Side-Channel Attacks on Quantum-Resistant Supersingular Isogeny Diffie-Hellman

Brian Koziel[1(✉)], Reza Azarderakhsh[2], and David Jao[3]

[1] Texas Instruments, Dallas, USA
kozielbrian@gmail.com
[2] CEECS Department and I-SENSE FAU, Boca Raton, USA
razarderakhsh@fau.edu
[3] C&O Department, University of Waterloo, Waterloo, Canada
djao@uwaterloo.ca

Abstract. In this paper, we present three side-channel attacks on the quantum-resistant supersingular isogeny Diffie-Hellman (SIDH) key exchange protocol. These refined power analysis attacks target the representation of a zero value in a physical implementation of SIDH to extract bits of the secret key. To understand the behavior of these zero-attacks on SIDH, we investigate the representation of zero in the context of quadratic extension fields and isogeny arithmetic. We then present three different refined power analysis attacks on SIDH. Our first and second attacks target the Jao, De Feo, and Plût three-point Montgomery ladder by utilizing a partial-zero attack and zero-value attack, respectively. Our third attack proposes a method to break the large-degree isogeny by utilizing zero-values in the context of isogenies. The goal of this paper is to illustrate additional security concerns for an SIDH static-key user.

Keywords: Side-channel attacks · Post-quantum cryptography
Isogeny-based cryptosystems · Elliptic curve cryptography

1 Introduction

Much of today's digital infrastructure relies on the security of key public-key cryptosystems, namely RSA and elliptic curve cryptography (ECC). The security assumption in both of these cryptosystems is effectively broken by a quantum computer using Shor's algorithm [1]. Thus, to counteract any potential crises with the emergence of a quantum computer, considerable research has gone into post-quantum cryptography (PQC), which studies cryptosystems that are infeasible to break in the presence of both quantum and classical computers.

The supersingular isogeny Diffie-Hellman (SIDH) key exchange protocol has been earning a large amount of attention since it resembles the elliptic curve Diffie-Hellman key exchange protocol, provides forward secrecy, and has much smaller key sizes in comparison to other quantum-resistant schemes. SIDH is slow compared to its competitors but the smaller key sizes allow for an efficient transmission of information over a public channel. This scheme's security

C. Adams and J. Camenisch (Eds.): SAC 2017, LNCS 10719, pp. 64–81, 2018.
https://doi.org/10.1007/978-3-319-72565-9_4

assumption is based on the difficulty to compute supersingular isogenies between supersingular elliptic curves. This is believed to be difficult for both classical and quantum computers. Compared to other quantum-resistant schemes, SIDH is the newest. Originally introduced by David Jao and Luca De Feo in 2011 [2], the theory and computational efficiency of SIDH has grown: undeniable signatures [3], digital signatures [4,5], key compression [6,7], projective isogeny formulas [8], and efficient software and hardware implementations [8–13].

Recently, there were two proposed fault attacks accepted at PQCrypto 2017 [14,15]. Otherwise, the literature is relatively sparse on side-channel attacks. Side-channel analysis (SCA) is a method by which an attacker circumvents the security assumption by analyzing a physical implementation of the cryptosystem. Unfortunately, as the cryptosystem performs its computations, it will leak certain pieces of information that can reveal security-critical underlying operations. For other cryptosystems, considerable investigation has gone on in regards to the timing, power, and electromagnetic residues that are revealed. Fault-based attacks are also interesting in that they try to make the cryptosystem fail by creating an invalid condition within the system.

Here, we analyze the applications of refined power analysis attacks on SIDH, which is also applicable to other isogeny-based cryptosystems. Our contributions can be summarized as follows:

- We introduce the concept of zero-value attacks in regards to quadratic finite fields.
- We analyze conditions for zero-values within the highly optimized Montgomery curve point and isogeny arithmetic.
- We propose partial-zero and zero-point attacks on the three-point Montgomery ladder.
- We propose the large-degree isogeny analogue of the zero-value attack in the context of SIDH.

2 Preliminaries

This serves as a quick introduction to elliptic curves, isogenies, and side-channel attacks. We point the reader to [16] for a complete look at elliptic curve theory and [17] for a summary of side-channel attacks on elliptic curve cryptography.

2.1 Elliptic Curve Theory

For our case study of elliptic curve formulas, we primarily focus on Montgomery curves [18]. Montgomery curves have been the primary target of SIDH implementations because they feature fast point arithmetic and isogeny operations. A Montgomery [18] curve defined over $\mathbb{F}_q$ can be written as:

$$E/\mathbb{F}_q : by^2 = x^3 + ax^2 + x,$$

where $a, b \in \mathbb{F}_q$ and $b(a^2 - 4) \neq 0$. A Montgomery curve is composed of all points (x, y) that satisfy the above equation as well as the point at infinity. It can

be shown that there is a one-to-one mapping from short Weierstrass curves to a Montgomery curve, so long as the short Weierstrass curve has points of order 4. As demonstrated in [18], this form of the curve allows for extremely efficient differential point additions by utilizing the Montgomery curve's Kummer line $(X : Z)$. By dropping the Y coordinate, this also results in extremely fast isogeny arithmetic, demonstrated in [8,9], making it currently the most efficient choice for SIDH. In addition to Montgomery curves, we also discuss applications to SIDH with short Weierstrass and Edwards [19] curves.

2.2 Isogeny Theory

Isogeny theory analyzes the relationship among various elliptic curves. The j-invariant of an elliptic curve characterizes various properties of a curve and places it into a specific elliptic curve isomorphism class. Over a specific finite field, we can move from one elliptic curve to another by utilizing a rational map over the identity element, or point at infinity. Moving from one elliptic curve to a curve with a different j-invariant is a curve *isogeny* and moving from one elliptic curve to a curve with the same j-invariant is called a curve *isomorphism*.

We formally define an isogeny over a finite field, $\mathbb{F}_q$, as $\phi : \ E \rightarrow E'$ as a non-constant rational map defined over $\mathbb{F}_q$ such that ϕ satisfies group homomorphism from $E(\mathbb{F}_q)$ to $E'(\mathbb{F}_q)$ [16]. SIDH uses isogenies among supersingular elliptic curves rather than their ordinary elliptic curve counterpart as they are more secure. Supersingular elliptic curves feature an endomorphism ring that is isomorphic to an order in a quaternion algebra [16]. Supersingular elliptic curves can be defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$, where p is a prime number. For every prime, $\ell \neq p$, there exist $\ell + 1$ isogenies of degree ℓ from a specific isomorphism class. These isogenies can be computed over a kernel, κ, such that $\phi : \ E \rightarrow E/\langle\kappa\rangle$ by utilizing Vélu's formulas [20]. SIDH efficiently computes large-degree isogenies of the form ℓ^e by decomposing them into a chain of degree ℓ isogenies and computing them iteratively.

2.3 Side-Channel Analysis

Side-channel analysis targets various physical phenomena that are emitted by a cryptographic implementation to reveal critical internal information of the device. Consider the use of gates to perform cryptographic computations as switches of 0's and 1's. Power, timing, and electromagnetic radiation are all emitted as such computations are performed. Simple power analysis (SPA) analyzes a single power signature of a device, while differential power analysis (DPA) statistically analyzes many power runs of a device. Timing analysis targets timing information of various portions of the computation. Electromagnetic radiation can be seen as an extension of power analysis attacks by analyzing electromagnetic emissions instead of power. Lastly, fault attacks attempt to inject a failing condition into the device to attempt to reveal secret information. In general, these attacks require physical access to a device and have been successful in breaking naively constructed cryptosystems.

Refined power analysis (RPA) techniques target computations involving a zero inside a device. Originally introduced by Goubin at PKC, an attacker can maliciously send base points that when pushed through a scalar multiplication produce a point of the form $(x, 0)$ or $(0, y)$ [21]. The conventional wisdom is that although DPA countermeasures produce a different set of intermediate computations, the computations with zero will be unchanged since zero multiplied with anything is zero. By recursively targeting bits of the scalar, an attacker can obtain an implementation's secret key. Later, Akishita and Takagi generalized this to a zero-value attack that targets conditions where a register holds zero [22]. They argue that since a multiplication is composed of a series of cascaded adders and an addition is a long XOR that the power consumption of these operations is significantly smaller when zero is one of the operands. Lastly, Smart notes that countermeasures to zero-point attacks include point blinding, key splitting, and an isogeny to an isomorphism class where there are no longer any zero points [23].

3 Supersingular Isogeny Diffie-Hellman Protocol

3.1 Background

Isogeny-based cryptography was first presented by Rostovtsev and Stolbunov in [24]. This work was based on isogenies of ordinary elliptic curves. The quantum resistance of this work was subsequently broken by Childs et al. [25]. Supersingular isogenies were first presented in the context of collision-resistant hash [26]. Later, Jao and De Feo proposed the isogeny-based cryptosystem to be based on isogenies of supersingular elliptic curves, which has not been shown to be easily broken with quantum computers as a result of the non-commutative endomorphism ring of supersingular elliptic curves [2]. Since then, several implementations of SIDH in both hardware and software have appeared in the literature [8–13].

3.2 SIDH Protocol

The supersingular isogeny Diffie-Hellman key exchange protocol is a public-key cryptosystem by which Alice and Bob can agree on a shared secret. The public parameters include:

- A prime p of the form $\ell_A^a \ell_B^b \cdot f \pm 1$ where ℓ_A and ℓ_B are small primes, a and b are positive integers, and f is a small cofactor
- A supersingular elliptic curve, $E_0(\mathbb{F}_{p^2})$
- A torsion basis $\{P_A, Q_A\}$ of $E_0[\ell_A^a]$ over $\mathbb{Z}/\ell_A^a\mathbb{Z}$ and a torsion basis $\{P_B, Q_B\}$ of $E_0[\ell_B^b]$ over $\mathbb{Z}/\ell_B^b\mathbb{Z}$

From these public parameters, the general idea of the protocol is that Alice and Bob perform separate walks on isogeny graphs of degree ℓ_A^a and ℓ_B^b, respectively, by computing a large-degree isogeny over a secret kernel. The security assumption is based on the difficulty of computing an isogeny between supersingular

elliptic curves, for which there is no subexponential algorithm known even for quantum computers. Alice generates private keys $m_A, n_A \in \mathbb{Z}/\ell_A^a\mathbb{Z}$ both not divisible by ℓ_A^a and Bob likewise generates private keys $m_B, n_B \in \mathbb{Z}/\ell_B^b\mathbb{Z}$ both not divisible by ℓ_B^b. The protocol consists of two rounds that can be broken down to:

1. Computing a secret kernel $R = \langle [m]P + [n]Q \rangle$ for torsion basis points $\{P, Q\}$, where m and n are private keys
2. Computing an isogeny over that secret kernel, $\phi : E \to E/\langle R \rangle$, using Vélu's formulas for a supersingular curve E
3. Computing the images of the other party's torsion basis, $\{\phi(P_{opp}), \phi(Q_{opp})\}$, for the first round.

Thus, for the first round, Alice and Bob perform the isogenies $\phi_A : E_0 \to E_A = E_0/\langle [m_A]P + [n_A]Q \rangle$ and $\phi_B : E_0 \to E_B = E_0/\langle [m_B]P + [n_B]Q \rangle$, respectively. They each also apply the isogeny to the other party's torsion basis. After the first round, Alice sends $(E_A, \{\phi_A(P_B), \phi_A(Q_B)\})$ and Bob sends $(E_B, \{\phi_B(P_A), \phi_B(Q_A)\})$ over a public channel. The second round consists of a similar isogeny computation, but over the exchanged public keys. Alice performs $\phi'_A : E_B \to E_{AB} = E_B/\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle$ and Bob performs $\phi'_B : E_A \to E_{BA} = E_A/\langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \rangle$. At this point, Alice and Bob have isomorphic curves since they separately performed a specific traversal of isogeny graphs of ℓ_A^a and ℓ_B^b, respectively, with their secret kernel construction. Since the resulting curves are isomorphic, the common j-invariant can be used as a shared secret [2].

3.3 SIDH Protocol Optimizations

Above, we recited the proper SIDH protocol. However, most of the implementations in the literature [7,9,11–13] take advantage of a few simplifications to the computations to make them more efficient. Notably, instead of performing a full double-point multiplication, $[m]P + [n]Q = R$, it is assumed that either m or n is 1. As noted in [9], any generator of $[m]P + [n]Q$ will produce a valid secret kernel. Thus, by assuming that m or n is invertible modulo the order of the group, $P + [m^{-1}n]Q = P + [m]Q$ is also a valid generator of all possible kernels. In terms of Montgomery curves, this simplification allows the use of a three-point Montgomery differential ladder [9], which is shown in Algorithm 1.

The three-point Montgomery differential ladder produces $[x]Q$, $[x+1]Q$, and $[x]Q+P$ at the end of each step. Thus, with the differentials Q and $Q-P$, we can take advantage of the efficient differential addition formulas over Montgomery curves. Although there have not yet been any SIDH implementations over other curves, it can be assumed that the above simplification would also be taken advantage of. However, instead of performing a double-point multiplication as above, the standard Montgomery ladder could be utilized to compute $[m]Q$, after which a simple projective addition would be performed to obtain $[m]Q + P$.

Otherwise, as originally proposed by [9], the majority of known implementations in the literature all feature primes of the form $2^a 3^b \cdot f - 1$. Over the

Algorithm 1 . Three-point differential ladder to compute $P + [t]Q$ [2]. "dadd$(P, Q, (P - Q).x)$" represents a differential point addition of P and Q, where the x-coordinate of $P - Q$ is known.

Input: Points P and Q on an elliptic curve E, scalar d which is k bits
1: Set $A = 0, B = Q, C = P$
2: Compute $Q - P$
3: **for** i decreasing **from** $|d|$ **downto** 1 **do**
4: Let d_i be the i-th bit of d
5: **if** $d_i = 0$ **then**
6: $B =$dadd(A, B, Q), $C =$dadd(A, C, P), $A = 2A$
7: **else**
8: $A =$dadd(A, B, Q), $C =$dadd$(B, C, Q - P)$, $B = 2B$
9: **end if**
10: **end for**
Ensure: $C = P + [t]Q$

Montgomery Kummer arithmetic, [8,9] produced efficient formulas to compute isogenies and apply isogenies of degree 2 and 3. Thus, we focus on this particular case, but the attacks we propose can easily be generalized to other isogeny degree bases.

4 Refined Power Analysis Model for SIDH

Here, we create a power analysis model to describe how the zero-point attack could be applied to SIDH.

4.1 Targeting Static Keys in SIDH

As originally proposed in [21], the zero-point attack is a form of differential power analysis, and thus, requires many runs of a device over the same key. At the time of its conception, this attack could be mounted against users with a long-term static key in ECDH, ECIES, and ECMQV (which is now broken). Based on the security assumption of supersingular isogenies, there is currently only an analogue to the SIDH with a user using a long-term static key. Here, we target the second round of SIDH, where Alice will compute the secret kernel point $R = P+[n]Q$ and perform the subsequent isogeny. SIDH is, in a sense, more dangerous than ECDH since the other party sends $\phi(P)$, $\phi(Q)$, and the supersingular elliptic curve E'. Not only does a malicious third-party get to choose two points to send over, they also can control which supersingular elliptic curve these points will lie on. From here on, we will assume that Alice has a long-term static key n_A and receives the public key tuple, $\{\phi_B(P_A), \phi_B(Q_A), E_B\}$ from Oscar and attempts to compute the shared secret $j(E_{AB}) = j(E_B/\langle\phi_B(P_A) + [n_A]\phi_B(Q_A)\rangle)$. For generality, the scalar n_A could apply to either $\phi_B(P_A)$ or $\phi_B(Q_A)$ and Alice does not specify.

In [8], Costello et al. introduce a method to validate the public keys sent over a public channel. This validation includes verifiying that the curve E_B is supersingular, of the proper cardinality, and is in the right supersingular isogeny class, as well as validating that the transmitted torsion basis points have the correct order and are independent. As they show in the results, the public key validation in [8] is rather expensive, and consumes approximately 40% of the time of a single round of the protocol.

As demonstrated by Galbraith et al. in [27], there is a simple adaptive oracle attack on a user with long-term static keys. Oscar will send public keys with maliciously crafted torsion basis points that will only match Oscar's shared secret oracle if the bits of Alice's keys are guessed correctly. Thus, over approximately $\log_2 p$ oracle queries, Oscar will have Alice's private key.

This above attack has been shown to bypass the public key validation proposed in [8], but fails to pass the Kirkwood et al. validation model [28] that ensures Oscar is producing public keys honestly. By utilizing a seed to a pseudo-random number generator to generate his private keys, Oscar must use Alice's public key to first generate the shared secret. Using this shared secret, Oscar will encrypt his PRNG seed and include it to Alice. From Alice's perspective, she will utilize Oscar's public keys to generate a shared secret and will retrieve Oscar's PRNG seed by decrypting it with the shared secret. Then, Alice will perform Oscar's computations with the derived private keys. If the public keys do not match those that Oscar sent, then Alice rejects the key-exchange since Oscar is not acting honestly.

We provide the above validation methods to analyze the additional overhead that a static key user must consider in return for increased security. The public key validation method ensures that the public keys *appear* valid at the cost of about 40% of a round, but still does not prevent the Galbraith et al. adaptive oracle attack. The Kirkwood et al. validation model does prevent the oracle attack and perhaps other dishonest public key attacks, but Alice must perform an additional round of SIDH. Thus, if Alice, the static-key user, decides to perform both of these validations, she must perform an additional 140% work (could be more if Oscar's isogeny computations are much more computationally intensive) as well as have any additional hardware or registers to support the additional functionality. Indeed, this overhead is much more than that of ECC, but certain devices may not be able to support or guarantee the security of an on-device random number generator, for instance.

In terms of the SIDH protocol, we recommend Alice to include both of these validations. We note that by itself, the Kirkwood et al. validation model will automatically start computations over the transmitted public keys. From a side-channel analysis perspective, this is incredibly weak as the public keys could produce any number of vulnerabilities. First, invalid torsion basis points can produce a kernel point that is not of the correct order, that a device might not handle gracefully. Second, Oscar can manipulate the torsion points that produce special points of interest (such as zero-points). We propose a simple attack of this in the following section. Third, an invalid elliptic curve can also produce

intermediate values of interest through manipulation. These are only some of the attacks that could be mounted if public key validation is used. Thus, public key validation serves as the primary defense against certain types of power analysis and fault attacks, while the Kirkwood et al. validation method serves as the primary defense against maliciously chosen, but valid public keys.

4.2 Zero-Value Representations in Quadratic Fields

First, we define a representation of zero in terms of a quadratic extension field, $\mathbb{F}_{p^2}$, which is the underlying finite prime field used in SIDH. Let $A, B \in \mathbb{F}_{p^2}$ such that $A = a_1x + a_0, B = b_1x + b_0$ and $a_1, a_0, b_1, b_0 \in \mathbb{F}_p$. We define an irreducible polynomial over this finite field of the form $x^2 + \alpha x + \beta$. We then define addition and multiplication with A and B as:

$$A + B = (a_1 + b_1)x + (a_0 + a_1) \tag{1}$$

$$A \times B = (a_0b_1 + a_1b_0 - \alpha a_1b_1)x + (a_0b_0 - \beta a_1b_1) \tag{2}$$

However, the known implementations in the SIDH literature utilize the primes of the form $2^a3^b \cdot f - 1$, for which -1 does not have a square root, so $x^2 + 1$ is an irreducible polynomial. The new multiplication formula becomes:

$$A \times B = (a_0b_1 + a_1b_0)x + (a_0b_0 - a_1b_1) \tag{3}$$

$$A \times B = (a_0b_1 + a_1b_0)x + ((a_0 + a_1)(b_0 - b_1) + a_0b_1 - a_1b_0) \tag{4}$$

We included Eq. (4) as the efficient way to perform the multiplication in $\mathbb{F}_{p^2}$, since we are only performing 3 multiplications in $\mathbb{F}_p$ rather than 4. We primarily focus on Eqs. (3) and (4), but further generalizations can be easily made. We give these equations to show that the behavior of zero will change slightly in $\mathbb{F}_{p^2}$. Interestingly, as the above equations show, both resulting $\mathbb{F}_p$ values from the multiplication in the extension field is dependent on all four input $\mathbb{F}_p$ values (a_1, a_0, b_1, b_0). Further, it is interesting to note that in the case of a squaring, the most significant element in $\mathbb{F}_{p^2}$ will only be zero if and only if the input element also has a most significant element of zero (since p is a prime number). We define the element A as being *fully zero* if $a_1, a_0 = 0$. We also define A as being *partially zero* if exactly $a_1 = 0$ or exactly $a_0 = 0$. Any other combinations for A are *non-zero*.

Consider that in projective coordinates, the x-coordinate is scaled by a Z value, i.e. $(x, y) \rightarrow (X : Y : Z)$ where $x = X/Z$ and $y = Y/Z$. In the SIDH scenario, as can be observed from Eqs. (3) and (4), a partially zero x-coordinate guaranteed produces a non-zero X-coordinate if scaled by a Z-coordinate that is non-zero. Non-zero Z-coordinates will only produce a partially zero X-coordinate if exactly $a_0b_1 = -a_1b_0$ or $a_0b_0 = a_1b_1$. If Alice performs her random curve isomorphism or randomizes the projective input coordinates, then Oscar has little control over what values of Z Alice will be using at various iterations of the scalar point multiplication.

The primary conclusion from above is that targeting partial zero values in the case of projective points is not very beneficial. Instead, an attacker can target the fully zero values since the projective representation of quadratic extension fields will not change these. For Montgomery [18] curves, the point with a zero x-coordinate is $(0, 0)$, which has order 2. Edwards [19] curves contain the point $(0, 1)$, which is the neutral element of the addition law, the point $(0, -1)$, which has order 2, and the points $(1, 0)$ and $(-1, 0)$, which have order 4. Lastly, short Weierstrass curves may have special points of the form $(0, \sqrt{b})$ if the square root of b exists and the special point $(x, 0)$ of order 2 if there is a solution to $x^3 + ax + b = 0$ [21]. Although the zero-point is not guaranteed for a specific short Weierstrass curve, one can apply an isogeny to an isomorphism class where the zero-points do exist. Thus, since Oscar can choose the supersingular elliptic curve and corresponding basis points, he can always choose a curve where there is a zero-point.

4.3 Zero-Values in Montgomery Curve Arithmetic

As proposed by [22], an implementation's arithmetic unit can be targeted to determine the existence of a zero-register. In the case of the quadratic extension field, arithmetic is primarily done in the base field. Thus, we target any partially-zero or fully-zero values that may be produced by the curve arithmetic.

In this work, we analyze the fastest SIDH arithmetic available in the literature, which is introduced in [8]. This work takes advantage of the fast Montgomery differential arithmetic for scalar point multiplication as well as fast projective isogenies of degree three and four. Table 1 contains a summary of the arithmetic. This arithmetic has been developed to work over the projectivized isogeny form of the Montgomery curve:

$$E_{(A:B:C)} : \; By^2 = Cx^3 + Ax^2 + Cx$$

Which can be converted to the original Montgomery curve form in the preliminaries with the relations: $a = \frac{A}{C}$, $b = \frac{B}{C}$. Here, C is a projectivized constant of the Montgomery curve to allow for projective isogeny formulas. Note that "get_iso" refers to computing an isogeny and "eval_iso" refers to pushing a point from one elliptic curve to its targeted isogenous curve. In the equations in Table 1, assume that (X_2, Z_2) and (X_3, Z_3) are input points P and Q for addition and doubling, (X_1, Z_1) is the normalized coordinate for $P - Q$, and $A_{24} = (A + 2)/4$. $(X_4, Z_4) = 2(X_2, Z_2)$ and $(X_5, Z_5) = (X_2, Z_2) + (X_3, Z_3)$. (P_{X3}, P_{Z3}) and (P_{X4}, P_{Z4}) are kernel points of order 3 and 4, respectively.

From this table we point out a few interesting calculations that could be used in a zero-value attack.

In terms of the double and addition formula, we point out the following calculations:

1. $X_2 + Z_2 = Z_2(x_2 + 1)$
2. $X_2 - Z_2 = Z_2(x_2 - 1)$

Table 1. Summary of projective Montgomery curve arithmetic from [8]

Operation	Equation
xDBL	$X_4 = (X_2 + Z_2)^2(X_2 - Z_2)^2$
	$Z_4 = (A_{24}((X_2 + Z_2) - (X_2 - Z_2)^2)$
	$+(X_2 + Z_2)^2)((X_2 + Z_2)^2 - (X_2 - Z_2)^2)$
xADD	$X_5 = ((X_2 + Z_2)(X_3 - Z_3) + (X_2 - Z_2)(X_3 + Z_3))^2$
	$Z_5 = X_1((X_2 + Z_2)(X_3 - Z_3) - (X_2 - Z_2)(X_3 + Z_3))^2$
get_iso_3	$(A', C') = (P_{Z3}^4 + 18P_{X3}^2P_{Z3}^2 - 27P_{X3}^4 : 4P_{X3}P_{Z3}^3)$
eval_iso_3	$(X', Z') = (X(P_{X3}X - P_{Z3}Z)^2 : Z(P_{Z3}X - P_{X3}Z)^2)$
get_iso_4	$(A', C') = (2(2P_{X4}^4 - P_{Z4}^4) : P_{Z4}^4)$
eval_iso_4	$X' = X(2P_{X4}P_{Z4}Z - X(P_{X4}^2 + P_{Z4}^2))(P_{X4}X - P_{Z4}Z)^2$
	$Z' = Z(2P_{X4}P_{Z4}X - Z(P_{X4}^2 + P_{Z4}^2))(P_{Z4}X - P_{X4}Z)^2$

We can expect to see a zero in an intermediate register holding this result if either $Z_2 = 0$, $x_2 = 1$, or $x_2 = -1$. $Z_2 = 0$ implies that we are trying to double the point at infinity, which is not expected in a valid run of this protocol. $x_2 = \pm 1$ is an interesting target point for the ladder since it will produce an intermediate zero. However, these points are not guaranteed on a Montgomery curve. For the standard curve equation, these points exist if there is a corresponding y that satisfies $(1, \sqrt{\frac{A+2}{B}})$ or $(-1, \sqrt{\frac{A-2}{B}})$. Roughly, this is a check if the square root exists in the underlying quadratic field to form a point.

Similarly, the differential addition formula utilizes:

1. $X_2 + Z_2 = Z_2(x_2 + 1)$
2. $X_2 - Z_2 = Z_2(x_2 - 1)$
3. $X_3 + Z_3 = Z_3(x_3 + 1)$
4. $X_3 - Z_3 = Z_3(x_3 - 1)$

Similar to the doubling formula, we can expect to see an intermediate zero if $Z_2 = 0$, $Z_3 = 0$, $x_2 = \pm 1$, or $x_3 = \pm 1$. If one of the intermediate Z values is 0, then we are adding with the point at infinity. We pinpoint these computations, since we can target the $x = \pm 1$ at the double-point multiplication level or at the large-degree isogeny level. The hidden kernel point is continously tripled (double and add) when computing an isogeny of base degree 3 or quadrupled (double and double) when computing an isogeny of base degree 4.

The isogeny formulas are only used in the large-degree isogeny computation that finishes the round. As was previously mentioned, an isogeny of a base degree is computed over a kernel and then any points on the old curve are converted to the new one through an isogeny evaluation. For computing an isogeny of degree 3, we can target:

1. $P_{Z3}^4 + 18P_{X3}^2P_{Z3}^2 - 27P_{X3}^4 = P_{Z3}^4(1 + 18P_{x3}^2 - 27P_{x3}^4) = P_{Z3}^4(1 + 9(2P_{x3}^2 - 3P_{x3}^4))$
2. $4P_{X3}P_{Z3}^3 = 4P_{Z3}^4(P_{x3})$

As the first equation shows, we will have a zero in the equation for A' if $P_{Z3} = 0$, $27P_{x3}^4 - 18P_{x3}^2 - 1 = 0$, or $3P_{x3}^4 - 2P_{x3}^2 = 0$. If $P_{Z3} = 0$, then we are using the point at infinity, which does not have order 3 and is an invalid isogeny kernel. The solutions to Eq. 1 are $P_{x3} = \pm\frac{1}{3}\sqrt{3+2\sqrt{3}}, \pm\frac{1}{3}\sqrt{-3-2\sqrt{3}}$ and the solutions to Eq. 2 are $P_{x3} = 0, \pm\sqrt{\frac{2}{3}}$. However, if $P_{x3} = 0$, then we are using the point $(0, 0)$ which has order 2, not 3, again invalidating the isogeny computation. The equation for C' is zero if either $P_{Z3} = 0$ or $P_{x3} = 0$, which are again invalid kernels, which is to be expected since a C coefficient of zero means that the curve does not exist.

For eval_iso_3:

1. $P_{X3}X - P_{Z3}Z = ZP_{Z3}(P_{x3}x - 1)$
2. $P_{Z3}X - P_{X3}Z = ZP_{Z3}(x - P_{x3})$

In the first case, $P_{x3}x - 1 = 0$ means that $P_{x3} = x^{-1}$. If either Z value is zero, then we are attempting to apply the isogeny to the point at infinity, which will again produce the point at infinity. For the second case, $x = P_{x3}$ implies that we are attempting to push the same point as our kernel point to the new curve, which will result in the point at infinity.

For get_iso_4:

1. $2P_{X4}^4 - P_{Z4}^4 = P_{Z4}^4(2P_{x4}^4 - 1)$

Here, we only look at $2P_{x4}^4 = 1$ to produce a zero value for A'. The only valid solutions are $P_{x4} = \pm\frac{1}{\sqrt{2}}$ and $P_{x4} = \pm\frac{1}{\sqrt{-2}}$.

Lastly, we summarize eval_iso_4 in-line:

1. $P_{X4}^2 + P_{Z4}^2 = P_{Z4}^2(P_{x4}^2 + 1) \implies P_{x4} = \pm\sqrt{-1}$
2. $2P_{X4}P_{Z4}Z - X(P_{X4}^2 + P_{Z4}^2) = P_{Z4}^2 Z(2P_{x4} - x(P_{x4}^2 + 1)) \implies P_{x4} = \pm\frac{\sqrt{4x^2+9}-3}{2x}$, OR $P_{x4} = x = 0$
3. $2P_{X4}P_{Z4}X - Z(P_{X4}^2 + P_{Z4}^2) = P_{Z4}^2 Z(2xP_{x4} - (P_{x4}^2 + 1)) \implies P_{x4} = \frac{1}{2}(3x \pm \sqrt{9y^2 + 4})$
4. $P_{X4}X - P_{Z4}Z = P_{Z4}Z(P_{x4}x - 1) \implies P_{x4} = x^{-1}$
5. $P_{Z4}X - P_{X4}Z = P_{Z4}Z(x - P_{x4}) \implies x = P_{x4}$ (Evaluating point same as kernel point)

5 Proposed Partial-Zero Attack on Three-Point Ladder

Here, we describe a simple attack on the three-point differential ladder proposed by Jao et al. in [9] and shown in Algorithm 1.

5.1 Partial-Zero Attack Targeting Differential Addition

Depending on the bit of the key we perform the following computations:

- if $d_i = 0$, then $C = \text{dadd}(A, C, P)$
- if $d_i = 1$, then $C = \text{dadd}(B, C, Q - P)$

In particular, we direct our attention to the differential point, either P or $Q-P$. An attacker may have little control over the projective coordinates based on the quadratic multiplication, but it has been typical to use a normalized differential point, i.e. $P = (x, y)$, for speed, so Oscar will know which values for $P.x$ and $(Q-P).x$ are generated. By determining a combination of P and Q that produces a non-zero $P.x$ and a partially-zero $(Q-P).x$, Oscar has created an oracle for each iteration of the three-point ladder, since a multiplication by zero will be observed if $(Q-P).x$ is used whereas a typical power observation will be observed for the non-zero $P.x$. Depending on the multiplication arithmetic in the implementation, Oscar can extract the entire key from Alice in a single attempt if there is a stark enough contrast between multiplying by $P.x$ and the partially-zero $(Q-P).x$.

Thus in the case of attacking a static-key SIDH user, let us assume that Oscar is attempting to find such a curve and valid torsion basis that can mount this attack. Initially, Oscar can perform a few walks on the graph of his supposed isogeny graph of degree ℓ_B. As he walks the isogeny graph, he computes the image of Alice's torsion basis, $\{P_A, Q_A\}$, as well as their difference, $(Q_A - P_A)$ on this new isogenous curve, to preserve a valid torsion basis. From here, Oscar checks if a valid elliptic curve isomorphism can convert either P_A or $(Q_A - P_A)$, but not both, to an affine coordinate with a partially-zero x-coordinate. If the isomorphism class does not have an available curve, then Oscar performs another walk on the isogeny graph of degree ℓ_B to an isomorphism class that may have the required condition.

5.2 Countermeasures

To thwart this attack, a static-key user can merely reject any torsion bases that produce a normalized P or $(Q-P)$ x-coordinates that are partially-zero. Otherwise, using a random projectivization of these differential coordinates would thwart the attack as long as it does not create a partially-zero result. Projectivizing the differential coordinates comes at the cost of two additional multiplication per step of the three-point ladder. Lastly, any other methods that would alter the representation of this partially-zero value would also thwart the attack, such as a random initial isomorphism.

6 Proposed Zero-Point Attack on Three-Point Ladder

Here, we apply the zero-point attack to the three-point differential ladder presented in [9] in a procedure that is similar to that produced in [21].

6.1 Zero-Point Attack with Points of Large Order

The three-point differential ladder computes $P + [n]Q$ with input points P, Q and $(P-Q).x$ is known. At the end of the ith step of the ladder, the following points are computed:

$$[x]Q = (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + d_i).Q$$

$$[x+1]Q = (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + d_i + 1).Q$$

$$P + [x]Q = P + (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + d_i).Q$$

Thus, it is simple to see that the $(i+1)$ step will produce the following values:

- $d_i = 0$ will always produce $(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 1).Q$ and then $(\sum_{j=i+1}^{n-1} d_j 2^{j-i}).Q$, $P + (\sum_{j=i+1}^{n-1} d_j 2^{j-i}).Q$ if $d_{i+1} = 0$ or $(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 2).Q$, $P + (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 1).Q$ if $d_{i+1} = 1$.
- $d_i = 1$ will always produce $(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 3).Q$ and then $(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 2).Q$, $P + (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 2).Q$ if $d_{i+1} = 0$ or $(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 4).Q$, $P + (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 3).Q$ if $d_{i+1} = 1$.

Next, we target the points that will always be produced by the guess of d_i. Let P_0 be a special point where the x-coordinate or y-coordinate is 0, which must be $(0, 0)$ for a Montgomery curve. However, rather than continuing with Goubin's methodology, we note that performing a scalar multiplication with a point of order 2 will either produce itself if the scalar is odd or the point at infinity if the scalar is even. Roughly, we need to find a point P_1 that satisfies the equation $P_0 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 1).P_1$ if we believe that $d_i = 0$ or $P_0 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 3).P_1$ if we believe that $d_i = 1$. Based on this setup, we know that P_1 is a point with order $2(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 1)$ if $d_i = 0$ or $2(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 3)$ if $d_i = 1$.

Thus, since such points have an invalid order, they will not pass the public-key validation. We propose instead to find curves with points $P_0 = (\pm 1, y)$ with a large order and solve for P_1 in the same way. After finding an appropriate point P_1, Alice will compute her shared secret and may produce the special point of interest, revealing bit i. The point with $x = \pm 1$ is interesting, as we noted that it would produce a zero condition when analyzing the Montgomery arithmetic. As noted in [21], this process is recursively repeated to reveal Alice's entire secret key. We note that although three points are used in this differential point ladder, we still target the points $(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 1).Q$ if we guess that $d_i = 0$ and $(\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 3).Q$ if we guess that $d_i = 1$, as was done in Goubin's original analysis [21].

As is shown above, the zero-point attack will not work against a static-key user that is validating public keys. However, this is primarily because the Montgomery curve arithmetic only uses the x-coordinate to perform a scalar point multiplication and there is only a single zero-point with order 2. Short Weierstrass curves, on the other hand, may have a point, $P_0 = (0, \sqrt{b})$. This

point does not have a specific order, thus Oscar can use isogenies and isomorphisms to force this point to have his desired order for the attack. In order to bypass the public-key validation, Oscar finds a point P_1 of the proper order as specified by the SIDH parameters such that $P_0 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 1).P_1$ or $P_0 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + 3).P_1$. In this case, Alice may produce the point of interest and Oscar discovers another bit of Alice's key. The difference here is that, this attack may succeed even in the case of public-key validation.

6.2 Countermeasures

The most noteworthy countermeasures to these zero-point attacks in the context of ECDH include an isogeny to a curve where the zero-point doesn't exist, randomization of the private exponent, and point blinding [23]. However, in regards to SIDH, we note that performing an initial random isogeny will change the resulting isomorphism class, but will work if the degree of the random isogeny is not ℓ_A or ℓ_B. Further, in the context of the Kirkwood et al. validation model, Alice will not know which random isogeny Oscar performed, so Oscar must perform a final isogeny in the reverse direction of the random initial isogeny to provide valid public keys.

7 Proposed Refined Power Analysis on Large-Degree Isogenies

Here, we discuss an analog of these zero-value attacks to large-degree isogenies. Roughly, we show that the iterative nature of the large-degree isogenies can be attacked by forcing zero conditions.

7.1 Using RPA on SIDH

As is shown in Fig. 1, the large-degree isogeny of a base degree can be visualized as traversing a complete graph where the vertices represent isomorphism classes and the edges represent isogenies. Each isomorphism class has $\ell + 1$ connecting isomorphism classes. From an initial isomorphism class, there are $\ell + 1$ possible isogenies of degree ℓ to a new isomorphism class. After that, we do not go backwards on an isogeny walk, so there are ℓ possible isogenies at every vertex after that. In this context, we are trying to determine which path Alice takes through the isogeny graph, rather than determine bits of Alice's key. The difficulty to compute a path between two distant isomorphism classes is considered to be inefficient even in the context of quantum computers, so each time an isogeny decision is revealed, the problem becomes that much easier.

We consider the idea of revealing that path through forced zero values. As is specified in the preliminaries, the large-degree isogeny is performed iteratively; we take a kernel point of sufficient order and iteratively perform a single walk on the isogeny graph. As we perform these isogenies of a base degree, we apply the isogeny to other stored multiples of the kernel point. For the first round, we also

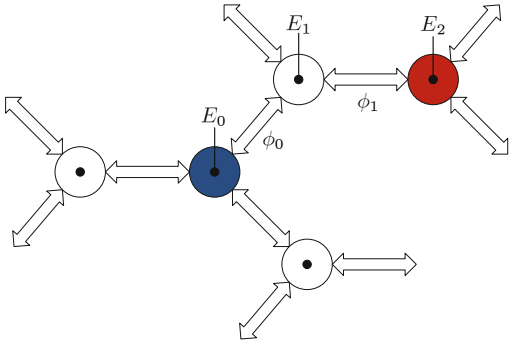

Fig. 1. Graph representing the space of all isogenies of degree 2 under a given field, $\mathbb{F}_{p^2}$. The vertices (circles) represent an isomorphism class, of which all curves within the class share the same j-invariant. The blue circle represents the initial supersingular elliptic curve isomorphism class of the isogeny. In SIDH, Oscar can choose which isomorphism class to send Alice. The red circle indicates the targeted path that Oscar is trying to determine. In this scenario, Oscar has discovered ϕ_0 and must subsequently determine ϕ_1 by injecting a zero condition into the two possibilities for E_2. This process is repeated iteratively to reveal Alice's static key.

apply the isogeny to the other party's basis. As we compute an isogeny, we are determining the coefficients for a new curve, thus we call refined power analysis attacks targeting curve coefficeints *zero-value isogeny coefficient attacks*. As we apply an isogeny, we are determining the representation of that point on the new curve, thus we call refined power analysis attacks targeting particular isogenous points *zero-value isogeny point attacks*.

Let us assume that Alice takes e_A walks on the isogeny graph of base ℓ_A starting at the supersingular curve E_0. We number these walks $\phi_0, \phi_1, \cdots \phi_{e_A-1}$. Thus, as is shown in Fig. 1 for $\ell_A = 2$, $\phi_0 : \ E_0 \rightarrow E_1$ and so on. Our goal is to determine which neighboring node isogeny ϕ_0 utilized. Since Vélu's formulas are deterministic and we know what elliptic curve Alice will start on, we can determine the $\ell + 1$ possible isogenous curves where Alice will end up. We can then target these elliptic curves by forcing a zero condition in one or more of the neighboring vertices. If this zero condition is experienced on the computation of ϕ_0 or becomes a coefficient or point coordinate in calculating ϕ_1, then we can confirm or reject some of the possible isogenies. After we have identified ϕ_0, we next target ϕ_1, which will have ℓ possibilities. From there, we iteratively use i known isogenies and target the $(i + 1)$ isogeny until we have discovered the entire isogeny path. Next, we explain this further in the context of zero-value coefficient and point attacks.

7.2 Zero-Value Isogeny Coefficient Attack

The first attack we look at is if a zero-value curve coefficient is produced from an isogeny. As is noted in Sect. 1, there are several ways to produce a zero-value for A' in the context of computing an isogeny of degree 3 or degree 4.

However, Oscar has little knowledge of the computation of the kernel point, so it is not easy to target the point of order 3 or 4. Instead, Oscar can target some curve in any isomorphism class that produces an isogenous curve with $A' = 0$. In this case, Oscar is checking if this edge of the isogeny graph is traversed by checking if a zero is experienced. In the case of Montgomery curves, the constant $A_{24} = (A+2)/4$ is used to perform point doubling. Thus, if Oscar can determine the power trace of an addition by zero, he can reveal information about the isogeny path. With Montgomery curves, the calculation of A_{24} is the only direct usage of A as the other formulas for computing and evaluating an isogeny do not utilize the Montgomery curve coefficients. This may not be the case for other optimized isogeny formulas for Montgomery and other curves.

7.3 Zero-Value Isogeny Point Attack

This attack pinpoints when applying an isogeny to a kernel point or basis point produces a zero-value. In the context of SIDH, Oscar has little control over intermediate representations of the kernel point, but can trick Alice to using his own torsion basis points in the first round of SIDH if Alice agrees to non-standardized parameters. Outside of SIDH, this could be interesting to other applications of supersingular isogenies that require applying the isogeny to points from another party. Anyways, the key here is to pick maliciously crafted torsion points that reveal a zero when pushed through the isogeny. Again, Oscar can determine all nearby curves with the deterministic Vélu's formulas, so he will know a few of the options that Alice will produce. In the context of Montgomery curves, the point $(0, 0)$ is not an option since that point will always be pushed to $(0, 0)$ on other Montgomery curves. However, in the context of other curve forms, this attack could again be interesting, as one can target the special points $(0, y)$ or $(x, 0)$ if they exist.

7.4 Countermeasures

The zero-value attack on isogenies requires knowledge of the nearby isogenous curves. Thus, anything that randomizes the resulting isogenous curves, such as performing a random curve isomorphism or an initial isogeny of a degree $\ell_r \neq \ell_A, \ell_B$, will defeat this assumption, since the scaling of the curve will produce different isogenous curves.

8 Conclusion

In this paper, we investigated refined power analysis attacks and their application to the supersingular isogeny Diffie-Hellman key exchange protocol. As we have shown, there are a few caveats to using zero-value attacks over quadratic extension fields and in Montgomery curve arithmetic. Nevertheless, we have proposed three different zero-value attacks on SIDH that can target static-key users. Since the Kirkwood et al. validation model does not protect against side-channel

attacks, the attacks proposed in this paper continue to question the safety of a static-key user in SIDH. The dual computations of a double-point multiplication and large-degree isogeny in the context of an elliptic curve and points that another party sends over is especially dangerous. As we move forward, it is necessary to survey the effectiveness of the attacks proposed here and any new side-channel attacks that are found in the future.

Acknowledgment. The authors would like to thank the reviewers for their comments. This work is supported in parts by the grants NIST-60NANB17D184, NIST-60NANB16D246, and NSF CNS-1661557.

References

1. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science (FOCS 1994), pp. 124–134 (1994)
2. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
3. Jao, D., Soukharev, V.: Isogeny-based quantum-resistant undeniable signatures. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 160–179. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_10
4. Yoo, Y., Azarderakhsh, R., Jalali, A., Jao, D., Soukharev, V.: A Post-Quantum Digital Signature Scheme Based on Supersingular Isogenies. Cryptology ePrint Archive, Report 2017/186 (2017)
5. Galbraith, S.D., Petit, C., Silva, J.: Signature Schemes Based On Supersingular Isogeny Problems. Cryptology ePrint Archive, Report 2016/1154 (2016)
6. Azarderakhsh, R., Jao, D., Kalach, K., Koziel, B., Leonardi, C.: Key compression for isogeny-based cryptosystems. In: Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography. AsiaPKC 2016, pp. 1–10. ACM, New York (2016)
7. Costello, C., Jao, D., Longa, P., Naehrig, M., Renes, J., Urbanik, D.: Efficient compression of SIDH public keys. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 679–706. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_24
8. Costello, C., Longa, P., Naehrig, M.: Efficient algorithms for supersingular isogeny Diffie-Hellman. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 572–601. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_21
9. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Math. Cryptol. **8**(3), 209–247 (2014)
10. Azarderakhsh, R., Fishbein, D., Jao, D.: Efficient Implementations of a Quantum-Resistant Key-Exchange Protocol on Embedded Systems. Technical report, University of Waterloo (2014)
11. Koziel, B., Jalali, A., Azarderakhsh, R., Jao, D., Mozaffari-Kermani, M.: NEON-SIDH: efficient implementation of supersingular isogeny Diffie-Hellman key exchange protocol on ARM. In: Foresti, S., Persiano, G. (eds.) CANS 2016. LNCS, vol. 10052, pp. 88–103. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_6

12. Koziel, B., Azarderakhsh, R., Mozaffari-Kermani, M.: Fast hardware architectures for supersingular isogeny Diffie-Hellman key exchange on FPGA. In: Dunkelman, O., Sanadhya, S.K. (eds.) INDOCRYPT 2016. LNCS, vol. 10095, pp. 191–206. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49890-4_11
13. Koziel, B., Azarderakhsh, R., Kermani, M.M., Jao, D.: Post-quantum cryptography on FPGA based on isogenies on elliptic curves. IEEE Trans. Circuits Syst. I Regul. Pap. **64**(1), 86–99 (2017)
14. Gélin, A., Wesolowski, B.: Loop-abort faults on supersingular isogeny cryptosystems. In: Lange, T., Takagi, T. (eds.) PQCrypto 2017. LNCS, vol. 10346, pp. 93–106. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59879-6_6
15. Ti, Y.B.: Fault attack on supersingular isogeny cryptosystems. In: Lange, T., Takagi, T. (eds.) PQCrypto 2017. LNCS, vol. 10346, pp. 107–122. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59879-6_7
16. Silverman, J.H.: The Arithmetic of Elliptic Curves. GTM, vol. 106. Springer, New York (1992). https://doi.org/10.1007/978-0-387-09494-6
17. Fan, J., Guo, X., Mulder, E.D., Schaumont, P., Preneel, B., Verbauwhede, I.: State-of-the-art of secure ECC implementations: a survey on known side-channel attacks and countermeasures. In: 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 76–87, June 2010
18. Montgomery, P.L.: Speeding the pollard and elliptic curve methods of factorization. Math. Comput. **48**(177), 243–264 (1987)
19. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards curves. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 389–405. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68164-9_26
20. Vélu, J.: Isogénies entre courbes elliptiques. Comptes Rendus de l'Académie des Sci. **273**, A238–A241 (1971). Paris Séries A-B
21. Goubin, L.: A refined power-analysis attack on elliptic curve cryptosystems. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 199–211. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_15
22. Akishita, T., Takagi, T.: Zero-value point attacks on elliptic curve cryptosystem. In: Boyd, C., Mao, W. (eds.) ISC 2003. LNCS, vol. 2851, pp. 218–233. Springer, Heidelberg (2003). https://doi.org/10.1007/10958513_17
23. Smart, N.P.: An analysis of Goubin's refined power analysis attack. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 281–290. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45238-6_23
24. Rostovtsev, A., Stolbunov, A.: Public-Key Cryptosystem Based on Isogenies. Cryptology ePrint Archive, Report 2006/145 (2006)
25. Childs, A.M., Jao, D., Soukharev, V.: Constructing elliptic curve isogenies in quantum subexponential time. J. Math. Cryptol. **8**(3), 1–29 (2014)
26. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. J. Cryptol. **22**(1), 93–113 (2009)
27. Galbraith, S.D., Petit, C., Shani, B., Ti, Y.B.: On the security of supersingular isogeny cryptosystems. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 63–91. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_3
28. Kirkwood, D., Lackey, B.C., McVey, J., Motley, M., Solinas, J.A., Tuller, D.: Failure is not an Option: Standardization Issues for Post-Quantum Key Agreement. Technical report, Workshop on Cybersecurity in a Post-Quantum World (2015)