Hard Isogeny Problems over RSA Moduli and Groups with Infeasible Inversion

Salim Ali Altuğ[1] and Yilei Chen[2]

[1] Boston University, Boston, USA
saaltug@bu.edu
[2] Visa Research, Palo Alto, USA
yilchen@visa.com

Abstract. We initiate the study of computational problems on elliptic curve isogeny graphs defined over RSA moduli. We conjecture that several variants of the neighbor-search problem over these graphs are hard, and provide a comprehensive list of cryptanalytic attempts on these problems. Moreover, based on the hardness of these problems, we provide a construction of groups with infeasible inversion, where the underlying groups are the ideal class groups of imaginary quadratic orders.
Recall that in a group with infeasible inversion, computing the inverse of a group element is required to be hard, while performing the group operation is easy. Motivated by the potential cryptographic application of building a directed transitive signature scheme, the search for a group with infeasible inversion was initiated in the theses of Hohenberger and Molnar (2003). Later it was also shown to provide a broadcast encryption scheme by Irrer et al. (2004). However, to date the only case of a group with infeasible inversion is implied by the much stronger primitive of self-bilinear map constructed by Yamakawa et al. (2014) based on the hardness of factoring and indistinguishability obfuscation (iO). Our construction gives a candidate without using iO.

1 Introduction

Let $\mathbb{G}$ denote a finite group written multiplicatively. The discrete-log problem asks to find the exponent a given g and $g^a \in \mathbb{G}$. In the groups traditionally used in discrete-log-based cryptosystems, such as $(\mathbb{Z}/q\mathbb{Z})^*$ [11], groups of points on elliptic curves [29,22], and class groups [3,28], computing the inverse $x^{-1} = g^{-a}$ given $x = g^a$ is easy. We say $\mathbb{G}$ is a *group with infeasible inversion* if computing inverses of elements is hard, while performing the group operation is easy (i.e. given g, g^a, g^b, computing g^{a+b} is easy).

The search for a group with infeasible inversion was initiated in the theses of Hohenberger [18] and Molnar [30], motivated with the potential cryptographic application of constructing a directed transitive signature. It was also shown by Irrer et al. [20] to provide a broadcast encryption scheme. The only existing candidate of such a group, however, is implied by the much stronger primitive of self-bilinear maps constructed by Yamakawa et al. [40], assuming the hardness of integer factorization and indistinguishability obfuscation (iO) [2,16].

In this paper we propose a candidate trapdoor group with infeasible inversion without using iO. The underlying group is isomorphic to the ideal class group of an imaginary quadratic order (henceforth abbreviated as "*the class group*"). In the standard representation of the class group, computing the inverse of a group element is straightforward. The representation we propose uses the volcano-like structure of the isogeny graphs of ordinary elliptic curves. In fact, the initiation of this work was driven by the desire to explore the computational problems on the isogeny graphs defined over RSA moduli.

1.1 Elliptic curve isogenies in cryptography

An isogeny $\varphi : E_1 \to E_2$ is a morphism of elliptic curves that preserves the identity. Given two isogenous elliptic curves E_1, E_2 over a finite field, finding an explicit rational polynomial that represents an isogeny from E_1 to E_2 is traditionally called the *computational isogeny problem*.

The best way of understanding the nature of the isogeny problem is to look at the *isogeny graphs*. Fix a finite field $\mathbf{k}$ and a prime ℓ different than the characteristic of $\mathbf{k}$. Then the isogeny graph $G_\ell(\mathbf{k})$ is defined as follows: each vertex in $G_\ell(\mathbf{k})$ is a j-invariant of an isomorphism class of curves; two vertices are connected by an edge if there is an isogeny of degree ℓ over $\mathbf{k}$ that maps one curve to another. The structure of the isogeny graph is described in the PhD thesis of Kohel [23]. Roughly speaking, a connected component of an isogeny graph containing ordinary elliptic curves looks like a *volcano* (termed in [15]). The connected component containing supersingular elliptic curves, on the other hand, has a different structure. In this article we will focus on the ordinary case.

A closer look at the algorithms of computing isogenies. Let $\mathbf{k}$ be a finite field of q elements, ℓ be an integer such that $\gcd(\ell, q) = 1$. Given the j-invariant of an elliptic curve E, there are at least two different ways to find all the j-invariants of the curves that are ℓ-isogenous to E (or to a twist of E) and to find the corresponding rational polynomials that represent the isogenies:

1. Computing kernel subgroups of E of size ℓ, and then applying Vélu's formulae to obtain explicit isogenies and the j-invariants of the image curves,
2. Calculating the j-invariants of the image curves by solving the ℓ^{th} modular polynomial Φ_ℓ over $\mathbf{k}$, and then constructing explicit isogenies from these j-invariants.

Both methods are able to find all the ℓ-isogenous neighbors over $\mathbf{k}$ in time $\mathsf{poly}(\ell, \log(q))$. In other words, *over a finite field*, one can take a stroll around the polynomial-degree isogenous neighbors of a given elliptic curve efficiently.

However, for two random isogenous curves over a sufficiently large field, finding an explicit isogeny between them seems to be hard, even for quantum computers. The conjectured hardness of computing isogenies was used in a key-exchange and a public-key cryptosystem by Couveignes [7] and independently by Rostovtsev and Stolbunov [31]. Moreover, a hash function and a key exchange scheme were proposed based on the hardness of computing isogenies over

supersingular curves [4,21]. Isogeny-based cryptography is attracting attention partially due to its conjectured post-quantum security.

1.2 Isogeny graphs over RSA moduli

Let p, q be primes and let $N = pq$. In this work we consider computational problems related to elliptic curve isogeny graphs defined over $\mathbb{Z}/N\mathbb{Z}$, where the prime factors p, q of N are unknown. An isogeny graph over $\mathbb{Z}/N\mathbb{Z}$ is defined first by fixing the isogeny graphs over $\mathbb{F}_p$ and $\mathbb{F}_q$, then taking a graph tensor product; obtaining the j-invariants in the vertices of the graph over $\mathbb{Z}/N\mathbb{Z}$ by the Chinese remainder theorem. Working over the ring $\mathbb{Z}/N\mathbb{Z}$ without the factors of N creates new sources of computational hardness from the isogeny problems. Of course, by assuming the hardness of factorization, we immediately lose the post-quantum privilege of the "traditional" isogeny problems. From now on all the discussions of hardness are with respect to the polynomial time classical algorithms.

Basic neighbor search problem over $\mathbb{Z}/N\mathbb{Z}$. When the factorization of N is unknown, it is not clear how to solve the basic problem of finding (even one of) the ℓ-isogenous neighbors of a given elliptic curve. The two algorithms over finite fields we mentioned seem to fail over $\mathbb{Z}/N\mathbb{Z}$ since both of them require solving polynomials over $\mathbb{Z}/N\mathbb{Z}$, which is hard in general when the factorization of N is unknown. In fact, we show that if it is feasible to find all the ℓ-isogenous neighbors of a given elliptic curve over $\mathbb{Z}/N\mathbb{Z}$, then it is feasible to factorize N.

Joint-neighbor search problem over $\mathbb{Z}/N\mathbb{Z}$. Suppose we are given several j-invariants over $\mathbb{Z}/N\mathbb{Z}$ that are connected by polynomial-degree isogenies, we ask whether it is feasible to compute their joint isogenous neighbors. For example, in the isogeny graph on the LHS of Figure 1, suppose we are given j_0, j_1, j_2, and the degrees ℓ between j_0 and j_1, and m between j_0 and j_2 such that $\gcd(\ell, m) = 1$. Then we can find j_3 which is m-isogenous to j_1 and ℓ-isogenous to j_2, by computing the polynomial $f(x) = \gcd(\Phi_m(j_1, x), \Phi_\ell(j_2, x))$ over $\mathbb{Z}/N\mathbb{Z}$. When $\gcd(\ell, m) = 1$ the polynomial $f(x)$ turns out to be linear with its only root being j_3, hence computing the (ℓ, m) neighbor in this case is feasible.

However, not all the joint-isogenous neighbors are easy to find. As an example, consider the following (ℓ, ℓ^2)-*joint neighbor problem* illustrated on the RHS of Figure 1. Suppose we are given j_0 and j_1 that are ℓ-isogenous, and asked to find another j-invariant j_{-1} which is ℓ-isogenous to j_0 and ℓ^2-isogenous to j_1. The natural way is to take the gcd of $\Phi_\ell(j_0, x)$ and $\Phi_{\ell^2}(j_1, x)$, but in this case the resulting polynomial is of degree $\ell > 1$ and we are left with the problem of finding a root of it over $\mathbb{Z}/N\mathbb{Z}$, which is believed to be computationally hard without knowing the factors of N.

Currently we do not know if solving this problem is as hard as factoring N. Neither do we know of an efficient algorithm of solving the (ℓ, ℓ^2)-joint neighbor problem. We will list our attempts in solving the (ℓ, ℓ^2)-joint neighbor problem in Section 5.2.

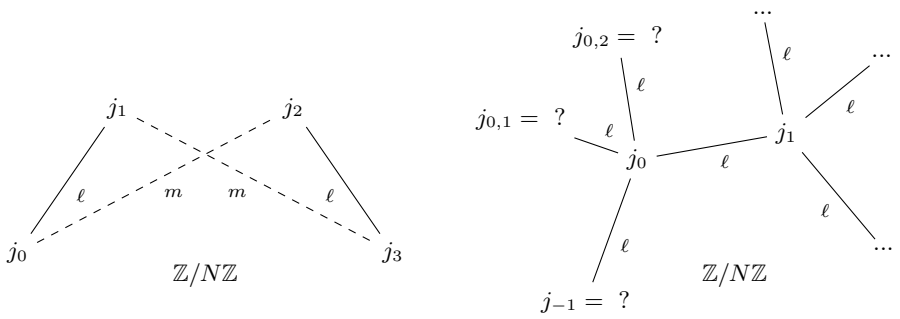

Fig. 1. Left: the (ℓ, m)-isogenous neighbor problem where $\gcd(\ell, m) = 1$. Right: the (ℓ, ℓ^2)-isogenous neighbor problem.

The conjectured computational hardness of the (ℓ, ℓ^2)-joint neighbor problem is fundamental to the infeasibility of inversion in the group we construct.

1.3 Constructing a trapdoor group with infeasible inversion

To explain the construction of the trapdoor group with infeasible inversion (TGII), it is necessary to recall the connection of the ideal class groups and elliptic curve isogenies. Let $\mathbf{k}$ be a finite field as before and let E be an elliptic curve over $\mathbf{k}$ whose endomorphism ring is isomorphic to an imaginary quadratic order $\mathcal{O}$. The group of invertible $\mathcal{O}$-ideals acts on the set of elliptic curves with endomorphism ring $\mathcal{O}$. The ideal class group $\mathcal{CL}(\mathcal{O})$ acts faithfully and transitively on the set

$$\mathrm{Ell}_{\mathcal{O}}(\mathbf{k}) = \{j(E) : E \text{ with } \mathrm{End}(E) \simeq \mathcal{O}\}.$$

In other words, there is a map

$$\mathcal{CL}(\mathcal{O}) \times \mathrm{Ell}_{\mathcal{O}}(\mathbf{k}) \to \mathrm{Ell}_{\mathcal{O}}(\mathbf{k}), \quad (\mathfrak{a}, j) \mapsto \mathfrak{a} * j$$

such that $\mathfrak{a} * (\mathfrak{b} * j) = (\mathfrak{a}\mathfrak{b}) * j$ for all $\mathfrak{a}, \mathfrak{b} \in \mathcal{CL}(\mathcal{O})$ and $j \in \mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$; for any $j, j' \in \mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$, there is a unique $\mathfrak{a} \in \mathcal{CL}(\mathcal{O})$ such that $j' = \mathfrak{a} * j$. The cardinality of $\mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$ equals to the class number $h(\mathcal{O})$.

We are now ready to provide an overview of the TGII construction with a toy example in Figure 2.

Parameter generation. To simplify this overview let us assume that the group $\mathcal{CL}(\mathcal{O})$ is cyclic, in which case the group $\mathbb{G}$ with infeasible inversion is exactly $\mathcal{CL}(\mathcal{O})$ (in the detailed construction we usually choose a cyclic subgroup of $\mathcal{CL}(\mathcal{O})$). To generate the public parameter for the group $\mathcal{CL}(\mathcal{O})$, we choose two primes p, q and curves $E_{0,\mathbb{F}_p}$ over $\mathbb{F}_p$ and $E_{0,\mathbb{F}_q}$ over $\mathbb{F}_q$ such that the endomorphism rings of $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$ are both isomorphic to $\mathcal{O}$. Let $N = p \cdot q$. Let E_0 be an elliptic curve over $\mathbb{Z}/N\mathbb{Z}$ as the CRT composition of $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$. The j-invariant of E_0, denoted as j_0, equals to the CRT composition of

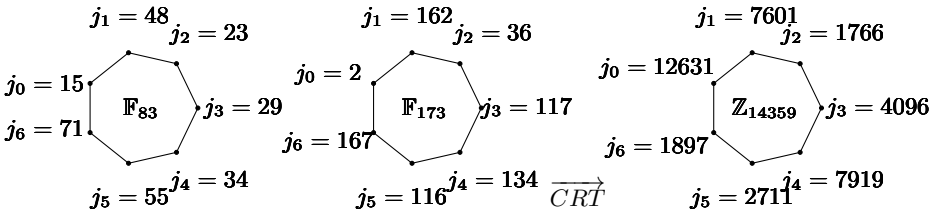

Fig. 2. A representation of $\mathcal{CL}(-251)$ by a 3-isogeny volcano over $\mathbb{Z}_{14359}$ of size $h(-251) = 7$. The $\mathbb{F}_{83}$ part is taken from [31].

the j-invariants of $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$. The identity of $\mathcal{CL}(\mathcal{O})$ is represented by j_0. The public parameter of the group is (N, j_0).

In the example of Figure 2, we set the discriminant D of the imaginary quadratic order $\mathcal{O}$ to be -251. The group order is then the class number $h(\mathcal{O}) = 7$. Choose $p = 83$, $q = 173$, $N = pq = 14359$. Fix a curve E_0 so that $j(E_{0,\mathbb{F}_p}) = 15$, $j(E_{0,\mathbb{F}_q}) = 2$, then $j_0 = \mathsf{CRT}(83, 173; 15, 2) = 12631$. The public parameter is $(14359, 12631)$.

The encodings. We provide two types of encodings for each group element: the canonical and composable embeddings. The *canonical encoding* of an element is uniquely determined once the public parameter is fixed and it can be directly used in the equivalence test. It, however, does not support efficient group operations. The *composable encoding* of an element, on the other hand, supports efficient group operations with the other composable encodings. Moreover, a composable encoding can be converted to a canonical encoding by an efficient, public extraction algorithm.

An element $x \in \mathcal{CL}(\mathcal{O})$ is canonically represented by the j-invariant of the elliptic curve $x * E_0$ (once again, obtained over $\mathbb{F}_p$ and $\mathbb{F}_q$ then composed by CRT), and we call $j(x * E_0)$ the canonical encoding of x. Note that the canonical encodings of all the elements are fixed once j_0 and N are fixed.

To make things concrete, let $a = \sqrt{-251}$ and consider the toy example above. The ideal class $x = [(3, \frac{a+1}{2})]$ acting on E_0 over $\mathbb{F}_p$ gives $j(x * E_{0,\mathbb{F}_p}) = 48$, over $\mathbb{F}_q$ gives $j(x * E_{0,\mathbb{F}_q}) = 162$. The canonical encoding of x is then $j_1 = \mathsf{CRT}(83, 173; 48, 162) = 7601$. Similarly, the canonical encodings of the ideal classes $[(7, \frac{a-1}{2})]$, $[(5, \frac{a+7}{2})]$, $[(5, \frac{a+3}{2})]$, $[(7, \frac{a+1}{2})]$, $[(3, \frac{a-1}{2})]$ are 1766, 4096, 7919, 2711, 1897.

The composable encodings and the composition law. To generate a composable encoding of $x \in \mathcal{CL}(\mathcal{O})$, we factorize x as $x = \prod_{x_i \in S} x_i^{e_i}$, where S denotes a generating set, and both the norms $N(x_i)$ and the exponents e_i being polynomial in size. The composable encoding of x then consists of the norms $N(x_i)$ and the j-invariants of $x_i^k * E_0$, for $k \in [e_i]$, for $i \in [|S|]$. The *degree* of a composable encoding is defined to be the product of the norms of the ideals $\prod_{x_i \in S} N(x_i)^{e_i}$. Note that the degree depends on the choice of S and the factorization of x, which is not unique.

As an example let us consider the simplest situation, where the composable encodings are just the canonical encodings themselves together with the norms of the ideals (i.e. the degrees of the isogenies). Set the composable encoding of $x = [(3, \frac{a+1}{2})]$ be $(3, 7601)$, the composable encoding of $y = [(7, \frac{a-1}{2})]$ be $(7, 1766)$.

Let us remark an intricacy of the construction of composable encodings. When the degrees of the composable encodings of x and y are coprime and polynomially large, the composition of x and y can be done simply by concatenating the corresponding encodings. To extract the canonical encoding of $x \circ y$, we take the gcd of the modular polynomials. In the example above, the canonical encoding of $x \circ y$ can be obtained by taking the gcd of $\Phi_7(7601, x)$ and $\Phi_3(1766, x)$ over $\mathbb{Z}/N\mathbb{Z}$. Since the degrees are coprime, the resulting polynomial is linear, with the only root being 4096, which is the canonical encoding of $[(5, \frac{a+7}{2})]$.

Note, however, that if the degrees share prime factors, then the gcd algorithm does not yield a linear polynomial, so the the above algorithm for composition does not go through. To give a concrete example to what this means let us go back to our example: if we represent $y = [(7, \frac{a-1}{2})]$ by first factorizing y as $[(3, \frac{a+1}{2})]^2$ we then get the composable encoding of y as $(3, (7601, 1766))$. In this case the gcd of $\Phi_{3^2}(7601, x)$ and $\Phi_3(1766, x)$ over $\mathbb{Z}/N\mathbb{Z}$ yields a degree 3 polynomial, where it is unclear how to extract the roots. Hence, in this case we cannot calculate the canonical embedding of $x \circ y$ simply by looking at the gcd.

Therefore, to facilitate the efficient compositions of the encodings of group elements, we will need to represent them as the product of *pairwise co-prime* ideals with polynomially large norms. This, in particular, means the encoding algorithm will need to keep track on the primes used in the degrees of the composable encodings in the system. In other words, the encoding algorithm is *stateful*.

The infeasibility of inversion. The infeasibility of inversion amounts to the hardness of the computation of the canonical embedding of an element $x^{-1} \in \mathbb{G}$ from a composable encoding of x, and it is based on the hardness of the (ℓ, ℓ^2)-isogenous neighbors problem for each ideal of a composable encoding.

Going back to our example, given the composable encoding $(3, 7601)$ of $x = [(3, \frac{a+1}{2})]$, the canonical encoding of $x^{-1} = [(3, \frac{a-1}{2})]$ is a root of $f(x) = \gcd(\Phi_{3^2}(7601, x), \Phi_3(12631, x))$. The degree of f, however, is 3, so that it is not clear how to extract the root efficiently over an RSA modulus.

The difficulty of sampling the class group invariants and its implications. Let us remark that the actual instantiation of TGII is more involved. A number of challenges arise solely from working with the ideal class groups of imaginary quadratic orders. To give a simple example of the challenges we face, efficiently generating a class group with a known large prime class number is a well-known open problem. Additionally, our construction requires more than the class number (namely, a short basis of the relation lattice of the class group) to support an efficient encoding algorithm.

In our solution, we choose the discriminant D to be of size roughly $\lambda^{O(\log \lambda)}$ and polynomially smooth, so as to make the parameter generation algorithm

and the encoding algorithm run in polynomial time. The discriminant D (i.e. the description of the class group $\mathcal{CL}(D)$) has to be hidden to preserve the plausible $\lambda^{O(\log \lambda)}$-security of the TGII. Furthermore, even if D is hidden, there is an $\lambda^{O(\log \lambda)}$ attack by first guessing D or the group order, then solving the discrete-log problem given the polynomially-smooth group order. Extending the working parameters regime seems to require the solutions of several open problems concerning ideal class groups of imaginary quadratic orders.

Summary of the TGII construction. To summarize, our construction of TGII chooses two sets of j-invariants that correspond to elliptic curves with the same imaginary quadratic order $\mathcal{O}$ over $\mathbb{F}_p$ and $\mathbb{F}_q$, and glues the j-invariants via the CRT composition as the canonical encodings of the group elements in $\mathcal{CL}(\mathcal{O})$. The composable encoding of a group element x is given as several j-invariants that represent the smooth ideals in a factorization of x. The efficiency of solving the (ℓ, m)-joint-neighbor problem over $\mathbb{Z}/N\mathbb{Z}$ facilitates the efficient group operation over coprime degree encodings. The conjectured hardness of the (ℓ, ℓ^2)-joint-neighbor problem over $\mathbb{Z}/N\mathbb{Z}$ is the main reason behind the hardness of inversion, but it also stops us from composing encodings that share prime factors.

The drawbacks of our construction of TGII are as follows.

1. Composition is only feasible for coprime-degree encodings, which means in order to publish arbitrarily polynomially many encodings, the encoding algorithm has to be stateful in order to choose different polynomially large prime factors for the degrees of the encoding (we cannot choose polynomially large prime degrees and hope they are all different).
2. In the definition from [18,30], the composable encodings obtained during the composition are required to be indistinguishable to a freshly sampled encoding. In our construction the encodings keep growing during compositions, until they are extracted to the canonical encoding which is a single j-invariant.
3. In addition to the (ℓ, ℓ^2)-joint-neighbor problem, the security of the TGII construction relies on several other heuristic assumptions. We will list our cryptanalytic attempts in §5.3. Moreover, even if we have not missed any attacks, the current best attack only requires $\lambda^{O(\log \lambda)}$-time, by first guessing the discriminant or the group order.

The two applications of TGII. Let us briefly mention the impact of the limitation of our TGII on the applications of directed transitive signature (DTS) [18,30] and broadcast encryption [20]. For the broadcast encryption from TGII [20], the growth of the composable encodings do not cause a problem. For DTS, in the direct instantiation of DTS from TGII [18,30], the signature is a composable encoding, so the length of the signature keeps growing during the composition, which is an undesirable feature for a non-trivial DTS. So on top of the basic instantiation, we provide an additional compression technique to shrink the composed signature.

Let us also remark that in the directed transitive signature [18,30], encodings are sampled by the master signer; in the broadcast encryption scheme [20], encodings are sampled by the master encrypter. At least for these two applications, having the master signer/encrypter being stateful is not ideal but acceptable.

Organization. The rest of the paper is organized as follows. Section 2 provides the background of imaginary quadratic fields, elliptic curves and isogenies. Section 3 defines the computational problems for isogeny graphs over composite moduli. Section 4 provides our basic construction of a trapdoor group with infeasible inversion. Section 5 provides a highlight of our cryptanalysis attempts.

2 Preliminaries

Notation and terminology. Let $\mathbb{C}, \mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{N}$ denote the set of complex numbers, reals, rationals, integers, and positive integers respectively. For any field K we fix an algebraic closure and denote it by $\bar{K}$. For $n \in \mathbb{N}$, let $[n] := \{1, ..., n\}$. For $B \in \mathbb{R}$, an integer n is called B-smooth if all the prime factors of n are less than or equal to B. An n-dimensional vector is written as a bold lower-case letter, e.g. $\mathbf{v} := (v_1, ..., v_n)$. For an index $k \in \mathbb{N}$, distinct prime numbers p_i for $i \in [k]$, and $c_i \in \mathbb{Z}/p_i\mathbb{Z}$ we will let $\mathsf{CRT}(p_1, ..., p_k; c_1, ..., c_k)$ to denote the unique $y \in \mathbb{Z}/(\prod_i^k p_i)\mathbb{Z}$ such that $y \equiv c_i \pmod{p_i}$, for $i \in [k]$. Given a lattice Λ with a basis $\mathbf{B}$, let $\tilde{\mathbf{B}}$ denote the Gram-Schmidt orthogonalization of $\mathbf{B}$.

Let λ denote the security parameter. In theory and by default, an algorithm is called "efficient" if it runs in probabilistic polynomial time over λ.

2.1 Ideal class groups of imaginary quadratic orders

We recall the necessary background of ideal class groups from [28,5,9].

Let K be an imaginary quadratic field. An *order* $\mathcal{O}$ in K is a subset of K such that (1) $\mathcal{O}$ is a subring of K containing 1; (2) $\mathcal{O}$ is a finitely generated $\mathbb{Z}$-module; (3) $\mathcal{O}$ contains a $\mathbb{Q}$-basis of K. The ring $\mathcal{O}_K$ of integers of K is always an order. For any order $\mathcal{O}$, we have $\mathcal{O} \subseteq \mathcal{O}_K$, in other words $\mathcal{O}_K$ is the maximal order of K with respect to inclusion.

The ideal class group (or class group) of $\mathcal{O}$ is the quotient group $\mathcal{CL}(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O})$ where $I(\mathcal{O})$ denotes the group of proper (i.e. invertible) fractional $\mathcal{O}$-ideals of, and $P(\mathcal{O})$ is its subgroup of principal $\mathcal{O}$-ideals. Let D be the discriminant of $\mathcal{O}$. Note that since $\mathcal{O}$ is quadratic imaginary we have $D < 0$. Sometimes we will denote the class group $\mathcal{CL}(\mathcal{O})$ as $\mathcal{CL}(D)$, and the class number (the group order of $\mathcal{CL}(\mathcal{O})$) as $h(\mathcal{O})$ or $h(D)$.

Let $D = D_0 \cdot f^2$, where D_0 is the *fundamental discriminant* and f is the *conductor* of $\mathcal{O}$ (or D). The following well-known formula relates the class number of an non-maximal order to that of the maximal one:

$$\frac{h(D)}{w(D)} = \frac{h(D_0)}{w(D_0)} \cdot f \prod_{p|f} \left(1 - \frac{\left(\frac{D_0}{p}\right)}{p}\right), \tag{1}$$

where $w(D) = 6$ if $D = -3$, $w(D) = 4$ if $D = -4$, and $w(D) = 2$ if $D < -4$. Let us also remark that the Brauer-Siegel theorem implies that $\ln(h(D)) \sim \ln(\sqrt{|D|})$ as $D \to -\infty$.

Representations. An $\mathcal{O}$-ideal of discriminant D can be represented by its generators, or by its binary quadratic forms. A binary quadratic form of discriminant D is a polynomial $ax^2+bxy+cy^2$ with $b^2-4ac = D$. We denote a binary quadratic form by (a, b, c). The group $SL_2(\mathbb{Z})$ acts on the set of binary quadratic forms and preserves the discriminant. We shall always be assuming that our forms are positive definite, i.e. $a > 0$. Recall that a form (a, b, c) is called *primitive* if $\gcd(a, b, c) = 1$, and a primitive form is called *reduced* if $-a < b \leq a < c$ or $0 \leq b \leq a = c$. Reduced forms satisfy $a \leq \sqrt{|D|/3}$.

A fundamental fact, which goes back to Gauss, is that in each equivalence class, there is a unique reduced form (see Corollary 5.2.6 of [5]). Given a form (a, b, c), denote $[(a, b, c)]$ as its equivalence class. Note that when D is fixed, we can denote a class simply by $[(a, b, \cdot)]$. Efficient algorithms of composing forms and computing the reduced form can be found in [28, Page 9].

2.2 Elliptic curves and their isogenies

In this section we will recall some background on elliptic curves and isogenies. All of this material is well-known and the main references for this section are [23,34,35,37,14].

Let E be an elliptic curve defined over a finite field $\mathbf{k}$ of characteristic $\neq 2, 3$ with q elements, given by its Weierstrass form $y^2 = x^3+ax+b$ where $a, b \in \mathbf{k}$. By the Hasse bound we know that the order of the $\mathbf{k}$-rational points $E(\mathbf{k})$ satisfies

$$-2\sqrt{q} \leq \#E(\mathbf{k}) - (q + 1) \leq 2\sqrt{q}.$$

Here, $t = q + 1 - \#E(\mathbf{k})$ is the trace of Frobenius endomorphism $\pi : (x, y) \mapsto (x^q, y^q)$. Let us also recall that Schoof's algorithm [32] takes as inputs E and q, computes t, and hence $\#E(\mathbf{k})$, in time $\mathsf{poly}(\log q)$.

The *j-invariant* of E is defined as $j(E) = 1728 \cdot \frac{4a^3}{4a^3+27b^2}$. The values $j = 0$ or 1728 are special and we will choose to avoid these two values throughout the paper. Two elliptic curves are isomorphic over the algebraic closure $\bar{\mathbf{k}}$ if and only if their j-invariants are the same. Note that this isomorphism may not be defined over the base field $\mathbf{k}$, in which case the curves are called twists of each other. It will be convenient for us to use j-invariants to represent isomorphism classes of elliptic curves (including their twists). In many cases, with abuse of notation, a j-invariant will be treated as the same to an elliptic curve over $\mathbf{k}$ in the corresponding isomorphism class.

Isogenies. An *isogeny* $\varphi : E_1 \to E_2$ is a morphism of elliptic curves that preserves the identity. Every nonzero isogeny induces a surjective group homomorphism from $E_1(\bar{\mathbf{k}})$ to $E_2(\bar{\mathbf{k}})$ with a finite kernel. Elliptic curves related by a nonzero

isogeny are said to be isogenous. By the Tate isogeny theorem [38, pg.139] two elliptic curves E_1 and E_2 are isogenous over $\mathbf{k}$ if and only if $\#E_1(\mathbf{k}) = \#E_2(\mathbf{k})$.

The degree of an isogeny is its degree as a rational map. An isogeny of degree ℓ is called an ℓ-isogeny. When $\mathrm{char}(\mathbf{k}) \nmid \ell$, the kernel of an ℓ-isogeny has cardinality ℓ. Two isogenies ϕ and φ are considered equivalent if $\phi = \iota_1 \circ \varphi \circ \iota_2$ for isomorphisms ι_1 and ι_2. Every ℓ-isogeny $\varphi : E_1 \to E_2$ has a unique dual isogeny $\hat{\varphi} : E_2 \to E_1$ of the same degree such that $\varphi \circ \hat{\varphi} = \hat{\varphi} \circ \varphi = [\ell]$, where $[\ell]$ is the multiplication by ℓ map. The kernel of the multiplication-by-ℓ map is the *ℓ-torsion subgroup*

$$E[\ell] = \left\{P \in E(\bar{\mathbf{k}}) : \ell P = 0\right\}.$$

When $\ell \nmid \mathrm{char}(\mathbf{k})$ we have $E[\ell] \simeq \mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z}$. For a prime $\ell \neq \mathrm{char}(\mathbf{k})$, there are $\ell + 1$ cyclic subgroups in $E[\ell]$ of order ℓ, each corresponding to the kernel of an ℓ-isogeny φ from E. An isogeny from E is defined over $\mathbf{k}$ if and only if its kernel subgroup G is defined over $\mathbf{k}$ (namely, for $P \in G$ and $\sigma \in \mathrm{Gal}(\bar{\mathbf{k}}/\mathbf{k})$, $\sigma(P) \in G$; note that this does not imply $G \subseteq E(\mathbf{k})$). If $\ell \nmid \mathrm{char}(\mathbf{k})$ and $j(E) \neq 0$ or 1728, then up to isomorphism the number of ℓ-isogenies from E defined over $\mathbf{k}$ is $0, 1, 2$, or $\ell + 1$.

Modular polynomials. Let $\ell \in \mathbb{Z}$, let $\mathbb{H}$ denote the upper half plane $\mathbb{H} := \{\tau \in \mathbb{C} : \mathrm{im}\ \tau > 0\}$ and $\mathbb{H}^* = \mathbb{H} \cup \mathbb{Q} \cup \{\infty\}$. Let $j(\tau)$ be the classical modular function defined on $\mathbb{H}$. For any $\tau \in \mathbb{H}$, the complex numbers $j(\tau)$ and $j(\ell\tau)$ are the j-invariants of elliptic curves defined over $\mathbb{C}$ that are related by an isogeny whose kernel is a cyclic group of order ℓ. The minimal polynomial $\Phi_\ell(y)$ of the function $j(\ell z)$ over the field $\mathbb{C}(j(z))$ has coefficients that are polynomials in $j(z)$ with inter coefficients. Replacing $j(z)$ with a variable x gives the *modular polynomial* $\Phi_\ell(x, y) \in \mathbb{Z}[x, y]$, which is symmetric in x and y. It parameterizes pairs of elliptic curves over $\mathbb{C}$ related by a cyclic ℓ-isogeny (an isogeny is said to be cyclic if its kernel is a cyclic group; when ℓ is a prime every ℓ-isogeny is cyclic). The modular equation $\Phi_\ell(x, y) = 0$ is a canonical equation for the modular curve $Y_0(\ell) = \mathbb{H}/\Gamma_0(\ell)$, where $\Gamma_0(\ell)$ is the congruence subgroup of $\mathrm{SL}_2(\mathbb{Z})$ defined by

$$\Gamma_0(\ell) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) \middle| \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \pmod{\ell} \right\}.$$

The time and space required for computing the modular polynomial Φ_ℓ are polynomial in ℓ, cf. [12, § 3] or [5, Page 386]. In this article we will only use $\{\Phi_\ell \in \mathbb{Z}[x, y]\}_{\ell \in \mathsf{poly}(\lambda)}$, so we might as well assume that the modular polynomials are computed ahead of time.

2.3 Isogeny volcanoes and the class groups

An isogeny from an elliptic curve E to itself is called an *endomorphism*. Over a finite field $\mathbf{k}$, $\mathrm{End}(E)$ is isomorphic to an imaginary quadratic order when E is ordinary, or an order in a definite quaternion algebra when E is supersingular. In this paper we will be focusing on the ordinary case.

Isogeny graphs. The thesis of [23] describes the graphs that capture the relation of being ℓ-isogenous among elliptic curves over a finite field $\mathbf{k}$.

Definition 1 (ℓ-isogeny graph). *Fix a prime ℓ and a finite field $\mathbf{k}$ such that $\mathrm{char}(\mathbf{k}) \neq \ell$. The ℓ-isogeny graph $G_\ell(\mathbf{k})$ has vertex set $\mathbf{k}$. Two vertices (j_1, j_2) have a directed edge (from j_1 to j_2) with multiplicity equal to the multiplicity of j_2 as a root of $\Phi_\ell(j_1, Y)$. The vertices of $G_\ell(\mathbf{k})$ are j-invariants and each edge corresponds to an (isomorphism classes of an) ℓ-isogeny.*

For $j_1, j_2 \notin \{0, 1728\}$, an edge (j_1, j_2) occurs with the same multiplicity as (j_2, j_1) and thus the subgraph of $G_\ell(\mathbf{k})$ on $\mathbf{k} \setminus \{0, 1728\}$ can be viewed as an undirected graph. $G_\ell(\mathbf{k})$ has super singular and ordinary components. The ordinary components of $G_\ell(\mathbf{k})$ look like *ℓ-volcanoes*:

Definition 2 (ℓ-volcano). *Fix a prime ℓ. An ℓ-volcano V is a connected undirected graph whose vertices are partitioned into one or more levels V_0, ..., V_d such that the following hold:*

1. *The subgraph on V_0 (the surface, or the crater) is a regular graph of degree at most 2.*
2. *For $i > 0$, each vertex in V_i has exactly one neighbor in level V_{i-1}.*
3. *For $i < d$, each vertex in V_i has degree $\ell + 1$.*

Let $\phi : E_1 \to E_2$ by an ℓ-isogeny of elliptic curves with endomorphism rings $\mathcal{O}_1 = \mathrm{End}(E_1)$ and $\mathcal{O}_2 = \mathrm{End}(E_2)$ respectively. Then, there are three possibilities for $\mathcal{O}_1$ and $\mathcal{O}_2$:

- If $\mathcal{O}_1 = \mathcal{O}_2$, then ϕ is called horizontal,
- If $[\mathcal{O}_1 : \mathcal{O}_2] = \ell$, then ϕ is called descending,
- If $[\mathcal{O}_2 : \mathcal{O}_1] = \ell$, then ϕ is called ascending.

Let E be an elliptic curve over $\mathbf{k}$ whose endomorphism ring is isomorphic to an imaginary quadratic order $\mathcal{O}$. Then, the set

$$\mathrm{Ell}_{\mathcal{O}}(\mathbf{k}) = \{j(E) \in \mathbf{k} \mid \text{ with } \mathrm{End}(E) \simeq \mathcal{O}\}$$

is naturally a $\mathcal{CL}(\mathcal{O})$-torsor as follows: For an invertible $\mathcal{O}$-ideal $\mathfrak{a}$ the $\mathfrak{a}$-torsion subgroup

$$E[\mathfrak{a}] = \left\{P \in E(\bar{\mathbf{k}}) : \alpha(P) = 0, \forall \alpha \in \mathfrak{a}\right\}$$

is the kernel of a separable isogeny $\phi_{\mathfrak{a}} : E \to E'$. If the norm $N(\mathfrak{a}) = [\mathcal{O} : a]$ is not divisible by $\mathrm{char}(\mathbf{k})$, then the degree of $\phi_{\mathfrak{a}}$ is $N(\mathfrak{a})$. Moreover, if $\mathfrak{a}$ and $\mathfrak{b}$ are two invertible $\mathcal{O}$-ideals, then $\phi_{\mathfrak{ab}} = \phi_{\mathfrak{a}}\phi_{\mathfrak{b}}$, and if $\mathfrak{a}$ is principal then $\phi_{\mathfrak{a}}$ is an isomorphism. This gives a faithful and transitive action of $\mathcal{CL}(\mathcal{O})$ on $\mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$.

Remark 1 (Linking ideals and horizontal isogenies). When ℓ splits in $\mathcal{O}$ we have $(\ell) = \mathfrak{l} \cdot \bar{\mathfrak{l}}$. Fix an elliptic curve $E(\mathbf{k})$ with $\mathrm{End}(E) \simeq \mathcal{O}$, the two horizontal isogenies $\phi_1 : E \to E_1$ and $\phi_2 : E \to E_2$ can be efficiently associated with the two ideals $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ when $\ell \in \mathsf{poly}(\lambda)$ (cf. [33]). To do so, factorize the characteristic polynomial of Frobenius π as $(x - \mu)(x - \nu) \pmod{\ell}$, where $\mu, \nu \in \mathbb{Z}/\ell\mathbb{Z}$. Given an ℓ-isogeny ϕ from E to E/G, the eigenvalue (say μ) corresponding to the eigenspace G can be verified by picking a point $P \in G$, then check whether $\pi(P) = [\mu]P$ modulo G. If so then μ corresponds to ϕ.

3 Isogeny graphs over composite moduli

Let p, q be distinct primes and set $N = pq$. We will be using elliptic curves over the ring $\mathbb{Z}/N\mathbb{Z}$. We will not be needing a formal treatment of elliptic curves over rings as such a discussion would take us too far afield. Instead, we will be defining objects and quantities over $\mathbb{Z}/N\mathbb{Z}$ by taking the CRT of the corresponding ones over $\mathbb{F}_p$ and $\mathbb{F}_q$, which will suffice for our purposes. This follows the treatment given in [27].

Since the underlying rings will matter, we will denote an elliptic curve over a ring R by $E(R)$. If R is clear from the context we shall omit it from the notation. To begin, let us remark that the number of points $\#(E(\mathbb{Z}/N\mathbb{Z}))$ is equal to $\#(E(\mathbb{F}_p))\cdot\#(E(\mathbb{F}_q))$, and the j-invariant of $E(\mathbb{Z}/N\mathbb{Z})$ is $\mathsf{CRT}(p, q; j(E(\mathbb{F}_p)), j(E(\mathbb{F}_q)))$.

3.1 Isogeny graphs over $\mathbb{Z}/N\mathbb{Z}$

Let N be as above. For every prime $\ell \nmid N$ the isogeny graph $G_\ell(\mathbb{Z}/N\mathbb{Z})$ can be defined naturally as the graph tensor product of $G_\ell(\mathbb{F}_p)$ and $G_\ell(\mathbb{F}_q)$.

Definition 3 (ℓ-isogeny graph over $\mathbb{Z}/N\mathbb{Z}$). *Let ℓ, p, and q be distinct primes and let $N = pq$. The ℓ-isogeny graph $G_\ell(\mathbb{Z}/N\mathbb{Z})$ has*

- *The vertex set of $G_\ell(\mathbb{Z}/N\mathbb{Z})$ is $\mathbb{Z}/N\mathbb{Z}$, identified with $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ by CRT,*
- *Two vertices $v_1 = (v_{1,p}, v_{1,q})$ and $v_2 = (v_{2,p}, v_{2,q})$ are connected if and only if $v_{1,p}$ is connected to $v_{2,p}$ in $G_\ell(\mathbb{F}_p)$ and $v_{1,q}$ is connected to $v_{2,q}$ in $G_\ell(\mathbb{F}_q)$.*

Let us make a remark for future consideration. In the construction of groups with infeasible inversion, we will be working with special subgraphs of $G_\ell(\mathbb{Z}/N\mathbb{Z})$, where the vertices over $\mathbb{F}_p$ and $\mathbb{F}_q$ correspond to j-invariants of curves whose endomorphism rings are the same imaginary quadratic order $\mathcal{O}$. Nevertheless, this is a choice we made for convenience, and it does not hurt to define the computational problems over the largest possible graph and to study them first.

3.2 The ℓ-isogenous neighbors problem over $\mathbb{Z}/N\mathbb{Z}$

Definition 4 (The ℓ-isogenous neighbors problem). *Let p, q be two distinct primes and let $N = pq$. Let ℓ be a polynomially large prime s.t. $\gcd(\ell, N) = 1$. The input of the ℓ-isogenous neighbor problem is N and an integer $j \in \mathbb{Z}/N\mathbb{Z}$ such that there exists (possibly more than) one integer j' that $\Phi_\ell(j, j') = 0$ over $\mathbb{Z}/N\mathbb{Z}$. The problem asks to find such integer(s) j'.*

The following theorem shows that the problem of finding *all* of the ℓ-isogenous neighbors is at least as hard as factoring N.

Theorem 1. *If there is a probabilistic polynomial time algorithm that finds all the ℓ-isogenous neighbors in Problem 4, then there is a probabilistic polynomial time algorithm that solves the integer factorization problem.*

The idea behind the reduction is as follows. Suppose it is efficient to pick an integer j over $\mathbb{Z}/N\mathbb{Z}$, let $j_p = j \pmod p$ and $j_q = j \pmod q$, such that j_p has at least two distinct neighbors in $G_\ell(\mathbb{F}_p)$, and j_q has at least two distinct neighbors in $G_\ell(\mathbb{F}_q)$. In this case if we are able to find *all* the integer solutions $j' \in \mathbb{Z}/N\mathbb{Z}$ such that $\Phi_\ell(j, j') = 0$ over $\mathbb{Z}/N\mathbb{Z}$, then there exist two distinct integers j'_1 and j'_2 among the solutions such that $N > \gcd(j'_1 - j'_2, N) > 1$. One can also show that finding *one* of the integer solutions is hard using a probabilistic argument, assuming the underlying algorithm outputs a random solution when there are multiple ones.

In the reduction we pick the elliptic curve E randomly, so we have to make sure that for a non-negligible fraction of the elliptic curves E over $\mathbb{F}_p$, $j(E) \in G_\ell(\mathbb{F}_p)$ has at least two neighbors. The estimate for this relies on the following lemma:

Lemma 1 ([27] (1.9)). *There exists an efficiently computable positive constant c such that for each prime number $p > 3$, for a set of integers $S \subseteq \left\{s \in \mathbb{Z} \mid |p+1-s| < \sqrt{p}\right\}$, we have*

$$\#'\left\{E \mid E \text{ is an elliptic curve over } \mathbb{F}_p,\ \#E(\mathbb{F}_p) \in S\right\}_{/\simeq_{\mathbb{F}_p}} \geq c\,(\#S - 2)\frac{\sqrt{p}}{\log p}.$$

where $\#'\{E\}_{/\simeq_{\mathbb{F}_p}}$ denotes the number of isomorphism classes of elliptic curves over $\mathbb{F}_p$, each counted with weight $(\#\mathrm{Aut}E)^{-1}$.

Theorem 2. *Let p, ℓ be primes such that $6\ell < \sqrt{p}$. The probability that for a random elliptic curve E over $\mathbb{F}_p$ (i.e. a random pair $(a, b) \in \mathbb{F}_p \times \mathbb{F}_p$ such that $4a^3 + 27b^2 \neq 0$) $j(E) \in G_\ell(\mathbb{F}_p)$ having at least two neighbors is $\Omega(\frac{1}{\log p})$.*

Due to the page limitation we refer the readers to the full version for the proof of Theorem 2.

Proof (Proof of Theorem 1). Suppose that there is a probabilistic polynomial time algorithm A that finds all the ℓ-isogenous neighbors in Problem 4 with non-negligible probability η. We will build a probabilistic polynomial time algorithm A' that solves factoring. Given an integer N, A' samples two random integers $a, b \in \mathbb{Z}/N\mathbb{Z}$ such that $4a^3+27b^2 \neq 0$, and computes $j = 1728 \cdot \frac{4a^3}{4a^3+27b^2}$. With all but negligible probability $\gcd(j, N) = 1$ and $j \neq 0, 1728$; if j happens to satisfy $1 < \gcd(j, N) < N$, then A' outputs $\gcd(j, N)$.

A' then sends N, j_0 to the solver A for Problem 4 for a fixed polynomially large prime ℓ, gets back a set of solutions $\mathcal{J} = \{j_i\}_{i\in[k]}$, where $0 \leq k \leq (\ell+1)^2$ denotes the number of solutions. With probability $\Omega(\frac{1}{\log^2 N})$, the curve $E : y^2 = x^3 + ax + b$ has at least two ℓ-isogenies over both $\mathbb{F}_p$ and $\mathbb{F}_q$ due to Theorem 2. In that case there exists $j, j' \in \mathcal{J}$ such that $1 < \gcd(j - j', N) < N$, which gives a prime factor of N.

3.3 The (ℓ, m)-isogenous neighbors problem over $\mathbb{Z}/N\mathbb{Z}$

Definition 5 (The (ℓ, m)-isogenous neighbors problem). *Let p and q be two distinct primes. Let $N := p \cdot q$. Let ℓ, m be two polynomially large integers s.t. $\gcd(\ell m, N) = 1$. The input of the (ℓ, m)-isogenous neighbor problem is the j-invariants j_1, j_2 of two elliptic curves E_1, E_2 defined over $\mathbb{Z}/N\mathbb{Z}$. The problem asks to find all the integers j' such that $\Phi_\ell(j(E_1), j') = 0$, and $\Phi_m(j(E_2), j') = 0$ over $\mathbb{Z}/N\mathbb{Z}$.*

When $\gcd(\ell, m) = 1$, applying the Euclidean algorithm on $\Phi_\ell(j_1, x)$ and $\Phi_m(j_2, x)$ gives a linear polynomial over x.

Lemma 2 ([13]). *Let $j_1, j_2 \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, and let $\ell, m \neq p$ be distinct primes with $4\ell^2 m^2 < |D|$. Then the degree of $f(x) := \gcd(\Phi_\ell(j_1, x), \Phi_m(j_2, x))$ is less than or equal to 1.*

When $\gcd(\ell, m) = d > 1$, applying the Euclidean algorithm on $\Phi_\ell(j_1, x)$ and $\Phi_m(j_2, x)$ gives a polynomial of degree at least d. We present a proof in the the case where $m = \ell^2$, which has the general idea.

Lemma 3. *Let $p \neq 2, 3$ and $\ell \neq p$ be primes, and let j_0, j_1 be such that $\Phi_\ell(j_0, j_1) = 0 \bmod p$. Let $\Phi_\ell(X, j_0)$ and $\Phi_{\ell^2}(X, j_1)$ be the modular polynomials of levels ℓ and ℓ^2 respectively. Then,*

$$(X - j_1) \cdot \gcd(\Phi_\ell(X, j_0), \Phi_{\ell^2}(X, j_1)) = \Phi_\ell(X, j_0)$$

in $\mathbb{F}_p[X]$. In particular,

$$\deg(\gcd(\Phi_\ell(X, j_0), \Phi_{\ell^2}(X, j_1))) = \ell$$

Proof. Without loss of generality we can, and we do, assume that $\Phi_\ell(X, j_0)$, $\Phi_\ell(X, j_1)$, and $\Phi_{\ell^2}(X, j_1)$ split over $\mathbb{F}_p$ (otherwise we can base change to an extension $k'/\mathbb{F}_p$, where the full ℓ^2-torsion is defined, this does not affect the degree of the gcd).

Assume that the degree of the gcd is $N_{\gcd}$. We have,

$$\deg(\Phi_\ell(X, j_0)) = \ell + 1, \qquad \deg(\Phi_{\ell^2}(X, j_1)) = \ell(\ell + 1). \tag{2}$$

Let E_0, E_1 denote the (isomorphism classes of) elliptic curves with j-invariants j_0 and j_1 respectively, and $\varphi_\ell : E_0 \to E_1$ be the corresponding isogeny. We count the number N_{ℓ^2} of cyclic ℓ^2-isogenies from E_1 two ways. First, N_{ℓ^2} is the number of roots of $\Phi_{\ell^2}(X, j_1)$, which, by (2) and the assumption that $\ell^2 + \ell < p$, is $\ell^2 + \ell$.

Next, recall (cf. Corollary 6.11 of [36]) that every isogeny of degree ℓ^2 can be decomposed as a composition of two degree ℓ isogenies (which are necessarily cyclic). Using this N_{ℓ^2} is bounded above by $N_{\gcd} + \ell^2$, where the first factor counts the number of ℓ^2-isogenies $E_1 \to E$ that are compositions $E_1 \xrightarrow{\hat{\varphi}_\ell} E_0 \to E$, and the second factor counts the isogenies that are compositions $E_1 \to E' \to E$, where $E' \ncong E_1$. Note that we are not counting compositions $E_1 \xrightarrow{\phi} \tilde{E} \xrightarrow{\hat{\phi}} E_1$ since these do not give rise to cyclic isogenies.

This shows that $\ell^2 + \ell \leq \ell^2 + N_{\ell^2} \Rightarrow N_{\gcd} \geq \ell$. On the other hand, by (2) $N_{\gcd} \leq \ell$ since $\Phi_\ell(X, j_0)/(X - j_0)$ has degree ℓ and each root except for j_1 gives a (possibly cyclic) ℓ^2-isogeny by composition with $\hat{\varphi}_\ell$. This implies that $N_{\gcd} = \ell$ and that all the ℓ^2-isogenies obtained this way are cyclic. In particular, we get that the gcd is $\Phi_\ell(X, j_0)/(X - j_1)$.

Discussions. Let us remark that we do not know if solving the (ℓ, ℓ^2)-isogenous neighbors problem is as hard as factoring. To adapt the same reduction in the proof of Theorem 1, we need the feasibility of sampling two integers j_1, j_2 such that $\Phi_\ell(j_1, j_2) = 0 \pmod N$, and j_1 or j_2 has to have another isogenous neighbor over $\mathbb{F}_p$ or $\mathbb{F}_q$. However the feasibility is unclear to us in general.

From the cryptanalytic point of view, a significant difference of the (ℓ, ℓ^2)-isogenous neighbors problem and the ℓ-isogenous neighbors problem is the following. Let ℓ be an odd prime. Recall that an isogeny $\phi : E_1 \to E_2$ of degree ℓ can be represented by a rational polynomial

$$\phi : E_1 \to E_2, \quad (x, y) \mapsto \left(\frac{f(x)}{h(x)^2}, \frac{g(x, y)}{h(x)^3} \right),$$

where $h(x)$ is its *kernel polynomial* of degree $\frac{\ell-1}{2}$. The roots of $h(x)$ are the x-coordinates of the kernel subgroup $G \subset E_1[\ell]$ such that $\phi : E_1 \to E_1/G$.

Given a single j-invariant j' over $\mathbb{Z}/N\mathbb{Z}$, it is infeasible to find a rational polynomial ϕ of degree ℓ that maps from a curve E with j-invariant j' to another curve j'', since otherwise j'' is a solution to the ℓ-isogenous neighbors problem. However, if we are given two j-invariants $j_1, j_2 \in \mathbb{Z}/N\mathbb{Z}$ such that $\Phi_\ell(j_1, j_2) = 0 \pmod N$, as in the (ℓ, ℓ^2)-isogenous neighbors problem; then it is feasible to compute a pair of curves E_1, E_2 such that $j(E_1) = j_1$, $j(E_2) = j_2$, together with an explicit rational polynomial of an ℓ-isogeny from E_1 to E_2. This is because the arithmetic operations involved in computing the kernel polynomial $h(x)$ mentioned in [8,33,12] works over $\mathbb{Z}/N\mathbb{Z}$ by reduction mod N, and does not require the factorization of N.

Proposition 1. *Given $\ell, N \in \mathbb{Z}$ such that $\gcd(\ell, N) = 1$, and two integers $j_1, j_2 \in \mathbb{Z}/N\mathbb{Z}$ such that $\Phi_\ell(j_1, j_2) = 0$ over $\mathbb{Z}/N\mathbb{Z}$, the elliptic curves E_1, E_2, and the kernel polynomial $h(x)$ of an isogeny ϕ from E_1, E_2 can be computed in time polynomial in $\ell, \log(N)$. From the kernel polynomial $h(x)$ of an isogeny ϕ, computing $f(x)$, $g(x, y)$, hence the entire rational polynomial of ϕ, is feasible over $\mathbb{Z}/N\mathbb{Z}$ via Vélu's formulae [39].*

However, it is unclear how to utilize the rational polynomial to solve the (ℓ, ℓ^2)-joint neighbors problem. We postpone further discussions on the hardness and cryptanalysis to Section 5.

4 Trapdoor group with infeasible inversion

In this section we present the construction of the trapdoor group with infeasible inversion. As the general construction is somewhat technical we will present it in

two steps: first we will go over the basic algorithms that feature a simple encoding and composition rule, which suffices for the instantiations of the applications; we will then move to the general algorithms that offer potential optimization and flexibility.

4.1 Definitions

Let us first provide the definition of a TGII, adapted from the original definition in [18,30] to match our construction. The main differences are:

1. The trapdoor in the definition of [18,30] is only used to invert an encoded group element, whereas we assume the trapdoor can be use to encode and decode (which implies the ability of inverting).
2. We classify the encodings of the group elements as *canonical encodings* and *composable encodings*, whereas the definition from [18,30] does not. In our definition, the canonical encoding of an element is uniquely determined once the public parameter is fixed. It can be directly used in the equivalence test, but it does not support efficient group operations. Composable encodings of group elements support efficient group operations. A composable encoding, moreover, can be converted into a canonical encoding by an efficient, public extraction algorithm.

Definition 6. *Let $\mathbb{G} = (\circ, 1_{\mathbb{G}})$ be a finite multiplicative group where $\circ$ denotes the group operator, and $1_{\mathbb{G}}$ denotes the identity. For $x \in \mathbb{G}$, denote its inverse by x^{-1}. $\mathbb{G}$ is associated with the following efficient algorithms:*

Parameter generation. $\mathsf{Gen}(1^\lambda)$ *takes as input the security parameter 1^λ, outputs the public parameter PP and the trapdoor τ.*

Private sampling. $\mathsf{TrapSam}(\mathsf{PP}, \tau, x)$ *takes as inputs the public parameter PP, the trapdoor τ, and a plaintext group element $x \in \mathbb{G}$, outputs a composable encoding $\mathsf{enc}(x)$.*

Composition. $\mathsf{Compose}(\mathsf{PP}, \mathsf{enc}(x), \mathsf{enc}(y))$ *takes as inputs the public parameter PP, two composable encodings $\mathsf{enc}(x), \mathsf{enc}(y)$, outputs $\mathsf{enc}(x \circ y)$. We often use the notation $\mathsf{enc}(x) \circ \mathsf{enc}(y)$ for $\mathsf{Compose}(\mathsf{PP}, \mathsf{enc}(x), \mathsf{enc}(y))$.*

Extraction. $\mathsf{Ext}(\mathsf{PP}, \mathsf{enc}(x))$ *takes as inputs the public parameter PP, a composable encoding $\mathsf{enc}(x)$ of x, outputs the canonical encoding of x as $\mathsf{enc}^*(x)$.*

The hardness of inversion requires that it is infeasible for any efficient algorithm to produce the canonical encoding of x^{-1} given a composable encoding of $x \in \mathbb{G}$.

Hardness of inversion. *For any p.p.t. algorithm A,*

$$\Pr[z = \mathsf{enc}^*(x^{-1}) \mid z \leftarrow A(\mathsf{PP}, \mathsf{enc}(x))] < \mathsf{negl}(\lambda),$$

where the probability is taken over the randomness in the generation of PP, x, $\mathsf{enc}(x)$, and the adversary A.

4.2 Construction details: basic

In this section we provide the formal construction of the TGII with the basic setting of algorithms. The basic setting assumes that in the application of TGII, the encoding sampling algorithm can be stateful, and it is easy to determine which encodings have to be pairwise composable, and which are not. Under these assumptions, we show that we can always sample composable encodings so that the composition always succeeds. That is, the degrees of the any two encodings are chosen to be coprime if they will be composed in the application, and not coprime if they will not be composed. The reader may be wondering why we are distinguishing pairs that are composable and those that are not, as opposed to simply assuming that every pairs of encoding are composable. The reason is for security, meanly due to the parallelogram attack in §5.3.

The basic setting suffices for instantiating the directed transitive signature [18,30] and the broadcast encryption schemes [20], where the master signer and the master encrypter are stateful. We will explain how to determine which encodings are pairwise composable in these two applications, so as to determine the prime degrees of the encodings (the rest of the parameters are not application-specific and follow the universal solution from this section).

For convenience of the reader and for further reference, we provide in Figure 3 a summary of the parameters, with the basic constraints they should satisfy, and whether they are public or hidden. The correctness and efficiency reasons behind these constraints will be detailed in the coming paragraphs, whereas the security reasons will be explained in §5.

Parameter generation. The parameter generation algorithm $\mathsf{Gen}(1^\lambda)$ takes the security parameter 1^λ as input, first chooses a non-maximal order $\mathcal{O}$ of an imaginary quadratic field as follows:

1. Select a square-free negative integer $D_0 \equiv 1 \bmod 4$ as the fundamental discriminant, such that D_0 is polynomially large and $h(D_0)$ is a prime.
2. Choose $k = O(\log(\lambda))$, and a set of distinct polynomially large prime numbers $\{f_i\}_{i\in[k]}$ such that the odd-part of $\left(f_i - \left(\frac{D_0}{f_i}\right)\right)$ is square-free and not divisible by $h(D_0)$. Let $f = \prod_{i\in[k]} f_i$.
3. Set $D = f^2 D_0$. Recall from Eqn. (1) that

$$h(D) = 2 \cdot \frac{h(D_0)}{w(D_0)} \prod_{i\in[k]} \left(f_i - \left(\frac{D_0}{f_i}\right)\right) \tag{3}$$

Let $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ be the odd part of $\mathcal{CL}(\mathcal{O})$, $h(D)_{\text{odd}}$ be largest odd factor of $h(D)$. Note that due to the choices of D_0 and $\{f_i\}$, $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ is cyclic, and we have $|D|, h(D)_{\text{odd}} \in \lambda^{O(\log\lambda)}$. The group with infeasible inversion $\mathbb{G}$ is then $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ with group order $h(D)_{\text{odd}}$.

We then sample the public parameters as follows:

1. Choose two primes p, q, and elliptic curves $E_{0,\mathbb{F}_p}$, $E_{0,\mathbb{F}_q}$ with discriminant D, using the CM method (cf. [25] and more).

Parameters	Basic constraints	Public?
The modulus N	$N = pq$, p, q are primes, $\lvert p\rvert, \lvert q\rvert \in \mathsf{poly}(\lambda)$	Yes
The identity $j(E_0)$	$\mathrm{End}(E_0(\mathbb{F}_p)) \simeq \mathrm{End}(E_0(\mathbb{F}_q)) \simeq \mathcal{O}$	Yes
$\#(E_0(\mathbb{F}_p))$, $\#(E_0(\mathbb{F}_q))$	not polynomially smooth	No
The discriminant D of $\mathcal{O}$	$D = D_0 \cdot f^2$, $D \approx \lambda^{O(\log \lambda)}$, D is poly smooth	No
The class number $h(D)$	follows the choice of D	No
A set S in an encoding:	$S = \{C_i = [(p_i, b_i, \cdot)]\}_{i\in[w]}$ generates $\mathcal{CL}(D)_{\mathrm{odd}}$	See below
The number w of ideals	$w \in O(\log \lambda)$	Yes
The degree p_i of isogenies	$p_i \in \mathsf{poly}(\lambda)$	Yes
The basis $\mathbf{B}$ of Λ_S	$\lVert\tilde{\mathbf{B}}\rVert \in \mathsf{poly}(\lambda)$	No

Fig. 3. Summary of the choices of parameters in the basic setting.

2. Check whether p and q are safe RSA primes (if not, then back to the previous step and restart). Also, check whether the number of points $\#(E_0(\mathbb{F}_p))$, $\#(E_0(\mathbb{F}_q))$, $\#(\tilde{E}_0(\mathbb{F}_p))$, $\#(\tilde{E}_0(\mathbb{F}_q))$ (where $\tilde{E}$ denotes the quadratic twist of E) are polynomially smooth (if yes, then back to the previous step and restart). p, q and the number of points should be hidden for security.
3. Set the modulus N as $N := p \cdot q$ and let $j_0 = \mathsf{CRT}(p, q; j(E_{0,\mathbb{F}_p}), j(E_{0,\mathbb{F}_q}))$. Let j_0 represent the identity of $\mathbb{G}$.

Output (N, j_0) as the public parameter PP. Keep (D, p, q) as the trapdoor τ (D and the group order of $\mathbb{G}$ should be hidden for security).

The sampling algorithm and the group operation of the composable encodings. Next we provide the definitions and the algorithms for the composable encoding.

Definition 7 (Composable encoding). *Given a factorization of x as $\prod_{i=1}^{w} C_i^{e_i}$, where $w \in O(\log \lambda)$; $C_i = [(p_i, b_i, \cdot)] \in \mathbb{G}$, $e_i \in \mathbb{N}$, for $i \in [w]$. A composable encoding of $x \in \mathbb{G}$ is represented by*

$$\mathsf{enc}(x) = (L; T_1, ..., T_w) = ((p_1, ..., p_w); (j_{1,1}, ..., j_{1,e_1}), ..., (j_{w,1}, ..., j_{w,e_w})),$$

where all the primes in the list $L = (p_1, ..., p_w)$ are distinct; for each $i \in [w]$, $T_i \in (\mathbb{Z}/N\mathbb{Z})^{e_i}$ is a list of the j-invariants such that $j_{i,k} = C_i^k * j_0$, for $k \in [e_i]$.

The degree of an encoding $\mathsf{enc}(x)$ is defined to be $d(\mathsf{enc}(x)) := \prod_{i=1}^{w} p_i^{e_i}$.

Notice that the factorization of $x = \prod_{i=1}^{w} C_i^{e_i}$ has to satisfy $e_i \in \mathsf{poly}(\lambda)$, for all $i \in [w]$, so as to ensure the length of $\mathsf{enc}(x)$ is polynomial. Looking ahead, we also require each p_i, the degree of the isogeny that represents the C_i-action, to be polynomially large so as to ensure Algorithm 3 in the encoding sampling algorithm and Algorithm 6 in the extraction algorithm run in polynomial time.

The composable encoding sampling algorithm requires the following subroutine:

Algorithm 3 $\mathsf{act}(\tau, j, C)$ *takes as input the trapdoor $\tau = (D, p, q)$, a j-invariant $j \in \mathbb{Z}/N\mathbb{Z}$, and an ideal class $C \in \mathcal{CL}(\mathcal{O})$, proceeds as follows:*

1. *Let* $j_p = j \mod p$, $j_q = j \mod q$.
2. *Compute* $j'_p := C * j_p \in \mathbb{F}_p$, $j'_q := C * j_q \in \mathbb{F}_q$.
3. *Output* $j' := \mathsf{CRT}(p, q; j'_p, j'_q)$.

Algorithm 4 (Sample a composable encoding) *Given as input the public parameter* $\mathsf{PP} = (N, j_0)$*, the trapdoor* $\tau = (D, p, q)$*, and* $x \in \mathbb{G}$*,* $\mathsf{TrapSam}(\mathsf{PP}, \tau, x)$ *produces a composable encoding of* x *is sampled as follows:*

1. *Choose* $w \in O(\log \lambda)$ *and a generation set* $S = \{C_i = [(p_i, b_i, \cdot)]\}_{i \in [w]} \subset \mathbb{G}$.
2. *Sample a short basis* $\mathbf{B}$ *(in the sense that* $\|\tilde{\mathbf{B}}\| \in \mathsf{poly}(\lambda)$*) for the relation lattice* Λ_S*:*
$$\Lambda_S := \left\{ \mathbf{y} \mid \mathbf{y} \in \mathbb{Z}^w, \prod_{i \in [w]} C_i^{y_i} = 1_{\mathbb{G}} \right\}. \tag{4}$$
3. *Given* x*,* S*,* $\mathbf{B}$*, sample a short vector* $\mathbf{e} \in \{\mathsf{poly}(\lambda) \cap \mathbb{N}\}^w$ *such that* $x = \prod_{i \in [w]} C_i^{e_i}$.
4. *For all* $i \in [w]$*:*
   (a) *Let* $j_{i,0} := j_0$.
   (b) *For* $k = 1$ *to* e_i*: compute* $j_{i,k} := \mathsf{act}(\tau, j_{i,k-1}, C_i)$.
   (c) *Let* $T_i := (j_{i,1}, ..., j_{i,e_i})$.
5. *Let* $L \in \mathbb{N}^w$ *be a list where the* i^{th} *entry of* L *is* p_i.
6. *Output the composable encoding of* x *as*
$$\mathsf{enc}(x) = (L; T_1, ..., T_w) = ((p_1, ..., p_w); (j_{1,1}, ..., j_{1,e_1}), ..., (j_{w,1}, ..., j_{w,e_w})).$$

Remark 2 (Thinking of each adjacent pair of j*-invariants as an isogeny).* In each T_i, each adjacent pair of the j-invariants can be thought of representing an isogeny ϕ that corresponds to the ideal class $C_i = [(p_i, b_i, \cdot)]$. Over the finite field, C_i can be explicitly recovered from an adjacent pair of the j-invariants and p_i (cf. Remark 1). Over $\mathbb{Z}/N\mathbb{Z}$, the rational polynomial of the isogeny ϕ can be recovered from the adjacent pair of the j-invariants and p_i (cf. Proposition 1), but it is not clear how to recover b_i in the binary quadratic form representation of C_i.

Remark 3 (The only stateful step in the sampling algorithm). Recall that the basic setting assumes the encoding algorithm is stateful, where the state records the prime factors of the degrees used in the existing composable encodings. The state is only used in the first step to choose the $\{p_i\}$ of the ideals in the generation set $S = \{C_i = [(p_i, b_i, \cdot)]\}_{i \in [w]}$.

Group operations. Given two composable encodings, the group operation is done by simply concatenating the encodings if their degrees are coprime, or otherwise outputting "failure".

Algorithm 5 *The encoding composition algorithm* $\mathsf{Compose}(\mathsf{PP}, \mathsf{enc}(x), \mathsf{enc}(y))$ *parses* $\mathsf{enc}(x) = (L_x; T_{x,1}, ..., T_{x,w_x})$*,* $\mathsf{enc}(y) = (L_y; T_{y,1}, ..., T_{y,w_y})$*, produces the composable encoding of* $x \circ y$ *as follows:*

- *If* $\gcd(d(\mathsf{enc}(x)), d(\mathsf{enc}(y))) = 1$*, then output the composable encoding of* $x \circ y$ *as*
$$\mathsf{enc}(x \circ y) = (L_x || L_y; T_{x,1}, ..., T_{x,w_x}, T_{y,1}, ..., T_{y,w_y}).$$
- *If* $\gcd(d(\mathsf{enc}(x)), d(\mathsf{enc}(y))) > 1$*, output "failure".*

The canonical encoding and the extraction algorithm.

Definition 8 (Canonical encoding). *The canonical encoding of* $x \in \mathbb{G}$ *is* $x * j_0 \in \mathbb{Z}/N\mathbb{Z}$*.*

The canonical encoding of x can be computed by first obtaining a composable encoding of x, and then converting the composable encoding into the canonical encoding using the extraction algorithm. The extraction algorithm requires the following subroutine.

Algorithm 6 (The "gcd" operation) *The algorithm* $\mathsf{gcd.op}(\mathsf{PP}, \ell_1, \ell_2; j_1, j_2)$ *takes as input the public parameter* PP*, two degrees* ℓ_1, ℓ_2 *and two* j*-invariants* j_1, j_2*, proceeds as follows:*

- *If* $\gcd(\ell_1, \ell_2) = 1$*, then it computes* $f(x) = \gcd(\Phi_{\ell_2}(j_1, x), \Phi_{\ell_1}(j_2, x))$ *over* $\mathbb{Z}/N\mathbb{Z}$*, and outputs the only root of* $f(x)$*;*
- *If* $\gcd(\ell_1, \ell_2) > 1$*, it outputs* $\perp$*.*

Algorithm 7 $\mathsf{Ext}(\mathsf{PP}, \mathsf{enc}(x))$ *converts the composable encoding* $\mathsf{enc}(x)$ *into the canonical encoding* $\mathsf{enc}^*(x)$*. The algorithm maintains a pair of lists* (U, V)*, where* U *stores a list of* j*-invariants* $(j_1, ..., j_{|U|})$*,* V *stores a list of degrees where the* i^{th} *entry of* V *is the degree of isogeny between* j_i *and* j_{i-1} *(when* $i = 1$*,* j_{i-1} *is the* j_0 *in the public parameter). The lengths of* U *and* V *are always equal during the execution of the algorithm.*

The algorithm parses $\mathsf{enc}(x) = (L; T_1, ..., T_w)$*, proceeds as follows:*

1. *Initialization: Let* $U := T_1$*,* $V := (L_1, ..., L_1)$ *of length* $|T_1|$ *(i.e. copy* L_1 *for* $|T_1|$ *times).*
2. *For* $i = 2$ *to* w*:*
   (a) *Set* $u_{\mathsf{temp}} := |U|$*.*
   (b) *For* $k = 1$ *to* $|T_i|$*:*
      i. *Let* $t_{i,k,0}$ *be the* k^{th} j*-invariant in* T_i*, i.e.* $j_{i,k}$*;*
      ii. *For* $h = 1$ *to* u_{temp}*:*
         – *If* $k = 1$*, compute* $t_{i,k,h} := \mathsf{gcd.op}(\mathsf{PP}, L_i, V_h; t_{i,k,h-1}, U_h)$*;*
         – *If* $k > 1$*, compute* $t_{i,k,h} := \mathsf{gcd.op}(\mathsf{PP}, L_i, V_h; t_{i,k,h-1}, t_{i,k-1,h})$*;*
      iii. *Append* $t_{i,k,u_{\mathsf{temp}}}$ *to the list* U*, append* L_i *to the list* V*.*
3. *Return the last entry of* U*.*

Example 1. Let us give a simple example for the composition and the extraction algorithms. Let ℓ, m, n be three distinct polynomially large primes. Let the composable encoding of an element y be $\mathsf{enc}(y) = ((\ell); (j_{1,1}, j_{1,2}, j_{1,3}))$, based on

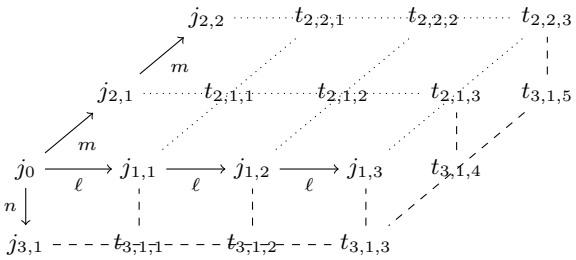

Fig. 4. An example for the composable encoding and the extraction algorithm.

the factorization of $y = C_1^{e_1} = [(\ell, b_\ell, \cdot)]^3$. Let the composable encoding of an element z be $\mathsf{enc}(z) = ((m, n); (j_{2,1}, j_{2,2}), (j_{3,1}))$, based on the factorization of $z = C_2^{e_2} \cdot C_3^{e_3} = [(m, b_m, \cdot)]^2 \cdot [(n, b_n, \cdot)]^1$. Then the composable encoding of $x = y \circ z$ obtained from Algorithm 5 is $\mathsf{enc}(x) = ((\ell, m, n); (j_{1,1}, j_{1,2}, j_{1,3}), (j_{2,1}, j_{2,2}), (j_{3,1}))$.

Next we explain how to extract the canonical encoding of x from $\mathsf{enc}(x)$. In Figure 4, the j-invariants in $\mathsf{enc}(x)$ are placed on the solid arrows (their positions do not follow the relative positions on the volcano). We can think of each gcd operation in Algorithm 6 as fulfilling a missing vertex of a parallelogram defined by three existing vertices.

When running $\mathsf{Ext}(\mathsf{PP}, \mathsf{enc}(x))$, the list U is initialized as $(j_{1,1}, j_{1,2}, j_{1,3})$, the list V is initialized as (ℓ, ℓ, ℓ). Let us go through the algorithm for $i = 2$ and $i = 3$ in the second step.

- When $i = 2$, u_{temp} equals to $|U| = 3$. The j-invariants $\{t_{2,k,h}\}_{k \in [|T_2|], h \in [u_{\mathsf{temp}}]}$ are placed on the dotted lines, computed in the order of $t_{2,1,1}$, $t_{2,1,2}$, $t_{2,1,3}$, $t_{2,2,1}$, $t_{2,2,2}$, $t_{2,2,3}$. The list U is updated to $(j_{1,1}, j_{1,2}, j_{1,3}, t_{2,1,3}, t_{2,2,3})$, the list V is updated to (ℓ, ℓ, ℓ, m, m)
- When $i = 3$, u_{temp} equals to $|U| = 5$. The j-invariants $\{t_{3,1,h}\}_{h \in [u_{\mathsf{temp}}]}$ are placed on the dashed lines, computed in the order of $t_{3,1,1}$, ..., $t_{3,1,5}$. In the end, $t_{3,1,5}$ is appended to U, n is appended to V.

The canonical encoding of x is then $t_{3,1,5}$.

On correctness and efficiency. We now verify the correctness and efficiency of the parameter generation, encoding sampling, composition, and the extraction algorithms.

To begin with, we verify that the canonical encoding correctly and uniquely determines the group element in $\mathcal{CL}(\mathcal{O})$. It follows from the choices of the elliptic curves $E_0(\mathbb{F}_p)$ and $E_0(\mathbb{F}_q)$ with $\mathrm{End}(E_0(\mathbb{F}_p)) \simeq \mathrm{End}(E_0(\mathbb{F}_q)) \simeq \mathcal{O}$, and the following bijection once we fix E_0:

$$\mathcal{CL}(\mathcal{O}) \to \mathrm{Ell}_{\mathcal{O}}(\mathbf{k}), \quad x \mapsto x * j(E_0(\mathbf{k})), \text{ for } \mathbf{k} \in \{\mathbb{F}_p, \mathbb{F}_q\}$$

Next, we will show that generating the parameters, i.e. the curves $E_{0,\mathbb{F}_p}$, $E_{0,\mathbb{F}_q}$ with a given fundamental discriminant D_0 and a conductor $f = \prod_i^k f_i$,

is efficient when $|D_0|$ and all the factors of f are of polynomial size. Let u be an integer such that $f \mid u$. Choose a p and t_p such that $t_p^2 - 4p = u^2 D_0$. Then, compute the Hilbert class polynomial H_{D_0} over $\mathbb{F}_p$ and find one of its roots j. From j, descending on the volcanoes $G_{f_i}(\mathbb{F}_p)$ for every f_i gives the j-invariant for the curve with desired discriminant. The same construction works verbatim for q.

We then show that sampling the composable encodings can be done in polynomial time heuristically:

1. Given a logarithmically large set $S = \{C_i = [(\ell_i, b_i, \cdot)] \in \mathcal{CL}(\mathcal{O})\}_{i\in[w]}$, a possibly big basis of the relation lattice Λ_S can be obtained by solving the discrete-log problem over $\mathcal{CL}(\mathcal{O})$, which can be done in polynomial time since the group order is polynomially smooth.
2. Suppose that the lattice Λ_S satisfies the Gaussian heuristic (this is the only heuristic we assume). That is, for all $1 \leq i \leq w$, the i^{th} successive minimum of Λ_S, denoted as λ_i, satisfies $\lambda_i \approx \sqrt{w} \cdot h(\mathcal{O})^{1/w} \in \mathsf{poly}(\lambda)$. Since $w = O(\log(\lambda))$, the short basis $\mathbf{B}$ of Λ_S, produced by the LLL algorithm, satisfies $\|\mathbf{B}\| \leq 2^{\frac{w}{2}} \cdot \lambda_w \in \mathsf{poly}(\lambda)$.
3. Given a target group element $x \in \mathcal{CL}(\mathcal{O})$, the polynomially short basis $\mathbf{B}$, we can sample a vector $\mathbf{e} \in \mathbb{N}^w$ such that $\prod_{i=1}^m C_i^{e_i} = x$ and $\|\mathbf{e}\|_1 \in \mathsf{poly}(\lambda)$ in polynomial time using e.g. Babai's algorithm [1]. (In §5.3, we will explain that the GPV sampler [17] is preferred for the security purpose.)
4. The unit operation $\mathsf{act}(\tau, j, C)$ is efficient when the ideal class C corresponds to a polynomial degree isogeny, since it is efficient to compute polynomial degree isogenies over the finite fields.
5. The length of the final output $\mathsf{enc}(x)$ is $(w + \|\mathbf{e}\|_1) \cdot \mathsf{poly}(\lambda) \in \mathsf{poly}(\lambda)$.

The algorithm $\mathsf{Compose}(\mathsf{PP}, \mathsf{enc}(x), \mathsf{enc}(y))$ concatenates $\mathsf{enc}(x)$, $\mathsf{enc}(y)$, so it is efficient as long as $\mathsf{enc}(x)$, $\mathsf{enc}(y)$ are of polynomial size.

The correctness of the unit operation $\mathsf{gcd.op}$ follows the commutativity of the endomorphism ring $\mathcal{O}$. The operation $\mathsf{gcd.op}(\mathsf{PP}, \ell_1, \ell_2; j_1, j_2)$ is efficient when $\gcd(\ell_1, \ell_2) = 1$, $\ell_1, \ell_2 \in \mathsf{poly}(\lambda)$, given that solving the (ℓ_1, ℓ_2) isogenous neighbor problem over $\mathbb{Z}/N\mathbb{Z}$ is efficient under these conditions.

When applying $\mathsf{Ext}()$ (Algorithm 7) on a composable encoding $\mathsf{enc}(x) = (L_x; T_{x,1}, ..., T_{x,w_x})$, it runs $\mathsf{gcd.op}$ for $\max_{i=1}^{w_x} |T_{x_i}| \cdot (\sum_{i=1}^{w_x} |T_{x_i}|)$ times. So obtaining the canonical encoding is efficient as long as all the primes in L_x are polynomially large, and $|T_{x,i}| \in \mathsf{poly}(\lambda)$ for all $i \in [w_x]$.

5 Cryptanalysis

We provide a highlight of the cryptanalytic attempts we have made and discuss the impacts and the countermeasures. The details of our cryptanalysis attempts can be found in the full version.

The security of our cryptosystem relies on the conjectured hardness of solving various problems over $\mathbb{Z}/N\mathbb{Z}$ without knowing the factors of N. So we start from the feasibility of performing several individual computational tasks over $\mathbb{Z}/N\mathbb{Z}$;

then focus on the (ℓ, ℓ^2)-isogenous neighbor problem over $\mathbb{Z}/N\mathbb{Z}$, whose hardness is necessary for the security of our candidate TGII; finally address all the other attacks in the TGII construction.

5.1 The (in)feasibility of performing computations over $\mathbb{Z}/N\mathbb{Z}$

Factoring polynomials over $\mathbb{Z}/N\mathbb{Z}$. The task of finding roots of polynomials of degree $d \geq 2$ over $\mathbb{Z}/N\mathbb{Z}$ sits in the subroutines of many potential algorithms we need to consider, so let us begin with a clarification on the status of this problem. No polynomial time algorithm is known to solve this problem in general. In a few special cases, finding at least one root is feasible. For example, if a root of a polynomial over $\mathbb{Z}/N\mathbb{Z}$ is known to be the same as the root over $\mathbb{Q}$, then we can use LLL [26]; or if a root is known to be smaller than roughly $O(N^{1/d})$, then Coppersmith-type algorithms can be used to find such a root [6]. However, these families of polynomials only form a negligible portion of all the polynomials with polynomially bounded degrees.

Feasible information from a single j-invariant From any $j \in \mathbb{Z}/N\mathbb{Z}, j \neq 0, 1728$, we can find the coefficients a and b of the Weierstrass form of an elliptic curve $E(\mathbb{Z}/N\mathbb{Z})$ with $j(E) = j$ by computing $a = 3j(1728-j), b = 2j(1728-j)^2$. But choosing a curve over $\mathbb{Z}/N\mathbb{Z}$ with a given j-invariant together with a point on the curve seems tricky. Nevertheless, it is always feasible to choose a curve together with the x-coordinate of a point on it, since a random $x \in \mathbb{Z}/N\mathbb{Z}$ is the x-coordinate of some point on the curve with probability roughly $\frac{1}{2}$. It is also known that computing the multiples of a point P over $E(\mathbb{Z}/N\mathbb{Z})$ is feasible solely using the x-coordinate of P (cf. [10]). The implication of this is that we should at the very least not give out the group orders of the curves involved in the scheme. More precisely, we should avoid the j-invariants corresponding to curves (or their twists) with polynomially smooth cardinalities over either $\mathbb{F}_p$ or $\mathbb{F}_q$. Otherwise Lenstra's algorithm [27] can be used to factorize N.

In our application we also assume that the endomorphism rings of $E(\mathbb{F}_p)$ and $E(\mathbb{F}_q)$ are isomorphic and not given out (the reason will be explained later). Computing the discriminant of $\mathcal{O} \simeq \mathrm{End}(E(\mathbb{F}_p)) \simeq \mathrm{End}(E(\mathbb{F}_q))$ or the number of points of E over $\mathbb{Z}/N\mathbb{Z}$ seems to be hard given only N and a j-invariant. In fact Kunihiro and Koyama (and others) have reduced factorizing N to computing the number of points of general elliptic curves over $\mathbb{Z}/N\mathbb{Z}$ [24]. However, these reductions are not efficient in the special case, where the endomorphism rings of $E(\mathbb{F}_p)$ and $E(\mathbb{F}_q)$ are required to be isomorphic. So, the result of [24] can be viewed as evidence that the polynomial time algorithms for counting points on elliptic curves over finite fields may fail over $\mathbb{Z}/N\mathbb{Z}$ without making use of the fact that the endomorphism rings of $E(\mathbb{F}_p)$ and $E(\mathbb{F}_q)$ are isomorphic.

Let ℓ be a prime. We will be concerned with degree ℓ isogenies. If we are only given a single j-invariant $j_1 \in \mathbb{Z}/N\mathbb{Z}$, then finding an integer j_2 such that $\Phi_\ell(j_1, j_2) = 0 \pmod N$ seems hard. Nevertheless, we remark that Theorem 1 does not guarantee that finding j_2 is as hard as factoring when the endomorphism

rings of $E(\mathbb{F}_p)$ and $E(\mathbb{F}_q)$ are isomorphic. However, as of now, we do not know how to make use of the condition that the endomorphism rings are isomorphic to mount an attack on the problem.

Feasible information from more j-invariants In the construction of a TGII we are not only given a single j-invariant, but many j-invariants with each neighboring pair of them satisfying the ℓ^{th} modular polynomial, a polynomial degree $\ell + 1$. We will study what other information can be extracted from these neighboring j-invariants.

In Proposition 1, we have explained that given two integers $j_1, j_2 \in \mathbb{Z}/N\mathbb{Z}$ such that $\Phi_\ell(j_1, j_2) = 0$ over $\mathbb{Z}/N\mathbb{Z}$, the elliptic curves E_1, E_2, and the kernel polynomial $h(x)$ of an isogeny ϕ from E_1, E_2 can be computed in time polynomial in $\ell, \log(N)$. However, it is not clear how to use the explicit expression of ϕ to break factoring or solve the inversion problem.

A natural next step is to recover a point in the kernel of ϕ, but it is also not clear how to recover even the x-coordinate of a point in the kernel when $\ell \geq 5$. For $\ell = 3$, on the other hand, the kernel polynomial does reveal the x-coordinate of a point P in the kernel $G \subset E_1[3]$ (note that $h(\cdot)$ is of degree 1 in this particular case). But revealing the x-coordinate of a point $P \in E_1[3]$ does not immediately break factoring, since $3P$ is O over both $\mathbb{F}_p$ and $\mathbb{F}_q$. At this moment we do not know of a full attack from a point in $\ker(\phi)$. Nevertheless, we still choose to take an additional safeguard by avoiding the use of 3-isogenies since it reveals the x-coordinate of a point in $E_1[3]$, and many operations on elliptic curves are feasible given the x-coordinate of a point.

5.2 Tackling the (ℓ, ℓ^2)-isogenous neighbor problem over $\mathbb{Z}/N\mathbb{Z}$

The (ℓ, ℓ^2)-isogenous neighbor problem is essential to the hardness of inversion in our TGII construction. In addition to Definition 5, we assume that the endomorphism rings of the curves in the problem are isomorphic to an imaginary quadratic order $\mathcal{O}$.

The Hilbert class polynomial attack We first note that the discriminant D of the underlying endomorphism ring $\mathcal{O}$ cannot be polynomially large, otherwise we can compute the Hilbert class polynomial H_D in polynomial time and therefore solve the (ℓ, ℓ^2)-isogenous neighbor problem. Given j_0, j_1 such that $\Phi_\ell(j_0, j_1) = 0$, compute the polynomial $\gamma(x)$,

$$\gamma(x) := \gcd(\Phi_\ell(j_0, x), \Phi_{\ell^2}(j_1, x), H_D(x)) \in (\mathbb{Z}/N\mathbb{Z})[x].$$

The gcd of $\Phi_\ell(j_0, x)$ and $\Phi_{\ell^2}(j_1, x)$ gives a polynomial of degree ℓ. The potential root they share with $H_D(x)$ is the only one with the same endomorphism ring with j_0 and j_1, which is j_{-1}. So $\gamma(x)$ is a linear function.

Survey of the Ionica-Joux algorithm Among the potential solutions to the (ℓ, ℓ^2)-isogenous neighbor problem, finding the one corresponding to the image of a horizontal isogeny would break our candidate group with infeasible inversion, so it is worth investigating algorithms which find isogenies with specific directions. However, the only known such algorithm over the finite fields, that of Ionica and Joux [19], does not seem to work over $\mathbb{Z}/N\mathbb{Z}$. In the full version we provide a detailed survey of this algorithm.

More about modular curves and characteristic zero attacks Given j, solving $\Phi_\ell(j, x)$ is not the only way to find the j-invariants of the ℓ-isogenous curves. Alternative complex analytic (i.e. characteristic zero) methods have been discussed, for instance, in [12, Section 3]. However, these methods all involve solving polynomials of degree ≥ 2 to get started.

As mentioned in Section 2.2, the curve $\mathbb{H}/\Gamma_0(\ell)$ parameterizes pairs of elliptic curves over $\mathbb{C}$ related by a cyclic ℓ-isogeny. The (ℓ, ℓ^2)-isogenous neighbor problem, on the other hand, concerns curves that are horizontally ℓ-isogenous, i.e. ℓ-isogenous and have the same endomorphism ring. To avoid an attack through characteristic zero techniques, we make sure that there is no immediate quotient of $\mathbb{H}$ that parametrizes curves which are related with an ℓ-isogeny and have the same endomorphism ring. Below, we first go over the well-known moduli description of modular curves to make sure that they don't lead to an immediate attack, and then show that there is indeed no quotient of $\mathbb{H}$ between $\mathbb{H}/\mathrm{SL}_2(\mathbb{Z})$ and $\mathbb{H}/\Gamma_0(\ell)$, so we don't have to worry about possible attacks on that end.

Let $\Gamma := \mathrm{SL}_2(\mathbb{Z})$, and let $\Gamma(\ell)$ and $\Gamma_1(\ell)$ denote the congruence subgroups,

$$\Gamma(\ell) := \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) \middle| \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \pmod{\ell} \right\},$$

$$\Gamma_1(\ell) := \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) \middle| \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \pmod{\ell} \right\}.$$

It is well-known that the curves $\mathbb{H}/\Gamma_1(\ell)$ and $\mathbb{H}/\Gamma(\ell)$ parametrize elliptic curves with extra data on their ℓ-torsion (cf. [23]). $\mathbb{H}/\Gamma_1(\ell)$ parametrizes (E, P), where P is a point on E having order exactly ℓ, and $\mathbb{H}/\Gamma(\ell)$ parametrizes triples (E, P, Q), where $E[\ell] = \langle P, Q \rangle$ and they have a fixed Weil pairing. These curves carry more information than the ℓ-isogenous relation and they are not immediately helpful for solving the (ℓ, ℓ^2)-isogenous neighbor problem.

As for the quotients between $\mathbb{H}/\mathrm{SL}_2(\mathbb{Z})$ and $\mathbb{H}/\Gamma_0(\ell)$, the following lemma shows that there are indeed none.

Lemma 4. *Let ℓ be a prime. If $H \leq \Gamma$ is such that $\Gamma_0(\ell) \leq H \leq \Gamma$, then either $H = \Gamma_0(\ell)$ or $H = \Gamma$.*

Proof. Let $\sigma_1 = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, $\sigma_2 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$, $\sigma_3 = \sigma_1 \sigma_2^{-1}$, and recall that $SL_2(\mathbb{Z}/\ell\mathbb{Z}) = \langle \sigma_1, \sigma_2 \rangle = \langle \sigma_1, \sigma_3 \rangle$. Recall that the natural projection $\pi : \Gamma \to SL_2(\mathbb{Z}/\ell\mathbb{Z})$ is surjective. Assume that $H \neq \Gamma_0(\ell)$. This implies that $\pi(H) = SL_2(\mathbb{Z}/\ell\mathbb{Z})$ (we shall give a proof below). Assuming this claim for the moment let $g \in \Gamma \backslash H$.

Since $\pi(\Gamma) = \pi(H)$ there exists $h \in H$ such that $\pi(g) = \pi(h)$. Therefore, $gh^{-1} \in \ker(\pi) = \Gamma(\ell) \subset H$. Therefore, $g \in H$ and $\Gamma = H$.

To see that $\pi(\Gamma) = \pi(H)$, first note that since $\Gamma_0(\ell) \subset H$ we have all the upper triangular matrices in $\pi(H)$. Next, let $h = \left(\begin{smallmatrix} h_1 & h_2 \\ h_3 & h_4 \end{smallmatrix}\right) \in H \backslash \Gamma_0(\ell)$ such that $\pi(h) = \left(\begin{smallmatrix} \bar{h}_1 & \bar{h}_2 \\ \bar{h}_3 & \bar{h}_4 \end{smallmatrix}\right) \in \pi(H) \backslash \pi(\Gamma_0(\ell))$ (note that this difference is non-empty since otherwise $\Gamma_0(\ell) = H$).
We have two cases depending on $\bar{h}_1 = 0$ or not. If $\bar{h}_1 = 0$ then $\bar{h}_3 \neq 0$ and $\sigma_3 = \left(\begin{smallmatrix} \bar{h}_3^{-1} & \bar{h}_4 \\ 0 & \bar{h}_3 \end{smallmatrix}\right) \bar{h}^{-1} \in \pi(H)$. On the other hand, if $\bar{h}_1 \neq 0$ multiplying on the right by $\left(\begin{smallmatrix} \bar{h}_1^{-1} & -\bar{h}_2 \\ 0 & \bar{h}_1 \end{smallmatrix}\right) \in \pi(H)$ we see that $\left(\begin{smallmatrix} 1 & 0 \\ \bar{h}_3 \bar{h}_1^{-1} & 1 \end{smallmatrix}\right) \in \pi(H)$. For any integer m, the m'th power of this matrix is $\left(\begin{smallmatrix} 1 & 0 \\ m\bar{h}_3 \bar{h}_1^{-1} & 1 \end{smallmatrix}\right) \in \pi(H)$. Taking $m \equiv \bar{h}_1 \bar{h}_3^{-1}$ shows that $\sigma_2 \in \pi(H)$. This shows that $\pi(H) = SL_2(\mathbb{Z}/\ell\mathbb{Z})$.

5.3 Cryptanalysis of the candidate group with infeasible inversion

We now cryptanalyze the concrete candidate TGII. Recall the format of an encoding of a group element x from Definition 7:

$$\begin{aligned} \mathsf{enc}(x) =& (L_x; T_{x,1}, ..., T_{x,w_x}) \\ =& ((p_{x,1}, ..., p_{x,w_x}); (j_{x,1,1}, ..., j_{x,1,e_{x,1}}), ..., (j_{x,w_x,1}, ..., j_{x,w_x,e_{x,w_x}})). \end{aligned}$$

The "exponent vector" $\mathbf{e}_x \in \mathbb{Z}^{w_x}$ can be read from the encoding as $\mathbf{e}_x = (|T_{x,1}|, ..., |T_{x,w_x}|)$.

We assume polynomially many composable encodings are published in the applications of a TGII. In down-to-earth terms it means the adversary is presented with polynomially many j-invariants on the crater of a volcano, and the explicit isogenies between each pair of the neighboring j-invariants (due to Proposition 1).

We will be considering the following model on the adversary's attacking strategy.

Definition 9 (The GCD attack model). *In the GCD attack model, the adversary is allowed to try to find the inverse of a target group element only by executing the unit gcd operation* $\mathsf{gcd.op}(\mathsf{PP}, \ell_1, \ell_2; j_1, j_2)$ *given in Algorithm 6 for polynomially many steps, where* $\ell_1, \ell_2; j_1, j_2$ *are from the published encodings or obtained from the previous executions of the gcd evaluations.*

We do not know how to prove the construction of TGII is secure even if the adversary is restricted to attack in the GCD model. Our cryptanalysis attempts can be classified as showing (1) how to prevent the attacks that obey the GCD evaluation law; (2) how to prevent the other attack approaches (by e.g. guessing the class group invariants).

Preventing the trivial leakage of inverses In applications we are often required to publish the encodings of elements that are related in some way. A

typical case is the following: for $x, y \in \mathcal{CL}(\mathcal{O})$, the scheme may require publishing the encodings of x and $z = y \circ x^{-1}$ without revealing a valid encoding of x^{-1}. As a toy example, let $x = [(p_x, b_x, \cdot)]$, $y = [(p_y, b_y, \cdot)]$, where p_x and p_y are distinct primes. Let j_0, the j-invariant of a curve E_0, represent the identity element in the public parameter. Let $((p_x); (j_x))$ be a composable encoding of x and $((p_y); (j_y))$ be a composable encoding of y.

Naively, a composable encoding of $z = y \circ x^{-1}$ could be $((p_x, p_y); (j_{x^{-1}}), (j_y))$, where $j_{x^{-1}}$ is the j-invariant of $E_{x^{-1}} = x^{-1}E_0$. Note, however, that $((p_x); (j_{x^{-1}}))$ is a valid encoding of x^{-1}. In other words such an encoding of $y \circ x^{-1}$ trivially reveals the encoding of x^{-1}.

One way of generating an encoding of $z = y \circ x^{-1}$ without trivially revealing $j_{x^{-1}}$ is to first pick a generator set of ideals where the norms of the ideals are coprime to p_x and p_y, then solve the discrete-log of z over these generators to compute the composable encoding. This is the approach we take in this paper.

Parallelogram attack In the applications we are often required to publish the composable encodings of group elements a, b, c such that $a \circ b = c$. If the degrees of the three encodings are coprime, then we can recover the encodings of a^{-1}, b^{-1}, and c^{-1} using the following "parallelogram attack". This is a non-trivial attack which obeys the gcd evaluation law in Definition 9.

Let us illustrate the attack via the examples in Figure 5, where the solid arrows represent the isogenies that are given as the inputs (the j-invariants of the target curves are written at the head of the arrows, their positions do not follow the relative positions on the volcano; the degree of the isogeny is written on the arrow); the dashed lines and the j-invariants on those lines are obtained from the gcd evaluation law.

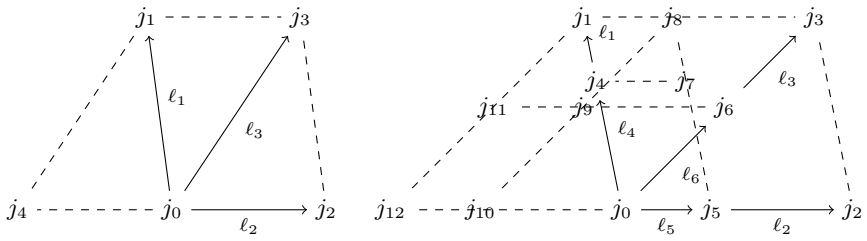

Fig. 5. The parallelogram attack.

For simplicity let us first look at the example in the left figure. Let composable encodings of a, b, c be given by $(\ell_1; (j_1))$, $(\ell_2; (j_2))$, $(\ell_3; (j_3))$, where ℓ_1, ℓ_2, ℓ_3 are polynomial and pairwise coprime. A composable encoding of b^{-1} then can be written as $(\ell_2; (j_4))$, where j_4 is the root of the linear equation $f(x) = \gcd(\Phi_{\ell_3}(j_1, x), \Phi_{\ell_2}(j_0, x))$. This is due to the relation $c \circ b^{-1} = a$, which, in particular implies that j_1 and j_4 are connected by an isogeny of degree ℓ_3.

The simple attack above uses the fact that the degrees of the entire encodings of a, b and c are polynomial. Let us use the example in the right figure to illustrate that even if the encodings are composed of many polynomial degree isogenies (so that the total degrees may be super-polynomial), the attack is still effective. The idea is to view the composition law as filling the missing edges of a parallelogram given the j-invariants on a pair of adjacent edges. The final goal is to find the missing corner j_{12} in the parallelogram $j_0 - j_3 - j_1 - j_{12}$. To arrive there we need the j-invariants on a pair of adjacent edges to begin with, so we first have to fill the j-invariants on, for instance, the edge $j_1 - j_3$. Therefore, as the first step, we consider the parallelogram $j_0 - j_2 - j_3 - j_1$. To fill the j-invariants on the edge $j_1 - j_3$, we first compute j_7 as the root of $f_7(x) = \gcd(\Phi_{\ell_4}(j_5, x), \Phi_{\ell_5}(j_4, x))$, then compute j_8 as the root of $f_8(x) = \gcd(\Phi_{\ell_1}(j_7, x), \Phi_{\ell_5}(j_1, x))$ (the polynomials f_7, f_8 are linear since the degrees of $\mathsf{enc}(a)$ and $\mathsf{enc}(b)$ are coprime). In the second step, we consider the parallelogram $j_0 - j_3 - j_1 - j_{12}$. To find j_{12} we use the gcd evaluation law to find j_9, j_{10}, j_{11}, j_{12} one-by-one (using the condition that the degrees of $\mathsf{enc}(c)$ and $\mathsf{enc}(b)$ are coprime).

The parallelogram attack is very powerful, in the sense that it is not preventable when application requires to publish the composable encodings of a, b, c such that $a \circ b = c$, and $\mathsf{enc}(a)$, $\mathsf{enc}(b)$, $\mathsf{enc}(c)$ to be pairwise composable. However, the parallelogram attack does not seem to work when 2 out of the 3 pairs of $\mathsf{enc}(a)$, $\mathsf{enc}(b)$ and $\mathsf{enc}(c)$ are not composable. In the applications of directed transitive signature and broadcast encryption, there are encodings of a, b, c such that $a \circ b = c$. Luckily, only one pair of the encodings among the three has to be composable to provide the necessary functionalities of these applications.

Hiding the class group invariants In the applications of TGII, it is reasonable to assume that the closure of the gcd-compositions of the published j-invariants covers all the $h(D)$ j-invariants. So inverting a group element can be done by solving the discrete-log problem over $\mathcal{CL}(D)$. However, the class number $h(D)$ is polynomially smooth, so the discrete-log problem over $\mathcal{CL}(D)$ can be solved in polynomial time once $h(D)$ is given, and $h(D)$ can be recovered from D or any basis of a relation lattice of $\mathcal{CL}(D)$. So we do need to hide the discriminant D, the class number $h(D)$, and any lattice Λ defined above. In the full version, we describe the details of how to hide these class group invariants.

Let us remark that if self-composition of an encoding is feasible, then one can efficiently guess all the polynomially smooth factors of $h(D)$. However for our construction self-composition is infeasible, due to the hardness of the (ℓ, ℓ^2)-isogenous neighbor problem. Nevertheless, one can still attack by first guessing D or $h(D)$, which takes $\lambda^{O(\log \lambda)}$ time according to the current setting of parameter.

Acknowledgments

The research of Salim Ali Altuğ is supported by the grant DMS-1702176. The research of Yilei Chen was conducted at Boston University supported by the NSF MACS project and NSF grant CNS-1422965.

References

1. László Babai. On lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
2. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *LNCS*, pages 1–18. Springer, 2001.
3. Johannes A. Buchmann and Hugh C. Williams. A key-exchange system based on imaginary quadratic fields. *J. Cryptology*, 1(2):107–118, 1988.
4. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.
5. Henri Cohen. A course in computational algebraic number theory. 1995.
6. Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.
7. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006.
8. Jean-Marc Couveignes and François Morain. Schoof's algorithm and isogeny cycles. In *International Algorithmic Number Theory Symposium*, pages 43–58. Springer, 1994.
9. David A Cox. *Primes of the form x^2+ny^2: Fermat, class field theory, and complex multiplication*, volume 34. John Wiley & Sons, 2011.
10. N. Demytko. A new elliptic curve based analogue of RSA. In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 40–49. Springer, 1993.
11. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.
12. Noam D Elkies et al. Elliptic and modular curves over finite fields and related computational issues. *AMS IP STUDIES IN ADVANCED MATHEMATICS*, 7:21–76, 1998.
13. Andreas Enge and Andrew V. Sutherland. Class invariants by the crt method. In *International Algorithmic Number Theory Symposium*, pages 142–156. Springer, 2010.
14. Luca De Feo. Mathematics of isogeny based cryptography. *arXiv preprint arXiv:1711.04062*, 2017.
15. Mireille Fouquet and François Morain. Isogeny volcanoes and the SEA algorithm. In *International Algorithmic Number Theory Symposium*, pages 276–291. Springer, 2002.
16. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49. IEEE Computer Society, 2013.
17. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.
18. Susan Rae Hohenberger. The cryptographic impact of groups with infeasible inversion. Master's thesis, Massachusetts Institute of Technology, 2003.
19. Sorina Ionica and Antoine Joux. Pairing the volcano. *Math. Comput.*, 82(281):581–603, 2013.
20. Jim Irrer, Satyanarayana Lokam, Lukasz Opyrchal, and Atul Prakash. Infeasible group inversion and broadcast encryption. *University of Michigan Electrical Engineering and Computer Science Tech Note CSE-TR-485-04*, 2004.
21. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2011.

22. Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.
23. David Russell Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.
24. Noboru Kunihiro and Kenji Koyama. Equivalence of counting the number of points on elliptic curve over the ring z_n and factoring n. In *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 47–58. Springer, 1998.
25. Georg-Johann Lay and Horst G Zimmer. Constructing elliptic curves with given group order over large finite fields. In *International Algorithmic Number Theory Symposium*, pages 250–263. Springer, 1994.
26. Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
27. Hendrik Willem Lenstra. Factoring integers with elliptic curves. *Annals of mathematics*, pages 649–673, 1987.
28. Kevin S McCurley. *Cryptographic key distribution and computation in class groups*. IBM Thomas J. Watson Research Division, 1988.
29. Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.
30. David Molnar. Homomorphic signature schemes. B.s. thesis, Harvard College, 2003.
31. Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145, 2006.
32. René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of computation*, 44(170):483–494, 1985.
33. René Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995.
34. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.
35. Joseph H Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151. Springer Science & Business Media, 2013.
36. Andrew V. Sutherland. Isogeny kernels and division polynomials. https://ocw.mit.edu/courses/mathematics/18-783-elliptic-curves-spring-2017/lecture-notes/MIT18_783S17_lec6.pdf. Accessed: 2018-09-03.
37. Andrew V. Sutherland. Isogeny volcanoes. *The Open Book Series*, 1(1):507–530, 2013.
38. John Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones mathematicae*, 2(2):134–144, 1966.
39. Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273:238–241, 1971.
40. Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 90–107, 2014.