Fault attack on Supersingular Isogeny Cryptosystems

Yan Bo Ti[1]

Mathematics Department, University of Auckland, NZ. yanbo.ti@gmail.com

Abstract. We present the first fault attack on cryptosystems based on supersingular isogenies. During the computation of the auxiliary points, the attack aims to change the base point to a random point on the curve via a fault injection. We will show that this would reveal the secret isogeny with one successful perturbation with high probability. We will exhibit the attack by placing it against signature schemes and key-exchange protocols with validations in place. Our paper therefore demonstrates the need to incorporate checks in implementations of the cryptosystem.

1 Introduction

Cryptosystems based on isogenies between supersingular elliptic curves were proposed by Jao and De Feo in 2011 [13] as a candidate for cryptographic protocols in the post-quantum world. Instead of relying on the discrete logarithm problem which is susceptible to Shor's algorithm [20], it is based on the number-theoretic problem of finding isogenies between supersingular elliptic curves.

Cryptosystems based on isogenies have their genesis in an unpublished manuscript by Couveignes [8] and were later rediscovered by Rostovtsev and Stolbunov [18]. A paper by Charles, Goren and Lauter [3] then used the isogeny graphs to construct a hash function. However, Childs, Jao and Soukharev [4] managed to find a quantum algorithm that was able to break the cryptosystems in [8,18] in sub-exponential time by reducing the problem of finding an isogeny between isogenous ordinary curves to a hidden shift problem which can be solved by a quantum algorithm (Kuperberg's algorithm [16]). The reduction is based on the abelian group action of the class group of the endomorphism ring of the elliptic curve. This action is absent in the supersingular case and hence their reduction does not apply.

Since the publication of [13], protocols such as the interactive identification protocol [9] and various signature schemes have been introduced [12,24,14,23,19] to add to the key-exchange and encryption protocols introduced in [13]. A cryptanalysis paper [11] has highlighted their vulnerability to adaptive attacks and the importance of countermeasures. Some implementation papers have introduced side-channel protection such as constant time operations [7]. However, threats posed by fault attacks have been absent in the literature.

Fault attacks exploit the leakage of sensitive information when the implementation operates under unexpected circumstances. Biehl, Meyer and Müller

[2] extended fault attacks on RSA cryptosystems to systems using elliptic curves. Ciet and Joye [5] then refined the methods and made the attack more practical. The key insight in both papers was the absence of the a_6 elliptic curve parameter in the scalar multiplication computation. The fault changed the base point P to some P'. This meant that the output point $[\lambda]P'$, where λ is the secret, might be in a group where solving the elliptic curve discrete logarithm problem was feasible, hence allowing for the recovery of some information about λ.

In this work, we will examine the effects of changing a point P to some random P' and attempt to recover the secret, which in this case is an isogeny ϕ. The attack would be able to recover the entire secret ϕ from a single output $\phi(P')$ with high probability. This compares well against the fault attack presented in [5] where a single successful perturbation only reveals partial information of the secret. We will present a fault attack in the context of several signature schemes and key-exchange protocols. The attack would work against the countermeasure proposed by Kirkwood et al. [15] which is based on the Fujisaki–Okamoto transform. The main observation that underlies the attack is that users should never reveal the image of random points under the secret isogeny.

The main result of the paper will be presented in Section 3. Prior to that, Section 2 will cover both the mathematical notions and the cryptographic protocols required to understand this paper. In Section 4 we will analyse the attack and discuss its feasibility.

2 Preliminaries

2.1 Mathematical background

Let E and E' be elliptic curves defined over a finite field $\mathbb{F}_q$ of characteristic p, then an *isogeny* between them is a non-zero morphism that maps the group identity of E to the group identity of E'. If $\phi : E \to E'$ is an isogeny, then it is a group homomorphism from $E(\overline{\mathbb{F}}_q)$ to $E'(\overline{\mathbb{F}}_q)$ [21, III.4.8] Equivalently, we are able to represent an isogeny ϕ as an algebraic morphism of the form

$$\phi(x, y) = \left(\frac{f_1(x, y)}{g_1(x, y)}, \frac{f_2(x, y)}{g_2(x, y)} \right)$$

where $\phi(\mathcal{O}) = \mathcal{O}$ and $f_i, g_i \in \mathbb{F}_q[x, y]$. In this case, we say that E and E' are *isogenous* over $\mathbb{F}_q$. The *degree* of an isogeny is defined to be its degree as an algebraic morphism and is denoted by $\deg \phi$. Isogenies with the same domain and range are known as *endomorphisms*. The map $[n] : E \to E$ given by

$$[n]P = \underbrace{P + \cdots + P}_{n \text{ times}}$$

is the multiplication-by-n map on E and is an example of an endomorphism. The kernel of this endomorphism is the set of n-torsion points which we denote by

$$E[n] = \left\{ P \in E\left(\overline{\mathbb{F}}_q\right) \;\middle|\; [n]P = \mathcal{O} \right\} .$$

If $p \nmid n$, then the set of n-torsion points of an elliptic curve has the group structure $E[n] \cong \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ [21, III.6.4].

Given an isogeny $\phi : E \to E'$, there exists a unique isogeny $\hat{\phi} : E' \to E$ such that

$$\phi \circ \hat{\phi} = [\deg \phi] = \hat{\phi} \circ \phi \,.$$

We call $\hat{\phi}$ the *dual isogeny* of ϕ [21, III.6.1]. Hence we can see that isogenous curves form an equivalence class.

An isogeny $\phi : E \to E'$ is *separable* if the induced extension of the function fields is separable. All of the isogenies that we will encounter in this paper will be separable. The size of the kernel of a separable isogeny is the same as the degree of the isogeny [21, III.4.10]. In fact, the link between a separable isogeny and its kernel goes deeper: the kernel of a separable isogeny uniquely defines the isogeny up to isomorphism [21, III.4.12]. To express this idea, we use the notation E/G to represent the codomain of some isogeny ϕ from E with kernel G. Given a finite subgroup G, an isogeny with kernel G can be computed using an algorithm by Vélu [22].

Given an elliptic curve E, the set of all endomorphisms over $\overline{\mathbb{F}}_q$, together with the zero isogeny, forms a ring. Addition in the ring is given by point-wise addition, and multiplication by composing endomorphisms. The endomorphism ring forms an algebra over $\mathbb{Z}$ and is of dimension at most 4 [21, III.4.2,III.7.5]. In fact $\dim_{\mathbb{Z}} \operatorname{End} E = 2$ or 4 and in the first case, we say that E is ordinary and in the second case, we say that E is *supersingular*. For the remainder of this paper, the elliptic curves we will encounter will be supersingular.

2.2 Supersingular isogeny cryptosystem

In this section, we will review the key-exchange protocol, interactive identification protocol and the various signature schemes. The key-exchange and the identification protocols were first introduced in [13,9]. Thereafter, signature schemes were introduced in [14,12,24], where the latter two are based on the identification protocol.

Key-exchange Suppose that Alice and Bob wish to establish a shared secret. There are three steps to the protocol that will achieve this objective.

Set-up: Fix a prime p of the form $p = \ell_A^{e_A} \cdot \ell_B^{e_B} \cdot f \pm 1$ where ℓ_A and ℓ_B are small distinct primes, f is a small cofactor, and e_A and e_B are positive integers such that $\ell_A^{e_A} \approx \ell_B^{e_B}$. Now fix a supersingular elliptic curve E over $\mathbb{F}_{p^2}$ and pick bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ for the $\ell_A^{e_A}$ and $\ell_B^{e_B}$-torsion subgroups.

Key generation: Alice picks random elements $a_1, a_2 \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A, and computes the subgroup $G_A = \langle [a_1]P_A + [a_2]Q_A \rangle$. She then uses the formula from Vélu to compute a curve $E_A = E/G_A$ and an isogeny $\phi_A : E \to E_A$, where $\ker \phi_A = G_A$. Alice also computes the points $\phi_A(P_B)$ and $\phi_A(Q_B)$. She then sends the tuple $(E_A, \phi_A(P_B), \phi_A(Q_B))$ to Bob. Bob performs the computation mutatis mutandis on his end.

Key derivation: Upon receipt of Bob's tuple $(E_B, \phi_B(P_A), \phi_B(Q_A))$, Alice computes the subgroup $G'_A = \langle [a_1]\phi_B(P_A) + [a_2]\phi_B(Q_A) \rangle$ and uses Vélu's formula to compute the elliptic curve $E_{AB} = E_B/G'_A$. She then uses the j-invariant of E_{AB} as the shared secret. Bob proceeding likewise would also obtain the j-invariant of E_{AB} to use as the shared secret. The protocol can be summarised in Fig. 1.

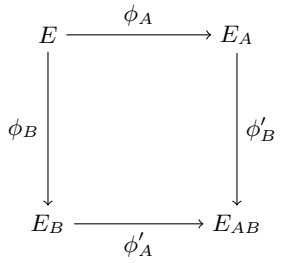


Fig. 1: Key-exchange protocol

Interactive identification protocol This interactive identification protocol has four steps: set-up, commitment, challenge and response.

Set-up: Fix a prime p of the form $p = \ell_A^{e_A} \cdot \ell_B^{e_B} \cdot f \pm 1$ where ℓ_A and ℓ_B are small distinct primes, f is a small cofactor and e_A and e_B are positive integers such that $\ell_A^{e_A} \approx \ell_B^{e_B}$. Now fix a supersingular elliptic curve E over $\mathbb{F}_{p^2}$.
The prover picks a random element $S \in E[\ell_A^{e_A}]$ with order $\ell_A^{e_A}$ and computes $\phi : E \to E/\langle S \rangle = E_S$. Then, the prover generates a basis $\{P_B, Q_B\}$ for $E[\ell_B^{e_B}]$. The prover then computes and publishes the tuple

$$(E, P_B, Q_B, E_S, \phi(P_B), \phi(Q_B))$$

as the public key.

The two parties then repeat the next three steps until a security threshold is reached.

Commitment: The prover chooses random elements $r_1, r_2 \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, not both divisible by ℓ_B and computes the point $R = [r_1]P_B + [r_2]Q_B$. The prover then computes the isogeny $\psi : E \to E/\langle R \rangle = E_R$ and the curve $E_{RS} = E_S/\langle \phi(R) \rangle = E_S/\langle [r_1]\phi(P_B) + [r_2]\phi(Q_B) \rangle = E/\langle R, S \rangle$. The prover sends (E_R, E_{RS}) to the verifier.
Challenge: The verifier sends the challenge bit $c \in \{0, 1\}$.

Response: In response, the prover reveals $(R, \phi(R))$[1] if $c = 0$ or $\psi(S)$ if $c = 1$. In the former case, the verifier would check that $E/\langle R\rangle \cong E_R$ and $E_S/\langle\phi(R)\rangle \cong E_{RS}$. In the latter case, the verifier checks that $E_R/\langle\psi(S)\rangle \cong E_{RS}$.

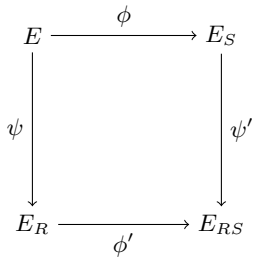


Fig. 2: Interactive identification protocol

Digital signature scheme This non-interactive signature scheme is the result of applying the Fiat–Shamir transform on the interactive identification protocol presented above. This scheme was introduced in [12] and [24]. The signature scheme uses the output of the hash as a string of challenge bits to generate a string of responses corresponding to the challenges. The verification step then involves verifying the response in the signature for each challenge bit.
Details of the scheme are given in §A.1.

Undeniable signature scheme The undeniable signature scheme [14] is a "three-dimensional" analogue to key-exchange protocol which is "two-dimensional" in the sense that we consider a commutative cube instead of a commutative square. Given a signature, the scheme is able to confirm the signature if the signature is valid, or disavow an invalid signature without having to reveal a valid signature.
Details of the scheme are given in §A.2.

2.3 The Kirkwood et al. Validation Method

Kirkwood et al. introduced a method to secure the key-exchange protocol of isogeny cryptosystems. This is based on the Fujisaki–Okamoto transform [10] which is also explained by Peikert [17, §5.2] and Galbraith et al. [11, §2.3]. The method allows for one party to validate the other, but for the ease of exposition, let us suppose that Alice is using a static secret and Bob needs to prove to her that he is performing the protocol correctly.

[1] It is also possible to compress $(R, \phi(R))$ by sending (r_1, r_2) instead (c.f. [1]). The verifier can then recover R and $\phi(R)$ given P_B, Q_B, $\phi(P_B)$ and $\phi(Q_B)$.

Bob would prove to Alice that he performed the protocol correctly by executing the key-exchange, encrypting the random seed used to generate his private key and sending this ciphertext to Alice for her to verify that the random seed leads to the correct keys.

Applied to the Jao–De Feo protocol, we will briefly explain how Bob can prove to Alice that he has executed the protocol correctly. This is especially applicable if Alice is using a static key and Bob is potentially a malicious party.

1. Alice computes and sends the public key $(E_A, \phi_A(P_B), \phi_A(Q_B))$.
2. Bob receives Alice's public key $(E_A, \phi_A(P_B), \phi_A(Q_B))$.
3. Bob obtains his random seed r_B from a random source and derives his private key using a key derivation function, KDF_1,

$$(b_1, b_2) = \mathrm{KDF}_1(r_B)\,.$$

   He uses the secret key to compute $G_B = \langle [b_1]P_B + [b_2]Q_B \rangle$, and uses the Vélu formula to compute ϕ_B and $E_B = E/G_B$.
4. Bob derives the shared secret $j(E_{AB})$ using his private key and Alice's public key. He then computes a session key (SK) and a validation key (VK) using a key derivation function, KDF_2,

$$SK \mid VK = \mathrm{KDF}_2(j(E_{AB}))\,.$$

5. Bob sends his public key $(E_B, \phi_B(P_A), \phi_B(Q_A))$ and $c_B = \mathrm{Enc}_{VK}(r_B \oplus SK)$ to Alice.
6. Using her private key and Bob's public key, Alice computes the shared secret $j(E'_{AB})$ and derives the session and validation keys SK' and VK'. She uses these to compute

$$r'_B = \mathrm{Dec}_{VK'}(c_B) \oplus SK'\,.$$

   She then computes Bob's secret keys from r'_B and recomputes all of Bob's operations and compares $(E'_B, \phi'_B(P_A), \phi'_B(Q_A))$ with $(E_B, \phi_B(P_A), \phi_B(Q_A))$. If they are equal, then Alice verifies that Bob has computed the protocol correctly and proceeds to use $SK' = SK$ for future communication with Bob. Else, the protocol terminates in a non-accepting state.

This validation method can be used for both the key-exchange and the encryption protocols. It also compels one party to reveal the secret used and so requires a change in secret keys after each verification. This protocol is summarised in Fig. 3.

3 Fault attack

Assume that the protocol under attack reveals the x-coordinate of the image of a point under the secret isogeny. The fault attack aims to force the implementation to output the image of a random point under the secret isogeny. This would allow

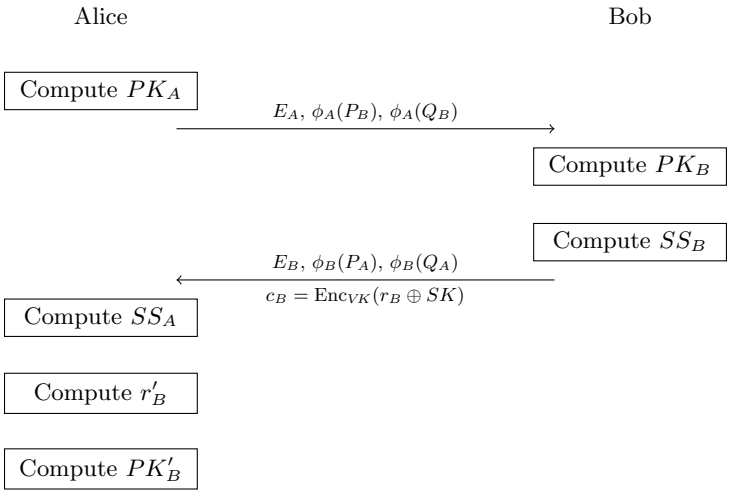


Fig. 3: The Kirkwood et al. validation method for supersingular key-exchange.

the adversary to recover the secret. We will see how this is accomplished and see the different scenarios where the fault attack may be employed.

Our first observation is that computations do not involve the y-coordinate of the points. Given a curve E and a point P, a perturbation in the x-coordinate of P would result in another point P' on the same curve over a quadratic extension. Indeed, given any x, we recover the y-coordinate of P' by solving a quadratic equation which always has a solution in $\mathbb{F}_{p^2}$. In particular, any $x \in \mathbb{F}_{p^2}$ either corresponds to a point on E or a point on its quadratic twist E'. In [7], the most efficient implementation of the cryptosystem thus far, computations do not distinguish between the curve E and its quadratic twist E', hence the isogeny will be evaluated correctly on any $x \in \mathbb{F}_{p^2}$. In a more general setting where the twists of the curves are treated separately, the faulted point will be on E with probability $1/2$ and on the twist with probability $1/2$. Hence the adversary may assume, after a series of faults, that a perturbed point will lie on E.

The perturbed point would be a random point on the curve. In §3.1, we will show how one recovers the secret isogeny given the image of the random point. This is not dissimilar to [14, Remark 3.1], where Jao and Soukharev noted that a party should never disclose any information that allows an adversary to evaluate ϕ_A on $E[\ell_A^{e_A}]$. The method to recover ϕ_A given the image of a random point in $E[\ell_A^{e_A}]$ is mentioned in [9, §5.1] and explained in detail in §3.1. In fact, we will show that a party should never reveal the image of random points under the secret isogeny.

3.1 Recovery of isogeny from image of random point

Let $E/\mathbb{F}_{p^2}$ be a supersingular elliptic curve where $p = \ell_A^{e_A} \cdot \ell_B^{e_B} \cdot f \pm 1$. Then with (P_A, Q_A), (P_B, Q_B), and (P_C, Q_C) being the generators of $E[\ell_A^{e_A}]$, $E[\ell_B^{e_B}]$, and

$E[f]$ respectively, a random point $X \in E(\mathbb{F}_{p^2})$ takes the form

$$X = [u]P_A + [v]Q_A + [w]P_B + [x]Q_B + [y]P_C + [z]Q_C$$

for some $u, v, w, x, y, z \in \mathbb{Z}$.

Now suppose that we are given the image of X under the secret isogeny ϕ_A, then we will show how one can use the knowledge of $\phi_A(X)$ to recover ϕ_A. Since ϕ_A is a group homomorphism and we know that X can be expressed as a linear combination of P_A, Q_A, P_B, Q_B, P_C, and Q_C, we have

$$\begin{aligned}\phi_A(X) &= \phi_A([u]P_A + [v]Q_A + [w]P_B + [x]Q_B + [y]P_C + [z]Q_C) \\ &= [u]\phi_A(P_A) + [v]\phi_A(Q_A) + [w]\phi_A(P_B) \\ &\quad + [x]\phi_A(Q_B) + [y]\phi_A(P_C) + [z]\phi_A(Q_C)\,.\end{aligned}$$

Now our aim is to isolate a linear combination of $\phi_A(P_A)$ and $\phi_A(Q_A)$. To that end, we perform the operation

$$[\ell_B^{e_B} \cdot f]\phi_A(X) = [\ell_B^{e_B} \cdot f]([u]\phi_A(P_A) + [v]\phi_A(Q_A)) \quad = [u']\phi_A(P_A) + [v']\phi_A(Q_A)\,,$$

and we find ourselves in the scenario described in [14, Remark 3.1] and [9, §5.1].

Once we have $[u']\phi_A(P_A) + [v']\phi_A(Q_A)$, the subgroup generated by this point will help with the construction of the dual isogeny of ϕ_A hence recovering ϕ_A.

Lemma 1. *Let E_1 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, where $p = \ell_A^{e_A}\ell_B^{e_B} f \pm 1$. Suppose $\phi : E_1 \to E_2$ is an isogeny of degree $\ell_A^{e_A}$ with a cyclic kernel and let $\{P, Q\}$ be generators of $E_1[\ell_A^{e_A}]$. Then for any $X \in E_1[\ell_A^{e_A}]$, define $\psi : E_2 \to E'$ such that $\ker\psi = \langle\phi(X)\rangle$, then there exists some $\theta : E' \to E_1$ of degree ℓ_A^{ϵ}, $\epsilon \leq e_A$, such that*

$$\hat{\phi} = \theta \circ \psi\,.$$

Proof. Using [11, Lemma 1], we may suppose that $\ker\phi = \langle P + [\alpha]Q\rangle$. Hence

$$\begin{aligned}\phi(P) &= \phi(P) - \phi(P + [\alpha]Q) \\ &= -[\alpha]\phi(Q)\,.\end{aligned}$$

Then expressing $X = [u]P + [v]Q$ for some u, v, we have

$$\langle\phi(X)\rangle = \langle[u]\phi(P) + [v]\phi(Q)\rangle = \langle[v - \alpha u]\phi(Q)\rangle = \langle[\ell_A^k]\phi(Q)\rangle\,,$$

where k is the ℓ_A-adic valuation of $(v - \alpha u)$.

Let $\psi : E_2 \to E'$ be an isogeny with kernel given by $\langle\phi(X)\rangle = \langle[\ell_A^k]\phi(Q)\rangle$. Pick any $Y \in E_1[\ell_A^{e_A}]$ and write $Y = [r]P + [s]Q$ for some r, s.

If $k = 0$, then

$$\begin{aligned}\psi \circ \phi(Y) &= \psi(\phi([r]P + [s]Q)) \\ &= \psi([s - r\alpha]\phi(Q)) \\ &= \mathcal{O}\,.\end{aligned}$$

So it is clear that $E_1[\ell_A^{e_A}] \subseteq \ker(\psi \circ \phi)$. The reverse inclusion is obvious since $\ker(\psi \circ \phi)$ does not contain any non-trivial element of order co-prime to ℓ_A. So $\psi \circ \phi = [\ell_A^{e_A}]$, which implies, by the uniqueness of the dual isogeny, that $\psi = \hat{\phi}$, and $\theta : E_1 \to E_1$ is the identity isogeny.

If $k > 0$,

$$\begin{aligned} \psi \circ \phi(Y) &= \psi(\phi([r]P + [s]Q)) \\ &= \psi([s - r\alpha]\phi(Q)) \,. \end{aligned}$$

Note that $\psi \circ \phi(Y)$ has order at most ℓ_A^k, since

$$[\ell_A^k]\psi \circ \phi(Y) = [s - r\alpha]\psi([\ell_A^k]\phi(Q)) = \mathcal{O} \,.$$

Now denote by $\gamma \in \mathbb{Z}_{\geq 0}$, the ℓ_A-adic valuation of $s - r\alpha$, then

$$\begin{aligned} \operatorname{ord}(\psi \circ \phi(Y)) &= \operatorname{ord}(\psi([s - r\alpha]\phi(Q))) \\ &= \ell_A^{k-\gamma} \,. \end{aligned}$$

[Note that $\epsilon = k - \gamma$.]

So choose Y such that $\gamma = 0$ and define $\theta : E' \to E_1$ such that $\ker \theta = \langle \psi \circ \phi(Y) \rangle$. Then using the above argument, we can see that $\theta \circ \psi = \hat{\phi}$. Furthermore, it is clear that $\deg \theta \leq \ell_A^{e_A}$. □

The lemma tells us that given the image of a point in $E_1[\ell_A^{e_A}]$ under an $\ell_A^{e_A}$-isogeny, ϕ, we are able to find an isogeny ψ which is close to the dual isogeny of ϕ. To obtain the dual isogeny, one has to first recover θ. If ϵ is sufficiently small, one will be able to recover θ by brute force. In fact, we will examine the size of ϵ in §4 and show that ϵ is small in most cases.

Hence we have the following algorithm to recover isogenies given the image of random points.

Algorithm 1: Recovering the dual isogeny after fault injection.

Data: $\phi(X)$
Result: $\hat{\phi}$
Set $\lambda \leftarrow \ell_B^{e_B} \cdot f$;
Set $T \leftarrow [\lambda]\phi(X)$;
Set $\psi : E_2 \to E'$ as the isogeny with kernel T;
if $\operatorname{ord}(T) = \ell_A^{e_A}$ **then**
  Return ψ;
else
  Brute force for θ;
Return $\theta \circ \psi$;

3.2 Fault Models

We will now demonstrate the fault attack in the following scenarios:

- Interactive identification protocol
- Digital signature scheme
- Undeniable signature protocol
- Static key-exchange protocol
- Static key-exchange protocol with the Kirkwood et al. validation method

The feasibility of each of these will be discussed in §4.1

Interactive identification protocol and signature schemes In the *interactive identification protocol*, to learn the prover's long-term secret S, the adversary needs to perturb the computation of the point $\phi(R)$. During the prover's computation, the adversary will introduce a perturbation immediately before the computation of $\phi(R)$. In particular the adversary would attempt to inject a fault into the fetching operation and cause a fault in R. This will cause the faulted point R' to be, with high probability, a point of full order. Successfully doing so would allow for the recovery of the secret isogeny ϕ. To obtain the output of the faulted point, the adversary needs the challenge bit to be 0 as described in §2.2. This would happen 50% of the time and since identification schemes typically require a large number of passes, this must happen with high probability. The adversary could check the order of the points in the responses (if the challenge bit is 0) and the faulted point would have order larger than $\ell_A^{e_A}$. Using this information, the adversary would be able to use Algorithm 1 to recover S.

Due to its similarity to the identification protocol, to learn the signer's long-term secret S in the *digital signature scheme*, the steps the adversary takes are identical to the process above. The aim now is to inject a fault during the computation of $\phi(R_i)$ (c.f. §A.1) for some i's. A successful fault coinciding with the challenge bit being 0 would produce a point of order larger than $\ell_A^{e_A}$, so the adversary has to find that point in the signature by testing the orders of the points in the signature.

In the *undeniable signature protocol* the adversary will be able to learn the long term secret ϕ_A by inducing a fault in $\phi_M(P_C)$ before the computation of $\phi_{M,AM}(\phi_M(P_C))$ (c.f. §A.2). Using $\phi_{M,AM}(X)$, the adversary would learn $\phi_{M,AM}$ and equivalently, $\phi_M(G_A)$. Since ϕ_M is computable from the message, the adversary would be able to recover G_A.

Static key-exchange protocol Consider the static key-exchange protocol described in §2.2. Suppose an adversary is trying to learn Alice's static secret isogeny and has the ability to cause a fault in Alice's computation. After introducing a fault in the computation of $\phi_A(P_B)$, Alice would then proceed to publish the public key tuple

$$(E_A, \phi_A(X), \phi_A(Q_B)) .$$

The adversary will then be able to recover ϕ_A using Algorithm 1.

Notice that this would not be prevented by the validation method presented in §2.3. Since the validation method will only be able to detect misdemeanours carried out by Bob, it will not be able to prevent the fault attack. In particular, throughout the validation process the public key of Alice is only computed once and is never checked by the method. Hence the fault attack would not be detected by this validation and an adversary would be able to recover the secret isogeny as previously described.

Remark 1. The attack may also be implemented on the ephemeral key-exchange protocol, but in both settings the attack would cause a failure to establish a shared secret key.

3.3 Countermeasures

A simple countermeasure to this attack is to implement order checking before the publication of the auxiliary points. Another countermeasure that can be placed on the identification protocol and hence the signature scheme is the compression of the points $R, \phi(R)$ if the challenge bit is 0. Sending r_1 and r_2 allows the verifier to recompute R and $\phi(R)$ using the public keys and will prevent the adversary from learning the faulted auxiliary point. Note that the compression of $\psi(S)$ will not be useful since the attack does not attack that point.

4 Analysis of attack

As seen in the proof of Lemma 1, to obtain the dual of the isogeny, we need $k = 0$, or failing that, have ϵ small. But since ϵ is dependent on k, we will study k instead.

We start by fixing some $\alpha \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and suppose that u and v are selected randomly in $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, then we have

$$\Pr\left(\ell_A^n \text{ divides } (u - \alpha v)\right) = \frac{1}{\ell_A^n}\,.$$

Indeed, it is clear that we can treat $\rho = u - \alpha v$ as a single random variable, so this reduces to finding $\Pr(\ell_A^n \text{ divides } \rho)$, where ρ is randomly selected from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$. Since one in every ℓ_A^n elements is divisible by ℓ_A^n, we have the claim.

So $k = 0$ with probability $1 - \frac{1}{\ell_A}$. More generally, $k = \kappa$ with probability $\frac{\ell_A - 1}{\ell_A^{\kappa+1}}$. So we see that the isogeny ψ obtained from the procedure in §3 will be close to being the dual isogeny and brute forcing for θ is feasible.

Lastly, we will address the issue of the faulted point $\phi(X)$ not having an order divisible by $\ell_A^{e_A}$. This would have the effect of decreasing the degree of ψ and so increase the degree of θ. But notice that we can repeat the same analysis as the above to conclude that the degree of θ would be small with high probability.

Hence we have shown that Algorithm 1 has a high probability of recovering the secret isogeny.

4.1 Feasibility of attack models

Let us now study the feasibility of the attacks discussed in §3.2. We will see that the attacks would work well against signature schemes but not against key-exchange protocols.

Signature schemes The presence of a long-term secret and the availability of auxiliary points makes the signature schemes extremely attractive for an adversary attempting a fault attack on the supersingular isogeny cryptosystem. Note that while a fault would affect the validity of the signatures, the signer will not change the long-term secret due to an invalid signature. Hence the adversary would be able to break the signature scheme. We have to add that the compression of points is an effective countermeasure that foils the attack and would also reduce the size of the responses.

Key-exchange protocols Suppose that one party is using a static key in the key-exchange protocol. An adversary would be able to recover the secret isogeny if the static public key is recomputed for each exchange. However, this is unlikely to happen since $\phi_A(P_B)$ and $\phi_A(Q_B)$ will be hardcoded for efficiency.

Now suppose that the adversary is attacking the key-exchange protocol with ephemeral keys. If the secrets are not authenticated, the adversary would be able to compute $\phi_A(P_B)$, and send that in place of $\phi_A(X)$. This way, both parties would be able to derive the same shared secret. Since recovering ϕ_A from $\phi_A(X)$ can be done efficiently, and computing $\phi_A(P_B)$ is also efficient, performing the substitution before a time-out in the connection is very feasible. However, it should be noted that without authentication, it might be better to use a man-in-the-middle attack.

Acknowledgements

I am grateful to Steven Galbraith and Craig Costello for insightful conversations and comments. I would also like to thank the referees for their helpful feedback and suggestions.

References

1. Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi. Key compression for isogeny-based cryptosystems. In *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography*, pages 1–10, 2016.
2. Ingrid Biehl, Bernd Meyer, and Volker Müller. Differential fault attacks on elliptic curve cryptosystems. In *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '00, pages 131–146. Springer-Verlag, 2000.

3. Denis X. Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic Hash Functions from Expander Graphs. *J. Cryptology*, 22(1):93–113, 2009.
4. Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Mathematical Cryptology*, 8(1):1–29, 2014.
5. Mathieu Ciet and Marc Joye. Elliptic curve cryptosystems in the presence of permanent and transient faults. *Designs, Codes and Cryptography*, 36(1):33–43, 2005.
6. Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of SIDH public keys. In *Advances in Cryptology - EUROCRYPT '17 Proceedings, Part I*, pages 679–706, 2017.
7. Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie–Hellman. In *Advances in Cryptology – CRYPTO '16 Proceedings, Part I*, pages 572–601, 2016.
8. Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006:291, 2006.
9. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology*, 8(3):209–247, 2014.
10. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99 Proceedings*, pages 537–554. Springer, 1999.
11. Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *Advances in Cryptology - ASIACRYPT '16 Proceedings, Part I*, pages 63–91, 2016.
12. Steven D. Galbraith, Christophe Petit, and Javier Silva. Signature schemes based on supersingular isogeny problems. Cryptology ePrint Archive, Report 2016/1154, 2016. `http://eprint.iacr.org/2016/1154`.
13. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto '11 Proceedings*, pages 19–34, 2011.
14. David Jao and Vladimir Soukharev. Isogeny-based quantum-resistant undeniable signatures. In *PQCrypto '14 Proceedings*, pages 160–179. Springer, 2014.
15. Daniel Kirkwood, Bradley C. Lackey, John McVey, Mark Motley, Jerome A. Solinas, and David Tuller. Failure is not an option: Standardization issues for post-quantum key agreement, 2015. Workshop on Cybersecurity in a Post-Quantum World.
16. Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005.
17. Chris Peikert. Lattice cryptography for the internet. In *PQCrypto '14 Proceedings*, volume 8772, pages 197–219. Springer, 2014.
18. Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. `http://eprint.iacr.org/`.
19. Srinath M. S. and V. Chandrasekaran. Isogeny-based quantum-resistant undeniable blind signature scheme. Cryptology ePrint Archive, Report 2016/148, 2016. `http://eprint.iacr.org/2016/148`.
20. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
21. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2009.

22. Jacques Vélu. Isogénies entre courbes elliptiques. *C.R. Acad. Sc. Paris, Série A.*, 273:238 – 241, 1971.
23. Sun Xi, Haibo Tian, and Yumin Wang. Toward quantum-resistant strong designated verifier signature from isogenies. *International Journal of Grid and Utility Computing*, 5(2):292–296, September 2012.
24. Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In *To appear in Financial Crypto 2017*, 2017.

A Signature schemes in detail

A.1 Digital signature scheme

The signature scheme has three steps: key generation, signing and verifying.

Set-up and key generation: Fix a prime p of the form $p = \ell_A^{e_A} \cdot \ell_B^{e_B} \cdot f \pm 1$ where ℓ_A and ℓ_B are small distinct primes, f is a small cofactor and e_A and e_B are positive integers such that $\ell_A^{e_A} \approx \ell_B^{e_B}$. Now fix an elliptic curve E over $\mathbb{F}_{p^2}$. Next, let $t = 0.5\lfloor \log_2 p \rfloor$ and fix a hash function $H : \{0,1\}^* \to \{0,1\}^t$. The signer picks a random element $S \in E[\ell_A^{e_A}]$ with order $\ell_A^{e_A}$ and computes $\phi : E \to E/\langle S \rangle = E_S$. The signer then generates a basis $\{P_B, Q_B\}$ for $E[\ell_B^{e_B}]$, and computes and publishes the tuple

$$(E, P_B, Q_B, E_S, \phi(P_B), \phi(Q_B))$$

as the public key.
Signing: The signer needs to produce t challenges. So for each $i = 1, \ldots, t$, choose random elements $r_{1,i}, r_{2,i} \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ such that not both are divisible by ℓ_B and computes the points

$$\begin{aligned} R_i &= [r_{1,i}]P_B + [r_{2,i}]Q_B \,, \\ T_i &= [r_{1,i}]\phi(P_B) + [r_{2,i}]\phi(Q_B) \end{aligned}$$

and the isogenies

$$\begin{aligned} \psi_i &: E \to E/\langle R_i \rangle = E_{R_i} \,, \\ \phi_i' &: E_S \to E_S/\langle T_i \rangle = E_{T_i} \,. \end{aligned}$$

Given a message m, the signer computes

$$h = H\left(m, E_{R_1}, \ldots, E_{R_t}, E_{T_1}, \ldots, E_{T_t}\right) .$$

The bit-string of h would serve as the sequence of challenge bits.
If the i-th bit of h is 0, the signer sets $z_i = (R_i, \phi(R_i))$[2]. If the i-th bit of h is 1, the signer sets $z_i = \psi_i(S)$. The signature would then be the tuple

$$(h, z_1, z_2, \ldots, z_t) \,.$$

Verifying: To verify the signature, the verifier would use the output of the hash as the challenge bits and use the same verification procedure as seen in §2.2 to verify each z_i as the response to the challenge bits.

[2] It is also possible to compress z_i by sending $r_{1,i}$ and $r_{2,i}$ instead. The verifier can then recover R and $\phi(R)$ given P_B, Q_B, $\phi(P_B)$ and $\phi(Q_B)$.

A.2 Undeniable signature scheme

This signature scheme has three steps: key generation, signing and verifying. The last step is split into confirmation or disavowal.

Set-up and key generation: Fix a hash function $H : \{0,1\}^* \to \mathbb{Z}$. Fix a prime p of the form $p = \ell_A^{e_A} \cdot \ell_M^{e_M} \cdot \ell_C^{e_C} \cdot f \pm 1$ and fix a supersingular elliptic curve E over $\mathbb{F}_{p^2}$. Now pick bases $\{P_A, Q_A\}$, $\{P_M, Q_M\}$ and $\{P_C, Q_C\}$ for the $\ell_A^{e_A}$, $\ell_M^{e_M}$ and $\ell_C^{e_C}$-torsion points respectively. The signer then randomly picks elements $a_1, a_2 \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ not both divisible by ℓ_A, computes the subgroup $G_A = \langle [a_1]P_A + [a_2]Q_A \rangle$ and uses Vélu's formula to compute $E_A = E/G_A$ and the isogeny $\phi_A : E \to E_A$. The signer computes the image of P_C and Q_C under this isogeny and publishes the tuple $(E_A, \phi_A(P_C), \phi_A(Q_C))$.
Signing: Given a message M, the signer computes the hash $h = H(M)$ and the subgroup $G_M = P_M + [h]Q_M$. Next, the signer computes the following isogenies:

- $\phi_M : E \to E_M = E/G_M$
- $\phi_{M,AM} : E_M \to E_{AM} = E/\phi_M(G_A)$
- $\phi_{A,AM} : E_A \to E_{AM} = E/\phi_A(G_M)$

The signature then consists of the tuple

$$(E_{AM}, \phi_{M,AM}(\phi_M(P_C)), \phi_{M,AM}(\phi_M(Q_C))) \ .$$

Verification: Since this is an undeniable signature scheme, there are two components to this: the *confirmation protocol* and the *disavowal protocol*.
In the former protocol, given the signature

$$(E_{AM}, \phi_{M,AM}(\phi_M(P_C)), \phi_{M,AM}(\phi_M(Q_C))) \ ,$$

the objective is to confirm E_{AM}. In the latter, given the signature (E_F, F_P, F_Q), the objective is to disavow the signature.

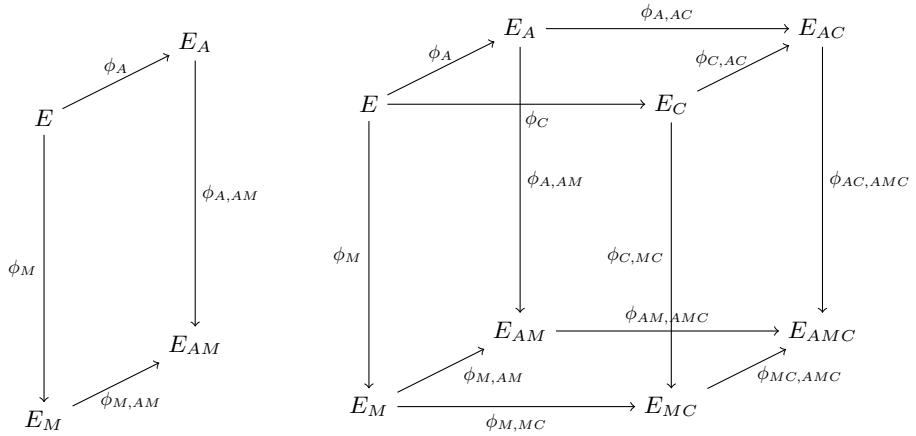
Confirmation:

1. The signer picks random elements $c_1, c_2 \in \mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$ not both divisible by ℓ_C, computes the subgroup $G_C = \langle [c_1]P_C + [c_2]Q_C \rangle$ and computes

$$E_C = E/G_C \,, \qquad E_{MC} = E_M/\phi_M(G_C) \,,$$
$$E_{AC} = E_A/\phi_A(G_C) \,, \qquad E_{AMC} = E_{MC}/\phi_{C,MC}(G_A) \,.$$

2. The signer publishes $(E_C, E_{AC}, E_{MC}, E_{AMC}, \phi_C(P_M) + [h]\phi_C(Q_M))$.
3. The verifier randomly selects $b \in \{0,1\}$.
   If $b = 0$: the signer outputs $\ker \phi_C$. The verifier then computes ϕ_C, $\phi_{M,MC}$, $\phi_{A,AC}$ and $\phi_F : E_F \to E_{FC}$ and checks that each isogeny maps between the curves in the commitment. The verifier also computes $\phi_{C,MC}$ and checks that it matches the commitment.
   If $b = 1$: the signer outputs $\ker \phi_{C,AC}$ and the verifier then computes $\phi_{MC,AMC}$, $\phi_{AC,AMC}$ and checks that E_{AMC} is the codomain.

Disavowal: The disavowal step is almost exactly the same as the confirmation step with the exception in the last step where if $b = 0$, the verifier would see that $E_{FC} \ncong E_{AMC}$.



(a) Signing (b) Confirmation/Disavowal

Fig. 4: Commutative diagrams generated during protocol