Hard and Easy Problems for Supersingular Isogeny Graphs

Christophe Petit and Kristin Lauter
University of Birmingham, Microsoft Research

September 29, 2017

Abstract

We consider the endomorphism ring computation problem for supersingular elliptic curves, constructive versions of Deuring's correspondence, and the security of Charles-Goren-Lauter's cryptographic hash function. We show that constructing Deuring's correspondence is easy in one direction and equivalent to the endomorphism ring computation problem in the other direction. We also provide a collision attack for special but natural parameters of the hash function, and we prove that for general parameters its preimage and collision resistance are also equivalent to the endomorphism ring computation problem. Our reduction and attack techniques are of independent interest and may find further applications in both cryptanalysis and the design of new protocols.

1 Introduction

The recent search for new "post-quantum" cryptographic primitives and the ongoing international PQC competition sponsored by NIST has motivated a new era of research in the mathematics of cryptography. Ideas for cryptographic primitives based on hard mathematical problems are being actively proposed and examined. This paper focuses on isogeny-based cryptography, and in particular on the hardness of computing endomorphism rings of supersingular elliptic curves and its possible applications in cryptography.

In 2006, Charles, Goren, and Lauter introduced the hardness of finding paths in Supersingular Isogeny Graphs into cryptography and used it for constructing cryptographic hash functions. Since then, this problem and related hard problems have been used as the basis for key exchange protocols [16], signature schemes [29, 13], and public key encryption [10].

There exists a one to one correspondence due to Deuring [9] between supersingular j-invariants and maximal orders in a quaternion algebra, up to some equivalence relations. Following this correspondence, candidate hard problems underlying the security of Charles-Goren-Lauter hash function can be naturally

translated into problems expressed in terms of elements and ideals of a quaternion algebra. An a priori plausible strategy to cryptanalyse the hash function would therefore involve the following three steps:

1. Translate an isogeny problem into its quaternion algebra equivalent.
2. Solve the problem in the quaternion algebra.
3. Translate the solution back into a solution for the isogeny problem.

This motivates explicit versions of Deuring's correspondence for completing the first and last step of this strategy, namely algorithms to translate j-invariants into maximal orders in the quaternion algebra and conversely.

In this paper, we report our successes and failures in implementing this strategy. We solved the second step with coauthors in [18], and the present paper contains additional techniques that essentially allow to solve the third step. On the other hand, we found that the first step and in fact the security of the hash function are equivalent to computing the endomorphism ring of a supersingular elliptic curve, a problem that is emerging as the core candidate hard problem in isogeny-based cryptography.

1.1 Contributions of this paper

More precisely, in this paper we consider the following five problems (we refer to subsequent sections for precise descriptions of these problems):

1. Constructing Deuring's correspondence from maximal orders to supersingular invariants.
2. Constructing Deuring's correspondence from supersingular invariants to maximal orders.
3. Computing endomorphism rings of supersingular elliptic curves.
4. Computing preimages and collisions for the hash function when the initial vertex is chosen at random.
5. Computing preimages and collisions for the hash function when the initial vertex is chosen by the attacker.

We provide efficient algorithms for the first and last of these problems, and efficient reductions between the other three problems.

1.2 Related work

The endomorphism ring computation problem and constructive versions of Deuring's corespondence have been studied in the past independently of their cryptographic applications [17, 5], and all known algorithms for these problems have required exponential time. Here we provide a polynomial time algorithm for computing Deuring's correspondence in one direction.

Computing Deuring's correspondence in the other direction has sometimes been identified with the endomorphism ring computation problem, since in a sense both of them output some description of the endomorphism ring. The output formats required for both problems are however very different and it is a priori not easy to go from one to another. Our reductions accomplish that by using the quaternion ℓ-isogeny algorithm from [18] and additional techniques.

To the best of our knowledge there has not been any progress on the security of Charles-Goren-Lauter hash function since the initial arguments and attempts presented in their paper [6]. While they based preimage and collision resistance on some isogeny degrees of a special form, we show that these properties in fact only rely on the hardness of computing endomorphism rings of supersingular elliptic curves. Paradoxically we also give a new partial attack on the construction for specific but natural parameters.

Relationships between various isogeny problems were discussed in the preliminary sections of [12, 13], based on an earlier version of our paper. Here we make some of the results that were mentioned there explicit.

Recently there have been several partial attacks on isogeny-based protocols [12, 25, 14, 21]. These attacks target the key exchange protocol of Jao-De Feo [16] in specific attack models and are complementary to our work.

1.3 Outline

We recall various preliminaries in Section 2. In Section 3 we give our reductions between computing endomorphism rings of supersingular elliptic curves, constructing Deuring's correspondence from j-invariants to maximal orders, and the security of Charles-Goren-Lauter hash function. In Section 4 we provide an algorithm to construct Deuring's correspondence from maximal orders to j-invariants and a partial attack on the hash function. Section 5 concludes the paper.

2 Preliminaries

2.1 Supersingular isogeny graphs

Let p be a prime, and let E be a supersingular elliptic curve defined over a field of characteristic p. Up to isomorphism the curve can in fact be defined over $\mathbb{F}_{p^2}$. An isogeny $\phi : E \to E'$ is a nonconstant morphism from E to E' that maps the identity into the identity. The degree of an isogeny φ is the degree of φ as a morphism. An isogeny of degree ℓ is called an ℓ-isogeny. An isogeny can be identified with its kernel [28]. Given a subgroup G of E, we can use Vélu's formulae [27] to compute an isogeny $\varphi : E \to E'$ with kernel G and such that $E' \cong E/G$. Given a prime ℓ, the torsion group $E[\ell]$ contains exactly $\ell+1$ cyclic subgroups of order ℓ, each one corresponding to a different isogeny of degree ℓ. For each isogeny $\varphi : E \to E'$, there is a unique isogeny $\hat{\varphi} : E' \to E$, which is called the *dual isogeny* of φ, satisfying $\varphi\hat{\varphi} = \hat{\varphi}\varphi = [\deg \varphi]$.

If we have two isogenies $\varphi : E \to E'$ and $\varphi' : E' \to E$ such that $\varphi\varphi'$ and $\varphi'\varphi$ are the identity in their respective curves, we say that φ, φ' are *isomorphisms*, and that E, E' are *isomorphic*. Isomorphism classes of elliptic curves over $\mathbb{F}_q$ can be labeled with their j-invariant [23, III.1.4(b)]. In this paper we write $j(E)$ for the j-invariant of E. By convention given a j-invariant $j \neq 0, 1728$ we write $E(j)$ for the curve defined by the equation $y^2 = x^3 + \frac{3j}{1728-j}x + \frac{2j}{1728-j}$. We also write $E(0)$ and $E(1728)$ for the curves with equations $y^2 = x^3 + 1$ and $y^2 = x^3 + x$ respectively.

For any prime $\ell \neq p$, one can construct a so-called *ℓ-isogeny graph*, where each vertex is associated to a supersingular j-invariant, and an edge between two vertices is associated to a degree ℓ isogeny between the corresponding curves. Isogeny graphs are regular with regularity degree $\ell + 1$; they are undirected since to any isogeny from j_1 to j_2 corresponds a dual isogeny from j_2 to j_1. Isogeny graphs are Ramanujan, i.e. they are optimal *expander graphs*, with the consequence that random walks on the graph quickly reach the uniform distribution [15].

2.2 Charles, Goren and Lauter hash function

The first cryptographic construction based on supersingular isogeny problems is a hash function proposed by Charles, Goren and Lauter [6]. The security of this construction relies on the hardness of computing some isogenies of special degrees between two supersingular elliptic curves.

More precisely, consider an ℓ-isogeny graph over $\mathbb{F}_{p^2}$, where p is a "large" prime and ℓ is a "small" prime. The authors suggest to take $p = 1 \bmod 12$ to avoid some annoying backtracking issues. The message is first mapped into $\{0, \ldots, \ell - 1\}^*$, with some padding if necessary. At each vertex, a deterministic ordering of the edges is fixed (this can be done by sorting the j-invariants of the $\ell+1$ neighbours). An initial vertex j_0 is also fixed, as well as an initial incoming direction.

Given a message $(m_1, m_2, \ldots, m_N) \in \{0, \ldots, \ell - 1\}^*$, an edge of j_0 (excluding the incoming edge) is first chosen according to the value of m_1, and the corresponding neighbour E_1 is computed. Then an edge of j_1 (excluding the edge between j_0 and j_1) is chosen according to the value of m_2, and the corresponding neighbour j_2 is computed, etc. The final invariant j_N reached by this computation is mapped to $\{0, 1\}^n$ in some deterministic way (here $n \approx \log p$) and the value obtained is returned as the output of the hash function.

Clearly the function is preimage resistant if and only if, given two supersingular invariants j_1 and j_2, it is computationally hard to compute a positive integer e and an isogeny $\varphi : E(j_1) \to E(j_2)$ of degree ℓ^e. Moreover it is collision resistant if and only if, given one supersingular invariant j, it is computationally hard to compute a positive integer e and an endomorphism $\varphi : E(j) \to E(j)$ of degree ℓ^e.

In this paper we give two new results on the security of this construction. On the one hand (Section 3.3), we show that for a randomly chosen starting point j_0 the function is preimage and collision resistant if and only if the endomorphism

ring computation problem is hard: loosely speaking this means computing some endomorphisms of $E(j)$ but not necessarily of the correct norms. The interest of this result lies in that computing endomorphisms of elliptic curves is a natural problem to consider from an algorithmic number theory point of view, and it has indeed been studied since Kohel's thesis in 1996. On the other hand (Section 4.2), we also show that the collision resistance problem is easy for some particular starting points.

2.3 Deuring correspondence

The endomorphism ring of a supersingular elliptic curve is isomorphic to a maximal order in the quaternion algebra $B_{p,\infty}$ over $\mathbb{Q}$ ramified only at p and ∞. A quaternion algebra is generated as a $\mathbb{Q}$-module by four elements $\{1, i, j, k\}$ where $i^2 = a$, $j^2 = b$, $ij = -ji$ and $k = ij$ for some integers a, b, and is often denoted by (a, b). We refer to Vignéras [26] for the arithmetic of quaternion algebras and the definitions and properties of the trace, reduced norm, orders and ideals.

Pizer [22] gave the following explicit description of $B_{p,\infty}$ for all p along with a basis for one maximal order.

Proposition 1 *[22, p 368–369] Let $p > 2$ be a prime. Then we can define $B_{p,\infty}$ and a maximal order $\mathcal{O}_0$ as follows:*

p	(a, b)	$\mathcal{O}_0$
3 mod 4	$(-p, -1)$	$\langle 1, j, \frac{j+k}{2}, \frac{1+i}{2} \rangle$
5 mod 8	$(-p, -2)$	$\langle 1, j, \frac{2-j+k}{4}, \frac{-1+i+j}{2} \rangle$
1 mod 8	$(-p, -q)$	$\langle \frac{1+j}{2}, \frac{i+k}{2}, \frac{j+ck}{q}, k \rangle$

where in the last row $q = 3 \bmod 4$, $(p/q) = -1$ and c is some integer with $q | c^2 p + 1$. Assuming that the generalized Riemann hypothesis is true, there exists $q = O(\log^2 p)$ satisfying these conditions [2].

We represent quaternion algebra elements as linear combinations of $1, i, j, k$, where moreover q is minimal in the case $p = 1 \bmod 8$. We stress that in all cases the maximal orders $\mathcal{O}_0$ given by Proposition 1 contain $\langle 1, i, j, k \rangle$ as a small index subring.

Deuring [9] showed that supersingular elliptic curves over $\overline{\mathbb{F}}_p$ (up to isomorphism) are in one-to-one correspondence with maximal orders of $B_{p,\infty}$ (up to conjugation by an invertible element of $B_{p,\infty}$). More precisely, Deuring's correspondence associates to a supersingular invariant j any maximal order $\mathcal{O}$ such that $\mathcal{O} \cong \mathrm{End}(E)$. Moreover any left ideal I of $\mathcal{O}$ corresponds to an isogeny $\phi_I : E \to E_I$ with kernel

$$\ker \phi_I = \{P \in E | \alpha(P) = 0, \forall \alpha \in I\}.$$

This is a 1-1 correspondence provided that the degree of ϕ_I is coprime to p. In addition, we can identify the right order of I, $\mathcal{O}_{\mathcal{R}}(\mathcal{I})$ with the endomorphism ring of E_I.

When $p = 3 \bmod 4$ the curve $y^2 = x^3 + x$ is supersingular with invariant $j = 1728$. This curve corresponds to a maximal order $\mathcal{O}_0$ with $\mathbb{Z}$-basis $\{1, i, \frac{1+k}{2}, \frac{i+j}{2}\}$ under Deuring's correspondence, and there is an isomorphism of quaternion algebras $\theta : B_{p,\infty} \to \mathrm{End}(E_0) \otimes \mathbb{Q}$ sending $(1, i, j, k)$ to $(1, \phi, \pi, \pi\phi)$ where $\pi : (x, y) \to (x^p, y^p)$ is the Frobenius endomorphism, and $\phi : (x, y) \to (-x, \iota y)$ with $\iota^2 = -1$. More generally, it is easy to compute j-invariants corresponding to the maximal orders given by Proposition 1.

Proposition 2 *There is a polynomial time algorithm that given a prime $p > 2$, computes a supersingular invariant $j_0 \in \mathbb{F}_p$ such that $End(E(j_0)) \cong \mathcal{O}_0$ (where $\mathcal{O}_0$ is as given by Proposition 1 together with a map $\phi \in End(E(j_0))$) such that $\theta : B_{p,\infty} \to End(E(j_0)) \otimes \mathbb{Q} : (1, i, j, k) \to (1, \phi, \pi, \pi\phi)$ is an isomorphism of quaternion algebras.*

Proof: Consider Algorithm 1 below. Step 1 can be executed in time polynomial in $\log p$ using complex multiplication, as in Bröker's algorithm [4]. The cardinality of $\mathcal{J}$ is equal to the class number of $\mathbb{Q}(\sqrt{-q})$, and this is bounded by q. To compute ϕ in Step 3 one can simply compute all isogenies of degree q using Vélu's formulae and identify the one corresponding to an endomorphism. The map ϕ defines an isomorphism of quaternion algebras $\theta : B_{p,\infty} \to \mathrm{End}(E(j_0)) \otimes \mathbb{Q} : (1, i, j, k) \to (1, \phi, \pi, \pi\phi)$. To perform the check in Step 4, one applies θ to the numerators of $\mathcal{O}_0$ basis elements, and check whether the resulting maps annihilate the D torsion, where D is the denominator. □

Algorithm 1 Computing Deuring correspondence for special orders

Require: A prime p.

Ensure: A supersingular invariant $j_0 \in \mathbb{F}_p$ such that $\mathcal{O}_0 \cong \mathrm{End}(E(j_0))$, and an endomorphism $\phi \in \mathrm{End}(E(j_0))$ such that $\mathrm{n}(\phi) = q$ and $\mathrm{Tr}(\phi) = 0$.

1: Compute $\mathcal{J}$ a set of supersingular invariants j such that $E(j)$ has complex multiplication by R_D, the integer ring of $\mathbb{Q}(\sqrt{-q})$.
2: **for** $j \in \mathcal{J}$ **do**
3: Compute ϕ an endomorphism of degree q of $E(j)$.
4: **if** $\mathrm{End}(E(j)) \cong \mathcal{O}_0$ **then**
5: **return** j and ϕ.
6: **end if**
7: **end for**

In this paper we will be interested in *constructing* Deuring's correspondence for arbitrary maximal orders and supersingular j invariants. This could a priori have three different meanings, given by Problems 1, 2 and 3 below.

Problem 1 (Deuring Correspondence List.) *Build a list of all pairs $(j, \mathcal{O})$ where j is a supersingular invariant and the endomorphism ring of $E(j)$ is isomorphic to $\mathcal{O}$.*

This problem was considered by Cerviño in [5]. His algorithm computes representation numbers for *all* supersingular elliptic curves and *all* maximal orders

of $B_{p,\infty}$, and then compares the two lists of representation numbers to realize Deuring's correspondence. Since representation numbers of size up to $O(p^{1/2})$ may be needed to distinguish any pair of orders, the algorithm runs in time at least $O(p^2)$ times a polynomial function of $\log p$. A similar approach was followed in [20].

As there are roughly $p/12$ supersingular invariants and they require $O(\log p)$ bits to represent, any algorithm for Problem 1 will at best run in a time $O(p \log p)$. We can hope for more efficient algorithms if we are only interested in constructing the correspondence for a given order or curve.

Problem 2 (Constructive Deuring Correspondence.) *Given a maximal order $\mathcal{O} \subset B_{p,\infty}$, return a supersingular j invariant such that the endomorphism ring of $E(j)$ is isomorphic to $\mathcal{O}$.*

Problem 3 (Inverse Deuring Correspondence.) *Given a supersingular invariant j, compute a maximal order $\mathcal{O} \in B_{p,\infty}$ such that the endomorphism ring of $E(j)$ is isomorphic to $\mathcal{O}$.*

The j-invariant is naturally represented as an element of $\mathbb{F}_{p^2}$, and it is unique up to Galois conjugation. The maximal order is unique up to conjugation by an invertible quaternion element, and it can be described by a $\mathbb{Z}$-basis, namely four elements $1, \omega_2, \omega_3, \omega_4 \in B_{p,\infty}$ such that $\mathcal{O} = \mathbb{Z} + \omega_2\mathbb{Z} + \omega_3\mathbb{Z} + \omega_4\mathbb{Z}$. Choosing a Hermite basis makes this description unique.

In this paper we will provide a polynomial time algorithm for Problem 2 (Section 4.1). We will also explicit connections between Problem 2 and the endomorphism ring computation problem, where instead of a maximal order in $B_{p,\infty}$ one needs to output a basis for $\mathrm{End}(E(j))$.

2.4 The endomorphism ring computation problem

Given an elliptic curve, it is natural to ask to compute its endomorphism ring.

Problem 4 (Endomorphism ring computation problem.) *Given a supersingular invariant j, compute the endomorphism ring of $E(j)$.*

The endomorphism ring can be returned as four rational maps that form a $\mathbb{Z}$-basis with respect to scalar multiplication (in fact 3 maps, since one of these maps can always be chosen equal to the identity map). The maps themselves can usually not be returned in their canonical expression as rational maps, as in general this representation will require a space larger than the degree, and the degrees can be as big as p.

Various representations of the maps are a priori possible. We believe that any valid representation should be *concise* and *useful*, in the sense that it must require a space polynomial in $\log p$ to store, and it must allow the evaluation of the maps at arbitrary elliptic curve points in a time polynomial in both $\log p$ and the space required to store those points. To the best of our knowledge these two conditions are sufficient for all potential applications of Problem 4. When

its degree is a smooth number, an endomorphism can be efficiently represented as a composition of small degree isogenies. In Section 3.1 we will consider a more general representation.

A first approximation to a solution to Problem 4 was provided by Kohel in his PhD thesis [17], and later improved by Galbraith [11] using a birthday argument. The resulting algorithm explores a tree in an ℓ-isogeny graph (for some small integer ℓ) until a collision is found, corresponding to an endomorphism. The expected cost of this procedure is $O(\sqrt{p})$ times a polynomial in $\log p$. Repeating this procedure a few times, possibly with different values of ℓ, we obtain a set of endomorphisms which generate a subring of the whole endomorphism ring, and heuristically one expects that they actually generate the whole endomorphism ring. The endomorphism ring computation problem was also considered in [8] for curves defined over $\mathbb{F}_p$. The identification protocol and signature schemes developed in [13] explicitly rely on its potential hardness for security. We remark that significant progress has been made since Kohel's thesis on the endomorphism ring computation problem in the ordinary case [3]. However these improvements use the commutative nature of the endomorphism ring of ordinary curves, and it is not clear how they could be adapted to supersingular curves.

We observe that Problems 3 and 4 take the same input, and their outputs are also "equal" in the sense they are isomorphic. For this reason the two problems have sometimes been referred to interchangeably. We stress, however, that being isomorphic does not a priori guarantee that the isomorphism is efficiently computable, the same way as discrete logarithms can be computed in the additive group $\mathbb{Z}_{p-1}$ but not in the multiplicative group $\mathbb{F}_p^*$. In particular, a solution to Problem 3 does not a priori provide a *uesful* description of the endomorphism ring so that one can for example evaluate endomorphisms at given points. Similarly, a solution to Problem 3 does not a priori provide a $\mathbb{Z}$-basis for an order in $B_{p,\infty}$, and this is necessary for example to apply the algorithms of [18].

It turns out that the two problems are equivalent: in Sections 3.1 and 3.2 below, we provide efficient algorithms to go from a representation of the endomorphism ring as a $\mathbb{Z}$ basis over $\mathbb{Q}$ to a representation as rational maps and conversely.

2.5 Quaternion ℓ-isogeny algorithm

The quaternion ℓ-isogeny problem was introduced and solved in [18] as a step forward in the cryptanalysis of Charles-Goren-Lauter hash function, following the general strategy outlined in Section 1.

We refer to [18, 13] for a full description of the algorithm and its powersmooth version as well as their analysis. For our purposes the following proposition will be sufficient.

Lemma 3 *[18, 13] Under various heuristic assumptions, there exist two polynomial time algorithms that given I a left ideal of $\mathcal{O}_0$, returns J another left*

ideal of $\mathcal{O}_0$ in the same class as I, with a norm N such that $N \approx p^{7/2}$. Moreover for the first algorithm we have $N = p_i^{e_i}$ with $p_i^{e_i} < \log p$ and for the second algorithm we have $N = \ell^e$ for some integer e and some small prime ℓ.

Interestingly, [13] also proves that (after a minor tweak) the outputs of these algorithms only depend on the ideal class of their inputs and not on the particular ideal class representative.

Many of our algorithms and reductions below will use these algorithms as black boxes. Their correctness will therefore rely on the same heuristics, and possibly some more.

2.6 Translating $\mathcal{O}_0$ ideals to isogenies

Let $\mathcal{O}_0$ be a maximal order given by Proposition 1, let E_0 be a corresponding supersingular elliptic curve, and let I be a left $\mathcal{O}_0$ ideal of norm N. Following Deuring's correspondence this ideal corresponds to an isogeny $\phi : E_0 \to E_1$ of degree N. This isogeny is uniquely defined by its kernel, which is a cyclic subgroup of order N in E_0. Following Waterhouse [28] one can identify the correct subgroup by evaluating the maps corresponding to an $\mathcal{O}_0$ basis at a generator of each subgroup. Moreover when N is composite, the kernel can be represented more efficiently as a product of cyclic subgroups whose orders are powers of primes, and similarly the isogenies are represented more efficiently as a composition of prime degree isogenies. The details of such an algorithm can be found in [13], which also analyzes its complexity. The following proposition will be sufficient for our purposes.

Proposition 4 *There exists an algorithm which, given an $\mathcal{O}_0$ left ideal I of norm $N = \prod_i p_i^{e_i}$, returns an isogeny $\phi : E_0 \to E_1$ corresponding to this ideal through Deuring's correspondence. Moreover the runtime complexity of this algorithm is polynomial in $\max_i p_i^{e_i}$.*

We stress that this translation algorithm requires to know the endomorphism ring of E_0, and that it is only efficient when $\max_i p_i^{e_i}$ is small.

2.7 Isogeny-based cryptography

A few years after Charles, Goren and Lauter designed their hash function, Jao and De Feo proposed a variant of the Diffie-Hellman protocol based on supersingular isogeny problems, which is now known as the supersingular isogeny key exchange protocol [16]. We briefly describe it here in a way to encompass both the original parameters and the generalization recently suggested by Petit [21].

The parameters include a large prime p, a supersingular curve E, and two coprime integers N_A and N_B. Alice and Bob select cyclic subgroups of E of order respectively N_A and N_B; they compute the corresponding isogenies and they exchange the values of the end vertices respectively E/G_A or E/G_B. The shared key is the value $j(E/\langle G_A, G_B\rangle)$. This shared key could a priori not be computed by any party from E/G_A, E/G_B and their respective secret keys only,

so Alice (resp. Bob) additionally sends the images of a basis of $E[N_B]$ by ϕ_A (resp. a basis of $E[N_A]$ by ϕ_B).

Jao-De Feo suggested to use $N_A = 2^{e_B} \approx p^{1/2} \approx N_B = 3^{e_B}$ such that $(p-1)/N_A N_B$ is a small integer for efficiency reasons; in [21] Petit argued that choosing $N_A \approx N_B \approx p^2$ both powersmooth numbers is a priori better from a security point of view while preserving polynomial time complexity for the protocol execution. It was shown by Gabraith-Petit-Shani-Ti [13] that computing the endomorphism ring of E and E_A is sufficient to break the key exchange for the parameters suggested by Jao-De Feo. The argument uses the fact that isogenies generated for Jao-De Feo's parameters are of relatively small degree, and this does not seem to apply to Petit's paramters.

The security of Jao-De Feo's protocol relies on the hardness of computing isogenies of a given degree between two given curves, when provided in addition with the action of the isogeny on a large torsion group. This problem is not known to be equivalent to the endomorphism ring computation problem. Recent results by Petit [21] show that revealing the action of isogenies on a torsion group does make some isogeny problems easier to solve, though at the moment his techniques do not apply to Jao-De Feo's original parameters. In this paper we show the equivalence of the endomorphism ring computation problem with some relevant problems, and we solve some other problems. We *believe* that the security of the key exchange protocol lies between these hard and easy problems, but leave its study to further work.

The interest in isogeny-based cryptography has recently increased in the context of NIST's call for post-quantum cryptography algorithms [1]: indeed at the moment the best algorithms to solve supersingular isogeny problems all require exponential time in the security parameter, even when including quantum algorithms. Besides the hash function and the key exchange protocols, there are now constructions based on isogeny problems for public key encryption, identification protocols and signatures [10, 29, 13]. Constructions in the first two papers build on the key exchange protocol and rely on similar assumptions. The second signature scheme in [13], however, only relies on the endomorphism computation problem.

3 Equivalent Hard Problems in Supersingular Isogeny Graphs

In this section we consider the following problems:

- A constructive version of Deuring's correspondence, from j invariants to maximal orders in $B_{p,\infty}$ (Problem 3).
- The endomorphism ring computation problem (Problem 4).
- The preimage and collision resistance of Charles-Goren-Lauter hash function, for a randomly chosen initial vertex.

We show that all these problems are heuristically equivalent, in the sense that there exist efficient reductions from one problem to another under plausible heuristics assumptions.

The first two problems have the same inputs and in a sense their outputs are also equal, so it is perhaps no surprise to the reader that they are equivalent. However, the two problems differ in the way the output should be represented: as a maximal order in $B_{p,\infty}$ for Problem 3, and as four rational maps for Problem 4. Sections 3.1 and 3.2 below clarify the steps from one representation to the other.

It should also be clear intuitively that (heuristically at least) an algorithm to find preimages or collisions for the hash function can be used to compute endomorphism rings. The other implication is perhaps not as intuitive, and our solution crucially requires the tools developed in [18]. These reductions are discussed in Section 3.3 below.

3.1 Endomorphism Ring Computation is not harder than Inverse Deuring Correspondence

Let us first assume that we have an efficient algorithm for Problem 3, returning a $\mathbb{Z}$ basis for a maximal order as discussed above. Algorithm 2 below uses this algorithm to solve Problem 4.

Algorithm 2 Reduction from Problem 4 to Problem 3

Require: A supersingular invariant j.
Ensure: Four maps that generate $\mathrm{End}(E(j))$.

1: Use an algorithm for Problem 3 to obtain a maximal order $\mathcal{O} \approx \mathrm{End}(E(j))$.
2: Compute an ideal I connecting $\mathcal{O}_0$ and $\mathcal{O}$.
3: Compute an ideal J with powersmooth norm in the same class as I.
4: Translate the ideal J into an isogeny $\varphi : E_0 \to E$.
5: Let N be the norm of J.
6: Let $1, \phi_2, \phi_3, \phi_4$ generating $\mathrm{End}(E(j_0))$.
7: Let $1, \omega_2, \omega_3, \omega_4$ generating $\mathcal{O}$, and let $1, \omega_{2,0}, \omega_{3,0}, \omega_{4,0} \in \mathcal{O}_0$ corresponding to $1, \phi_2, \phi_3, \phi_4$.
8: Find integers c_{ij} such that $\omega_i = \frac{\sum_j c_{ij}\omega_{j,0}}{N}$.
9: **return** N, φ, c_{ij} implicitly representing the maps $\frac{\sum_{i=1}^4 c_{ij}\varphi^{-1}\phi_i\varphi}{N}$ for each i.

The maps returned by Algorithm 2 are of the form $\phi = \frac{\sum_{i=1}^4 c_{ij}\varphi^{-1}\phi_i\varphi}{N}$ where N is a smooth number, $c_{ij} \in \mathbb{Z}$, $\{\phi_i\}_{i=1,2,3,4}$ form a basis for the endomorphism ring of a special curve E_0, and $\varphi : E_0 \to E(j)$ is an isogeny of degree N, given as a composition of low degree isogenies. This is arguably not the most natural representation of endomorphisms, but it still allows to efficiently evaluate them at arbitrary points, as shown by Algorithm 3 and Lemma 5 below.

Lemma 5 *Let K be an extension of $\mathbb{F}_{p^2}$ where P lies. Assume that $\log N$ and $\max_i p_i^{e_i}$ are polynomial in $\log p$. Then Algorithm 3 can be implemented to run in time polynomial in $\log |K|$.*

Algorithm 3 Endomorphism evaluation

Require: A curve E, an isogeny $\varphi : E_0 \to E$ with powersmooth degree N, and integers a, b, c, d defining an endomorphism $\phi = \frac{\varphi^{-1}(a+b\phi_2+c\phi_3+d\phi_4)\varphi}{N} \in \mathrm{End}(E)$.
Require: A point $P \in E$.
Ensure: $\phi(P)$.
1: Let $N = \prod_i {p_i}^{e_i}$ and let $m_i = N/p_i^{e_i}$.
2: **for** all i **do**
3: Compute Q_i such that $p_i^{e_i} Q_i = P$.
4: Compute $S_i = \varphi^{-1}(a + b\phi_2 + c\phi_3 + d\phi_4)\varphi(Q_i)$
5: **end for**
6: Compute S such that $S_i = m_i S$.
7: **return** S.

PROOF: Let Q such that $Q_i = m_i Q$. We have $NQ = P$, hence $S = \phi(P)$ since ϕ is a homomorphism. This proves correctness of the algorithm. Although Q may lie on a very large extension of $\mathbb{F}_{p^2}$, each of the Q_i lies on a reasonably small extension, namely the extension degree is polynomial in $\log p$. Note that S lies in K, so Step 6 is efficient. Step 3 involves some univariate polynomial factorization, a task that is polynomial in both the degree of the polynomial and the logarithm of the field order. In Step 4 the isogeny φ and its inverse can be evaluated stepwise, and evaluating the map $a + b\phi_2 + c\phi_3 + d\phi_4$ at an arbitrary point involves 4 scalar multiplications, three additions and the evaluation of the maps $\phi_i \in \mathrm{End}(E(j_0))$ at certain points. □

Proposition 6 *Under plausible heuristic assumptions, the reduction from Problem 4 to Problem 3 provided by Algorithm 2 can be implemented to run in a time polynomial in* $\log p$.

PROOF: In Step 2, the ideal I can be computed as $I = N\mathcal{O}_0 + N\mathcal{O}_0\mathcal{O}$, where N is the index of $\mathcal{O}_0 \cap \mathcal{O}$ in either $\mathcal{O}_0$ or $\mathcal{O}$. This can be done in a time polynomial in $\log B$. By Lemma 3 the output of Step 3 is an ideal of norm $N = \prod p_i^{e_i}$ such that $S = \max_i p_i^{e_i} = O(\log p)$. The translation algorithm runs in a time polynomial in S, hence in $\log p$. The other steps also run in polynomial time. □

3.2 Inverse Deuring Correspondence is not harder than Endomorphism Ring Computation

Let us now assume that we have an efficient algorithm for Problem 4, returning four maps generating the endomorphism ring, in some format that allows efficient evaluation of the maps at arbitrary points. Algorithm 4 below uses this algorithm and then constructs a sequence of linear transformations that bring

$1, \alpha, \beta, \gamma$ to four orthogonal maps $1, \iota, \lambda, \iota\lambda$ corresponding to $1, i, j, k \in B_{p,\infty}$. Composing the inverses of these maps then gives a $\mathbb{Z}$-basis for $\mathcal{O}$.

Algorithm 4 Reduction from Problem 3 to Problem 4

Require: A supersingular invariant j.
Ensure: A maximal order $\mathcal{O} \subset B_{p,\infty}$ such that $\mathrm{End}(E(j)) \approx \mathcal{O}$.

1: Use an algorithm for Problem 3 to obtain four maps $1, \alpha, \beta, \gamma$ generating $\mathrm{End}(E(j))$, in a format that allows efficient evaluation at elliptic curve points.
2: Compute the Gram matrix associated to the sequence $(1, \alpha, \beta, \gamma)$.
3: Find a rational invertible linear transformation sending $(1, \alpha, \beta, \gamma)$ to some $(1, \alpha', \beta', \alpha'\beta')$, where $1, \alpha', \beta', \alpha'\beta'$ generate an orthogonal basis for $B_{p,\infty}$ over $\mathbb{Q}$.
4: **if** the numerators and denominators of $\mathrm{n}(\alpha)$ and $\mathrm{n}(\beta)$ are not easy to factor **then**
5: Apply a random invertible linear transformation to (α, β, γ).
6: Go to Step 3.
7: **end if**
8: Find $a, b, c \in \mathbb{Q}$ such that $\mathrm{n}(\iota) = q$, where $\iota = a\alpha' + b\beta' + c\alpha'\beta'$.
9: Find a rational invertible linear transformation sending $(1, \alpha', \beta', \alpha'\beta')$ to $(1, \iota, \delta, \iota\delta)$ for some $\delta \in B_{p,\infty}$ where $1, \iota, \delta, \iota\delta$ generate an orthogonal basis for $B_{p,\infty}$ over $\mathbb{Q}$.
10: **if** the numerator and denominator of $\mathrm{n}(\delta)$ is not easy to factor **then**
11: Apply a random invertible linear transformation to (α, β, γ).
12: Go to Step 3.
13: **end if**
14: Find $a, b \in \mathbb{Q}$ such that $\mathrm{n}(\delta)(a^2 + b^2 q) = p$. Let $\lambda = a\delta + b\iota\delta$.
15: Find a rational invertible linear transformation sending $(1, \iota, \delta, \iota\delta)$ to $(1, \iota, \lambda, \iota\lambda)$.
16: Invert and compose all linear transformations to express $1, \alpha, \beta, \gamma$ in the basis $(1, \iota, \lambda, \iota\lambda)$, and deduce a basis of $\mathcal{O}$ in $B_{p,\infty}$.
17: **return** the basis of $\mathcal{O}$.

Let B be a bound on the degrees of the maps α, β, γ returned in Step 1 of Algorithm 4. We analyze the complexity of the algorithm through the following lemmas and proposition.

Lemma 7 *There exists an algorithm for Step 2 that runs in time polynomial in* $\log p$ *and* $\log B$.

Proof: Given two endomorphisms α, β, one can compute their inner product $\langle \alpha, \beta \rangle = \alpha\bar{\beta} + \beta\bar{\alpha} \in \mathbb{Z}$ by evaluating it on an appropriate set of small prime order torsion points then applying the Chinese remainder theorem, following a strategy similar to Schoof's point counting algorithm (see [17, Theorem 81]). Applying this algorithm to every pair of maps from $(1, \alpha, \beta, \gamma)$ gives the result. □

Lemma 8 *There exists an algorithm for Steps 3 and 9 that runs in time polynomial in* $\log p$ *and* $\log B$*.*

PROOF: We focus on Step 3, and Step 9 is similar. Given the Gram matrix one can apply the Gram-Schmidt orthogonalization process to obtain a new basis $(1, \alpha', \beta', \gamma')$. It remains to show that $\alpha'\beta'$ is a scalar multiple of γ' so that we can normalize γ' to obtain the result. It suffices to show that $\alpha'\beta'$ is orthogonal to 1, α' and β'. Indeed we have $\langle \alpha'\beta', 1\rangle = \alpha'\beta' + \bar{\beta}'\bar{\alpha}' = \langle \alpha', \bar{\beta}'\rangle = -\langle \alpha', \bar{\beta}'\rangle = 0$; we have $\langle \alpha'\beta', \alpha'\rangle = \alpha'\beta'\bar{\alpha}' + \alpha'\bar{\beta}'\bar{\alpha}' = \mathrm{n}(\alpha')\mathrm{Tr}(\beta') = 0$; and similarly $\langle \alpha'\beta', \beta'\rangle = \alpha'\beta'\bar{\beta}' + \beta'\bar{\beta}'\bar{\alpha}' = \mathrm{n}(\beta')\mathrm{Tr}(\alpha') = 0$. □

Lemma 9 *Given the factorizations of the numerators and denominators of both* $n(\alpha')$ *and* $n(\beta')$*, there exists an algorithm for Step 8 that runs in time polynomial in* $\log p$ *and* $\log B$*.*

PROOF: Finding such $a, b, c \in \mathbb{Q}$ satisfying the condition amounts to finding $a', b', c', d \in \mathbb{Z}$ such that $a'^2\mathrm{n}(\alpha') + b'^2\mathrm{n}(\beta') + c'^2\mathrm{n}(\alpha')\mathrm{n}(\beta') = d^2q$. According to Denis Simon [24, Section 8] his algorithm solves this Diophantine equation in polynomial time. □

Lemma 10 *Given the factorizations of the numerator and denominator of* $n(\delta)$*, there exists an algorithm for Step 14 that runs in time polynomial in* $\log p$ *and* $\log B$*.*

PROOF: Note that $\langle \delta, \iota\delta\rangle$ is by construction the orthogonal space of $\langle 1, \iota\rangle$, and this space must contain an element of norm p, so the equation has a solution. Given factorizations for both the numerator and the denominator of δ one can use Cornacchia's algorithm [7] to solve Step 14. □

Proposition 11 *Under plausible heuristic assumptions, the reduction provided by Algorithm 4 can be implemented to run in polynomial time.*

PROOF: In Steps 4 and 10 the algorithm requires that some numbers are easy to factor. In Step 4 we may expect these numbers to behave like random numbers of the same sizes. In Step 10 p must divide numerator of $\mathrm{n}(\delta)$. We may expect that both the cofactor and the denominator factor like random numbers of the same size. One can require all those numbers to be large primes, or a product of large primes and small cofactors, two properties that will be satisfied with a probability inversely proportional to a polynomial function of $\log p$. Steps 5 and 11 randomize α, β, γ so that we expect the conditions to be satisfied after a number of steps that is polynomial in $\log p$. By the four lemmas before we then expect that the whole reduction runs in a time polynomial in $\log p$. □

The reduction provided by Algorithm 4 and its runtime analysis relies on several heuristics, namely the probability to obtain suitable norms in Steps 4 and 10 as discussed in the above proposition, and the runtime algorithm of Denis Simon's algorithm for Step 8.

3.3 Preimage and Collision Resistance of CGL Hash Function

In this section we show that hardness of the endomorphism ring computation problem is equivalent to the security of Charles-Goren-Lauter hash function.

Proposition 12 *Assume there exists an efficient algorithm for the endomorphism ring computation problem. Then there is an efficient algorithm to solve the preimage and collision problems for Charles-Goren-Lauter hash function.*

PROOF: By standard arguments on hash functions it is enough to focus on preimage resistance. Our reduction of this problem to the endomorphism ring computation problem is given in Algorithm 5. Besides two black box calls to an algorithm for the endomorphism ring computation problem, it uses other efficient algorithms described in this paper, including Algorithm 2 to translate a description of an endomorphism ring as rational maps into a description of a maximal order in $B_{p,\infty}$, both the ℓ power and the powersmooth versions of the quaternion isogeny algorithm, and the translation algorithm from ideals to isogenies. All these routines are efficient by the lemmas and propositions of this paper. □

Algorithm 5 Reduction from preimage resistance to endomorphism ring computation

Require: Two supersingular invariants $j_s, j_t \in \mathbb{F}_{p^2}$.
Ensure: A sequence of j invariants $j_s = j_0, j_1, \ldots, j_e = j_t$ such that for any i there exists an isogeny of degree ℓ from $E(j_i)$ to $E(j_{i+1})$.
1: Compute $\mathrm{End}(E(j_s))$ and $\mathrm{End}(E(j_t))$.
2: Compute $\mathcal{O}_s \approx \mathrm{End}(E(j_s))$ and $\mathcal{O}_t \approx \mathrm{End}(E(j_t))$ with Algorithm 2.
3: Compute ideals I_s and I_t connecting $\mathcal{O}_0$ respectively to $\mathcal{O}_s$ and $\mathcal{O}_t$.
4: Compute ideals J_s and J_t with norm ℓ^e for some e, in the same classes as I_s and I_t respectively.
5: **for** $J \in \{J_s, J_t\}$ and corresponding $E \in \{E(j_s), E(j_t)\}$ **do**
6: Compute a sequence of ideals $J_i = \mathcal{O}_0 q + \mathcal{O}_0 \ell^i$ for $i = 0, \ldots, e$
7: **for** all i **do**
8: Compute K_i with powersmooth norm in the same class as I_i.
9: Translate K_i into an isogeny $\varphi_i : E_0 \to E_i$.
10: **end for**
11: Deduce a sequence $(j_0, j(E_1), j(E_2), \ldots, j_e = j(E))$.
12: **end for**
13: **return** $(j(E_s), \ldots, j_0, \ldots, j(E_t))$ the concatenation of both paths.

The reverse direction may look easier a priori. By standard arguments on hash functions it is sufficient to prove the claim with respect to a collision algorithm. A collision for Charles-Goren-Lauter's hash function gives a non scalar endomorphism of the curve; four linearly independent endomorphisms

give a full rank subring of the endomorphism ring; and heuristically one expects that a few of such maps will be sufficient to generate the whole ring. To compute the endomorphism ring one would therefore calls the collision finding algorithms multiple times until the resulting maps generate the full endomorphism rings. This strategy, however, has a potential caveat: the collision algorithm might be such that it always returns the same endomorphism. In Algorithm 6 we get around this problem by performing a random walk from the input invariant j, calling the collision algorithm on the end vertex of the random walk, and concatenating paths to form endomorphisms of $E(j)$.

Proposition 13 *Assume there exists an efficient preimage or collision algorithm for Charles-Goren-Lauter's hash function. Then under plausible heuristic assumptions there is an efficient algorithm to solve the endomorphism ring computation problem.*

PROOF: The reduction algorithm for collision resistance is given by Algorithm 6 below. Note that in Step 7 the discriminant can be computed from the Gram matrix, which by Lemma 7 can be efficiently computed. Heuristically, one expects that the loop will be executed at most $O(\log p)$ times. Indeed let us assume that after adding some elements to the subring we have a subring of index N. Then we can heuristically expect any new randomly generated endomorphism to lie in this subring with a probability only $1/N$. Moreover when it does not lie in the subring, the element will decrease the index by a non trivial integer factor of N. □

Algorithm 6 Reduction from endomorphism ring computation to collision resistance

Require: A supersingular invariant $j \in \mathbb{F}_{p^2}$.
Ensure: The endomorphism ring of $E(j)$.
1: Let $\mathcal{R} = \langle 1 \rangle \subset \mathrm{End}(E(j))$.
2: **repeat**
3: Perform a random walk in the graph, leading to a new vertex j'.
4: Apply a collision finding algorithm on j', leading to an endomorphism of $E(j')$.
5: Deduce an endomorphism ϕ of $E(j)$ by concatenating paths.
6: Set $\mathcal{R} \leftarrow \langle \mathcal{R}, \phi \rangle$.
7: Compute the discriminant of $\mathcal{R}$.
8: **until** $\mathrm{disc}(\mathcal{R}) = 4p^2$.
9: **return** a $\mathbb{Z}$ basis for $\mathcal{R}$.

4 Some Easy Problems in Supersingular Isogeny Graphs

The previous section relied heavily on the quaternion ℓ-isogeny algorithm of Kohel-Lauter-Petit-Tignol to derive the computational equivalence of several problems. In this section, we provide two additional applications of this algorithm. First, we give an algorithm for constructing Deuring correspondence from maximal orders in $B_{p,\infty}$ to supersingular j-invariants. Second, we give a polynomial time collision algorithm against Charles-Goren-Lauter hash function when a special curve is chosen as the initial point.

4.1 Constructive Deuring's correspondence, from quaternion orders to j-invariants

In this section we provide an efficient algorithm to solve Problem 2. Algorithm 7 first computes an ideal connecting $\mathcal{O}_0$ to $\mathcal{O}$. Then it uses the quaternion ℓ-isogeny algorithm from ANTS 2014 (or rather, its powersmooth version) to compute another ideal in the same class but with a norm $N = \prod p_i^{e_i}$ such that $\max_i p_i^{e_i}$ is small. It finally translates that ideal into an isogeny $\phi : E_0 \to E_1$ that corresponds to it via Deuring's correspondence.

Algorithm 7 Constructive Deuring correspondence, from quaternions to maximal orders.

Require: Maximal order $\mathcal{O} \subset B_{p,\infty}$.
Ensure: Supersingular invariant j such that $\mathrm{End}(E(j)) \approx \mathcal{O}$.

1: Compute an ideal I that is a left ideal of $\mathcal{O}_0$ and a right ideal of $\mathcal{O}$.
2: Compute an ideal J in the same class as I but with powersmooth norm.
3: Compute an isogeny $\phi : E_0 \to E_I$ that corresponds to J via Deuring's correspondence.
4: **return** $j(E_I)$.

Let $\langle 1, \omega_2, \omega_2, \omega_3 \rangle$ be a basis for $\mathcal{O}$, let $M \in GL(4, \mathbb{Q})$ such that $(1, \omega_2, \omega_2, \omega_3) = M(1, i, j, k)$, and let B be a bound on the numerators and denominators of all coefficients of M.

Proposition 14 (Constructive Deuring Correspondence.) *Under plausible heuristic assumptions, Algorithm 7 can be implemented to run in a time polynomial in both* $\log B$ *and* $\log p$.

Proof: The analysis is similar to the proof of Proposition 6. □

We remark that this algorithm is implicitly used in the recent identification protocol of Galbraith, Silva and Petit [13].

4.2 An attack on CGL hash function

It was shown in [6] that computing collisions or preimages for Charles-Goren-Lauter hash function amounts to computing large ℓ-power degree isogenies between two (possibly isomorphic) elliptic curves. The hardness arguments for these problems then essentially relied on the following arguments:

1. In general, these isogenies must have a degree so large that they cannot be efficiently computed with current algorithms.

2. The best known algorithms for these problems were variants of birthday searches, with an exponential complexity in the parameter's size [11].

Paradoxically, the quaternion ℓ-isogeny algorithm [18] leads to both the security arguments of Section 3.3 and to a partial attack against the hash function. More precisely, in this section we present a collision attack for the hash function when the initial point used in the random walk is a special elliptic curve E_0 as defined in Section 2.3.

Our attack is summarized by Algorithm 8 below. We first compute a collision for a "quaternion version" of Charles-Goren-Lauter hash function. This then essentially amounts to finding $q \in \langle 1, i, j, k\rangle \subset \mathcal{O}_0$ with $\mathrm{n}(q) = \ell^e$ for some e, which defines a sequence of ideals I_i corresponding to a path from $\mathcal{O}_0$ to $q^{-1}\mathcal{O}_0 q \approx \mathcal{O}_0$. Applying the translation algorithm directly to this sequence of ideals would have a prohibitive cost because ℓ^e is larger than p. To solve this problem we first replace each ideal in the sequence by another ideal in the same class but with powersmooth norm, and we apply the translation algorithm to each of them individually to obtain corresponding isogenies. The end vertices of these isogenies form a sequence of j-invariants that define a collision for the original elliptic curve version of Charles-Goren-Lauter hash function.

Algorithm 8 Collision attack on CGL hash function for special initial points

Require: Special j_0 and $\mathcal{O}_0$ as defined in Section 2.3.
Ensure: A sequence of j invariants $j_0, j_1, \ldots, j_e = j_0$ such that for any i there exists an isogeny of degree ℓ from $E(j_i)$ to $E(j_{i+1})$.
1: Compute $e \in \mathbb{N}$ and $q \in \langle 1, i, j, k\rangle \subset \mathcal{O}_0$ with $\mathrm{n}(q) = \ell^e$.
2: Compute a sequence of ideals $I_i = \mathcal{O}_0 q + \mathcal{O}_0 \ell^i$.
3: **for** all i **do**
4: Compute J_i with powersmooth norm in the same class as I_i.
5: Translate J_i into an isogeny $\varphi_i : E_0 \to E_i$.
6: **end for**
7: **return** $(j_0, j(E_1), j(E_2), \ldots, j(E_e) = j_0)$.

To obtain an element with a power of ℓ norm in Step 1, we fix e large enough then pick random values of y and z until the equation $w^2 + qx^2 = \ell^e - p(y^2 + qz^2)$ can be solved with Cornacchia's algorithm. This solution is described in Algorithm 9.

Algorithm 9 Power of ℓ norm element in $\mathcal{O}_0$

Require: Maximal order $\mathcal{O}_0 \subset B_{p,\infty}$ as defined in Section 2.3.
Ensure: $e \in \mathbb{N}$ and $q \in \mathcal{O}_0$ with $\mathrm{n}(q) = \ell^e$.

1: Let $e = \lceil 2 \log p \rceil$.
2: Choose random y, z smaller than $\sqrt{p/q}$.
3: Let $N \leftarrow \ell^e - p(y^2 + qz^2)$.
4: Find $w, x \in \mathbb{Z}$ such that $w^2 + qx^2 = N$ if there are some, otherwise go to Step 2.
5: **return** $q = w + xi + yj + zk$.

Proposition 15 *There exists an algorithm that computes a collision for Charles-Goren-Lauter hash function when the initial vertex is a special curve, in a time polynomial in* $\log p$.

Proof: In Algorithm 9 we expect that the Equation in Step 4 will have solution for a proportion $1/2q \log p$ of the random choices (y, z), so we expect this algorithm to run in time polynomial in $\log p$. Note that $e = \lceil 2 \log p \rceil$, and that Steps 4 and 5 in Algorithm 8 both run in a time polynomial in $\log p$. We conclude that the runtime of Algorithm 8 is also polynomial in $\log p$. □

We remark that we described our attack only for the maximal orders $\mathcal{O}_0$ defined in Section 2.3, but it can be extended to other maximal orders as long as the corresponding curve is known or can be computed, and as long as elements of norm a power of ℓ can be found in the order. This is the case for "special" orders, as defined in [18].

The attack provided by Algorithm 8 can be extended into a "backdoor attack" where an entity in charge of deciding the initial vertex for the hash function play the role of the attacker. This entity could take a random walk from j_0 to another curve E and publish this $j(E)$ as the initial vertex for the hash function. Due to the random walk the vertex $j(E)$ will be uniformly distributed, hence the function will be collision resistant based on the assumption that the endomorphism ring computation problem is hard (see Proposition 13). However, the entity can concatenate the path from j_0 to j and the collision around j_0 to obtain a collision around j.

To the best of our knowledge, there exists no efficient algorithm to sample supersingular j invariants that does not involve this random walk procedure, so the backdoor attack cannot really be avoided. On the other hand, by inspecting such a collision, it is easy to recover a path to $\mathcal{O}_0$ and that will reveal that a backdoor was inserted. In that sense, the backdoor mechanism may not be too much of an issue in practice.

5 Conclusion

In the context of NIST's post-quantum cryptography call [1], there is a lot of interest in cryptosystems based on isogeny problems. In this paper we build on the quaternion ℓ-isogeny algorithm of Kohel-Lauter-Petit-Tignol [18] to solve some relevant problems in this area and to develop reductions between other relevant problems.

One consequence of our work is a new one-way function based on the hardness of computing the endomorphism ring of supersingular elliptic curves. Indeed on the one hand we provided an efficient algorithm to compute Deuring's correspondence in one direction, from maximal orders to j invariants, and on the other hand we showed the equivalence of the other direction with solving the endomorphism ring computation problem.

We also considered the security of Charles, Goren and Lauter's hash function [6]. We showed that for randomly chosen initial vertices both collision and resistance of this function are equivalent to hardness of the endomorphism ring computation problem, and we provided an efficient collision algorithm for special parameters and discussed the potential impact of this attack.

Our work confirms that the endomorphism ring computation problem is a key problem for isogeny-based cryptography, as was already suggested in [12, 13, 21]. The problem has now been studied for more than twenty years under different forms [17, 5], but clearly not as extensively as classical problems like integer factorization. We stress that the security of Jao and De Feo's key exchange protocol relies on problems that are potentially easier to solve than the endomorphism ring computation problem. All these problems should now be under additional scrutiny in the context of NIST's call.

Most of our results are true under the general heuristic assumption that "numbers generated following certain distributions behave like random numbers of the same size, unless there is a good reason for this to be false". We believe that any failure of this general heuristic will suggest improvements to our algorithms rather than modify our general conclusions.

6 Historical Comments and Acknowledgements

This work started in 2010 when Christophe Petit, then at Université catholique de Louvain, visited Kristin Lauter in San Diego. Most of the results of this paper were obtained between then and 2012. A draft including an early version of the ANTS algorithm [18] was circulated among a few experts; talks were given on its content; and parts of it were either cited or developed in other publications that needed it. This has led to a situation were, for example, the equivalence of Problem 4 and isogeny problems is well-accepted among experts [13, Section 2.1] but no justification of this fact is publicly accessible.

As the importance of our results has grown with the community interest in isogeny-based cryptography, we have finally come to revisit our draft and make its contents publicly available, so that cryptographers and cryptanalysts can

build on our techniques in their own work.

Between 2008 and now we have had many interesting discussions on the problems discussed in this paper and on isogeny-based cryptography in general, in particular with Steven Galbraith, David Kohel, Luca De Feo, Jérome Plût, Damien Robert and Yan Bo Ti. These discussions have improved our understanding of these problems, fueled our research, and influenced this new write-up of our results. Many thanks to all of you.

References

[1] Post-quantum crypto project. `http://csrc.nist.gov/groups/ST/post-quantum-crypto/`.

[2] N. C. Ankeny. The least quadratic non residue. *Annals of Mathematics*, 55(1):65– 72, 1952.

[3] Gaetan Bisson and Andrew V. Sutherland. Computing the endomorphism ring of an ordinary elliptic curve over a finite field. *Journal of Number Theory*, 131(5):815–831, May 2011.

[4] Reinier Bröker. Constructing supersingular elliptic curves.

[5] Juan Marcos Cerviño. On the correspondence between supersingular elliptic curves and maximal quaternionic orders. `http://arxiv.org/abs/math/0404538v1`, 2004.

[6] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.

[7] 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 Matematiche di Battaglini*, 46:33–90, 1903.

[8] Christina Delfs and Steven D. Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Des. Codes Cryptography*, 78(2):425–440, 2016.

[9] Max Deuring. Die typen der multiplikatorenringe elliptischer funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941. 10.1007/BF02940746.

[10] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology*, 8(3):209–247, 2014.

[11] Steven D. Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS Journal of Computation and Mathematics*, 2:118–138, 1999.

[12] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 63–91, 2016.

[13] Steven D. Galbraith, Christophe Petit, and Javier Silva. Signature schemes based on supersingular isogeny problems. Cryptology ePrint Archive, Report 2016/1154 (to appear at Asiacrypt 2017), 2016. `http://eprint.iacr.org/2016/1154`.

[14] Alexandre Gélin and Benjamin Wesolowski. Loop-abort faults on supersingular isogeny cryptosystems. In Lange and Takagi [19], pages 93–106.

[15] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006.

[16] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, pages 19–34, 2011.

[17] David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.

[18] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17A:418–432, 2014.

[19] Tanja Lange and Tsuyoshi Takagi, editors. *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *Lecture Notes in Computer Science*. Springer, 2017.

[20] Ken Mac Murdy and Kristin Lauter. Explicit generators for endomorphism rings of supersingular elliptic curves. `https://phobos.ramapo.edu/~kmcmurdy/research/ss_endomorphisms.pdf`, 2004.

[21] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. *IACR Cryptology ePrint Archive*, 2017:571, 2017.

[22] Arnold Pizer. An algorithm for computing modular forms on $\gamma_0(n)^*$. *Journal of Algebra*, 64:340–390, 1980.

[23] Joseph Silverman. *The Arithmetic of Elliptic Curves*. Springer Verlag, 1986.

[24] Denis Simon. Quadratic equations in dimensions 4, 5 and more. Preprint, `http://www.math.unicaen.fr/~simon/`, 2005.

[25] Yan Bo Ti. Fault attack on supersingular isogeny cryptosystems. In Lange and Takagi [19], pages 107–122.

[26] Marie-France Vignéras. *Arithmétique des algébres de quaternions*. Springer-Verlag, 1980.

[27] Jacques Vélu. Isogénies entre courbes elliptiques. *Communications de l'Académie royale des Sciences de Paris*, 273:238–241, 1971.

[28] William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'E.N.S.*, 2:521–560, 1969.

[29] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. Financial Crypto, 2017.