SIGNITC: Supersingular Isogeny Graph Non-Interactive Timed Commitments

Knud Ahrens
University of Passau, Germany
knud.ahrens@uni-passau.de

Abstract

Non-Interactive Timed Commitment schemes (NITC) allow to open any commitment after a specified delay t_{fd}. This is useful for sealed bid auctions and as primitive for more complex protocols. We present the first NITC without repeated squaring or theoretical black box algorithms like NIZK proofs or one-way functions. It has fast verification, almost arbitrary delay and satisfies IND-CCA hiding and perfect binding. Our protocol is based on isogenies between supersingular elliptic curves making it presumably quantum secure, and all algorithms have been implemented as part of SQISign or other well-known isogeny-based cryptosystems. Additionally, it needs no trusted setup and can use known primes for SIKE or SQISign.

Keywords: Non-interactive timed commitments, post-quantum, isogeny walks, Deuring correspondence.

1 Introduction

The concept of time-lock puzzles [21] has been around for more than twenty years, but timed commitments [5] are rather new and we will use the definition of Non-Interactive Timed Commitment schemes (NITC) by Katz, Loss, and Xu [19] from the year 2020. These protocols satisfy binding or non-malleability properties and efficient verification just like usual commitment schemes, but a commitment can be opened by anyone after some delay t_{fd}. So hiding only lasts for this time t_{fd} and there are additional algorithms: one to verify that a commitment can be opened by others and another one to open the commitment forcefully in time at least t_{fd}. A possible application is a sealed bid auction, where all bids can be revealed after time t_{fd} even if some of the bidders refuse to open their commitment. Other applications are listed in Katz et al. [19].

Our approach uses random walks in the isogeny graph of supersingular elliptic curves to construct a NITC, hence the name Supersingular Isogeny Graph Non-Interactive Timed Commitments or SIGNITC[1] for short. The main idea is that computing isogenies of large or non-smooth degree is slow, but if we know the endomorphism ring of the starting curve, we can find a smooth shortcut. So we use a secret isogeny to a curve with known endomorphism ring for fast

[1] pronounced like “signets”

commitment and verification, but the forced decommitment has to compute the long isogeny and thus it needs time at least t_{fd}.

The advantage of isogeny-based cryptography is that it is presumably quantum secure and relatively slow compared to other fields of post-quantum cryptography. Since we need a delay, this is a good thing. The field has undergone thorough scrutiny due to the candidates SIKE [18] and SQISign [9] in NIST competitions for post-quantum protocols and it is well-studied by now. The protocol only uses (known) isogeny-based cryptography, so we do not need to know several fields and this facilitates correct and secure implementations. This also means that we have no theoretical black box algorithms like zero knowledge proofs, succinct non-interactive arguments of knowledge or one-way functions. In addition, all needed calculations have already been implemented as subroutines in other cryptosystems. To our knowledge this is the first quantum secure NITC scheme with explicit algorithms. The only drawbacks are that some algorithms are still quite involved and that we need to differ slightly from the original definition for hiding.

Related Work Thyagarajan et al. [23] present an approach based on class groups using non-interactive zero knowledge (NIZK) proofs. Katz et al. [19] and Chvojka and Jager [10] use protocols based on repeated squaring in a group of unknown order and NIZK proofs. Finally Ambrona et al. [2] avoid NIZK proofs but still use repeated squaring. None of these is quantum secure.

NITC schemes are related to verifiable delay functions (VDF) [6] in the sense that both have fast verification and a function that needs a long time to evaluate. The main difference is the handling of secrets. For VDFs finding the correct response for a given challenge has to be slow for everyone. For NITC schemes however someone has to compute the commitment and therefore already knows the output of the slow task, namely finding the message to a given commitment. So we can construct NITC schemes from VDFs, but the contrary is difficult or impossible, depending on the protocol.

VDFs have direct applications to blockchains and there already are several approaches. Many are based on repeated squaring for the delay. A new publication [4] suggests that this might not be sequential. So contrary to current belief, repeated squaring could be parallelizable, disqualifying it as a delay function. Additionally, this is not quantum secure. There are even some isogeny-based candidates for VDFs, but they all still have some flaws. The pairing-based approach [11] is not quantum secure. Chavez-Saab et al. [8] use SNARGS and their verification time increases for larger delays. Finally there is one base on Kani's criterion for abelian surfaces [13], but the authors state that it is not clear how to implement it. A different approach based on endomorphism rings [1] has the problem that the generation of a challenge also gives (a significant advantage in finding) the response. So it is closer to a NITC scheme and gave the initial idea for this article.

Structure of this Article The remainder of this paper is structured as follows. First we give a definition of NITC schemes and discuss their properties. Next we recall the necessary definitions and fix the notations of isogeny-based cryptography. Readers familiar with one of these topics can briefly skim through the respective sections as we aimed to use standard notations. The sole difference is a slight variation in Definition 3.5 of an IND-CCA security game. In Section 4 we present our protocol in full detail. Its security and its properties are discussed in Section 5. Finally we give a short conclusion and outlook.

2 Non-Interactive Timed Commitments

In this section we recall NITC schemes and their properties. In their paper Katz et al. [19] gave the first formal definition of this concept.

Definition 2.1 (NITC [19])**.** *A* $(t_{\mathrm{com}}, t_{\mathrm{cv}}, t_{\mathrm{dv}}, t_{\mathrm{fd}})$- non-interactive timed commitment scheme (NITC) *is a tuple* $\mathtt{TC} = (\mathtt{PGen}, \mathtt{Com}, \mathtt{ComVrfy}, \mathtt{DecVrfy}, \mathtt{FDecom})$ *of five algorithms with the following behaviour:*

- *The randomized parameter generation algorithm* $\mathtt{PGen}$ *takes as input the security parameter* 1^κ *and outputs a common reference string* $\mathbf{crs}$.
- *The randomized commit algorithm* $\mathtt{Com}$ *takes as input a string* $\mathbf{crs}$ *and a message* m*. It outputs a commitment* $\mathbf{C}$ *and proofs* $\pi_{\mathrm{com}}, \pi_{\mathrm{dec}}$ *in time at most* t_{com}.
- *The deterministic commitment verification algorithm* $\mathtt{ComVrfy}$ *takes as input a string* $\mathbf{crs}$*, a commitment* $\mathbf{C}$ *and a proof* π_{com}*. It outputs* **accept** *(if* $\mathbf{C}$ *could be forcefully decommitted) or* **reject** *in time at most* t_{cv}.
- *The deterministic decommitment verification algorithm* $\mathtt{DecVrfy}$ *takes as input a string* $\mathbf{crs}$*, a commitment* $\mathbf{C}$*, a message* m *and a proof* π_{dec}*. It outputs* **accept** *or* **reject** *in time at most* t_{dv}.
- *The deterministic forced decommitment algorithm* $\mathtt{FDecom}$ *takes as input a string* $\mathbf{crs}$ *and a commitment* $\mathbf{C}$*. It outputs a message* m *or* **invalid** *in time* at least t_{fd}.

We require that for all κ*, all* $\mathbf{crs}$ *output by* $\mathtt{PGen}(1^\kappa)$*, all* m *and all* $\mathbf{C}, \pi_{\mathrm{com}}, \pi_{\mathrm{dec}}$ *output by* $\mathtt{Com}(\mathbf{crs}, m)$*, it holds that*

$$\mathtt{ComVrfy}(\mathbf{crs}, \mathbf{C}, \pi_{\mathrm{com}}) = \mathbf{accept} = \mathtt{DecVrfy}(\mathbf{crs}, \mathbf{C}, m, \pi_{\mathrm{dec}})$$

and $\mathtt{FDecom}(\mathbf{crs}, \mathbf{C}) = m$.

To be relevant for applications a NITC also needs to satisfy three further properties. First we give a proper definition of *practicality* and then recall definitions for *hiding* and *binding* in our notation.

Definition 2.2 (Practicality)**.** *A NITC scheme is* practical*, if verification is much faster than forcefully opening the commitment, so* $t_{\mathrm{cv}}, t_{\mathrm{dv}} \ll t_{\mathrm{fd}}$*. If in addition the commitment is also much faster than forced decommitment, i.e.* $t_{\mathrm{com}} \ll t_{\mathrm{fd}}$*, we call it* perfectly practical.

We present two IND-CCA security games and define hiding in terms of the probability that an adversary $\mathcal{A}$ wins the games. In both cases the adversary has access to an oracle for $\mathtt{FDecom}$ and a query is considered to have only a small computational cost. The first game is the one used by Katz et al. [19].

Definition 2.3 (IND-CCA original [19]). *For a NITC scheme* $\mathtt{TC}$ *and an algorithm* $\mathcal{A}$*, define the game* $\mathtt{IND\text{-}CCA}^{\mathcal{A}}_{\mathtt{TC}}$ *as follows:*

1. *Compute* $\mathbf{crs} \leftarrow \mathtt{PGen}(1^\kappa)$.
2. *Run* $\mathcal{A}(\mathbf{crs})$ *in a preprocessing phase with access to* $\mathtt{FDecom}(\mathbf{crs}, \cdot)$.
3. *When* $\mathcal{A}$ *outputs* (m_0, m_1)*, choose a uniform bit* $b \leftarrow \{0, 1\}$ *and then compute* $(\mathbf{C}_b, \pi_{\text{com}}, \pi_{\text{dec}}) \leftarrow \mathtt{Com}(\mathbf{crs}, m_b)$. *Give* $(\mathbf{C}_b, \pi_{\text{com}})$ *to* $\mathcal{A}$*, who continues to have access to* $\mathtt{FDecom}(\mathbf{crs}, \cdot)$ *except that it may not query the oracle on the given commitment* $\mathbf{C}_b$.
4. *When* $\mathcal{A}$ *outputs a bit* b'*, it wins iff* $b' = b$.

The commitment $\mathbf{C}$ in our approach is a tuple $\mathbf{C} = (E_s, K_T, u)$ and not a single value. Because of that we can only satisfy a slightly weaker variation of the IND-CCA security game. The new Definition 3.5 is given in Section 3.2 and is discussed in more detail in Section 5.1. Hiding is defined with respect to an IND-CCA game. This allows us to evaluate the security of our NITC in terms of both the original and our adapted definition. Broadly speaking hiding guarantees that it is impossible to infer information about the message from the commitment. In our case hiding should hold at least for the time t_{fd} it takes to open a commitment by force, so for all $t_o < t_{\text{fd}}$ in the following definition.

Definition 2.4 (Hiding [19]). *A NITC scheme* $\mathtt{TC}$ *is* (t_p, t_o, ε)-CCA-secure *if for all adversaries* $\mathcal{A}$ *running in time at most* t_p *in the preprocessing phase and time at most* t_o *in the subsequent online phase,*

$$\Pr\left[\mathcal{A} \text{ wins } \mathtt{IND\text{-}CCA}^{\mathcal{A}}_{\mathtt{TC}}\right] \leq \frac{1}{2} + \varepsilon.$$

Similar to hiding, binding is defined in terms of the probability that $\mathcal{A}$ wins a BND-CCA security game. This time we do not need to adapt this for our approach.

Definition 2.5 (BND-CCA [19]). *For a NITC scheme* $\mathtt{TC}$ *and an algorithm* $\mathcal{A}$*, define the game* $\mathtt{BND\text{-}CCA}^{\mathcal{A}}_{\mathtt{TC}}$ *as follows:*

1. *Compute* $\mathbf{crs} \leftarrow \mathtt{PGen}(1^\kappa)$.
2. *Run* $\mathcal{A}(\mathbf{crs})$ *with access to* $\mathtt{FDecom}(\mathbf{crs}, \cdot)$.
3. $\mathcal{A}$ *outputs* $(m, \mathbf{C}, \pi_{\text{com}}, \pi_{\text{dec}}, m', \pi'_{\text{dec}})$ *and wins iff* $\mathtt{ComVrfy}(\mathbf{crs}, \mathbf{C}, \pi_{\text{com}}) =$ **accept** *and either:*
   - $m \neq m'$*, yet* $\mathtt{DecVrfy}(\mathbf{crs}, \mathbf{C}, m, \pi_{\text{dec}})$ *and* $\mathtt{DecVrfy}(\mathbf{crs}, \mathbf{C}, m', \pi'_{\text{dec}})$ *both output* **accept***;*
   - $\mathtt{DecVrfy}(\mathbf{crs}, \mathbf{C}, m, \pi_{\text{dec}}) =$ **accept** *but* $\mathtt{FDecom}(\mathbf{crs}, \mathbf{C}) \neq m$.

Binding makes sure that a commitment can not be opened to two different messages and that FDecom gives the correct messages for valid commitments.

Definition 2.6 (Binding [19]). *A NITC scheme* $\mathtt{TC}$ *is* (t, ε)-BND-CCA-secure *if for all adversaries* $\mathcal{A}$ *running in time* t*,*

$$\Pr\left[\mathcal{A} \text{ wins } \mathtt{BND\text{-}CCA}^{\mathcal{A}}_{\mathtt{TC}}\right] \leq \varepsilon.$$

3 Isogeny-based Cryptography

In this section we provide the necessary basics for isogeny-based cryptography, quaternion algebras and the Deuring correspondence. We also discuss some computational problems in this area.

3.1 Elliptic Curves and the Quaternion Algebra

Elliptic curves have ties to different fields resulting in several equivalent definitions. We will mostly follow the notation of Silverman [22], but restrict ourselves to aspects relevant for this paper.

Definition 3.1 (Elliptic Curve)**.** *An* elliptic curve *is a pair* (E, ∞)*, where* E *is a curve of genus one and* $\infty \in E$*. It is* defined over a field K*, if it is defined over* K *as a curve and* $\infty \in E(K)$*.*

We can define an addition of points on the curve making $(E, +)$ an additive group where ∞ is the neutral element. This permits scalar multiplication written as $[m]: E \to E$ and torsion subgroups $E[m] := \{P \in E \mid [m]P = \infty\}$.

Definition 3.2 (Isogeny)**.** *Let* E *and* E' *be elliptic curves. Then a morphism* $\varphi: E \to E'$ *such that* $\varphi(\infty) = \infty$ *is called an* isogeny*. If a non-zero isogeny* $\varphi: E \to E'$ *exists, then* E *and* E' *are called* isogenous*.*

In fact, every isogeny between two curves is also a group homomorphism. The isogenies from a curve E into itself form the endomorphism ring $\operatorname{End} E$. Isogenies can be written as rational maps and their degree is defined by this map. Thus, the degree $\deg(\varphi \circ \varphi') = \deg \varphi \deg \varphi'$ is multiplicative. In addition, each isogeny $\varphi: E \to E'$ has a unique dual isogeny $\hat{\varphi}: E' \to E$ such that the composition $\hat{\varphi} \circ \varphi = [\deg \varphi]$ is the multiplication by the degree. The isogenies of degree 1 are the isomorphisms, and each isomorphism class can be labelled by the so-called j-invariant. This allows to construct the ℓ-isogeny graph that has those j-invariants as vertices and isogenies of degree ℓ as edges.

Definition 3.3 (Supersingularity)**.** *Let* K *be a field of characteristic* $p > 0$ *and* E *an elliptic curve defined over* K*. The curve* E *is* supersingular *if the torsion group* $E[p]$ *is trivial. Equivalently, this means that the endomorphism ring* $\operatorname{End} E$ *is an order in a quaternion algebra.*

For the rest of this paper $p > 3$ will be a large prime. This allows us to write every elliptic curve in short Weierstraß form as $E: y^2 = x^3 + Ax + B$ with $j(E) = 108(4A)^3/(4A^3 + 27B^2)$. For supersingular curves there is always a representation with A, B, j in $\mathbb{F}_{p^2}$. There are only $\lfloor p/12 \rfloor + \varepsilon$ supersingular elliptic curves for fields with characteristic p where $\varepsilon \in \{0, 1, 2\}$. Hence, the subset $J_{SS} \subset \mathbb{F}_{p^2}$ of supersingular j-invariants has cardinality at least $\lfloor p/12 \rfloor$.

We have already seen in Definition 3.3 that supersingular curves are related to quaternion algebras. We are interested in the quaternion algebra $\mathcal{B}_{p,\infty}$ ramified at p and infinity with $\mathbb{Q}$-basis $\{1, \mathrm{i}, \mathrm{j}, \mathrm{k}\}$ such that

$$\mathrm{i}^2 = -1, \quad \mathrm{j}^2 = -p, \quad \mathrm{k} = \mathrm{ij} = -\mathrm{ji}.$$

The (reduced) norm of an element $\mathfrak{a} = a_0 + a_1\mathrm{i} + a_2\mathrm{j} + a_3\mathrm{k} \in \mathcal{B}_{p,\infty}$ is given by $\operatorname{nrd}(\mathfrak{a}) = \mathfrak{a}\bar{\mathfrak{a}}$ for $\bar{\mathfrak{a}} = a_0 - a_1\mathrm{i} - a_2\mathrm{j} - a_3\mathrm{k}$. The bilinear form $f(\mathfrak{a}, \mathfrak{b}) =$

$(\mathfrak{a}\bar{\mathfrak{b}} + \mathfrak{b}\bar{\mathfrak{a}})/2$ satisfies $f(\mathfrak{a}, \mathfrak{a}) = \mathfrak{a}\bar{\mathfrak{a}} = \mathrm{nrd}(\mathfrak{a})$, and two elements $\mathfrak{a}, \mathfrak{b} \in \mathcal{B}_{p,\infty}$ are called orthogonal if $f(\mathfrak{a}, \mathfrak{b}) = 0$. An order in $\mathcal{B}_{p,\infty}$ is a lattice that is also a subring, and it is maximal if its discriminant equals p. Now an elliptic curve E is supersingular if and only if $\operatorname{End} E$ is isomorphic to a maximal order $\mathcal{O}$ in $\mathcal{B}_{p,\infty}$.

Theorem 3.4 (Deuring Correspondence [14]). *The isomorphism classes of supersingular elliptic curves correspond to the isomorphism classes of invertible left $\mathcal{O}$-ideals in the quaternion algebra, for a fixed maximal order $\mathcal{O}$.*

This so-called Deuring correspondence also gives us that an ℓ-isogeny φ starting at E corresponds to a left ideal I_φ of norm ℓ in $\mathcal{O} \cong \operatorname{End} E$, and the image curve has an endomorphism ring isomorphic to the right order $\mathcal{O}_R(I_\varphi) = \{\mathfrak{a} \in \mathcal{B}_{p,\infty} \mid I_\varphi \mathfrak{a} \subseteq I_\varphi\}$ of I_φ, see [24, Ch. 42] for more details.

3.2 Application to Cryptography

Many isogeny-based protocols rely on secret walks in isogeny graphs of supersingular elliptic curves. The fact that the endomorphism ring is non-commutative gives rise to presumably quantum secure protocols. Moreover, the graphs have fast mixing properties, meaning that we reach an almost uniform distribution on the graph after a short random walk [16].

Taking n steps in the ℓ-isogeny graph corresponds (up to isomorphism) to an isogeny $\varphi\colon E \to E'$ of degree $d = \ell^n$. For our purposes the degree of such isogenies will always be coprime to the characteristic p of the field and the isogeny φ is determined by a point K of order d on the staring curve E. This point generates the kernel of φ and we write $E' \cong E/\langle K \rangle$. In this case the d-torsion group $E[d]$ has d^2 elements and can be generated by two points P, Q of order d on E. This allows us to efficiently choose and describe a random walk by two integers a, b such that $K = [a]P + [b]Q$. We can even use this to define a random walk starting on a different curve. For an isogeny $\psi\colon E \to E''$ with degree coprime to the degree of φ the pushforward $[\psi]_\star\varphi$ is determined by the kernel $\langle \psi(K) = [a]\psi(P) + [b]\psi(Q) \rangle$ and starts at the codomain E'' of ψ. Note that although every supersingular elliptic curve has a representation in $\mathbb{F}_{p^2}$, the kernel of an isogeny and hence its generators might be elements of extensions $\mathbb{F}_{p^{2e}}$. With this notation we can define the adapted security game mentioned in Section 2.

Definition 3.5 (IND-CCA adapted). *For a NITC scheme* $\mathtt{TC}$ *and an algorithm* $\mathcal{A}$*, define the game* $\mathtt{IND\text{-}CCA}^{\mathcal{A}}_{\mathtt{TC}}$ *as follows:*

1. *Compute* $\mathbf{crs} \leftarrow \mathtt{PGen}(1^\kappa)$.
2. *Run* $\mathcal{A}(\mathbf{crs})$ *in a preprocessing phase with access to* $\mathtt{FDecom}(\mathbf{crs}, \cdot)$.
3. *When* $\mathcal{A}$ *outputs* (m_0, m_1)*, choose a uniform bit* $b \leftarrow \{0, 1\}$ *and then compute* $(\mathbf{C}_b, \pi_{\mathrm{com}}, \pi_{\mathrm{dec}}) \leftarrow \mathtt{Com}(\mathbf{crs}, m_b)$. *Give* $(\mathbf{C}_b, \pi_{\mathrm{com}})$ *to* $\mathcal{A}$*, who continues to have access to* $\mathtt{FDecom}(\mathbf{crs}, \cdot)$ *except that it may not query the oracle on* $(E', K', \cdot)$ *for* $E'/\langle K' \rangle \cong E_s/\langle K_T \rangle$ *and* $\mathbf{C}_b = (E_s, K_T, u_b)$.
4. *When* $\mathcal{A}$ *outputs a bit* b'*, it wins iff* $b' = b$.

Now we list some computational tasks that are relevant for isogeny-based cryptosystems. First we present tasks that can be solved efficiently and have a polynomial or even constant complexity.

Task 1: Compute isogenies given their kernels.

Task 2: Given two elliptic curves E, E', an isogeny $\varphi\colon E \to E'$ as well as the corresponding order $\mathcal{O} \cong \operatorname{End} E$ and ideal I_φ, compute $\mathcal{O}' \cong \operatorname{End} E'$.

Task 3: Given two elliptic curves E, E', and the corresponding orders $\mathcal{O} \cong \operatorname{End} E$, $\mathcal{O}' \cong \operatorname{End} E'$, compute a connecting ideal I corresponding to an isogeny $\varphi_I\colon E \to E'$.

Task 4: Given a left ideal I of a maximal order $\mathcal{O} \subset \mathcal{B}_{p,\infty}$, find an equivalent ideal $J = I\mathfrak{b}$ (for $\mathfrak{b} \in \mathcal{B}_{p,\infty}^*$) such that its norm is small(er) and smooth.

Task 5: Given $\mathcal{O} \cong \operatorname{End} E$, translate between isogenies $\varphi\colon E \to E'$ and their corresponding left $\mathcal{O}$-ideals I_φ.

Depending on the degree, Task 1 can be solved using Vélu's formulae [25] or the $\sqrt{\text{é}}\text{lu}$ algorithm [3]. For Task 2 we can compute $\mathcal{O}'$ as $\mathcal{O}_R(I_\varphi)$ and the connecting ideal I in Task 3 satisfies $\mathcal{O} = \mathcal{O}_L(I)$, where the left order $\mathcal{O}_L(I)$ is defined analogously to the right order $\mathcal{O}_R(I) = \mathcal{O}'$. Task 4 is solved by the KLPT algorithm [20] and Task 5 is addressed by subroutines of SQISign [12].

Note that these are not very specific and might have significantly different running times for special cases or when given additional information. For example, Task 1 is more efficient for smooth degrees than for non-smooth ones of similar size (see Section 5.2) and Task 5 can be done faster when given additional information.

In general, Task 5 only requires corresponding generators $\mathcal{O} = \langle \mathfrak{a}_0, \ldots, \mathfrak{a}_3 \rangle \cong \langle \alpha_0, \ldots, \alpha_3 \rangle = \operatorname{End} E$. Given an ideal I, the kernel of the corresponding isogeny φ_I is the set of points $K \in E$ such that $\alpha(K) = \infty$ for all $\alpha \in \operatorname{End} E$ corresponding to an element of I. Given an isogeny $\varphi\colon E \to E'$ with kernel $\langle K \rangle$, the corresponding ideal I_φ is the set of elements $\mathfrak{a} \in \mathcal{O}$ such that the corresponding $\alpha \in \operatorname{End} E$ satisfy $\alpha(K) = \infty$. If we know the norm or degree d (coprime to the characteristic p) of the ideal or isogeny, we can (pre)compute the action of the generators $(\alpha_0, \ldots, \alpha_3)$ of $\operatorname{End} E$ on the torsion group $E[d]$ and write it as 2×2 matrices $(A_0, \ldots, A_3)$ with respect to a basis (P, Q) of $E[d]$. Also, it suffices to find one point or one quaternion to generate the kernel or the ideal, respectively. Finding a generator is then reduced to finding a solution to a system of linear equations modulo d. If we additionally know two non-trivial endomorphisms θ, η such that $\langle K, \theta(K) \rangle = E[d]$ and the corresponding quaternions $\mathfrak{a}_\theta$, $\mathfrak{a}_\eta$ are orthogonal, we can solve $[a]K + [b]\theta(K) = \eta(K)$ (as matrix equation) to get $I_\varphi = \mathfrak{a}\mathcal{O} + d\mathcal{O}$ for $\mathfrak{a} = a + b\mathfrak{a}_\theta - \mathfrak{a}_\eta$ [9, Algorithm 23].

To create a delay we need moderately hard problems, which are still polynomial in complexity but might take a considerable time to compute. In Section 5.2 we show that Task 1 can be made sufficiently slow. The following hard problems have a conjectured exponential complexity (see Section 5.2) and are equivalent [26]. They are the basis for encryption or signature schemes like CSIDH [7] or SQISign [12]. In our case they ensure that there are no shortcuts for the forced decommitment.

Problem 3.6 (Isogeny Path Problem)**.** *Given two (isogenous) supersingular elliptic curves E, E' and a prime ℓ, find a path from E to E' in the ℓ-isogeny graph.*

Problem 3.7 (Endomorphism Ring Problem)**.** *Given a supersingular elliptic curve E, find four endomorphisms that generate* $\operatorname{End} E$ *as a lattice.*

Problem 3.8 (Maximal Order Problem)**.** *Given a supersingular elliptic curve E, find four quaternions in $\mathcal{B}_{p,\infty}$ that generate a maximal order $\mathcal{O}$ such that $\mathcal{O} \cong \operatorname{End} E$.*

Remark 3.9. *Knowledge of endomorphism rings can break the hard problems. If we know both endomorphism rings the first hard problem becomes polynomial using Tasks 3 - 5. If we know an isogeny from a curve with known endomorphism ring to our curve also the second hard problem becomes polynomial by Tasks 2 & 5. The third hard problem reduces to the second via Task 5.*

Finding supersingular elliptic curves can basically be done in two ways. We can reduce an elliptic curve in characteristic 0 modulo a prime and check if the resulting curve is supersingular, or take a random isogeny starting at one of these curves. In both cases the endomorphism ring of the final curve can be computed either via reduction or by transport along the isogeny. But as discussed in Remark 3.9 this weakens the hard problems. Hence many cryptosystems require curves with unknown endomorphism ring. This in turn forces them to use a multi-party computation or a trusted authority in their setup to ensure that no single participant knows a complete path from a curve with known endomorphism ring to the one used. See [**?**] for more information. Note that the present cryptosystem does not have this problem.

4 The Protocol

Now we can combine the previous two sections and present our construction. First we give a high-level overview and discuss some challenges. Then we look at the algorithms and choices for the parameters.

4.1 Overview

At the heart of our protocol is an isogeny φ_T of large degree d_T. Its domain is a public supersingular elliptic curve E_s with secret $\mathcal{O}_s \cong \operatorname{End} E_s$ and its kernel is generated by a publicly known point K_T on E_s. We use the j-invariant j_T of the codomain E_T of φ_T to hide the message $m \in M$. Therefore an adversary needs to compute E_T (or rather $j_T = j(E_T)$) in order to break hiding or to open the commitment by force. We can choose how long the commitment should be kept secret by setting the degree d_T accordingly. This gives us hiding. Since E_s and K_T are part of the commitment, the codomain $E_T \cong E_s/\langle K_T \rangle$ is fixed (up to isomorphism) and we have perfect binding.

For verification to be faster than forced opening, we need a more efficient way to compute j_T. The starting curve E_0 has a known endomorphism ring, which allows us to compute elements of $\operatorname{End} E_0 \cong \mathcal{O}_0$ efficiently using precomputations. During the commitment we choose a long isogeny $\varphi'_T \colon E_0 \to E'_T$ and use these precomputations to find another isogeny $\widetilde{\varphi}'_T \colon E_0 \to E'_T$ of much smaller and smooth degree (Tasks 4 & 5 from Section 3.2). Finally we use φ_s to push (the kernels of) φ'_T and $\widetilde{\varphi}'_T$ forward to (the kernels of) φ_T and $\widetilde{\varphi}_T$, respectively. We give φ_s and φ'_T to the verifier as part of the decommitment proof, so the commitment and the verification can compute E_T as the codomain of $\widetilde{\varphi}_T$. An adversary only knows E_s, but not φ_s or φ'_T and hence can neither compute $\mathcal{O}_s \cong \operatorname{End} E_s$ nor the pushforward $[\varphi_s]_\star \widetilde{\varphi}'_T = \widetilde{\varphi}_T$. Therefore it has no efficient way to compute a shortcut $\widetilde{\varphi}_T$. This gives us the preferred difference in speed for verification and forced opening. This is visualized in Figure 1.

$$\begin{tikzcd}
E_0 \arrow[r, "\varphi_s"] & E_s \cong E_0/\langle K_s \rangle \arrow[r, "\widetilde{\varphi}_T"] \arrow[r, bend left, "\varphi_T"] & E_T \cong E_s/\langle K_T \rangle
\end{tikzcd}$$

Figure 1: Walk in the isogeny graph with smooth $\deg(\varphi_s) = d_s \ll d_T = \deg(\varphi_T)$ and smooth $\deg(\widetilde{\varphi}_T) \ll d_T$ for the shortcut isogeny $\widetilde{\varphi}_T$.

To efficiently verify the validity of a commitment, we need to map the j-invariant j_T into the group of messages M. This map has to satisfy the following property. Otherwise the commitment might leak information about j_T.

Definition 4.1 (Inverse Resistant Functions)**.** *A function $f\colon X \to Y$ is λ-*inverse resistant*, if for uniform $x \in X$ the probability $\Pr[\mathcal{A}(f(x)) = x]$ is at most $2^{-\lambda}$ for all algorithms $\mathcal{A}$.*

This definition is weaker than one-way functions, since finding an element in the preimage is allowed as long as the probability to find the correct one is sufficiently small. It also differs from hash functions, which are mostly considered to be collision resistant. A simple projection with a sufficiently large preimage set satisfies this definition but is neither a one-way function nor a proper hash function.

4.2 Algorithms

As seen in Definition 2.1 we have five algorithms `PGen`, `Com`, `ComVrfy`, `DecVrfy` and `FDecom`. In this subsection we give pseudocode for each algorithm and discuss their (relative) speed and some subroutines.

4.2.1 Parameter Generation

The parameter generation `PGen` defines the security of the whole protocol and fixes the delay t_{fd}. It sets all general parameters like the characteristic p of the finite fields, the starting curve E_0, $\mathcal{O}_0 \cong \operatorname{End} E_0$, the degrees d_s and d_T, as well as the message group $(M, \oplus)$ and the inverse resistant function $F\colon J_{SS} \to M$. It also provides some precomputations that improve the speed of the commitment and the decommitment verification. These are bases of the d_s and d_T torsion groups of E_0, and matrices that correspond to the action of two endomorphisms θ and η on $E_0[d_T]$. It may also include an integer d_t dividing $p+1$ (or p^2-1) but coprime to d_s and matrices that correspond to the action of $\operatorname{End} E_0$ on $E_0[d_t]$. Its output is the common reference string **crs**. A detailed description can be found in Algorithm 1. The speed is dominated by finding generators of $E_0[d_T]$ and computing the action of θ and η on these points. The bottlenecks are checking the order and linear independency of two random points in $E_0(\mathbb{F}_{p^{2e}})$

and decomposing the images of this basis under the endomorphisms in terms of this basis.

Algorithm 1 Parameter generation algorithm `PGen`

Require: Security parameter 1^κ
Ensure: $\mathbf{crs} = \big((p, E_0, \operatorname{End} E_0, \mathcal{O}_0, d_s, P_s, Q_s, M, F), (d_T, P'_T, Q'_T, e, A_\theta, A_\eta)\big)$
Choose prime p of right size
Choose supersingular elliptic curve E_0 with known $\mathcal{O}_0 \cong \operatorname{End} E_0$
Find corresponding bases $\mathcal{O}_0 = \langle \mathfrak{a}_0, \ldots, \mathfrak{a}_3 \rangle$ and $\operatorname{End} E_0 = \langle \alpha_0, \ldots, \alpha_3 \rangle$
Choose $d_s \in \mathbb{N}$ such that $E_0[d_s] \subseteq E_0(\mathbb{F}_{p^2})$
Find $P_s, Q_s \in E_0$ such that $\langle P_s, Q_s \rangle = E_0[d_s]$
Choose a group $(M, \oplus)$ with efficient membership testing as message space
Choose an efficient, inverse resistant function $F\colon J_{SS} \to M$
Set $\mathbf{crs}_0 = (p, E_0, \operatorname{End} E_0, \mathcal{O}_0, d_s, P_s, Q_s, M, F)$ ▷ *Depends only on* κ
Choose $e, d_T \in \mathbb{N}$ such that d_T is coprime to d_s and $E_0[d_T] \subseteq E_0(\mathbb{F}_{p^{2e}})$
Find $P'_T, Q'_T \in E_0(\mathbb{F}_{p^{2e}})$ such that $\langle P'_T, Q'_T \rangle = E_0[d_T]$
Find endomorphisms $\theta, \eta \in \operatorname{End} E_0$ such that $\langle R, \theta(R) \rangle = E_0[d_T]$ for all $R \in E_0[d_T]$ of maximal order and the corresponding quarternions $\mathfrak{a}_\theta, \mathfrak{a}_\eta$ are orthogonal
Compute the action of θ, η on $E_0[d_T]$ as matrices $A_\theta, A_\eta \in \operatorname{Mat}_{2\times 2}(\mathbb{Z}/d_T\mathbb{Z})$ with respect to the basis (P'_T, Q'_T)
Set $\mathbf{crs}_T = (d_T, P'_T, Q'_T, e, A_\theta, A_\eta)$ ▷ *Fixes delay* t_{fd}
return $(\mathbf{crs}_0, \mathbf{crs}_T)$

4.2.2 Commitment

The commitment algorithm `Com` takes as input a message $m \in M$ and outputs a tuple $(\mathbf{C}, \pi_{\mathrm{com}}, \pi_{\mathrm{dec}})$. First it chooses a random isogeny $\varphi_s\colon E_0 \to E_s$ of degree d_s and a second random isogeny $\varphi'_T\colon E_0 \to E'_T$ of large degree d_T. It uses $\mathcal{O}_0$ and the corresponding ideal I'_T to find an equivalent ideal $\widetilde{I}'_T$ and corresponding isogeny $\widetilde{\varphi}'_T\colon E_0 \to E'_T$ of smooth and much smaller degree $\widetilde{d}_T$. Here we use algorithms from SQISign [9] to translate between isogenies and ideals and to find equivalent ideals of specific norm. Finally it pushes $\widetilde{\varphi}'_T$ forward to the shortcut $\widetilde{\varphi}_T = [\varphi_s]_\star \widetilde{\varphi}'_T\colon E_s \to E_T$ of degree $\widetilde{d}_T$ for the long isogeny $\varphi_T = [\varphi_s]_\star \varphi'_T\colon E_s \to E_T$ of degree d_T. This allows it to efficiently compute the j-invariant $j_T = j(E_T)$ and $u = m \ominus F(j_T) \in M$. The commitment itself $\mathbf{C} = (E_s, K_T, u)$ is again a tuple of a supersingular elliptic curve E_s, a point K_T on E_s that generates the kernel of φ_T and $u \in M$. While the commitment proof π_{com} is empty, the decommitment proof π_{dec} allows to reconstruct the secret isogeny φ_s and to use the same method of finding a shortcut $\widetilde{\varphi}_T$ for φ_T as in `Com`. The individual steps are given in Algorithm 2. In SQISign [9] the authors state that converting between isogenies and ideals is the bottleneck of their computation. Therefore we assume that the slowest part of this algorithm is computing $\widetilde{\varphi}_T$. Note however, that SQISign is still fast and our commitment algorithm will be faster than computing the long isogeny φ_T (over $\mathbb{F}_{p^{2e}}$) directly. A more detailed discussion can be found in Section 5.3.

Remark 4.2. *We can force the degree $\widetilde{d}_T$ of the shortcut isogeny to divide some d_t coprime to d_s and precompute the action of* $\operatorname{End} E_0$ *on* $E_0[d_t]$ *as matrices to*

Algorithm 2 Commitment algorithm `Com`

Require: Common reference string **crs**, message $m \in M$

Ensure: $(\mathbf{C}, \pi_{\text{com}}, \pi_{\text{dec}}) = \big((E_s, K_T, u), (), (s, t)\big)$

1: Choose random $s \in [0, d_s)$ and compute $K_s = P_s + [s]Q_s \in E_0[d_s]$
2: Compute $E_s \cong E_0/\langle K_s \rangle$ via Vélu's formulae
3: Choose random $t \in [0, d_s)$ and set $v = (1, t)^\top$, $v' = A_\theta v$
4: Set $A = (v, v') \in \mathrm{GL}_2(\mathbb{Z}/d_T\mathbb{Z})$ and compute $(w_1, w_2)^\top = A^{-1} A_\eta v$
5: Compute ideal $I'_T = (w_1 + w_2 \mathfrak{a}_\theta - \mathfrak{a}_\eta)\mathcal{O}_0 + d_T \mathcal{O}_0$ corresponding to isogeny $\varphi'_T \colon E_0 \to E'_T$ of degree d_T with kernel $\langle K'_T = P'_T + [t]Q'_T \rangle$
6: Try to compute equivalent ideal $\widetilde{I}'_T$ with smooth norm $\widetilde{d}_T \ll d_T$ coprime to d_s such that $\widetilde{d}_T \mid (p+1)$ (or $\widetilde{d}_T \mid (p^2 - 1)$) and go back to step 3 if it fails
7: Compute corresponding isogeny $\widetilde{\varphi}'_T$ of degree $\widetilde{d}_T$ and kernel $\langle \widetilde{K}'_T \rangle$
8: Compute pushforward $\widetilde{\varphi}_T = [\varphi_s]_\star \widetilde{\varphi}'_T \colon E_s \to E_T$ with kernel $\langle \varphi_s(\widetilde{K}'_T) \rangle$
9: Compute $K_T = \varphi_s(P'_T + [t]Q'_T) \in E_s[d_T]$
10: Compute $E_T \cong E_s/\langle K_T \rangle$ as codomain of $\widetilde{\varphi}_T$
11: Compute $j_T = j(E_T)$ and $u = m \ominus F(j_T) \in M$
12: Set $\mathbf{C} = (E_s, K_T, u)$ ▷ *Commitment*
13: Set $\pi_{\text{com}} = ()$ ▷ *Commitment proof (empty)*
14: Set $\pi_{\text{dec}} = (s, t)$ ▷ *Decommitment proof*
15: **return** $(\mathbf{C}, \pi_{\text{com}}, \pi_{\text{dec}})$

improve the speed of translating $\widetilde{I}'_T$ into $\widetilde{\varphi}'_T$. If we choose d_t such that $\widetilde{d}_T$ divides $(p+1)$ or (p^2-1), the kernel of $\widetilde{\varphi}_T$ is $\mathbb{F}_{p^2}$- or $\mathbb{F}_{p^4}$-rational, respectively. This allows for fast evaluation of $\widetilde{\varphi}_T$.

4.2.3 Commitment Verification

Algorithm 3 shows the commitment verification `ComVrfy`. It is fast since it only needs to check if the three parts of the commitment are of the correct form. Namely, E_s is an elliptic curve, K_T is a point on that curve and u is an element of the group M. All of these membership tests can be done efficiently. If we want to assure that forced opening does not take too long, we can also check if $K_T \in \mathbb{F}^2_{p^{2e}}$. This sets an upper bound of $d_T \leq p^e - (-1)^e$ for the degree d_T of φ_T.

Algorithm 3 Commitment verification algorithm `ComVrfy`

Require: Common reference string **crs**, commitment $\mathbf{C}$ and proof π_{com}

1: Check if E_s is an elliptic curve over $\mathbb{F}_{p^2}$, $K_T \in E_s$ and $u \in M$
Optional: check $K_T \in \mathbb{F}^2_{p^{2e}}$ ▷ *Check upper bound for degree of* φ_T
2: **return** (**accept**/**reject**)

4.2.4 Decommitment Verification

The decommitment verification `DecVrfy` (Algorithm 4) is similar to the commitment algorithm. It first reconstructs φ_s from π_{dec} and verifies $\varphi_s \colon E_0 \to E_s$ and $K_T = \varphi_s(P'_T + [t]Q'_T)$. Then it uses $\mathcal{O}_0$ to find a shortcut isogeny $\widetilde{\varphi}'_T$ for φ'_T of smooth and much smaller degree $\widetilde{d}_T$ and pushes it forward to $\widetilde{\varphi}_T =$

$[\varphi_s]_\star \widetilde{\varphi}'_T \colon E_s \to E_T$. With this short isogeny it computes $j_T = j(E_T)$ and checks if $u \oplus F(j_T) = m$. As stated above, we assume the slowest part of this algorithm to be the computation of $\widetilde{\varphi}_T$. Again, this is still faster than forced decommitment, as $\deg \widetilde{\varphi}_T$ is smooth and smaller than $\deg \varphi_T$ (cf. Section 5.3). Here the same improvements as in Remark 4.2 for the commitment can be applied.

Algorithm 4 Decommitment verification algorithm `ComVrfy`

Require: Common reference string **crs**, commitment **C**, message m, decommitment proof π_{dec}

Compute $K_s = P_s + [s]Q_s \in E_0[d_s]$ and check $E_s \cong E_0/\langle K_s \rangle$
Compute $K'_T = P'_T + [t]Q'_T \in E_0[d_T]$ and check $\varphi_s(K'_T) = K_T$
Set $v = (1,t)^\top$, $v' = A_\theta v$ and $A = (v, v') \in \mathrm{GL}_2(\mathbb{Z}/d_T\mathbb{Z})$
Compute $(w_1, w_2)^\top = A^{-1} A_\eta v_1$
Compute ideal $I'_T = (w_1 + w_2 \mathfrak{a}_\theta - \mathfrak{a}_\eta)\mathcal{O}_0 + d_T \mathcal{O}_0$ corresponding to isogeny $\varphi'_T \colon E_0 \to E'_T$ of degree d_T with kernel $\langle K'_T \rangle$
Compute equivalent ideal $\widetilde{I}'_T$ with smooth norm $\widetilde{d}_T \ll d_T$ coprime to d_s such that $\widetilde{d}_T \mid (p+1)$ (or $\widetilde{d}_T \mid (p^2-1)$)
Compute corresponding isogeny $\widetilde{\varphi}'_T$ of degree $\widetilde{d}_T$ and kernel $\langle \widetilde{K}'_T \rangle$
Compute pushforward $\widetilde{\varphi}_T = [\varphi_s]_\star \widetilde{\varphi}'_T \colon E_s \to E_T$ with kernel $\langle \varphi_s(\widetilde{K}'_T) \rangle$
Compute $E_T \cong E_s/\langle K_T \rangle$ as codomain of $\widetilde{\varphi}_T$
Compute $j_T = j(E_T)$ and check $u \oplus F(j_T) = m$
return (**accept**/**reject**)

4.2.5 Forced Decommitment

In terms of the number of tasks the forced decommitment algorithm is rather simple. It just computes E_T as codomain of the isogeny φ_T given by the point K_T that generates its kernel. From there it recovers the message $m = u \oplus F(j(E_T))$. Computing an isogeny φ_T of large degree d_T is slow (cf. Theorem 5.11), especially when the calculations have to be done in a field extension $\mathbb{F}_{p^{2e}}$. This allows us to make Algorithm 5 (almost) arbitrarily slow.

Algorithm 5 Forced decommitment algorithm `FDecom`

Require: Common reference string **crs**, commitment **C**
Ensure: Message m

Compute $E_T \cong E_s/\langle K_T \rangle$ via Vélu's formulae or $\sqrt{\text{élu}}$ algorithm
Compute $j_T = j(E_T)$ and $m = u \oplus F(j_T)$
return m

4.3 Parameter Sizes and other Choices

The algorithms above do not specify all properties of the parameters. Therefore we now discuss the necessary and some optional choices. For example, the hiding property sets requirements on the size of some parameters and we also propose some choices for implementing this protocol.

The delay t_{fd} should be large, but it has to be polynomial in κ (or $\log p$). On one hand the main idea of NITC schemes is that we can forcefully open a commitment (in polynomial time) with `FDecom`, if someone refuses to open it themselves. On the other hand generic algorithms to solve Problems 3.6 - 3.8 could be faster than `FDecom` and therefore violate hiding, if t_{fd} were superpolynomial. In particular, we need $t_{\mathrm{fd}} < \min\{d_s^{1/4}, p^{1/4}\}$ due to Assumptions 5.9 and 5.10 of Section 5.2 for quantum security and $t_{\mathrm{fd}} \gg t_{\mathrm{cv}}, t_{\mathrm{dv}}$.

4.3.1 Prime p, starting curve E_0 and isogenies φ_s and φ_T

In order to satisfy the hiding property, p and d_s have to have a certain size. It has to be infeasible to precompute $\mathcal{O}_s$ for all possible E_s or to find an isogeny from E_0 to E_s in time less than t_{fd} in the online phase. Therefore we choose $p \approx 2^{2\kappa}$ and $\sqrt{p} \lesssim d_s \lesssim p$ or equivalently $2^{\kappa} \lesssim d_s \lesssim 2^{2\kappa}$. The degree d_T is chosen such that computing an isogeny of degree d_T takes at least time t_{fd}, but not much more, and we require $d_s \leq d_T$. Since $P_T = \varphi_s(P_T')$ and $Q_T = \varphi_s(Q_T')$ have to generate $E_s[d_T]$, we need the degree d_s of φ_s to be coprime to d_T. In Section 5 we give a more detailed justification of these numbers.

The starting curve could be any supersingular elliptic curve E_0 with a known efficient representation of $\mathcal{O}_0$. For our protocol we choose E_0 to be the curve $E_0\colon y^2 = x^3 + x$ with $(p+1)^2$ points over $\mathbb{F}_{p^2}$ and $\mathcal{O}_0 = \langle 1, \mathrm{i}, \frac{\mathrm{i}+\mathrm{j}}{2}, \frac{1+\mathrm{k}}{2}\rangle_{\mathbb{Z}}$ for $p \equiv 3 \bmod 4$. In this case the endomorphisms $[\mathrm{i}]\colon (x, y) \mapsto (-x, \mathrm{i}y)$, $\phi\colon (x, y) \mapsto (x^p, y^p)$ (Frobenius map), θ and η correspond to i, j, $(\mathrm{j}+\frac{1+\mathrm{k}}{2})$ and i, respectively[2]. Usually, we would want d_s and d_T to be smooth numbers both dividing $p+1$ in order to have fast evaluation of the corresponding isogenies. So d_s should be smooth and divide $p+1$ (or p^2-1). However, evaluating φ_T does not need (in fact should not) be efficient, since it is only evaluated by `FDecom`. Therefore d_T can contain large prime factors and should be large.

For a supersingular curve with $(p+1)^2$ points over $\mathbb{F}_{p^2}$ we have $(p^e - (-1)^e)^2$ points over $\mathbb{F}_{p^{2e}}$ and the largest fully $\mathbb{F}_{p^{2e}}$-rational torsion group is the $(p^e - (-1)^e)$-torsion. This means that for large d_T we need to go to extensions of $\mathbb{F}_{p^2}$ to find a basis for the d_T-torsion group of E_0 or E_s. Higher extensions and larger p slow down the computations, therefore we want to minimize the degree of the extension and the size of p to increase efficiency. Since the size of p affects almost all computations, whereas the size of e only influences computations related to the K_T or φ_T, it can be beneficial to choose a smaller p and a larger e when dealing with large d_T.

For an implementation we can choose a prime p such that $p+1$ contains a smooth factor $d_s = 2^{\kappa}$. This ensures that the first isogeny $\varphi_s\colon E_0 \to E_s$ can be evaluated efficiently. After choosing a prime, we find an extension degree e such that $p^e - (-1)^e$ contains a sufficiently large factor d_T that is coprime to d_s. For example, we can choose d_T odd with prime factorization $d_T = \prod q_i^{e_i}$ such that $\sum e_i \sqrt{q_i} > t_{\mathrm{fd}}$ (cf. Section 5.2). The primes used in SQISign and SIKE allow to choose d_s (and d_T) this way. So there are already known primes with the right properties for different security levels. The primes for SQISign even allow $d_s \approx p$, if the degree d_s is just required to divide $p^2 - 1$ instead of $p+1$. Then t_{fd} can be almost as large as $p^{1/4}$ instead of $p^{1/8}$. This could be a good trade-off for large delays.

[2]This is not a typo. We have $\eta = [\mathrm{i}]$.

4.3.2 Message space M and function F

We choose M to be a finite group $M = \mathbb{Z}/N\mathbb{Z}$ for an integer $N \in \mathbb{N}$. This gives us very efficient membership testing and group operations. The size of N depends on the needed length of a message m and the prime p. If N is larger than $\lfloor p/12 \rfloor + 2$, then $F\colon J_{SS} \to M$ can not be surjective and therefore $u = m \ominus F(j_T)$ might leak information about the message m.

As mentioned before, computing j_T from $F(j_T)$ has to be infeasible or at least slow. In order to satisfy hiding we choose the function F to be λ-inverse resistant with $\lambda = \kappa \approx \log \sqrt{p}$. In addition, it has to be fast since `Com` and `DecVrfy` have to compute $F(j_T)$. An easy way to accomplish this is to take a function that is not injective. The larger the kernel of F, i.e. smaller N, the more information is lost. A simple projection $\mathbb{F}_{p^2} \supset J_{SS} \to \mathbb{F}_p$ onto one of the components or even their sum will leak information, since there is a subset of j-invariants that already are in $\mathbb{F}_p$. If we use a simple map like $(a, b) \mapsto b \bmod N$ or $(a, b) \mapsto a + b \bmod N$, we thus need to use $N \ll p$. For an implementation we can identify $J_{SS} \subset \mathbb{F}_{p^2}$ with a subset of $\mathbb{F}_p[\mathrm{i}] \cong \mathbb{F}_{p^2}$ and choose

$$F\colon J_{SS} \to M = \mathbb{Z}/N\mathbb{Z}, \quad a + b\mathrm{i} \mapsto a + b \bmod N$$

with $N = \lfloor \sqrt{p}/12 \rfloor$. Then we can expect every element in M to be the image of about $\sqrt{p} \approx 2^\kappa$ elements in J_{SS}. There is no direct way of finding the supersingular j-invariants. Hence, one would have to compute the preimage in $\mathbb{F}_{p^2}$ (about $12p^{3/2}$ elements) and check if they are j-invariants of supersingular elliptic curves. This is sufficiently inverse resistant in practice.

Remark 4.3. *If $d_s = N \approx \sqrt{p}$ (or d_s slightly smaller) we can add $v = s \ominus F(j_T)$ to the commitment $\mathbf{C}$ to make the scheme publicly verifiable. In this case $s \in [0, d_s)$ can be uniquely recovered from v if we know $F(j_T)$, allowing to compute $\varphi_s(P'_T), \varphi_s(Q'_T)$ and finding t. So* `FDecom` *could also provide the decommitment proof $\pi_{\text{dec}} = (s, t)$ and everyone could use* `DecVrfy` *to verify the output of* `FDecom` *instead of computing it themselves. Since s can be considered as a random number in M (in this case), the additional v in the commitment will neither leak information about $F(j_T)$ nor about s unless we already know $F(j_T)$.*

5 Security

We show that our protocol satisfies the Definition 2.1 of a NITC scheme by Katz et al. [19] and prove the three properties practicality, hiding and binding. In order to prove practicality, we need assumptions for the relative speed of some algorithms. Remember that our timings are the number of operations rather than real world times.

Remark 5.1. *Operations in $\mathbb{F}_{p^{2e}}$ are slower than operations in $\mathbb{F}_{p^2}$. In particular, the majority of operations of* `FDecom` *are in extension fields, but for* `Com`*,* `ComVrfy` *and* `DecVrfy` *most operations can be done in $\mathbb{F}_{p^2}$. So our timings are rather conservative.*

Our algorithms have the correct input and output arguments and for all κ and $m \in M$ every set of honestly generated $(\kappa, m, \mathbf{crs}, \mathbf{C}, \pi_{\text{com}}, \pi_{\text{dec}})$ satisfies verification $\texttt{ComVrfy}(\mathbf{crs}, \mathbf{C}, \pi_{\text{com}}) = \mathbf{accept} = \texttt{DecVrfy}(\mathbf{crs}, \mathbf{C}, m, \pi_{\text{dec}})$ and forced decommitment $\texttt{FDecom}(\mathbf{crs}, \mathbf{C}) = m$. This makes it a NITC scheme.

5.1 Hiding and Binding

For hiding we use the same (non-malleability) Definition 2.4 as Katz et al. [19]. First we show why we need an adapted security game. In Definition 2.3 the adversary $\mathcal{A}$ sends two messages m_0, m_1 and receives the commitment $\mathbf{C}_b = (E_s, K_T, u_b)$ corresponding to message m_b for a uniform $b \in \{0,1\}$. It is allowed to query an oracle for $\texttt{FDecom}(\cdot)$ except for $\texttt{FDecom}(\mathbf{crs}, \mathbf{C}_b)$.

Lemma 5.2. *An adversary $\mathcal{A}$ can break hiding with the original security game from Definition 2.3.*

Proof. Since $m_{1-b} \ominus m_b \oplus u_b = u_{1-b}$, querying $\texttt{FDecom}(\mathbf{crs}, (E_s, K_T, u_\pm))$ with $u_+ = (m_0 \ominus m_1) \oplus u_b$ and $u_- = \ominus(m_0 \ominus m_1) \oplus u_b$ gives m_{1-b} and a random message m'. For $|M| = 2$ we have $u_+ = u_-$ and get m_{1-b}. For $|M| > 2$ however, we can assume $m_0 \neq m' \neq m_1$. This allows $\mathcal{A}$ to output the correct $b' = b$ with high probability.

Even worse, if we replace K_T by any other point K' such that $\langle K' \rangle = \langle K_T \rangle$, e.g. $K' = [\ell]K_T$ for ℓ coprime to d_T, or apply an isomorphism such that $E'/\langle K' \rangle \cong E_T \cong E_s/\langle K_T \rangle$ then $\texttt{FDecom}(\mathbf{crs}, (E', K', u_b))$ will return m_b. $\square$

Thus, it is reasonable to disallow queries of the form $\texttt{FDecom}(\mathbf{crs}, (E', K', \cdot))$ for $E'/\langle K' \rangle \cong E_s/\langle K_T \rangle$, i.e. using the adapted security game in Definition 3.5. This is still in the spirit of the original definition, since it prohibits the "decryption" of the commitment in question. In our case the security arises from the secret isogeny $\varphi_s \colon E_0 \to E_s$ and the long isogeny $\varphi_T \colon E_s \to E_T$ with kernel $\langle K_T \rangle$, and the "key" is $F(j_T)$ for $j_T = j(E_T)$. Such queries would enable $\mathcal{A}$ to find $F(j_T)$ and would hence basically allow to query $\texttt{FDecom}(\mathbf{crs}, (E_s, K_T, u_b))$ by proxy, which is forbidden in the original definition.

Assumption 5.3. *We assume that the probability to find the correct output in the online phase (step 3) of the security game from Definition 3.5 in time $t_o < t_{\mathrm{fd}}$ is less than $2^{-\kappa}$ if F is a κ-inverse resistant function.*

Let us justify this assumption by looking at the security game from Definition 3.5. Assume that F is a κ-inverse resistant function as specified in the protocol. In the online phase $\mathcal{A}$ sends two messages m_0, m_1 and receives the output (E_s, K_T, u_b) of $\texttt{Com}(\mathbf{crs}, m_b)$ for a uniform $b \in \{0,1\}$. The adversary $\mathcal{A}$ knows that $F(j_T)$ is equal to $F_0 = \ominus u_b \oplus m_0$ or $F_1 = \ominus u_b \oplus m_1$, but for each $i \in \{0,1\}$ there are at least 2^κ j-invariants such that $F(j) = F_i$ and none of them is more likely than the other. To verify one of them, $\mathcal{A}$ would have to compute $E_s/\langle K_T \rangle$. But since this is equivalent to computing $\texttt{FDecom}(\mathbf{crs}, (E_s, K_T, u_b))$, it can not be done in time less than t_{fd}. Similarly, querying $\texttt{FDecom}(\mathbf{crs}, (E_s, [\ell]K_T, u_b))$ for $\ell \mid d_T$ gives $m_\ell = u_b \oplus F(j_\ell)$ and hence $F(j_\ell) = \ominus u_b \oplus m_\ell$ for j-invariants j_ℓ of intermediate curves of the long isogeny φ_T. But since F is an κ-inverse resistant function, there are at least 2^κ undistinguishable candidates for each j_ℓ. For a (small) prime ℓ a match between the $(\ell+1)$ neighbours of each candidate for j_ℓ in the ℓ-isogeny graph and the candidates for j_T from each F_0 and F_1 has to be found. The probability to find such a match is less than $t_o 2^{-2\kappa} < 2^{-\kappa}$ using $t_o < t_{\mathrm{fd}} < p^{1/4} < 2^\kappa$ and the fact that not all of time t_o can be spent on this task. Replacing E_s and K_T by a curve E' and point K' such that $E'/\langle K' \rangle$ is unrelated to $E_s/\langle K_T \rangle$ or intermediate curves the query $\texttt{FDecom}(\mathbf{crs}, (E', K', u_b))$ will give completely unrelated results.

Theorem 5.4. *For a κ-inverse resistant function F and under Assumption 5.3, SIGNITC is (t_p, t_o, ε)-CCA-secure (satisfies hiding) with security game from Definition 3.5 for $t_p \ll 2^\kappa$ polynomial in κ, $t_o < t_{\mathrm{fd}}$ and $\varepsilon = 2^{-\kappa}$.*

Proof. The precomputation phase can only provide a negligible advantage for an adversary $\mathcal{A}$. The computation of $\mathtt{Com}(\mathbf{crs}, m)$ includes choosing random $K_s = P_s + [s]Q_s \in E_0[d_s]$ and $K_T = P_T + [t]Q_T \in E_s[d_T]$ with $s, t \in [0, d_s)$. Since $2^\kappa \approx \sqrt{p} \lesssim d_s$, it is infeasible to precompute (and store) a significant subset of all possibilities in time $t_p \ll 2^\kappa$ polynomial in κ. For the online phase Assumption 5.3 gives us that the advantage over guessing is less than $2^{-\kappa}$. □

The proof for binding works with the original Definition 2.6 and security game from Definition 2.5. With our protocol we even achieve perfect binding.

Theorem 5.5. *SIGNITC is $(\infty, 0)$-BND-CCA-secure (satisfies binding) with security game from Definition 2.5.*

Proof. If the commitment $\mathbf{C}$ is accepted by $\mathtt{ComVrfy}$, then it contains an elliptic curve E_s, a point K_T on E_s and an element u of an additive group M. Since $\mathtt{DecVrfy}$ verifies that the codomain of $\varphi_T = [\varphi_s]_\star \varphi'_T$ is $E_T \cong E_s/\langle K_T \rangle$, we have that acceptance of both $(\mathbf{crs}, \mathbf{C}, m, \pi_{\mathrm{dec}})$ and $(\mathbf{crs}, \mathbf{C}, m', \pi'_{\mathrm{dec}})$ by $\mathtt{DecVrfy}$ implies $m \ominus F(j_T) = u = m' \ominus F(j_T)$ and hence $m = m'$. The speedup does not change this. Similarly, if $\mathtt{DecVrfy}$ accepts $(\mathbf{crs}, \mathbf{C}, m, \pi_{\mathrm{dec}})$ then $u = m \ominus F(j_T)$ and $\mathtt{FDecom}$ computes $F(j_T)$ from E_s and K_T and thus outputs the correct message $m = u \oplus F(j_T)$. □

5.2 Relative Running Times

Computing isogenies of prime degree q can be done using Vélu's formulae in time $O(q)$, or the $\sqrt{\text{élu}}$ algorithm [3] in time $\sqrt{q}(\log q)^{2+o(1)}$ or $\widetilde{O}(\sqrt{q})$ for short. Here $\widetilde{O}$ may also contain additional logarithmic terms $\widetilde{O}(n) = O(n \operatorname{poly}(\log n))$. The crossover point for optimized algorithms is at $q \approx 100$, and we denote the time it takes to compute an isogeny of prime degree q with $\mathrm{eval}_{\mathrm{prime}}(q)$. Computing isogenies efficiently is a well-studied topic and we will assume that these timings are close to optimal.

Lemma 5.6. *There is a (small) constant c_p such that evaluating an isogeny of prime degree q takes time $\mathrm{eval}_{\mathrm{prime}}(q) \leq c_p q$.*

Now let us look at an isogeny φ with a kernel that is generated by a point K'_0 of order q^ℓ. We can decompose $\varphi = \varphi_\ell \circ \cdots \circ \varphi_1$ into isogenies φ_i of degree q. In each step we compute the points $K_i = [q^{\ell-i}]K'_{i-1}$ generating the kernel of φ_i and $K'_i = \varphi_i(K'_{i-1})$ generating the kernel of $\varphi'_i = \varphi_\ell \circ \cdots \circ \varphi_{i+1}$. So every step takes time $\mathrm{eval}_{\mathrm{prime}}(q)$ plus the time it takes to compute the point multiplication. Generalizing this to isogenies of arbitrary composite degree gives us bounds for the time $\mathrm{eval}(d)$ it takes to compute an isogeny of degree d. If we ignore the multiplications for the lower bound we get the following lemma.

Lemma 5.7. *Let $d = \prod_{i=1}^r q_i^{e_i}$ be the prime factorization of the degree d. There is a (small) constant $c_c \geq 1$ such that the time $\mathrm{eval}(d)$ it takes to evaluate an isogeny of degree d is bounded by*

$$\sum_{i=1}^r e_i \,\mathrm{eval}_{\mathrm{prime}}(q_i) \leq \mathrm{eval}(d) \leq c_c \sum_{i=1}^r e_i \,\mathrm{eval}_{\mathrm{prime}}(q_i).$$

This allows us to choose d_T such that $\mathrm{eval}(d_T) \geq t_{\mathrm{fd}}$. Combining these results we get an upper bound for the computation time of isogenies of smooth degree.

Lemma 5.8. *Evaluating an isogeny of B-smooth degree d with prime factorization $d = \prod_{i=1}^{r} q_i^{e_i}$ takes time* $\mathrm{eval}(d) \in O(\frac{B}{\log B} \log d)$.

Proof. We use Lemmas 5.7 and 5.6 to write

$$\mathrm{eval}(d) \leq c_c \sum_{i=1}^{r} e_i \, \mathrm{eval}_{\mathrm{prime}}(q_i) \leq c_c c_p \sum_{i=1}^{r} e_i q_i.$$

Since $q_i < B$ for all $1 \leq i \leq r$, we get $q_i \leq \log q_i \frac{B}{\log B}$ and

$$\mathrm{eval}(d) \leq c_c c_p \sum_{i=1}^{r} e_i \log q_i \frac{B}{\log B} = c_c c_p \frac{B}{\log B} \log d. \qquad \square$$

According to Eisenträger et al. [15] the fastest (currently known) algorithms for solving the (equivalent) general Isogeny Path Problem, general Endomorphism Ring Problem or general Maximal Order Problem (cf. Section 3.2) over $\mathbb{F}_{p^2}$ take time $\widetilde{O}(p^{1/2})$ for classical computations and $\widetilde{O}(p^{1/4})$ with a quantum computer. Since E_0 and E_s are known to be connected by a d_s-isogeny there is also a meet-in-the-middle or claw-finding attack in classical time $\widetilde{O}(d_s^{1/2})$ and $\widetilde{O}(d_s^{1/4})$ when applying Grover's Algorithm [17].

Assumption 5.9 (General Isogeny Assumption)**.** *We assume that the fastest algorithms to solve the general Isogeny Path Problem, the general Endomorphism Ring Problem or the general Maximal Order Problem over $\mathbb{F}_{p^2}$ need at least $p^{1/2}$ or $p^{1/4}$ operations for classical or quantum algorithms, respectively.*

Assumption 5.10 (Special Isogeny Assumption)**.** *We assume that the fastest algorithms to find an isogeny between two d-isogenous curves over $\mathbb{F}_{p^2}$ with $d < p$ take at least $d^{1/2}$ or $d^{1/4}$ operations for classical or quantum algorithms, respectively.*

With these assumptions we can prove that computing the codomain of an isogeny can be made almost arbitrarily slow.

Theorem 5.11. *Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ with unknown $\mathcal{O} \cong \operatorname{End} E$, but d'-isogenous to a curve E_0 with known endomorphism ring. Let further K be a point on E of order d, such that computing the corresponding isogeny takes at least time t, according to Lemma 5.7. Then for $t < \min\{d'^{1/4}, p^{1/4}\}$ and under Assumptions 5.9 and 5.10, computing $E_K \cong E/\langle K \rangle$ takes at least time t.*

Proof. The isogeny $\varphi \colon E \to E_K$ with kernel $\langle K \rangle$ has degree d. Efficiently calculating a shortcut isogeny $\widetilde{\varphi} \colon E \to E_K$ requires knowledge of $\mathcal{O} \cong \operatorname{End} E$. Finding the endomorphism ring $\operatorname{End} E$ or the order $\mathcal{O} \cong \operatorname{End} E$ without an isogeny $\varphi' \colon E_0 \to E$, or finding an isogeny $\widetilde{\varphi}$ without $\mathcal{O} \cong \operatorname{End} E$ are hard problems. By Assumption 5.9 solving these problems takes time at least $p^{1/4} > t$. Finding an isogeny φ' needs at least time $d'^{1/4} > t$ by Assumption 5.10 if $d' < p$ or $p^{1/4} > t$ by Assumption 5.9 if $d' \geq p$. Therefore computing $E_T \cong E_s/\langle K_T \rangle$ takes at least time t. $\square$

The algorithm `FDecom` only has **crs** and $\mathbf{C} = (E_s, K_T, u)$ as input. In order to compute $m = u \oplus F(j_T)$ it has to calculate the j-invariant j_T of the secret curve $E_T \cong E_s/\langle K_T \rangle$. Theorem 5.11 gives us the following corollary:

Corollary 5.12. *For $t_{\mathrm{fd}} < \min\{d_s^{1/4}, p^{1/4}\}$ and under the Assumptions 5.9 and 5.10, the forced decommitment* `FDecom` *takes at least time t_{fd}.*

Note that the restriction $t_{\mathrm{fd}} < \min\{d_s^{1/4}, p^{1/4}\}$ is based on the quantum timings in Assumptions 5.9 and 5.10. For classical algorithms $t_{\mathrm{fd}} < \min\{d_s^{1/2}, p^{1/2}\}$ would be sufficient, but since our protocol should be quantum secure we chose the more general bound including quantum algorithms.

5.3 Practicality

We show that `Com`, `ComVrfy` and `DecVrfy` can be computed efficiently and that we achieve a perfectly practical NITC scheme. We chose $p \approx 2^{2\kappa}$, $\sqrt{p} \lesssim d_s \lesssim p$ and $t_{\mathrm{fd}} < \min\{d_s^{1/4}, p^{1/4}\}$ to get κ bits of classical and $\kappa/2$ bits of quantum security for the precomputation phase in hiding. In this subsection "efficiently" means a running time at most polynomial in $\log p$.

Lemma 5.13. *The commitment* `Com` *takes time $t_{\mathrm{com}} \in \mathrm{poly}(\log p)$.*

Proof. The number of operations on E_0 for computing $K_s = P_s + [s]Q_s$ and $K_T' = P_T' + [t]Q_T'$ is linear in $\log d_s$ since $0 \leq s, t < d_s$. By Lemma 5.8 we can find $E_s \cong E_0/\langle K_s \rangle$ and $K_T = \varphi_s(K_T')$ via Vélu's formulae in time $O(\frac{B}{\log B} \log d_s)$ if d_s is B-smooth. We adapted Algorithm 23 from [9] to compute I_T' from K_T' using one inversion and a few additions and multiplications modulo d_T. By heuristics from SQISign, the norm of an equivalent ideal can be expected to be roughly $\sqrt{p}$ or smaller. We can see that finding an equivalent ideal of smooth norm $\widetilde{d}_T$ and translating it into an isogeny can be done efficiently via the algorithms from [9]. Evaluating this isogeny to find $E_T \cong E_s/K_T$ is in $O(\frac{B}{\log B} \log \sqrt{p})$ if $\widetilde{d}_T$ is B-smooth. Finally we have to compute $j_T = j(E_T)$ and $u = m \ominus F(j_T)$. Since we chose F and the group operation in M to be efficiently computable, $d_s \lesssim p$ and $d_s, \widetilde{d}_T$ smooth, we get that the algorithm takes time $t_{\mathrm{com}} \in \mathrm{poly}(\log p)$. $\square$

Lemma 5.14. *The maximal number of operations t_{cv} for algorithm* `ComVrfy` *is a small constant.*

Proof. The algorithm has to complete three tasks. First it has to check if E_s is an elliptic curve. To do that, it suffices to check that the discriminant is non-zero. For curves in short Weierstraß form $E\colon y^2 = x^3 + Ax + B$ this is just $4A^3 \neq -27B^2$. To check if K_T is a point on E_s it can simply compute if K_T satisfies the curve equation. Finally, membership testing for $u \in M$ is efficient by definition of M. For $M = \mathbb{Z}/N\mathbb{Z}$ this means checking if u is an integer (and if $0 \leq u < N$). So all this can be done in very few operations and their number is independent of the size of d_s, p and κ. $\square$

Lemma 5.15. *The decommitment verification algorithm* `DecVrfy` *takes time $t_{\mathrm{dv}} \in \mathrm{poly}(\log p)$.*

Proof. The decommitment verification has the same steps as the commitment. There are only two differences: Firstly, it gets s, t from π_{dec} instead of choosing them and hence does not need to try again for bad choices of t. And secondly, it has to compare the E_s and K_T it computes to the ones in the commitment and m to the decommitment. Since these steps are computationally insignificant we get that the algorithm also takes time $t_{\text{dv}} \in \text{poly}(\log p)$. $\square$

Note that the running times of `Com`, `ComVrfy` and `DecVrfy` are not dominated by d_T. Even for low security levels like $\kappa = 128$ we get that $\log p \ll p^{1/8} \lesssim d_s^{1/4}$. Since t_{fd} can be almost as large as $\min\{d_s^{1/4}, p^{1/4}\}$ and $d_s \lesssim p$, the previous Lemmas 5.14 and 5.15 show that we can choose $\log d_T < \text{eval}(d_T) < t_{\text{fd}}$ such that $t_{\text{com}}, t_{\text{cv}}, t_{\text{dv}} \ll t_{\text{fd}}$. This gives us the following theorem:

Theorem 5.16. *SIGNITC is perfectly practical under Assumptions 5.9 and 5.10.*

Conclusion

We showed that SIGNITC is a perfectly practical NITC that satisfies hiding and perfect binding. It is the first NITC without repeated squaring or black box algorithms, it needs no trusted setup and all subroutines have already been implemented for other cryptosystems. Since it uses only isogeny-based cryptography, it is presumably quantum secure. Since repeated squaring might not be a good candidate for creating a delay anymore, this could also be an interesting starting point for isogeny-based delay in other settings. We leave it as open research to implement this protocol to get some benchmarks for (relative) real world timings and to find (computational) optimizations.

Acknowledgments

The author would like to thank Antonio Sanso for pointing out the concept of NITC schemes, Valerie Gilchrist and Lorenz Panny for their comments on a previous version of this article, and Jens Zumbrägel for his support during the work on this article.

References

[1] Knud Ahrens and Jens Zumbrägel. DEFEND: Towards verifiable delay functions from endomorphism rings. Cryptology ePrint Archive, Paper 2023/1537, 2023. URL `https://eprint.iacr.org/2023/1537`.

[2] Miguel Ambrona, Marc Beunardeau, and Raphaël R. Toledo. Timed commitments revisited. Cryptology ePrint Archive, Paper 2023/977, 2023. URL `https://eprint.iacr.org/2023/977`.

[3] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. In Steven D. Galbraith, editor, *Proceedings of the Fourteenth Algorithmic Number Theory Symposium*, pages 39–55, Berkeley, 2020. Mathematical Sciences Publishers. doi: 10.2140/obs.2020.4.39.

[4] Alex Biryukov, Ben Fisch, Gottfried Herold, Dmitry Khovratovich, Gaëtan Leurent, María Naya-Plasencia, and Benjamin Wesolowski. Cryptanalysis of algebraic verifiable delay functions. Cryptology ePrint Archive, Paper 2024/873, 2024. URL `https://eprint.iacr.org/2024/873`.

[5] Dan Boneh and Moni Naor. Timed commitments. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, pages 236–254, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. doi: 10.1007/3-540-44598-6_15.

[6] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 757–788, Cham, 2018. Springer International Publishing. doi: 10.1007/978-3-319-96884-1_25.

[7] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 395–427, Cham, 2018. Springer International Publishing. doi: 10.1007/978-3-030-03332-3_15.

[8] Jorge Chavez-Saab, Francisco Rodríguez-Henríquez, and Mehdi Tibouchi. Verifiable isogeny walks: Towards an isogeny-based postquantum VDF. In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography*, pages 441–460, Cham, 2022. Springer International Publishing. doi: 10.1007/978-3-030-99277-4_21.

[9] Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign algorithm specifications and supporting documentation. Project Homepage, 2023. `https://sqisign.org/spec/sqisign-20230601.pdf`.

[10] Peter Chvojka and Tibor Jager. Simple, fast, efficient, and tightly-secure non-malleable non-interactive timed commitments. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography – PKC 2023*, pages 500–529, Cham, 2023. Springer Nature Switzerland. doi: 10.1007/978-3-031-31368-4_18.

[11] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019*, pages 248–277, Cham, 2019. Springer International Publishing. doi: 10.1007/978-3-030-34578-5_10.

[12] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 64–93, Cham, 2020. Springer International Publishing. doi: 10.1007/978-3-030-64837-4_3.

[13] Thomas Decru, Luciano Maino, and Antonio Sanso. Towards a quantum-resistant weak verifiable delay function. In Abdelrahaman Aly and Mehdi Tibouchi, editors, *Progress in Cryptology – LATINCRYPT 2023*, pages 149–168, Cham, 2023. Springer Nature Switzerland. doi: 10.1007/978-3-031-44469-2_8.

[14] Max Deuring. Die Typen der Multiplikatorenringe elliptischer funktionenkörper. *Abh. Math. Sem. Hansischen Univ.*, 14:197–272, 1941. doi: 10.1007/BF02940746.

[15] Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 329–368, Cham, 2018. Springer International Publishing. doi: 10.1007/978-3-319-78372-7_11.

[16] Luca De Feo. Mathematics of isogeny based cryptography. Preprint, 2017. URL `https://arxiv.org/abs/1711.04062`.

[17] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. Association for Computing Machinery. doi: 10.1145/237814.237866.

[18] D. Jao et al. Supersingular isogeny key encapsulation, 2020. `https://sike.org/files/SIDH-spec.pdf`.

[19] Jonathan Katz, Julian Loss, and Jiayu Xu. On the security of time-lock puzzles and timed commitments. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography*, pages 390–413, Cham, 2020. Springer International Publishing. doi: 10.1007/978-3-030-64381-2_14.

[20] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS J. Comput. Math.*, 17:418–432, 2014. doi: 10.1112/S1461157014000151.

[21] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, USA, 1996.

[22] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer New York, 1986. doi: 10.1007/978-1-4757-1920-8.

[23] Sri Aravinda Krishnan Thyagarajan, Guilhem Castagnos, Fabian Laguillaumie, and Giulio Malavolta. Efficient CCA timed commitments in class groups. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 2663–2684, New York, NY, USA, 2021. Association for Computing Machinery. doi: 10.1145/3460120.3484773.

[24] John Voight. *Quaternion algebras*, volume 288 of *Graduate Texts in Mathematics*. Springer Cham, 2021. doi: 10.1007/978-3-030-56694-4.

[25] Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences, Série A*, 273, N°4:238–241, 1971. URL `https://gallica.bnf.fr/ark:/12148/bpt6k56191248`.

[26] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1100–1111, 2022. doi: 10.1109/FOCS52979.2021.00109.