A Candidate Group with Infeasible Inversion

Salim Ali Altuğ* Yilei Chen†

September 27, 2018

Abstract

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 in 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).

We propose a candidate trapdoor group with infeasible inversion without using the heavy machinery of iO. The underlying group is isomorphic to the ideal class group of an imaginary quadratic order, and is represented by the elliptic curve isogeny graph. The hardness of group inversion relies on the conjectured hardness of several problems on the isogeny graphs defined over composite moduli with unknown factorization.

*Boston University. saaltug@bu.edu. Research supported by the grant DMS-1702176.

†Visa Research. chenyilei.ra@gmail.com. Research conducted while the author was at Boston University supported by the NSF MACS project and NSF grant CNS-1422965.

Contents

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 the discrete-log-based cryptosystems, such as $(\mathbb{Z}/q\mathbb{Z})^*$ [DH76], the elliptic curve group [Mil85, Kob87], and the class group [BW88, McC88], 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 [Hoh03] and Molnar [Mol03], motivated with the potential cryptographic application of constructing a directed transitive signature. It was also shown by Irrer et al. [ILOP04] to provide a broadcast encryption scheme. However, the only existing candidate of such a group is implied by the much stronger primitive of self-bilinear maps constructed by Yamakawa et al. [YYHK14], assuming the hardness of integer factorization and indistinguishability obfuscation (iO) [BGI+01, GGH+13].

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[1] 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 isogeny volcanoes defined over composite moduli with unknown factorization.

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 the isogeny from E_1 to E_2 is traditionally called the *computational isogeny problem*.

The study of computing explicit isogenies began with a rather technical motivation of improving Schoof's polynomial time algorithm [Sch85] of computing the number of points on an elliptic curve over a finite field (the improved algorithm is usually called Schoof-Elkies-Atkin algorithm, cf. [CM94, Sch95, E+98] and references therein). A more straightforward use of computing explicit isogenies is to transfer the elliptic curve discrete-log problem from one curve to the other [Gal99, GHS02, JMV05]. If for any two isogenous elliptic curves computing an isogeny from one to the other is efficient, then it means the discrete-log problem is equally hard among all the isogenous elliptic curves.

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 $G_\ell(\mathbf{k})$ contains a j-invariant of an isomorphism class of curves (and their twists); 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 [Koh96]. Roughly speaking, a connected component of an isogeny graph containing ordinary elliptic curves looks like a *volcano* (termed in [FM02]). The connected component containing supersingular

[1] By emphasizing the "representation", we would like to remind the readers that the hardness of group theoretical problems (like the discrete-log problem) depends on the group representation rather than the group structure. After all, most of the cryptographically interesting finite groups are chosen to be isomorphic to the innocent looking additive group $\mathbb{Z}/n\mathbb{Z}$, $n \in \mathbb{N}$. However, the isomorphism is typically hard to compute.

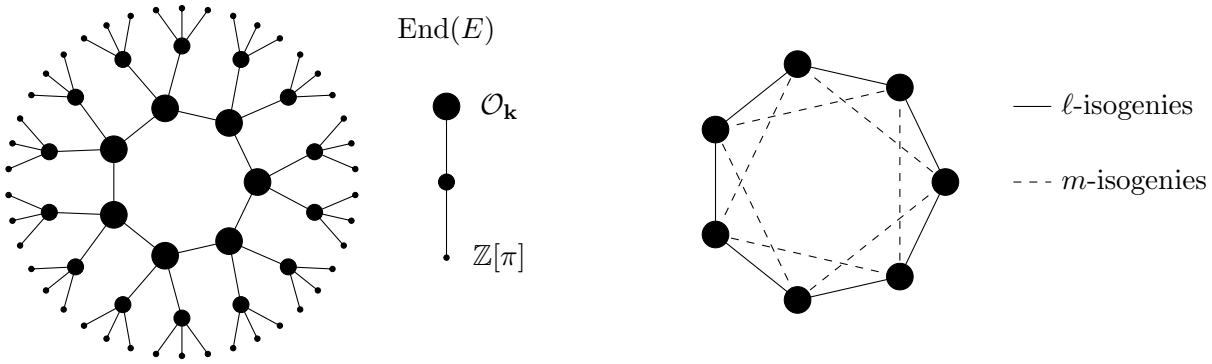

Figure 1: Examples of isogeny graphs. Left: a connected component of $G_3(\mathbf{k})$, and the corresponding tower of imaginary quadratic orders [Feo17]; Right: the vertex set is $\mathrm{Ell}_{\mathcal{O}}(\mathbb{C})$ for an imaginary quadratic order $\mathcal{O}$, the edges represent (isomorphic classes of) isogenies of degrees ℓ, m.

elliptic curves, on the other hand, has a different structure. In this article we will focus only on the ordinary case.

A closer look at the algorithms of computing isogenies. As above, let $\mathbf{k}$ be a finite field of q elements, and an integer ℓ 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. To compute kernel subgroups of E of size ℓ, then to apply Vélu's formulae to obtain the explicit isogenies and the j-invariants of the image curves.

2. To obtain the j-invariants of the image curves by solving the modular polynomial over $\mathbf{k}$, then to construct the 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 [Cou06] (written in 1997 but not published until 2006) and independently by Rostovtsev and Stolbunov [RS06]. Moreover, a hash function and a key exchange scheme were proposed based on the hardness of computing isogenies over supersingular curves [CLG09, JF11]. Isogeny-based cryptography is attracting attention partially due to their conjectured post-quantum security.

1.2 Isogeny volcanoes over a composite modulus with unknown factorization

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

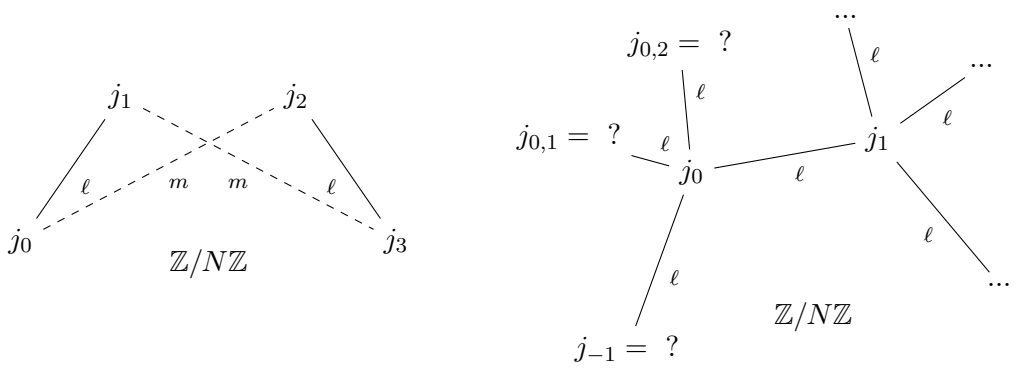

Figure 2: Left: the (ℓ, m)-isogenous neighbor problem where $\gcd(\ell, m) = 1$. Right: the (ℓ, ℓ^2)-isogenous neighbor problem.

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 2, 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 2. Suppose we are given j_0 and j_1 that are ℓ-isogenous, and asked to find 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.

Besides the gcd method described above, currently we do not know of another way of solving the (ℓ, ℓ^2)-joint neighbor problem. Neither do we know if solving this problem is as hard as factoring N. We will list a few attempts we have made in solving or showing the hardness of this problem.

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

1.3 Representing ideal class groups by isogeny volcanoes

To explain the construction of the trapdoor group with infeasible inversion, 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 $j \in \mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$; and 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 $\mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$ is equal to the class number $h(\mathcal{O})$.

To represent the ideal class group of an imaginary quadratic order $\mathcal{O}$, we choose curves $E_{0,\mathbb{F}_p}$ over $\mathbb{F}_p$ and $E_{0,\mathbb{F}_q}$ over $\mathbb{F}_q$ such that their endomorphism rings over $\mathbb{F}_p$ and $\mathbb{F}_q$ are both isomorphic to $\mathcal{O}$. From now on, by abuse of notation, we will be referring to both $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$ as E_0 unless we explicitly need to distinguish between the two curves. We hope that the individual curve under discussion will be clear from the context. We further remark that we can also start with a single curve E_0 over $\mathbb{Z}$ and go through the whole construction by simply reducing the curve modulo p and modulo q.

Let $N = p \cdot q$ and let j_0 be the j-invariant of E_0 over $\mathbb{Z}/N\mathbb{Z}$ defined by the CRT composition of the j-invariants of $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$. The public parameter of the group is then (N, j_0), where j_0 represents the identity of $\mathcal{CL}(\mathcal{O})$. An element $\mathfrak{a} \in \mathcal{CL}(\mathcal{O})$ is canonically represented by the j-invariant $\mathfrak{a} * j_0$ (once again, obtained over $\mathbb{F}_p$ and $\mathbb{F}_q$ then composed by CRT).

As in the (ℓ, m)-joint neighbor search problem of §1.2, given two class group elements represented by coprime degree isogenies, the group operation can be performed efficiently by taking the gcd of two modular polynomials. On the other hand, given a group element represented by a j-invariant j_1 that is ℓ-isogenous to j_0, computing an encoding of the group inversion is equivalent to computing a j-invariant that is ℓ-isogenous to j_0 and ℓ^2-isogenous to j_1, and lies in the same endomorphism ring with j_0 and j_1. It is one of the solutions of the (ℓ, ℓ^2)-isogenous neighbor problem.

Let us remark that the actual instantiation of the trapdoor group with infeasible inversion (TGII) is rather involved. A serious amount 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 an open problem. Our construction, however, requires more than knowing the class number to support an efficient encoding algorithm. Due to various constraints, currently we can only choose the parameters from a narrow range so as to support an efficient parameter generation algorithm, an efficient encoding algorithm, and to preserve the plausible security of the TGII. Extending the working parameters regime seems to require the solutions of several open problems concerning ideal class groups of imaginary quadratic orders.

We also note that our concrete instantiation deviates in several places from the ideal interface of a TGII. One of the deviations is that computing the self-composition of a group element is inefficient, due exactly to the hardness of the (ℓ, ℓ^2)-joint neighbor problem.

As a result of the complication from the class groups and all the deviations, additional engineering efforts have to be made when instantiating the applications of a TGII from their designs under the

ideal interface. In the instantiations of a directed transitive signature and a broadcast encryption scheme, we will specify the choices of parameters so as to provide both the functionality and plausible security. The hardness of the (ℓ, ℓ^2)-joint neighbor problem is merely a necessary condition for security. We will mention our cryptanalysis attempts and list the other problems related to the security of our TGII candidate.

1.4 Further discussions

Note that given g and g^a over the ring $(\mathbb{Z}/n\mathbb{Z})^*$, computing g^{-a} is feasible for any n. On the other hand, computing $g^{1/a}$ is infeasible for suitable subgroup $\mathbb{G}$ of $(\mathbb{Z}/n\mathbb{Z})^*$. However, in general, it is not clear how to efficiently perform the multiplicative operation "in the exponent".

The only existing candidate of (T)GII that supports a large number of group operations is implied by the self-bilinear maps constructed by Yamakawa et al. [YYHK14] using general purpose indistinguishability obfuscation [BGI+01]. The existence of iO is currently considered a strong assumption in the cryptography community. Over the past five years many candidates (since [GGH+13]) and attacks (since [CHL+15]) were proposed for iO. Basing iO on a clearly stated hard mathematical problem is still an open research area.

Nevertheless, the self-bilinear maps construction from iO is conceptually simple. Here we sketch the idea. Given an integer N with unknown factorization, a group element $a \in (\mathbb{Z}/\frac{\phi(N)}{4}\mathbb{Z})^*$ is represented by $g^a \in QR^+(N)$ (QR^+ denotes the signed group of quadratic residues), together with an obfuscation of the circuit $C_{2a,N}$:

$$C_{2a,N} : QR^+(N) \to QR^+(N), \quad x \mapsto x^{2a}.$$

Given g^a, $\mathrm{Obf}(C_{2a,N})$, g^b, $\mathrm{Obf}(C_{2b,N})$, everyone is able to compute g^{2ab}. [YYHK14] proves that under the hardness of factoring and assuming that the obfuscator satisfy the security of indistinguishable obfuscation, it is infeasible for the adversary to compute g^{ab}. Such a result implies that under the same assumption, it is infeasible to compute $g^{1/x}$ given g^x and $\mathrm{Obf}(C_{2x,N})$.

The obfuscated circuit is referred to as "auxiliary input" in [YYHK14], so what [YYHK14] constructed is precisely called "self-bilinear maps with auxiliary input". The downside of having auxiliary inputs is that the encodings of the group elements keep growing after the compositions. Self-bilinear maps without auxiliary input is recently investigated by [YYHK18] in the context of rings with infeasible inversion, but constructing them is still open even assuming iO.

The thesis of Hohenberger. Hohenberger [Hoh03] studies the sufficient and necessary conditions of constructing a group with infeasible inversion. Given that our construction deviates in several places from the ideal interface of a (T)GII, not all the conditions from [Hoh03] hold for our construction. For example, it is mentioned in [Hoh03] that the group order cannot be released in a GII, since the group inversion can be trivially computed by making self-compositions once the group order is known. But the reason does not apply to our construction since our construction does not support self-composition. However, we still need to hide the group order because the group order is chosen to be polynomially smooth, and revealing the group order allows the adversary to solve the discrete-log problem efficiently.

[Hoh03] also studies the relations of (T)GII to the other cryptographic primitives such as associative one-way functions. Again, due to the deviation of our candidate from the ideal interface of a (T)GII, the relations or implications do not necessary hold for our candidate.

2 Preliminaries

Notations and terminology. Let $\mathbb{C}, \mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{N}$ be the set of complex numbers, reals, rationals, integers, and positive integers. For any field K we denote its algebraic closure 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 integer $y \in \mathbb{Z}/(\prod_i^k p_i)\mathbb{Z}$ such that $y \equiv c_i \pmod{p_i}$, for $i \in [k]$.

In cryptography, the security parameter (denoted by λ) is a variable that is used to parameterize the computational complexity of the cryptographic algorithm or protocol, and the adversary's probability of breaking security. In theory and by default, an algorithm is called "efficient" if it runs in probabilistic polynomial time over λ. Exceptions may occur in reality and we will explicitly discuss them when they come up in our applications.

An n-dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$ that generate it as a vector space over $\mathbb{R}$. Given n linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{R}^n\}$, the lattice generated by $\mathbf{B}$ is

$$\Lambda(\mathbf{B}) = \Lambda(\mathbf{b}_1, ..., \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \cdot \mathbf{b}_i, x_i \in \mathbb{Z} \right\}.$$

Let $\tilde{\mathbf{B}}$ denote the Gram-Schmidt orthogonalization of $\mathbf{B}$.

Let $\mathbb{G}$ denote a finite abelian group, and let the prime factorization of its order $|\mathbb{G}|$ be $|\mathbb{G}| = \prod_{i \in [k]} p_i^{w(p_i)}$. For each p_i, let $H(p_i) := |\mathbb{G}|/p_i^{w(p_i)}$, and $G(p_i) := \left\{ g^{H(p_i)}, g \in \mathbb{G} \right\}$. We have the isomorphism

$$\mathbb{G} \to \mathbb{G}(p_1) \times ... \times \mathbb{G}(p_k), \quad g \mapsto (g^{H(p_1)}, ..., g^{H(p_k)}).$$

For a cyclic group $\mathbb{G}$, the *discrete-log problem* asks to find the exponent $a \in [|\mathbb{G}|]$ given a generator g and a group element $x = g^a \in \mathbb{G}$. The Pohlig-Hellman algorithm [PH78] solves the discrete-log problem in time $O\left(\sum_i w(p_i)(\log|\mathbb{G}| + \sqrt{p_i})\right)$ if the factorization of $|\mathbb{G}|$ is known.

Over a possibly non-cyclic group $\mathbb{G}$, the discrete-log problem is defined as follows: given a set of elements $g_1, ..., g_k$ and a group element $x \in \mathbb{G}$, output a vector $\mathbf{e} \in \mathbb{Z}^k$ such that $x = \prod_{i=1}^k g_i^{e_i}$, or decide that x is not in the subgroup generated by $\{g_1, ..., g_k\}$. A generalization of the Pohlig-Hellman algorithm works for non-cyclic groups with essentially the same cost plus an $O(\log|\mathbb{G}|)$ factor (the algorithm is folklore [PH78] and is explicitly given in [Tes99]). A further improvement is given by Sutherland [Sut11b].

2.1 Ideal class groups of imaginary quadratic orders

There are two equivalent ways of describing ideal class groups of imaginary quadratic orders: via the theory of ideals or quadratic forms. We will be using these two view points interchangeably. The main references for these are [McC88, Coh95, Cox11].

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. The standard representation of an $\mathcal{O}$-ideal of discriminant D uses 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 [Coh95]). 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 [McC88, Page 9].

Computing $h(D)$ and solving discrete-log problem over $\mathcal{CL}(D)$. The problem of computing the class number (namely, given the discriminant $D < 0$ of an imaginary quadratic order, computing $h(D)$) is only known to have polynomial-size witnesses under the Generalized Riemann Hypothesis (GRH) [McC88]. It follows from the existence of a polynomial-size generation set of the class group under GRH.

Lemma 2.1 ([Sch82] Corollary 6.2)**.** *Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D. Let $p_i \in \mathbb{N}$ be the i^{th} prime with $\left(\frac{D}{p_i}\right) = 1$, and let $b_i = \min\left\{b \in \mathbb{N} : b^2 \equiv D \pmod{4p_i}\right\}$. Assuming GRH there exists a constant c_0 such that the classes $[(p_i, b_i, \cdot)]$, $1 \leq i \leq m$ generate $\mathcal{CL}(\mathcal{O})$ where $p_m < c_0 \log^2 |D|$.*

Let $B \in \mathbb{N}$. Let $\mathcal{P}_B$ be the set of primes s.t. $\mathcal{P}_B = \left\{p \mid p \text{ is a prime}, p \leq B, \left(\frac{D}{p}\right) = 1\right\}$. Let the corresponding reduced forms be $C_i := [(p_i, b_i, \cdot)]$. From Lemma 2.1 it follows that if $m := |\mathcal{P}_B| \geq c_0 \log^2 |D|$ then the map

$$\psi : \mathbb{Z}^m \to \mathcal{CL}(D), \qquad \mathbf{e} \mapsto \prod_{i \in [m]} C_i^{e_i}$$

is a surjective group homomorphism. Hence the kernel Λ of ψ is a sublattice of $\mathbb{Z}^m$, and $\mathbb{Z}^m/\Lambda \simeq \mathcal{CL}(D)$ and $|\det(\Lambda)| = h(D)$. Λ is also called the *relation lattice*.

Lemma 2.2 ([McC88, HM89])**.** *Assuming GRH, there exists a Las Vegas algorithm that computes the invariants (a basis of Λ, $h(D)$, and the group structure) of $\mathcal{CL}(D)$ in an expected running time of $L(|D|)^{\sqrt{2}+o(1)}$, where $L(x) = e^{\sqrt{\log x \log\log x}}$.*

Once we have the class group invariants, solving the discrete-log problem over $\mathcal{CL}(D)$ takes $L(|D|)^{\frac{1}{2}+o(1)}$ time per instance [BD90].

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 [Koh96, Sil09, Sil13, Sut13a, Feo17].

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 [Sch85] 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 $E_1(\mathbf{k})$ to $E_2(\mathbf{k})$ with a finite kernel. Elliptic curves related by a nonzero isogeny are said to be isogenous. By the Tate isogeny theorem [Tat66, 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 $\mathbb{H} := \{\tau \in \mathbb{C} : \text{im } \tau > 0\}$ 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 $\text{SL}_2(\mathbb{Z})$ defined by

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

The time and space required for computing the modular polynomial Φ_ℓ are polynomial in ℓ, cf. [E+98, § 3] or [Coh95, 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]. In reality the coefficients of Φ_ℓ over $\mathbb{Z}[x, y]$ grow significantly with ℓ, so computing Φ_ℓ over $\mathbf{k}[x, y]$ directly is preferable using the improved algorithms of [CL05, BLS12], or even $\Phi_\ell(j_1, y)$ over $\mathbf{k}[y]$ using [Sut13b].

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 $\mathbf{k}$, $\text{End}(E)$ 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. These are graphs capturing the relation of being ℓ-isogenous among elliptic curves over a finite field $\mathbf{k}$.

Definition 2.3 (ℓ-isogeny graph)**.** *Fix a prime ℓ and a finite field $\mathbf{k}$ such that $\text{char}(\mathbf{k}) \neq \ell$. 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. Every curve in the isogeny class of a supersingular curve is supersingular. Accordingly, $G_\ell(\mathbf{k})$ has super singular and ordinary components. The ordinary components of $G_\ell(\mathbf{k})$ look like *ℓ-volcanoes*:

Definition 2.4 (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}.*

[2]The modular polynomials Φ_ℓ for $1 \leq \ell \leq 300$ are available at https://math.mit.edu/~drew/ClassicalModPolys.html.

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 $\mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$.

Every horizontal ℓ-isogeny arises this way from the action of an invertible $\mathcal{O}$-ideal $\mathfrak{l}$ of norm ℓ. Let K denote the fraction field of $\mathcal{O}$ and $\mathcal{O}_K$ be its ring of integers. If $\ell \mid [\mathcal{O}_K : \mathcal{O}]$ then no such ideal exists. Otherwise, $\mathcal{O}$ is said to be maximal at ℓ and there are $1 + \left(\frac{D}{\ell}\right)$ horizontal ℓ-isogenies.

Remark 2.5 (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. [Sch95]). 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 $\pi(P) = [\mu]P$ module G. If so then μ corresponds to ϕ.*

The following fundamental result of Kohel summarizes the above discussion and more.

Lemma 2.6 ([Koh96])**.** *Let ℓ be a prime. Let V be an ordinary component of $G_\ell(\mathbb{F}_q)$ that does not contain 0 or 1728. Then V is an ℓ-volcano for which the following hold:*

1. *The vertices in level V_i all have the same endomorphism ring $\mathcal{O}_i$.*
2. *The subgraph on V_0 has degree $1 + \left(\frac{D_0}{\ell}\right)$, where $D_0 = \mathrm{disc}(\mathcal{O}_0)$.*
3. *If $\left(\frac{D_0}{\ell}\right) \geq 0$, then $|V_0|$ is the order of $[\mathfrak{l}]$ in $\mathcal{CL}(\mathcal{O}_0)$; otherwise $|V_0| = 1$.*
4. *The depth of V is d, where $2d$ is the largest power of ℓ dividing $(t^2 - 4q)/D_0$, and $t^2 = \mathrm{tr}(\pi_{\mathrm{E}})^2$ for $j(E) \in V$.*
5. *$\ell \nmid [\mathcal{O}_K : \mathcal{O}_0]$ and $[\mathcal{O}_i : \mathcal{O}_{i+1}] = \ell$ for $0 \leq i < d$.*

Let $\mathcal{G}_{\mathcal{O},m}(\mathbf{k})$ be the regular graph whose vertices are the elements of $\mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$, and whose edges are the equivalence classes of horizontal isogenies defined over $\mathbf{k}$ of prime degrees $\leq m$. The following result states that under suitable assumptions $\mathcal{G}_{\mathcal{O},m}(\mathbf{k})$ is an expander graph.

Lemma 2.7 ([JMV05])**.** *Let $q = \#\mathbf{k}$ and δ be a fixed constant. Let m be such that $m \geq (\log q)^{2+\delta}$. Assuming GRH, a random walk on $\mathcal{G}_{\mathcal{O},m}(\mathbf{k})$ will reach a subset of size h with probability at least $\frac{h}{2|\mathcal{G}_{\mathcal{O},m}(\mathbf{k})|}$ after polylog(q) many steps.*

More about the endomorphism ring from a computational perspective. Given an ordinary curve E over $\mathbf{k}$, its endomorphism ring $\mathcal{O}$ can be determined by first computing the trace t of Frobenius endomorphism π, then computing $t^2 - 4q = v^2 D_0$, where $v^2 D_0$ is the discriminant of $\mathbb{Z}[\pi]$, $\mathbb{Z}[\pi] \subseteq \mathcal{O} \subseteq \mathcal{O}_K$, and $K = \mathbb{Q}(\sqrt{D_0})$. The discriminant of $\mathcal{O}$ is then $u^2 D_0$ for some $u \mid v$. When v has only few small factors, determining the endomorphism ring can be done in time polynomial in $\log(q)$ [Koh96]. In general it can take up to subexponential time in $\log(q)$ under GRH [BS11, Bis11].

Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D. Let $H_D(x)$ be the Hilbert class polynomial defined by

$$H_D(x) = \prod_{j(E) \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{C})} (x - j(E)).$$

H_D has integer coefficients and is of degree $h(D)$. Furthermore, it takes $O(|D|^{1+\epsilon})$ bits of storage. Under GRH, computing H_D mod q takes $O(|D|^{1+\epsilon})$ time and $O(|D|^{1/2+\epsilon} \log q)$ space [Sut11a]. In reality H_D is only feasible for small $|D|$ since it takes a solid amount of space to store H_D. Over $\mathbb{Z}[x]$, [Sut11a] is able to compute H_D for $|D| \approx 10^{13}$ and $h(D) \approx 10^6$. Over $\mathbb{F}_q[x]$, [Sut12] is able to compute H_D for $|D| \approx 10^{16}$ with $q \approx 2^{256}$.

3 Isogeny volcanoes 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 [Len87].

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 3.1 (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 3.2 (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 3.3. *If there is a probabilistic polynomial time algorithm that finds all the ℓ-isogenous neighbors in Problem 3.2, 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 a curve E over[3] $\mathbb{F}_p$ such that the vertex $j(E) \in G_\ell(\mathbb{F}_p)$ has at least two distinct neighbors. If we are able to find *all* the integer solutions $j' \in \mathbb{Z}/N\mathbb{Z}$ such that $\Phi_\ell(j(E), j') = 0$ 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 3.4 ([Len87] (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*

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

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 3.5. *Let p, ℓ be primes such that $6\ell < \sqrt{p}$. Then, there exists a constant $c > 0$ such that 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 $\Omega(\frac{1}{\log p})$.*

Proof of Theorem 3.5. We first give a lower bound on the number of ordinary elliptic curves over $\mathbb{F}_p$ whose endomorphism ring has discriminant D such that $\left(\frac{D}{\ell}\right) = 1$. If for some pair of (ℓ, p) there are not enough elliptic curves over $\mathbb{F}_p$ with two horizontal ℓ-isogenies then we count the elliptic curves with vertical ℓ-isogenies.

We start with estimating the portion of $t \in [\ell]$ that satisfies $\left(\frac{t^2-4p}{\ell}\right) = 1$:

$$\Pr_{t\in[\ell]}\left[\left(\frac{t^2 - 4p}{\ell}\right) = 1\right] = \begin{cases} 0 & \ell = 2 \\ \frac{1}{2} - \frac{3}{2\ell} & \ell > 2 \text{ and } \left(\frac{4p}{\ell}\right) = 1 \\ \frac{1}{2} - \frac{1}{2\ell} & \ell > 2 \text{ and } \left(\frac{4p}{\ell}\right) = -1 \end{cases} \tag{2}$$

[3] The choice of $\mathbb{F}_p$ over $\mathbb{F}_q$, obviously, does not matter.

where the last two equations follows the identity[4] $\sum_{t=1}^{\ell}\left(\frac{t^2-4p}{\ell}\right)=-1$. Hence for $\ell\geq 5$ or $\ell=3$ and $\left(\frac{4p}{3}\right)=\left(\frac{p}{3}\right)=-1$, no less than $\frac{1}{2}-\frac{3}{2\ell}$ of the $t\in[\ell]$ satisfy $\left(\frac{t^2-4p}{\ell}\right)=1$.

We now estimate the number of elliptic curves over $\mathbb{F}_p$ whose discriminant of the endomorphism ring D satisfies $\left(\frac{D}{\ell}\right)=1$. To do so we set $r=\lfloor\sqrt{p}/\ell\rfloor$, and use Lemma 3.4 by choosing the set S as

$$S=\left\{s\;\middle|\;\left(\frac{(p+1-s)^2-4p}{\ell}\right)=1, s\in\{(p+1)-r\cdot\ell,...,(p+1)+r\cdot\ell\}\setminus\{p+1\}\right\}.$$

Note that $\#S\geq r(\ell-3)$. Therefore, Lemma 3.4, there exists an effectively computable constant c such that the number of isomorphism classes of elliptic curves over $\mathbb{F}_p$ with the number of points in the set S is greater or equal to

$$c\cdot(r(\ell-3)-2)\cdot\frac{\sqrt{p}}{\log p}\geq c\cdot((\frac{\sqrt{p}}{\ell}-1)(\ell-3)-2)\cdot\frac{\sqrt{p}}{\log p}>c\cdot(\frac{\sqrt{p}}{2\ell}\cdot\frac{\ell}{3}-2)\cdot\frac{\sqrt{p}}{\log p}\geq c\cdot\frac{p}{18\log p}. \quad (3)$$

Since the total number of elliptic curves over $\mathbb{F}_p$ is p^2-p; the number of elliptic curves isomorphic to a given elliptic curve E is $\frac{(p-1)}{\#\mathrm{Aut}E}$ [Len87, (1.4)]. So for $\ell\geq 5$ or $\ell=3$ and $\left(\frac{4p}{3}\right)=-1$, the ratio of elliptic curves over $\mathbb{F}_p$ with discriminant D such that $\left(\frac{D}{\ell}\right)=1$ is $\Omega(\frac{1}{\log p})$.

To finish the treatment of the case, where $\ell\geq 5$, or $\ell=3$ and $\left(\frac{4p}{3}\right)=\left(\frac{p}{3}\right)=-1$, we will show that among such curves the proportion of the j-invariants $j(E)$ on the crater of the volcano having one or two neighbors is $o(\frac{1}{\log p})$. Recall that we are in the case $\ell\nmid D$ and $\ell=\mathfrak{l}_1\mathfrak{l}_2$ in $\mathbb{Q}(\sqrt{D})$, and that the crater has size equals to the order of $\mathfrak{l}_1$ (which is the same as the order of $\mathfrak{l}_2$) in $\mathcal{CL}(\mathcal{O})$.

If the crater has 1 or 2 vertices, $\mathfrak{l}_1$ must have order dividing 2 in $\mathcal{CL}(\mathcal{O})$. If $\mathfrak{l}_1$ has order 1 in $\mathcal{CL}(\mathcal{O})$ then we have $x^2-Dy^2=\ell$ for some $x,y\in\mathbb{Z}$. Since ℓ is prime we necessarily have $y\neq 0$. Moreover, since $6\ell<\sqrt{p}$ we have $-D<\sqrt{p}$ and therefore $4p-\sqrt{p}<t^2$. On the other hand, by the Hasse bound we have $t^2\leq 4p$, hence $4p-\sqrt{p}<t^2\leq 4p$. Therefore, there are at most $O(p^{\frac{1}{4}})$ j-invariants for which the the top of the volcano consists of a single vertex. This handles the case of $\mathfrak{l}_1$ having order 1.

The remaining case of $\mathfrak{l}_1$ having order 2, on the other hand, cannot happen because of genus theory. More precisely, let D_0 be the discriminant of $\mathbb{Q}(\sqrt{D})$. Since $\mathfrak{l}_1$ has order 2 in $\mathcal{CL}(D)$ and $\mathfrak{l}_1\nmid D$ it has order 2 in the class group $\mathcal{CL}(D_0)$ of $\mathbb{Q}(\sqrt{D})$. Now, recall that, by genus theory, the 2-torsion in $\mathcal{CL}(D_0)$ is generated by primes dividing D_0. Therefore, if $\mathfrak{l}_1$ has order 2, then $\mathfrak{l}_1\mid D_0$, which gives a contradiction.

Therefore, in the case of $\left(\frac{D}{\ell}\right)=1$ the probability that a random $j(E)\in\mathbb{F}_p$ having ≤ 2 neighbors on the crater of the volcano is $O(p^{-\frac{3}{4}})=o(\frac{1}{\log p})$, which finishes the treatment of the case $\ell\geq 5$ or $\ell=3$ and $\left(\frac{4p}{3}\right)=\left(\frac{p}{3}\right)=-1$.

For the remaining cases, where $\ell=2$, or $\ell=3$ and $\left(\frac{p}{3}\right)=1$, we count the number of vertical isogenies. Following the formula for the depth of an isogeny volcano in Lemma 2.6, for an elliptic curve $E(\mathbb{F}_p)$ of trace t with $\ell^2\mid t^2-4p$, the curve lives on the part of the volcano of depth ≥ 1. In this case we only need to make sure that the curve does not live at the bottom of the volcano because otherwise it will necessarily have at least ℓ-neighbors (if it is at the bottom it has only one neighbor).

When $\ell=2$, every $t=2t_1$ satisfies $4\mid t^2-4p$. So every $t\in[-2\sqrt{p},2\sqrt{p}]\cap 2\mathbb{Z}$ corresponds to trace of an elliptic curve over $\mathbb{F}_p$ with at least two neighbors.

[4]Derived from Theorem 19 in http://www.imomath.com/index.php?options=328&lmm=0.

When $\ell = 3$ and $\left(\frac{p}{3}\right) = 1$, Hensel's lemma implies that $\frac{2}{9}$ of the $t \in [-2\sqrt{p}, 2\sqrt{p}] \cap \mathbb{Z}$ satisfy $t^2 \equiv 4p \bmod 9$. These all correspond to traces of an elliptic curves over $\mathbb{F}_p$ with at least two neighbors.

Finally, in both cases using Lemma 3.4 with a set S that takes a constant fraction from $[p+1-\sqrt{p}, p+1+\sqrt{p}] \cap \mathbb{Z}$, we see that the $O(\frac{1}{\log p})$ lower bound also applies for $\ell = 2$ or $\ell = 3$ and $\left(\frac{p}{3}\right) = 1$. □

Proof of Theorem 3.3. Suppose that there is a probabilistic polynomial time algorithm A that finds all the ℓ-isogenous neighbors in Problem 3.2 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 3.2 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 3.5. 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 3.6 (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 3.7 ([ES10])**.** *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 $f(x) := \gcd(\Phi_\ell(j_1, x), \Phi_m(j_2, x))$ is a linear polynomial over x.*

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 3.8. *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 effect the degree of the gcd).

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

$$\deg(\Phi_\ell(X, j_0)) = \ell + 1, \qquad \deg(\Phi_{\ell^2}(X, j_1)) = \ell(\ell + 1). \tag{4}$$

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 (4) and the assumption that $\ell^2 + \ell < p$, is $\ell^2 + \ell$.

Next, recall (cf. Corollary 6.11 of [Sut]) 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 $E_0 \to E' \to E$, where $E' \not\cong E_1$. 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 (4) $N_{\gcd} \leq \ell$ 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 $\Phi_\ell(X, j_0)/(X - j_1)$. □

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 3.3, 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. We postpone further discussions on the hardness and cryptanalysis to Section 5.

4 A trapdoor group with infeasible inversion

In this section we present our construction of a trapdoor group with infeasible inversion (TGII).

4.1 The syntax of TGII

We first recall the ideal syntax of a TGII from [Hoh03, Mol03].

Definition 4.1. *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.*

Public sampling (optional). $\mathsf{Sam}(\mathsf{PP}, x)$ *takes as inputs the public parameter* PP *and a plaintext group element $x \in \mathbb{G}$, outputs an encoding* $\mathsf{encoding}(x)$.

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 an encoding* $\mathsf{encoding}(x)$.

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

Equivalence testing. $\mathsf{Equiv.test}(\mathsf{PP}, \mathsf{encoding}(x), \mathsf{encoding}(y))$ *takes as inputs the public parameter* PP*, two encoded elements* $\mathsf{encoding}(x), \mathsf{encoding}(y)$*, outputs* 1 *if* $x = y$*,* 0 *otherwise.*

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

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

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

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

4.2 Construction of TGII from isogenies

Let $\mathcal{CL}(\mathcal{O})$ denote the ideal class group of an imaginary quadratic order $\mathcal{O}$. The underlying group $\mathbb{G}$ with infeasible inversion will be a subgroup of $\mathcal{CL}(\mathcal{O})$. In most cases this subgroup will be clear from the context so by abuse of notation we usually denote it with $\mathcal{CL}(\mathcal{O})$ as well. If the need to distinguish the subgroup from the full class group arises we will explicitly clarify the relevant group.

We provide two formats of the encodings for a group element in $\mathcal{CL}(\mathcal{O})$. The *canonical encoding* of an element is uniquely determined once the public parameter is fixed. It can be used in the equivalence test, but does not support efficient group operations. The *composable encoding* of an element supports efficient group operations with the other composable encodings. It is not unique. The canonical encoding can be publicly obtained from the composable encoding, but we do not know how to obtain a composable encoding publicly from the canonical encoding.

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 their endomorphism rings over $\mathbb{F}_p$ and $\mathbb{F}_q$ are both isomorphic to $\mathcal{O}$. From now on, by abuse of notation, we will be referring to both $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$ as E_0 unless we explicitly need to distinguish between the two curves. Let $N = p \cdot q$ and let j_0 be the j-invariant of E_0 over $\mathbb{Z}/N\mathbb{Z}$ defined by the CRT composition of 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 then (N, j_0). The description of the group $\mathcal{CL}(\mathcal{O})$ and the class number $h(\mathcal{O})$ are not public under the current choices of parameters for the security purpose that we will explain later.

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 encoding is unique once j_0 and N are fixed.

To facilitate efficient compositions of the encodings of group elements, we need to represent them as the product of pairwise co-prime ideals with polynomially large norms. Such a representation is called the composable encoding. For example, for $x, y \in \mathcal{CL}(\mathcal{O})$, we first write $x = \prod_{i \in S_x} x_i^{e_i}$, $y = \prod_{j \in S_y} y_j^{f_j}$ where the norms $N(x_i)$ for $i \in S_x$ and $N(y_j)$ for $j \in S_y$ are pairwise relatively prime, and all the elements in $\{N(x_i), e_i\}_{i \in S_x}$ and $\{N(y_j), f_j\}_{j \in S_y}$ are polynomially large. Then, the composable encoding of x is a list of j-invariants, each representing the canonical encoding of an element in $\left\{\left\{x_i^k\right\}_{k \in [1, e_i]}\right\}_{i \in S_x}$; similarly for the composable encoding of y.

Formally, the algorithms in our TGII construction are given as follows.

Parameter generation. The parameter generation algorithm $\mathsf{Gen}(1^\lambda)$ takes the security parameter 1^λ as input and proceeds as follows:

1. Choose an imaginary quadratic order $\mathcal{O}$ of discriminant D together with a polynomially large set of ideal classes $S = \{C_i = [(\ell_i, b_i, \cdot)]\}_{i\in[m]}$ that generate $\mathcal{CL}(\mathcal{O})$. Let $\mathbb{G} := \mathcal{CL}(\mathcal{O})$. Define the relation lattice w.r.t. the set S as
$$\Lambda_{\mathcal{O}} := \left\{ \mathbf{e} \mid \mathbf{e} \in \mathbb{Z}^m, \prod_{i\in[m]} C_i^{e_i} = 1_{\mathbb{G}} \right\}. \tag{5}$$
Let $\mathbf{B}$ be a short basis of $\Lambda_{\mathcal{O}}$.
2. Choose two primes p, q, and elliptic curves $E_{0,\mathbb{F}_p}$, $E_{0,\mathbb{F}_q}$ such that $f^2\,D = t_p^2 - 4p$, $f^2\,D = t_q^2 - 4q < 0$, where D is a fundamental discriminant, and t_p and t_q are the traces of Frobenius endomorphisms of $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$ respectively. 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 $1_{\mathcal{CL}(\mathcal{O})}$.
3. Keep $(D, S, \mathbf{B}, p, q)$ as the trapdoor τ. Output (N, j_0) as the public parameter PP.

Producing more generation sets (optional). After fixing the discriminant D of $\mathcal{O}$, we can produce more generation sets, say $S' = \{C'_i = [(\ell'_i, b'_i, \cdot)]\}_{i\in[m']}$ together with a basis $\mathbf{B'}$ for the relation lattice
$$\Lambda' := \left\{ \mathbf{e} \mid \mathbf{e} \in \mathbb{Z}^{m'}, \prod_{i\in[m']} C_i'^{e_i} = 1_{\mathbb{G}} \right\}. \tag{6}$$
For certain choices of parameters, generating more short bases can be done in polynomial time, and can be included in the encoding phase instead of the parameter generation phase.

The canonical and composable encodings and their sampling algorithms. Next we explain the formats of the canonical encoding and the composable encodings, and their sampling algorithms.

We first define a unit operation that is commonly used in the encoding sampling algorithms.

Algorithm 4.2 (An ideal class C act on a j-invariant)**.** $\mathsf{act}(\tau, j, C)$ *takes as input the trapdoor* $\tau = (D, S, \mathbf{B}, p, q)$*, 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* $j'_p := C * j_p \in \mathbb{F}_p$*,* $j'_q := C * j_q \in \mathbb{F}_q$ *(recall* $* : \mathcal{CL}(\mathcal{O}) \times \mathrm{Ell}_{\mathcal{O}}(\mathbf{k}) \to \mathrm{Ell}_{\mathcal{O}}(\mathbf{k})$ *denotes the action of class group elements on the* j*-invariants). If any one of the computations fails then output* $\perp$*.*
3. *Output* $j' := \mathsf{CRT}(p, q; j'_p, j'_q)$*.*

Here is the definition of a canonical encoding.

Definition 4.3 (Canonical encoding)**.** *The canonical encoding of* $x \in \mathcal{CL}(\mathcal{O})$ *is* $x * j_0 \in \mathbb{Z}/N\mathbb{Z}$*.*

The canonical encoding of $x \in \mathcal{CL}(\mathcal{O})$ is computed by first obtaining a composable encoding of x, which will be defined next; then converting the composable encoding into the canonical encoding using Algorithm 4.8, which will be described a few paragraphs later.

Definition 4.4 (Composable encoding and its sampling algorithm)**.** *Given as input the public parameter* $\mathsf{PP} = (N, j_0)$*, the trapdoor* $\tau = (D, S, \mathbf{B}, p, q)$*, and a group element* $x \in \mathcal{CL}(D)$*, a composable encoding of* x *is defined and produced as follows:*

1. *Choose* $w_x \in \mathbb{N}$ *and a generation set* $S_x = \{C_{x,i} = [(p_{x,i}, b_{x,i}, \cdot)]\}_{i \in [w_x]}$*. Note that* S_x *is not necessarily a subset of* S*, and is, in fact, usually chosen not to be a subset of* S*.*
2. *Let* $L_x \in \mathbb{Z}^{w_x}$ *be a list where the* i^{th} *entry of* L *is* $p_{x,i}$*, equivalently, the degree of the isogeny that represents the* $C_{x,i}$ *action.*
3. *Sample a vector* $\mathbf{e}_x \in \mathbb{Z}^{w_x}$ *such that* $x = \prod_{i \in [w_x]} C_{x,i}^{e_{x,i}}$*, and* $e_{x,i} \geq 0$ *for all* $i \in [w_x]$*.*
4. *For* $i = 1$ *to* w_x*:*
   (a) *Parse the* i^{th} *ideal class* $C_{x,i} = [(p_{x,i}, b_{x,i}, \cdot)]$*.*
   (b) *Let* $j_{x,i,0} := j_0$*.*
   (c) *For* $k = 1$ *to* $e_{x,i}$*: compute* $j_{x,i,k} := \mathsf{act}(\tau, j_{x,i,k-1}, C_{x,i})$*.*
   (d) *Return* $T_{x,i} := (j_{x,i,1}, ..., j_{x,i,e_{x,i}})$*.*
5. *Output the composable encoding of* x *as*

$$\mathsf{encoding}(x) = (L_x; T_{x,1}, ..., T_{x,w_x}) = ((p_{x,1}, ..., p_{x,w_x}); (j_{x,1,1}, ..., j_{x,1,e_{x,1}}), ..., (j_{x,w_x,1}, ..., j_{x,w_x,e_{x,w_x}})).$$

The degree of an encoding $\mathsf{encoding}(x)$ *is defined to be* $d(\mathsf{encoding}(x)) := \prod_{i=1}^{w_x} p_{x,i}^{e_{x,i}}$*.*

Including ladders in the public parameter (optional). As mentioned in the beginning of §4.2, the group operation is natively feasible only for encodings with relatively prime degrees. To support the compositions of non-relatively prime degree encodings, say the shared prime degree is ℓ, we can include in the public parameter the encoding of x^k, where $x = [(\ell, b, \cdot)]$, k is a polynomial. It then supports the composition of two encodings whose sum of exponents on the degree ℓ is $\leq k$. In general it is feasible to support bounded number of shared-degree compositions.

Algorithm 4.5 (Sampling a ladder)**.** *Given a prime* ℓ*, a polynomial* k*, and an element* $x = [(\ell, b, \cdot)]$*, which fixes the direction in the isogeny circle, sample a ladder for degree* ℓ *of length* k *as follows:*

1. *For* $i = 1$ *to* k*: compute* $j_i := \mathsf{act}(\tau, j_{i-1}, x)$*.*
2. *Let* $\mathsf{ladder}(\ell) := (j_1, ..., j_k)$*. Include* $\mathsf{ladder}(\ell)$ *in the public parameter.*

Group operations. For the convenience of the description, let $\iota(\ell, L)$ take as input an integer ℓ, a list L, and output the index of ℓ in L.

Algorithm 4.6 (The group operation over two composable encodings)**.** *The encoding composition algorithm* $\mathsf{Compose}(\mathsf{PP}, \mathsf{encoding}(x), \mathsf{encoding}(y))$ *parses* $\mathsf{encoding}(x) = (L_x; T_{x,1}, ..., T_{x,w_x})$*,* $\mathsf{encoding}(y) = (L_y; T_{y,1}, ..., T_{y,w_y})$*, produces the composable encoding of* $z = x \circ y$ *as follows:*

1. *Let* $L_z = L_x \cup L_y$*.*
2. *For all* $\ell \in L_x \backslash L_x \cap L_y$*, let* $T_{z,\iota(\ell,L_z)} = T_{x,\iota(\ell,L_x)}$*; for all* $\ell \in L_y \backslash L_x \cap L_y$*, let* $T_{z,\iota(\ell,L_z)} = T_{y,\iota(\ell,L_y)}$*.*

3. *For all* $\ell \in L_x \cap L_y$*:*

   - *If* $|\mathsf{ladder}(\ell)| \geq |T_{x,\iota(\ell,L_x)}| + |T_{y,\iota(\ell,L_y)}|$*, then let* $T_{z,\iota(\ell,L_z)}$ *be the list of the first* $|T_{x,\iota(\ell,L_x)}| + |T_{y,\iota(\ell,L_y)}|$ *elements in* $\mathsf{ladder}(\ell)$*.*
   - *If* $|\mathsf{ladder}(\ell)| < |T_{x,\iota(\ell,L_x)}| + |T_{y,\iota(\ell,L_y)}|$*, then the composition is infeasible. Return "failure".*

4. *Output the composable encoding of* z *as* $\mathsf{encoding}(z) = (L_z; T_{z,1}, ..., T_{z,|L_z|})$*.*

Algorithm 4.7 (The unit operation)**.** *The algorithm* $\mathsf{op}(\mathsf{PP}, \ell_1, \ell_2; j_1, j_2)$ *takes as input the public parameter* PP*, and 4 integers* $\ell_1, \ell_2; j_1, j_2)$*, proceeds as follows:*

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

Algorithm 4.8 (Converting a composable encoding to the canonical encoding)**.** $\mathsf{Convert}(\mathsf{PP}, \mathsf{encoding}(x))$ *converts a composable encoding into the canonical encoding. The algorithm maintains two lists* (T, H)*, where* T *stores a list of* j*-invariants* $(j_1, ..., j_{|T|})$*, and* H *stores a list of degrees where the* i^{th} *entry of* H *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* H *and* T *are always equal.*

The algorithm parses $\mathsf{encoding}(x) = (L_x; T_{x,1}, ..., T_{x,w_x})$*, proceeds as follows:*

1. *Initiate* $T := T_{x,1}$*,* $H := (L_{x,1}, ..., L_{x,1})$ *of length* $|T_{x,1}|$*.*
2. *For* $i = 2$ *to* w_x*:*
   (a) *Set* $t_{\mathsf{temp}} := |T|$*.*
   (b) *For* $k = 1$ *to* $|T_{x,i}|$*:*
      i. *Let* $j_{k,0} := T_{x,i,k}$*;*
      ii. *For* $v = 1$ *to* t_{temp}*:*
         - *If* $k = 1$*, compute* $j_{k,v} := \mathsf{op}(\mathsf{PP}, L_{x,i}, H_v; j_{k,v-1}, T_v)$*;*
         - *If* $k > 1$*, compute* $j_{k,v} := \mathsf{op}(\mathsf{PP}, L_{x,i}, H_v; j_{k,v-1}, j_{k-1,v})$*;*
      iii. *Append* $j_{k,t_{\mathsf{temp}}}$ *to the list* T*, append* $L_{x,i}$ *to the list* H*.*
   (c) *Remove all the* $j_{k,v}$*, for* $k \in \{1, ..., |T_{x,i}|\}$*,* $v \in \{0, ..., t_{\mathsf{temp}}\}$*, from storage.*
3. *Return the last entry of* T*.*

Equivalence testing. $\mathsf{Equiv.test}(\mathsf{PP}, \mathsf{encoding}(x), \mathsf{encoding}(y))$ first converts $\mathsf{encoding}(x)$, $\mathsf{encoding}(y)$ into the canonical encodings. If the canonical encodings are the same, output 1; otherwise output 0.

Sampling the composable encoding of a random element (optional). In the applications we are required to trapdoor sample the composable encoding of a random element from $\mathcal{CL}(\mathcal{O})$ given a generation set $S' = \{C'_i := [(\ell'_i, b'_i, \cdot)]\}_{i \in [m']}$. Here we provide two solutions for the algorithm $\mathsf{RandomSam}(\tau, S')$, and assume in the applications we are always able to choose one of the solutions.

When $m' = O(\log(\lambda))$, we can compute a short basis for Λ' (cf. Eqn. (6)). In this case we can pick a random $x \in \mathcal{CL}(\mathcal{O})$ first (given the class group invariants in the trapdoor), then find a short vector $\mathbf{e}'$ such that $x = \prod_{i \in [m']} C_i'^{e'_i}$.

When $m' = \Omega(\log(N)^2)$, we can pick a random vector $\mathbf{e}' \in [-B, B]^{m'}$ where $B = \text{polylog}(N)$, and let $x = \prod_{i\in[m']} C_i'^{e_i'}$. Heuristically x is with high min-entropy, but figuring out the exact distribution of x is difficult in general. When all the ℓ_i' are small, then under GRH, x is statistically close to uniform over $\mathcal{CL}(\mathcal{O})$.

For both solutions we can then derive the composable encoding from S' and $\mathbf{e}'$.

The hardness assumption. We restate the hardness assumption for our candidate TGII for completeness. The hardness of inversion assumption says it is hard for any p.p.t. adversary to find any canonical encoding of x^{-1} given a composable encoding of x.

The hardness of inversion assumption. For any p.p.t. algorithm A,

$$\Pr[z = j(x^{-1} * E_0) \mid z \leftarrow A(\mathsf{PP}, \mathsf{encoding}(x))] < \mathsf{negl}(\lambda),$$

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

4.3 The choices of parameters

We now explain how to set the parameters for the scheme to get correctness, efficiency, and security. Here security means the hardness of inversion assumption holds for all the composable encodings of degree ≥ 5 (or > 7 as a precautionary measure). The security reasons behind the choices of parameters will be detailed in §5.

The parameters are chosen under the following constraints:

- The modulus N is chosen as the product of two large primes p, q so that factorizing N is hard.
- The imaginary quadratic order $\mathcal{O}$ is chosen with a super-polynomial discriminant D.
- All the $\mathcal{O}$-ideals used in generating the encodings are invertible. The norms of these $\mathcal{O}$-ideals (i.e. the degrees of the corresponding horizontal isogenies) are polynomially large primes ≥ 5.
- The parameters of $\mathcal{O}$ and the set of ideal classes $S = \{C_i = [(\ell_i, b_i, \cdot)] \in \mathcal{CL}(\mathcal{O})\}_{i\in[m]}$ are chosen so that $m \in \mathsf{poly}(\lambda)$, and given $x \in \mathcal{CL}(D)$ and the trapdoor, finding a vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\prod_{i=1}^m C_i^{e_i} = x$ is feasible in polynomial time.
- The basis $\mathbf{B}$ of $\Lambda_{\mathcal{O}}$ (cf. Eqn. (5)) and the basis $\mathbf{B}'$ of Λ' (cf. Eqn. (6)) are chosen such that $\|\tilde{\mathbf{B}}\|, \|\tilde{\mathbf{B}}'\| \in \mathsf{poly}(\lambda)$.
- 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) should be hidden and not polynomially smooth.

The degrees of all the isogenies are chosen to be ≥ 5. From the proof of Theorem 3.5 it is implicit that when $\ell = 2$ or $\ell = 3$ and $\left(\frac{p}{3}\right) = \left(\frac{q}{3}\right) = 1$, there are no horizontal isogenies forming a loop of length > 2, which is not interesting in our setting. When $\ell = 3$ and $\left(\frac{p}{3}\right) = \left(\frac{q}{3}\right) = -1$, there are horizontal 3-isogenies. However we choose to avoid $\ell = 3$ due to a security concern detailed in § 5.

On correctness and efficiency. We first show that the algorithms for encoding sampling and group operation are correct and efficient under the constraints of the parameters, then explain the more complicated part which is how to generate the parameters that satisfy the constraints.

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:

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

The unit operation $\mathsf{act}(\tau, j, C)$ is efficient when the ideal class C is represented by an ideal of polynomial norm, since it is efficient to compute polynomial degree isogenies over the finite fields.

The efficiency of sampling the composable encodings can be verified as follows. First, the parameter constraint guarantees that given $x \in \mathcal{CL}(D)$ and the trapdoor, finding a vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\prod_{i=1}^m C_i^{e_i} = x$ is feasible in polynomial time. The entries in $\mathbf{e}$ are not polynomially bounded. But given the basis $\mathbf{B}$ of the lattice $\Lambda_{\mathcal{O}}$ such that $\|\tilde{\mathbf{B}}\| \in \mathsf{poly}(\lambda)$, we can sample a vector $\mathbf{e}'$ in $\Lambda_{\mathcal{O}} + \mathbf{e}$ with polynomially large entries. Then we conclude by observing that the length of the composable encoding is $m \cdot (\|\mathbf{e}'\|_1 + 1) \in \mathsf{poly}(\lambda)$, and the time to produce such an encoding is the time to find $\mathbf{e}'$ and $m \cdot \|\mathbf{e}'\|_1$ times the runtime of the unit operation $\mathsf{act}()$, which is polynomial.

The algorithm $\mathsf{Compose}(\mathsf{PP}, \mathsf{encoding}(x), \mathsf{encoding}(y))$ simply concatenates $\mathsf{encoding}(x)$, $\mathsf{encoding}(y)$, plus the polynomially many information in the ladder (if needed). So it is efficient as long as $\mathsf{encoding}(x)$, $\mathsf{encoding}(y)$ are of polynomial size.

The correctness of the unit operation op follows the commutativity of the endomorphism ring $\mathcal{O}$. The operation $\mathsf{op}(\mathsf{PP}, \ell_1, \ell_2; j_1, j_2)$ 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 $\mathsf{Convert}()$ (Algorithm 4.8) on a composable encoding $\mathsf{encoding}(x) = (L_x; T_{x,1}, ..., T_{x,w_x})$, it runs 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 $i \in [w_x]$.

Two choices for the class group. Ideally we would like to efficiently sample an imaginary quadratic order $\mathcal{O}$ of discriminant D, together with the class number $h(D)$, a generation set $\{C_i\}$ and a short basis $\mathbf{B}$ for $\Lambda_{\mathcal{O}}$, and two large primes p, q as well as curves $E_{0,\mathbb{F}_p}, E_{0,\mathbb{F}_q}$ such that $\mathrm{End}(E_{0,\mathbb{F}_p}) \simeq \mathrm{End}(E_{0,\mathbb{F}_q}) \simeq \mathcal{O}$. In Section 5 we will show that if $|D|$ is polynomial then computing the group inversion takes polynomial time, so we are forced to choose a super-polynomially large $|D|$. On the other hand, working with a super-polynomially large D creates difficulties. For example, there is no polynomial time solution for the task of choosing a fundamental discriminant D with a large known $h(D)$ (see, for instance, [HM00]).

We describe two choices for the class group $\mathcal{CL}(\mathcal{O})$ (in fact, both choices use the odd part of the class group, but we abuse the notation $\mathcal{CL}(\mathcal{O})$ to represent its odd part). Both choices lead to heuristic polynomial time algorithms for generating and composing encodings. The first choice sets $\mathcal{O}$ as the maximal order of an imaginary quadratic field K, which is arguably more convenient to work with, but we do not know any algorithm that generates the parameters in polynomial time. The second choice sets $\mathcal{O}$ as a non-maximal order of an imaginary quadratic field K, and the class number $h(\mathcal{O})$ is chosen to be polynomially-smooth, which introduces additional challenges in the group representation and security analysis. But for the second choice we do provide a heuristic polynomial time parameter generation algorithm.

Choice 4.9 (Choice I)**.** *A maximal order $\mathcal{O}$ of an imaginary quadratic field is chosen as follows:*

- *Choose a fundamental discriminant $D < 0$ such that $|D| \approx \lambda^{O(\log^{1-\epsilon}(\lambda))}$, and let $\mathcal{O}$ be the ring of integers of $\mathbb{Q}(\sqrt{D})$.*
- *Pick a set of ideal classes $S = \{C_i = [(\ell_i, b_i, \cdot)]\}_{i\in[m]}$ that generate $\mathcal{CL}(\mathcal{O})$, where $m = O(\log(\lambda))$.*
- *Compute $h(D)$, and then a basis $\mathbf{B}$ of $\Lambda_{\mathcal{O}} = \left\{\mathbf{e} \mid \mathbf{e} \in \mathbb{Z}^m, \prod_{i\in[m]} C_i^{e_i} = 1_{\mathbb{G}}\right\}$ by solving discrete-log over $\mathcal{CL}(\mathcal{O})$. Run LLL on $\mathbf{B}$ to obtain a short basis.*

According to the Cohen-Lenstra heuristics [CL84], about 97.7575% of the imaginary quadratic fields K have the odd part of $\mathcal{CL}(\mathcal{O}_K)$ cyclic. If we choose D such that $|D| \equiv 3 \bmod 4$ and is a prime, then $h(D)$ is odd (by genus theory). So we might as well assume that $\mathcal{CL}(\mathcal{O}_K)$ is cyclic with odd order.

For a fixed discriminant D, heuristically about half of the primes ℓ satisfies $\left(\frac{D}{\ell}\right) = 1$, so there are polynomially many ideals of polynomially large norm that can be used in the generation set.

Suppose that the lattice $\Lambda_{\mathcal{O}}$ satisfies the Gaussian heuristic. That is, for all $1 \le i \le m$, the i^{th} successive minimum of $\Lambda_{\mathcal{O}}$ λ_i satisfies $\lambda_i \approx \sqrt{m} \cdot h(\mathcal{O}_K)^{1/m}$. Since we choose $m = O(\log(\lambda))$, $|D| \approx \lambda^{O(\log^{1-\epsilon}(\lambda))}$. Then $h(D) \approx O(\sqrt{|D|}) = \lambda^{O(\log^{1-\epsilon}(\lambda))}$, and the discrete-log problem over $\mathcal{CL}(D)$ can be solved in time $e^{O(\sqrt{\log^{2-\epsilon}(\lambda)\log(\log^2(\lambda))})} \in \mathsf{poly}(\lambda)$, which means we can efficiently generate a (possibly large) basis of $\Lambda_{\mathcal{O}}$. The short basis $\mathbf{B}$ of $\Lambda_{\mathcal{O}}$, produced by the LLL algorithm, satisfies $\|\mathbf{B}\| \le 2^{\frac{m}{2}} \cdot \lambda_m \in \mathsf{poly}(\lambda)$. So, under all the heuristics, the system parameter does imply polynomial time algorithms for sampling and composing encodings.

However, it is not clear how to efficiently choose p, q and curves $E_{0,\mathbb{F}_p}$, $E_{0,\mathbb{F}_q}$ such that the endomorphism rings of $E_{0,\mathbb{F}_p}$ and $E_{0,\mathbb{F}_q}$ have the given discriminant D of size $\lambda^{O(\log^{1-\epsilon}(\lambda))}$ (note that we do not specify the number of points $\#(E_{0,\mathbb{F}_p}(\mathbb{F}_p))$ and $\#(E_{0,\mathbb{F}_q}(\mathbb{F}_q))$). We remark that the classical CM method for choosing a curve with prescribed number of points (cf. [LZ94] and more) requires computing the Hilbert class polynomial H_D, whose cost grows proportional to $|D|$.

Choice 4.10 (Choice II)**.** *A non-maximal order $\mathcal{O}$ of an imaginary quadratic field is chosen as follows:*

- *Select a polynomially large negative square-free integer $D_0 \equiv 1 \bmod 4$ such that $h(D_0)$ is a prime.*
- *Choose an integer $n = O(\log(\lambda))$, and choose a set of polynomially large prime numbers $\{p_i\}_{i\in[n]}$ such that the odd-part of $\left(p_i - \left(\frac{D_0}{p_i}\right)\right)$ is square-free and nor divisible by $h(D_0)$ for all $i \in [n]$. Let $f = \prod_{i\in[n]} p_i$.*
- *Set $D = f^2 D_0$. Recall from Eqn. (1) that*

$$h(D) = 2 \cdot \frac{h(D_0)}{w(D_0)} \prod_{i\in[n]} \left(p_i - \left(\frac{D_0}{p_i}\right)\right) \tag{7}$$

  Let $\mathcal{CL}(\mathcal{O})_{odd}$ be the odd part of $\mathcal{CL}(\mathcal{O})$, and correspondingly $h(D)_{odd}$ be the odd part of $h(D)$.
- *Pick a set of ideal classes $S = \{C_i = [(\ell_i, b_i, \cdot)]\}_{i\in[m]}$ that generates $\mathcal{CL}(\mathcal{O})_{odd}$, and $\ell_i \in \mathsf{poly}(\lambda)$.*

- *Compute a basis* $\mathbf{B}$ *of* $\Lambda_{\mathcal{O}} = \left\{\mathbf{e} \mid \mathbf{e} \in \mathbb{Z}^m, \prod_{i\in[m]} C_i^{e_i} = 1_{\mathbb{G}}\right\}$ *by solving discrete-log over* $\mathcal{CL}(\mathcal{O})_{odd}$. *Run LLL on the basis* $\mathbf{B}$ *to obtain a short basis.*

Remark that Choice II gives $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ of cardinality $h(D)_{\text{odd}} \approx \lambda^{O(\log(\lambda))}$. Moreover, by construction, $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ is cyclic since its order is square-free, and the class number $h(D)_{\text{odd}}$ is polynomially smooth so that the discrete-log problem over $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ can be solved using the Pohlig-Hellman algorithm. Same as Choice I, the short basis of $\Lambda_{\mathcal{O}}$ produced by LLL algorithm satisfies $\|\mathbf{B}\| \leq 2^{\frac{m}{2}} \cdot \lambda_m \in \mathsf{poly}(\lambda)$.

Generating the curves $E_{0,\mathbb{F}_p}$, $E_{0,\mathbb{F}_q}$ with a given fundamental discriminant D_0 and a conductor f with factorization $f = \prod_i^k p_i$ 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 H_{D_0} over $\mathbb{F}_p$ and find one of its roots j. From j, descending on the volcanoes $G_{p_i}(\mathbb{F}_p)$ for every f_i gives the j-invariant for the curve with desired discriminant. The same construction works verbatim for q.

Remark 4.11. *We choose to work with cyclic groups for convenience. In fact, with more effort, one can iron out the details for the case where the odd part of* $\mathcal{CL}(\mathcal{O})$ *is not necessarily cyclic, with the possible benefit of being more flexible in the choices of parameters. The readers may turn to [Cox11, HJPT98] for further references.*

Remark 4.12. *In both Choice 4.9 and Choice 4.10, the group sizes are asymptotically upper bounded by* $\lambda^{O(\log(\lambda))}$, *which implies a* $\lambda^{O(\log(\lambda))}$*-time attack — first guess the group size and then solve the discrete-log problem. The bottleneck appears in the step of producing a short basis of the relation lattice* $\Lambda_{\mathcal{O}}$ *(and* Λ' *if in the optional mode where more generation sets are used) — the dimension of* $\Lambda_{\mathcal{O}}$ *is set to be* $O(\log(\lambda))$ *so that the basis reduction algorithms terminate in polynomial time.*

Let us remark that if an application of TGII can afford an unbounded parameter generation phase, then $O(\log(\lambda))$ *is no longer the restriction on the dimension of the relation lattice* $\Lambda_{\mathcal{O}}$. *The rest of the parameters can be chosen accordingly so that the best attack we know is to factorize* N. *Setting the dimension of* $\Lambda_{\mathcal{O}}$ *to be larger than the asymptotic bound is also reasonable in reality since the basis reduction algorithms are known to perform much better in practice.*

5 Cryptanalysis

We discuss our cryptanalysis attempts, and the countermeasures.

Central to the security of our cryptosystem is the conjectured hardness of solving various problems over $\mathbb{Z}/N\mathbb{Z}$ without knowing the factors of N. So the discussion of this section is organized by first analyzing the feasibility of performing several individual computational tasks over $\mathbb{Z}/N\mathbb{Z}$, then putting them altogether in the context of the candidate trapdoor group with infeasible inversion.

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. No polynomial time algorithm is known to solve this problem in general. The hardness of a few special instances have been extensively studied. They can be classified into three cases:

1. For certain families of polynomials, it is known that finding a root of them over $\mathbb{Z}/N\mathbb{Z}$ is as hard as factorizing N. For example, the family of polynomials $\left\{f_a(x) = x^2 - a\right\}_{a\in(\mathbb{Z}/N\mathbb{Z})^\times}$ whose potential roots are the solutions for the quadratic residue problem [Rab79].

2. There are families of polynomials, where 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 Q, then we can use LLL [LLL82]; 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 [Cop97]. But these families of polynomials only form a negligible portion of all the polynomials with polynomially bounded degrees.

3. The majority of the polynomials seem to live in the third case, where finding a root is conjectured to be hard, however the hardness is not known to be based on integer factorization. Among them, some specific families are even conjectured to be unlikely to have a reduction from integer factorization. For example, when $\gcd(3, \phi(N)) = 1$, the family of polynomials $\{f_a(x) = x^3 - a\}_{a \in (\mathbb{Z}/N\mathbb{Z})^\times}$ is conjectured to be hard to solve, and unlikely to be as hard as integer factorization [BV98].

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

5.1.1 Feasible information from a single j-invariant.

Let $N = pq$, as before, where p and q are large primes. From any $j \in \mathbb{Z}/N\mathbb{Z}, j \neq 0, 1728$, we can easily 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 $a = 3j(1728 - j), b = 2j(1728 - j)^2$. However, this method does not guarantee that the curve belongs to a specific isomorphism class (there are four of them). By choosing a value $u \in (\mathbb{Z}/N\mathbb{Z})^\times$ and let $a^* = u^4 a, b^* = u^6 b$ one gets the coefficient of another curve with the same j-invariant, each belonging to one of the four isomorphism classes.

On the other hand, 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 a point P over $E(\mathbb{Z}/N\mathbb{Z})$ is feasible solely using the x-coordinate of P (cf. [Dem93]). 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 [Len87] 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 to begin with. 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 $\mathbb{Z}/N\mathbb{Z}$ [KK98]. 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 [KK98] 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.

5.1.2 Computing explicit isogenies over $\mathbb{Z}/N\mathbb{Z}$ given more than one j-invariant

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 3.3 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.

Of course 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.

Recall that an isogeny $\phi : E_1 \to E_2$ 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*. 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 the kernel polynomial $h(x)$ of the isogeny ϕ, computing j_2, $f(x)$, and $g(x, y)$ is feasible over $\mathbb{Z}/N\mathbb{Z}$ via Vélu's formulae [Vél71].

Kernel polynomials are factors of the ℓ^{th} division polynomial Ψ_ℓ of E_1. Given a Weierstrass equation for E_1, computing Ψ_ℓ (of degree $\frac{1}{2}(\ell-1)$) is feasible over $\mathbb{Z}/N\mathbb{Z}$ using the recursive formulas. Finding a factor of Ψ_ℓ, on the other hand, seems to be hard.

If we are given two j-invariants $j_1, j_2 \in \mathbb{Z}/N\mathbb{Z}$ such that $\Phi_\ell(j_1, j_2) = 0 \pmod N$, then it is feasible to recover the curves E_1, E_2, together with an explicit rational polynomial that represents the isogeny from E_1 to E_2. This is simply because the arithmetic operations involved in computing the kernel polynomial $h(x)$ mentioned in [CM94, Sch95, E+98] works over $\mathbb{Z}/N\mathbb{Z}$ by reduction $\bmod N$, and does not require the factorization of N.

Claim 5.1. *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 we can recover the rational expression of ϕ. However, it is not clear how to use the explicit expression of ϕ to 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 0 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

The hardness of infeasible inversion of our group representation in fact relies on the hardness of the following generalization of the (ℓ, ℓ^2)-isogenous neighbor problem (cf. Definition 3.6). Let the modulus $N = pq$, where the large prime factors p and q are hidden. Let ℓ be the degree of the isogenies. For a polynomially large $k \in \mathbb{N}$ and a given a sequence of integers j_0, j_1, j_2, ... j_k such that $\Phi_\ell(j_{i-1}, j_i) = 0 \pmod N$ for all $i \in [k]$, the problem asks to find an integer j_{-1} such that $\Phi_{\ell^{i+1}}(j_{-1}, j_i) = 0 \pmod N$ for all $i \in [k]$. In addition, for all $i \in \{0, 1, ..., k\}$ the endomorphism rings of $E_{i,\mathbb{F}_p}$, $E_{i,\mathbb{F}_q}$ are isomorphic to an imaginary quadratic order $\mathcal{O}$ ($\mathcal{O}$ is supposed to be hidden for reasons to be explained later). See Figure 3 for the pictorial description of the problem.

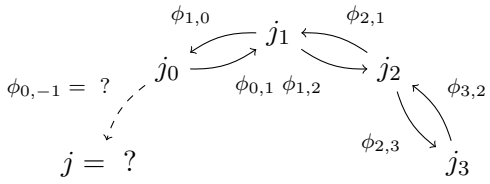

Figure 3: A pictorial description of the (ℓ, ℓ^2)-isogenous neighbor problem.

From the discussion in § 5.1.2 we know that it is feasible to compute the kernel polynomials of the isogenies $\phi_{i,i+1}$ and their duals $\phi_{i+1,i}$, for $i = 0, ..., k-1$. Denote these kernel polynomials by $h_{i,i+1}$ and $h_{i+1,i}$ respectively. If one can compute the kernel polynomial $h_{0,-1}$ of the isogeny $\phi_{0,-1}$ that maps from j_0 to j one can then solve the (ℓ, ℓ^2)-problem. Since $h_{0,1}$ can be recovered, there is a chance of obtaining $h_{0,-1}$ from $h_{0,1}$. As mentioned in Remark 2.5, over a finite field, the two kernel polynomials $h_{0,-1}$ and $h_{0,1}$ of the two horizontal isogenies can be explicitly related via the Frobenius endomorphism. However it is not clear how to use the relation over $\mathbb{Z}/N\mathbb{Z}$.

Another attempt is to map the kernel $G_{1,2}$ of $\phi_{1,2}$ and $G_{1,0}$ of $\phi_{1,0}$ from E_1 to E_0, and see if one of them corresponds to $\ker(\phi_{0,-1})$. Clearly $\phi_{1,0}(G_{1,0}) = O$, which gives nothing; applying $\phi_{1,0}$ on $G_{1,2}$ gives $\ker(\phi_{0,1})$, since $\phi_{0,1} \circ \phi_{1,0}(G_{1,2}) = [\ell]G_{1,2} = O$, so it does not give the desired subgroup either.

5.2.1 Attack by solving the Hilbert class polynomial and its implications

Let D be the discriminant of the imaginary quadratic order $\mathcal{O}$ that we are working with. If computing the Hilbert class polynomial H_D is feasible, then we can 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.

We remark that this attack seems to be infeasible when D is chosen to be super-polynomial.

5.2.2 Deciding the direction of an isogeny on the volcano

Fix an isogeny volcano $G_\ell(\mathbb{F}_q)$ over a finite field $\mathbb{F}_q$. As we mentioned, given a j-invariant j on $G_\ell(\mathbb{F}_q)$ there are efficient algorithms that output all the ℓ-isogenous neighbors of j. When there are 1 or 2 neighbors, we know j is at the bottom of the volcano (the case of 2 neighbors corresponding to when the volcano consists just of the surface). When there are more than 2 (i.e. $\ell + 1$) neighbors, can we decide which isogeny is ascending, horizontal, or descending? The method mentioned in [Koh96, FM02, Sut13a] takes a trial and error approach. It picks a random neighbor and goes forward, until the path reaches one of the terminating conditions. For instance, if it reaches a point where there is only one neighbor, then that means the path is descending; if the path forms a loop, or takes longer than the estimated maximum depth of the volcano, then the initial step is ascending or horizontal.

The only algorithm that is able to produce an isogeny with a designated direction is given by Ionica and Joux [IJ13]. They recognize an invariant related to the group structure of the curves lying on the same level of the volcano. The highlevel structure of the algorithm is as follows:

1. First decide the group structure via a pairing ([IJ13] uses reduced Tate pairing).
2. Then find the kernel subgroup with a property that depends on whether the desired isogeny is ascending, descending, or horizontal.

Either steps seem to carry out over $\mathbb{Z}/N\mathbb{Z}$. One of the main barriers is to compute (even the x-coordinate of) a point that sits in the specific subgroup of the curve.

The precise descriptions of the invariant and the algorithm are rather technical. So we only sketch the main theorem from [IJ13], skipping many technical details. Let $n \geq 0$, let $E[\ell^n](\mathbb{F}_{q^k})$ be the subgroup of points of order ℓ^n defined over an extension field over $\mathbb{F}_q$. Let $E[\ell^\infty](\mathbb{F}_q)$ be the ℓ-Sylow subgroup of $E(\mathbb{F}_q)$.

Let m be an integer such that $m \mid \#E(\mathbb{F}_q)$. Let k be the embedding degree, i.e. the smallest integer that $m \mid q^k - 1$. Let $T_m : E[m](\mathbb{F}_{q^k}) \times E(\mathbb{F}_{q^k})/mE(\mathbb{F}_{q^k}) \to \mu_m$ be the reduced Tate pairing, where μ_m denotes the m^{th} roots of unity. Furthermore, define the symmetric pairing

$$S(P, Q) = (T_{\ell^n}(P, Q)\, T_{\ell^n}(Q, P))^{1/2},$$

and call $S(P, P) = T_{\ell^n}(P, P)$ the self-pairing of P.

Theorem 5.2 (The main theorem of [IJ13], informally)**.** *Let E be an elliptic curve defined a finite field $\mathbb{F}_q$ and let $E[\ell^\infty](\mathbb{F}_q)$ be isomorphic to $\mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$ with $n_1 \geq n_2 \geq 1$. Then*

- *Suppose P is a ℓ^{n_2}-torsion point such that $T_{\ell^{n_2}}(P, P)$ is a primitive ℓ^{n_2}th root of unity. Then the ℓ-isogeny whose kernel is generated by $\ell^{n_2-1}P$ is descending.*
- *Suppose certain condition holds, and let P be a ℓ^{n_2}-torsion point with degenerate self-pairing. Then the ℓ-isogeny whose kernel is generated by $\ell^{n_2-1}P$ is either ascending or horizontal. Moreover, for any ℓ^{n_2}-torsion point Q whose self-pairing is non-degenerate, the isogeny with kernel spanned by $\ell^{n_2-1}Q$ is descending.*

Carrying out the algorithm of finding an isogeny with a specific direction (say horizontal) requires finding (even merely the x-coordinate of) a point in the designated subgroup. Over a finite field, such points can be found efficiently by picking a random point R and compute $P = \frac{\#(E(\mathbb{F}_q))}{r}R$, where r is the order of the designated subgroup; then test if the candidate point P lies in the correct subgroup by taking pairing. Over $\mathbb{Z}/N\mathbb{Z}$ it seems hard even to find a point with a specific order since we do not know $\#E(\mathbb{Z}/N\mathbb{Z})$. There are additional technicalities such as computing pairing over $\mathbb{Z}/N\mathbb{Z}$ (cf. [GM05]), which require further investigations after we get around the first barrier of finding points in the specific subgroups.

In our application we are mostly interested in finding the unvisited neighbor over $\mathbb{Z}/N\mathbb{Z}$ that lies on the same level of the volcano, which equals to finding the horizontal isogeny of a given curve. As we conjectured, the algorithm in [IJ13] do not extend to $\mathbb{Z}/N\mathbb{Z}$. So if the task of finding an isogeny with a designated direction is feasible over $\mathbb{Z}/N\mathbb{Z}$, then it is likely to imply a new algorithm of the same task over the finite field, which seems to be challenging on its own.

5.2.3 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 [E+98, 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[5] 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. [Koh96]). $\mathbb{H}/\Gamma_1(\ell)$ 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 5.3. *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] Recall that a modular curve is a quotient of the extended upper half plane by a congruence subgroup, and a congruence subgroup is a subgroup of $\mathrm{SL}_2(\mathbb{Z})$, which contains $\Gamma(\ell)$.

Let us also remark that for special values of ℓ the (ℓ, ℓ^2)-isogenous neighbor problem may be more prone to characteristic zero attacks (although we do not know of such an attack). For instance, for those ℓ for which $\mathbb{H}/\Gamma_0(\ell)$ (more precisely, its compactification $X_0(\ell)$) is genus 0 or 1, it (may) have many rational points. These points, in turn, can be used to get points over $\mathbb{Z}/N\mathbb{Z}$ without knowing the factorization of N. Nevertheless, we may just avoid these values in the system. As mentioned above, we currently do not see an attack based on this. This point is brought up just as an extra precautionary measure.

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 4.4:

$$\mathsf{encoding}(x) = (L_x; T_{x,1}, ..., T_{x,w_x}) = ((p_{x,1}, ..., p_{x,w_x}); (j_{x,1,1}, ..., j_{x,1,e_{x,1}}), ..., (j_{x,w_x,1}, ..., j_{x,w_x,e_{x,w_x}})).$$

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 (due to Claim 5.1) between each pair of the neighboring j-invariants. We assume the security goal is to prevent the feasibility of computing the canonical encoding of the inverse of any of the encoded group elements presented in the system, which is sufficient (but not necessary) for the applications presented in this paper.

5.3.1 Preventing trivial leakage of inverses from encodings

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 the composable encoding of x be $((p_x); (j_x))$ and the composable encoding of y be $((p_y); (j_y))$.

Naively, a composable encoding of $z = y \circ x^{-1}$ could then 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.

5.3.2 Hiding the class group invariants: why and how

With the current choice of parameters (cf. Choices 4.9 and 4.10), the discrete-log problem over $\mathcal{CL}(D)$ can be solved efficiently once $h(D)$ is given, and conversely $h(D)$ can be recovered from D or any basis of a lattice Λ' of dimension m' such that $\mathbb{Z}^{m'}/\Lambda' \simeq \mathcal{CL}(D)$. In the applications, on the other hand, we do need the discrete-log problem over $\mathcal{CL}(D)$ to be hard to support the infeasibility of inversion, since we assume polynomially many composable encodings are published. Recall from Lemma 2.7 that the graph $\mathcal{G}_{\mathcal{O},m}(\mathbf{k})$ for a sufficiently large m is an expander under GRH, which means it is reasonable

to assume that the closure of the composition of polynomially many j-invariants covers all the $h(D)$ j-invariants. Also, finding a composition of the published j-invariants that reaches a specific vertex (e.g. the vertex that represents the inverse of some encoded element) in the isogeny graph is feasible by solving the discrete-log problem over $\mathcal{CL}(D)$. So we do need to hide the discriminant D, the class number $h(D)$, and any lattice Λ' defined above.

We remark that if it is feasible to choose a super-polynomially large square-free discriminant D with a trapdoor τ_D so that solving discrete-log over $\mathcal{CL}(D)$ given D is hard but feasible given the trapdoor τ_D, then we might be able to maintain the security of the system even if D and $h(D)$ are public. This would avoid all the complications in hiding the class group invariants, however currently we do not know of such a method, so we need to hide the class group invariants.

The possibility of recovering the discriminant D. Recall that for an elliptic curve over a finite field $\mathbb{F}_q$, the discriminant D can be obtained by first computing the trace t of Frobenius, and then computing the integers v and D_0 such that $v^2 D_0 = t^2 - 4q$, and D_0 is square-free. Then $D = u^2 D_0$, where $u \mid v$. When v is smooth D can be recovered efficiently.

Over $\mathbb{Z}/N\mathbb{Z}$, on the other hand, since we loose the notion of the Frobenius automorphism, we do not know how to apply the previous algorithm. Also, just a set of j-invariants over $\mathbb{Z}/N\mathbb{Z}$, corresponding to curves with the same endomorphism rings over $\mathbb{F}_p$ and $\mathbb{F}_q$, does not seem to allow us to recover the discriminant.

However, we do know that the range of D is bounded by $|D| \leq 4p$ and $|D| \leq 4q$, so that $|D| < \sqrt{N}$. From the encodings we also learn the set $\mathcal{P}_B = \left\{ p \leq B \mid p \text{ is a prime, } \left(\frac{D}{p}\right) = 1 \right\}$, where $B \in \mathsf{poly}(\lambda)$. This brings us to the following problem of independent interest:

Definition 5.4. *Given an integer N and a set of primes $\mathcal{P} = \{p_1, ..., p_m\}$, find a negative value D such that $|D| < \sqrt{N}$ and $\left(\frac{D}{p}\right) = 1$ for all $p \in \mathcal{P}$.*

Note that each condition $\left(\frac{D}{p}\right) = \pm 1$ cuts the possible D's roughly by half. Hence, for a sequence $\epsilon_1, \epsilon_2, \cdots, \epsilon_{\omega(\log(N))}$, $\epsilon_j \in \{\pm 1\}$, with high probability there is at most one D with $|D| < \sqrt{N}$ that satisfies $\epsilon_j = \left(\frac{D}{p_j}\right)$. To find such a D, on the other hand, is a separate problem and the only known methods require $\sqrt{|D|}$ of the values of the sequence $\{\epsilon_j\}$ (cf. pg. 6 of [Hof14], also see pg. 14 of loc. cit. and [GH93] for the analogue of the same problem in the context of modular forms)

We also remark that the following similar problem, first mentioned by Damgård [Dam88], is conjectured to be hard:

Definition 5.5 (Problem P1 in [Dam88])**.** *The Legendre sequence with length ℓ and starting point a is the ± 1 sequence*

$$L = \left(\frac{a}{p}\right), \left(\frac{a+1}{p}\right), ..., \left(\frac{a+\ell}{p}\right).$$

Given L (with a polynomial ℓ) but not a and p, the problem asks to determine $\left(\frac{a+\ell+1}{p}\right)$.

If Problem 5.4 can be solved efficiently, an obvious safeguard is to choose less than $\log(|D|)$ many primes for the degrees of the isogenies involved in the composable encodings. However, it is not always possible to restrict the number of distinct prime degrees in applications. In that case, for the parameter choice II (cf. 4.10), where D is not square-free, we can first choose an enlarged set of k polynomially large primes $R = \{p_1, ..., p_k\}$ (satisfying the same conditions in 4.10), and let R' be a

random subset of R. We can then set $D = (\prod_{p \in R'} p)^2 D_0$, where D_0 is a polynomial size fundamental discriminant. Then in the composable encodings, we avoid the primes from the entire set R as the degrees of the isogenies.

The possibility of revealing Λ'. Revealing any full-rank (not necessarily short) basis $\mathbf{B}'$ of a relation lattice Λ' of dimension m' such that $\mathbb{Z}^{m'}/\Lambda' \simeq \mathcal{CL}(D)$ implies the disclosure of the class number $h(\mathcal{O}) = |\det(\mathbf{B}')|$. So we should prevent leaking any full-rank bases of such lattices.

An immediate consequence is that we cannot give out many non-trivial encodings of $1_{\mathbb{G}}$ (where $\mathbb{G} = \mathcal{CL}(\mathcal{O})$) that form a full-rank basis of a potential relation lattice Λ'. Recall that the trivial (i.e. canonical) encoding of $1_{\mathbb{G}}$ is j_0. A non-trivial encoding of $1_{\mathbb{G}}$ can be obtained by composing encodings which lead to a non-zero exponent vector. As an example, suppose that the encodings of $x, y, z \in \mathcal{CL}(\mathcal{O})$ share the same prime generation set of norm from L, and $x \circ y \circ z = 1_{\mathbb{G}}$, and denote them by $\mathsf{encoding}(v) = (L; T_{v,1}, ..., T_{v,|L|})$, $v \in \{x, y, z\}$. Then, the following vector is in Λ'

$$\mathbf{e}_{x \circ y \circ z} = (|T_{x,1}| + |T_{y,1}| + |T_{z,1}|, ..., |T_{x,|L|}| + |T_{y,|L|}| + |T_{z,|L|}|).$$

Note that in the example we are not required to compute the composable encoding of $x \circ y \circ z$. We only need to read off the exponents from the lengths of $T_{*,\star}$.

In the applications we do face the situation where the general security setting does not restrict the number of non-trivial encodings of $1_{\mathbb{G}}$. A countermeasure is to enforce each distinct non-zero encoding of $1_{\mathbb{G}}$ to have an new ideal with distinct prime norm, so that the dimension of the potential basis is larger than the possible number of vectors (relations) to be collected. We will illustrate how to apply this countermeasure in the applications.

Even for the encodings of non-identity elements, sampling the vector $\mathbf{e}$ from an arbitrary distribution over the cosets of Λ' might leak a basis. For example, [NR06] shows that if there are enough short vectors from the parallelepiped of a short basis, then one can find the parallelepiped and, therefore, recover the basis.

The countermeasure is to sample the vector $\mathbf{e}_x \in \mathbb{Z}^{w_x}$ from the discrete-Gaussian distribution [GPV08]. The sampler is known of being basis-independent.

Formally, for any $\sigma \in \mathbb{R}^+$, $\mathbf{c} \in \mathbb{R}^n$, define the (non-normalized) Gaussian function on $\mathbb{R}^n$ with center $\mathbf{c}$ and standard deviation σ as follows:

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \rho_{\sigma,\mathbf{c}}(\mathbf{x}) = e^{-\pi \|\mathbf{x}-\mathbf{c}\|^2/\sigma^2}$$

For any n-dimensional lattice Λ, define the discrete Gaussian distribution over Λ as:

$$\forall \mathbf{x} \in \Lambda,\ D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\Lambda)},$$

where $\rho_{\sigma,\mathbf{c}}(\Lambda) := \sum_{\mathbf{y} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{y})$ is the normalization factor.

With these definitions we have the following:

Lemma 5.6 ([MR07]). *Let $\mathbf{B}$ be a basis of an n-dimensional lattice Λ, and let $\sigma \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\log n)$, then $\Pr_{\mathbf{x} \leftarrow D_{\Lambda,\sigma,\mathbf{0}}}[\|\mathbf{x}\| \geq \sigma \cdot \sqrt{n} \vee \mathbf{x} = \mathbf{0}] \leq \mathsf{negl}(n)$.*

Lemma 5.7 ([GPV08, BLP+13]). *There is a p.p.t. algorithm that, given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda(\mathbf{B})$, $\mathbf{c} \in \mathbb{R}^n$, and $\sigma \geq \|\tilde{\mathbf{B}}\| \sqrt{\ln(2n+4)/\pi}$, outputs a sample from $D_{\Lambda,\sigma,\mathbf{c}}$.*

Note, however, that sampling from discrete-Gaussian distribution could lead to a vector with negative entries. Having negative entries as the exponent vector might trivially leak the inverse of other encodings (as mentioned in Section 5.3.1). There are at least two solutions to this problem. One is to sample a vector $\mathbf{v}_0$ with positive large entries in Λ', then add $\mathbf{v}_0$ on any other vectors $\mathbf{v}'$ sampled from discrete-Gaussian so that $\mathbf{v}_0 + \mathbf{v}'$ have all positive entries (this solution does leak one vector $\mathbf{v}_0$ in Λ'). The other solution is to pick at least one new prime ideal in the generation set of each composable encoding, so that the inverse won't be trivially obtained from the other encodings.

The possibility of leaking $h(D)$ from other sources. We have discussed the possibilities of leaking D and a basis of $\Lambda_{\mathcal{O}}$ from the encodings and the corresponding countermeasures. It remains to check whether there are other possibilities of leaking $h(D)$.

In Choice 4.10 of the parameter setting, $h(D)$ is always set to be polynomially smooth. The immediate consequence is that in the encodings, we shall not use prime degrees ℓ such that the order of the ideal class $[(\ell, b, \cdot)]$ is polynomially large in $\mathcal{CL}(D)$, to avoid the risk of unnecessarily leaking any factors of $h(D)$.

Additionally, we ask:

1. Given an encoding, is it feasible to recognize that it encodes an element of polynomial order in the group $\mathcal{CL}(\mathcal{O})$?
2. Given an encoding of an element that is known to be of polynomial order, is the order explicitly revealed (instead of having an $1/\mathsf{poly}(\lambda)$ chance of being guessed correctly)?

If self-composition is feasible, then the answers to both questions are yes. But for our construction, self-composition of the encoded group elements is infeasible.

Denote the degree of an encoding $\mathsf{encoding}(x)$ by d and the order of x in $\mathcal{CL}(\mathcal{O})$ by r. Let the canonical encoding of $\mathsf{encoding}(x)$ be j_x. In the special case where d^r is polynomially large, then we can efficiently recognize that $\mathsf{encoding}(x)$ has order r by first guessing d^r and then testing whether $\Phi_d(j_0, j_x) = 0 \pmod N$, and $\Phi_{d^{r-1}}(j_0, j_x) = 0 \pmod N$. The presence of such an encoding then leaks r as a factor of the group order.

One way of minimizing the possibility of having an element with small order is to choose the prime factors of $h(\mathcal{O})$ to be as large as possible. For example, we can choose D_0 and the odd prime factors of $f_i - 1$ (where f_i is a prime factor of the conductor) to be larger than $O(\lambda^3)$.

5.4 Miscellaneous

On the decisional version of the inversion problem. Like the typical hard problems in cryptography, a search problem usually comes with a decisional variant. For the hardness of inversion, the natural way of defining the decision problem is to say that given a group element x, it is hard to decide whether a string s represents x^{-1} or a random group element. While in the ideal interface of (T)GII the decisional variant is easy, simply due to the fact that one can compose x and s, then check if the result is equal to $1_{\mathbb{G}}$ or not. In our concrete instantiation, the following variant of the decisional problem has a chance to be hard:

Definition 5.8 (Decisional inversion problem)**.** *Given the public parameter* PP *and a (composable) encoding* $\mathsf{encoding}(x)$*, decide whether a string s represents the canonical encoding of x^{-1} (namely $j(x^{-1} * E_0)$) or a random value in $(\mathbb{Z}/N\mathbb{Z})^\times$.*

Given that the canonical encoding is not composable, there is a chance that the decisional problem is hard. However, when the degree of $\mathsf{encoding}(x)$ is a polynomial d, Problem 5.8 is easy, since we can test whether $\Phi_d(j_0, s) = 0 \pmod N$ or not (in contrast, we conjecture the search problem is hard even for polynomial degree encodings). If the encoding is of super-polynomial degree, then the decisional problem seems hard.

We remark that the applications of this paper do not necessarily rely on the hardness of the decisional version of the inversion problem, so the cryptanalysis effort we spend on the decisional problem is not as much as the search problem.

Rational points on $X_0(\ell)$. Rational points over $X_0(\ell)$ give solutions to $\Phi_\ell(x, y) = 0$ over $\mathbb{Z}/n\mathbb{Z}$ for any $n \in \mathbb{N}$, unless the denominators of x or y are not in $(\mathbb{Z}/n\mathbb{Z})^\times$. Recall that the genus of the modular curve $X_0(\ell)$ grows linearly with ℓ, hence for large ℓ, by Faltings' theorem, there are only finitely many rational points on $X_0(\ell)$. So for large ℓ, one cannot hope to find ℓ-isogenous neighbors over $\mathbb{Z}/N\mathbb{Z}$ by first finding points over $\mathbb{Q}$ and then reducing them mod N. For small ℓ, on the other hand, there are rational points that seem to be easy to find.

Let $P = (x, y)$ be such a ($\mathbb{Q}$-rational) point on $X_0(\ell)$, where neither x nor y is equal to 0 or 1728. The presence of these points, on one hand, allows one to easily find one j-isogenous neighbor of x mod N (if the denominator of x is invertible in $\mathbb{Z}/N\mathbb{Z}$), that is y mod N.

On the other hand, for an attack towards the (ℓ, ℓ^2)-neighbor problem one would like to find two $\mathbb{Q}$-rational points P, Q sharing a common coordinate (i.e. either $x_P = x_Q$ or $y_P = y_Q$). We remark that having such $\mathbb{Q}$-rational points is highly unlikely for large ℓ. More precisely, wlog assume that $x_P = x_Q$. Then y_P and y_Q correspond to the j-invariants of ℓ^2-isogenous curves. They, moreover, are both in $\mathbb{Q}$, and hence the point (y_P, y_Q) defines a $\mathbb{Q}$-rational point on $X_0(\ell^2)$. If $\ell > 7$, $X_0(\ell^2)$ has genus strictly great than 1 and therefore has finitely many rational points. Therefore, for $\ell > 7$ even if one can find a pair of Q-rational points P and Q on $X_0(\ell)$, it is highly unlikely that they would share a common coordinate.

Although we do not see an immediate attack through rational points on $X_0(\ell)$, we can always take $\ell > 7$ for extra precaution.

5.5 Summary

We summarize the potential attacks and the countermeasures.

The adversary is given the public parameters N, j_0, and polynomially many composable encodings, each containing several j-invariants connected by isogenies of polynomial degrees. Following Claim 5.1 we can recover all the explicit isogenies between neighboring j-invariants. Given that we choose all the prime degrees ℓ of the isogenies to be ≥ 5 (or > 7 for extra precaution), the explicit isogenies do not trivially leak the x-coordinates of the points in the kernel of the isogeny. It is not clear to us if there are other ways of computing x-coordinates of the points in any specific subgroup of size ℓ.

We should avoid leaking the number of points on those elliptic curves over $\mathbb{F}_p$ and $\mathbb{F}_q$ with endomorphism ring $\mathcal{O}$. Otherwise the adversary can pick a random x as the x-coordinate of a random point on the curve, and run Lenstra's factoring algorithm to factorize N.

Central to the hardness of inversion is the (ℓ, ℓ^2)-isogenous neighbor problem over a composite modulus N with unknown factorization. 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, that of [IJ13], does not seem to work over $\mathbb{Z}/N\mathbb{Z}$.

Under GRH the isogeny graph is an expander, which means given polynomially many composable encodings, it is reasonable to assume that the closure of the composition covers all the $h(D)$ j-invariants. However, finding a path of composition to reach a specific point (e.g. the inverse of some encoding) may still require solving the discrete-log problem over $\mathcal{CL}(D)$. Under the current choice of parameters, the discrete-log problem over $\mathcal{CL}(D)$ can be solved efficiently once $h(D)$ is given, and $h(D)$ can be recovered efficiently given D or any relation lattice Λ' of dimension m' such that $\mathbb{Z}^{m'}/\Lambda' \simeq \mathcal{CL}(D)$. So we should prevent leaking any of Λ', D, or $h(D)$.

The discriminant D cannot be polynomially large for two reasons: First, if $D \in \mathsf{poly}(\lambda)$, then we can guess D and compute the class number in polynomial time, which enables us to efficiently solve the discrete-log problem over $\mathcal{CL}(D)$. Second, we can also compute the Hilbert class polynomial H_D in polynomial time and therefore solve the (ℓ, ℓ^2)-isogenous neighbor problem.

Even if D is super-polynomial, it should be kept hidden. This is because under either of the two choices of the system parameters, anyone can compute the class number $h(D)$ and solve the discrete-log problem over $\mathcal{CL}(D)$ if D is given. Since D has to be hidden, we cannot give out a plaintext group element of $\mathcal{CL}(D)$ that allows efficient group operation, e.g. the quadratic form representation of an ideal class $[(a, b, c)]$ (cf. Section 2.1), since the discriminant D can be read off from $b^2 - 4ac$. Given that we shall hide the class number $h(D)$, we shall also hide any of Λ'. We do have a stateful universal solution for this problem, where the encoding algorithm keeps track of the prime ideals used in the prior encodings. However, at the moment we do not have a stateless universal solution.

Finally, let us remark that if we were able to choose a super-polynomially large square-free discriminant D with a trapdoor τ_D, so that solving discrete-log over $\mathcal{CL}(D)$ is hard if one is given only D, but feasible if one is given the trapdoor τ_D, then we can construct a system with D and $h(D)$ being public, which would avoid all the complications in hiding D and $h(D)$.

6 Broadcast encryption

A broadcast encryption scheme allows the encrypter to generate ciphertexts that are decryptable by a designated subset of all the receivers. A trivial solution for broadcast encryption is to simply encrypt the message many times for each user's decryption key. Although this does provide a solution, it clearly is not effective as the number of receivers grow. A meaningful broadcast encryption scheme needs to be more efficient than the trivial solution on either of the cost of encryption, size of the public parameters, or the sizes of users' decryption keys (cf. [FN93]).

We give a concrete instantiation of the broadcast encryption scheme of Irrer et al. [ILOP04], which was designed under the ideal interface of GII. Our instantiation supports private-key encryption and allows any numbers of users to collude. The encryption overhead and the users' secret keys are independent of the total number of the users n, however the public parameter size is linear in n. Let us remark that our aim here is to give a concrete application of our construction and hence we did not try to optimize the public parameter overhead. This is not the most efficient scheme in theory and other schemes that achieve $\log(n)$ public parameter overhead are available if one assumes the security of multilinear maps [BWZ14] or iO [Zha14]. We leave the challenge of achieving smaller public parameter blowup from GII or TGII to the interested reader.

6.1 Definition

A private-key broadcast encryption scheme consists of a tuple of efficient algorithms:

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm takes as input the security parameter 1^λ, then generates the public parameters PP and the master secret key MSK.
- $\mathsf{Gen}(\mathsf{PP}, \mathsf{MSK}, u)$: The user key generation algorithm includes the user ID, u, in the list of users $\mathcal{U}$. It generates a secret key SK_u and (possibly) a public key PK_u for user u. PK_u is included in the public parameters.
- $\mathsf{Enc}(\mathsf{PP}, \mathsf{MSK}, \Gamma, m)$: The encryption algorithm takes as input a polynomial size set $\Gamma \subseteq \mathcal{U}$ of recipients and produces (using $\mathsf{PP}, \mathsf{MSK}, \Gamma$) a key K of a symmetric-key encryption scheme. Denoting the encryption of the message m under K by $\mathsf{CT}_{K,m}$, it outputs $\mathsf{CT} = (\Gamma, \mathsf{CT}_{K,m})$.
- $\mathsf{Dec}(\mathsf{PP}, u, \mathsf{SK}_u, \mathsf{CT})$: The decryption algorithm first parses CT as $(\Gamma, \mathsf{CT}_{K,m})$ and derives the symmetric-key K from $\mathsf{PP}, \mathsf{SK}_u$, and Γ. It then uses K to decrypt $\mathsf{CT}_{K,m}$.

The scheme is said to be correct if for all subsets of the users $\Gamma \subseteq \mathcal{U}$ and for all messages m, any user u from the set Γ decrypts the correct message, i.e. $\mathsf{Dec}(\mathsf{PP}, u, \mathsf{SK}_u, \Gamma, \mathsf{Enc}(\mathsf{PP}, \mathsf{MSK}, \Gamma, m)) = m$.

For security we consider the simplest form of the key-recovery attack, defined by the following game between an adversary and a challenger.

1. The challenger runs the setup algorithm to generate a master secret key MSK and the public parameters. Challenger then picks a set of users $\mathcal{U}$ and generates the public parameters and the users' decryption keys. The adversary is given $\mathcal{U}$ and the public parameters.
2. The adversary picks a subset $\Gamma \subseteq \mathcal{U}$ of receivers where it wants to attack. The challenger gives all users' keys SK_u for $u \notin \Gamma$.
3. The challenger runs the Encrypt algorithm to obtain $\mathsf{Enc}(\mathsf{PP}, \mathsf{MSK}, \Gamma, m)$. The adversary is given $\mathsf{CT} = (\Gamma, \mathsf{CT}_{K,m})$.
4. The adversary outputs a key K' and wins the game if $K' = K$, loses otherwise.

A broadcast encryption scheme is said to be secure if for all the polynomial time adversary the advantage of winning the key-recover game is negligible.

6.2 A private-key broadcast encryption scheme from TGII

We first present the construction in the abstract syntax of a trapdoor group with infeasible inversion, then specify the detailed parameters in the instantiation using isogeny volcanoes.

Construction 6.1 (Broadcast encryption under an abstract TGII)**.** *Given a trapdoor group with infeasible inversion* TGII *and a symmetric-key encryption scheme* Sym*, construct a broadcast encryption scheme* BE *as follows:*

- $\mathsf{BE.Setup}(1^\lambda)$*: The setup algorithm takes as input the security parameter* 1^λ*, runs the TGII parameter generation algorithm to produce the public parameters* $\mathsf{TGII.PP}$ *and the trapdoor* τ *of a group* $\mathbb{G} = (\circ, 1_\mathbb{G})$*. Then sample a random element* $s \in \mathbb{G}$*. The public parameter* $\mathsf{BE.PP}$ *is set to be* $\mathsf{TGII.PP}$*. The master secret key* $\mathsf{BE.MSK}$ *includes* τ *and* s*.*

Construction	|param|	|user key|	|CT|	based on	PK	CR
[FN93]	$O(t^2 n \log n)$	$O(t \log^2 t \log n)$	$O(t^2 \log^2 t \log n)$	RSA assumption	No	$\leq t$ users
[ILOP04]	$O(n)$	$O(1)$	$O(1)$	Ideal GII	No	Arbitrary
[BGW05] I	$O(n)$	$O(1)$	$O(1)$	Bilinear maps	Yes	Arbitrary
[BGW05] II	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(1)$	Bilinear maps	Yes	Arbitrary
[BWZ14]	$O(\log n)$	$O(\log n)$	$O(\log n)$	Mmaps	Yes	Arbitrary
[Zha14]	$O(\log n)$	$O(\log n)$	$O(\log n)$	iO	Yes	Arbitrary
This work	$O(n)$	$O(1)$	$O(1)$	A concrete TGII	No	Arbitrary

Figure 4: A brief summary of the existing collusion resistant broadcast encryption schemes. n represents the number of users; "PK" stands for supporting public key encryption; "CR" stands for being collusion resistant. All the parameters ignore the (possibly multiplicative) $\mathsf{poly}(\lambda)$ factors.

- $\mathsf{BE.Gen}(\mathsf{BE.PP}, \mathsf{BE.MSK}, u)$*: The user secret key generation algorithm parses* τ, s *from* $\mathsf{BE.MSK}$*. It samples a random element* $x \in \mathbb{G}$*, computes* $\mathsf{encoding}(x) \leftarrow \mathsf{TGII.TrapSam}(\mathsf{TGII.PP}, \tau, x)$ *as the user's public key* PK_u*; computes* $\mathsf{encoding}(s \circ x) \leftarrow \mathsf{TGII.TrapSam}(\mathsf{TGII.PP}, \tau, s \circ x)$ *and treats it as the user's secret key* SK_u*.*

- $\mathsf{BE.Enc}(\mathsf{BE.PP}, \mathsf{BE.MSK}, \Gamma, m)$*: The encryption algorithm takes as input a polynomial size set* $\Gamma \subseteq \mathcal{U}$ *of recipients and a message* m*. It first computes the message encryption key* $K = \left(\prod_{i \in \Gamma} x_i\right) \circ s$*, then computes* $\mathsf{Sym.CT}_{K,m} := \mathsf{Sym.Enc}(K, m)$ *and outputs* $\mathsf{BE.CT} = (\Gamma, \mathsf{Sym.CT}_{K,m})$*.*

- $\mathsf{BE.Dec}(\mathsf{BE.PP}, u, \mathsf{SK}_u, \mathsf{CT})$*: The decryption algorithm extracts the set* Γ *from* $\mathsf{BE.CT}$*, takes the public keys* PK_i *for users* $i \in \Gamma \setminus u$*, the secret key* SK_u *for user* u*, and computes* $K' = \prod_{i \in \Gamma \setminus u} (\mathsf{PK}_i) \circ \mathsf{SK}_u$*. It then decrypts* $\mathsf{Sym.CT}_{K,m}$ *using* K'*.*

Instantiation from the concrete TGII and choices of parameters. To instantiate the scheme based on the concrete TGII, the encryption algorithm uses the canonical encoding of $K = \left(\prod_{i \in \Gamma} x_i\right) \circ s$ as the key for the symmetric encryption scheme. For $u \in \Gamma$, the decryption algorithm first computes the composable encoding of $K' = \prod_{i \in \Gamma \setminus u} (\mathsf{PK}_i) \circ \mathsf{SK}_u$, then extracts the canonical encoding of K' as the symmetric decryption key ($K' = K$ follows the uniqueness of the canonical encoding). To be more cautious, we can also apply a randomness extractor on K so as to derive a key that is statistically close to the uniform in distribution.

As mentioned in the construction and cryptanalysis sections (cf. Sections 4.3 and 5.3), currently we do not have a stateless universal solution for the distribution of the composable encodings. Here we provide the parameters for a broadcast encryption scheme, where the master encryption algorithm is stateful. We give the details for the case where the group is $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ for a non-maximal order $\mathcal{O}$ (i.e. Choice 4.10). The case where the group is $\mathcal{CL}(\mathcal{O})_{\text{odd}}$ for a maximal order $\mathcal{O}$ (Choice 4.9) follows follows similar lines.

- Choose a $D_0 < 0$, of large polynomial size (say $|D_0| \approx O(\lambda^3)$), such that $h(D_0)$ is a prime.

- For the conductor f: choose a set R of $2\log(\lambda)$ primes such that for all $r \in R$, $\frac{r - \left(\frac{D_0}{r}\right)}{2}$ is a polynomially large prime of size $\approx O(\lambda^3)$. Randomly pick $\log(\lambda)$ numbers from R, denote them by $f_1, ..., f_{\log(\lambda)}$. Set $f := \prod_{i \in [\log(\lambda)]} f_i$ to be the conductor. Let $\mathcal{O}$ be an order of discriminant

$D = D_0 f^2$ (choosing the factors of f from a random subset of R is mentioned as a safeguard in § 5.3.2).

- Let $m = O(\log(\lambda))$. Pick a set of ideal classes $S = \{C_i = [(\ell_i, b_i, \cdot)]\}_{i\in[m]}$, such that for all i $[(\ell_i, b_i, \cdot)]$ generates $\mathcal{CL}(\mathcal{O})_{\text{odd}}$, $7 < \ell_i \leq \mathsf{poly}(\lambda)$, and $\ell_i \notin R$. Generate a short basis of $\Lambda_{\mathcal{O}}$ (the relation lattice w.r.t. S) as usual.

- To generate the master secret key of the broadcast encryption scheme, pick a random element $s \in \mathcal{CL}(\mathcal{O})_{\text{odd}}$ using the trapdoor (note that we do not need to compute any encoding of s; the information needed from the trapdoor is the description of the class group $\mathcal{CL}(\mathcal{O})_{\text{odd}}$, which is not available from the public parameter).

- To generate each user u's public key, choose a set of $m_u = O(\log^2(N))$ prime ideals $S_u = \{[(\ell_{u,i}, b_{u,i}, \cdot)]\}_{i\in[m_u]}$, where the primes $\{\ell_{u,i}\}$ have never been used in the system. Then generate PK_u as a composable encoding of a random group element x_u by running $\mathsf{encoding}(x_u) \leftarrow \mathsf{RandomSam}(\tau, S_u)$.

- To generate each user u's secret key, choose a set of $m'_u = O(\log^2(N))$ prime ideals $S'_u = \left\{[(\ell'_{u,i}, b'_{u,i}, \cdot)]\right\}_{i\in[m'_u]}$, where the primes $\left\{\ell'_{u,i}\right\}$ have never been used in the system. Generate the composable encoding of a random group element x'_u: $\mathsf{encoding}(x'_u) \leftarrow \mathsf{RandomSam}(\tau, S'_u)$.

  Next, solve the discrete-log problem of $s \circ x_u \circ (x'_u)^{-1}$ under the generation set S, obtain a short solution $\mathbf{e}_u \in \mathbb{Z}^m$. Using this get a composable encoding of $s \circ x_u \circ (x'_u)^{-1}$. Then get the secret key of user u as a composable encoding of $s \circ x_u$:

$$\mathsf{SK}_u = \mathsf{encoding}(s \circ x_u) = \mathsf{Compose}(\mathsf{PP}, \mathsf{encoding}(s \circ x_u \circ (x'_u)^{-1}), \mathsf{encoding}(x'_u)).$$

Here all the public keys are encoded with relatively prime degrees. The encodings of secret keys do share common prime factors in their degrees, which does not create a problem for the decryption since the decryption algorithm only requires the composition of a secret key with many public keys (we also can encode the secret keys with relatively prime degrees, if it turns out to be necessary for security reasons).

In the encoding of user u's secret key we insert a random x'_u with fresh prime ideal generators for two reasons: First it prevents trivially leaking an inverse of SK_u. Note that solving discrete-log with the generator set S could lead to negative entries in the exponent vector $\mathbf{e} \in \mathbb{Z}^m$. Without the fresh prime ideals in $\mathsf{encoding}(x'_u)$, all the other generators in the encoding of SK_u are in the set S shared with other users' secret keys, so the inverse might be trivially leaked in the other encodings of the secret keys. The prime ideals in S'_u are used only once, which prevents trivially leaking any encoding of $(x'_u)^{-1}$.

The second reason is to prevent revealing any full-rank basis of a relation lattice Λ' of dimension m' such that $\mathbb{Z}^{m'}/\Lambda' \simeq \mathcal{CL}(\mathcal{O})$ (i.e. to address the problem mentioned in Section 5.3.2). More precisely, let us start from the following relation:

$$\mathsf{SK}_{u_1} \circ \mathsf{PK}_{u_2} \circ (\mathsf{SK}_{u_2} \circ \mathsf{PK}_{u_1})^{-1} = x_1 \circ s \circ x_2 \circ (x_2 \circ s \circ x_1)^{-1} = 1_{\mathcal{CL}(\mathcal{O})}.$$

This means if the users u_1 and u_2 collude, they can read off the exponents from the encodings of SK_{u_1}, PK_{u_2}, SK_{u_2}, and PK_{u_1} to form a non-zero vector in the potential Λ'. Note that we are not computing a valid encoding of $(\mathsf{SK}_{u_2} \circ \mathsf{PK}_{u_1})^{-1}$, which is supposed to be hard.

In general, let n be the number of users in the system. As above, we can obtain vectors in some Λ' from the exponents of the following relations:

$$\mathsf{SK}_{u_1} \circ \mathsf{PK}_{u_i} \circ (\mathsf{SK}_{u_i} \circ \mathsf{PK}_{u_1})^{-1} = x_1 \circ s \circ x_i \circ (x_i \circ s \circ x_1)^{-1}, \quad \forall\ 2 \le i \le n.$$

In fact all the vectors in the potential relation lattice Λ' can be obtained from the linear combinations of these $n-1$ vectors. So by enforcing distinct prime ideals in the encoding of each user's secret key, the dimension of the potential relation lattice Λ' (i.e. the number of distinct prime degrees involved) becomes $(n-1)+m+n\cdot O(\log(N))$, whereas the number of linearly independent relations is $n-1$. Therefore, it is not clear how to recover a full-rank basis of Λ' from the encodings.

Remark 6.2. *The scheme we presented above makes a slight change over the original scheme of Irrer et al. [ILOP04]. In [ILOP04] the user's secret key is x and the user's public key is $s \circ x$. The encryption key with respect to the set Γ is $K = \left(\prod_{i\in\Gamma} x_i\right) \circ s^{|\Gamma|-1}$. Whereas in our scheme we flip the public key and secret key, and change the encryption key accordingly. The purpose of the change is to provide a candidate instantiation where there is no need to produce another relation lattice (and its short basis) other than $\Lambda_{\mathcal{O}}$ for the generation set S.*

Note that this change does not affect the security analysis from [ILOP04], which shows that a key recovery attack to the broadcast encryption scheme implies the ability of computing inverses.

Remark 6.3. *The CPA-style security definition in [BGW05] requires that it is hard to distinguish a correct decryption key from a random key. Our scheme is also a candidate that satisfies the CPA definition when the decisional inversion problem is hard, which is plausible under the current setting of parameters.*

7 Directed transitive signature

The concept of transitive signature was introduced by Rivest and Micali [MR02]. In a transitive signature scheme the master signer is able to sign on the edges of a graph G using the master signing key. Given the signatures on a specific set S of edges, say $S = \{(u, v), (v, w)\}$, everyone can compute the signature on the edge (u, w), and in general, any edge in the transitive closure of S, but not for any edge beyond the transitive closure of S. Transitive signatures are useful in the scenario where the graph represents some authorization relationship. The master signer has limited availability, so it has the intention of signing a limited number of edges ahead of time. New users can then dynamically join the graph, build edges, and obtain the composed signatures if the edges live in the transitive closure of the existing ones.

Transitive signatures for undirected graphs are constructed in [MR02, BN02] and many others. But for directed graphs, only the special case of directed trees is achieved by Yi [Yi07] from the RSA assumption. However, Neven later gives a construction of DTS for directed trees from any standard digital signature, which shows that directed trees are indeed simpler to achieve [Nev08].

In this section we present the concrete instantiation of the directed transitive signature from TGII. We first recall the definition and the construction, from an ideal TGII, of [Hoh03, Mol03], then give the concrete instantiation.

7.1 Definition

We adopt the definition of a directed transitive signature from [Hoh03, Mol03].

Definition 7.1. *A directed transitive signature scheme* $\mathsf{DTS} = (\mathsf{Gen}, \mathsf{Cert}, \mathsf{Sig}, \mathsf{Ver}, \mathsf{Compose})$ *consists of the following tuple of efficient algorithms:*

- Gen*: The key generation algorithm* Gen *takes as input the security parameter* 1^λ*, returns the master public-key secret-key pair* $(\mathsf{MPK}, \mathsf{MSK})$*.*
- Cert*: The node certification algorithm* Cert *takes as input the master secret key* MSK *and a node* $i \in \mathbb{N}$*, returns* $(\mathsf{PK}(i), \mathsf{SK}(i))$*, a public value and a secret value for node* i*.*
- Sig*: The edge signing algorithm* Sig *takes as input the master secret key* MSK*, the source node* i *and destination node* j*, and the associated* $(\mathsf{PK}(i), \mathsf{SK}(i))$*,* $(\mathsf{PK}(j), \mathsf{SK}(j))$*, outputs a signature* $\sigma_{i,j}$ *of the edge* $i \mapsto j$*.*
- Ver*: The verification algorithm* Ver *takes as input two multisets of node public values* P_{src}*,* P_{dest} *(*$|P_{src}| = |P_{dest}|$*), and a potential signature* σ'*, returns either* 1 *or* 0 *according to the following rule: It returns* 1 *iff* σ' *is a valid signature of a set of directed edges* $E_v = \{e_1, e_2, ..., e_m\}$ *such that* P_{src} *is the multiset of public values for source nodes in* E_v*, and* P_{dest} *is the analogous multiset for destination nodes. The validity of an edge is taken relative to* MPK*,* P_{src} *and* P_{dest}*.*
- $\mathsf{Compose}$*: The composition algorithm* $\mathsf{Compose}$ *takes as input* MPK*, two pairs of multisets of node public values* $(P1_{src}, P1_{dest})$*,* $(P2_{src}, P2_{dest})$*, and values* σ_1*,* σ_2*. Let* $\Gamma_{src} = P1_{src} \cup P2_{src}$ *and* $\Gamma_{dest} = P1_{dest} \cup P2_{dest}$*. Then* $\mathsf{Compose}$ *returns either the composed signature on edge sets corresponding to* $(\Gamma_{src} \setminus (\Gamma_{src} \cap \Gamma_{dest}), \Gamma_{dest} \setminus (\Gamma_{src} \cap \Gamma_{dest}))$*, or declares failure.*

We consider the simplest definitions of correctness and security (the more formal definitions can be found in [Hoh03, Mol03]). Correctness requires that the verification algorithm output 1 *on all the signatures obtained from the compositions in the transitive closure of the edges signed by the master signing key. Security requires that for any p.p.t. adversary, it is infeasible to forge a signature beyond the transitive closure of the signed edges. In both of the definitions of correctness and unforgeability, the adversary is allowed to dynamically add nodes and edges in the graph, and request the challenger to sign, as long as the target edge for forgery does not trivially fall in the transitive closure of the signed edges.*

Remark 7.2. *The* Ver *and* $\mathsf{Compose}$ *algorithms take multisets as sources and destinations, since the definition allows the existence of signatures of non-consecutive paths. A signature with* $|P_{src}| > 1$ *is called a waiting signature, since it will need additional compositions before it can authenticate an consecutive path. If we only allow signatures that authenticate consecutive paths, then it is always the case that* $|P_{src}| = |P_{dest}| = 1$*, which simplifies the interfaces of* Ver *and* $\mathsf{Compose}$*. For example,* Ver *takes input* $(\mathsf{MPK}, \{\mathsf{PK}(i)\}, \{\mathsf{PK}(j)\}, \sigma)$ *and returns* 1 *iff* σ *is a valid signature on edge* $i \mapsto j$*.*

7.2 A directed transitive signature scheme from TGII

We first present the construction in the abstract syntax of a TGII, then provide the parameters for the concrete instantiation.

Construction 7.3 (DTS under an abstract TGII)**.** *Given a trapdoor group with infeasible inversion* TGII*, a regular digital signature scheme* DS*, construct a directed transitive signature* DTS *as follows:*

- Gen*: The key generation algorithm* Gen *takes as input the security parameter* 1^λ *and runs the TGII parameter generation algorithm to produce the public parameters* TGII.PP *and the trapdoor* τ *of a group* $\mathbb{G} = (\circ, 1_{\mathbb{G}})$*. It also generates the signing and verification keys for the regular signature scheme* $\mathsf{DS.Gen}(1^\lambda) \to \mathsf{DS.SK}, \mathsf{DS.VK}$*. It returns the master public-key* $\mathsf{DTS.MPK} = (\mathsf{TGII.PP}, \mathsf{DS.VK})$ *and the master secret-key* $\mathsf{DTS.MSK} = (\tau, \mathsf{DS.SK})$*.*

- Cert*: The node certification algorithm* Cert *takes as input the master secret key* $\mathsf{DTS.MSK} = (\tau, \mathsf{DS.SK})$ *and a node* $i \in \mathbb{N}$*, samples a random element* $x_i \in \mathbb{G}$ *and its encoding* $\mathsf{encoding}(x_i)$*, and sets the encoding to be the public information for* i*:*

$$\mathsf{PK}(i) := \mathsf{encoding}(x_i) = \mathsf{TGII.TrapSam}(\mathsf{TGII.PP}, \tau, x_i).$$

  The secret information on node i *can be set as* $\mathsf{SK}(i) = x_i^{-1}$*, or simply left as* $\perp$ *since the master trapdoor holder can invert* $\mathsf{PK}(i) = \mathsf{encoding}(x_i)$ *to get* x_i^{-1}*. Then* Cert *produces the signature* $\Sigma(i) = \mathsf{DS.Sig}(\mathsf{DS.SK}, i||\mathsf{PK}(i))$*, takes* $\Sigma(i)$ *as the certificate of node* i*.*

- Sig*: The edge signing algorithm* Sig *takes as input the master secret key* $\mathsf{DTS.MSK} = (\tau, \mathsf{DS.SK})$*, a source node* i *and a destination node* j*, and the associated* $(\mathsf{PK}(i), \mathsf{SK}(i))$*,* $(\mathsf{PK}(j), \mathsf{SK}(j))$*. It first verifies the node certificates, then recovers* x_i *from* $\mathsf{PK}(i)$ *and* x_j^{-1} *from* $\mathsf{SK}(j)$*. It then outputs*

$$\sigma_{i,j} := \mathsf{encoding}(x_i \circ x_j^{-1}) = \mathsf{TGII.TrapSam}(\mathsf{TGII.PP}, \tau, x_i \circ x_j^{-1}).$$

- Ver*: The verification algorithm* Ver *takes as input* P_{src}*,* P_{dest}*, and a potential signature* σ'*. Set* $k := |P_{src}| = |P_{dest}|$*. For each node* a_i *in* P_{src} *and each node* b_j *in* P_{dest}*, it parses the public information as* $(a_i, \mathsf{encoding}(a_i), \Sigma(a_i))$*,* $(b_j, \mathsf{encoding}(b_j), \Sigma(b_j))$*. If any of the signatures does not pass the verification it returns 0. Then it checks whether*

$$\sigma' \circ \mathsf{encoding}(b_1) \circ ... \circ \mathsf{encoding}(b_k) = \mathsf{encoding}(a_1) \circ ... \circ \mathsf{encoding}(a_k).$$

  If so returns 1, otherwise returns 0.

- Compose*: The composition algorithm* Compose *takes as input* MPK*, two pairs of multisets of node public values* $(P1_{src}, P1_{dest})$*,* $(P2_{src}, P2_{dest})$*, and values* σ_1*,* σ_2*. It first checks if all the certificates on the nodes in* $P1_{src} \cap P2_{src} \cap P1_{dest} \cap P2_{dest}$ *are valid. If so then it sets the composed signature to* $\sigma_1 \circ \sigma_2$*.*

Instantiation from the concrete TGII and the choices of parameters. Similar to the instantiation of the broadcast encryption from the concrete TGII, we provide a stateful DTS instantiation. The crux is to understand which encodings are composable in the system, and how many non-trivial encodings of $1_{\mathcal{CL}(\mathcal{O})}$ with linearly independent exponent vectors can be derived from the compositions. The answers to these two questions determine how to choose the generator ideals for the encodings so as to provide both functionality (namely the composability of the encodings) and security (esp. for hiding any basis of Λ' of dimension m' such that $\mathbb{Z}^{m'}/\Lambda' = \mathcal{CL}(\mathcal{O})$).

We consider the worst-case scenario where n nodes with indexes $[1, 2, ..., n]$ in the graph is fully bidirectionally connected, i.e. each pair of nodes are connected by two edges from both directions. The system publishes the encodings $\{\mathsf{PK}(i) = \mathsf{encoding}(x_i)\}_{i\in[n]}$, $\left\{\sigma_{i,j} = \mathsf{encoding}(x_i \circ x_j^{-1})\right\}_{i,j\in[n],i\neq j}$. The non-trivial encodings of zero can be obtained from the linear combinations of the exponents of

$$\left\{\mathsf{PK}(j) \circ \sigma_{i,j} \circ \mathsf{PK}(i)^{-1}\right\}_{i,j\in[n],i\neq j}.$$

So if we choose relative prime degree encodings for each edge signature $\sigma_{i,j}$ and for each node public information $\mathsf{PK}(i)$, then there is always more dimensions than the number of linearly independent relations in the potential lattice Λ', which is likely to hide Λ'.

The details are given as follows:

- Select $D_0 < 0$ of a large polynomial size (say $|D_0| \approx O(\lambda^3)$) such that $h(D_0)$ is prime.
- For the conductor f: choose a set R of $2\log(\lambda)$ primes such that for all $r \in R$, $\frac{r-\left(\frac{D_0}{r}\right)}{2}$ is a polynomially large prime of size $\approx O(\lambda^3)$. Randomly pick $\log(\lambda)$ numbers from R and let them be $f_1, ..., f_{\log(\lambda)}$. Set the conductor $f := \prod_{i\in[\log(\lambda)]} f_i$. Let $\mathcal{O}$ be an order of discriminant $D = D_0 f^2$.
- For a node of index i, generate its public and secret information as follows: Choose a set of $m_i = O(\log^2(N))$ prime ideals $S_i = \{[(\ell_{i,k}, b_{i,k}, \cdot)]\}_{k\in[m_i]}$, where the primes $\{7 < \ell_{i,k} \leq \mathsf{poly}(\lambda)\}$ have never been used in the system. Then generate $\mathsf{PK}(i)$ as a composable encoding of a random element x_i by running $\mathsf{encoding}(x_i) \leftarrow \mathsf{RandomSam}(\tau, S_i)$.
- Generating the signature $\sigma_{i,j}$, i.e. an composable encoding of $x_i \circ x_j^{-1}$: Choose $m_{i,j} = O(\log(\lambda))$ primes $S_{i,j} = \{7 < \ell_{i,j,k} \leq \mathsf{poly}(\lambda)\}_{k\in[m_{i,j}]}$ that have never been used in the system, and such that the set of ideal classes $\{C_{i,j,k} = [(\ell_{i,j,k}, b_{i,j,k}, \cdot)]\}_{k\in[m_{i,j}]}$ generates $\mathcal{CL}(\mathcal{O})_{\text{odd}}$. Generate a short basis of $\Lambda_{i,j} = \left\{\mathbf{e} \mid \prod_{k\in[m_{i,j}]} C_{i,j,k}^{e} = 1\right\}$.

  Next, solve the discrete-log problem of $x_i \circ x_j^{-1}$ under the generator set $S_{i,j}$, obtain a short solution $\mathbf{e}_{i,j} \in \mathbb{Z}^{m_{i,j}}$. Then derive the composable encoding of $x_i \circ x_j^{-1}$ as usual.

Remark 7.4. *The definition of [Hoh03, Mol03] additionally requires that for any edge in the graph , the composed signature should be indistinguishable from a signature produced by the master signer. We remark that the DTS from our TGII does not achieve this property, since the signatures grow as they are composed. The same phenomenon happens in the DTS from the TGII built from iO in [YYHK14].*

8 Future directions

We conclude the paper with several future directions.

Further investigation of the (ℓ, ℓ^2)-isogenous neighbors problem. The hardness of the (ℓ, ℓ^2)-isogenous neighbor problem over $\mathbb{Z}/N\mathbb{Z}$ is necessary for the security of our candidate trapdoor group with infeasible inversion. Let us remark that once the adversary is given two j-invariants that are ℓ-isogenous, she can recover the rational polynomial of the isogeny, which is not available in the ℓ-isogenous neighbor problem where the adversary is given a single j-invariant as input. Although it is not clear how to use the explicit rational polynomial of the isogeny to mount an attack, it should serve as a warning sign that the (ℓ, ℓ^2)-isogenous neighbors problem might be easier than the ℓ-isogenous neighbors problem.

Looking for alternative constructions of GII or TGII. Given the complications and the limitations of our construction of TGII, one might want to look for a simpler or a different construction of GII or TGII. Some concrete directions to study further are:

1. A construction, where the encoding is stateless.
2. A construction, where the size of the encoding does not grow with composition.
3. A (T)GII candidate for a non-commutative group, which is likely to give a candidate indistinguishability obfuscator following [CV13].

Acknowledgments

We thank Andrew V. Sutherland for his helpful comments.

References

[BD90] Johannes Buchmann and Stephan Düllmann. On the computation of discrete logarithms in class groups. In *Conference on the Theory and Application of Cryptography*, pages 134–139. Springer, 1990.

[BGI+01] 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.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.

[Bis11] Gaetan Bisson. Computing endomorphism rings of elliptic curves under the grh. *Journal of Mathematical Cryptology*, 5(2):101–113, 2011.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.

[BLS12] Reinier Bröker, Kristin Lauter, and Andrew V Sutherland. Modular polynomials via isogeny volcanoes. *Mathematics of Computation*, 81(278):1201–1231, 2012.

[BN02] Mihir Bellare and Gregory Neven. Transitive signatures based on factoring and rsa. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 397–414. Springer, 2002.

[BS11] Gaetan Bisson and Andrew V Sutherland. Computing the endomorphism ring of an ordinary elliptic curve over a finite field. *Journal of Number Theory*, 131(5):815–831, 2011.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 59–71. Springer, 1998.

[BW88] Johannes A. Buchmann and Hugh C. Williams. A key-exchange system based on imaginary quadratic fields. *J. Cryptology*, 1(2):107–118, 1988.

[BWZ14] Dan Boneh, Brent Waters, and Mark Zhandry. Low overhead broadcast encryption from multilinear maps. In *CRYPTO (1)*, volume 8616 of *Lecture Notes in Computer Science*, pages 206–223. Springer, 2014.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 3–12. Springer, 2015.

[CL84] Henri Cohen and Hendrik Willem Lenstra. Heuristics on class groups of number fields. 1984.

[CL05] Denis Charles and Kristin Lauter. Computing modular polynomials. *LMS Journal of Computation and Mathematics*, 8:195–204, 2005.

[CLG09] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.

[CM94] Jean-Marc Couveignes and François Morain. Schoof's algorithm and isogeny cycles. In *International Algorithmic Number Theory Symposium*, pages 43–58. Springer, 1994.

[Coh95] Henri Cohen. A course in computational algebraic number theory. 1995.

[Cop97] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006.

[Cox11] David A Cox. *Primes of the form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication*, volume 34. John Wiley & Sons, 2011.

[CV13] Ran Canetti and Vinod Vaikuntanathan. Obfuscating branching programs using black-box pseudo-free groups. Cryptology ePrint Archive, Report 2013/500, 2013.

[Dam88] Ivan Damgård. On the randomness of legendre and jacobi sequences. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 163–172. Springer, 1988.

[Dem93] 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.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[E+98] 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.

[ES10] Andreas Enge and Andrew V Sutherland. Class invariants by the crt method. In *International Algorithmic Number Theory Symposium*, pages 142–156. Springer, 2010.

[Feo17] Luca De Feo. Mathematics of isogeny based cryptography. *arXiv preprint arXiv:1711.04062*, 2017.

[FM02] Mireille Fouquet and François Morain. Isogeny volcanoes and the SEA algorithm. In *International Algorithmic Number Theory Symposium*, pages 276–291. Springer, 2002.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.

[Gal99] Steven D Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS Journal of Computation and Mathematics*, 2:118–138, 1999.

[GGH+13] 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.

[GH93] Dorian Goldfeld and Jeffrey Hoffstein. On the number of Fourier coefficients that determine a modular form. In Marvin Knopp and Mark Sheingorn, editors, *A Tribute to Emil Grosswald: Number Theory and Related Analysis*, volume 143 of *Contemporary Mathematics*, pages 385–393, Providence, Rhode Island, 1993. AMS.

[GHS02] Steven D. Galbraith, Florian Hess, and Nigel P. Smart. Extending the GHS weil descent attack. In *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 29–44. Springer, 2002.

[GM05] Steven D Galbraith and James F McKee. Pairings on elliptic curves over finite commutative rings. In *IMA International Conference on Cryptography and Coding*, pages 392–409. Springer, 2005.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[HJPT98] Detlef Hühnlein, Michael J Jacobson, Sachar Paulus, and Tsuyoshi Takagi. A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–307. Springer, 1998.

[HM89] James L Hafner and Kevin S McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American mathematical society*, 2(4):837–850, 1989.

[HM00] Safuat Hamdy and Bodo Möller. Security of cryptosystems based on class groups of imaginary quadratic orders. In *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 234–247. Springer, 2000.

[Hof14] Jeffrey Hoffstein. A History of the Development of NTRU. Eurocrypt talk, 2014.

[Hoh03] Susan Rae Hohenberger. *The cryptographic impact of groups with infeasible inversion.* PhD thesis, Massachusetts Institute of Technology, 2003.

[IJ13] Sorina Ionica and Antoine Joux. Pairing the volcano. *Math. Comput.*, 82(281):581–603, 2013.

[ILOP04] 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.

[JF11] 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.

[JMV05] David Jao, Stephen D. Miller, and Ramarathnam Venkatesan. Do all elliptic curves of the same order have the same difficulty of discrete log? In *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2005.

[KK98] 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.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[Koh96] David Russell Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.

[Len87] Hendrik Willem Lenstra. Factoring integers with elliptic curves. *Annals of mathematics*, pages 649–673, 1987.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LZ94] 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.

[McC88] Kevin S McCurley. *Cryptographic key distribution and computation in class groups*. IBM Thomas J. Watson Research Division, 1988.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[Mol03] David Molnar. *Homomorphic Signature Schemes*. PhD thesis, Harvard College, 2003.

[MR02] Silvio Micali and Ronald L. Rivest. Transitive signature schemes. In *CT-RSA*, volume 2271 of *Lecture Notes in Computer Science*, pages 236–243. Springer, 2002.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measure. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[Nev08] Gregory Neven. A simple transitive signature scheme for directed trees. *Theoretical Computer Science*, 396(1-3):277–282, 2008.

[NR06] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 271–288. Springer, 2006.

[PH78] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 24(1):106–110, 1978.

[Rab79] Michael O Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE, 1979.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145, 2006.

[Sch82] René Schoof. Quadratic fields and factorization. *Mathematisch Centrum Computational Methods in Number Theory, Pt. 2 p 235-286(SEE N 84-17999 08-67)*, 1982.

[Sch85] René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of computation*, 44(170):483–494, 1985.

[Sch95] René Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.

[Sil13] Joseph H Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151. Springer Science & Business Media, 2013.

[Sut] 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.

[Sut11a] Andrew V Sutherland. Computing Hilbert class polynomials with the Chinese remainder theorem. *Mathematics of Computation*, 80(273):501–538, 2011.

[Sut11b] Andrew V Sutherland. Structure computation and discrete logarithms in finite abelian p-groups. *Mathematics of Computation*, 80(273):477–500, 2011.

[Sut12] Andrew V Sutherland. Accelerating the CM method. *LMS Journal of Computation and Mathematics*, 15:172–204, 2012.

[Sut13a] Andrew V Sutherland. Isogeny volcanoes. *The Open Book Series*, 1(1):507–530, 2013.

[Sut13b] Andrew V Sutherland. On the evaluation of modular polynomials. *The Open Book Series*, 1(1):531–555, 2013.

[Tat66] John Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones mathematicae*, 2(2):134–144, 1966.

[Tes99] Edlyn Teske. The Pohlig–Hellman method generalized for group structure computation. *Journal of Symbolic Computation*, 27(6):521–534, 1999.

[Vél71] Jean Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273:238–241, 1971.

[Yi07] Xun Yi. Directed transitive signature scheme. In *CT-RSA*, volume 4377 of *Lecture Notes in Computer Science*, pages 129–144. Springer, 2007.

[YYHK14] 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.

[YYHK18] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Generic hardness of inversion on ring and its relation to self-bilinear map. *IACR Cryptology ePrint Archive*, 2018:463, 2018.

[Zha14] Mark Zhandry. Adaptively secure broadcast encryption with small system parameters. *IACR Cryptology ePrint Archive*, 2014:757, 2014.