whose degree is $\ell + 1$.

(ii) Take a root $t_1 = \xi$ of $m(t_1)$, and recover a kernel polynomial $F_S(x)$ for some subgroup S of $E_0[\ell]$ with order ℓ. Specifically, we perform the below steps:
- Substitute $t_1 = \xi, a, b$ into the RUR formula (7) to compute Weierstrass coefficients $\hat{a}, \hat{b}$ of $\widehat{E}_0$.
- Compute coefficients $t_2, \ldots, t_k$ and recover a kernel polynomial $F_S(x)$ using Equation (4).

(iii) Take a root $x = \alpha_1$ of the kernel polynomial $F_S(x)$ to obtain a point $P = (\alpha_1, \beta_1)$ in $E_0[\ell]$ for some $\beta_1 \in \overline{\mathbb{F}}_{p^2}$ satisfying $\beta_1^2 = \alpha_1^3 + a\alpha_1 + b$. By factorization of $F_S(x)$ over $\mathbb{F}_{p^2}$, one of its irreducible factor $G(x)$ shall be taken as the defining polynomial of α_1 over $\mathbb{F}_{p^2}$ and also, by factorization of $y^2 - \alpha_1^3 - a\alpha_1 - b$ over $\mathbb{F}_{p^2}[x]/(G(x))$, its irreducible factor is taken as the defining polynomial of β_1 over $\mathbb{F}_{p^2}[x]/(G(x))$.

(iv) In the same way, we can obtain another point $Q = (\alpha_2, \beta_2)$ in $E_0[\ell]$ by taking another root $t_1 = \eta$ of the polynomial $m(t_1)$ for another kernel polynomial $F_{S'}$. Then the two points P, Q span the ℓ-torsion group as $E_0[\ell] = \langle P, Q \rangle$ since $Q \notin \langle P \rangle$ due to $\xi \neq \eta$. In these computation, by factorization of $F_{S'}(x)$ and that of $y^2 - \alpha_2^3 - a\alpha_2 - b$, the coordinates α_2, β_2 of Q are expressed as polynomials in α_1, β_1 over $\mathbb{F}_{p^2}$. (It can be shown that $F_{S'}(x)$ is factorized into linear factors.)

We note that $m_\ell(t_1)$ is factorized into linear factors over $\mathbb{F}_{p^2}$ and its factorization can be done in $O(\ell^3 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$ (see [19]). For each root of $m_\ell(t_1)$, we compute $\hat{a}, \hat{b}$ by the RUR formula (7) and recover $F_S(x)$ by using the formula (4) like (5). By using an efficient computation in [3], the computation of $F_S(x)$ can be done in $O(\ell^2)$ arithmetic operations in $\mathbb{F}_{p^2}$ (see also Remark 5 below). While the ℓ-th division polynomial $\psi_\ell(x)$ has degree $\frac{\ell^2-1}{2}$, our method computes two kernel polynomials of degree $\frac{\ell-1}{2}$ which are factors of $\psi_\ell(x)$ (recall Equation (6)). In particular, while the complexity of factorization of $\psi_\ell(x)$ is $O(\ell^6 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$, factorization of $F_S(x)$ only requires $O(\ell^3 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$. This shows that our computation is much faster than using the division polynomial $\psi_\ell(x)$ for large ℓ. But for (iv), the complexity of factorization $F_{S'}(x)$ over $\mathbb{F}_{p^2}[x]/(G(x))$ is $O(\ell^3 \deg(G(x))^3 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$. Hence it would be much better if we could avoid such factorization over the extended field. For this purpose, we can use some endomorphism of E_0, since actions of the endomorphism group are given in explicit and computable manner in our experimental setting (see the next subsection for our experiments).

3.4.1 SPECIAL CASE

Here we consider a special case where E_0 is defined over $\mathbb{F}_p$. (We take such a curve in our experiments. See Equation (10) below.) In this case, the p-th Frobenius map π on E_0 satisfies $\pi^2 + p = 0$ as endomorphisms of E_0 (see [39]). This implies that π^2 acts as scalar multiplication by $-p$ on E_0. Therefore any subgroup S of $E_0[\ell]$ of order ℓ is stable by the action of π^2, that is, $\pi^2(S) = S$. Thus any kernel polynomial $F_S(x)$ is also stable by the p^2-th Frobenius action, and all coefficients of $F_S(x)$ are defined over $\mathbb{F}_{p^2}$, namely, $F_S(x) \in \mathbb{F}_{p^2}[x]$. In particular, the p-th Frobenius action on $F_S(x)$ generates another kernel polynomial $F_{S'}(x)$ if $F_S(x) \in \mathbb{F}_{p^2}[x] \setminus \mathbb{F}_p(x)$. We can obtain a basis of the ℓ-torsion group $E_0[\ell]$ from roots of different kernel polynomials $F_S(x)$ and $F_{S'}(x)$. More generally, given a point $P \in E_0[\ell]$, another point $Q \notin \langle P \rangle$ can be obtained as $Q = f(P)$ for some endomorphism f of E_0 when the structure of $\mathrm{End}(E_0)$ is explicitly known (e.g., see Section 4.2.1 below for input data in our experiments).

Remark 1. *A kernel polynomial $F_S(x)$ can be applied to computing a point of E_0 of ℓ-power order. Indeed, we can recover two polynomials $D_S(x)$ and $N_S(x)$ from $F_S(x)$, and compute the isogeny $\phi_S(x, y)$ defined by Equation* (1). *Then we can obtain a point R of order ℓ^2 by taking it such that $\phi_S(R) = P$ for an ℓ-torsion point P. We repeat this procedure to compute a point of E_0 of order ℓ^k and a basis of $E_0[\ell^k]$ (see also [28] for details).*

3.4.2 COMPARISON WITH OTHER METHODS

Using modular polynomials Same as in the SEA algorithm for counting points on an elliptic curve, we may use the ℓ-th modular polynomial $\Phi_\ell(x, y)$ for recovering a kernel polynomial $F_S(x)$. Specifically, given an elliptic curve E_0 with j-invariant j_0, the roots of $\Phi_\ell(x, j_0)$ are in correspondence with the $\ell + 1$ cyclic subgroups of $E_0[\ell]$. In a method using modular polynomials, we take such a root to recover its corresponding kernel polynomial $F_S(x)$ by using derivatives $\partial\Phi_\ell/\partial x$ and $\partial\Phi_\ell/\partial y$ like [16, Algorithm 27] (see also [28] for such idea). The order of computational complexity is the same as ours. But our method is much faster in practice since [16, Algorithm 27] uses more computational steps such as computation of derivatives. (In other words, symbolic formulas of isogenies in [32] simplifies complicated calculation steps in advance.) Furthermore, the method using modular polynomials is applicable only to a case where $\Phi_\ell(x, j_0)$ has a *single* root since it uses derivatives of $\Phi_\ell(x, y)$. In particular, the method using modular polynomials cannot be applied to our experiments with $j_0 = 1728$ in the next section. Indeed, we verified from our preliminary experiments with $j_0 = 1728$ that the roots of $\Phi_\ell(x, j_0)$ are all multiple (see Subsection 4.2.1 for input curves in our experiments).

Random sampling method This method is probabilistic while ours and the method using modular polynomials are deterministic. This method starts to find the smallest integer r such that the order of $E_0(\mathbb{F}_{p^{2r}})$ is divisible by ℓ^2. The order $\#E_0(\mathbb{F}_{p^{2r}})$ is efficiently computed from the order $\#E_0(\mathbb{F}_{p^2})$. Precisely, we have $\#E_0(\mathbb{F}_{p^{2r}}) = (1 - \alpha^r)(1 - \beta^r)$ when we represent $\#E_0(\mathbb{F}_{p^2}) = (1 - \alpha)(1 - \beta)$ with $\alpha, \beta \in \mathbb{C}$. We then take a point R in $E_0(\mathbb{F}_{p^{2r}})$ randomly, and compute $P = cR$ for the cofactor $c = \#E_0(\mathbb{F}_{p^{2r}})/\ell^e$, where e is maximal such that c is an integer. Then the order of R is exactly equal to ℓ with a high probability for large ℓ. Since $\#E_0(\mathbb{F}_{p^{2r}}) = O(p^{2r})$, this method requires $O(r \log p)$ additions on $E(\mathbb{F}_{p^{2r}})$. Moreover, since $r = O(\ell)$, the complexity is $O(\ell^3 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$, which is the same as that of ours. When $r \ll \ell$, the random sampling method is much faster than ours in practice. When $r \approx \ell$, the running time of the random sampling method depends on the cost of addition on $E_0(\mathbb{F}_{p^{2r}})$, and our method is comparable to the random sampling method in performance.

Remark 2. *Factorization of the division polynomial $\psi_d(x)$ with $O(d) = O(\log p)$ is considered in the proof of [18, Lemma 5] to obtain a basis of the d-torsion group $E_0[d]$ in complexity analysis of computing the isogeny φ_I corresponding to a given left $\mathcal{O}_0$-ideal I under the Deuring correspondence with $\mathcal{O}_0 = \mathrm{End}(E_0)$ (see also [18, Algorithm 2] for its procedure). The proof of [18, Lemma 5] estimates that factorization of $\psi_d(x)$ can be done in $\widetilde{O}(\log^4 p)$ bit operations by fast polynomial factorization in [24] since $\deg \psi_d = O(\log^2 p)$. In contrast, our method enables us to handle kernel polynomials $F_S(X)$ with $\deg F_S(x) = O(\log p)$ for obtaining a basis of $E_0[d]$, and factorization of $F_S(x)$ can be done in much less complexity than that of $\psi_d(x)$. In particular, by using our method, the basis computation part might be not dominant in [18, Algorithm 2]. Indeed, our experimental results in Subsection 4.3.1 show that the basis computation part is not dominant in solving the constructive Deuring correspondence (the isogeny computation part is dominant in our experiments).*

4 SOLVING THE CONSTRUCTIVE DEURING CORRESPONDENCE

Given a maximal order $\mathcal{O}$ in a quaternion algebra $B_{p,\infty}$, the *constructive Deuring correspondence* asks us to compute the j-invariant of a supersingular elliptic curve such that its endomorphism ring is isomorphic to $\mathcal{O}$. As in the previous section, take a supersingular elliptic curve E_0 over $\mathbb{F}_{p^2}$ and set $\mathcal{O}_0 = \mathrm{End}(E_0)$ that is a maximal order in $B_{p,\infty}$. By [14, Algorithm 12], the constructive Deuring correspondence can be reduced to the following problem [1]; "Given a left $\mathcal{O}_0$-ideal I, compute the j-invariant of the supersingular elliptic curve $E_I = E_0/E_0[I]$ corresponding to I." In this section, we present how to solve this problem *in practice* via the KLPT algorithm [26] that finds an equivalent ideal J of an input left $\mathcal{O}_0$-ideal I with smooth reduced norm $\mathrm{Nrd}(J)$. A key idea is to compute the isogeny φ_J, alternative to the isogeny φ_I, to obtain the target j-invariant $j(E_I)$. Since $\deg \varphi_J = \mathrm{Nrd}(J)$ [41, Proposition 42.2.16], the isogeny φ_J can be factored as a composition of isogenies of degrees $\ell_i^{e_i}$ when $\mathrm{Nrd}(J) = \prod_{i=1}^{r} \ell_i^{e_i}$ with distinct small primes ℓ_i and $e_i \geq 1$. (We always assume that $p \neq \ell_i$ for all $1 \leq i \leq r$.) Thus the isogeny φ_J can be computed more efficiently as the reduced norm $\mathrm{Nrd}(J)$ is smaller and more smooth.

[1]Specifically, given two maximal orders $\mathcal{O}_0, \mathcal{O}$ in $B_{p,\infty}$, consider the set $I = I(\mathcal{O}_0, \mathcal{O}) = \{\alpha \in B_{p,\infty} : \alpha\mathcal{O}\bar{\alpha} \subseteq M\mathcal{O}_0\}$ where $M = [\mathcal{O}_0 : \mathcal{O}_0 \cap \mathcal{O}]$ denotes the index of the Eichler order $\mathcal{O}_0 \cap \mathcal{O}$ in $\mathcal{O}_0$. Then I is both a left $\mathcal{O}_0$-ideal and a right $\mathcal{O}$-ideal of reduced norm M [26, Lemma 8]. (The set I is called a *connecting ideal*.) Thus the endomorphism ring of E_I is isomorphic to $\mathcal{O}$ since $\mathrm{End}(E_I) \simeq \{\alpha \in B_{p,\infty} : I\alpha \subseteq I\} = \mathcal{O}$ (see [41, Chapter 17] for details).

4.1 OUTLINE OF PROCEDURE

Given a left $\mathcal{O}_0$-ideal I with $\mathcal{O}_0 = \mathrm{End}(E_0)$, the main procedure to compute the j-invariant $j(E_I)$ is divided into the below two steps:

Step A: By applying the KLPT algorithm [26], find an equivalent ideal J of the input ideal I such that the reduced norm of J is smooth.

Step B: Compute the isogeny $\varphi_J : E_0 \longrightarrow E_J$ corresponding to J to obtain the j-invariant $j(E_J) = j(E_I)$ since $E_I \simeq E_J$ (see the below commutative diagram).

$$\begin{tikzcd}
E_0 \arrow[rr, "\varphi_I"] \arrow[dr, "\varphi_J"'] & & E_I \arrow[dl, equal, "\simeq"] \\
& E_J &
\end{tikzcd}$$

4.1.1 DETAILS OF STEP A: THE KLPT ALGORITHM

Given a bound $B > 0$, an integer $n = \prod_{i=1}^{r} \ell_i^{e_i}$ with distinct primes ℓ_i is said *B-powersmooth* if $\ell_i^{e_i} \leq B$ for all $1 \leq i \leq r$. The KLPT algorithm takes a maximal order $\mathcal{O}_0$ in $B_{p,\infty}$ and a left $\mathcal{O}_0$-ideal I as input, and finds an equivalent left $\mathcal{O}_0$-ideal J of I with B-powersmooth reduced norm $\mathrm{Nrd}(J)$ for $B \approx \frac{7}{2}\log p$ (see [26] for heuristic analysis on B). The below lemma plays a key role in the KLPT algorithm:

Lemma 1 ([26]). *Let I be a left $\mathcal{O}_0$-ideal and α an element of I. Then an equivalent ideal $J = I\gamma$ with $\gamma = \bar{\alpha}/\mathrm{Nrd}(I)$ is a left $\mathcal{O}_0$-ideal of reduced norm* $\mathrm{Nrd}(\alpha)/\mathrm{Nrd}(I)$.

A basic procedure of Step A is as below (see [18, Algorithm 1] or [23, Section 3.1]). We take $B = \frac{7}{2}\log p$ as an initial smooth bound, and we increase it until we find an equivalent ideal J of I with smooth reduced norm $\mathrm{Nrd}(J)$.

Step A-1: Find $\delta \in I$ such that an equivalent left $\mathcal{O}_0$-ideal $I' := I\bar{\delta}/\mathrm{Nrd}(I)$ of I has a prime reduced norm N.

(i) Compute a Minkowski-reduced basis $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of I as a $\mathbb{Z}$-lattice.

(ii) Generate a random element $\delta = \sum_{i=1}^{4} x_i\alpha_i$ with small integers x_1, x_2, x_3, x_4, until the reduced norm of δ is equal to $\mathrm{Nrd}(I)$ times a prime N. Then $\mathrm{Nrd}(I') = \mathrm{Nrd}(\delta)/\mathrm{Nrd}(I) = N$ by Lemma 1.

Step A-2: Find $\beta \in I'$ with reduced norm NS for some odd B-powersmooth S.

(i) Find $\alpha \in I'$ with $I' = \mathcal{O}_0 N + \mathcal{O}_0\alpha$, by taking α as a small linear combination of a basis of I' until the condition $\gcd(\mathrm{Nrd}(\alpha), N^2) = N$ is satisfied.

(ii) Find $\beta_1 = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \in \mathcal{O}_0$ with odd reduced norm NS_1 for some B-powersmooth number S_1. Specifically, for a large enough B-powersmooth number S_1, generate a pair of small random integers (c, d) until the norm equation $a^2 + b^2 = NS_1 - p(c^2 + d^2)$ can be efficiently solved by Cornacchia's algorithm [9] to find a pair of integral solutions (a, b).

(iii) Find $\beta_2 = C\mathbf{j} + D\mathbf{k}$ with $C, D \in \mathbb{Z}$ satisfying $\alpha \equiv \beta_1\beta_2 \pmod{N\mathcal{O}_0}$ by linear algebra.

(iv) Find $\beta_2' \in \mathcal{O}_0$ with a powersmooth reduced norm S_2 and $\lambda \in \mathbb{Z}$ such that $\beta_2' \equiv \lambda\beta_2 \pmod{N\mathcal{O}_0}$. Write $\beta_2' = \lambda\beta_2 + N(a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k})$ to search five integers a, b, c, d, λ with $\lambda \notin N\mathbb{Z}$ satisfying

$$N^2(a^2 + b^2) + p\left\{(\lambda C + cN)^2 + (\lambda D + dN)^2\right\} = S_2 \tag{8}$$

for a large enough powersmooth number S_2. Specifically, given S_2, we perform the below steps:

- Consider Equation (8) modulo N as $p\lambda^2(C^2 + D^2) \equiv S_2 \pmod{N}$, from which a solution of λ is obtained. (We multiply S_2 by small primes if the equation cannot be solved.)
- Once λ is obtained, consider Equation (8) modulo N^2 as

$$p\lambda^2(C^2 + D^2) + 2p\lambda N(cC + dD) \equiv S_2 \pmod{N^2}. \tag{9}$$

  Pick $c \in \mathbb{Z}$ randomly to solve the equation for d.
- Given a triple of integers (λ, c, d), solve the equation

$$a^2 + b^2 = \frac{S_2 - p\left\{(\lambda C + cN)^2 + (\lambda D + dN)^2\right\}}{N^2}$$

  by Cornacchia's algorithm for a pair of integral solutions (a, b). (We pick a different pair of integers (c, d) if the equation cannot be solved.)

(v) Set $\beta = \beta_1\beta_2'$, whose reduced norm is NS_1S_2.

Step A-3: Output an equivalent ideal $J := I'\bar{\beta}/N$ of I, whose reduced norm is $S := S_1S_2$ by Lemma 1.

4.1.2 DETAILS OF STEP B: COMPUTING ISOGENIES

For the equivalent ideal J of the input I with smooth reduced norm, we compute the isogeny $\varphi_J : E_0 \longrightarrow E_J$ in Step B. A basic procedure of Step B is as below (see [34, Section 3.2] or [18, Section 2]):

Step B-1: Factor the reduced norm of J as $\mathrm{Nrd}(J) = \prod_{i=1}^{r} \ell_i^{e_i}$ with distinct primes ℓ_i and $e_i \geq 1$, and find a set of generators of J as $J = \langle g_1, g_2, g_3, g_4 \rangle_{\mathbb{Z}}$.

Step B-2: Set $\varphi_0 = \mathrm{id}_{E_0}$, and repeat the following procedure for $1 \leq i \leq r$:

(i) Compute a basis $\{P_i, Q_i\}$ of the torsion group $E_0[\ell_i^{e_i}]$ by our method described in the previous section.

(ii) Compute $g_j(P_i)$ and $g_j(Q_i)$ for every generator element g_j of J, and find a point R_i of order $\ell_i^{e_i}$ satisfying $g_j(R_i) = \infty_{E_0}$ for all generators g_j. The point R_i generates the group $\ker \varphi_J \cap E_0[\ell_i^{e_i}]$.

(iii) Compute an isogeny $\phi_i : E_{i-1} \longrightarrow E_i$ with kernel generated by the point $\varphi_{i-1}(R_i)$, and then compute a composition map $\varphi_i = \phi_i \circ \varphi_{i-1} : E_0 \longrightarrow E_i$.

Step B-3: The target curve E_J can be obtained by computing $\varphi_r : E_0 \longrightarrow E_r$, since $\ker \varphi_r = E_0[J]$ and hence $\varphi_r = \varphi_J$. (It holds $\ker \varphi_r \subseteq E_0[J]$ and $\deg \varphi_r = \mathrm{Nrd}(J) = \deg \varphi_J$ by construction, and $\ker \varphi_r = E_0[J]$.)

$$\begin{tikzcd}
R_i \in E_0 \arrow[rr, "\varphi_r = \varphi_J"] \arrow[d, "\varphi_{i-1}"'] \arrow[drr, "\varphi_i"] & & E_J \simeq E_I \\
E_{i-1} \arrow[rr, "\phi_i"] & & E_{i-1}/\langle \varphi_{i-1}(R_i) \rangle = E_i
\end{tikzcd}$$

4.2 IMPLEMENTATION

4.2.1 INPUT CURVES AND IDEALS

For an odd prime p such that $p \equiv 3 \pmod 4$, we fix a supersigular elliptic curve

$$E_0 : y^2 = x^3 + x \tag{10}$$

over $\mathbb{F}_{p^2}$ satisfying $j(E_0) = 1728$ and $\#E_0(\mathbb{F}_{p^2}) = (p+1)^2$. The endomorphism ring of E_0 is isomorphic to a maximal order $\mathcal{O}_0 = \left\langle 1, \mathbf{i}, \dfrac{1+\mathbf{k}}{2}, \dfrac{\mathbf{i}+\mathbf{j}}{2} \right\rangle_{\mathbb{Z}}$ in the quaternion algebra $B_{p,\infty}$. The explicit isomorphism is given by

$$B_{p,\infty} \longrightarrow \mathrm{End}(E_0) \otimes_{\mathbb{Z}} \mathbb{Q} \quad \text{with} \quad (1, \mathbf{i}, \mathbf{j}, \mathbf{k}) \longmapsto (1, \phi, \pi, \pi\phi), \tag{11}$$

where $\pi : (x, y) \mapsto (x^p, y^p)$ is the p-th Frobenius map and $\phi : (x, y) \mapsto (-x, uy)$ with $u^2 = -1$. To generate an input left $\mathcal{O}_0$-ideal I, we begin with using the method described in [34, Chapter 4]. Specifically, we repeat to randomly generate an integral square matrix $\mathbf{U}$ of size 4 with coefficients in $[-\lceil \log p \rceil, \lceil \log p \rceil]$, until the $\mathbb{Z}$-lattice of rank 4 spanned by the row vectors of $\mathbf{Ub}$ forms a left $\mathcal{O}_0$-ideal $\mathfrak{I}$, where a column vector $\mathbf{b} = \left(1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2}\right)^{\top}$ represents a canonical $\mathbb{Z}$-basis of the maximal order $\mathcal{O}_0$. The absolute value of $\mathbf{U}$ must be at least a square integer for $\mathfrak{I}$ to form a left $\mathcal{O}_0$-ideal. In particular, we have $\mathrm{Nrd}(\mathfrak{I}) = \sqrt{|\det(\mathbf{U})|}$ when the rows of $\mathbf{Ub}$ generate a left $\mathcal{O}_0$-ideal $\mathfrak{I}$. In general, the reduced norm of such an ideal $\mathfrak{I}$ is very small, and we add the following procedure to obtain an input ideal I with a large reduced norm; We select a random element $\gamma = \sum_{i=1}^{4} \gamma_i b_i$ in $\mathfrak{I}$ with coefficients γ_i in $[-\sqrt{p}\log p, \sqrt{p}\log p]$ such that $\mathrm{Nrd}(\gamma)/\mathrm{Nrd}(\mathfrak{I}) > \sqrt{p}$, where b_i denotes the i-th entry of $\mathbf{b}$ for each $1 \leq i \leq 4$. Then we take an equivalent ideal $\mathfrak{I}(\bar{\gamma}/\mathrm{Nrd}(\mathfrak{I}))$ of $\mathfrak{I}$ as an input left $\mathcal{O}_0$-ideal I. It follows from Lemma 1 that the reduced norm of the input ideal I is guaranteed to be greater than $\sqrt{p}$. As well as for $\mathfrak{I}$, the input ideal I is spanned over $\mathbb{Z}$ by the rows of $\mathbf{Vb}$ for some integral 4×4 matrix $\mathbf{V}$, and its reduced norm is given by $\sqrt{|\det(\mathbf{V})|}$.

Remark 3. *We denote the normalized norm map associated with $\mathfrak{I}$ by $q : \mathfrak{I} \longrightarrow \mathbb{Z}$ with $q(\alpha) = \frac{\mathrm{Nrd}(\alpha)}{\mathrm{Nrd}(\mathfrak{I})}$. For simplicity, we assume that integers of form $q(\alpha)$ behave like random numbers. Under this assumption, we expect that the above method could find an ideal I such that* $\mathrm{Nrd}(I) > \sqrt{p}$ *with high probability. The below experiments show that it can find an ideal I with a very large reduced norm* $\mathrm{Nrd}(I) \gg p$ *in practice.*

4.2.2 IMPLEMENTATION DETAILS AND COMPLEXITY ANALYSIS

For Step A Our implementation for Step A is based on the modified KLPT algorithm in [23]. Different from the original KLPT algorithm [26], we perform an exhaustive search for Step A-1 to take the minimum prime for $N = \mathrm{Nrd}(I')$. Specifically, we make the list of all pairs (N, δ), where $\delta = \sum_i^4 x_i \alpha_i \in I$ is an element with $x_i \in [-\lceil \log p \rceil, \lceil \log p \rceil]$ and $N = \mathrm{Nrd}(\delta)/\mathrm{Nrd}(I)$ is a prime. We then run the remaining part of the algorithm with the smallest N. If the remaining part does not find a solution, then we return to Step A-1 to change N to the next one until we obtain a solution. We also adopt Petit-Smith's improvement [33] that finds a small integral solution of Equation (9) for us to take a small size of S_2 in Step A-2 (iv). Specifically, if we have S_2, N, C, D on Equation (9), then the solutions for $c, d \in \mathbb{Z}$ form a two-dimensional affine lattice in $\mathbb{Z}^2$. Since it follows from Equation (8) that $S_2 \geq p\left\{(\lambda C + cN)^2 + (\lambda D + dN)^2\right\}$, we want to choice an integral pair (c, d) such that $\left\{(\lambda C + cN)^2 + (\lambda D + dN)^2\right\}$ is smallest among the solutions. It gives an instance of the closest vector problem, so we can find desired solution (c, d) by Babai's algorithms [1]. As in the original KLPT algorithm [26], we take $B = \frac{7}{2}\log p$ as an initial smooth bound, but we often increase it to find an integral solution of Equation (8) in Step A-2. (As seen from the below numerical examples, we increased a smooth bound up to about twice the initial bound $\frac{7}{2}\log p$ in our experiments.) We implemented Step A in SageMath [37], and ran this step on Intel Corei7-8750H@2.20GHz with 32GByte RAM. Since the running time of Step A-2 is dominant in the KLPT algorithm, we estimate that the running time of our program follows the complexity of the original KLPT algorithm. The complexity of the KLPT algorithm is heuristically known as $\widetilde{O}(\log^3 p)$ [18, Lemma 4].

Remark 4. *According to the implementation report in [23], the modified KLPT algorithm enables us to take smaller N and S_2 than the original algorithm. Specifically, Petit-Smith's improvement [33] enables us to find S_2 of size $O(pN^3)$, and hence an exhaustive search for small N can reduce the output quality of the KLPT algorithm. More specifically, the modified algorithm outputs a norm* $\mathrm{Nrd}(J)$ *that is about* 50 *bits smaller than the original algorithm for primes p from 15 to 45 bits. As for the running time, it is reported in [23] that the modified KLPT algorithm is slightly slower than the original algorithm for small primes p up to* 35 *bits due to an exhaustive search for Step A-1 (note that the exhaustive search is not dominant for the whole algorithm). On the other hand, it is faster in total for large primes such as* 45 *bits, since taking smaller N accelerates the processing of Step A-2.*

For Step B For an output ideal J of Step A, it is not computationally expensive to factorize smooth $\mathrm{Nrd}(J)$, and a set of generators $\{g_1, g_2, g_3, g_4\}$ of J can be obtained from input generators of I by construction in the KLPT algorithm. As described in Subsection 3.4, for every large prime factor ℓ of $\mathrm{Nrd}(J)$, we use symbolic formulas in [32] to recover a kernel polynomial $F_S(x)$ for some subgroup S of $E_0[\ell]$. (For small ℓ, we can use the division polynomial $\psi_\ell(x)$.) Note that symbolic formulas for E_0 had been computed in [32] for odd primes ℓ up to 81 with parameters a, b. Thus, we added "special" symbolic formulas for E_0 up to 131 with a parameter $a = 1$ and $b = 0$. We first find roots of $m_\ell(t_1)$ by its factorization into linear factors over $\mathbb{F}_{p^2}$, which can be done in $O(\ell^3 \log p)$ arithmetic operations in $\mathbb{F}_{p^2}$ (see [19]). For each root of $m_\ell(t_1)$, we compute $\hat{a}, \hat{b}$ by the RUR formula (7) and recover $F_S(x)$ by using the formula (4). By using an efficient computation in [3], the computation of $F_S(x)$ can be done in $O(\ell^2)$ arithmetic operations in $\mathbb{F}_{p^2}$ (see Remark 5 for the cost of recovering $F_S(x)$). We then factor $F_S(x)$ into irreducible polynomials over $\mathbb{F}_{p^2}$ to obtain an ℓ-torsion point $P = (\alpha_1, \beta_1) \in E_0[\ell]$. In our implementation, we represent $\mathbb{F}_{p^2}$ as $\mathbb{F}_p[u]/(u^2+1)$, and take an irreducible factor of $F_S(x)$ as the minimal polynomial of α_1 over $\mathbb{F}_{p^2}$. Since the degree of $F_S(x)$ is $k = \frac{\ell-1}{2}$, the complexity of this factorization is $O(\ell^3 \log^3(p^2)) = O(\ell^3 \log^3 p)$ bit operations in using classical arithmetic in $\mathbb{F}_{p^2}$. We take a square-root of $\alpha_1^3 + \alpha_1$ for β_1 since $P \in E_0$. Its complexity is $O(\ell^2 \log^3 p)$ bit operations. Thus it requires $O(\ell^3 \log^3 p)$ bit operations to find a point P in $E_0[\ell]$. As discussed in Subsection 3.4.1, we can generate another point $Q \in E_0[\ell]$ from P by computing $Q = f(P)$ for some endomorphism f of E_0. We often used the endomorphism ϕ corresponding to $\mathbf{i} \in B_{p,\infty}$ to take $Q = \phi(P) = (-\alpha_1, u\beta_1)$ (see the explicit isomorphism (11)). Indeed, the set $\{P, Q\}$ gives a basis of $E_0[\ell]$ if $F_S(-\alpha_1) \neq (-1)^k F_S(\alpha_1)$. It is not computationally expensive to compute $\phi(P)$, which is ignorable in complexity analysis.

Given a basis $\{P, Q\}$ of $E_0[\ell]$, we compute $g_j(P), g_j(Q)$ for every generator g_j of J. We then find a point R in a form P, Q or $P + sQ$ $(s = 1, \ldots, \ell - 1)$ such that $g_j(R) = \infty_{E_0}$ for all generators g_j of J. For checking $g_j(R) = \infty_{E_0}$, we first examine if $g_j(P), g_j(Q) = \infty_{E_0}$, and then find an integer s such that $g_j(P) + sg_j(Q) = \infty_{E_0}$ by using the idea of so-called the Baby-Step Giant-Step (BSGS) method. Specifically, we consider two sets $\{g_j(P) + g_j(Q), \ldots, g_j(P) + ug_j(Q)\}$ and $\{vg_j(Q), 2vg_j(Q), \ldots\}$ for $u, v \sim \sqrt{\ell}$ and find a pair of the same x-coordinate. The point R generates $\ker(J) \cap E_0[\ell]$. Let K denote the extension field of $\mathbb{F}_p$ defining all elements of $E_0[\ell]$, and let $d = [K : \mathbb{F}_p]$ denote its extension degree. The procedure of finding R requires $O(\ell)$ additions on the ℓ-torsion group $E_0[\ell]$ and $O(\log p)$ arithmetic operations in $\mathbb{F}_{p^d}$ for evaluating $g_j(P), g_j(Q)$ (see Remark 5 for the cost of evaluating $g_j(P), g_j(Q)$), and $O\left(\sqrt{\ell}^{1+\varepsilon}\right)$ additions on the ℓ-torsion group $E_0[\ell]$ for finding an integer s such that $g_j(P) + sg_j(Q) = \infty_{E_0}$. Thus its computational cost depends on the degree $d \leq 4k = 2(\ell - 1)$. The complexity of finding R is at most $O((\ell + \log p)\log^2(p^{4k})) = O(\ell^3 \log^2 p + \ell^2 \log^3 p)$ bit operations. Finally, we compute the isogeny $\phi_C : E_0 \longrightarrow E_0/C$ for the subgroup $C = \langle R \rangle$ of $E_0[\ell]$. Due to Vélu's formula (1), the isogeny computation is almost equivalent to recovering the kernel polynomial $F_C(x)$ associated with C in theory. In our implementation for computing $F_C(x) = \prod_{i=1}^{k}(x - x_{iR})$, we use a naive method where we simply multiply $x - x_R$, $x - x_{2R}$ and so on. This naive method requires k elliptic additions over $\mathbb{F}_{p^d}$, and it requires at most $O(\ell^3 \log^2 p)$ bit operations. We implemented Step B in Risa/Asir [35], a computer algebra system, and ran this step on MacBookPro16 (2019). We remark that the complexity of making the RUR formula (7) over $\mathbb{Z}$ to that over $\mathbb{F}_p$ is omitted since the size of coefficients of the RUR formula (7) for smaller ℓ are not so large and its timings are ignorable in our experiments.

Remark 5. *We give remarks for some detailed steps.*

- *The cost of evaluating $g_j(P)$ and $g_j(Q)$: Each g_j is given as a linear sum of endmorphisms $1, \phi, \pi, \pi\phi$. Since P is of exact order ℓ, we can reduce coefficients modulo ℓ and the computation can be done in $O(\ell)$ point additions among four points $P, \phi(P), \pi(P), \pi\phi(P)$. In particular, two points $\pi(P), \pi\phi(P)$ involve the Frobenius calculation that takes $O(\log p)$ operations over $\mathbb{F}_{p^d}$. Thus, in total, the cost of evaluation is done in $O(\log(p)) + \ell)$ arithmetic operations in $\mathbb{F}_{p^d}$ and in $O((\log p + \ell)d^2 \log^2 p)$ bit operations.*
- *The cost of evaluating formulae on the coefficients t_i's of $F_S(x)$: According to [3], once $t_1, \hat{a}, \hat{b}$ are known, all evaluations can be done in $O(\ell^2)$ arithmetic operations over $\mathbb{F}_{p^2}$ by using fast algorithms for power series expansion of the Weierstrass $\wp$-function (see also [2, VII.4.1]). Thus this part is not dominant. In our experiments, we used a naive method using a polynomial representation of each t_i shown in* (5) *that can be obtained as bi-product of isogeny formulas.*

Remark 6. *As described above, the minimal polynomial of the x-coordinate of an ℓ-torsion point $P = (\alpha_1, \beta_1)$ is given by an irreducible factor $G(x)$ of a kernel polynomial over $\mathbb{F}_{p^2}$. In our implementation, we represent α_1 as T in the extension field $\mathbb{F}_{p^2}[T]/(G(T))$ with a symbolic variable T. We can also find another ℓ-torsion point Q by factoring another kernel polynomial $F_{S'}(x)$ into irreducible factors over $\mathbb{F}_{p^2}$. But such polynomial factorization should be performed over the extension field $\mathbb{F}_{p^2}[T]/(G(T))$ to find an extension field where we can represent $uP + vQ$ for any integers u, v. The complexity of factoring $F_{S'}(x)$ over $\mathbb{F}_{p^2}[T]/(G(T))$ is $O(\ell^6 \log^3 p)$ bit operations in the worst-case since $\deg G(T) \leq k = \frac{\ell-1}{2}$. However, in our experimental settings, we can avoid such factorization over $\mathbb{F}_{p^2}[T]/(G(T))$ by generating another point $Q = f(P)$ for some endomorphism f of E_0 as mentioned in Subsection 3.4.1.*

4.3 EXPERIMENTS

4.3.1 NUMERICAL EXAMPLES

Below we present numerical examples for primes p of 15, 20, and 25 bits for Steps A and B in solving the constructive Deuring correspondence.

Example 1. *We take a* 15*-bit prime $p = 28499$, and consider a left $\mathcal{O}_0$-ideal I spanned by the rows of* $\mathbf{Vb}$ *with*

$$\mathbf{V} = \begin{pmatrix} -12706 & 14940 & -2267 & 15696 \\ 30636 & 14973 & -15696 & -2267 \\ -111803364 & -16137402 & -30636 & -14973 \\ 16152375 & -111834000 & -14973 & 30636 \end{pmatrix},$$

where $\mathbf{b}$ *is the same column vector as in Subsection 4.2.1. The reduced norm of I is given by* $\mathrm{Nrd}(I) = \sqrt{|\det(\mathbf{V})|}$, *decomposed into prime factors as* $3 \cdot 597537282301$. *Our implementation of the KLPT algorithm (Step A) took about 30 seconds to find an equivalent ideal J of I spanned by the rows of* $\mathbf{Wb}$ *with*

$$\mathbf{W} = \begin{pmatrix} 8512322886 & 375980085 & 39824784 & -75837522 \\ 300142563 & -8552147670 & 75837522 & 39824784 \\ 540642486813 & 275199438330 & -300142563 & 8552147670 \\ -283751586000 & 540342344250 & 8552147670 & 300142563 \end{pmatrix}.$$

In particular, we selected $N = 5$ in Step A-1 of the KLPT algorithm. The reduced norm of J is factored as

$$\mathrm{Nrd}(J) = 3^3 \cdot 5^2 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 43 \cdot 47 \cdot 53,$$

whose maximal prime factor does not exceed twice an initial smooth bound $B = \frac{7}{2} \log p \approx 35.9$ of the KLPT algorithm. In Table 1, we show the average running times of main procedures in Step B for every prime factor ℓ of $\mathrm{Nrd}(J)$. *(We ran each procedure 5 times, and show its average running time in the table.) As described in 4.2.2, the extension degree $[K : \mathbb{F}_p]$ affects running times of finding a generator of R of* $\ker(J) \cap E_0[\ell]$ *and computing the kernel polynomial $F_C(x)$ corresponding to the isogeny $\phi_C : E_0 \longrightarrow E_0/C$ for the subgroup $C = \langle R \rangle$ of E_0. From $F_C(x)$, the formula* (1) *can be computed immediately. We see from Table 1 that the running time of computing a basis of $E_0[\ell]$ monotonically increases with the size of ℓ, but the running times of other procedures depend on the extension degree $[K : \mathbb{F}_p]$. The total running time of Step B is approximately equal to the sum of the running times in Table 1.*

Table 1: Average running times (seconds) of main procedures in Step B in the case of a 15-bit prime (We also display the extension degree $[K : \mathbb{F}_p]$, where K denotes the field of defining all elements of $E_0[\ell]$)

Prime factors $\ell \geq 17$ of $\mathrm{Nrd}(J)$	17	19	23	29	31	37	43	47	53
Extension degree $[K : \mathbb{F}_p]$	32	2	44	56	60	36	42	92	26
Time of computing a basis of $E_0[\ell]$	0.02	0.03	0.04	0.05	0.05	0.07	0.12	0.11	0.15
Time of finding a generator of kernel	0.23	0.01	0.55	0.67	1.03	0.46	0.22	5.28	0.17
Time of computing an isogeny	0.06	0.003	0.19	0.46	0.60	0.23	0.11	2.88	0.33

Example 2. *We take a* 20*-bit prime* $p = 795299$*, and consider a left* $\mathcal{O}_0$*-ideal* I *spanned by the rows of* $\mathbf{Vb}$ *with*

$$\mathbf{V} = \begin{pmatrix} 36588 & 20732 & -12737 & 16125 \\ -36857 & 23851 & 16125 & 12737 \\ 1266198584 & -1603014637 & 19988 & 24797 \\ 1603026428 & 1266239304 & 12060 & -3863 \end{pmatrix}$$

as an input of the KLPT algorithm. The reduced norm of I *is factored as* $29 \cdot 1447497510289$*. Our implementation of the KLPT algorithm (Step A) took about 266 seconds to output an ideal* J *spanned by the rows of* $\mathbf{Wb}$ *with*

$$\mathbf{W} = \begin{pmatrix} -471644229843708735 & -607934833983252567 & -314526437963564 & -777357662522713 \\ 608712191645775280 & -471958756281672299 & -777357662522713 & 314526437963564 \\ 315722156098756937900 & 770430892473983699630 & -236368056972097506 & -304198832603905858 \\ -772105346446381728400 & 307279127189203437170 & -304513359041869422 & 235590699309574793 \end{pmatrix}.$$

In particular, we selected $N = 99431$ *in the KLPT algorithm. The reduced norm of* J *is factored as*

$$\mathrm{Nrd}(J) = 3^4 \cdot 5^2 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \cdot 53 \cdot 61 \cdot 67 \cdot 73 \cdot 79 \cdot 97 \cdot 101 \cdot 103,$$

whose maximal prime factor is around twice an initial smooth bound $B = \frac{7}{2} \log p \approx 47.6$*. In Table 2, we summarize the average running times of main procedures in Step B for every prime factor* ℓ *of* $\mathrm{Nrd}(J)$*.*

Table 2: Same as Table 1, but in the case of a 20-bit prime

Prime factors $\ell \geq 47$ of $\mathrm{Nrd}(J)$	47	53	61	67	73	79	97	101	103
Extension degree $[K : \mathbb{F}_p]$	92	104	60	44	36	78	48	100	204
Time of computing a basis of $E_0[\ell]$	0.11	0.14	0.21	0.25	0.28	0.62	0.65	0.82	0.86
Time of finding a generator of kernel	4.83	6.59	1.93	0.93	0.52	1.75	1.40	6.87	55.41
Time of computing an isogeny	2.89	4.58	1.37	0.73	0.54	4.73	1.55	9.66	70.73

Example 3. *We take a* 25*-bit prime* $p = 17795587$*. As an input of the KLPT algorithm, we take a left* $\mathcal{O}_0$*-ideal* I *spanned by the rows of* $\mathbf{Vb}$ *with*

$$\mathbf{V} = \begin{pmatrix} 3409696 & 661453 & -2562520 & 2805198 \\ 3466651 & -847176 & -2805198 & -2562520 \\ 3800130335697 & -4160012604594 & -652674 & 301377 \\ -4160012303217 & -3800129683023 & -301377 & -652674 \end{pmatrix}.$$

The reduced norm of I *is given by* $\mathrm{Nrd}(I) = 3 \cdot 11 \cdot 19 \cdot 101 \cdot 338048020593727$*. Our implementation of the KLPT (Step A) took about 162 seconds to find an equivalent ideal* J *of* I *generated by the rows of* $\mathbf{Wb}$ *with*

$$\mathbf{W} = \begin{pmatrix} -162793662102251031955209 6 & -254340478231797114580368 3 & -77554844880647944276 2 & 71084140616232319942 5 \\ -254269394091180882260425 8 & 162871216947131679899485 8 & -71084140616232319942 5 & -77554844880647944276 2 \\ 114926415776962938821430975 2 & -105361049572728995938442978 9 & -54314100362582637406476 1 & -84782316312020510068234 0 \\ -105445831889041016448511212 9 & -114872101676600356184024499 1 & 84782316312020510068234 0 & -54314100362582637406476 1 \end{pmatrix}.$$

In particular, we selected $N = 1482967$ *in the KLPT algorithm. The reduced norm of* J *is factorized as*

$$3^4 \cdot 5^3 \cdot 7^2 \cdot 11^2 \cdot 13 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 43 \cdot 47 \cdot 53 \cdot 61 \cdot 67 \cdot 73 \cdot 79 \cdot 83 \cdot 89 \cdot 97 \cdot 101 \cdot 103 \cdot 107 \cdot 109 \cdot 113 \cdot 127 \cdot 131,$$

whose maximal prime factor is around twice an initial smooth bound $B = \frac{7}{2} \log p \approx 58.4$ *as in the* 20*-bit case. In Table 3, we summarize the average running times of main procedures in Step B for every prime factor* ℓ *of* $\mathrm{Nrd}(J)$*.*

Table 3: Same as Table 1, but in the case of a 25-bit prime

Prime factors $\ell \geq 97$ of $\mathrm{Nrd}(J)$	97	101	103	107	109	113	127	131
Extension degree $[K : \mathbb{F}_p]$	16	200	102	106	108	56	126	130
Time of computing a basis of $E_0[\ell]$	0.65	0.85	1.41	1.63	1.07	1.15	3.44	4.23
Time of finding a generator of kernel	0.15	39.16	2.87	3.81	1.35	1.87	7.03	8.23
Time of computing an isogeny	0.17	65.97	11.82	13.53	3.71	2.66	24.81	27.86

4.3.2 SUMMARY AND DISCUSSION

In Table 4, we give a summary of numerical examples presented in Subsection 4.3.1 for solving the constructive Deuring correspondence. Note that the running times for Step B are approximately consistent with the sums of running times of Tables 1, 2 and 3, respectively, since the running times for small ℓ are not dominant for the total time of Step B. We see from Table 4 that as well as the KLPT algorithm [26] for Step A, our method for Step B can run in practice for primes p of up to 25 bits. In particular, our implementation of the KLPT algorithm outputs an ideal J whose reduced norm $\mathrm{Nrd}(J)$ roughly has size $O(p^4) \sim O(p^6)$, depending on the size of the prime N selected in Step A-1. Moreover, as mentioned in Subsection 4.3.1, the size of the maximum prime factor of $\mathrm{Nrd}(J)$ is around twice an initial smooth bound $B = \frac{7}{2} \log p$. From this, we estimate that the complexity of our method for Step B is roughly $O(\log^6 p)$ bit operations since the factorization of a kernel polynomial requires $O(\ell^3 \log^3 p)$ bit-complexity (in using classical arithmetic operations in $\mathbb{F}_{p^2}$) for every prime factor ℓ of $\mathrm{Nrd}(J)$ from Subsection 4.2.2. In our method for Step B, symbolic formulas related to isogenies are the most important ingredient for us to obtain a kernel polynomial $F_S(x)$ that is a factor of the ℓ-th division polynomial $\psi_\ell(x)$ with $\deg F_S(x) = \frac{\ell-1}{2}$. If we have no such symbolic formulas, we must factor $\psi_\ell(x)$ with $\deg \psi_\ell(x) = \frac{\ell^2-1}{2}$, which requires $O(\ell^6 \log^3 p) = O(\log^9 p)$ bit-complexity since it requires at most $\ell \approx \log p$. Indeed, in his master thesis [34], Ray directly factorized division polynomials $\psi_\ell(x)$ to obtain bases of torsion groups and reported that it took about 718 seconds to compute Step B in an 11-bit prime p. (As mentioned in Remark 2, factorization of $\psi_\ell(x)$ is considered also in the proof of [18, Lemma 5].) Table 4 shows that our method is much faster than the implementation report [34, Figure 4.1]. However, our method is *currently* only for primes p of up to around 25 bits since symbolic formulas related to isogenies are available in [32] for odd primes ℓ up to 131 for the curve E_0 defined by (10). For example, a case of a 30-bit prime p (resp., 40-bit prime p) requires symbolic formulas for primes around $2B = 7 \log p \approx 146$ (resp., $7 \log p \approx 194$).

Table 4: A summary of numerical examples (Examples 1, 2, 3) of cases of 15, 20, and 25-bit primes p

Bit-size of p	Running time (seconds): Step A	Step B	Total time	Bit-size of $\mathrm{Nrd}(J)$	Maximum prime factor of $\mathrm{Nrd}(J)$
15	30	15	45	67	53
20	266	181	447	119	103
25	162	230	392	162	131

5 CONCLUSION AND FUTURE WORK

The constructive Deuring correspondence is a central problem in computational number theory, and it is also closely connected to the security of some isogeny-based cryptosystems (see [14]). When we fix a supersingular elliptic curve E_0 over $\mathbb{F}_{p^2}$ with $\mathcal{O}_0 = \mathrm{End}(E_0)$, it is equivalent to the problem that computes the j-invariant of the supersingular elliptic curve E_I corresponding to a given left $\mathcal{O}_0$-ideal I under the Deuring correspondence. We demonstrated by experiments that we can solve the equivalent problem via the KLPT algorithm [26] *in practice* for primes p of up to around 25-bits. Specifically, we used the modified KLPT algorithm in [23] to output an equivalent ideal J of I with smaller reduced norm $\mathrm{Nrd}(J)$. Compared to the implementation report of [34], our key ingredient was to make use of symbolic formulas of isogenies in [32]. (Such formulas are available like modular polynomials.) For every prime ℓ dividing $\mathrm{Nrd}(J)$, such formulas allow us to recover a factor of the ℓ-th division polynomial $\psi_\ell(x)$ to efficiently obtain an ℓ-torsion point in E_0. However, our method has a limit since symbolic formulas for the elliptic curve (10) are available only for odd primes up to $\ell = 131$ in [32].

As future work, there are two research directions for larger characteristics p. A direction is to compute symbolic formulas for larger primes ℓ like [32]. Another direction is to improve the output quality of the KLPT algorithm. In particular, for the latter direction, we might be able to make use of the generalized KLPT algorithm in [12] to find a small and smooth reduced norm $\mathrm{Nrd}(J)$ in large characteristics p.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments to improve this work. In particular, we thank the reviewers telling us the methods mentioned in Subsection 3.4.2. The fifth author was supported by JST, ACT-X Grant Number JPMJAX2001, Japan. This work was also supported by JSPS KAKENHI Grant Numbers 19K22847 and 20K14301, Japan.

REFERENCES

[1] L. Babai, On Lovász' lattice reduction and the nearest lattice point problem, Combinatorica 6(1) (1986), 1–13.

[2] I.F. Blake, G. Seroussi and N.P. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, 1999.

[3] A. Bostan, F. Morain, B. Salvy and É. Schost, *Fast algorithms for computing isogenies between elliptic curves*, Mathematics of Computation 77 (2008), no. 263, 1755–1778.

[4] W. Castryck, T. Lange, C. Martindale, L. Panny and J. Renes, *CSIDH: An efficient post-quantum commutative group action*, in: Advances in Cryptology–ASIACRYPT 2018, Springer LNCS 11274 (2018), 395–427.

[5] J.M. Cerviño, *On the correspondence between supersingular elliptic curves and maximal quaternionic orders*, arXiv preprint math/math/0404538 (2004), available at `https://arxiv.org/abs/math/0404538`.

[6] I. Chevyrev and S.D. Galbraith, *Constructing supersingular elliptic curves with a given endomorphism ring*, LMS Journal of Computation and Mathematics 17 (2014), no. A, 71–91.

[7] D. Charles, K. Lauter and E. Goren, *Cryptographic hash functions from expander graphs*, Journal of Cryptology 22 (2009), no. 1, 93–113.

[8] L. Colò and D. Kohel, *Orienting supersingular isogeny graphs*, Journal of Mathematical Cryptology 14 (2020), no. 1, 414–437.

[9] G. Cornacchia, *Su di un metodo per la risoluzione in numeri interi dell' equazione* $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$, Giornale di Mathematiche di Battaglini 46 (1903), 33–90.

[10] M. Deuring, *Die Typen der Multiplikatorenringe elliptischer Funktionenkörper*, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 14 (1941), 197–272.

[11] L. De Feo and S.D. Galbraith, *SeaSign: Compact Isogeny Signatures from Class Group Actions*, in: Advances in Cryptology–EUROCRYPT 2019, Springer LNCS 11478 (2019), 759–789.

[12] L. De Feo, D. Kohel, A. Leroux, C. Petit and B. Wesolowski, *SQISign: Compact post-quantum signatures from quaternions and isogenies*, in: Advances in Cryptology–ASIACRYPT 2020, Springer LNCS 12491 (2020), 64–93.

[13] N. D. Elkies, *Elliptic and modular curves over finite fields and related computational issues*, AMS IP STUDIES IN ADVANCED MATHEMATICS 7 (1998), 21–76.

[14] K. Eisenträger, S. Hallgren, K. Lauter, T. Morrison and C. Petit, *Supersingular isogeny graphs and endomorphism rings: Reductions and solutions*, in: Advances in Cryptology–EUROCRYPT 2018, Springer LNCS 10822 (2018), 329–368.

[15] K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison and J. Park, *Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs*, in: Algorithmic Number Theory Symposium (ANTS XIV), Mathematical Sciences Publishers, Open Book Series 4 (2020), 215–232.

[16] S. D. Galbraith, *Mathematics of public key cryptography*, Cambridge University Press, 2012.

[17] S. D. Galbraith, C. Petit, B. Shani and Y. B. Ti, *On the security of supersingular isogeny crytosystems*, in: Advances in Cryptology–ASIACRYPT 2016, Springer LNCS 10031 (2016), 63–91.

[18] S. D. Galbraith, C. Petit and J. Silva, *Identification protocols and signature schemes based on supersingular isogeny problems*, in: Advances in Cryptology–ASIACRYPT 2017, Springer LNCS 10624 (2017), 3–33.

[19] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, 1999.

[20] C.D. de Saint Guilhem, P. Kutas, C. Petit and J. Silva, *SÉTA: Supersingular Encryption from Torsion Attacks*, IACR Cryptology ePrint Archive: Report 2019/1291 (2019), available at `https://eprint.iacr.org/2019/1291`.

[21] D. Jao and L. De Feo, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, in: International Workshop on Post-Quantum Cryptography (PQCrypto 2011), Springer LNCS 7071 (2011), 19–34.

[22] D. Jao et al., *SIKE: Supersingular Isogeny Key Encapsulation*, Submission to the NIST standardization process on post-quantum cryptography, available at `https://sike.org/`.

[23] Y. Kambe, Y. Aikawa, M. Kudo, M. Yasuda, K. Takashima and K. Yokoyama, *Implementation report of the Kohel-Lauter-Petit-Tignol algorithm for the constructive Deuring correspondence*, in: International Conference on Mathematics and Computing (ICMC 2021), Springer AISC 1412 (2022), 953–966.

[24] K. S. Kedlaya and C. Umans, *Fast polynomial factorization and modular composition*, SIAM Journal on Computing 40, no. 6 (2011), 1767–1802.

[25] D. Kohel, *Endomorphism rings of elliptic curves over finite fields*, PhD thesis, University of California, Berkeley (1996).

[26] D. Kohel, K. Lauter, C. Petit and J.-P. Tignol, *On the quaternion ℓ-isogeny path problem*, LMS Journal of Computation and Mathematics 17A (2014), 418–432.

[27] K. McMurdy and K. Lauter, *Explicit generators for endomorphism rings of supersingular elliptic curves*, preprint (2004), available at `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.601.43&rep=rep1&type=pdf`.

[28] J. Miret, R. Moreno, A. Rio and M. Valls, *Computing the ℓ-power torsion of an elliptic curve over a finite field*, Mathematics of computation 78, no. 267 (2009), 1767–1786.

[29] D. Moody et al., *NISTIR 8309: Status report on the second round of the NIST Post-Quantum Cryptography standardization process*, available at `https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf`.

[30] T. Moriya, H. Onuki and T. Takagi, *SiGamal: A supersingular isogeny-based PKE and its application to a PRF*, in: Advances in Cryptology–ASIACRYPT 2020, Springer LNCS 12492 (2020), 551–580.

[31] The National Institute of Standards and Technology (NIST), *Post-Quantum Cryptography*, available at `https://csrc.nist.gov/projects/post-quantum-cryptography`.

[32] M. Noro, M. Yasuda and K. Yokoyama, *Symbolic computation of isogenies of elliptic curves by Vélu's formula*, Commentarii Mathematici Universistatis Sancti Pauli 68 (2020), 93–130, available at `https://rikkyo.repo.nii.ac.jp/?action=repository_uri&item_id=20429`.

[33] C. Petit and S. Smith, *An improvement to the quaternion analogue of the ℓ-isogeny problem* (slides), presented at MathCrypt 2018.

[34] D. Ray, *Constructing the Deuring correspondence with applications to supersingular isogeny-based cryptography*, Master Thesis, Eindhoven University of Technology (2018).

[35] M. Noro and T. Takeshima, *Risa/Asir–A computer algebra system*, in: International Symposium on Symbolic and Algebraic Computation (ISSAC 1992), ACM (1992), 387–396.

[36] F. Rouillier, *Solving zero-dimensional systems through the rational univariate representation*, Applicable Algebra in Engineering, Communication and Computing 9 (1999), no. 5, 433–461.

[37] The Sage Developers, *SageMath, the Sage Mathematics Software System* (Version 9.0), available at `https://www.sagemath.org/`.

[38] R. Schoof, *Counting points on elliptic curves over finite fields*, Journal de théorie des nombres de Bordeaux 7 (1995), no. 1, 219–254.

[39] J. H. Silverman, *The arithmetic of elliptic curves*, Springer GTM 106, Second Edition, 2009.

[40] J. Vélu, *Isogénies entre courbes elliptiques*, CR Acad. Sci. Paris, Séries A 273 (1971), 238–241.

[41] J. Voight, Quaternion algebras (v.0.9.21), available at `https://math.dartmouth.edu/~jvoight/quat-book.pdf`, (2020).