The Security of All Private-key Bits in Isogeny-based Schemes

Barak Shani
University of Pennsylvania

Abstract

We study the computational hardness of recovering single bits of the private key in the supersingular isogeny Diffie–Hellman (SIDH) key exchange and similar schemes. Our objective is to give a polynomial-time reduction between the problem of computing the private key in SIDH to the problem of computing any of its bits. The parties in the SIDH protocol work over elliptic curve torsion groups of different order N. Our results depend on the parity of N. Our main result shows that if N is odd, then each of the top and lower $O(\log\log N)$ bits of the private key is as hard to compute, with any noticeable advantage, as the entire key. A similar, but conditional, result holds for each of the middle bits. This condition can be checked, and heuristically holds almost always. The case of even N is a bit more challenging. We give several results, one of which is similar to the result for an odd N, under the assumption that one always succeeds to recover the designated bit. To achieve these results we extend the solution to the chosen-multiplier hidden number problem, for domains of a prime-power order, by studying the Fourier coefficients of single-bit functions over these domains.

1. INTRODUCTION

Isogeny-based cryptography is a relatively new area in public-key cryptography. It is based on the problem of computing isogenies between a given pair of isogenous elliptic curves. There is no (publicly) known polynomial-time quantum (or classical) algorithm for this problem, as opposed to factoring and computing discrete logarithms, and therefore it may serve in the future as a fundamental problem to build cryptographic schemes upon. See [5, 10] for introduction to isogeny-based cryptography.

The use of isogenies for public-key cryptography was first considered in the unpublished work of Couveignes [4] and was later rediscovered by Rostovtsev and Stolbunov [18] (see also Stolbunov [21]), who proposed a key exchange protocol with ordinary elliptic curves. Subsequently Jao and De Feo [13] proposed a key exchange protocol with supersingular elliptic curves, that does not suffer from some weaknesses in the ordinary cryptosystem.

The main isogeny-based scheme so far is Jao–De Feo's Diffie–Hellman-type key exchange, known as the *supersingular isogeny Diffie–Hellman* (SIDH). Other schemes [6, 9, 25] build upon it. As such, it is often convenient to introduce this area with a comparison to the original Diffie–Hellman key exchange (see for example the introduction of [23]). We follow this analogy. In the rest of the paper we let H

be some group where its operation is written multiplicatively. The analogue to exponentiation in H is applying an isogeny on the elliptic curve E, and the analogue to computing discrete logarithms in H is computing an explicit isogeny $\phi : E \to E'$ (given E, E').

One of the important aspects in the security of public-key cryptosystems is the bit security of the private key (a comprehensive overview, including the relation to pseudorandom functions, is given in the survey [11]). The study of bit security aims to determine whether the public key reveals partial information about the private key, where the main interest is in the bits of the private key. Subsequently, it may affect the choice of security parameters in implementations of the cryptosystems. Ideally one would like to know that each bit of the private key is hard to predict, or at least have a precise classification of which bits are easy to compute and which are hard.

This paper studies the bit security of the private key in SIDH and in similar isogeny-based cryptosystems. We remark that other security aspects of SIDH were investigated in [8, 22, 17]. The setup in these cryptosystems consists of a supersingular elliptic curve E and two points of a known smooth order $P, Q \in E[\ell^e]$, for some prime ℓ. Our main question of interest is the following:

> *Given Alice's public key E/G, where $G = \langle [a]P + [b]Q \rangle$, how hard is it to retrieve bits of the private key a, b?*

A similar question can be asked when also two auxiliary points $\phi(P_1), \phi(Q_1) \in E/G$ are given, as in SIDH (see Section 2), where $\phi : E \to E/G$ is Alice's secret isogeny.

For this problem to be well defined, we assume the uniqueness (up to automorphism) of the isogeny $\phi : E \to E/G$ of degree dividing ℓ^e; this is typically the case, and it holds when the auxiliary points are given (see [23, Lemma 3.2]). The equivalent problem for the discrete logarithm problem (DLP) is: given $g \in H$ and g^a, how hard is it to retrieve bits of a?

One issue arises from the fact that G is not uniquely defined.[1] For every $c \in \mathbb{Z}_{\ell^e}^*$ we have that $G = \langle [ca]P + [cb]Q \rangle$. Thus there are many pairs a', b' that define G, i.e. $G = \langle [a']P + [b']Q \rangle$, and there is no benefit in determining the exact values a, b chosen by Alice: recovering any such pair breaks the cryptosystem. In particular, if a, b are chosen at random, then with high probability at least one of these values, say a, is invertible (that is $a \in \mathbb{Z}_{\ell^e}^*$) and so for $c = a^{-1}$ we get that $G = \langle P + [a^{-1}b]Q \rangle$. Therefore with high probability one can determine all the bits of one of the coefficients. In particular we see that even though the private key may consist of two randomly chosen n-bit integers, they provide at most $n + 1$ bits of security.[2] In fact it is suggested in the literature [6, 3] to set either $a = 1$ or $b = 1$ to

[1] We remark that a similar problem arises in the study of bit security in LUC and XTR cryptosystems, and is a major obstacle in getting results for XTR (see [15] for details).

[2] It is suggested to make sure that at least one of the coefficients a, b is invertible to have (at most) n bits of security (in some cases the order of the group G can be computed and so if both coefficients are not invertible it can be detected and exploited to have a smaller search space for the coefficients; see Section 4.3.1). The extra bit comes from the freedom to choose the invertible coefficient.

speed-up the implementation.

Contribution: Suppose without loss of generality that $G = \langle P + [b']Q \rangle$. We want to know how hard it is to compute, or predict, a single bit of b', the coefficient of Q, with a non-negligible advantage (over the guessing strategy). A standard approach to showing that it is infeasible to predict bits of b' is to show that predicting a single bit of b' leads to an algorithm that computes b' (with some noticeable probability). Since $G = \langle [r]P + [rb']Q \rangle$ for any $r \in \mathbb{Z}_{\ell^e}^*$, the ability to predict a bit of the coefficient of Q allows one to predict a bit of rb' for any $r \in \mathbb{Z}_{\ell^e}^*$. Thus, we get (a variant of) the *chosen-multiplier hidden number problem*, which is a very useful tool in the study of bit security. However, here the problem takes place in a domain of highly composite order, $\mathbb{Z}_{\ell^e}$, a case that has not been previously addressed, to the best of our knowledge.

In Theorem 3 we provide a solution to the chosen-multiplier hidden number problem with single-bit functions in $\mathbb{Z}_{\ell^e}$, for $\ell \neq 2$. We divide the solution into three cases: the top $\log\log \ell^e$ bits (where the case $\ell = 2$ also holds), the lower $\log\log \ell^e$ bits, and the rest of the bits – the middle bits – where the solution is conditional. This result may be of independent interest.

In Section 4.1 we consider the case $\ell \neq 2$. Building on the former result we prove that computing any of the top or lower $\log\log \ell^e$ bits of b' with non-negligible advantage is as hard as computing G (or equivalently the entire value b'); the top-bits result also holds for $\ell = 2$. For the middle bits, we give a general condition for a similar result to hold. The condition, which can be checked, depends on the particular domain size ℓ^e, and heuristically holds with overwhelming probability. These results hold in a model where an oracle takes a pair of points P', Q', and returns a bit of the (minimal positive) coefficient s of Q' such that $G = \langle P' + [s]Q' \rangle$ (if such s exists).

In Section 4.2 we consider the case $\ell = 2$. For this case we consider a more restrictive model that does not enable to specify the generating points P', Q' of $E[\ell^e]$ as in the previous case, instead they are given upon specifying the torsion group (which can be determined by the isogeny degree). We show that recovering the least significant bit of both coefficients is as hard as recovering the entire group G. In addition, in a similar model to the one in Section 4.1, we show that computing – with probability 1 – a single bit of b' is as hard as computing b'.

These results do not exploit the auxiliary points in the SIDH public key. We further show how to use these points to randomise these reductions and to learn bits of the private key when the group G is not generated properly.

2. ISOGENY-BASED CRYPTOGRAPHY

We state some facts on isogenies between elliptic curves. For general background on elliptic curves see Silverman [20] and for mathematical background on isogeny-based cryptography see any of [5, 6, 8, 9,

10, 13, 17, 22, 23].

An *isogeny* between two elliptic curves E, E' is a non-constant morphism $\phi : E \to E'$ that maps the identity in E to the identity in E'. An isogeny is a group homomorphism, and it is separable if its extension to function fields is separable. An isogeny is defined by its kernel: for every finite subgroup $G \subseteq E$, there is a unique curve (up to isomorphism), denoted by E/G, and a separable isogeny $\phi : E \to E/G$ such that $\ker \phi = G$. The degree of ϕ satisfies $\deg \phi = \# \ker \phi$. An isogeny of a composite degree can be written as a composition of smaller-degree isogenies: if $\deg \phi = n_1 n_2$ then $\phi = \phi_2 \circ \phi_1$ where $\deg \phi_1 = n_1$ and $\deg \phi_2 = n_2$.

An elliptic curve E over a finite field $\mathbb{F}_{p^k}$, for a prime p, is said to be supersingular if $|E(\mathbb{F}_{p^k})| \equiv 1 \pmod{p}$. Every supersingular curve is isomorphic to a curve over $\mathbb{F}_{p^2}$. The j-invariant of a curve is an invariant of its isomorphism class, thus all isomorphic curves have the same j-invariant.

Given E and a subgroup $G \subseteq E$, Vélu's formulas [24] can be used to compute the curve E/G and the isogeny $\phi : E \to E/G$ in time $O(|G|)$. Specifically, if the order of G is smooth one can efficiently compute the sequence of small-degree isogenies in the composition, and so also ϕ and E/G, in time polylogarithmic in $|G|$, as we now show for a specific case of interest.

Let $R \in E$ be of order ℓ^e, then $G := \langle R \rangle \subseteq E$ is a cyclic group of order ℓ^e. The isogeny with kernel G factors to a chain of isogenies, each of degree ℓ. Set $E_0 = E$ and $R_0 = R$, and define for $0 \leq i < e$

$$E_{i+1} = E_i / \langle \ell^{e-i-1} R_i \rangle \,, \qquad \phi_{i+1} = E_i \to E_{i+1} \,, \qquad R_{i+1} = \phi_i(R_i) \,.$$

Then $E/\langle R \rangle = E_e$ and $\phi = \phi_e \circ \cdots \circ \phi_1$.

On the other hand, given two curves E, E', computing an isogeny between them is believed to be a hard problem, known as the (general) *isogeny problem*. The hardness of this problem has been used as a basis for several cryptosystems, most notably using supersingular elliptic curves. SIDH key exchange is the core construction; encryption, identification and signature schemes [6, 9, 25] build upon it. Thus, our study of the bit security of the private key in SIDH applies to these schemes as well.

We now turn to describe SIDH. Note that the security of the cryptosystem relies a stronger problem, known as the *explicit isogeny problem*: computing an isogeny of a given degree. On the other hand, in addition to E, E' two auxiliary points are also being published.

Key Exchange Protocol Alice and Bob agree on a prime of the form $p = \ell_A^n \ell_B^m f \pm 1$, where ℓ_A, ℓ_B are small primes such that $\ell_A^n \approx \ell_B^m$ and f is small, a supersingular elliptic curve $E(\mathbb{F}_{p^2})$ with some specific representation and two pairs of independent generators $P_A, Q_A \in E[\ell_A^n]$ and $P_B, Q_B \in E[\ell_B^m]$: the group $\langle P_A, Q_A \rangle$ generated by P_A and Q_A has (full) order ℓ_A^{2n}, and similarly for $\langle P_B, Q_B \rangle$.

The supersingular isogeny Diffie–Hellman key exchange protocol proceeds as follows:

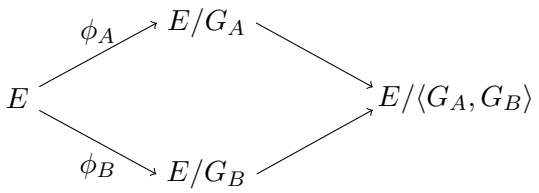
1. Alice chooses random integers $0 \leq a_1, a_2 < \ell_A^n$, not both divisible by ℓ_A, computes $G_A := \langle [a_1]P_A + [a_2]Q_A \rangle$ and an isogeny ϕ_A from E with kernel G_A. She then obtains the curve $E_A = E/G_A$ and the points $\phi_A(P_B)$, $\phi_A(Q_B)$ on it, and sends this triple to Bob.

2. Bob chooses random integers $0 \leq b_1, b_2 < \ell_B^m$, not both divisible by ℓ_B, computes $G_B := \langle [b_1]P_B + [b_2]Q_B \rangle$ and an isogeny ϕ_B from E with kernel G_B. He then obtains the curve $E_B = E/G_B$ and the points $\phi_B(P_A)$, $\phi_B(Q_A)$ on it, and sends this triple to Alice.

3. Alice computes $\phi_B(G_A) = \langle \phi_B([a_1]P_A + [a_2]Q_A) \rangle = \langle [a_1]\phi_B(P_A) + [a_2]\phi_B(Q_A) \rangle$ and an isogeny from E_B with kernel $\phi_B(G_A)$. She then obtains the curve $E_B/\langle \phi_B(G_A) \rangle = E/\langle G_A, G_B \rangle$ and computes its j-invariant.

4. Bob computes $\phi_A(G_B) = \langle \phi_A([b_1]P_B + [b_2]Q_B) \rangle = \langle [b_1]\phi_A(P_B) + [b_2]\phi_A(Q_B) \rangle$ and an isogeny from E_A with kernel $\phi_A(G_B)$. He then obtains the curve $E_A/\langle \phi_A(G_B) \rangle = E/\langle G_A, G_B \rangle$ and computes its j-invariant.

Both parties obtain the curve $E_{AB} := E/\langle G_A, G_B \rangle$. It is not guaranteed that they use the same representation of the curve, so using the j-invariant guarantees they share the same key. The protocol revolves around the following commutative diagram:



Our analysis makes use of the following facts (see [3, Section 2] or [6, Section 3]). Let $\ell \in \{\ell_A, \ell_B\}$ and let $e \in \{n, m\}$ the corresponding exponent. We write $\mathbb{Z}_{\ell^e}$ for $\mathbb{Z}/\ell^e\mathbb{Z}$. Since ℓ is coprime to p we have $E[\ell^e] \simeq \mathbb{Z}_{\ell^e} \times \mathbb{Z}_{\ell^e}$. Let $P, Q \in E[\ell^e]$ be two independent generators (a basis). A point $[a]P + [b]Q$ is of full order ℓ^e if and only if (at least) one of a, b is not divisible by ℓ. There are $\ell^{e-1}(\ell + 1)$ distinct cyclic subgroups of order ℓ^e and $\ell^{2e-2}(\ell^2 - 1)$ points of full order ℓ^e.

Thus, the probability that a point in $E[\ell^e]$, drawn uniformly at random (for example $[a]P + [b]Q$ for randomly chosen $a, b \in \mathbb{Z}_{\ell^e}$), is of full order is at least $3/4$. Moreover, the probability that two uniformly sampled points P', Q' form a basis of $E[\ell^e]$ is at least $3/8$. Indeed, P' is of full order with probability $1 - 1/\ell^2$; suppose that P', P'' is a basis, then for $Q' = [a]P' + [b]P''$, the pair P', Q' is a basis if and only if b is invertible, which holds with probability $1 - 1/\ell$.

3. THE HIDDEN NUMBER PROBLEM AND BIT SECURITY

This section introduces the *hidden number problem* and its application in the study of bit security. We remark that this is not the *isogeny hidden number problem* introduced in [8]. Previous results on the hidden number problem, which are presented in this section, involve discrete Fourier analysis in the group $\mathbb{Z}_N$. We refer to [7] for relevant background, however this background is not necessary for the rest of the paper. We represent the group $\mathbb{Z}_N$ by the set of integers $\{0, \ldots, N-1\}$, and define $|a| = \min\{a, N-a\}$ for $a \in \mathbb{Z}_N$. The i-th bit function bit_i is defined by $\mathrm{bit}_i(x) = -1^{x_i}$ for an integer $x = \sum_{i=0}^{n} x_i 2^i$, with $x_i \in \{0,1\}$. We set $\mathrm{LSB} := \mathrm{bit}_0$.

The hidden number problem was introduced by Boneh and Venkatesan in the study of the hardness of computing bits of Diffie–Hellman keys [2]. This problem has been proven to be very fruitful, with applications in different areas (see the survey [19]). In particular a variant of this problem can be used to study the hardness of computing bits of various secret values. Håstad and Näslund seem to be the first to make this connection explicit, in their study of the RSA and the discrete logarithm problems [12]. We define this variant, called *chosen-multiplier hidden number problem*, as follows.

Definition 1 (CM-HNP)**.** Let $\mathrm{bit}_i : \mathbb{Z}_N \to \{-1, 1\}$ be the i-th bit function and let $s \in \mathbb{Z}_N$ be unknown ("hidden"). Recover s given query access to a predictor P satisfying

$$\Pr_{x \in \mathbb{Z}_N}[P(x) = \mathrm{bit}_i(sx)] \geq \frac{1 + \beta_i(N)}{2} + \epsilon\,,$$

where $\epsilon > 0$ and $\beta_i(N)$ denotes the bias[3] of bit_i, i.e. $\beta_i(N)$ satisfies $\Pr_{x \in \mathbb{Z}_N}[\mathrm{bit}_i(x) = 1] = \frac{1+\beta_i(N)}{2}$.

Håstad and Näslund solved this problem for every i when N is prime or an RSA modulus.[4] Their technique is adaptive (requires choosing queries with respect to the outcome of previous queries) and is based on complicated manipulation of bits. A more general solution is given in the work of Akavia, Goldwasser and Safra [1]. This solution is non-adaptive and based on tools from Fourier analysis (the work [7] surveys these tools and their applications). It is based on the fact that the function bit_i has (a few) large Fourier coefficients and that for $s \in \mathbb{Z}_N^*$ the function $x \mapsto \mathrm{bit}_i(sx)$ permutes the Fourier coefficients of bit_i by the scaler s^{-1} (one then computes only the set of large coefficients for both functions and finds the permutation; see Theorem 2 below).

Theorem 1 ([1, Theorem 6])**.** *Let $f : \mathbb{Z}_N \to \{-1, 1\}$ and denote by $\widehat{f}$ its Fourier transform. Given query access to f, a threshold $\tau > 0$ and $\delta > 0$, there exists an algorithm that outputs a list L of size at most $1/\tau$ such that L contains all $a \in \mathbb{Z}_N$ such that $|\widehat{f}(a)|^2 > \tau$ (and no a with $|\widehat{f}(a)|^2 \leq \tau/2$) with probability at least $1 - \delta$. The algorithm runs in polynomial time in $\log N$, $1/\tau$ and $\log(1/\delta)$.*

[3] For some properties of the bias see [12, Section 2].

[4] More precisely, their applications are for such domains. Their algorithm either returns s or a non trivial factor of N.

It is shown by Morillo and Ràfols [16] that for all i there exists $a \in \mathbb{Z}_N$ such that bit_i satisfies $|\widehat{\text{bit}_i}(a)|^2 > \tau$ for $\tau^{-1} = O(\log N)$ (see also [14] or [7, Section 4.1]).[5] In other words, each single-bit function has a significantly large Fourier coefficient. Using this property of bit_i, an application of [1, Theorem 2] gives a solution to CM-HNP. We restate this result and sketch its proof.

Theorem 2. *Let p be prime, let $\epsilon, \delta > 0$, let s be an invertible element in $\mathbb{Z}_p$, i.e. $s \in \mathbb{Z}_p^*$, and consider the chosen-multiplier hidden number problem (CM-HNP) with "hidden number" s and some integer $0 \leq i \leq \log p$. There exists an algorithm for CM-HNP that returns a list of size $O(\log p/\epsilon)$ that contains s with probability $1 - \delta$. The algorithm runs in polynomial time in $\log p$, $1/\epsilon$ and $\log(1/\delta)$.*

Proof sketch. First suppose that $\epsilon = 1/2 - \beta_i(p)/2$, that is $P(x) = \text{bit}_i(sx)$ for all x. Since s is invertible, the "scaling property" of the Fourier transform asserts that $\widehat{P}(a) = \widehat{\text{bit}_i}(as^{-1})$. In other words, P and bit_i have the same set of Fourier coefficients, ordered differently by a scale of s. To solve CM-HNP it is sufficient to find this scaling factor. As we now explain, this is done by locating only the 'large' Fourier coefficients (for both functions) and pairing them until s is found.

Apply the algorithm from Theorem 1 with the same threshold τ on P and bit_i to obtain the (short) lists L_P, L_i, respectively. The value τ can be chosen experimentally (in polynomial time in $\log p$ due to the result of [16] mentioned above) until the returned lists are non-empty. By the scaling property, for every $a \in L_P$ there exists $b \in L_i$ such that $b = as^{-1}$. The secret s can be thus recovered efficiently, as $s = ab^{-1}$ and $a, b \in \mathbb{Z}_p$ are invertible (non zero) [6].

Consider now $\epsilon < 1/2 - \beta_i(p)/2$. The Fourier coefficients of P may differ from those of bit_i, but the difference is bounded by a constant proportional to ϵ, see [7, Section 3.3] for the exact details. One should therefore lower the threshold τ proportionally to ϵ, and the same solution holds. This algorithm runs in polynomial time in $\log p$ and ϵ^{-1}. ∎

We remark that Theorem 2 holds in any domain $\mathbb{Z}_N$, as long as a significant Fourier coefficient of bit_i, which is guaranteed to exist by [16], is at an invertible element. That is, for $a \in \mathbb{Z}_N^*$ and $\tau^{-1} = O(\log N)$ we have $|\widehat{\text{bit}_i}(a)|^2 > \tau$. For a prime modulus this is trivial, and for an RSA modulus N, if the large coefficient is not coprime to N, then after finding it (using the algorithm in Theorem 1) one can factor N, similar to the result of Håstad and Näslund mentioned above. The following section shows when this condition holds for domains of order $N = \ell^e$.

3.1 Highly composite domains

We now present an extension to Theorem 2 for domains that are highly composite. For our applications we are concerned with $N = \ell^e$ for some (small) prime ℓ. First, we need a characterisation of the Fourier

[5] The result is even greater, the functions are *concentrated* (see [16, 14]).

[6] An exception is the most-significant-bit function in the case that the bias is too large, i.e. the function is almost constant, therefore $a = 0$. In this case the advantage ϵ is negligible.

coefficients of the function bit_i, when defined over $\mathbb{Z}_{\ell^e}$. The case $\ell = 2$ was considered in [14], where the following result is given.

Lemma 1 ([14, Lemma 6.2])**.** *Let $k \in \mathbb{N}$ and $0 \leq i < k$. Define $bit_i : \mathbb{Z}_{2^k} \to \{-1, 1\}$ by $bit_i(x) = (-1)^{x_i}$ where $x = \sum_{j=0}^{k-1} x_j 2^j$ and $x_j \in \{0, 1\}$. Let $\alpha \in \mathbb{Z}_{2^k}$. Then $\widehat{bit_i}(\alpha) = 0$ unless α is an odd multiple of 2^{k-i-1}, in which case $|\widehat{bit_i}(\alpha)| = O(2^{k-i}/|\alpha|)$.*

This result can be summarised as follows: the Fourier coefficient $\widehat{\text{bit}_i}(\alpha)$ is zero when α is not an odd multiple of 2^{k-i-1}, and when it is non-zero it has magnitude bounded by $2^{k-i}/|\alpha|$.

This result gives an upper bound on the magnitude of the coefficients of bit_i. From Parseval's identity, since $\|\text{bit}_i\| = 1$, we can also obtain a lower bound for these coefficients. In fact, from [7, Claim 4.1] (combined with the proof of the Lemma 1), it can be shown that $|\widehat{\text{bit}_i}(\alpha)| \approx 2^{k-i}/\pi|\alpha|$ for sufficiently small α satisfying the above (i.e. for the largest coefficients).

To illustrate this result, for $i = 0$, i.e. the least significant bit, there is one non-zero coefficient at $\alpha = 2^{k-1}$ of magnitude 1 (indeed, the corresponding character alternates between 1 and -1); for $i = 1$ the non-zero coefficients are at $\alpha = \pm 2^{k-2}$, of magnitude $1/\sqrt{2}$; for $i = k - 1$, i.e. the most significant bit, the large coefficients are at odd values close to 0: the largest is at $\alpha = \pm 1$, and the size decreases with $1/|\alpha|$.

We see that except for the case $i = k - 1$ (the most significant bit), all large coefficients of bit_i are at even values. Therefore, one cannot apply the algorithm in Theorem 2 in domains $\mathbb{Z}_{2^e}$. We now turn to domains $\mathbb{Z}_{\ell^e}$, where ℓ is an odd prime. The Fourier coefficients of bit_i, over these domains, have not been previously studied in the literature, to the best of our knowledge.

Recall that our goal is to be able to apply the algorithm from Theorem 2, therefore we need to show that bit_i has a large coefficient at an invertible value α. We separate our analysis into three cases: the lower $O(\log \log N)$ bits, the top $O(\log \log N)$ bits and the middle bits. For the lower bits, we show that bit_i has a large coefficient at an invertible value. For the middle bits, we give a characterisation when bit_i has such coefficients. For the top bits, we show how to solve the hidden number problem, even if bit_i does not have such a coefficient.

We make use of the "changes in domain" of Laity and Shani [14]. The main idea is that we can use the function bit_i over $\mathbb{Z}_{2^k}$ to study the Fourier coefficients of bit_i restricted to $\mathbb{Z}_{\ell^e}$, where $2^{k-1} < \ell^e < 2^k$. Thus, the classification of the large Fourier coefficients in Lemma 1 becomes useful even in other domains. In particular, we show the following. Suppose that the i-th bit function over $\mathbb{Z}_{2^k}$ has a large Fourier coefficient at the value a. For the lower- and the middle-bit cases, we show that the i-th bit function over $\mathbb{Z}_{\ell^e}$ has a large Fourier coefficient at the value $\lfloor \frac{\ell^e}{2^k} a \rceil$, the closest integer to $\frac{\ell^e}{2^k} a$. Moreover, the magnitude of the latter coefficient gets closer to the magnitude of the former coefficient, the closer $\frac{\ell^e}{2^k} a$ is to an integer. In addition, there is a large coefficient at the second closest integer to

$\frac{\ell^e}{2^k}a$, if this value is not too close to an integer, i.e. when $|\frac{\ell^e}{2^k}a - \lfloor \frac{\ell^e}{2^k}a \rceil|^{-1} < O(\log N)$.

Claim 1. Let ℓ be an odd prime, let e be an integer and let $N = \ell^e$, then for every $0 \leq i \leq O(\log \log N)$ there exists $z \in \mathbb{Z}_N^*$ and $\tau^{-1} = O(\log N)$ such that $|\widehat{\text{bit}}_i(z)|^2 > \tau$.

Proof. We show that bit_i has two consecutive large coefficients (in fact it is the case for all large coefficients) and therefore one of them has to be at an invertible element. Let k such that $2^{k-1} < \ell^e < 2^k$. Consider $\text{bit}_i : \mathbb{Z}_{\ell^e} \to \{-1, 1\}$ as the restriction of the i-th bit function $\text{bit}'_i : \mathbb{Z}_{2^k} \to \{-1, 1\}$ to $\mathbb{Z}_{\ell^e}$. By Lemma 1 the significantly large coefficients of the latter function are exponentially far apart. Therefore, we can apply the results from [14, Section 3] even though bit'_i has more than one non-zero coefficient, as the weight of all of the coefficients of bit'_i, except for at most one, on a specific coefficient of bit_i is negligible (see also [14, Section 5]).

Suppose that $\widehat{\text{bit}'}_i(a)$ is a large Fourier coefficient. It is sufficient to assume that $0 < a \leq 2^{k-1}$, since $|\widehat{\text{bit}'}_i(-a)| = |\widehat{\text{bit}'}_i(a)|$, as the function is real valued. From Lemma 1, $a = j2^{k-i-1}$ for odd j. Let $z = \lfloor \frac{\ell^e}{2^k}a \rceil = \lfloor \frac{\ell^e}{2^{i+1}}j \rceil$. Lemma 3.1 in [14] shows that $\widehat{\text{bit}}_i(z)$ is a large Fourier coefficient. If $z \in \mathbb{Z}_N^*$, we are done. Otherwise, let z' be the second closest integer to $\frac{\ell^e}{2^k}a$, and define $r = |\frac{\ell^e}{2^k}a - z'|$, then $1/2 \leq r < 1$. Moreover, r is a multiple of $1/2^{i+1}$ since $\frac{\ell^e}{2^k}a = \frac{\ell^e}{2^{i+1}}j$. Combining with [14, Remark 2], we get that $|\widehat{\text{bit}}_i(z')| \geq 2|\widehat{\text{bit}'}_i(a)|/(\pi 2^{i+1}r) \geq |\widehat{\text{bit}'}_i(a)|/O(\log N)$. Hence, $\widehat{\text{bit}}_i(z')$ is a large Fourier coefficient, and either $z' \equiv 1 \pmod{\ell}$ or $z' \equiv -1 \pmod{\ell}$, so $z' \in \mathbb{Z}_N^*$. ∎

Claim 2. Let ℓ be an odd prime, let e be an integer and let $N = \ell^e$. For $O(\log \log N) < i < \log N - O(\log \log N)$, denote $z = \lfloor \frac{\ell^e}{2^{i+1}} \rceil$ and $r = |\frac{\ell^e}{2^{i+1}} - z|$, the distance between $\frac{\ell^e}{2^{i+1}}$ to its closest integer. Then there exists $z' \in \mathbb{Z}_N^*$ and $\tau^{-1} = O(\log N)$ such that $|\widehat{\text{bit}}_i(z')|^2 > \tau$ if and only if $r^{-1} = O(\log N)$ or $z \in \mathbb{Z}_N^*$.

Proof. We rely on the ideas from the previous claim. As before we can apply the results from [14, Section 3], since the Fourier coefficients of bit'_i are sufficiently far apart. By Lemma 1 the coefficient $\widehat{\text{bit}'}_i(2^{k-i-1})$ is large, thus for $\lfloor \frac{\ell^e}{2^k}2^{k-i-1} \rceil = \lfloor \frac{\ell^e}{2^{i+1}} \rceil = z$, the coefficient $\widehat{\text{bit}}_i(z)$ is large. If $z \in \mathbb{Z}_N^*$, we are done. Otherwise, if $r^{-1} = O(\log N)$, then as before, using the first inequality in [14, Remark 2], we get that $\widehat{\text{bit}}_i(z')$ is a large Fourier coefficient, where z' is the second closest integer to $\frac{\ell^e}{2^{i+1}}$.

On the other hand, suppose that $r^{-1} > O(\log N)$ (and z non-invertible), then from the second inequality in [14, Remark 2] we get that $|\widehat{\text{bit}}_i(z')| \leq |\widehat{\text{bit}'}_i(a)|\frac{\pi}{2}\frac{r}{(1-r)} < \pi|\widehat{\text{bit}'}_i(a)|/O(\log N)$, so it is not sufficiently large. Moreover, this is the case for all z' in a short interval around z. Since all the large coefficients of bit'_i are in small multiples of 2^{k-i-1}, we get the same conclusions by repeating the argument. That is, for a small odd j we get that $\widehat{\text{bit}}_i(jz)$ is a large coefficient, and jz is non-invertible since z is non-invertible, and that any other coefficient in a small interval around jz is small, since $r' = |\frac{\ell^e}{2^{i+1}}j - jz|$ satisfies $r'^{-1} > O(\log N)$. ∎

An example of the previous claim can be seen by considering $N = 3^9$ and $i = 4$. We have $\frac{\ell^e}{2^{i+1}} = \frac{19683}{32} \approx 615.09$ (so $r \approx 0.9$), and the only coefficients of bit_i that are larger than 0.0805 are at the values $\pm 615, \pm 3 \cdot 615, \pm 5 \cdot 615$, which are all non-invertible in $\mathbb{Z}_N$.[7] We remark that since there are at most $\log N$ values for i (there are $\log N$ bits), the condition $r^{-1} > O(\log N)$ is expected to hold with negligible probability. Hence, heuristically, for almost all N we expect bit_i to have a large coefficient at an invertible value for all $O(\log \log N) < i < \log N - O(\log \log N)$.

The case of the top bits is a bit more challenging, since now the large coefficients are very close to each other and may affect each other (see [14, Section 5]). It is possible to show that for the most significant bit and the second most significant bit, the function bit_i has a large coefficient at an invertible value, and in fact it seems that it is possible to give a conditional result as in Claim 2 for all the top bits.[8] However, since we are mainly interested in proving that we can solve the hidden number problem, we show this could be done unconditionally.

Claim 3. Let ℓ be an odd prime, let e be an integer, let $N = \ell^e$, and let $\log N - O(\log \log N) \leq i \leq \log N$. Then, there exists z and $\tau^{-1} = O(\log N)$ such that $|\widehat{\text{bit}_i}(z)|^2 > \tau$. Moreover, every z such that $|\widehat{\text{bit}_i}(z)|^2 > \tau$ for some $\tau^{-1} = O(\log N)$ satisfies $z = \ell^j z'$, where $z' \in \mathbb{Z}_N^*$ and $0 \leq j \leq O(\log_\ell \log N)$, where $\log_\ell$ is the logarithmic function in base ℓ.

Proof. We sketch the proof. From [16], the function bit_i has large Fourier coefficients. All the large coefficients of bit_i' are at values of small size (small multiples of 2^{k-i-1}). From [14], the large coefficients of bit_i are also in a short interval, of polynomial size, around 0. ∎

Using these three claims, we can now show that one can solve the hidden number problem in $\mathbb{Z}_{\ell^e}$, for an odd prime ℓ, for all top and lower bits unconditionally, and for the middle bits under the condition in Claim 2. Moreover, the solution for the top bits applies to $\ell = 2$.

Theorem 3. *Let ℓ be an odd prime unless stated otherwise, let e be an integer, let $N = \ell^e$ and let $\epsilon, \delta > 0$. Let s be an invertible element in $\mathbb{Z}_N$, i.e. $s \in \mathbb{Z}_N^*$, and consider the chosen-multiplier hidden number problem (CM-HNP) with "hidden number" s and some integer $0 \leq i \leq \log N$. Denote $z = \lfloor \frac{\ell^e}{2^{i+1}} \rceil$ and $r = |\frac{\ell^e}{2^{i+1}} - z|$. Then, if*

- *(lower bits) $0 \leq i \leq O(\log \log N)$, or*
- *(top bits) $\log N - O(\log \log N) \leq i \leq \log N$ and ℓ is any prime, or*

[7] For comparison $|\text{bit}_4(615)| > 0.62, |\text{bit}_4(1845)| > 0.18, |\text{bit}_4(3075)| > 0.089$.

[8] To sketch the proof: we can bound from below the magnitude of the largest coefficient and bound from above the other large coefficient, such that the other coefficients cannot reduce the magnitude of the largest one too much, so it has to remain large. That is, the largest coefficient can be shown to be sufficiently large such that even if the other coefficients reduce its size, it still remains significantly large.

- *(middle bits,* $O(\log\log N) < i < \log N - O(\log\log N)$*)* $r^{-1} = O(\log N)$ *or* $z \in \mathbb{Z}_N^*$,

there exists an algorithm for CM-HNP that returns a list of size $O(\log N/\epsilon)$ *that contains* s *with probability* $1-\delta$*. The algorithm runs in polynomial time in* $\log N$*,* $1/\epsilon$ *and* $\log(1/\delta)$*.*

Proof. To use the algorithm in Theorem 2 we need to show that bit_i has a significant Fourier coefficient at an invertible value. The case of the lower bits follows from Claim 1, and the case of the middle bits is shown in Claim 2. For the top bits, we do not require the large coefficients to be at invertible values. Instead, if all the values in the lists L_p, L_i (see Theorem 2) are non-invertible, then from Claim 3 (or Lemma 1 if $\ell = 2$) we can recover all bits of s except of at most $O(\log\log N)$ top bits. Indeed, write $s = \ell^{e-j}s_1 + s_2$, for $s_2 < \ell^{e-j}$. From Theorem 2 we have $b = as^{-1}$, and from Claim 3, $a = \ell^j a'$ and $b = \ell^j b'$, where $a', b' \in \mathbb{Z}_N^*$ and $0 \le j \le O(\log_\ell \log N)$. Thus, we can recover s_2. The remaining bits of s are brute forced. ∎

4. HARDNESS OF THE PRIVATE KEY BITS

We present our main results. In Section 4.1, which addresses the case $\ell \neq 2$, we show that (almost) all bits of the private key in SIDH and similar isogeny-based schemes are as hard to compute (predict) as the private key. In Section 4.2, which addresses the case $\ell = 2$, we consider a more restricted problem where one gets bits of the coefficients of given generating points (as opposed to the ability to choose the generating points). We show that computing the least significant bit of both coefficients is as hard as computing the secret isogeny. Both of these results do not use the auxiliary points in the SIDH public key. We further show how to use these points to self randomise the problem and to compute bits of the private key if the secret isogeny is not of the prescribed degree.

4.1 The case $\ell \neq 2$: bits that are hardcore

We consider an oracle $\mathcal{O}$ that takes as input a curve E (the origin curve), a curve E' (the image curve) and two points $P', Q' \in E$. If there are coefficients a, b such that $G = \langle [a]P' + [b]Q' \rangle$ and $E' = E/G$, we would like the oracle to output some information about the private key. As explained above (see also [8, Lemma 1] and its preceding paragraph) there are no unique coefficients a, b (as modular integers) that determine G, and so we need to specify some relation between the points. Our convention is that the coefficient of P' is set to be 1 (there is no restriction in considering P', as by relabeling the points we can consider the coefficient of Q'). Assuming $E' = E/G$, the oracle outputs a bit of the minimal positive s such that $G = \langle P' + [s]Q' \rangle$. If such s does not exist, i.e. $E' \neq E/G$, the oracle outputs a bit arbitrarily. The oracle's probability of success is taken over the input points P', Q' (in section 4.3.2 we present a randomisation technique for the origin and image curves, so one can also consider the case where the oracle fails on some curves).

For comparison with DLP, the analogous oracle takes as input the tuple (H, h, g) and outputs a bit of a, for the minimal positive a if exists, such that $h = g^a$, and the probability of success is over h, g (DLP is easily self randomised; the trick is to consider h^r, g^r, instead of h, g, for random integers r). We see that in both cases the oracle takes the defining group, the public key and a specific generator for the space of the public keys.

Moreover, while not typical, it is possible that two different isogenies of degree (dividing) ℓ^e exist between E and E/G. In order to make sure that our problem is well defined, we assume that such an isogeny is unique, up to automorphism. In particular, if the auxiliary points are also considered, this assumption holds (see [23, Lemma 3.2]).

First, we give the following lemma, where we consider a relaxed oracle that takes as input also the coefficient of the point P' (as opposed to replace the input point P' by the point $[r]P'$), and its probability of success is taken over the input coefficient.

Lemma 2. *Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, let ℓ be an odd prime, let $P, Q \in E[\ell^e]$ be two linearly independent generators of $E[\ell^e]$, and let $G = \langle [a]P + [b]Q \rangle$ of order ℓ^e be unknown. Suppose that there is a unique isogeny from E to E/G of degree dividing ℓ^e. Let $n = \lfloor e \log(\ell) \rfloor$, let $0 \leq i \leq n$ and let $\epsilon, \delta > 0$. Suppose $\mathcal{O}$ satisfies*

$$\Pr_{r \in \mathbb{Z}_{\ell^e}} [\mathcal{O}(E, E/G, P', Q'; r) = \mathit{bit}_i(s)] \geq \frac{1 + \beta_i(N)}{2} + \epsilon \,,$$

for the minimal $s \in \mathbb{Z}_{\ell^e}$, if exists, such that $G = \langle [r]P' + [s]Q' \rangle$. Then, under the conditions in Theorem 3 and given the curve E/G, the group $G \subseteq E$ can be computed with probability $1 - \delta$ in polynomial time in $e \log(\ell), 1/\epsilon$ and $\log(1/\delta)$.

Proof. Since G is of order ℓ^e either $a \in \mathbb{Z}^*_{\ell^e}$ or $b \in \mathbb{Z}^*_{\ell^e}$. Suppose $b \in \mathbb{Z}^*_{\ell^e}$, otherwise switch P and Q. If $a = 0$ then $G = \langle Q \rangle$. Otherwise write $a = \ell^k a'$ for some $0 \leq k < e$ such that $a' \not\equiv 0 \pmod{\ell}$, then $a' \in \mathbb{Z}^*_{\ell^e}$. Let $b' = a'^{-1} b \in \mathbb{Z}^*_{\ell^e}$. Then $G = \langle [\ell^k]P + [b']Q \rangle$. For now suppose that k is known. We show how to compute b', which is sufficient to recover G.

Note that for any $r \in \mathbb{Z}^*_{\ell^e}$ we have $G = \langle [r\ell^k]P + [rb']Q \rangle$. Thus $\mathcal{O}(E, E/G, [\ell^k]P, Q; r)$ outputs $\mathrm{bit}_i(rb')$ with advantage ϵ if $r \in \mathbb{Z}^*_{\ell^e}$. For any other r we make a coin flip for the i-th bit of rb'. We define $\mathcal{O}'$ as follows

$$\mathcal{O}'(r) = \begin{cases} \mathcal{O}(E, E/G, [\ell^k]P, Q; r), & \text{when } r \in \mathbb{Z}^*_{\ell^e}, \\ \text{coin flip}, & \text{otherwise.} \end{cases}$$

Since ℓ is prime, at least half of the values $r \in \mathbb{Z}_{\ell^e}$ satisfy $r \in \mathbb{Z}^*_{\ell^e}$. For the other values, we expect coin flips to give the correct bit with probability $1/2$. Thus the advantage of $\mathcal{O}'$ in predicting $\mathrm{bit}_i(rb')$ is at least $\epsilon/2$. In other words

$$\Pr_{r \in \mathbb{Z}_{\ell^e}} [\mathcal{O}'(r) = \mathrm{bit}_i(rb')] \geq \frac{1 + \beta_i(N)}{2} + \epsilon/2 \,.$$

The problem of recovering b' is now reduced to CM-HNP in $\mathbb{Z}_{\ell^e}$ with predictor $\mathcal{O}'$. By Theorem 3 we can obtain a bounded list of candidates for b'. Let $\bar{b}$ be such a candidate, then one can compute $\bar{G} = \langle [\ell^k]P + [\bar{b}]Q \rangle$, and check whether $E/\bar{G} \simeq E/G$ (for example by computing their j-invariant). With probability $1 - \delta$ one of the candidates is b', and so the claim follows. Finally, since we don't know k a priori, we run this algorithm for each $k < e$. ∎

By restricting the oracle's probability space to the coefficients r in the lemma, we know that for *any pair of points* P', Q' that generates G, the oracle has a non-negligible advantage in outputting the correct bit. We now show that it is possible to obtain points P', Q' for which the oracle has a non-negligible advantage in outputting the correct bit, even when its success probability is taken over the pair of points (P', Q') (and the coefficient of P' is considered to be 1). For this result we need the following claim, known as the reverse Markov inequality.

Claim 4. Let X be a random variable that satisfies $\Pr[X \leq a] = 1$ for some constant a. Then, for $d < \mathrm{E}[X]$

$$\Pr[X > d] \geq \frac{\mathrm{E}[X] - d}{a - d}\,.$$

Theorem 4. *Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, let ℓ be an odd prime, let $P, Q \in E[\ell^e]$ be two linearly independent generators of $E[\ell^e]$, and let $G = \langle [a]P + [b]Q \rangle$ of order ℓ^e be unknown. Suppose that there is a unique isogeny from E to E/G of degree dividing ℓ^e. Let $n = \lfloor e \log(\ell) \rfloor$, let $0 \leq i \leq n$ and let $\epsilon, \delta' > 0$. Suppose $\mathcal{O}$ satisfies*

$$\Pr_{P',Q'}[\mathcal{O}(E, E/G, P', Q') = bit_i(s)] \geq \frac{1 + \beta_i(N)}{2} + \epsilon\,,$$

for the minimal $s \in \mathbb{Z}^*_{\ell^e}$, if exists, such that $G = \langle P' + [s]Q' \rangle$.[9] *Then, given the curve E/G and under the conditions in Theorem 3, namely*

- $0 \leq i \leq O(\log\log N)$, *or*
- $\log N - O(\log\log N) \leq i \leq \log N$, *or*
- $O(\log\log N) < i < \log N - O(\log\log N)$ *and the "middle bits" condition in Theorem 3 holds,*

the group $G \subseteq E$ can be computed with probability $1 - \delta'$ in polynomial time in $e\log(\ell), 1/\epsilon$ and $\log(1/\delta')$.

[9] The general case of oracles with advantage over the entire $\mathbb{Z}_{\ell^e}$ is left open. The potential problem is an oracle that outputs the correct bit when s is non-invertible (and otherwise a random bit), a case in which we cannot directly follow the proof of Lemma 2.

Proof. Let $r \in \mathbb{Z}^*_{\ell^e}$ and note that if $\bar{P} \in E[\ell^e]$ is chosen uniformly at random, also $P' := [r]\bar{P}$ is. Indeed, multiplication by an invertible element r is a permutation of $E[\ell^e]$, as if $[r]P_1 = [r]P_2$ then $[r](P_1 - P_2) = 0$, but $\ell \nmid r$ so $P_1 = P_2$. We have

$$\Pr_{\bar{P},Q',r}[\mathcal{O}(E, E/G, [r]\bar{P}, Q') = \text{bit}_i(s)] = \Pr_{P',Q'}[\mathcal{O}(E, E/G, P', Q') = \text{bit}_i(s)]\,.$$

Fix $\bar{P}, Q'$ and let X be the probability over r that $\mathcal{O}(E, E/G, [r]\bar{P}, Q') = \text{bit}_i(s)$, for the minimal positive $s \in \mathbb{Z}^*_{\ell^e}$, conditioned that such s exists. That is,

$$X = \Pr_{r \in \mathbb{Z}^*_{\ell^e}}[\mathcal{O}(E, E/G, [r]\bar{P}, Q') = \text{bit}_i(s) \mid \exists s \in \mathbb{Z}^*_{\ell^e}, G = \langle [r]\bar{P} + [s]Q' \rangle]\,.$$

By the assumption on $\mathcal{O}$, we have $E_{\bar{P},Q'}[X] \geq \frac{1+\beta_i(N)}{2} + \epsilon$, where the expectation is taken over all pairs $(\bar{P}, Q')$ such that $G = \langle \bar{P} + [b']Q' \rangle$ for some $b' \in \mathbb{Z}^*_{\ell^e}$ (note that since $r \in \mathbb{Z}^*_{\ell^e}$ then also $G = \langle [r]\bar{P} + [rb']Q' \rangle$, and $s = rb' \in \mathbb{Z}^*_{\ell^e}$). We now apply Claim 4 with $a = 1$ and $d = \frac{1+\beta_i(N)}{2} + \epsilon/2$. We get that

$$\Pr_{\bar{P},Q'}\left[\Pr_r[\mathcal{O}(E, E/G, [r]\bar{P}, Q') = \text{bit}_i(s) \mid \exists s \in \mathbb{Z}^*_{\ell^e}, G = \langle [r]\bar{P} + [s]Q' \rangle] \geq \frac{1+\beta_i(N)}{2} + \frac{\epsilon}{2}\right] \geq \frac{\epsilon}{2}\,.$$

Next, we show how to obtain points $\bar{P}, Q' \in E[\ell^e]$ such that with probability $\epsilon/2$ we have $\Pr_r[\mathcal{O}(E, E/G, [r]\bar{P}, Q') = \text{bit}_i(s)] \geq \frac{1+\beta_i(N)}{2} + \frac{\epsilon}{2}$. By the previous inequality, it is sufficient to show that $G = \langle \bar{P} + [b']Q' \rangle$ for some $b' \in \mathbb{Z}^*_{\ell^e}$. Draw at random $\tilde{P}, \tilde{Q} \in E[\ell^e]$. Suppose that they are linearly independent generators, so $G = \langle [c]\tilde{P} + [d]\tilde{Q} \rangle$ for some c, d. As above, since G is of order ℓ^e either $c \in \mathbb{Z}^*_{\ell^e}$ or $d \in \mathbb{Z}^*_{\ell^e}$.

We now consider $2e - 1$ alternatives. First, set $\bar{P} = \tilde{P}$, and for each $0 \leq k < e$ set $Q' = [\ell^k]\tilde{Q}$. If $c \in \mathbb{Z}^*_{\ell^e}$, then for one of the Q' we have $G = \langle \bar{P} + [b']Q' \rangle$, where $b' \in \mathbb{Z}^*_{\ell^e}$ (except for the negligible case where $G = \langle \bar{P} \rangle$). Secondly, set $Q' = \tilde{Q}$, and for each $0 < k < e$ set $\bar{P} = [\ell^k]\tilde{P}$. If $c \notin \mathbb{Z}^*_{\ell^e}$, then for one of the $\bar{P}$ we have $G = \langle \bar{P} + [b']Q' \rangle$, where $b' \in \mathbb{Z}^*_{\ell^e}$ (with the exception $G = \langle Q' \rangle$).

Hence, one of these alternatives satisfies $G = \langle \bar{P} + [b']Q' \rangle$, so with probability at least $\epsilon/2$

$$\Pr_r[\mathcal{O}(E, E/G, [r]\bar{P}, Q') = \text{bit}_i(s)] \geq \frac{1+\beta_i(N)}{2} + \frac{\epsilon}{2}\,, \tag{1}$$

where $r \in \mathbb{Z}^*_{\ell^e}$ and $s = rb'$. As explained in the end of Section 2, the probability that the points $\tilde{P}, \tilde{Q}$ are linearly independent generators is at least $3/8$. Hence, the overall probability that (1) holds is at least $3\epsilon/16$.

Conditioning on (1), we recover s similarly to the recovery procedure in Lemma 2 (note that while in Lemma 2 the probability is taken over all r, we only query $\mathcal{O}$ on $r \in \mathbb{Z}^*_{\ell^e}$, as for other r the advantage is not assumed not hold).

Finally, to recover G with probability $1-\delta'$ we repeat this procedure (drawing $\tilde{P}, \tilde{Q}$ and creating the $2e-1$ alternatives on which we apply Lemma 2) w times. For each recovered group G', we compute the isogeny $E \to E/G'$ and check if $E/G' \simeq E/G$. We fail to get the desired $\bar{P}, Q'$ with probability at most $(1-3\epsilon/16)^w$. Therefore, the probability that one of the candidates G' satisfy $E/G' \simeq E/G$ is at least $(1-(1-3\epsilon/16)^w)(1-\delta)$, where $(1-\delta)$ is the sucess probability in Lemma 2. Set $\delta = \delta'/2$, then we need $1-(1-3\epsilon/16)^w > \frac{1-\delta'}{1-\delta'/2} = 1 - \frac{\delta'/2}{(1-\delta'/2)}$. Hence, by expressing w, we get that the number of repetitions is polynomial in $\log(1/\delta'), \log(1/\epsilon)$. ∎

4.2 The case $\ell = 2$: the hardness of LSB for non-chosen points

We first consider the case where the torsion points P', Q' cannot be chosen, as opposed to the previous case. Instead, they are given by the oracle upon specifying the torsion subgroup. For example, this can be thought of the case where some "canonical" basis of $E[\ell^e]$ is set (see for example [26] where such bases are used for compression).

We consider the special case $\ell = 2$. We remark that for efficiency it is suggested to take the minimal possible ℓ, i.e. $\ell = 2$ (in SIDH key exchange the other party takes $\ell = 3$). We show that an algorithm that computes the least significant bit of both coefficients can be turned into an algorithm that computes the entire coefficients. Our recovery method resembles the one in [8, Section 3].

Theorem 5. *Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, let $P, Q \in E[2^n]$ be two linearly independent generators of $E[2^n]$, and let $G = \langle [a]P + [b]Q \rangle$ of order 2^n be unknown. Suppose $\mathcal{O}$ satisfies*

$$\mathcal{O}(E', E'', n') = (P', Q', \mathit{LSB}(a_1), \mathit{LSB}(a_2)),$$

for $P', Q' \in E'[2^{n'}]$ two linearly independent generators of $E'[2^{n'}]$, if there exist $a_1, a_2 \in \mathbb{Z}_{2^{n'}}$ such that $E'' = E'/\langle [a_1]P' + [a_2]Q' \rangle$. Then, given the curve E/G, the group $G \subseteq E$ can be computed in polynomial time in n, if there is a unique isogeny from E to E/G of degree dividing 2^n; otherwise, the output is one of the defining kernels (not necessarily G).

Proof. As above we may assume that $G = \langle P + [b']Q \rangle$ where $b' = a^{-1}b$. It is sufficient to recover b' in order to compute G. We show how to recover b', bit by bit, from the least significant bit to the most significant bit. Write $b' = \epsilon_0 + \epsilon_1 2 + \cdots + \epsilon_{n-1}2^{n-1}$. Let $\phi : E \to E/G$ be an isogeny of degree 2^n. Then $\phi = \phi_n \circ \cdots \circ \phi_1$ where $\deg \phi_i = 2$. Recall the notation from Section 2. The main idea of the recovery construction is to use the information from $\mathcal{O}$ to compute ϕ_i given E_{i-1}. We first show how to recover ϵ_0, E_1 and $\phi_1 : E_0 \to E_1$.

At first make the query $\mathcal{O}(E, E/G, n)$. We recieve P', Q' that generate $E[2^n]$, hence there are $\alpha, \beta, \gamma, \delta$ such that

$$P = \alpha P' + \beta Q', \qquad Q = \gamma P' + \delta Q',$$

where at least one of α, β and one of γ, δ are odd, since P, Q are of full order. The values $\alpha, \beta, \gamma, \delta$ can be efficiently computed as a two-dimensional discrete logarithm problem (see for example [26]). Moreover, since $G \subseteq E[2^n]$ there are a_1, a_2 such that $G = \langle [a_1]P' + [a_2]Q' \rangle$, and so we receive $\text{LSB}(a_1), \text{LSB}(a_2)$.

We have

$$R := P + [b']Q = [\alpha + b'\gamma]P' + [\beta + b'\delta]Q' = [\bar{a}]P' + [\bar{b}]Q' \,,$$

where $\bar{a} = \alpha + b'\gamma, \bar{b} = \beta + b'\delta$. Note that there exists $c \in \mathbb{Z}_{2^n}^*$ such that $a_1 = c\bar{a}, a_2 = c\bar{b}$, and since c is odd we have $\text{LSB}(a_1) = \text{LSB}(\bar{a}), \text{LSB}(a_2) = \text{LSB}(\bar{b})$. Suppose γ is odd, then $\alpha + b' \equiv \text{LSB}(a_1) \pmod 2$, and since α is known we can compute $\epsilon_0 = \text{LSB}(b') = b' \mod 2 = \text{LSB}(a_1) - \alpha \mod 2$. If γ is even then δ is odd and we compute ϵ_0 in a similar way.

We show how to use ϵ_0 to compute E_1 and ϕ_1. Recall that $E_1 = E/\langle 2^{n-1}R \rangle$. Thus

$$2^{n-1}R = 2^{n-1}P + 2^{n-1}[b']Q = 2^{n-1}P + 2^{n-1}[\epsilon_0]Q \,.$$

Since the kernel of ϕ_1 is generated by $2^{n-1}P + 2^{n-1}[\epsilon_0]Q$, we see that ϵ_0 is sufficient to determine ϕ_1 and the curve $E_1 = E/\langle 2^{n-1}R \rangle$.

Suppose $\epsilon_0, \ldots, \epsilon_{i-1}$, $\phi_1, \ldots, \phi_i$ and $E_0, \ldots, E_i$ are known. We show how to recover ϵ_i, ϕ_{i+1} and E_{i+1}. Define $\bar{\phi}_i = \phi_i \circ \cdots \circ \phi_1$. Since $\phi_1, \ldots, \phi_i$ are known, we can compute $\bar{\phi}_i(P), \bar{\phi}_i(Q)$. Note that

$$R_i := \phi_i(R_{i-1}) = \bar{\phi}_i(R) = \bar{\phi}_i(P + [b']Q) = \bar{\phi}_i(P) + [b']\bar{\phi}_i(Q) \,.$$

Therefore $E_{i+1} = E_i/\langle 2^{n-i-1}R_i \rangle = E_i/\langle 2^{n-i-1}\bar{\phi}_i(R) \rangle = E_i/\langle 2^{n-i-1}\bar{\phi}_i(P + [b']Q) \rangle$.

Let $K = \epsilon_0 + 2\epsilon_1 + \cdots + 2^{i-1}\epsilon_{i-1}$. We have

$$R_i = \bar{\phi}_i(P) + [b']\bar{\phi}_i(Q) = \bar{\phi}_i(P + [K]Q) + \left[\frac{b' - K}{2^i}\right] \bar{\phi}_i(2^i Q) \,.$$

It is clear that $\bar{\phi}_i(2^i Q) \in E_i[2^{n-i}]$, as $[2^{n-i}]\bar{\phi}_i(2^i Q) = \bar{\phi}_i([2^n]Q) = 0$. Moreover, $\bar{\phi}_i(2^i Q)$ is of order 2^{n-i}, otherwise Q is not of full order. Similarly, $\bar{\phi}_i(P + [K]Q) \in E_i[2^{n-i}]$ since $[2^{n-i}]\bar{\phi}_i(P + [K]Q)) = \phi_i(2^{n-i}R_{i-1}) = 0$ by definition of ϕ_i. Moreover, $\bar{\phi}_i(P + [K]Q))$ is of order 2^{n-i}, otherwise $\phi_i(R_{i-1})$ is of order smaller than 2^{n-i} which contradicts the fact that the degree of ϕ_i is 2 (as the group generated by $2^{n-i}R_{i-1}$ is not of order 2).

Make the query $\mathcal{O}(E_i, E/G, n-i)$ to obtain two points P', Q' that generate $E_i[2^{n-i}]$ and $\text{LSB}(a_1), \text{LSB}(a_2)$ such that $E/G = E_i/\langle [a_1]P' + [a_2]Q' \rangle$ (by the problem's construction, it is clear that an isogeny of degree 2^{n-i} exists between E_i and E/G).

Since P', Q' are of full order in $E[2^{n-i}]$, there are $\alpha, \beta, \gamma, \delta$ such that

$$\bar{\phi}_i(P + [K]Q)) = \alpha P' + \beta Q' \,, \qquad \bar{\phi}_i(2^i Q) = \gamma P' + \delta Q' \,,$$

where at least one of α, β and one of γ, δ are odd, since $\bar{\phi}_i(P + [K]Q)), \bar{\phi}_i(2^i Q) \in E_i[2^{n-i}]$ are of full order. The values $\alpha, \beta, \gamma, \delta$ can be efficiently computed as a two-dimensional discrete logarithm problem. We have

$$R_i = \bar{\phi}_i(P+[K]Q)+\left[\frac{b'-K}{2^i}\right]\bar{\phi}_i(2^iQ) = \left[\alpha + \frac{b'-K}{2^i}\gamma\right]P' + \left[\beta + \frac{b'-K}{2^i}\delta\right]Q' = [\bar{a}]P' + [\bar{b}]Q' \,,$$

where $\bar{a} = \alpha + \frac{b'-K}{2^i}\gamma, \bar{b} = \beta + \frac{b'-K}{2^i}\delta$. Similar to the arguments above, we can recover $\epsilon_i = \frac{b'-K}{2^i} \mod 2$.

Finally,

$$2^{n-i-1}R_i = [2^{n-i-1}]\bar{\phi}_i(P) + [2^{n-i-1}b']\bar{\phi}_i(Q) = [2^{n-i-1}]\bar{\phi}_i(P) + [2^{n-i-1}\epsilon_0 + \cdots + 2^{n-1}\epsilon_i]\bar{\phi}_i(Q)\,.$$

Since the kernel of ϕ_{i+1} is generated by $2^{n-i-1}R_i$, we see that $\epsilon_0, \ldots, \epsilon_i$ are sufficient to determine ϕ_{i+1} and the curve $E_{i+1} = E_i/\langle 2^{n-i-1}R_i\rangle$. ∎

We can use the approach from the previous theorem to show that an oracle, as in Section 4.1, that computes any single bit of the private key, can be used to fully compute the secret isogeny. This is done by a standard method of shifting the bits to the computable position, and so we only sketch the proof.

Proposition 1. *Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, let $P, Q \in E[2^n]$ be two linearly independent generators of $E[2^n]$, let $G = \langle [a]P + [b]Q\rangle$ of order 2^n be unknown, and let $0 \le i < n$. Suppose that there is a unique isogeny from E to E/G of degree dividing 2^n. Suppose $\mathcal{O}$ satisfies*

$$\mathcal{O}(E, E/G, P', Q') = \mathit{bit}_i(s)\,,$$

for the minimal $s \in \mathbb{Z}_{\ell^e}$, if exists, such that $G = \langle P' + [s]Q'\rangle$. Then given the curve E/G, the group $G \subseteq E$ can be computed with n calls to $\mathcal{O}$.

Proof sketch. Suppose $G = \langle P + [b']Q\rangle$, we recover b'. The recovery procedure is divided into $\lceil n/(i+1)\rceil$ steps. At each step we recover a block of $i+1$ (consecutive) bits, except for the last step that may have less bits to recover. This is done by shifting each bit into the i-th position. First, for every $0 \le j \le i$ query $\mathcal{O}(E, E/G, P, [2^j]Q) = \mathrm{bit}_{i-j}(b')$. This gives the lowest $i+1$ bits of b'. Secondly, we 'shift' the bits of b' by $i+1$ positions as follows. Let $K = \epsilon_0 + 2\epsilon_1 + \cdots + 2^i\epsilon_i$ be the recovered bits, i.e. $\epsilon_j = \mathrm{bit}_j(b')$. Then $G = \langle P + [b']Q\rangle = \langle P + [K]Q + \left[\frac{b'-K}{2^{i+1}}\right]([2^{i+1}]Q)\rangle$. Now, repeat the first step with $P \to P + [K]Q$ and $Q \to [2^{i+1}]Q$). That is, for every $0 \le j \le i$ query $O(E, E/G, P + [K]Q, [2^j]([2^{i+1}]Q))$ to obtain the next block of $i+1$ bits. Repeat until all bits are recovered. ∎

4.3 Making use of the auxiliary points

We show how the auxiliary points $\phi(P_1), \phi(Q_1)$ in SIDH can be used to detect if the group G is not of full order, thus revealing some low order bits of the private key, and to randomise our reductions. We remark that the ideas presented here are well known, and so we keep this presentation brief.

4.3.1 Bits that are easy to compute

We first show that if the order of $G = \langle [a]P + [b]Q \rangle \subseteq E[\ell^e]$ is less than ℓ^e, then one can compute some lower bits of a and b. This is due to the fact that G is not of full order in $E[\ell^e]$ if and only if $a \equiv b \equiv 0 \pmod{\ell}$. Moreover, if $a \equiv 0 \pmod{\ell^\alpha}$ and $b \equiv 0 \pmod{\ell^\beta}$, then for $\gamma = \min\{\alpha, \beta\}$ we have that $G = \langle \ell^\gamma([a']P + [b']Q) \rangle$ and so G is a cyclic group of order (at most) $\ell^{e-\gamma}$.

Since $\deg(\phi) = |G|$ we can use the Weil pairing to compute γ. Indeed, a standard fact is that if $\phi : E \to E'$ is an isogeny and if $P_1, Q_1 \in E[N]$ then

$$e_N(\phi(P_1), \phi(Q_1)) \;=\; e_N(P_1, Q_1)^{\deg(\phi)}$$

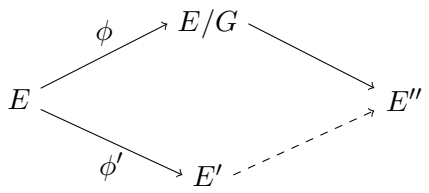
where the first Weil pairing is computed on E' and the second on E. Therefore, one can range over the possible γ and test this equality. Once γ is found we have partial knowledge on a, b, i.e. $a, b \equiv 0 \pmod{\ell^\gamma}$. For more details see for example [8, Remark 1] and reference within.

Interestingly a similar result holds for DLP, albeit with a different technique, in case the group generated by g is of order $2^s r$, $s > 0$ and r odd. To learn the least significant bit of a given g^a we see that $(g^a)^{2^{s-1}r}$ is 1 if a is even and -1 if a is odd (this is simply the Jacobi symbol which tells us whether g^a is a quadratic residue). It is an easy exercise to generalise this technique to learn all s least significant bits of a.

4.3.2 Randomisation

Note that during the reduction in Section 4.1 the curves $E, E/G$ are fixed, while in Section 4.2 the curve E/G is fixed (and $\mathcal{O}$ always succeeds in the prediction). It is interesting to consider the case where $\mathcal{O}$ fails on some set of origin curves E or image curves E/G. That is, $\mathcal{O}$ has no noticeable advantage on these curves. It is therefore desirable to be able to randomise the input curves.

The auxiliary points allow us to change the origin and image curves. In fact, this is in the heart of the key exchange that allows Alice and Bob to apply "their" isogenies (i.e. taking their secret path on the isogeny graph) on the other party's public curve. Consider the following diagram.



Let $G = \langle [a]P + [b]Q \rangle \subseteq E[\ell^e]$ as above. We suppose that $E, E/G$ and $\phi(P_1), \phi(Q_1) \in E/G$ are public, for some known $P_1, Q_1 \in E[\ell_1^{e_1}]$, $\ell_1 \neq \ell$. Therefore, we can choose m, n and compute an isogeny $\phi' : E \to E'$ with kernel $\langle [m]P_1 + [n]Q_1 \rangle$ (and the points $\phi'(P), \phi'(Q)$). Since $\phi(P_1), \phi(Q_1)$

are given, we can also compute the isogeny from E/G with kernel $\langle [m]\phi(P_1) + [n]\phi(Q_1)\rangle$ to a curve denoted by E''. A similar idea is described in [8, Section 5], which we refer to for more details.

It holds that there is an isogeny of degree ℓ^e from E' to E'' with kernel $\langle [a]\phi'(P) + [b]\phi'(Q)\rangle$. Hence, one can replace $E, E/G$ with E', E'' to get partial information on the *same* private key a, b. Moreover, this allows us to have reduction even if the success probability of $\mathcal{O}$ in Section 4.2 is low, by repeating each step on different pair of curves, and applying majority rule on the recovered bits.

5. Summary and open problems

In this paper we studied the bit security of the private key in (supersingular) isogeny-based schemes. While the private key consists of two coefficients, one coefficient is essentially known, and so our study focuses on the hardness of obtaining single bits of the other coefficient, which we now refer to as "the private key". For a party that works on a torsion group of an odd order N, we showed that each of the top and lower $O(\log\log N)$ bits of the private key is as hard to compute, with any non-negligible advantage, as the entire private key. For the middle bits, we gave a condition for the same result to hold. This condition can be checked and we believe that it holds almost always – proving this is left for the future. For the even torsion group, the same result holds for the top bits. Moreover, we showed that computing any bit of the private key, with probability 1, is as hard as computing the private key. We also showed, in a less flexible model, that computing, with probability 1, the least significant bit of both coefficients is as hard as computing the entire secret isogeny. We remark that since the odd-case result is stronger, if both parties want to guarantee full bit security of their private keys, they may prefer to agree on non-even primes ℓ_A, ℓ_B (for which the middle-bits condition holds for each bit).[10]

Our work leaves a few open problems. When the middle-bits condition does not hold for a certain bit, is it possible to show that this bit cannot be predicted? Is it possible to obtain a more general result for the even torsion group? One approach would be to use the randomisation technique in Section 4.3.2, however this technique only holds for small primes ℓ_1. Achieving a total randomisation is an open problem, with applications to other works.

We remark that the result in Theorem 5 may hold where one is given the most significant bit, instead of the least significant bit; since the most significant bit is only used for the very last 2-isogeny, the approach would be to start at the image curve and "walk" towards the origin. Finally, while not directly related to the bit security of SIDH, it is interesting to consider a full solution to CM-HNP for every secret $s \in \mathbb{Z}_{\ell^e}$. This could be obtained if the functions $g_i^k(x) := \text{bit}_i(\ell^k x)$ can be shown to have large coefficients. One natural approach to (try to) achieve this result would be to walk along the proof of [14, Lemma 6.2]. Filling the details for both of these remarks may be a nice student project.

[10] We clarify that other factors, like execution time, may also be taken into account in the choice of the SIDH primes.

Acknowledgements The author thanks two anonymous Eurocrypt 2019 reviewers for pointing out several issues.

REFERENCES

[1] Akavia, A., Goldwasser, S., and Safra, S. (2003) "Proving Hard-Core Predicates Using List Decoding," in FOCS 2003, pp. 146–157. IEEE Computer Society, Washington, DC.

[2] Boneh, D., and Venkatesan, R. (1996) "Hardness of Computing the Most Significant Bits of Secret Keys in Diffie–Hellman and Related Schemes," in Koblitz, N. (ed.) Advances in Cryptology – CRYPTO 1996. LNCS, vol. 1109, pp. 129–142. Springer, Heidelberg.

[3] Costello, C., Longa, P., and Naehrig, M. (2016) "Efficient Algorithms for Supersingular Isogeny Diffie-Hellman," in Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016. LNCS, vol. 9814, pp. 572–601. Springer, Heidelberg.

[4] Couveignes, J.-M. (2006) "Hard Homogeneous Spaces," in Cryptology ePrint Archive, Report 2006/291. http://eprint.iacr.org/2006/291.

[5] De Feo, L. (2017) "Mathematics of Isogeny Based Cryptography," in arXiv:1711.04062 [cs.CR]. https://arxiv.org/abs/1711.04062.

[6] De Feo, L., Jao, D., and Plût, J. (2014) "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies," in Journal of Mathematical Cryptology, **8**(3), 209–247.

[7] Galbraith, S.D., Laity, J., and Shani, B. (2018) "Finding Significant Fourier Coefficients: Clarifications, Simplifications, Applications and Limitations," in Chicago Journal of Theoretical Computer Science, **2018**(6), 1–38.

[8] Galbraith, S.D., Petit, C., Shani, B., and Ti, Y.B. (2016) "On the Security of Supersingular Isogeny Cryptosystems," in Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016. LNCS, vol. 10031, pp. 63–91. Springer, Heidelberg.

[9] Galbraith, S.D., Petit, C., and Silva, J. (2017) "Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems," in Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. LNCS, vol. 10624, pp. 3–33. Springer, Heidelberg.

[10] Galbraith, S.D., and Vercauteren, F. (2017) "Computational Problems in Supersingular Elliptic Curve Isogenies," in Quantum Information Processing, 17:256.

[11] González Vasco, M.I., and Näslund, M. (2001) "A Survey of Hard Core Functions," in Lam, K.-Y., Shparlinski, I., Wang, H., Xing, C. (eds.) Proc. Workshop on Cryptography and Computational Number Theory 1999. Progress in Computer Science and Applied Logic, vol. 20, pp. 227–255. Birkhäuser, Basel.

[12] Håstad, J., and Näslund, M. (2003) "The Security of all RSA and Discrete Log Bits," in Journal of the ACM, **51**(2), 187–230.

[13] Jao, D., and De Feo, L. (2011) "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies," in Yang, B.-Y. (ed.) Post-Quantum Cryptography – PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer, Heidelberg.

[14] Laity, J., and Shani, B. (2018) "On Sets of Large Fourier Transform Under Changes in Domain," in Applied and Computational Harmonic Analysis **45**, 216–232.

[15] Li, W.-C.W., Näslund, M., and Shparlinski, I.E. (2002) "Hidden Number Problem with the Trace and Bit Security of XTR and LUC," in Yung, M. (ed.) Advances in Cryptology – CRYPTO 2002. LNCS, vol. 2442, pp. 433–448. Springer, Heidelberg.

[16] Morillo, P., and Ràfols, C. (2009) "The Security of All Bits Using List Decoding," in Jarecki, S., Tsudik, G. (eds.) Public Key Cryptography – PKC 2009: Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography. LNCS, vol. 5443, pp. 15–33. Springer, Heidelberg.

[17] Petit, C. (2017) "Faster Algorithms for Isogeny Problems Using Torsion Point Images," in Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. LNCS, vol. 10625, pp. 330–353. Springer, Heidelberg.

[18] Rostovtsev, A., and Stolbunov, A. (2006) "Public-Key Cryptosystem Based on Isogenies," in Cryptology ePrint Archive, Report 2006/145. http://eprint.iacr.org/2006/145.

[19] Shparlinski, I.E. (2005) "Playing "Hide-and-Seek" with Numbers: The Hidden Number Problem, Lattices and Exponential Sums," in Garrett, P., Lieman, D. (eds.) Public-Key Cryptography; Proceedings of Symposia in Applied Mathematics, vol. 62, AMS, pp. 153–177.

[20] Silverman, J.H. (2009) The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics, vol. 106 (2nd edition). Springer-Verlag, New York.

[21] Stolbunov, A. (2010) "Constructing Public-key Cryptographic Schemes Based on Class Group Action on a Set of Isogenous Elliptic Curves," in Advances in Mathematics of Communications, **4**(2), 215–235.

[22] Ti, Y.B. (2017) “Fault Attack on Supersingular Isogeny Cryptosystems,” in Lange, T. Takagi, T., (eds.) Post-Quantum Cryptography – PQCrypto 2017. LNCS, vol. 10346, pp. 107–122. Springer, Heidelberg.

[23] Urbanik, D., and Jao, D. (2018) “SoK: The Problem Landscape of SIDH,” in APKC ’18 – Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop, pp. 53–60. ACM, New York, NY, USA.

[24] Vélu, J. (1971) “Isogénies entre courbes elliptiques,” in Comptes Rendus de l’Académie des Sciences de Paris, Série A, **273**, 238–241.

[25] Yoo, Y., Azarderakhsh, R., Jalali, A., Jao, D., and Soukharev, V. (2017) “A Post-quantum Digital Signature Scheme Based on Supersingular Isogenies,” in Kiayias, A. (ed.) Financial Cryptography 2017. LNCS, vol. 10322, pp. 163–181. Springer, Heidelberg.

[26] Zanon, G.H.M., Simplicio Jr., M.A., Pereira, G.C.C.F., Doliskani, J., and Barreto, P.S.L.M. (2018) “Faster Isogeny-Based Compressed Key Agreement,” in Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography – PQCrypto 2018. LNCS, vol. 10786, pp. 248–268. Springer, Heidelberg.