Isogeny-based time-release cryptography

Parsa Tasbihgou

School of Computer and Communication Sciences

Semester Project

June 2024

Responsible
Prof. Serge Vaudenay
EPFL / LASEC

Supervisor
Dr. Tako Boris Fouotsa
EPFL / LASEC

E-mail: parsa.tasbihgou@epfl.ch (Parsa Tasbihgou)

Isogeny-based time-release cryptography

Parsa Tasbihgou

EPFL, Lausanne, Switzerland

Abstract. In this report we give a review of some concepts in time-release cryptography. We focus on constructions based on isogenies of supersingular elliptic curves we also look at some constructions based on squaring in groups of unknown order. We discuss in detail a paper by Luca De Feo and Jeffrey Burdges titled "Delay Encryption" [BDF21].
An important task in many isogeny-based systems is generating a supersingular elliptic curve with unknown endomorphism ring. We look at methods to generate such a curve with the extra constraint of being defined over $\mathbb{F}_p$, which turns out to be challenging.
Finally, we look at necessary assumptions for existence of time-release primitives and the relations between them.

Keywords: Isogenies · Supersingular elliptic curves · Delay Encryption · Groups of unknown order

Contents

1 Introduction

1.1 Motivation

Some cryptosystems can benefit from having a lifespan and releasing their information after a set amount of time. For example:

- We would like to be able to open sealed bids from an auction after everyone has submitted their bid, independent of their cooperation.
- We would like to be able to open encrypted votes after the voting period is finished.
- We would like to encrypt and publish sensitive information in way that they automatically become public after a set amount of time.

There are also less obvious applications of such cryptosystems; trusted randomness beacons and consensus from proof of resources in blockchains are some such applications. For more details about applications we refer to section 2 of [BBBF18] and section 2 of [Med23].
The idea of time-release cryptography was first introduced with a third party that would release the information after a set amount of time. But as usual it is inconvenient to rely on a long-term trusted party so we are interested in constructions that don't rely on a long-term trusted third party (TTP).
The first public time-release cryptosystem was introduced by Rivest, Shamir and Wagner in 1996 [RSW96]. They introduced Time-lock puzzles were a probabilistic algorithm would create a "puzzle" that requires some amount of sequential work (that we refer to as "time"). After the puzzle is solved a secret is revealed that allows an otherwise infeasible computation. Since time-lock puzzles (TLP), a lot of work has been done in this field. In this report we look at verifiable delay functions (VDF), which have been a subject of study for some time and delay encryption that was recently introduced by De Feo and Burdges.

1.2 Related works

Similar to other cryptographic primitives, time-release primitives are constructed from some assumptions. Some of the frameworks that have been studied to construct time-release primitives are: squaring in groups of unknown order, isogenies of supersingular elliptic curves, univariate permutation polynomials and random oracles. Pietrzak [Pie19] gave an unconditionally secure VDF in the random oracle model which is interesting from a theoretical point of view but is inefficient and impractical in practice. There are other constructions based on more familiar cryptographic assumptions like RSA and the endomorphism ring problem.
Wesolowski [Wes20] gave a VDF construction based on squaring in groups of unknown order that is efficient and is used in the Etherium system. There are some drawbacks to his construction, for example the requirement of a trusted third party or distributed RSA modulus generation which is costly in practice and doesn't scale well.
De Feo et al. [DFMPS19] gave a VDF construction based on isogenies and pairings of supersingular elliptic curves. While their construction also relies on either a TTP or distributed generation of random supersingular elliptic curves, this task is more efficient compared to RSA modulus generation. Moreover in contrast to Wesolowski's construction, there is no evidence that random sampling of supersingular curves with unknown endomorphism ring is impossible.
There are many previous works on the theory of elliptic curves and isogenies, inspired by their applications in cryptography. This background is crucial for our discussions on isogeny-based approaches to time-release cryptography, hence we dedicate Section 2 to this background.

1.3 Contributions

Our contribution in this project is resolving two problems from the delay encryption paper [BDF21]. The watermarking method proposed in that paper has inverse linear soundness error in the proof length. In Subsection 4.4 we propose a new watermarking method that has constant length and prover time with zero soundness error. This is a considerable improvement over the original method since it eliminates any advantage for a malicious party with no extra cost to honest parties.
The authors propose a proof of knowledge of isogenies over $\mathbb{F}_p$ that has constant proof length. They note that their proof system is not proven to be sound. In Subsection 5.3 we show that a malicious prover can convince the verifier without knowledge of any isogeny but the knowledge of a single discrete logarithm that depends only the statement and not the challenge.

1.4 Outline

Our discussions will include concepts from the theory of elliptic curves over finite fields and the theory of quadratic number fields, in Section 2 we give a brief reminder of these concepts.
In Section 3 we introduce the main time-release primitives that are the subject of study in this project along with their security definitions. In Section 4 we look at instantiations of time-release primitives introduced in Section 3 based on isogenies of supersingular elliptic curves and pairings. In Section 5 we look at the problem of sampling a supersingular elliptic curve with unknown endomorphism ring. A problem that is used in both isogeny-based systems from Section 4 and many other isogeny-based systems that rely on hardness of endomorphism ring computation. The main interest in this section is to prove the knowledge of a secret isogeny without revealing other information.
The existence and relations between both public key and symmetric key primitives are well-studied however, time-release primitives are relatively young and their relations are not understood very well. In Section 7 we review some of the interesting relations and necessary assumptions required for existence of time-release cryptography.
Squaring in groups of unknown order is another important framework for instantiating time-release cryptographic primitives and the work on isogeny-based approaches is partly motivated by resolving short comings in this framework. In Appendix A we look at constructions in this framework.

2 Preliminaries on elliptic curves and isogenies

In this section we will recall some facts and definitions from the theory of elliptic curves. For more detailed discussions we refer to Panny's Ph.D thesis [Pan] and Silverman's book [Sil09].

2.1 Elliptic curves as algebraic curves

Elliptic curves are smooth projective curves of genus 1. Any elliptic curve defined over a field k with characteristic not equal to 2 or 3 can be specified by a short Weierstrass equation in the following form.

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{1}$$

This equation defines a projective variety in $\mathbb{P}^2(k)$ and on the XY affine chart has a point at infinity at $\mathcal{O} = [0 : 1 : 0]$. Ee can set $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$ and write Equation 1 in the

XY affine chart in the following form.

$$y^2 = x^3 + ax + b \tag{2}$$

By definition elliptic curves should be smooth and therefore they can't have singular points. This condition can be verified for a curve in the short Weierstrass form by checking $4a^3 + 27b^2 \neq 0$.

2.2 Points on elliptic curves

An elliptic curve E is defined by an equation $E : y^2 = x^3 + ax + b$. We say E is defined over field k if $a, b \in k$. Notice that E is also defined on any extension of k, so E can have points defined in all extensions of k up to $\overline{k}$ (also further extensions but no additional point can be found). We denote the points on E that are defined over some extension f/k by $E(f)$.
When E is defined over a field with non-zero characteristic, the number of points on E is bounded by Hasse's theorem, $\#E(\mathbb{F}_q) = q + 1 - t$ where t is called the trace of the curve and $|t| \leq 2\sqrt{q}$. Additionaly, the trace can be computed in polynomial time using Schoof's algorithm.

2.3 Elliptic curves as Abelian varieties

In addition to being algebraic varieties, elliptic curves are Abelian varieties meaning that there is an Abelian group structure on points on elliptic curves.
From Bezout's theorem every line in the projective space intersects any cubic curve in exactly three points, furthermore the line that passes through two points $P, Q \in E(k)$ passes through a third point also defined on k. We use this fact to define a group operation on the points on E. Take the point at infinity $\mathcal{O} = [0 : 1 : 0]$ to be the identity element. Since we use additive notation for the group, the identity element $\mathcal{O}$ is sometimes called *zero*.
For any non-zero point $P = [x : y : 1]$ we set the inverse element $-P = [x : -y : 1]$. Let $P, Q \in E(k)$ and let $R \in E(k)$ be the third intersection point of E and the line that passes through P and Q, we require that the sum of every three points on a line be zero, so $P + Q + R = \mathcal{O}$ and by rearranging we get $P + Q = -R$. We don't formally prove that the defined operation is a group, a formal proof can be found in III.2 [Sil09].
Figure 1 illustrates an example of addition and scalar multiplication on elliptic curves.

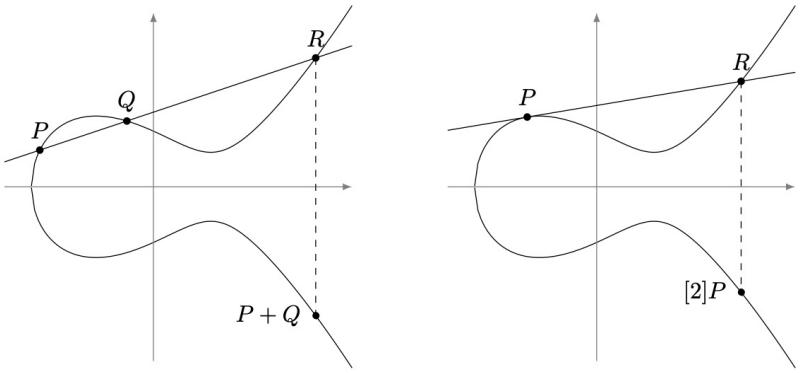


Figure 1: Addition and scalar product on elliptic curves

$E(k)$ is an Abelian group, now we want to study the subgroups of this group. For any $n \in \mathbb{N}$ all the points $P \in E(k)$ such that $nP = 0$, form a group denoted by $E[n]$, and called the m-torsion subgroup.
If $char(k) = 0$ then $E[n] \simeq (\mathbb{Z}_n)^2$. If $char(k) = p > 0$ let $n = mp^r$ where p and m are coprime, then $E[n] \simeq \mathbb{Z}_m \times \mathbb{Z}_n$ or $E[n] \simeq \mathbb{Z}_m \times \mathbb{Z}_m$. In simpler terms:

1. $E[p] \simeq \mathbb{Z}_p$
2. $E[p] \simeq \{\mathcal{O}\}$.

The curves that fall in the first case are called ***ordinary*** and the curves in the second case are called ***supersingular***.
An elliptic curve is supersingular if and only if its trace is zero modulo p, equivalently $\#E(\mathbb{F}_q) \equiv 1 \bmod p$.

2.4 Isogenies of elliptic curves

Definition 1 (Rational map)**.** For projective curves C and C',
Let $\phi = (\phi_1, \phi_2, \phi_3) \in \mathbb{P}^2(k(C))$. For any $P \in C(\overline{k})$, let $\phi_1(P), \phi_2(P)$ and $\phi_3(P)$ be defined and at least one of them be non-zero. If $\phi_1(P), \phi_2(P), \phi_3(P) \in C'(\overline{k})$, then $\phi : C \to C'$ is called a rational map from C to C'.
ϕ is said to be defined at point P if and only if there exists $\lambda \in k(C)^*$ such that $\lambda\phi_1, \lambda\phi_2$ and $\lambda\phi_3$ are all defined at P and at least one of them is non-zero.

Definition 2 (Morphism)**.** A rational map over curve C that is defined at every point is called a morphism.

Theorem 1. *When C is a smooth projective curve, any rational map $\phi : C \to C'$ is a morphism, independent of C'.*

Corollary 1. *Any rational map on an elliptic curve is a morphism.*

Definition 3 (Isomorphism)**.** Two projective curves C and C' are isomorphic if and only if there exists an invertable morphism $\phi : C \to C'$ such $\phi \circ \phi^{-1}$ is the identity map on C' and $\phi^{-1} \circ \phi$ is the identity map on C. Such a morphism is called an isomorphism.

Notice that a rational function can be defined on a higher extension than where the curves are defined. Consequently, morphisms can also be defined on higher extensions. For example two curves defined over k might not be isomorphic over k but be isomorphic over $\overline{k}$.

Definition 4 (Isogeny)**.** A morphism $\phi : E \to E'$ on elliptic curves is called an **isogeny** if it is a surjective homomorphism of Abelian groups.
The set of all isogenies from a curve E to a curve E' is denoted by $Hom(E, E')$.

Theorem 2. *Any morphism on a projective curve is either constant or surjective.*

Theorem 3. *Any morphism on two Abelian varieties that takes the identity element to the identity element is a homomorphism.*

Corollary 2. *A non-constant rational map $\phi : E \to E'$ on elliptic curves that takes $\mathcal{O}_E$ to $\mathcal{O}_{E'}$ is an isogeny.*

Definition 5 (Endomorphism)**.** An isogeny from a curve to itself is called and endomorphism.

Definition 6. There are two special endomorphisms:

p-power Frobenius: Let E/k be an elliptic curve defined over k and $char(k) = p$ the p-power Frobenius π_p is defined $\pi_p(x, y) = (x^p, y^p)$.

Frobenius: Let $E/\mathbb{F}_q$ be an elliptic curve, the Frobenius endomorphism $\pi : E \to E$ is defined $\pi(x, y) = (x^q, y^q)$.
When $\#E(\mathbb{F}_q) = q + 1 - t$, t is the called the trace of curve E and the characteristic equation of Frobenius is $\pi^2 - t\pi + q = 0$.

Definition 7 (Automorphism)**.** An endomorphism that is an isomorphism is called an automorphism.

Definition 8 (Standard form of isogenies)**.** Any isogeny $\phi : E \to E'$ can be written as $(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)}y)$ where $gcd(u, v) = gcd(s, t) = 1$.
In this representation polynomials u, v, s, t are uniquely determined up to scalar factors. Also v and t have the same set of roots.
The affine points $(x : y : 1) \in \text{Ker}\phi$ are exactly the points such that $v(x) = 0$. This implies the kernel of any isogeny is finite subgroup.

Definition 9 (Degree of isogenies)**.** Let $\phi : E \to E'$ be an isogeny with standard form $(\frac{u(x)}{v(x)}$, the degree of ϕ is defined as $max\{deg(u), deg(v)\}$.

Definition 10 (Separable isogenies)**.** Let $\phi : E \to E'$ be an isogeny with standard form $(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)}y)$. ϕ is called separable if the derivative of $\frac{u(x)}{v(x)}$ is non-zero, otherwise its called inseparable isogeny.

The degree and separability of isogenies can alternatively be defined in the following way. Any isogeny $\phi : E/k \to E'/k$ induces an injective map between their function fields $\phi^* : k(E') \to k(E); \phi^*(f) = f \circ \phi$.
$k(E)$ is a finite extension of $\phi^*(E')$. The degree of ϕ is the degree of this extension and ϕ is separable if this extension is separable.

Theorem 4. *If ϕ is a separable isogeny then, $\#Ker\phi) = deg(\phi)$.*

Definition 11 (Dual isogeny)**.** Any isogeny $\phi : E \to E'$ has a dual isogeny $\hat{\phi}$ of the same degree and defined over the same field such that $\phi \circ \hat{\phi} = [deg\phi]id_{E'}$ and $\hat{\phi} \circ \phi = [deg\phi]id_E$.

Theorem 5 (Tate)**.** *Two elliptic curves E/k and E'/k are isogenous over k iff $\#E(k) = \#E'(k)$.*

Definition 12 (Kernel isogeny)**.** Let E/k be an elliptic curve and $G \leq E(k)$ a finite subgroup. There is separable isogeny $\phi_G : E \to E/G$ with kernel G. If the order of G is smooth, this isogeny can be computed using Velu's formula.

2.5 Isomorphism classes

Definition 13 (j-invariant)**.** An elliptic curve defined by the Weierstrass equation $E : y^2 = x^3 + ax + b$ has j-invariant $j(E) = \frac{4a^3}{4a^3+27b^2}$.

Theorem 6. *Two supersingular elliptic curves are isomorphic over $\overline{\mathbb{F}_p}$ if and only if they have the same j-invariant.*

Corollary 3. *If two curves have different j-invariants they are not isomorphic on any extension of the base field. Meanwhile two curves with the same j-invariant might not be isomorphic on their field of definition but be isomorphic on some extension field.*

Theorem 7. *For any $j \in \mathbb{F}_q$ there is some elliptic curve defined over $\mathbb{F}_q$ with j-invariant equal to j.*

Theorem 8. *The j-invariant of any supersingular elliptic curve is defined over* F_{p^2}.

Corollary 4. *Any supersingular elliptic curve is isomorphic to some curve defined over* $\mathbb{F}_{p^2}$.

Definition 14 (Equivalence of isogenies)**.** Two isogenies $\phi, \psi : E \to E'$ are equivalent if an automorphism α on E' exists such that $\phi = \alpha \circ \psi$.

2.6 Endomorphism rings

The set of all endomorphisms of an elliptic curve E plus the zero map forms a ring under addition and composition. The ring is called the endomorphism ring and denoted by $\mathrm{End}(E)$. There are the following possibilities for endomorphism rings of elliptic curves:

$char(k) = 0$. $\mathrm{End}(E) = \mathbb{Z}$ or its isomorphic to an order in an imaginary quadratic field.

$char(k) = p$. There are two cases:

Ordinary: $\mathrm{End}(E)$ is isomorphic to an order in some imaginary quadratic field.

Supersingular: $\mathrm{End}(E)$ is isomorphic to a maximal order in the quaternion algebra ramified at p and ∞, $B_{p,\infty}$.

It is clear that multiplication by any non-zero scalar is an endomorphism, therefore the endomorphism ring always embeds the integer ring $\mathbb{Z}$.
The Frobenius endomorphism π is usually non-scalar $\pi \notin \mathbb{Z}$ then $\mathbb{Z}[\sqrt{t^2 - 4q}] \simeq \mathbb{Z}[\pi] \subseteq \mathrm{End}(E)$. Sometimes this is the whole story and $\mathrm{End}(E) = \mathbb{Z}[\pi]$.
But the endomorphism ring can be bigger in several ways. A simple way is that $\pi - a$ is divisible by some integer which implies $\mathbb{Z}[\pi] \not\subseteq \mathbb{Z}[\frac{\pi - a}{d}] \subseteq \mathrm{End}(E)$. We can resolve this issue by allowing denominators.
However the set $\{1, \pi\}$ might fail to span the whole endomorphism ring even when denominators are allowed. To separate the cases where the issue is resolved by allowing denominators from the cases where the rank of the ring is larger that 2, we create a coarser object.

Definition 15 (Endomorphism algebra)**.** Define the endomorphism algebra of E/k as $\mathrm{End}^o_k(E) = \mathrm{End}_k(E) \otimes_{\mathbb{Z}} \mathbb{Q}$. Intuitively $\mathrm{End}^o_k(E)$ consists of elements $\frac{\alpha}{d}$ such that $\alpha \in \mathrm{End}_k(E)$ and $d \in \mathbb{Z}$.
The endomorphism algebra is a $\mathbb{Q}$-Algebra with the same rank as $\mathrm{End}(E)$.

Definition 16 (Conjugate endomorphism)**.** Let $\theta \in \mathrm{End}(E)$ be an endomorphism, the conjugate endomorphism of θ denoted by $\overline{\theta}$ corresponds to the conjugate element of θ in the endomorphism algebra of E, equivalently it is the dual of θ as an isogeny.

Definition 17 (Norm of endomorphism)**.** Let $\theta \in \mathrm{End}(E)$ be an endomorphism, norm of θ denoted by $N(\theta)$ is defined as $\theta\overline{\theta}$. Remember that as an isogeny $\theta\overline{\theta}$ is multiplication by degree of θ, so it is an integer scalar.

Definition 18 (Trace of endomorphism)**.** Let $\theta \in \mathrm{End}(E)$ be an endomorphism, the trace of θ denoted by $Tr(\theta)$ is defined as $\theta + \overline{\theta}$. The trace is always an integer.

Theorem 9. *Any endomorphism is an algebraic integer of degree at most 2. In particular any* $\theta \in End(E)$ *satisfies* $\theta^2 - Tr(\theta)\theta + N(\theta) = 0$.

Notice that the endomorphism algebra no longer has the problem of denominators. In addition the isomorphism class of endomorphism algebra is isogeny invariant, meaning that two isogenous curves have the same endomorphism algebra.

Proof. Let $\phi : E \to E'$ be an isogeny. The map $\tau_\phi : End^o_k(E) \to End^o_k(E'); \tau_\phi(\alpha) = \frac{\phi\alpha\hat{\phi}}{deg\phi}$ is an isomorphism of $\mathbb{Q}$-Algebras. □

2.7 Classification of endomorphism algebras

In this section we study the structure of the Endomorphism algebras. Remember that isomorphism classes of endomorphism algebras are isogeny invariant, so each isogeny class has a unique endomorphism algebra up to isomorphism. Let E/k be a supersingular elliptic curve, its endomorphism algebra can be classified as follows:

char$(k) = 0$. Either $\mathrm{End}_k(E) = \mathbb{Z}$ meaning $\mathrm{End}_k^o(E) = \mathbb{Q}$, or $\mathrm{End}_k^o(E)$ is an imaginary quadratic field.

$k = \mathbb{F}_q$. $\mathrm{End}_k^o(E)$ is either an imaginary quadratic field, or a quaternion algebra.

Definition 19 (Lattice)**.** A lattice L in a finite dimensional $\mathbb{Q}$-Algebra A is a finitely generated subgroup that spans A over $\mathbb{Q}$.

Definition 20 (Order)**.** An order $\mathcal{O}$ in a finite dimensional $\mathbb{Q}$-Algebra is a lattice that is also a subring.
Orders are partially ordered under inclusion. An order is maximal iff its not properly included in any non-trivial order.

2.7.1 Quadratic endomorphism algebras

Let $E/\mathbb{F}_q$ be an elliptic curve, the Frobenius endomorphism satisfies $\pi^2 - t\pi + q$. As previously mentioned π is usually non-scalar so $\pi = \sqrt{t^2 - 4q} \notin \mathbb{Z}$. In this case the $\mathrm{End}_k(E)^o = \mathbb{Q}(\pi) = \mathbb{Q}(\sqrt{t^2 - 4q})$ is a 2-dimensional $\mathbb{Q}$-algebra and $\mathrm{End}_k(E)$ is an order containing $\mathbb{Z}[\pi]$ in $\mathbb{Q}(\pi)$.
This case applies to all ordinary supersingular elliptic curves over finite fields and supersingular curves over prime fields with characteristic at least 5 since $\pi \notin \mathbb{Z}$ is always true. For supersingular curves over prime fields with characteristic $p \geq 5$, the trace is 0 modulo p so $\pi = 2\sqrt{-p}$.
Notice that in the case of quadratic endomorphism algebras all endomorphisms commute freely.

2.7.2 Quaternionic endomorphism algebras

If π is scalar $\mathbb{Z}[\pi] = \mathbb{Z}$ and there are more endomorphism than expected, also not all these endomorphisms commute.

Theorem 10. *Let $E/\mathbb{F}_q$ be a supersingular elliptic curve. $End^o(E) = B_{p,\infty}$ and $End(E)$ is a maximal order in this quternion algebra. Moreover, $End_k(E)$ is non-commutative (i.e. a maximal order in $B_{p,\infty}$) iff $\pi \in \mathbb{Z}$.*

In contrast to imaginary quadratic fields, there are many maximal orders in quaternion algebras, so it reasonable to ask "to what extent can a curve be recovered from its endomorphism ring?". During answered this question.

Theorem 11 (During correspondence)**.** *Let p be a prime and $\sigma : x \to x^p$ the non-trivial automorphism of $\mathbb{F}_{p^2}$. Taking endomorphisms induces a bijection.*

$$\{j(E) | E/\overline{\mathbb{F}_p} \text{ supersingular}\}/\langle\sigma\rangle \leftrightarrow \{\text{Maximal orders in } B_{p,\infty}\}$$

Specifically, for any maximal order $\mathcal{O}$ in $B_{p,\infty}$ there are one or two isomorphism classes of supersingular curves over $\overline{\mathbb{F}_p}$ (equivalently over $\mathbb{F}_{p^2}$). There is only one such curve E iff $j(E) \in \mathbb{F}_p$ otherwise, the two curves have conjugate j-invariant in $j(E) \in \mathbb{F}_{p^2} \backslash \mathbb{F}_p$ and are connected by the p-power Frobenius isogeny $\pi_p : (x, y) \to (x^p, y^p)$.

2.8 Isogenies and ideals

In this section we study the connection between isogenous curves and ideal classes of endomorphism rings. Through out this section E is an elliptic curve defined over k and $\mathcal{O} = \mathrm{End}_k(E)$ is its ring of rational endomorphisms.
Any non-zero left ideal $I \subseteq \mathcal{O}$ defines a finite subgroup of E denoted by $E[I]$.

$$E[I] = \bigcap_{\alpha \in I} \mathrm{Ker}(\alpha).$$

Notice that it suffices to iterate over the generators of I.
When $I = J\pi^r$ such that $J \not\subseteq \mathcal{O}\pi$ then the cardinality of $E[I]$ is equal to the norm of J which is defined as the gcd of the norms of all elements in J.
As mentioned earlier for any finite subgroup of E there is a unique separable isogeny with that kernel so we can define the isogeny $\phi_{E[I]}$ as the isogeny with kernel $E[I]$. To simplify the notation we just write ϕ_I and the target curve is denoted by E/I.
From the definition of $E[I]$ one can show that multiplying I by $\gamma \in \mathcal{O}^*$ from the right doesn't change E/I up to k-isomorphism. Therefore the codomain defined by an ideal only depends on its ideal class.
Similarly, one can concretely define the kernel isogeny for some subgroup $G \leq E(k)$. Define $I(G) = \{\alpha \in \mathrm{End}(E) | \alpha(G) = 0\}$. $I(G)$ is a left $\mathrm{End}(E)$-ideal and the corresponding isogeny $\phi_{I(G)}$ is the unique separable isogeny with kernel G.

Definition 21 (Right (left) order of a lattice)**.** Let $I \subset B_{p,\infty}$ be a lattice, the right order of L is the largest subring of $B_{p,\infty}$ for which L is a fractional ideal.

$$\mathcal{O}_R(I) = \{\alpha \in B_{p,\infty} | I\alpha \subseteq I\}.$$

The left order is defined similarly with multiplication from left.

Theorem 12. *Let E_0 be a supersingular elliptic curve with endomorphism ring $\mathcal{O}_0$. For any supersingular curve E with endomorphism $\mathcal{O}$ there is unique left ideal class of $\mathcal{O}_0$, $[I]$ such that $E_0/I \simeq E$ and $\mathcal{O}$ is isomorphic to the right order of I.*
Furthermore $\mathcal{O}_0 \cdot \mathcal{O}$ is an ideal in that ideal class. Such an ideal is called a connecting ideal because it is a left-$\mathcal{O}_0$ and right-$\mathcal{O}$ ideal.

The set of all isogenies between E_0 and E denoted by $Hom(E_0, E)$ corresponds to the set of all connecting ideals of $\mathrm{End}(E_0)$ and $\mathrm{End}(E)$ and this is exactly a left-$\mathcal{O}_0$ right-$\mathcal{O}$ ideal class. This ideal class is a lattice in $B_{p,\infty}$ in other words a free rank 4 $\mathbb{Z}$-module.

Theorem 13. *Any left $\mathcal{O}$-ideal I has a representative in its ideal class with ℓ-power norm for any prime $\ell \neq p$.*

Corollary 5. *This theorem implies that any pair of isogenous supersingular elliptic curves have an isogeny of ℓ-power degree. Consequently, the ℓ-isogeny graph is connected for all primes $\ell \neq p$.*

Let E, E' be supersingular elliptic curves with endomorphism rings $\mathcal{O}, \mathcal{O}'$. We know $I = \mathcal{O} \cdot \mathcal{O}'$ corresponds to an isogeny $\phi_I : E \to E'$ however this isogeny is generally not smooth and therefore unfit for computation. The corollary states that there is a ℓ-power ideal J equivalent to I. Wesolowski gave a polynomial algorithm to find this representative in polynomial time, also there is the KLPT algorithm that finds J in heuristic polynomial time. The KLPT algorithm is faster than Wesolowski's algorithm in practice. Additionally under some heuristic assumption the norm of J is approximately $p^{7/2}$ therefore the corresponding isogeny has degree approximately $p^{7/2}$.

2.9 Supersingular isogeny graph

In this section we study the supersingular isogeny graph. There are approximately $\frac{p}{12}$ supersingular isomorphism classes in characteristic p and they all have a representative defined over $\mathbb{F}_{p^2}$. Moreover all isogenies between supersingular curves defined over $\mathbb{F}_{p^2}$ are also defined over $\mathbb{F}_{p^2}$. So the supersingular graph over $\mathbb{F}_{p^2}$ is equivalent to the graph over $\overline{\mathbb{F}_p}$.
The supersingular isogeny graph has a vertex for each isomorphism class usually identified by its j-invariant. Two vertices are connected iff there is an isogeny between the curves they represent. Sometimes we are interested in a specific subgraph of this graph that only contains edges corresponding to ℓ-isogenies.
Recall that a curve is supersingular iff its trace is 0 mod p. There are 5 options for the trace of curves defined over $\mathbb{F}_{p^2}$: $0, \pm p, \pm 2p$. The number of isomorphism classes of curves with trace 0 or $\pm p$ is at most 6, so they are less interesting in our discussion. The remaining curves have trace $2p$ when $\#E(\mathbb{F}_{p^2}) = (p-1)^2$ and $-2p$ when $\#E(\mathbb{F}_{p^2}) = (p+1)^2$. Remember that two curves are isogenous iff they have the same number of points, therefore these two classes of curves form exactly two connected components in the graph. These components are isomorphic and have the same j-invariants, in fact each curve in one component has its twist in the other component. We focus on one of the components and call it the **supersingular ℓ-isogeny graph**.
This graph is $\ell + 1$-regular and Ramanujan, meaning its an optimal expander, in other words all its non-trivial eigenvalues are less that or equal to $2\sqrt{\ell}$ in absolute value.

2.10 Orders in quadratic number fields

The $\mathbb{F}_p$-rational endomorphism ring of a supersingular elliptic curves is isomorphic to an order in an imaginary quadratic field. The relation between orders in quadratic fields and their ideals reveals a lot of information regarding the structure of the $\mathbb{F}_p$-restricted supersingular isogeny graph. In this section we look at parts of the theory of quadratic field relevant to our work. Most of the materials in this section is taken from [Kla12].
Throughout this section let d be a square-free integer and $k = \mathbb{Q}(\sqrt{d})$ a quadratic number field.

Definition 22 (Ring of integers)**.** In any number field F the set of elements $\alpha \in F$ such that α is the root of a monic polynomial with integer coefficients is called the integer ring and denoted by $\mathcal{O}_F$. The integer ring has rank equal to the rank of $F/\mathbb{Q}$ so it is a lattice and therefore an order in F.

The discriminant (also called fundamental discriminant) of k, denoted Δ_k is d in case $d \equiv 1 \mod 4$ and $4d$ otherwise. The ring if integers in a quadratic field k is the unique maximal order in k. Additionally it has the integer basis $(1, \omega_k)$, where $\omega_k = \frac{\Delta_k + \sqrt{\Delta_k}}{2}$.
Let $\mathcal{O}$ be an order in k since $\mathcal{O}_k$ is the unique maximal order, $\mathcal{O} \subseteq \mathcal{O}_k$, moreover $f = [\mathcal{O}_k : \mathcal{O}]$ is finite and is called the conductor of $\mathcal{O}$. $\mathcal{O}$ has the integer basis $(1, f\omega_k)$.
Remark. Let p be an odd prime and $k = \mathbb{Q}(\sqrt{-p})$.

- If $p \equiv 1 \mod 4$, the discriminant of k is $-4p$ and its ring of integers is $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$.
- If $p \equiv 3 \mod 4$, the discriminant of k is $-p$ and its ring of integers is $\mathbb{Z}[\sqrt{-p}]$.

Definition 23 (Fractional ideal)**.** Let $\mathcal{O}$ be an order in k. A fractional $\mathcal{O}$-ideal is a $\mathcal{O}$-submodule $a \subset k$ such that, there is some $r \in \mathcal{O}\backslash\{0\}$ so $ra \subseteq \mathcal{O}$. All $\mathcal{O}$-fractional ideals are of the form αa, where $\alpha \in k^*$ and a is a $\mathcal{O}$-ideal.

Definition 24 (Norm)**.** For all non-zero $\mathcal{O}$-ideals a, $|\mathcal{O}/a|$ is finite. We define the norm of a denoted by $N(a)$, to be this integer value. Equivalently, this is equal to the gcd of all elements in a.

Definition 25 (Inevitable ideal)**.** A fractional $\mathcal{O}$-ideal is invertable if there exists a fractional $\mathcal{O}$-ideal b such that $ab = \mathcal{O}$.

Definition 26 (Proper ideal)**.** A fractional $\mathcal{O}$-ideal a is called proper if $\mathcal{O} = \{\beta \in k | \beta a \subset a\}$. Notice that $\mathcal{O} \subseteq \{\beta \in k | \beta a \subset a\}$ is always true.

Theorem 14. *A fractional $\mathcal{O}$-ideal is proper if and only if it is invertable.*

Theorem 15. *All fractional ideals of the integer ring are proper therefore invertable.*

Definition 27 (Ideal class group)**.** Let $\mathcal{O}$ be an order in k. The set of proper $\mathcal{O}$-fractional ideals denoted $I(O)$, is an abelian group under multiplication. The set of principal $\mathcal{O}$-fractional ideals denoted $P(O)$, is a normal subgroup of $I(O)$.
The quotient $I(O)/P(O)$ is called the **ideal class group** of order $\mathcal{O}$ denoted by $\mathrm{cl}(\mathcal{O})$.
The class group of $\mathcal{O}_k$ is also referred to as the fundamental class group or the class group of k.
Size of the class group is denoted by $h(\mathcal{O})$ and for all orders $\mathcal{O}$ in k, $h(\mathcal{O}_k)|h(\mathcal{O})$.

The $\mathbb{F}_p$-rational endomorphism ring of a supersingular elliptic curve in characteristic p is isomorphic to an order in $\mathbb{Q}(\sqrt{-p})$ and the equivalence classes of their isogenies correspond to the class group of the order, so the structure of the class group of imaginary quadratic orders are important in studying rational isogenies. The following theorem due to Bach [Bac89] is a very useful statement about the rank of the class group.

Theorem 16. *Assuming GRH, ideal of norm less that or equal to $6\log(|d|)^2$ generate the class group of an order $\mathcal{O}$, where d is the discriminant of $\mathcal{O}$.*

This implies that the $\mathbb{F}_p$-restricted supersingular isogeny graph with isogenies of prime degree less that or equal to $6\log(4p)^2$ is connected. Since the conductor of the rational endomorphism ring of any supersingular curve is at most 2.
We don't always need all these isogenies to connect the graph. For some $B \in \mathbb{N}$, let $L = \{\ell < B | \ell \text{ is prime and } \left(\frac{-p}{\ell}\right) = 1\}$. In sub-exponential time we can decide if ideals with norm in L generate the class group.

From now on we assume k to be an imaginary quadratic field $\mathbb{Q}(\sqrt{-d})$, where d is positive and square-free. Also $\mathcal{O}$ is an order in k with conductor f.

Theorem 17. *Let $m \neq 0$ be an integer.Any ideal class in $cl(O)$ has a representative of norm coprime to m.*

Definition 28. An $\mathcal{O}$-ideal a is prime to f iff $a + f\mathcal{O} = \mathcal{O}$.

Theorem 18. *An $\mathcal{O}$-ideal a is prime to f iff $N(a)$ is coprime to f. Additionally, all ideals prime to f are proper.*

Corollary 6. *The set of ideals prime to f denoted by $I(O, f)$ is subset of $I(O)$ and closed under multiplication. Additionally, the set of principal ideals prime to f denoted by $P(O, f)$ is a normal subgroup of $I(O, f)$.*

It turns out $\mathrm{cl}(O)$ can be described in terms of ideals coprime to f.

Theorem 19.

$$I(O, f)/P(O, f) \simeq I(O)/P(O) = cl(O)$$

This is intuitively true since all ideal classes have a representative prime to f, so restricting to ideals prime to f shouldn't change the structure of the class group.
Now we extend the concept of ideals prime to conductor, to the integer ring.

Definition 29. Let $m > 0$ be an integer. A $\mathcal{O}_k$-ideal a is prime to m iff $a + m\mathcal{O}_k = \mathcal{O}_k$. This is equivalent to $gcd(N(a), m) = 1$.

Theorem 20. *There is an isomorphism between $I(\mathcal{O}, f)$ and $I(\mathcal{O}_k, f)$ that preserves norms. $\phi : I(\mathcal{O}, f) \to I(\mathcal{O}_k, f), \phi(a) = a\mathcal{O}, \phi^{-1}(b) = b \cap \mathcal{O}$.*

Let $P = \phi(P(\mathcal{O}, f))$, the class group $\mathrm{cl}(\mathcal{O})$ can be specified only in terms of ideals of the integer ring. $\mathrm{cl}(\mathcal{O}) = I(\mathcal{O}_k, f)/P$.

2.11 $\mathbb{F}_p$-restricted supersingular isogeny graph

About $\sqrt{p}$ supersingular j-invariants are defined over $\mathbb{F}_p$. The trace of these curves is 0 therefore $\#E(\mathbb{F}_p) = p + 1$ and there is a unique isogeny class containing all curves defined over $\mathbb{F}_p$.
Considering only the isogenies defined over $\mathbb{F}_p$, the isomorphism classes of curves and their isogenies form the **$\mathbb{F}_p$-restricted supersingular isogeny graph**. This graph is a volcano with depth at most 2.
The $\mathbb{F}_p$-restricted isogeny graph is not a subgraph of the full supersingular isogeny graph. Every supersingular elliptic curve E/p has the same j-invariant as its quadratic twist and since all supersingular curves have $p + 1$ points over $\mathbb{F}_p$ they are isogenous, however E is not isomorphic to its twist over $\mathbb{F}_p$. For this reason it doesn't make sense to identify the vertices of the $\mathbb{F}$-restricted supersingular isogeny graph by their j-invariant.

The $\mathbb{F}_p$-restricted endomorphism ring of $E/\mathbb{F}_p$ denoted $\mathrm{End}_{\mathbb{F}_p}(E)$ is isomorphic to either $\mathbb{Z}[\sqrt{-p}]$ or $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$. An isogeny $\phi : E \to E'$ is called ***horizontal*** if $\mathrm{End}_{\mathbb{F}_p}(E) \simeq \mathrm{End}_{\mathbb{F}_p}(E')$. By this definition horizontal isogeny classes are formed. There are two horizontal isogeny classes:

1. The curves with endomorphism ring isomorphic to $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$ are called the *surface*.
2. The curves with endomorphism ring isomorphic to $\mathbb{Z}[\sqrt{-p}]$ are the *floor*.

Notice that when $p \equiv 1 \mod 4$ the surface and the floor coincide since $\mathbb{Z}[\sqrt{-p}] = \mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$. The size of the horizontal isogeny classes depends on p modulo 4.

1. If $p \equiv 1 \mod 4$ then all curves have endomorphism ring isomorphic to $\mathbb{Z}[\sqrt{-p}]$ so there is only one horizontal isogeny class and this class has h curves up to $\mathbb{F}_p$-isomorphism where h is the class number of $\mathbb{Q}(\sqrt{-p})$. This imaginary quadratic field is the $\mathbb{F}_p$-restricted endomorphism algebra of all curves.

2. If $p \equiv 3 \mod 4$ then the surface has h curves (again up to $\mathbb{F}_p$-isomorphism) and the floor has h or $3h$ curves depending on whether $p \equiv 3$ or $p \equiv 7 \mod 8$ respectively.

The structure of the $\mathbb{F}_p$-restricted ℓ-isogeny graph depends on ℓ:

- If ℓ is an odd prime and $\left(\frac{-p}{\ell}\right) = -1$, no ℓ-isogeny is defined over $\mathbb{F}_p$ so the graph is empty.
- If ℓ is an odd prime and $\left(\frac{-p}{\ell}\right) = 1$, ever curve has exactly two horizontal ℓ-isogenies, so each horizontal isogeny class is a bunch of cycles.
- If $\ell = 2$ and $p \equiv 1 \mod 4$, there is only one horizontal isogeny class and every curve has exactly one horizontal ℓ-isogeny. See Figure 2.

- If $\ell = 2$ and $p \equiv 3 \bmod 4$, there are two horizontal isogeny classes. Every curve on the floor has exactly one non-horizontal ℓ-isogeny going to the surface. The curves on the surface:
  - If $p \equiv 7 \bmod 8$, they have exactly two horizontal isogenies and one non-horizontal isogeny going to the floor that is the dual of the one coming from the floor. See Figure 3.
  - If $p \equiv 3 \bmod 8$, they have three non-horizontal ℓ-isogenies going to the floor that are duals of isogenies coming form the floor. See Figure 4.

Figure 2: $\mathbb{F}_{101}$-restricted 2-isogeny graph

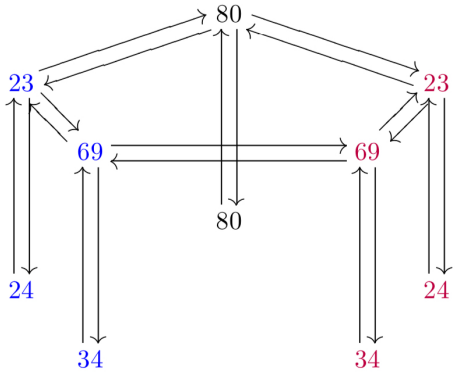


Figure 3: $\mathbb{F}_{103}$-restricted 2-isogeny graph

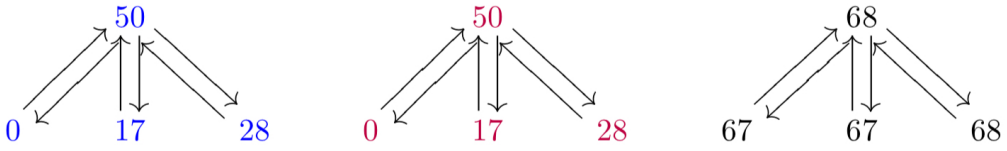


Figure 4: $\mathbb{F}_{83}$-restricted 2-isogeny graph

These illustrations are taken from [DG16]. More examples can be found in appendix A of their paper.

When working with $\mathbb{F}_p$-restricted graphs we are mainly interested in cases that there are large cycles in the graph. As we saw, cycles exist when ℓ is an odd prime and $\left(\frac{-p}{\ell}\right) = 1$ or $\ell = 2$ and $p \equiv 7 \bmod 8$. In both of these cases the graph consists of a number of cycles therefore a non-backtracking walk of arbitrary length T can be identified by its starting

vertex, length and direction of travel on the cycle. Notice that this identification has logarithmic length in the length of the walk.

Let $E/\mathbb{F}_p$ be a a supersingular elliptic curve. The $\mathbb{F}_p$-isogenies of degree ℓ on E correspond to Galois invariant subgroups of $E[\ell]$.

Definition 30 (Galois invariance)**.** Let k be a field and f/k some extension field. $Gal(f/k)$ is the set of all automorphisms of f that fix k. This set forms a group under composition. A set S is called Galois invariant if S is invariant under the action of $Gal(f/k)$, in other words $\sigma(S) = S$ for all $\sigma \in Gal(f/k)$.

We know the characteristic equation of the frobenius endomorphism is $\pi^2 - tr(\pi)\pi + q = 0$. Since E is supersingular $tr(\pi) \equiv 0 \mod p$, on $\mathbb{F}_p$ this equation reduces to $\pi^2 + p = 0$. There are three possibilities:

1. $\pi^2 + p$ is irreducible
2. $\pi^2 + p = (\pi - a)(\pi - b), a \neq b$
3. $\pi^2 + p = (\pi - a)^2$

Recall that $E[\ell] \simeq (\mathbb{Z}_\ell)^2$ so its a $\mathbb{Z}_\ell$-vector space of dimension 2 and $\pi : E[\ell] \to E[\ell]$ is a linear map. Let M_π be the matrix representation of π then $M_\pi^2 + pI_2 \equiv 0 \mod \ell$.
Let $G = \langle P \rangle$ be a non-trivial cyclic Galois invariant subgroup of $E[\ell]$, then there exists some $a \in \mathbb{F}_p$ such that $\pi(P) = a \cdot P$. So M_π has eigenspace $\langle P \rangle$ with eigenvalue a and the other eigenvalue is p/a. Let $\{P, Q\}$ be a basis for $E[\ell]$ then we can see $M_\pi = \begin{pmatrix} a & 0 \\ b & a \end{pmatrix}$ or $M_\pi = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}$.
In case (1) $E[\ell]$ can have 1 or $\ell + 1$ cyclic Galois invariant subgroup depending on the matrix M_π. In case (2) $E[\ell]$ has 2 Galois invariant subgroups and in case (3) it has non. For more details on supersingular elliptic curves over prime fields we refer to [DG16].

1. Case (1) happens when ℓ is an odd prime and $-p$ does not split mod ℓ. In this case, the ideal $\ell\mathcal{O}$ is irreducible in $\mathbb{Q}(\pi)$ therefore no isogeny is defined over $\mathbb{F}_p$.
2. In this case $\ell\mathcal{O} = I\overline{I}$, where $I = (\ell, \pi - a)$ and $\overline{I} = (\ell, \pi - b)$ are prime ideals and each define an ℓ-isogeny. So each curve has exactly two horizontal isogenies.
3. The polynomial $\pi^2 + p \mod \ell$ has a repeated root only when $\ell = 2$. In this case $|E(\mathbb{F}_p)[2]|$ is 2 if $\mathrm{End}_{\mathbb{F}_p}(E) = \mathbb{Z}[\sqrt{-p}]$ and it is 4 if $\mathrm{End}_{\mathbb{F}_p}(E) = \mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$. In the former case E has only one non-trivial subgroup defined over $\mathbb{F}_p$ and in the later case $E(\mathbb{F}_p)[2] = E[2] \simeq (\mathbb{Z}_l)^2$, so it has 3 non-trivial subgroups.

2.12 CSIDH

In this section we look at CSIDH, a group action on supersingular elliptic curves define over $\mathbb{F}_p$ proposed by Castryck and Decru [CLM+18]. Let $\mathcal{O}$ be an order in $\mathbb{Q}(\sqrt{-p})$ and $\pi \in \mathcal{O}$, we denote the set of supersingular elliptic curves defined over $\mathbb{F}_p$ with $\mathrm{End}_{\mathbb{F}_p}(E) \simeq \mathcal{O}$ and frobenius endomorphism corresponding to π by $\mathcal{E}(\mathcal{O}, \pi)$.
cl($\mathcal{O}$ acts on $\mathcal{E}(\mathcal{O}, \pi)$ freely and transitively by the action of ideal isogenies on the curves:

$$\mathrm{cl}(\mathcal{O}) \times \mathcal{E}(\mathcal{O}, \pi) \to \mathcal{E}(\mathcal{O}, \pi)$$

$$(a, E) \to E/a = E/\phi_a$$

a is an integer representative of its ideal class.

By Theorem 16 all ideals can be represented as a product of small prime ideal, hence computing the action of an ideal class can be reduced to computing action of ideals of small prime norm. As discussed before if ℓ is split then $\ell\mathcal{O} = I\overline{I}$. $I = (\ell, \pi - \lambda)$ and $\overline{I} = (\ell, \pi - p/\lambda)$ are prime ideals of norm ℓ where λ and p/λ are eigenvalues of π on $E[\ell]$. An efficient way to compute the action of an ℓ-ideal is to compute a basis of the ℓ-torsion on some extension field and compute the eigenspace of π on that basis. Then apply Velu's formulas to a generator in the correct eigenspace. Let $\{P, Q\}$ be a generator of $E[\ell]$, λ an eigenvalue of π with eigenspace $\langle G\rangle$ then ϕ_G can be computed using Velu's formulas and it corresponds to the ideal $I = (\ell, \pi - \lambda)$.
An important factor in efficiency of this method is the dimension of the extension where the torsion basis is defined. The optimal case is when $\lambda = 1$ and $p/\lambda = -1$. $\lambda = 1$ corresponds to a $\mathbb{F}_p$-rational ℓ-torsion point and p/λ to a point with x-coordinate in $\mathbb{F}$ and y-coordinate in $\mathbb{F}_{p^2}\backslash\mathbb{F}_p$.

To get the optimal performance, CSIDH is used with a prime of form $p = 4\cdot\ell_1\cdot\ell_2\cdot....\cdot\ell_n - 1$ where ℓ_is are odd primes. $p \equiv -1 \mod \ell_i$ and since $0 \equiv \pi^2 + p \equiv \pi^2 - 1 \mod \ell_i$, ℓ_i is split and $\ell_i\mathcal{O} = I_i\overline{I_i}$, where $I_i = (\ell_i, \pi - 1)$ and $\overline{I_i} = (\ell_i, \pi + 1)$. Notice that this is the optimal case mentioned above.
For sampling uniform elements in $\mathrm{cl}(\mathcal{O})$ we want to know it's structure however, computing the class group of a imaginary quadratic field with large discriminant is a classical hard problem. So generally for large p, $\mathrm{cl}(\mathcal{O})$ is not known. Another method for picking random elements is to specify ideals by their representation as product of prime ideal with small norm, and pick random exponents for each prime ideal. Let $e_i \in [-B, +B]$ be randomly selected integers then $a = \ell_1^{e_1} \cdot \ell_2^{e_2} \cdot ... \cdot \ell_n^{e_n}$ is a random element in $\mathrm{cl}(\mathcal{O})$. However a is not uniformly distributed, its distribution depends on the conductor of the order $\mathcal{O}$ and the range $[-B, +B]$, but usually for a reasonably small B the support of a is large enough to provide cryptographic security.

Since $p \equiv 3 \mod 4$, $E_0 : y^2 = x^3 + x$ is supersingular and $\mathrm{End}_{\mathbb{F}_p}(E_0) \simeq \mathbb{Z}[\pi]$. The current setup is an efficient **commutative** group action and finding the inverse of an action is hard.
This problem is called "Group Action Inversion Problem (GAIP)" and is defined as follows:

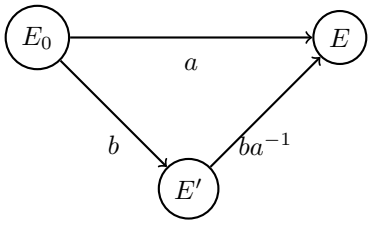
Definition 31 (GAIP)**.** Given a curve $E/\mathbb{F}_p$ such that $\mathrm{End}_{\mathbb{F}_p} \simeq \mathcal{O}$, find an integral ideal $a \subset \mathcal{O}$ so $E = aE_0$.

Assuming hardness of GAIP, we can design a zero-knowledge proof of knowledge for ideal classes. Let $a \in \mathrm{cl}(\mathcal{O})$ and $E = aE_0$, a prover P knowing a wants to prove its knowledge to a verifier V without reveling any extra information.
This proof is similar to the graph isomorphism proof.

1. The prover chooses a random $b \in \mathrm{cl}(\mathcal{O})$ and sends $E' = bE_0$ as its commitment.

2. The verifier chooses a random bit $c \in \{0, 1\}$ as the challenge.

3. If $c = 0$ the prover sends $r = b$ as its response and verifier checks $rE_0 = E'$,
   if $b = 1$ the prover sends $r = ba^{-1}$ as its response and the verifier checks $rE' = E$.



Completeness: If the prover knows a clearly she can convince the verifier.

Special soundness: Assume $(E', 0, r)$ and $(E', 1, r')$ are two accepting transcripts (Since the challenge space is $\{0, 1\}$ the different challenges are uniquely determined). A witness can be extracted as follows: $r'(rE) = r'(E') = E \to (r'r)E = E$.

HVZK: If challenge bit c is 0, simulator picks a random $b \in \mathrm{cl}(\mathcal{O})$ and outputs (bE_0, c, b). If $c = 1$, simulator picks random $b \in \mathrm{cl}(\mathcal{O})$ and outputs $(b^{-1}E, c, b)$. Notice that the first message is uniformly distributed so the simulator's output is identical to the view of the honest verifier, hence, this proof is perfectly zero-knowledge for the honest verifier.

This proof system is a Σ-protocol so using the Fiat-Shamir transform it can be made into a non-interactive ZKPoK and a signature. In fact CSI-FiSh signature is a variant of this proof system.

In practice a variant of CSIDH called CSIDH-512 is used. CSIDH-512 uses 74 small prime (so $n = 74$) and random exponents are picked from $[-5, +5]$. The authors of CSIDH [CLM+18] assume that this random sampling covers approximately 2^{256} elements which is about half the entire class group and it provides about 128-bit classical security.

2.13 Orientations

In this section we look at oriented elliptic curves and isogenies. Studying orientations is a generalization of studying curves and isogenies defined over $\mathbb{F}_p$ and it will help us understand the concepts better. Throughout this section let k be a quadratic number field, $\mathcal{O}_k$ its ring of integers and $\mathcal{O}$ an arbitrary order in k.

Definition 32 (Orientation on elliptic curve)**.** A k-orientation on elliptic curve E is an embedding $\tau : k \to \mathrm{End}(E)^o$. τ is an $\mathcal{O}$-orientation if $\tau(\mathcal{O}) = \tau(k) \cap \mathrm{End}(E)$. Such (E, τ) is called an $\mathcal{O}$-oriented elliptic curve and E is said to be $\mathcal{O}$-orientable.
Notice that an $\mathcal{O}$-orientation τ, naturally extends to a k-orientation, since $\mathcal{O}$ spans k.

An isogeny $\phi : (E, \tau) \to E'$ induces an orientation τ' on E'.

$$\phi_*(\tau)(\alpha) = (\phi \circ \tau(\alpha) \circ \hat{\phi}) \frac{1}{\deg \phi}$$

From an oriented curve (E, τ) we can define two other oriented curves:

Oriented twist: The twist of (E, τ) is $(E, \overline{\tau})$, where $\overline{\tau}(\alpha) = \tau(\overline{\alpha})$.

Frobenius: The Frobenius of (E, τ) is $(E, \overline{\tau})^{(p)} = (E^{(p)}, \pi_*(\overline{\tau}))$.

Definition 33. Oriented isogeny An isogeny $\phi : (E, \tau) \to (E', \tau')$ is k-oriented if $\tau' = \phi_*(\tau)$. Furthermore, when $\deg \phi$ is prime and τ is an $\mathcal{O}$-orientation and τ' is a $\mathcal{O}'$-orientation,

1. If $\mathcal{O} = \mathcal{O}'$, then ϕ is called *horizontal*.
2. If $\mathcal{O} \subsetneq \mathcal{O}'$, then ϕ is called ascending.

3. If $\mathcal{O} \supsetneq \mathcal{O}'$, then ϕ is called descending.

The set of k-oriented isomorphism classes of $\mathcal{O}$-oriented supersingular elliptic curves defined over $\overline{\mathbb{F}_p}$ is denoted by $SS_\mathcal{O}$. We abuse notation by writing $(E,\tau) \in SS_\mathcal{O}$ and meaning (E,τ) is a representative of an isomorphism class in $SS_\mathcal{O}$.

Theorem 21. *$SS_\mathcal{O}$ is non-empty if and only if p doesn't split in k and doesn't divide the conductor of $\mathcal{O}$.*

In the previous section we looked at the $\mathbb{F}_p$-restricted supersingular isogeny graph. The $\mathbb{F}_p$-rational endomorphism ring of a supersingular curve $E/\mathbb{F}_p$ is isomorphic to $\mathbb{Z}[\pi]$. This isomorphism is an embedding and therefore supersingular curves defined over $\mathbb{F}_p$ are $\mathbb{Z}[\pi]$-oriented. For easier notation we only write the non-trivial generator of the order, so we can write π-oriented. Furthermore, the isogenies defined over $\mathbb{F}_p$ are $\mathbb{Q}(\pi)$-oriented.
Now we look at the case of general orientations. Let $(E,\tau) \in SS_\mathcal{O}$, Δ be the discriminant of $\mathcal{O}$ and $\ell \neq p$ be a prime. The k-oriented ℓ-isogenies of (E,τ) are as follows:

- There are $\ell - \left(\frac{\Delta}{\ell}\right)$ descending isogenies.
- If $\mathcal{O}$ is maximal at ℓ, there are $\left(\frac{\Delta}{\ell}\right) + 1$ horizontal and no ascending isogenies.
- If $\mathcal{O}$ is non-maximal at ℓ, there is 0 horizontal and one ascending isogeny.

Similar to the π-orientation ($\mathbb{F}_p$-restricted) other orientations also give rise to a group action. Let $(E,\tau) \in SS_\mathcal{O}$, an $\mathcal{O}$-ideal I defines a subgroup $E[I] = \bigcap_{\alpha \in I} \mathrm{Ker}(\tau(\alpha))$ and an isogeny $\phi_I : E \to E^I$ with kernel $E[I]$ and degree $N(I)$.
This construction induces the action of $\mathcal{O}$-ideals on $SS_\mathcal{O}$ by

$$I \cdot (E,\tau) = (E^I, (\phi_I)_*(\tau)).$$

The action of an ideal is invariant under multiplication from right by some $\gamma \in \mathcal{O}^*$ so it factors to the ideal class group and we get the group action:

$$\begin{gathered} \mathrm{cl}(\mathcal{O}) \times SS_\mathcal{O}(p) \to SS_\mathcal{O}(p) \\ [I] \cdot (E,\tau) \to I \cdot (E,\tau) = (E^I, (\phi_I)_*(\tau)) \end{gathered}$$

2.14 Pairings

Definition 34 (Pairing of abelian groups)**.** Let G_1 and G_2 be abelian groups with additive notation and $N \in \mathbb{N}$ such that the exponents of both groups divide N, also let G_3 be the cyclic group of order N with multiplicative notation.
A pairing is a non-degenerate, bilinear map $e : G_1 \times G_2 \to G_3$.

Non-degeneracy: For all $P \in G_1 \backslash \{0\}$ there is $Q \in G_2$ such that $e(P,Q) \neq 1$. And vice versa.

Bilinearity: For all $P, P' \in G_1$ and $Q, Q' \in G_2$, $e(P+P',Q) = e(P,Q)e(P',Q)$ and $e(P,Q+Q') = e(P,Q)e(P,Q')$.
Notice that this implies for all $a \in \mathbb{N}$, $e(aP,Q) = e(P,Q)^a = e(P,aQ)$.

Elliptic curves are abelian groups so naturally we want pairings on elliptic curves. There are several pairings on elliptic curves for example Tate or Weil pairing. We only introduce the Weil pairing.

Definition 35 (Weil pairing.)**.** Let E be an elliptic curve defined over $\mathbb{F}_q$ of characteristic p and $N \in \mathbb{Z}$ be coprime to p.
Let $\mathbb{F}_{q^k} = \mathbb{F}_q(E[N])$ be the extension of $\mathbb{F}_q$ generated by the coordinates of N-torsion points of E and μ_N denotes the group of N-th roots of unity in $\mathbb{F}_{q^k}$. The N-th Weil

pairing is $e_N : E[N] \times E[N] \to \mu_N \subseteq \mathbb{F}^*_{q^k}$.
Furthermore for a fixed non-zero $Q \in E[N]$, $f_Q(P) = e_N(P, Q)$ and $g_Q(P) = e_N(Q, P)$ are isomorphisms.

Theorem 22. *Let E be an elliptic curve defined over $\mathbb{F}_q$ of characteristic p. Let $N \in \mathbb{N}$ be coprime to p such that $N|\#E(\mathbb{F}_q)$ but does not divide $q-1$. $E(\mathbb{F}_{q^k})$ contains $E[N]$ iff $N|q^k - 1$.*

Corollary 7. *When $N \nmid q-1$ The extension degree k in definition of the Weil pairing is the smallest $k \in \mathbb{N}$ such that $N|q^k - 1$.*

Theorem 23 (Pairing equation of isogenies)**.** *Let $\phi : E \to E'$ be an isogeny of elliptic curves defined in characteristic p and N a natural number coprime to p. Let $P \in E[N]$ and $Q \in E'[N]$. e and e' are respectively the Weil pairing on E and E'.*

$$e_N(P, \hat{\phi}(Q)) = e'_N(\phi(P), Q) \tag{3}$$

3 Time-release cryptographic primitives

3.1 The origin: Time-lock puzzles

The first time-release encryption primitive without using long term trusted agents was proposed in [RSW96]. In this scheme a "puzzle" is created that requires some fixed amount of sequential computations to solve. Solving the puzzle reveals some information that enables an otherwise infeasible computations. The following is the formal definition of a Time-lock puzzle.

Definition 36 (Time-lock puzzle (TLP))**.** A TLP consists of three algorithms.

- $Setup(1^\lambda, T) \to pp$, takes a security parameter λ and a delay parameter T and outputs the public parameters. *Setup* has to run in $poly(\lambda, \log(T))$.
- $Gen(pp, s) \to Z$, takes public parameters pp and a secret s and outputs a puzzle Z. *Gen* has to run in $poly(\lambda)$.
- $Sol(pp, Z) \to s$, takes the public parameters and the puzzle and solves the puzzle to retrieve the secret s. *Sol* has to run in $poly(\lambda, T)$.

The three algorithms form the functionality of a TLP, in addition they have to be correct and sound.

Correctness. A TLP is correct if for all $\lambda, T \in \mathbb{N}$ and secret s and pp generated by *Setup*, we have $Pr[Sol(pp, Gen(pp, s)) = s] = 1$.

Soundness. The security game is defined as follow. An adversary consists of two poly-bounded algorithms A_1, A_2.

1. Challenger samples $Setup(1^\lambda, T) \to pp$, and sends pp to the adversary.
2. The adversary runs $A_1(pp) \to (st, s_0, s_1)$ and generates two secrets s_0, s_1 and sends them to the challenger.
3. The Challenger samples a random bit $b \in \{0, 1\}$ and sends the puzzle $Gen(pp, s_b) \to Z_b$ to the adversary.
4. The adversary runs $A_2(st, Z_b) \to b'$ and outputs b'.
5. The adversary wins if A_2 terminates in time less than T and $b' = b$.

A TLP is sound if any adversary running A_1 in polynomial time has negligible advantage.

Since the introduction of time-lock puzzles some new time-release primitives have been introduced. One of the limitations of TLPs is that if there are many participants each with a secret and the goal is to compute some information from these secrets (for example in an auction), a large number of puzzles need to be solved individually. A proposed method to overcome this limitation is homomorphic TLP.

A **homomorphic TLP** (HTLP for short) is similar to a regular TLP in terms of security and functionality with the added feature that puzzles can be aggregated and only solved once, so when the final puzzle is solved the aggregated information is revealed.
Unfortunately, all known efficient HTLPs are single homomorphic and the only known fully homomorphic TLPs are either based on FHE or indistinguishablity obfuscation (IO). FHE is not optimal for this purpose as current methods are slow. Also known IO constructions are inefficient.
De Feo and Burdges introduced a new time-release primitive called **Delay Encryption**. The goal of this scheme is that each participant selects a public key, encrypt their messages and publish the ciphertext. After a delay function is computed on a single input the secret key corresponding to each public key can be computed efficiently and ciphertexts can be decrypted.

3.2 Verifiable Delay functions

Before discussing the details of delay encryption, first we introduce another time-release cryptographic primitive called **Verifiable Delay function**. A function $f : X \to Y$ is a VDF if computing $f(x)$ is a slow and sequential process for all $x \in X$ but for any $y \in Y$ verifying $f(x) = y$ is efficient. We define a VDF concretely as follows.

Definition 37 (Verifiable Delay Function)**.** A VDF consists of three algorithms:

- $Setup(\lambda, T) \to (ek, vk)$, takes security parameter λ and delay parameter T and outputs evaluation key ek and verification key vk. $Setup$ should run in time $poly(\lambda, T)$.
- $Eval(ek, x) \to (y, \pi)$, takes evaluation key and x as input and outputs y and some proof π of $f(x) = y$. This process is meant to infeasible in time less than T.
- $Verify(vk, x, y, \pi) \to True, False$, takes verification key, x, y and proof π and outputs a truth predicate, if π is a valid proof that $f(x) = y$.

A VDF has to satisfy three security notions:

Completeness: The honest evaluator always convinces the verifier. This notion is concretely captured in the VDF-completeness game:

1. V chooses delay parameter T and security parameter λ.
2. V generates the public parameters $Setup(\lambda, T) \to (ek, vk)$.
3. V chooses some random $x \in X$ and sends (ek, T, x) to P.
4. P honestly runs $Eval(ek, x)$ to get y and sends it to V.
5. V runs $Verify(vk, x, y, \pi)$. P succeeds if $Verify$ returns true.

A VDF is complete if the honest evaluator has always wins the VDF-completeness game.

Soundness: The dishonest evaluator can't pass verification with non-negligible probability. A VDF has soundness error ϵ if for any PPT algorithm A and $x \in X$ the following holds.

$$Pr[Verify(vk, x, y', \pi') = true | (vk, ek) \leftarrow Setup(T, \lambda), (y', \pi') \leftarrow A(ek, x), y' \neq f(x)] \leq \epsilon$$

Sequentiality: It is impossible the compute $f(x)$ for any $x \in X$ in time less that $o(T)$ even with $poly(T)$ many CPUs.

Notice that according to the original definition of VDFs in [BBBF18] the *Setup* process has to run in $poly(\lambda)$ but De Feo et al. relax this requirement because the *Setup* process in their isogeny-based VDF runs in $poly(\lambda, T)$.

3.3 Delay encryption

In delay encryption there are no sender/receiver pairs, messages are encrypted in a session identified by a session identifier. Each session has a session key that has to be computed through a sequential function, the session key allows the decryption of all ciphertexts in that session. Delay encryption is concretely defined as follows.

Definition 38 (Delay encryption)**.** A delay encryption scheme consists of four algorithms:

- $Setup(\lambda, T) \rightarrow (ek, pk)$, takes security parameter λ and delay parameter T and outputs extraction key ek and public (encryption) key pk. *Setup* has to run in $poly(\lambda, T)$, the extraction key has to be have size $poly(\lambda, T)$ but the public key has to have size $poly(\lambda)$.
- $Extract(ek, id) \rightarrow idk$, takes the extraction key and the session id $id \in \{0,1\}^*$ and outputs the session key idk. The extraction process is expected to take time exactly T.
- $Encaps(pk, id) \rightarrow (c, k)$, takes as input a public key pk and a session id $\in \{0,1\}^*$ and outputs a key $k \in K$ and ciphertext $c \in C$. c can be viewed as the encryption of the key k (Similar to KEM). The encapsulation process should run in $poly(\lambda)$ time.
- $Decaps(pk, id, idk, c) \rightarrow k$, takes as input public key, session id, session key and encapsulated key. It outputs a key k. The decapsulation processes should run in time $ploy(\lambda)$.

Encaps and *Decaps* can be used alongside a symmetric key encryption to form a hybrid encryption scheme. In this scheme k will be the symmetric key and c will be the public key.Similar to any cryptographic primitive we define notions of soundness and correctness.

Correctness: A delay encryption scheme is correct if for any (ek, pk) generated by *Setup* and any session id we have

$$Pr[Decaps(pk, id, idk, c) = k | idk = Extract(id, ek) \wedge (c, k) = Encaps(pk, id)] = 1$$

Soundness: The security of a delay encryption scheme is defined similar to that of public key encryption schemes, specifically identity based ones. However, notice that there are no secrets in a delay encryption scheme and anyone can run the extraction algorithm, hence the usual notion of indistinguishablity is not relevant, we define indistinguishablity relative to time: no adversary can distinguish a random string from the key k in time less than $o(T)$, but obviously anyone can do it in time T. Here by time we mean amount of sequential computation on a machine with polynomialy many CPUs.
We have the following $\Delta - IND - CPA$ security game:

1. Precomputation
   The adversary receives (ek, pk) and outputs algorithm D.
2. Challenge
   The challenger receives a random id and computes (c, k_0) ← Encaps(pk, id). It also chooses a random $k_1 \in K$ and a bit b ∈ {0, 1} and outputs (id, c, k_b).
3. Guess
   Algorithm D is run on (id, c, k_b). The adversary wins if D halts in time less than Δ and D(id, c, k_b) = b.

A delay encryption is Δ-indistinguishable CPA secure if any adversary running precomputation in poly(λ, T) has negligible advantage.
We say a delay encryption is Δ-indistinguishable CPA **quantum annoying** if any quantum adversary running precomputation in poly(λ, T) and outputting a classical algorithm has negligible advantage.

4 Delay from isogenies

4.1 Isogeny-based VDF

De Feo et al. introduced a VDF based on chains of low degree isogenies and pairings [DFMPS19]. Let $p = Nf - 1$ and N be primes such that discrete log problem in the group of N-th roots of unity in $\mathbb{F}_{p^2}$ is hard. Let $\phi : E/\mathbb{F}_p \to E'/\mathbb{F}_p$ be an isogeny defined over $\mathbb{F}_p$ and $P \in E[N], Q \in E'[N]$ then $e_N(P, \hat{\phi}(Q)) = e'_N(\phi(P), Q)$. We can use this to create a VDF.

1. The **setup** algorithm has two phases:
   (a) generating a curve $E/\mathbb{F}_p$ with hard endomorphism ring.
   (b) Doing a walk of length T in the $\mathbb{F}_p$-restricted ℓ-isogeny graph to get an isogeny $\phi : E/\mathbb{F}_p \to E'/\mathbb{F}_p$ of degree ℓ^T and computing the image of a random point $P \in E[N, \pi - 1]$ [1].

   The evaluation key is $ek = (\hat{\phi})$ and the verification key is $vk = (E, E', P, \phi(P))$
2. The **evaluation** algorithm takes a point $Q \in E'[N, \pi + 1]$ and outputs $\hat{\phi}(Q)$.
3. The **verification** check is $e_N(P, \hat{\phi}(Q)) = e'_N(\phi(P), Q)$.

It is clear that this VDF is **not** secure against an adversary with a discrete log oracle during the evaluation since the pairing equation can be cheated by solving a discrete log. For classical adversaries the security of this scheme is based on two assumptions.

1. Computing a chain of ℓ-isogenies is a sequential process and can not be performed in parallel.
2. No adversary can find a shorter isogeny with the same action on the N-torsion.

The first assumption is related to the inherent sequentiality of isogenies, it is a reasonable assumption however, there are no concrete arguments for it. It is important to remember that for time-release cryptography we always need some sequentiality assumptions in addition to other cryptographic assumptions since a provably sequential function implies

[1] $E[\pi - 1]$ is the kernel of the $\pi - 1$ endomorphism, these are the points with both XY coordinates defined over $\mathbb{F}_p$. Similarly $E[\pi + 1]$ are the points with X coordinate define over $\mathbb{F}_p$ but Y coordinate defined on a strictly higher extension.

$NC \neq P$ which is not currently proven.
The second assumption on the other hand, is more concrete since breaking this assumption implies solving the endomorphism ring problem (though only for curves defined over prime fields).
It is easily verifiable that an honest evaluator will always convince the honest verifier. We now prove that this VDF is perfectly sound.

Theorem 24. *The isogeny-based VDF is perfectly sound.*

Proof. This map $g_P : E[N, \pi - 1] \to \mu_N \subset \mathbb{F}_{p^2}; g_P(R) = e_N(P, R)$ is a group isomorphism so if $R \neq \hat{\phi}(Q)$ then $g(R) \neq g_P(\hat{\phi}(Q))$. □

On the topic of sequentiality of this we define the following game.

Definition 39. Isogeny shortcut game

1. Precomputaion.
   The adversary receives the public parameters (N, p, E, E', ϕ) and outputs an algorithm S in time $poly(\lambda, T)$.

2. Challenge.
   The Challenger outputs a uniform point $Q \in E'[N]$.

3. Guess.
   Algorithm S is run on input Q. The adversary wins iff S halts in time less than Δ and $S(Q) = \hat{\phi}(Q)$.

The isogeny shortcut game is said to be Δ-hard if whenever S runs in time less that Δ the adversary has negligible advantage. The isogeny-based VDF is Δ-sequential if the isogeny shortcut game is Δ-hard.
A VDF is **Δ-quantum annoying** if the isogeny shortcut game is Δ-hard for any quantum adversary outputting a classical algorithm.

There are some similarities between this VDF and the VDF based on squaring in groups of unknown order. We are using horizontal ℓ-isogenies defined over $\mathbb{F}_p$ with the same directions. These isogenies correspond to an ideal α of norm ℓ in the imaginary quadratic order $\mathcal{O} \simeq \mathrm{End}_{\mathbb{F}_p}(E)$. Composing isogenies correspond to multiplying ideals, so computing an isogeny chain ϕ of length T corresponds to computing α^T and $\hat{\phi}$ corresponds to $\alpha^{-T} = \overline{\alpha}^T$. While the other approach raises elements to power 2^T we only raise to power T because no analogue of the square-and-multiply method is known for composing isogenies. The main difference of these approaches is the proof of correctness.

4.1.1 Known class group

We have discussed walking on the $\mathbb{F}_p$-restricted graph, in this case an adversary that has access to quantum computation before the evaluation but not during it can still break this scheme.
$\mathrm{End}_{\mathbb{F}_p}(E)$ is an order in an imaginary quadratic field so its class group can be computed by a quantum adversary in polynomial time. Since the isogeny ϕ is defined over $\mathbb{F}_p$ it corresponds to a fractional ideal of this order, equivalently an element in its class group. Another representative of this ideal class with smaller norm can be computed.
When the class group is known, we can find a "short" basis, meaning that any element has a representation in this basis with small l_∞-norm. Such representation allows computing the action quickly.
Let $\mathcal{O}$ be an imaginary quadratic order and its class group $\mathrm{cl}(\mathcal{O})$ known, also let $B =$

$(I_{\ell_1}, I_{\ell_2}, ..., I_{\ell_n})$ be a generator of $\mathrm{cl}(\mathcal{O})$, where I_{ℓ_i}s are known prime ideals. Notice that B is not necessarily independent, in fact in many cases $\mathrm{cl}(\mathcal{O})$ is cyclic.
We define the relation lattice of B denoted by L_B as follows:

$$L_B = \{x \in \mathbb{Z}^n | \prod_{1 \leq i \leq n} I_{\ell_i}^{x_i} = (1)\}$$

Any ideal $a \in \mathrm{cl}(\mathcal{O})$ has a representation x in basis B and that representation reduced by the relation lattice gives a lattice vector x' in the fundamental region of the lattice so $||x'||_\infty \leq \lambda_n(B)$ where $\lambda_n(B)$ is the n-th successive minimum. $a' = \prod I_{\ell_i}^{x'_i}$ is in ideal in the same ideal class as a and it's action can be computed in time $O(n\lambda_n)$. Furthermore, notice that finding the reduced representation x' is an instance if the closest vector problem.
When $\mathrm{cl}(\mathcal{O})$ is known an LLL-reduced basis can be computed. Using this basis a representation with l_∞-norm approximately $\log |\Delta_\mathcal{O}|$ can be computed. For further discussion see 2.6 in [Sto12]

4.1.2 VDF on the full supersingular isogeny graph

This problem can be solved by walking on the full ℓ-isogeny graph, in that case the pairing equation used for verification is

$$e_N(P, (Tr \circ \hat{\phi})(Q)) = e'_N(\phi(P), Q)^2$$
$$Tr(P) = P + \pi(P)$$

The previous equation no longer works because the curve E' is no longer defined over $\mathbb{F}_p$ therefore the eigenvalues of π are not ± 1 so $E[\ell, \pi - 1]$ and $E[\ell, \pi + 1]$ are not orthogonal groups. See 5.2 in [DFMPS19] for details.

A disadvantage of using long walks in the general graph is that an ℓ-isogeny walk can't be described only by its length and a single bit even when $\ell = 2$.
De Feo and Burdges [BDF21] point out that for a one hour delay, an isogeny walk of length at least $T \simeq 7 \times 10^{10}$ should be used. For each 2-isogeny a coefficient in $\mathbb{F}_p$ needs to be stored, occupying around 1500 bits. Hence a total of approximately 12TiB is needed to store the isogeny walk. Even though only the evaluators need to store the walk and not the verifiers, it is still undesirable.
A possible solution to this issue that is not mentioned by the authors is to use the roots of the modular polynomial to specify an isogeny walk. In this method each step of the isogeny walk is determined by a single bit. The sequence of bits can be generated by a pseudo-random generator and the only information that needs to be stored is the seed of the generator. The details of this method are discussed in Section 6.

The following table is the summery of best known attacks for breaking sequentiality of the isogeny VDF assuming the endomorphism rings are hard.

	Classical over $\mathbb{F}_p$	Classical over $\mathbb{F}_{p^2}$	Quantum over $\mathbb{F}_p$	Quantum over $\mathbb{F}_{p^2}$
Computing shortcut	$L_p(1/2)$	$O(p^{1/2})$	$polylog(p)$	$O(p^{1/4})$
Pairing inversion	$L_p(1/3)$	$L_p(1/3)$	$polylog(p)$	$polylog(p)$

To achieve λ bits of security, the authors recommend using primes p with λ^3 bits and N with 2λ bits.

4.2 Isogeny-based delay encryption

In this section we look at the isogeny-based instantiation of delay encryption primitive introduced by De Feo and Burdges [BDF21].
This scheme is similar to a key encapsulation scheme. There are three algorithms:

- **TrustedSetup**(λ) is the same as in isogeny-VDF. It generates a random supersingular elliptic curve defined over $\mathbb{F}_p$.
- **Setup**$(E, E', P, \phi(P), id)$:
  1. Perform an ℓ-isogeny walk $\phi : E \to E'$.
  2. Choose a random point $P \in E[N]^o$ and compute $\phi(P)$.
  3. output $ek = (E', \phi)$ and $pk = (E', P, \phi(P))$.
- **Extraction** algorithm is the same as the evaluation algorithm.
  The session key is $sk = \hat{\phi}(Q)$ where $Q = H_1(id)$.
- The **Encaps** and **Decaps** algorithms work as follows:

Encaps$(E, E', P, \phi(P), id)$:

1. $r \leftarrow^{\$} \mathbb{Z}_N$
2. $Q \leftarrow H_1(id)$
3. $k = e'_N(\phi(P), Q)^r$
4. output (rP, k)

Decaps$(E, E', \hat{\phi}(Q), rP)$:

1. $k = e_N(rP, \hat{\phi}(Q))$
2. output k

$H_1 : \{0,1\}^\lambda \to E'[N]^o$ is a hash function mapping binary strings of length λ to points of order N in $E'(\mathbb{F}_{p^2})$.
After a user u runs Encaps, a pair (rP, k_u) is generated. Then u publishes rP and keeps k_u secret. Then k_u can be used as the key of a symmetric key encryption to encrypt messages. Before running Decaps on one has any information about k_u so the encrypted messages can't be decrypted. However, after running Decaps k_u can be quickly computed and all messages encrypted using k_u can be decrypted. Notice that if there are multiple users, each will generate their own symmetric key and encrypt messages, and once Decaps is run all symmetric keys can be computed quickly.
The completeness of this scheme follows from the pairing equation and bilinearity of pairings, $k = e'_N(\phi(P), Q)^r = e_N(P, \hat{\phi}(Q))^r = e_N(rP, \hat{\phi}(Q)) = k$. Now we discuss the security of this scheme

4.2.1 Security of isogeny-based delay encryption

There are four ways to attack this scheme.

1. **Attacking the computation**: An adversary could use efficient algorithms or specialized hardware to compute chains of isogenies faster than expected.
   We assumed that evaluating chains of isogenies is a sequential process, one could break this assumption and evaluate isogenies in parallel, however it is possible to evaluate isogenies faster than expected without breaking this assumption. At the moment the most efficient formula given in [BDF21] to evaluate a 2-isogeny requires two sequential multiplications and a parallel squaring, however we have no proof that this is optimal.
   Ideally we would like to prove evaluating a 2-isogeny requires two sequential multiplications but proving the lower bound of two parallel multiplications seems hard.
2. **Discrete logarithm attack**: Notice that if an adversary knows the randomness r used in the encapsulation process, he can compute $k = e'_N(\phi(P), Q)^r$. r can be found by solving the discrete log of rP in base P.
   This attack is why N has to be chosen large.

3. **Isogeny shortcut**: An adversary can find a shorter isogeny with the same torsion action and therefore evaluating a lower degree isogeny. Finding an isogeny $\psi : E \to E'$ independent of that ϕ implies a non-trivial endomorphism of E which is assumed hard to find. So no poly-bounded adversary can find a shorter isogeny.

4. An adversary could compute $k = e'_N(\phi(rP), Q)$, but this computation still takes time T.

The first attack method concerns how the notion of sequential computation is related to the wall-clock time. An adversary attacking this scheme using that method is still taking the expected number of sequential steps but taking said steps faster. This types of attacks aren't discussed in the original paper any further.
The discrete log attack is defeated by choosing a large enough torsion and Δ-hardness of the isogeny shortcut game guarantees sequentiality of $Extract$, but this is not enough to prove Δ-IND-CPA for this scheme.
The **Δ-Bilinear isogeny shortcut game** is defined as follows.

Definition 40. Bilinear isogeny shortcut game

1. Precomputaion.
   The adversary receives the public parameters (N, p, E, E', ϕ) and outputs an algorithm S in time $poly(\lambda, T)$.

2. Challenge.
   The Challenger outputs uniform points $R \in E[N, \pi - 1], Q \in E'[N]$.

3. Guess.
   Algorithm S is run on input R, Q. The adversary wins iff $S(R, Q) = e'_N(\phi(R), Q) = e_N(P, \hat{\phi}(Q))$.

The bilinear isogeny shortcut game is said to be Δ-hard if whenever S runs in time less that Δ the adversary has negligible advantage.

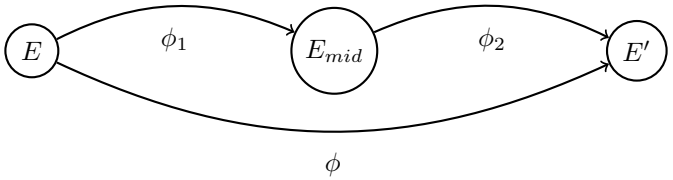
Theorem 25. *The isogeny-based delay encryption is Δ-IND-CPA secure, assuming Δ'-hardness of the bilinear isogeny shortcut game when $\Delta \in \Delta' - o(\Delta')$ and H_1 is modeled as a random oracle.*

Proof of this theorem is given in Theorem 1 of [BDF21]

4.3 Watermarking

In a distributed setup where several parties attempt the extraction process, it is desirable to be able to tie a computation to the party that performed said computation. For example, on the blockchain platform the participant that incurs the cost of computation receives some compensation for its work. The process of tying an evaluation to a participant is called **watermarking**.
In the context of VDFs based on squaring in groups of unknown order the results of the evaluation is the computed value and a proof of correctness. Watermarking is done by *proof watermarking*, where the participant generating the proof signs the proof so any other participant can verifier who did the computation.
However, in the isogeny-based VDF which underlies the extraction process of isogeny-based delay encryption, there is no proof, so proof watermarking is irrelevant. De Feo and Burdges propose the following method for watermarking an evaluation.
The isogeny walk $\phi : E \to E'$ is split into two equal length halves $\phi_1 : E \to E_{mid}$ and $\phi_2 : E_{mid} \to E'$ such that $\phi = \phi_2 \circ \phi_1$ and $(E, E', E_{mid}, P, P_{mid} = \phi_1(P), \phi(P))$ is published as the setup.

Participant i selects a random element $s_i \in \mathbb{Z}_N$ as secret key and publishes $S_i = s_i\phi(P)$ as their public key along with a proof of knowledge of the secret key which is a proof of knowledge of discrete log.



Notice that $\hat{\phi}(Q) = \hat{\phi_1}(\hat{\phi_2}(Q))$ so an honest party i will at some point compute $Q_{mid} = \hat{\phi}_2(Q)$. i publishes $Q^i_{mid} = s_iQ_{mid}$ as her proof. The proof can be verified by the pairing equation $e^{mid}_N(P_{mid}, Q^i_{mid}) = e'_N(S_i, Q)$.
If the verifier is satisfied it means that the prover has computed Q_{mid} correctly, which means that she has evaluated at least half of the isogeny walk.

De Feo and Burdges don't give an exact security definition for watermarking. We propose that a watermarking method similar to any other proof system has to be complete and sound.

Definition 41 (Watermarking completeness)**.** A watermarking method is complete if the honest evaluator can always generate a watermarking that convinces the honest verifier.

Definition 42 (Watermarking soundness error)**.** A watermarking method has soundness error ϵ if there is an adversary that, given $\hat{\phi}(Q)$ and a watermarking can generate a new valid watermarking in time $(1 - \epsilon)T$.

Since delay encryption relies on the notion of time, we should take the time required to generate the watermarking into account. Ideally we would like to have a watermarking that is generated in constant time and has negligible soundness error.
Watermarking method proposed by De Feo and Burdges has soundness error $\frac{1}{2}$ since a malicious prover can convince the verifier by only doing $1 - \frac{1}{2}$ of the honest evaluation. If ϕ is splited into n segments and require a proof for each intermediate point, soundness error is reduced to $\frac{1}{n}$. However, this is not efficient as the soundness error is linear in proof length.

4.4 New watermarking method

We present a watermarking method that has constant proof length and zero soundness error. Additionally, both the prover and the verifier have constant complexity in field operations.

1. The setup for the delay encryption doesn't change.

2. Participant i chooses a secret key $s_i \in \mathbb{Z}^*_N$ and publishes her public key $S_i = (vk_i, ek_i) = (s_i\phi(P), s_i^{-1}(rP))$. vk_i is her validation key, used to verify her identity and correctness of computation. ek_i is her extraction key, used to extract the session key from her watermarked evaluation.

3. Each participant also publishes a non-interactive zero-knowledge proof of knowledge for s_i. This proof can be any NIZKPoK of discrete log.

4. When i computes $\hat{\phi}(Q)$ she publishes $Q_i = s_i\hat{\phi}(Q)$ as her watermarked evaluation.

5. To verify the evaluation is done correctly, one checks $e_N(P, Q_i) = e'_N(vk_i, Q)$.
6. The session key is $k = e_N(s_i^{-1} rP, Q_i)$.

Proof. We must show that this watermarking system is complete and sound against a malicious prover who knows a correct proof for an unknown secret key.

Completeness. If the prover is honest $Q_i = s_i\hat{\phi}(Q)$, then we can rewrite the left hand side to get the right hand side: $e_N(P, Q_i) = e_N(P, s_i\hat{\phi}(Q)) = e_N(P, \hat{\phi}(Q))^{s_i} = e'_N(\phi(P), Q)^{s_i} = e'_N(s_i\phi(P), Q) = e'_N(vk_i, Q)$

Soundness. The check in line 5 ensures that the evaluator knows $s_i\hat{\phi}(Q)$ because P, Q and $s_i\phi(P)$ are fixed so the fourth point is determined uniquely. In addition, since the evaluator previously proved the knowledge of s_i (her secret key) she must also know $\hat{\phi}(Q)$.
Assuming discrete log is hard, no malicious party can forge an evaluation without computing $\hat{\phi}(Q)$, because as we established a valid evaluation implies knowledge of $\hat{\phi}(Q)$ so the malicious party has to learn $\hat{\phi}(Q)$ from $s_i\hat{\phi}(Q)$.

□

5 Supersingular elliptic curves with unknown endomorphism ring

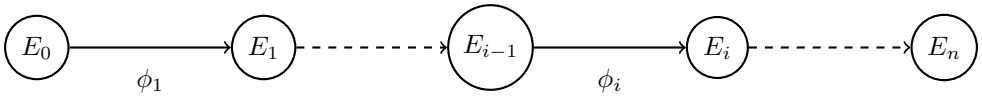
Sampling a random supersingular elliptic curve with unknown endomorphism ring is an open problem. Several attempts has been made and all have failed. This problem is very useful, if a method of generating such a curve is discovered many protocols that require a TTP can lose that requirement.
Currently we know two methods to generate a random supersingular curve.

1. Sampling a random maximal order $\mathcal{O}$ in quaternion algebra $B_{p,\infty}$ and constructing a curve E with that endomorphism ring. To construct E, one can compute a connecting ideal $I = \mathcal{O} \cdot \mathcal{O}_0$, where $\mathcal{O}_0$ is the known endomorphism ring of the curve E_0 with efficient endomorphism ring. Then using the KLPT algorithm a corresponding smooth isogeny $\phi_I : E_0 \to E$ can be computed such that $\mathrm{End}(E) = \mathcal{O}$.
2. Starting from a curve with known endomorphism ring like E_0 and going on a random isogeny walk of sufficient length and using the end curve. Since the supersingular isogeny graph is Ramanujan its mixing time is small, therefore a relatively short walk will result in a uniformly random final curve.

Both of these methods obviously don't hide the endomorphism ring of the final curve however, the latter can be done in a distributed manner so the trust can be distributed.

5.1 Distributed supersingular curve generation

Start from $E_0 : y^2 = x^3 - x$. Participant i verifies all the previous proofs then starts from E_{i-1} and performs a random isogeny walk of length $c \cdot \log(p)$ (c depends on the mixing parameter of the graph and is less than 10) to get an isogeny $\phi_i : E_{i-1} \to E_i$. She publishes E_i and a proof of knowledge of isogeny ϕ_i.



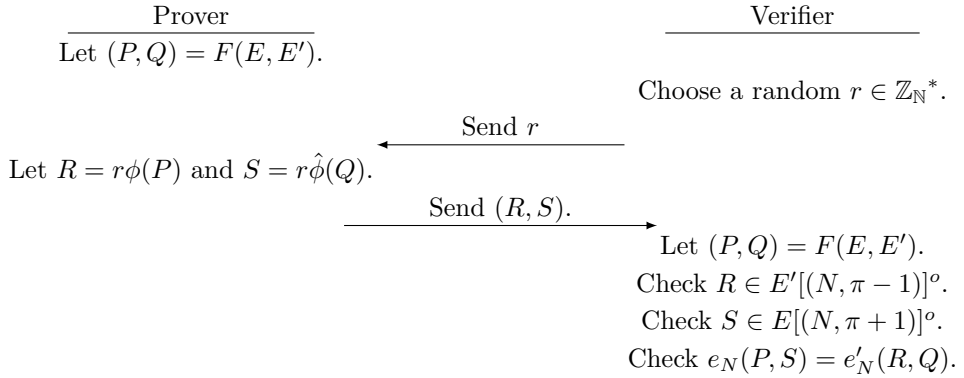
The final curve has unknown endomorphism ring as long as each participant doesn't know at least one other participants walk.

Proof. Assume participant i is honest in the sense that its isogeny walk ϕ_i is sufficiently long and doesn't share it with other participants, then the curve E_i is uniformly random to other participants and therefore computing an isogeny from $E_0 \to E_i$ is hard, even if an isogeny $\psi : E_0 \to E_{i-1}$ is known.
Additionally, there is a participant j such that i doesn't know ϕ_j then i can't compute an isogeny $E_0 \to E_j$. Therefore no participant can compute an isogeny $E_0 \to E_n$. □

The given zero-knowledge proof has to be knowledge-sound even if the prover knows one of the $\mathrm{End}(E_{i-1})$ and $\mathrm{End}(E_i)$, otherwise if the last participant is malicious he can select E_n so that he knows its endomorphism ring $\mathcal{O}_n$. He generally can't find an isogeny $\phi_n : E_{n-1} \to E_n$ however, since the proof is not knowledge-sound he might be able to convince the verifier without knowing such isogeny.

5.2 Proof of isogeny knowledge

The following protocol is proposed in [BDF21] as a proof of knowledge (not zero-knowledge in the current state) of an isogeny $\phi : E \to E'$ over prime field $\mathbb{F}_p$.
Let F be a deterministic function that takes two curves E, E' as input and outputs two points $P \in E[(N, \pi - 1)]^o$, $Q \in E'[(N, \pi + 1)]^o$.



Notice that the verifier is not checking if the prover has used the correct challenge, so a malicious prover can send the same response for any challenge and if the response is valid for any challenge it is valid for all challenges.

5.3 How to cheat?

A malicious prover can cheat by knowing a single discrete logarithm.

1. $(P, Q) = F(E, E')$ is fixed (for fixed E and E').
2. The prover picks $P' \in E'[(N, \pi - 1)]$ and $Q' \in E[(N, \pi + 1)]$. This is done once and P' and Q' are fixed.
3. Let $\alpha = e_N(P, Q')$ and $\beta = e'_N(P', Q)$ and $x = \log_\alpha \beta$ then $\alpha^x = \beta$ and $e_N(P, xQ') = e'_N(P', Q)$.
4. Now if the prover sets $R = P'$ and $S = xQ'$ the verifier will be convinced.

5. Also notice that (aR, aS) is a valid response for any $a \in \mathbb{Z}_N$ so the malicious prover can generate many valid responses from a single discrete log.

Recall that a malicious prover must have negligible advantage of convincing the verifier without knowing an isogeny for **any** pair of elliptic curves. This shows that the knowledge of an isogeny is not strictly needed to convince the verifier so the proof is not a proof of knowledge of isogeny, however the cheating prover has to solve a single discrete log for any (E, E') pair.

More worryingly, even assuming knowledge soundness of this proof, the statement that is proven is not the knowledge of an isogeny $\phi : E \to E'$ but it is **at most** the evaluation of an isogeny on the N-torsion and its degree. Let $\{P, Q\}$ be a basis of $E[N]$ and it is known that there is an isogeny ϕ of degree d with action $\phi(P) = P'$ and $\phi(Q) = Q'$, then $e_N(P, dQ) = e'_N(P', Q')$ holds since $\hat{\phi}(Q') = \hat{\phi}(\phi(Q)) = dQ$.
Also notice that the knowledge of degree is crucial as knowledge of the action alone is trivial, in other words for any $\{R, S\} \in E'[N]$ there is an isogeny taking P to R and Q to S, but this isogeny may not be defined over $\mathbb{F}_p$ or might have a very large degree.
The natural question is "Is this enough to imply knowledge of isogeny?".
For this to be true an extractor must be able to recover some oriented isogeny $\psi : E \to E'$.
From the torsion point attacks on SIDH, we know if enough torsion point images are known an isogeny can be recovered. The best known bound is that knowledge of approximately $\sqrt{\deg \phi}$ torsion points is enough to recover a specific isogeny.
In this case an extractor need not recover a specific isogeny but any smooth isogeny $E \to E'$. Moreover, the $\sqrt{\deg \phi}$ bound is for general isogenies but we are working with $\mathbb{F}_p$-isogenies which are oriented, and therefore fewer and possibly more structured, so an isogeny may be extracted from fewer torsion points.
The isogenies that are supposed to be proven have degree approximately p. In the current setup of the parameters, $p \simeq 2^{\lambda^3}$ and $N \simeq 2^{2\lambda}$ so $N << \sqrt{p}$ and the current known' bounds for recovering an isogeny from torsion points don't apply.

5.4 Sketch of a proof

We propose the following proof sketch to prove the knowledge of an isogeny action on torsion points.

- Prover:
  1. choose random $P \in E[(N, \pi - 1)]$.
  2. choose random $r_1, r_2, r, r' \in \mathbb{Z}_N^*$.
  3. send $(C_P = Com(P, r_1), C_{\phi(P)} = Com(\phi(P), r_2), rP, r'\phi(P))$ along with ZKPoK of r and r'.
- Verifier:
  1. check $rP \in E[(N, \pi - 1)] \wedge r'\phi(P) \in E'[(N, \pi - 1)]$. Also check the proofs of knowledge.
  2. Choose a random bit $b \in \{0, 1\}$ and send it to the prover.
  3. If $b = 0$:
     choose a random $Q \in E'[(N, \pi + 1)]$ and send it to the Prover.
  4. If $b = 1$:
     choose a random $Q \in E[(N, \pi + 1)]$ and send it to the Prover.
- Prover:

1. If $b = 0$ check $Q \in E'[(N, \pi + 1)]$ otherwise check $Q \in E[(N, \pi + 1)]$.
2. If $b = 0$: Send $(P, r_1, r'\hat{\phi}(Q))$.
3. If $b = 1$: Send $(\phi(P), r_2, \frac{r}{d}\phi(Q))$, where $d = \deg \phi$.

- Verifier:

1. If $b = 0$:
   Check $Com(P, r_1) = C_P$ and $e_N(P, r'\hat{\phi}(Q)) = e'_N(r'\phi(P), Q)$.
2. If $b = 1$:
   Check $Com(\phi(P), r_2) = C_{\phi(P)}$ and $e_N(rP, Q) = e'_N(\phi(P), \frac{r}{d}\phi(Q))$.

Completeness. of the presented scheme in case $b = 0$ directly follows the pairing equation and when $b = 1$, $\hat{\phi}(\frac{1}{d}\phi(Q)) = Q$ so $e_N(r'P, Q) = e'_N(\phi(P), \frac{r'}{d}\phi(Q))$.

Knowledge soundness. When $b = 0$, the verification is $e_N(P, r\hat{\phi}(Q)) = e'_N(r\phi(P), Q)$. $P, r\phi(P)$ are fixed and Q is chosen by the verifier so the only response from the prover that satisfies the equation is $r\hat{\phi}(Q)$, since the prover proved knowledge of r it means that he also knows $\hat{\phi}(Q)$. A similar argument works for the case $b = 1$. If the extractor has two interactions with the same first message and different b then it has both P and $\phi(P)$.

Zero-knowledge. In case $b = 0$ the verifier only learns P which is a random point in $E[N, \pi - 1]$ and not $\phi(P)$, also $r\hat{\phi}(Q)$ is random to the verifier since she doesn't know r so a simulator can be built. A similar argument works when $b = 1$.

This protocol uses an unusual trick. In provers first message he sends a commitment $C_p = C(P, r_1)$ of point P and some masked point rP, then he is supposed to send a proof of knowledge of r however, notice that the verifier doesn't know P, instead she knows a commitment of P. This means that proving knowledge of r isn't a simple proof of knowledge of discrete log.
We point out that said statement is an NP statement ((P, r_1, r) is a witness for it) so it does have a proof, but we don't know any proof system that is efficient enough to justify using this scheme.

5.5 Curves we can trust

In this section we look at secure methods that can be used to generate supersingular elliptic curves. Authors in [BCC+22] present a distributed protocol for generating supersingular elliptic curves over $\mathbb{F}_{p^2}$. Their approach follows the same general idea of each participant doing a long enough random walk on the graph, however their main contribution is giving an efficient zero-knowledge proof of knowledge of isogeny that works for any field and walk length. They use SIDH squares glued together to create random walks that are statistically close to random distribution.
Unfortunately, their approach doesn't work for curves defined over prime fields, because the required SIDH squares don't exist. Jao and Mokrani [MJ23] additionally point out that the proof in [BCC+22] reveals the degree of the secret isogeny. For curves over $\mathbb{F}_{p^2}$ this isn't a problem since there are many isogenies of a specific degree if it is large enough. However, Jao and Mokrani claim the knowledge of the degree is too much information when working with isogenies over prime fields.
When the degree of an oriented isogeny ϕ is known the absolute value of exponents e_i of the prime ideals such that $I_\phi = I_{\ell_1}^{e_i} \cdot I_{\ell_2}^{e_2} \cdot \ldots \cdot I_{\ell_n}^{e_n}$ can be computed and this massively reduces the possible isogenies. Concretely, in case of CSIDH-512 there are 74 small primes and when the absolute values of the exponents are known there are at most 2^{74} possible

secret isogenies, which is much smaller than our desired 128-bit security.

This problem can be solved in several ways. One of the simplest ways is to just use more primes in the CSIDH prime. Generally, when n primes are used the expected number of possible ideals with prescribed $|e_i|$ is 2^n. In this solution the size of p and hence all field elements will be larger but not so much that it is prohibitively costly.

5.5.1 CSI-FiSh

Another solution is to use a CSIDH based signature scheme, for example CSI-FiSH [BKV19]. The advantage of this method is that the distributed protocol is still non-interactive because the proof of knowledge is non-interractive, furthermore CSI-FiSh is reasonably fast and has very small proof and key length. When optimized for public key and proof size combined, CSI-FiSh has total size of 1468 bytes and takes 390ms to generate/verify the proof. A draw back of CSI-FiSh is that the class group has to be known.
The authors have computed the class group for CSDIH-512 that has a 154-digit (512 bit) discriminant. The discriminant used in CSIDH-512 is $p = 4 \cdot \ell_1 \cdot \ell_2 \cdot ... \cdot \ell_{74} - 1$, where $\ell_1, \ell_2, ..., \ell_{73}$ are the smallest odd primes and $\ell_{74} = 587$ is the smallest prime such that p is prime. It turns out the fundamental class group is cyclic with size

$$37 \cdot 1407181 \cdot 51593604295295867744293584889 \cdot 3159941450468199585300827874558783220 49$$

$\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ has conductor 2 and since 2 doesn't split in $\mathbb{Q}(\sqrt{-p})$,
$\#\mathrm{cl}(\mathbb{Z}[\sqrt{-p}]) = 3 \cdot \#\mathrm{cl}(\mathcal{O}_{\mathbb{Q}(\sqrt{-p})}) \simeq 2^{257.136}$ and therefore it is also cyclic. Further more $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ is generated by $(3, \pi - 1)$.
For the purpose of isogeny-based VDF and delay-encryption CSI-FiSh is not useful. Recall that the generated random curve is going to be used as the staring curve of a long isogeny walk. If the class group is known then one can compute a short basis, then an ideal with small l_∞ norm in the short basis in the same ideal class as the long isogeny walk can be found by solving a closest vector problem. So the class group should remain unknown.

SeaSign is another signature scheme on CSIDH that doesn't require the class group to be known, however the computation time required is in the order of hundreds of seconds, so it is only practical when there is a lot if time to generate a curve with unknown endomorphism ring and use it for a long time.
Jao and Mokrani [MJ23] propose an interactive proof to prove knowledge of an isogeny in the generation process. Their proof is still based on item 2.12 and uses a random oracle. They also provide a variant without a random oracle but with an extra round of interaction. The amount of communications for each participant is linear in the number of participant for both variations which compared to the signature based protocols is undesirable.

6 Quantum-secure VDF

As mentioned before the isogeny-based VDF on the $\mathbb{F}_p$-restricted graph completely breaks when faced with a quantum adversary. The variant on the general supersingular isogeny graph is quantum-annoying but not quantum-secure. Furthermore, the construction based on squaring in groups of unknown order discussed in Appendix A, also don't resist a quantum adversary.
The first quantum secure VDF construction was proposed in [CSHT21]. Their sequential function is computing the j-invariant of a long isogeny walk on the supersingular isogeny graph similar to the CGL hash function. They adapt a general purpose SNARG for this computation to prove its correctness. Their VDF has prover time $\widetilde{O}(T)$ with $O(\log^4(T))$

parallelization and $\widetilde{O}(\log(T)^{4+\log\log T})$ verifier time with no parallelization.
The input of the VDF is given to a pseudo-random generator (PRG) and the output of the PRG is used to specify the isogeny walk "on the fly".

Two elliptic curves E and E' with j-invariants j and j' respectively, are ℓ-isogenous if and only if the modular polynomial $\Phi_\ell(j, j')$ vanishes. Therefore, for a fixed curve E with $j(E) = j$ its neighbours on the ℓ-isogeny graph are given by the roots of $\Phi_\ell(j, x)$, where $\Phi_\ell(j, x)$ is a univariate polynomial if degree ℓ.
When walking in the 2-isogeny graph let E_0 be the starting curve and E_i be the current vertex and $j_i = j(E_i)$. To chose the next vertex we look at the roots of $\Phi_2(j_i, x)$, there are 3 roots as the graph is 3-regular, bot notice that on of the roots corresponds to the previous curve E_{i-1} so $\Phi_2(j_i, x) = (x - j_{i-1})(x^2 + ax + b)$.
The other two roots are given by

$$j_{i+1} = \frac{1}{2}(j_i^2 - 1488j_i - j_{i-1} + 162000 \pm \sqrt{D_i}) \tag{4}$$

where

$$\begin{aligned} D_i \quad &= j_i^4 - 2976j_i^3 + 2j_i^2 j_{i-1} + 2532192j_i^2 - 2976j_i j_{i-1} \\ &\quad -645205500j_i - 3j_{i-1}^2 + 324000j_{i-1} - 8748000000 \end{aligned} \tag{5}$$

A canonical square root of D_i can be deterministically computed using Kong's method.
Since Equation 4 requires the j-invariant of two steps into the past for the initial state we have to set two vertices. The supersingular isogeny graph has no repeated edges or loops except at $j = 1728$ and $j = 0$. $j = 1728$ has on loop and two edges to $j = 287496$, so we set $j_{-1} = 1728$ and $j_0 = 287496$ to avoid going to the non-regular vertex. Afterwards, at each step the next isogeny is specified by the sign of $\sqrt{D_i}$ which is determined by the output of the PRG.

To prove the the correctness of the computation, the evaluator keeps track of (j_i, D_i) at each step as its computation history. Then each sequence j and D is turned into an instance of the sumcheck problem and a probabilistic checkable proof (PCP) is created for each. Then standard methods are used to transform a PCP into a SNARG. For the details of arithmatization of the sequences see 4.1 in [CSHT21]. The transformation from PCP to SNARG is briefly explained in appendix B in [CSHT21] but a more detailed discussion can be found in Micali's original paper [Mic00].

An advantage of this construction is that no trusted setup is required since the isogeny walk is changed every time and the isogeny is not represented in an efficient form (i.e. as a kernel or a quaternionic ideal) no one can compute the final j-invariant in time less than T even is the endomorphism ring of the starting curve is known. Additionaly, compared to the VDF proposed by De Feo et al. the setup algorithm doesn't require a sequential function evaluation and is independent of T.
This being said, SNARG-based VDFs are only interesting in theory and are impractical in practice because the computational overhead caused by arithmatization and the PCP to SNARG transformation is in the order of tens of thousands. Even though, authors in [CSHT21] give specialized constructions for isogeny evaluation to reduce the computational overhead, but the improvements are still not enough to make this construction viable in practice. Unfortunately, they don't provide performance evaluations in their paper.

7 Existence and relation of time-release primitives

In this section we look at the relation between time-release cryptographic primitives and the minimal assumptions to create them.

Looking at classical cryptographic primitives we can see some relations. The existence of symmetric key encryption implies existence of one-way functions, on the the other hand existence of a one-way function implies existence of pseudo-random functions and in turn it implies existence of symmetric key encryption. Therefore symmetric key encryption and one-way functions are in some sense equivalent.
Moreover, Levin [Lev03] gave an explicit instance of a universal one-way function, meaning that Levin's function is one-way if and only if one-way functions exist. This result is theoretically very significant, because combined with previous results on symmetric key encryption and one-way functions, we can construct a universally secure symmetric key encryption scheme.

Furthermore, existence of public key encryption implies existence of trap-door one-way permutation that in turn implies $P \neq NP$.
Conversely, existence of a injective one-way permutation implies existence of public key encryption. Therefore these two are also in some sense equivalent. This also tells us that $P \neq NP$ is a necessary assumption for having public key encryption.

We are interested in understanding what kind of relations exist between time-release primitives and what are the necessary assumptions for having time-release cryptography.

7.1 $NC \neq P$

NC is the complexity class of problems decidable in polylogarithmic time and polynomial parallelization in other words, circuits with polylog depth and polynomial width. It is easy to see $NC \subseteq P$. However, if a sequential function exists $NC \subsetneq P$.
So $NC \neq P$ is a necessary condition for existence of time-release primitives, since any form of delay function is a sequential function.
Interestingly, existence of a sequential function implies existence of VDFs.

7.2 VDF from sequential functions

Let f be a sequential function. Its evaluation is a polynomial-bounded computation, therefore it has a succinct non-interactive zero-knowledge proof of correctness. This function along with its proof is a VDF, and because the proof is zero-knowledge, if the proof is watermarked no one can generate another valid proof with a different watermarking. This construction shows that $NC \neq P$ implies existence of VDF.

7.3 TLP from trapdoor sequential functions

Let $f : X \to Y$ be a trapdoor sequential function with trapdoor τ. This means if τ is unknown f is a normal VDF, however someone knowing τ can compute $f(x)$ for any $x \in X$ quickly.
We can define the following TLP. The Setup algorithm handles specifying f and giving the trapdoor τ to the sender. The Gen algorithm chooses a $x \in X$ at random and computes $y = f(x)$ using τ, then y can be used as the key to a symmetric key encryption scheme to encrypt the secret s and get cipher z, then the puzzle is (x, z).
The Sol algorithm computes $y = f(x)$ but without the knowledge of the trapdoor τ this computation is slow. Once it has y, it can decrypt z to get the secret s.

7.4 Future work

An interesting question for future work on time-release cryptography is to further explore the relation between primitives. For example one can easily create a VDF from a delay

encryption scheme, but it is unclear whether the converse is possible.
Finding a universal VDF is very interesting but maybe far from reach as it reduces to finding a NC-hard problem.

References

[Bac89] Eric Bach. Number theoretic algorithms. 1989. URL: https://minds.wisconsin.edu/bitstream/handle/1793/59118/TR844.pdf?sequence=1.

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. pages 757–788, 2018. URL: https://eprint.iacr.org/2018/601.pdf.

[BCC+22] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. Cryptology ePrint Archive, Paper 2022/1469, 2022. https://eprint.iacr.org/2022/1469. URL: https://eprint.iacr.org/2022/1469.

[BDF21] Jeffrey Burdges and Luca De Feo. Delay encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 302–326. Springer, 2021. URL: https://eprint.iacr.org/2020/638.pdf.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. Csi-fish: Efficient isogeny based signatures through class group computations. Cryptology ePrint Archive, Paper 2019/498, 2019. https://eprint.iacr.org/2019/498. URL: https://eprint.iacr.org/2019/498.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24*, pages 395–427. Springer, 2018.

[CSHT21] Jorge Chavez-Saab, Francisco Rodríguez Henríquez, and Mehdi Tibouchi. Verifiable isogeny walks: Towards an isogeny-based postquantum vdf. Cryptology ePrint Archive, Paper 2021/1289, 2021. https://eprint.iacr.org/2021/1289. URL: https://eprint.iacr.org/2021/1289.

[DFMPS19] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. pages 248–277, 2019. URL: https://eprint.iacr.org/2019/166.pdf.

[DG16] Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over f _p f p. *Designs, Codes and Cryptography*, 78:425–440, 2016. URL: https://arxiv.org/pdf/1310.7789.

[Kla12] Janis Klaise. Orders in quadratic imaginary fields of small class number. *preprint*, 2012. URL: https://warwick.ac.uk/fac/cross_fac/complexity/people/students/dtc/students2013/klaise/janis_klaise_ug_report.pdf.

[Lev03] Leonid A Levin. The tale of one-way functions. *Problems of Information Transmission*, 39(1):92–103, 2003. URL: https://arxiv.org/abs/cs/0012023.

[LMOQ22] Angelique Faye Loe, Liam Medley, Christian O'Connell, and Elizabeth A Quaglia. Tide: a novel approach to constructing timed-release encryption. pages 244–264, 2022. URL: https://eprint.iacr.org/2021/1293.pdf.

[Med23] Liam Medley. A good use of time: Techniques and applications of delay-based cryptography. 2023.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.

[MJ23] Youcef Mokrani and David Jao. Generating supersingular elliptic curves over $\mathbb{F}_p$ with unknown endomorphism ring. Cryptology ePrint Archive, Paper 2023/984, 2023. https://eprint.iacr.org/2023/984. URL: https://eprint.iacr.org/2023/984, doi:10.1007/978-3-031-56232-7_8.

[Pan] Lorenz Panny. *Cryptography on Isogeny Graphs*. PhD thesis. URL: https://yx7.cc/docs/phd/thesis.pdf.

[Pie19] Krzysztof Pietrzak. Simple verifiable delay functions. 2019. URL: https://drops.dagstuhl.de/storage/00lipics/lipics-vol124-itcs2019/LIPIcs.ITCS.2019.60/LIPIcs.ITCS.2019.60.pdf.

[RSW96] Ronald L Rivest, Adi Shamir, and David A Wagner. Time-lock puzzles and timed-release crypto. 1996. URL: https://people.csail.mit.edu/rivest/pubs/RSW96.pdf.

[Sil09] Joseph H. Silverman. *The Arithmetic of Elliptic Curves 2nd Edition*. 2009.

[Sto12] Anton Stolbunov. *Cryptographic Schemes Based on Isogenies*. PhD thesis, 01 2012. doi:10.13140/RG.2.2.20826.44488.

[Wes20] Benjamin Wesolowski. Efficient verifiable delay functions. *Journal of Cryptology*, 33:2113–2147, 2020. URL: https://hal.science/hal-02945371/document.

A Delay from squaring

In this section we discuss squaring in groups of unknown order as a candidate for sequential computation that can be used to create instantiate VDFs or delay encryption.

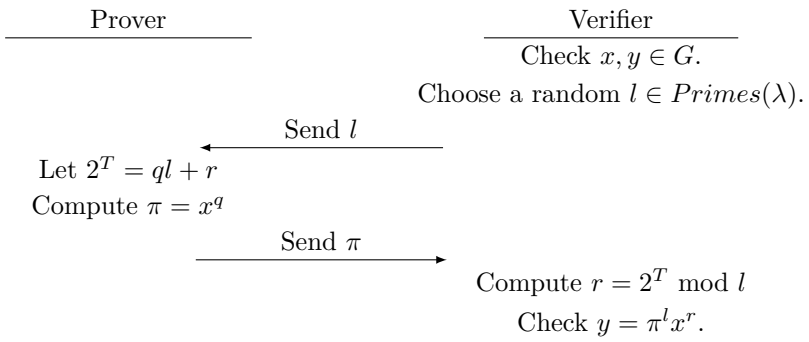
A.1 Squaring in groups of unknown order

Let G be group of unknown order and $f_G(x,T) = x^{2^T}$. Assuming that computing x^{2^T} requires T sequential squarings we can use this to create a delay function. Notice that this is a trap-door delay function since someone who knows $n = |G|$ can compute $k = 2^T \bmod n$ in time $\log(T)$ then compute $f_G(x,T) = x^k$ in time $\log(k) \simeq \log(n)$.
The setup algorithm generates a group of unknown order G and Evaluation algorithm is computing $y = f_N(x,T)$ and a proof π that $y = x^{2^T}$.
The evaluator has to publish a proof because there is no efficient way (indipendant of T) to verify the computation is done correctly without the help of the evaluator. There are two proposed methods for this proof by Pietrzak [Pie19] and Wesolowski [Wes20].

- **Pietrzak succinct proof**
  Let $u = x^{2^{T/2}}$, then $y = u^{2^{T/2}}$. The prover computes u and sends it to the verifier and recursively proves that he computed u and y correctly. Notice that the exponent

of the statement that is to be proved is halfed each time so the the depth of recursion is logarithmic in T and when the exponent is constant the verifier can do the computation himself and verify the prover's computation.
Furthermore the exponent of the statements to be proved are equal so we can consider a random linear combination of them $u^r y = (x^r u)^{2^{T/2}}$ for some random $r \in \{1, ..., 2^\lambda\}$, so in each recursive step the number of statements to be proved doesn't change (is one).
Verifier time. The verifier computation time is dominated by two exponentiations in G which takes 2λ so the total time is $2\lambda \log(T)$.
Prover time. The prover has to compute $x^{2^{T/2}}$ at each level of recursion, with some prepossessing he can compute the output of the VDF and its proof in $(1 + \frac{2}{\sqrt{T}})T$.
Proof length. The proof generated using this method consists of $\log(T)$ group elements.
Soundness. The soundness of this proof is based on the **low order assumption**. The low order assumption claims that any adversary that finds a non-trivial element $g \in G$ and $1 < d < 2^\lambda$ such that $g^d = 1$, has negligible advantage.

- **Wesolowski succinct proof**



Verifier time. The verifier has to compute $r = 2^T \bmod l$ which takes $\log(T)$ multiplications in Z_l^+, beyond that he performs two exponentiations in G that take at most λ so in total about $\log(T) + \lambda$.
Prover time. The prover has to compute $\pi = x^q$. He can compute the output of the VDF and the proof in $(1 + \frac{1}{s})T$ using s processors in parallel. See [Wes20] for details.
Proof length. The proof created using this method consists of a single group element.
Soundness. Soundness of this proof relies on the **adaptive root assumption**. Adaptive root assumption roughly claims that no adversary can find an element $w \in G - \{1\}$ such that for a random prime $l \in Primes(\lambda)$ he can compute an l-th root for w with non-negligible probability.

Remark. In terms of assumptions, the adaptive root assumption implies the low order assumptions whereas the reverse direction is not known to be true.

A.2 RSA groups

We know that knowledge of the order of the group $\mathbb{Z}_N^*$ is equivalent to knowing the factorization of N, so we can use the RSA groups as groups of unknown order. In this case the delay function would be $f_N(x, T) = x^{2^T} mod N$.
In fact this was the original candidate for creating time-release cryptography by Rivest,

Shamir and Wagner [RSW96]. However generating RSA moduli in a distributed setup is inefficient and scales badly, furthermore some proposed optimization have led to security problems.

A.3 Ideal class group of imaginary quadratic fields

Another candidate for groups of unknown order are the ideal class group of quadratic imaginary fields $\mathbb{Q}(\sqrt{-p})$ where p is a prime. This group is the quotient group of the fractional ideals of the integer ring $\mathcal{J}_{\mathcal{O}_R}$ over the group of principal ideals of the integer ring $\mathcal{P}_{\mathcal{O}_R}$.
Using this group doesn't give us security against a quantum-adversary however, generating this group in a distributed setup is more efficient.

A.4 Delay encryption from squaring in RSA groups

In this section we review a delay encryption method named TIDE [LMOQ22]. In this method an RSA modulus $N = pq$ is generated and used as the public key, and some information is provided so after someone has evaluated the delay function $f_N(x, T)$ everyone can factorize N and decrypt ciphertexts.
We will use that fact that if x and x' are know such that $x \neq \pm x'$ and $x^2 = x'^2$ then $gcd(x - x', N)$ is a non-trivial factor of N.
The public parameters are $pp = (N, x, x_0, x_{-t})$, where N is a blum integer [2] $\left(\frac{x}{N}\right) = -1$ and $x_0 = x^2$ and $x_{-t}^{2^T} = x_0$.
Notice that $x' = x_{-t}^{2^{T-1}}$ is a square root of x_0 however, it is by definition a quadratic residue therefore $\left(\frac{x'}{N}\right) = 1$ and since N is a Blum integer $\left(\frac{-1}{N}\right) = 1$ so $\left(\frac{-x'}{N}\right) = 1$ now remember $\left(\frac{x}{N}\right) = -1$ so x and x' are both square roots of x_0 and $x \neq \pm x'$, so they can be used to factorize N.
Now we describe how to generate the public parameters.

1. Generating N is simple, just sample enough random primes until we have two primes p and q congruent to 3 mod 4.

2. To generate x, we sample random $x \in Z_n^*$ until $\left(\frac{x}{N}\right) = -1$. Notice that this happens with probability $\frac{1}{2}$. Then set $x_0 = x$.

3. The non-trivial part is generating x_{-t}. x_{-t} is a T-th square root of x_0 mod N. Using the Chinese reminder theorem we can generate x_{-t} from α_t and β_t, where α_t is a T-th square root of x_0 mod p and β_t is a T-th square root of x_0 mod q.
   Furthermore time complexity of computing α_t and β_t is logarithmic in T: $\alpha_t = x_0^{\frac{p+1}{4}^T}$ mod p and $\alpha_t = x_0^{\frac{q+1}{4}^T}$ mod q. Since the orders of groups Z_p^* and Z_q^* are known we can compute α_t and β_t in time $\log(T) + \log(p) + \log(q)$.

Evaluation is computing $x' = f_N(x_{-t}, T - 1) = x_{-t}^{2^{T-1}}$. Optionally a proof of computation of x' can also be published.
After x' is computed anyone can factorize N and decrypt any ciphertext.

[2] $N = pq$, where p and q are prime and $p = q = 3$ mod 4