SILBE: an Updatable Public Key Encryption Scheme from Lollipop Attacks

Max Duparc, Tako Boris Fouotsa, and Serge Vaudenay

EPFL, Lausanne, Switzerland
{max.duparc,tako.fouotsa,serge.vaudenay}@epfl.ch

Abstract. We present a new post-quantum Public Key Encryption scheme (PKE) named Supersingular Isogeny Lollipop Based Encryption or SILBE. SILBE is obtained by leveraging the generalized lollipop attack of Castryck and Vercauteren on the M-SIDH Key exchange by Fouotsa, Moriya and Petit. Doing so, we can in fact make of SILBE a post-quantum secure Updatable Public Key Encryption scheme (UPKE). SILBE is the first isogeny-based UPKE which is not based on group actions. In its core, SILBE extensively uses both the Deuring Correspondence and Kani's Lemma, two central concepts in Isogeny-Based Cryptography.

Keywords: Post-Quantum Cryptography · Supersingular Isogenies · M-SIDH · Generalized Lollipop Attacks · UPKE

1 Introduction

The notion of Updatable Public Key Encryption (UPKE) was initially introduced in [7] as a relaxation of Forward Secure Public Key Encryption (FSPKE), given the inherent complexity of constructing FSPKE systems and the shared advantageous properties between the two. In addition to functioning as PKE, UPKE allows for secure asynchronous key updates. Several schemes have been proposed based on discrete logarithm [18], DRC [1], LWE [18,2], and on isogenies [19,28].

In this later case, an in-depth exploration of the question was performed in 2020 by Eaton, Jao, Komlo, and Mokrani in [19]. They proposed two designs of isogeny-based UPKE, respectively based on SIDH [22,14] and CSIDH [9]. For the former protocol, the authors suggested that "a viable construction in practice is hindered by existing mathematical limitations" and described a relaxed variant of UPKE which they named "online UPKE". The online UPKE was then instantiated in the SIDH setting. Follow-up developments on isogeny-based UPKE have been, to the best of our knowledge, focused on CSIDH and more generally on group actions [28].

In the meantime, the SIDH was shown insecure in [8,29,34] by leveraging the accessible images of torsion points to construct a high-dimensional isogeny using Kani's Lemma [24], one then extracts the secret isogeny from the high-dimensional one. This tool is revolutionary and has enables a breath of new schemes such as SQISignHD [12], FESTA and QFESTA [5,32], IS-CUBE [31] or

Leroux's VRF [27] and spawn many countermeasures such as M-SIDH [20,21] and binSIDH/terSIDH [4]. Proposed by Fouotsa, Moriya and Petit, the M-SIDH countermeasure prevents the attack of [8,29,34], at the cost of a lesser efficiency.

The idea we wanted to explore was therefore whether Kani's Lemma could be leveraged to construct an UPKE not based on isogeny group action. To answer this question, we strongly rely on two central tools:

- The first is the *generalized lollipop attack* of [10] as proposed by Castryck and Vercauteren, and especially how it can be used to attack some specific instances of the M-SIDH.
- The second is the *Deuring correspondence*, that links isogeny between supersingular elliptic curves and ideals between maximal orders of quaternion algebras, and more specifically its algorithms applications detailed in Leroux's thesis [26].

Contributions: Our main contribution is to turn the generalized lollipop attack [10] over M-SIDH into not only a PKE but an UPKE, therefore overcoming the mathematical limitations described in [19] when attempting to design SIDH based UPKE. Our UPKE, named SILBE[1] for Supersingular Isogeny Lollipop Based Encryption, follows the same principle as SETA [35] where the Petit torsion point attacks were used to design a PKE. However, this adaptation is not without its challenges and necessitates numerous adaptations and optimizations. The crux of the challenge lies in devising a key update mechanism that safeguards against information leakage about the secret keys. This is achieved by leveraging the diverse capabilities offered by various isogeny representations, coupled with the pseudorandom nature of walks on the supersingular isogeny graph. We also generate examples of secure prime for SILBE at different security levels.

Technical overview: Let $\phi : E_0 \to E_1$ be a secret isogeny in M-SIDH. The images of torsion points of highly composite order N through the isogeny ϕ are revealed up to a secret scalar α. Concretely, if $E_0[N] = \langle P, Q \rangle$, then public key is $(E_1, [\alpha]\phi(P), [\alpha]\phi(Q))$, where α is a secret scalar. Due to the compatibility between isogenies and pairings, it is sufficient to choose α as a square root of unity modulo N. The SIDH attacks are avoided by choosing N in such a way that the number of square roots of unity modulo N is exponential, meaning that N is highly composite. We refer to [21] for further details.

In [10], Castryck and Vercauteren show that if the curve E_0 is defined over $\mathbb{F}_p$, then one can use a generalisation of the so called "lollipop attack" to recover the secret isogeny ϕ when given $(E_1, [\alpha]\phi(P), [\alpha]\phi(Q))$. One thing to note here is that among all supersingular curves in characteristic p, very few are defined over $\mathbb{F}_p$. In fact, a uniformly random supersingular elliptic curve in characteristic p is defined over $\mathbb{F}_p$ with probability $\approx p^{-1/2}$. Moreover, given a uniformly random supersingular elliptic curve E, finding an isogeny connecting E to a supersingular curve defined over $\mathbb{F}_p$ is known to be hard [16]. If the latter problem was solved,

[1] "syllable" in German.

it would lead to a sub-exponential quantum algorithm for computing isogenies between supersingular elliptic curves [16], endangering the security of several isogeny-based protocols on its way, non group action ones more precisely.

The above observation hints that one could use the generalised lollipop attack to design a public key encryption scheme by proceeding as follows: the public key is a uniformly random supersingular elliptic curve E_A, and the secret key is an isogeny connecting E to a supersingular curve defined E_0 defined over $\mathbb{F}_p$ and of known endomorphism ring, say $\phi_A : E_0 \to E_A$. To encrypt a message m, a square root of unity modulo N, one translates this message into an M-SIDH isogeny $\phi_B : E_A \to E_B$, and the ciphertext is $(E_B, [\mathsf{m}]\phi_B(P), [\mathsf{m}]\phi_B(Q))$ where $E[N] = \langle P, Q\rangle$. To decrypt a ciphertext $(E_B, [\mathsf{m}]\phi_B(P), [\mathsf{m}]\phi_B(Q))$, one runs the generalized lollipop attack on the isogeny $\phi_B \circ \phi_A : E_0 \to E_B$ using the masked torsion point information available in the ciphertext. In practice, for the key generation, one samples E by performing a very long walk from E_0 (E_0 can be set to $j(E) = 1728$), then one uses the endomorphism ring of E_0 to compute a shorter isogeny $\phi_A : E_0 \to E_A$ which is used in the decryption. The fact that N is highly composite implies computing (higher dimensional) isogenies of relatively large prime degrees (say few thousands), which makes the resulting scheme not practical (yet). Nevertheless, the most interesting fact about this design is that it can be turned into an UPKE.

In fact, since the public key is solely composed of a uniformly random supersingular elliptic curve E_A, updating the public key is straightforward: one simply samples a very long uniformly random walk $\rho : E_A \to E'_A$, and set E'_A to be the new public key. To update the secret key which consists of a relatively short isogeny $\phi_A : E_0 \to E_A$ with E_0 defined over $\mathbb{F}_p$ and of known endomorphism ring, one translates $\rho \circ \phi : E_0 \to E'$ into a relatively short isogeny $\phi'_A : E_0 \to E'_A$ (this requires the endomorphism ring of E_0), and ϕ'_A is the new secret key. This leads to the very first secure isogeny-based UPKE which is not based on isogeny group actions. Hence overcoming the limitations highlighted in [19].

Outline: The remainder of this paper is organised as follows. In section 2 we give a detailed recall of isogenies, UPKE, M-SIDH and of the standard algorithms that we utilize to define SILBE. In section 3, we detail how we construct the PKE part of SILBE. In section 4 we explain how we build the key update mechanism of SILBE and show that its security remains unchanged. Finally, in section 5, we discuss how we find good public parameters and discuss SILBE's efficiency.

2 Preliminaries

Throughout this paper, we denote as λ the security parameter. We also say that $f(x) \leq \mathsf{negl}(x)$ if $|f(x)| \leq x^{-c}$ for any positive integers c for x big enough. A $\mathsf{PPT}(x)$ is a probabilistic algorithm that is $\mathsf{poly}(x)$, meaning that that its running time is polynomial in x. Let p be a prime, $\mathbb{F}_p$ be the finite field of characteristic p and $\overline{\mathbb{F}_p}$ its algebraic closure. Let E and E' be elliptic curves over $\overline{\mathbb{F}_p}$.

2.1 Isogenies Background

Here is a concise recapitulation of isogeny. For a more comprehensive exploration, we recommend referring to De Feo's notes [13] and Silverman's book [36] for a general understanding of elliptic curves and isogenies. For insights into the Deuring Correspondence, Leroux's thesis [26] is an excellent resource, while [34] provides valuable details on Kani's Lemma.

Basic Facts: An isogeny $\phi : E \to E'$ is given as a surjective projective rational map between $E(\overline{\mathbb{F}_p})$ and $E'(\overline{\mathbb{F}_p})$ that preserves the group structure. The degree of this rational map defines the *degree* of the isogeny. This induces that the degree of a composition of isogeny is the product of the respective degree of each isogeny. We will consider isogeny up to isomorphism, meaning that two isogenies $\phi : E \to F$ and $\psi : E' \to F'$ are isomorphic if they are equal up to pre- and post-composition of isomorphisms. Note that E and E' are isomorphic induces that they share the same j-invariant, with both notions being equivalent when seen in $\overline{\mathbb{F}_p}$. For every isogeny $\phi : E \to E'$, there exists an unique *dual isogeny* $\widehat{\phi} : E' \to E$ such that $\phi \circ \widehat{\phi} = \widehat{\phi} \circ \phi = [\deg(\phi)]$, with $[n]$ the scalar multiplication map for any $n \in \mathbb{Z}$. Using duality, we can define the *n-torsion group*, noted $E[n] = \ker([n])$ with $E[n] \cong \mathbb{Z}_n^2$ for n coprime to p.

Additionally, an isogeny $\phi : E \to E'$ is *separable* if $\deg(\phi) = |\ker(\phi)|$. Following the fundamental theorem of isomorphism, we have that any separable isogeny is defined up to isomorphism by its kernel, meaning that $\phi : E \to E$ and $\psi : E \to E/\ker(\phi)$ are isomorphic, we also have that for any isogeny, $\ker(\phi) \subset E[\deg(\phi)]$.

The characterisation of isogeny by their kernel enables to define the notion of *pushforwards*. Let $\phi : E \to F$ and $\psi : E \to F'$ be two isogenies of coprime degree. The pushforward of ψ by ϕ is the isogeny $\phi_*\psi : F \to E'$ defined by $\ker(\phi_*\psi) = \phi\big(\ker(\psi)\big)$.

Deuring Correspondence: Among isogenies, endomorphisms have important additional properties. First, $\mathrm{End}(E)$, the set of all endomorphisms for an elliptic curve E is an integral ring of characteristic zero, under the operations of pointwise addition and composition. An elliptic curve is said to be *ordinary* if $\mathrm{End}(E)$ is isomorphic to an order of a imaginary quadratic field. Otherwise, they are *supersingular* and $\mathrm{End}(E)$ is isomorphic to a maximal order of $\mathbf{B}_{p,\infty}$ a quaternion algebra ramified exactly at p and ∞. An order $\mathcal{O}$ of $\mathbf{B}_{p,\infty}$ is a subring such that $\mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q} = \mathbf{B}_{p,\infty}$ with $\mathbf{B}_{p,\infty}$ of the form $\mathbb{Q} + \mathbb{Q}\mathbf{i} + \mathbb{Q}\mathbf{j} + \mathbb{Q}\mathbf{ij}$ with $\mathbf{j}^2 = -p$, $\mathbf{i}^2$ depending of p and such that $\mathbf{ij} = -\mathbf{ji}$. An important example is the curve E_0 with j-invariant 1728. If $p = 3 \mod 4$, then it is supersingular and its endomorphism ring correspond to the maximal order $\mathcal{O}_0 = \mathbb{Z} + \mathbf{i}\mathbb{Z} + \frac{\mathbf{i}+\mathbf{j}}{2}\mathbb{Z} + \frac{1+\mathbf{ij}}{2}\mathbb{Z}$ with $i : (x, y) \to (-x, \sqrt{-1}y)$ and $j = \pi$ the Frobenius endomorphism.

Supersingularity is an important property as is preserved by isogenies and as all supersingular curves are defined in $\mathbb{F}_{p^2}$ and connected together. Importantly, Deuring proved in [17] that there is an equivalence between supersingular curves and maximal orders of $\mathbf{B}_{p,\infty}$ such that an isogeny ϕ between two curves E_0 and E_1, with $\mathrm{End}(E_0) \cong \mathcal{O}_0$ and $\mathrm{End}(E_1) \cong \mathcal{O}_1$, can be represented as a integral ideal I connecting $\mathcal{O}_0$ and $\mathcal{O}_1$. Integral ideals are fractional ideals such that $I \subseteq \mathcal{O}_L(I)$, with $\mathcal{O}_L(I) = \{\alpha \in \mathbf{B}_{p,\infty} | \alpha I \subseteq I\}$. Similarly, there exists $\mathcal{O}_R(I) = \{\alpha \in \mathbf{B}_{p,\infty} | I\alpha \subseteq I\}$. All ideals can be seen as $(\mathcal{O}_L(I), \mathcal{O}_R(I))$-ideal with both $\mathcal{O}_L(I)$ and $\mathcal{O}_R(I)$ maximal orders whenever I is integral. The *norm* of an ideal is defined as $n(I) = \gcd(\{n(\alpha) | \alpha \in I\})$.

Let $\phi : E \to E'$ be an isogeny between two supersingular curves. Let $\mathcal{O}_E$ and $\mathcal{O}_{E'}$ be the maximal orders of $\mathbf{B}_{p,\infty}$ corresponding to $\mathrm{End}(E)$ and $\mathrm{End}(E')$. The *kernel ideal* of ϕ is defined as $I_\phi = \{\alpha \in \mathcal{O}_E | \alpha(\ker(\phi)) = 0\}$ Conversely, given I an $(\mathcal{O}_E, \mathcal{O}_{E'})$-ideal, it induces an isogeny $\phi_I : E \to F$ given by $\ker \phi_I = E[I] = \{P \in E \mid \alpha(P) = 0 \ \forall \alpha \in I\}$. The Deuring correspondence induces the following equivalences:

supersingular j-invariants over $\mathbb{F}_{p^2}$	**maximal orders in $\mathbf{B}_{p,\infty}$**
$j(E)$	$\mathcal{O}_E$
$\phi \circ \psi$	$I_\psi I_\phi$
$\deg(\phi)$	$n(I_\phi)$
$\widehat{\phi}$	$\overline{I_\phi}$
$\psi_* \phi$	$[I_\psi]_* I_\phi = \frac{1}{n(I_\psi)} \overline{I_\psi}(I_\psi \cap I_\phi)$
$\gamma \in \mathrm{End}(E)$	$\mathcal{O}_E \gamma$

Kani's Lemma: Lastly, an important recent notion in isogenies is Kani's Lemma [24] and especially its usage to break SIDH, as proposed in [8,29,34], where it was used to embed isogenies between elliptic curve into higher dimensional isogenies. Inside this paper, we soly focus on principally polarized abelian varieties and therefore omit the notion of polarization. We refer the interested reader to Milne's book [30]. The only exception is that we denote the dual of an high dimension isogeny ϕ as $\tilde{\phi}$, its polarized dual. We give here the Kani's Lemma as defined in [34, lemma 3.2].

Lemma 1. *: Let $f : A \to B$, $g : A \to A'$, $f' : A' \to B'$ and $g' : B \to B'$, be polarized separable isogenies such that $g' \circ f = f' \circ g$. Then, the map $F : B \times A' \to A \times B'$ given by the matrix $\begin{pmatrix} \tilde{f} & -\tilde{g} \\ g' & f' \end{pmatrix}$ is a polarised separable isogeny with* $\deg(F) = \deg(f) + \deg(g)$, $\ker(F) = \{(f(P), -g(P)) | P \in A[D]\}$ *and* $\ker(\widetilde{F}) = \{(-\tilde{g}(P), f'(P)) | P \in A'[D]\}$.

Furthermore, given $\deg(F) = d_1 d_2$, then we can write $F = F_2 \circ F_1$ with $\deg(F_1) = d_1$ and $\deg(F_2) = d_2$ such that

$$\begin{tikzcd} & V & \\ B \times A' \arrow[ur, "F_1"] \arrow[rr, "F"] & & A \times B' \arrow[ul, "\widetilde{F_2}"'] \end{tikzcd}$$

$$\ker(F_1) = \left\{(f(P), g(P)) \middle| P \in A[d_1]\right\} \ \& \ \ker(\widetilde{F_2}) = \left\{(\tilde{f}(P), g'(P)) \middle| P \in B[d_2]\right\}$$

2.2 UPKE

We base our definition of UPKE on the notion of symmetric UPKE of [19].

Definition 1. *Given λ a security parameter, an* UPKE *scheme is given by a set of 6* $\mathsf{PPT}(\lambda)$ *together with a setup algorithm* $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ *with* pp *the public parameters.*

- $\mathsf{KG}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{sk}, \mathsf{pk})$
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \xrightarrow{\$} \mathsf{ct}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \longrightarrow \mathsf{m}$
- $\mathsf{UG}(\mathsf{pp}) \xrightarrow{\$} \mu$
- $\mathsf{Upk}(\mathsf{pk}, \mu) \longrightarrow \mathsf{pk}'$
- $\mathsf{Usk}(\mathsf{sk}, \mu) \longrightarrow \mathsf{sk}'$

Likewise to PKE, they also must ensure correctness*: For all $i \in \mathbb{N}$, we have that*

$$\mathbb{P}\left[\mathsf{Dec}\big(\mathsf{sk_i}, \mathsf{Enc}(\mathsf{pk_i}, \mathsf{m})\big) = \mathsf{m} \;\middle|\; \begin{array}{c} (\mathsf{sk_0}, \mathsf{pk_0}) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}), \mu_\mathsf{i} \xleftarrow{\$} \mathsf{UG}(\mathsf{pp}), \\ \big(\mathsf{sk_i}, \mathsf{pk_i}\big) \longleftarrow \big(\mathsf{Usk}(\mathsf{sk_{i-1}}, \mu_\mathsf{i}), \mathsf{Upk}(\mathsf{pk_{i-1}}, \mu_\mathsf{i})\big) \end{array}\right] = 1$$

We make a slight abuse of notation, as all algorithms know of pp, but this choice is made to clarify already heavy notations. The idea behind the security of an UPKE is to be a secure PKE with a key update mechanism that ensures both *Forward Security* and *Post-Compromise Security*. The first notion means that if the adversary learns about sk_i, then it can not use this information to retrieve sk_j for $j < i$ without knowing the update values $\mu_{j+1}, \cdots, \mu_i$. Similarly, the second notion induces that the adversary is not able to retrieve sk_j for $j > i$ without knowing the update values $\mu_{i+1}, \cdots, \mu_j$.

To ensure that those security notions are respected and to enable the adversary to adaptively choose updates, we use the following oracles and lists. We denote as $\mathsf{Oracles}$ the list of all oracles.

- $\mathsf{Upd\_list}$ and $\mathsf{Cor\_list}$ are two lists that respectively store the updates made by the adversaries and what keys are corrupted.
- $\mathsf{Fresh\_Upd}$: The *Fresh-Update oracle* samples a random update μ_i, computes the updated keys $(\mathsf{sk_{i+1}}, \mathsf{pk_{i+1}})$ and return $\mathsf{pk_{i+1}}$.
- $\mathsf{Given\_Upd}$: The *Given-Update oracle* computes the keys $(\mathsf{sk_{i+1}}, \mathsf{pk_{i+1}})$ corresponding to a given update μ_i and return $\mathsf{pk_{i+1}}$. The update $(i, i+1)$ is added to $\mathsf{Upd\_list}$.
- $\mathsf{Corrupt}$: The *Corruption oracle* that receive an index j and return sk_j. It marks j as corrupted together with all others keys of index i such that there is no fresh update in-between.
- $\mathsf{Plaintext\_Check}$ the *plaintext checking oracle* that receives a plaintext and a ciphertext and returns if the ciphertext is a valid encryption of the plaintext.

We use the later oracles to construct the security notion of One-wayness *One-Wayness under Plaintext Checking Attack with Updatability* (OW-PCA-U). Here, instead of distinguishing between the ciphers of two chosen messages, as it is done in IND-CPA-U [19, Figure 1], the adversaries have to decrypt a challenge ciphertext. An UPKE is *OW-PCA-U secure* if for any given $(\mathcal{A}_1, \mathcal{A}_2)$ $\mathsf{poly}(\lambda)$ adversaries, we have that

$$\mathsf{Adv}^{\mathsf{IND\text{-}PCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2) = \mathbb{P}\left[\mathcal{G}^{\mathsf{OW\text{-}PCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2) = 1\right] \leqslant \mathsf{negl}(\lambda)$$

with $\mathcal{G}^{\mathsf{OW\text{-}PCA\text{-}U}}$ the cryptographic game given by figure 1.

```
$\mathcal{G}^{\mathsf{OW\text{-}PCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2)$
1 :  $i = 0$
2 :  $\mathsf{Upd\_list} = \mathsf{Cor\_list} = \emptyset$
3 :  $\mathsf{sk}_0, \mathsf{pk}_0 \xleftarrow{\$} \mathsf{KG}(1^\lambda)$
4 :  $\mathsf{j}, \mathsf{st} \longleftarrow \mathcal{A}_1^{\mathsf{Oracles}}(\mathsf{pk}_0)$
5 :  if $j > i$ do
6 :      return $\perp$
7 :  $\mathsf{m} \xleftarrow{\$} \mathcal{M}$
8 :  $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_\mathsf{j}, \mathsf{m})$
9 :  $\mathsf{n} \longleftarrow \mathcal{A}_2^{\mathsf{Oracles}}(\mathsf{ct}, \mathsf{st})$
10 : if $\mathsf{IsFresh}(j)$ do
11 :     return $m \overset{?}{=} n$
12 : return $\perp$

$\mathsf{Plaintext\_Check}(\mathsf{m}, \mathsf{c}, \mathsf{i}) \to b$
1 :  if $\mathsf{m} \notin \mathcal{M}$ do
2 :      return $\perp$
3 :  else do
4 :      return $\mathsf{m} \overset{?}{=} \mathsf{Dec}(\mathsf{sk}_\mathsf{i}, \mathsf{c})$

$\mathsf{IsFresh}(j)$
1 :  return not $j \overset{?}{\in} \mathsf{Cor\_list}$

$\mathsf{Fresh\_Upd}() \to \mathsf{pk}_\mathsf{i}$
1 :  $\mu \xleftarrow{\$} \mathsf{UG}(1^\lambda)$
2 :  $\mathsf{sk}_{\mathsf{i}+1} \xleftarrow{\$} \mathsf{Usk}(\mathsf{sk}_\mathsf{i}, \mu)$
3 :  $\mathsf{pk}_{\mathsf{i}+1} \xleftarrow{\$} \mathsf{Upk}(\mathsf{pk}_\mathsf{i}, \mu)$
4 :  $i \leftarrow i + 1$
5 :  return $\mathsf{pk}_\mathsf{i}$

$\mathsf{Given\_Upd}(\mu) \to \mathsf{pk}_\mathsf{i}$
1 :  $\mathsf{sk}_{\mathsf{i}+1} \xleftarrow{\$} \mathsf{Usk}(\mathsf{sk}_\mathsf{i}, \mu)$
2 :  $\mathsf{pk}_{\mathsf{i}+1} \xleftarrow{\$} \mathsf{Upk}(\mathsf{pk}_\mathsf{i}, \mu)$
3 :  $\mathsf{Upd\_list} \longleftarrow \mathsf{Upd\_list} \cup \{(i, i+1)\}$
4 :  $i \leftarrow i + 1$
5 :  return $\mathsf{pk}_\mathsf{i}$

$\mathsf{Corrupt}(j) \to \mathsf{sk}_\mathsf{j}$
1 :  $\mathsf{Cor\_list} = \mathsf{Cor\_list} \cup \{j\}$
2 :  $i, k \leftarrow j$
3 :  while $(i-1, i) \in \mathsf{Upd\_list}$ do :
4 :      $\mathsf{Cor\_list} = \mathsf{Cor\_list} \cup \{i-1\}$
5 :      $i \leftarrow i - 1$
6 :  while $(k, k+1) \in \mathsf{Upd\_list}$ do :
7 :      $\mathsf{Cor\_list} = \mathsf{Cor\_list} \cup \{k+1\}$
8 :      $k \leftarrow k + 1$
9 :  return $\mathsf{sk}_\mathsf{j}$
```

Fig. 1. OW-PCA-U Game

2.3 Used Algorithms

SILBE often alternate between different representations of isogenies, more specifically its kernel, ideals and higher dimensional representations. To do so, we use the following standard algorithms in Isogeny Based Cryptography:

- **KernelToIsogeny**: Takes as input E, K with E a supersingular curve and $K \in E[d]$ and return ϕ the isogeny of degree d whose kernel is generated by K together with E', its codomain. To do so, it uses Vélu's Formulas [37] and factorises ϕ as a composition of prime degree isogenies. To be efficient, this requires for d to be smooth.
- **CanonicalTorsionBasis**: Takes as input E a supersingular curve and N an integer such that $N|(p^2-1)$ and return $\langle P, Q\rangle = E[N]$. To do so, it simply samples points at random in $E(\mathbb{F}_{p^2})$ or its quadratic twist and multiplies it by the right cofactor. To ensure that this method is deterministic, the sampling is performed deterministically.
- **PushEndRing** [12, algorithm 8]: Takes as input $\mathfrak{O}_E$ an evaluation basis of $\text{End}(E)$, $\varphi : E \to F$ an isogeny of degree d that is efficiently computable together with its ideal I_φ. It outputs $\mathfrak{O}_F$ a d-evaluation basis of $\text{End}(F)$. An evaluation basis [12, definition A.4.1] consist in an isomorphism between the endomorphism ring and a maximal order such that every element of the basis is efficiently computable [12, definition 1.1.1].
- **KernelToIdeal**[12, algorithm 9]: Takes as input $\mathfrak{O}_E$ a N-evaluation basis of $\text{End}(E)$ and K a generator of the kernel of an isogeny ϕ of smooth degree d coprime to N and return I_ϕ.
- **EvalTorsion**[12, algorithm 11]: It takes as input $\mathfrak{O}_F$ an evaluation basis of $\text{End}(F)$, $\rho_1 : F \to E$ of degree $d_1, \rho_2 : F \to E'$ of degree d_2, both efficiently computable isogenies together with their respective ideals I_1 and I_2. It also takes as input J an $(\mathcal{O}_E, \mathcal{O}_{E'})$-ideal of norm N coprime to d_1 and d_2. It outputs $\phi_J(P)$, with P any point whose order is coprime to $d_1 d_2$.
- **RandomEquivalentIdeal**[26, algorithm 6]: It takes as input a $(\mathcal{O}_E, \mathcal{O}_F)$-ideal I and returns J another $(\mathcal{O}_E, \mathcal{O}_F)$-ideal such that $n(J)$ is a "small" prime, meaning that $n(J) \in [\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$ with extremly high probability, as shown by [26, lemma 3.2.3 & 3.2.4].
- **ConstructKani** [33]: It takes as input d the degree of an isogeny $\phi : E \to E'$ together with N_1 and N_2 two divisors of N such that $N_1 N_2 \geq d$. It also takes as input $P, Q, \phi(P), \phi(Q)$ with the first two points a basis of $E[N]$. It returns F an isogeny of dimension $2g$ with $g = 1, 2$ or 4 that is in fact the Kani's isogeny induced by the following diagram:

$$\begin{tikzcd}
E^g \arrow[r, "\phi^g"] \arrow[d, "\alpha"'] & F^g \arrow[d, "\alpha"] \\
E^g \arrow[r, "\phi^g"] & F^g
\end{tikzcd}$$

with α an endomorphism of dimension $1, 2$ or 4 depending of $N_1N_2 - d$. We also denote as **EvalKani** the algorithm that uses this high dimension isogeny to evaluate $\phi(R)$ for any $R \in E$.

2.4 M-SIDH

Following the breaking of SIDH in [8,29,34], some countermeasures were porposed among which is the Masked-SIDH or M-SIDH [20,21]. The central idea comes from the fact that masking the sent torsion points in the SIDH still enables to compute the pushforwards while protecting against **EvalKani**, as the received torsion points describe the isogeny $[m]\phi$ whose degree is greater than N. Nevertheless, using Weil pairing and given $[m]\phi\binom{P}{Q}$ with P, Q a basis of $E[A]$, we can retrieve $m^2 \mod A$. Finding the mask m is therefore equivalent to finding the right square root of m^2 in $\mathbb{Z}_A$. Thus, to be secure, we need for A to be such that $\mathbb{Z}_A$ has many roots of the unity, i.e. that $A = \prod_{i=1}^{n} p_i$ with n large and p_i distinct odd primes. This is the general idea behind the M-SIDH that we now describe as presented in [21]. The M-SIDH is given in figure 2 and its public parameter be as follows:

- $p = ABf - 1$ a prime number such that with $A = \prod_{i=1}^{n_A} p_i$ and $B = \prod_{j=1}^{n_B} q_j$ coprime such that $A \simeq B$ and $n_A \simeq n_B$.
- E a starting supersingular curve with $\langle P_A, Q_A\rangle$ a basis of $E[A]$ and $\langle P_B, Q_B\rangle$ a basis of $E[B]$.
- Both Alice and Bob that can efficiently sample at random over $\mu_2(A) = \left\{x \in \mathbb{Z}_A \middle| x^2 = 1\right\}$ and $\mu_2(B)$.

It was shown in [21] that the key security of M-SIDH reduces to the following problem with adequate N and d.

Problem 1. Supersingular isogeny problem with masked torsion point information: Let $\phi : E \to E'$ be an isogeny of degree d, let $\langle P, Q\rangle$ be a basis of $E[N]$ with $N = \prod_{i=1}^{n} p_i$ coprime to d and let $m \in \mu_2(N)$ be a random element. Given $P, Q, [m]\phi(P), [m]\phi(Q)$, compute ϕ.

Importantly, it is not sufficient to ask for n_A and n_B to be around λ, as following [21, Theorem 7], it suffices to find $m \mod N_t$, with $N_t = \prod_{i=t}^{n} p_i$ such that $N_t \geq \sqrt{d}$. This is because we have enough torsion points on N_t to use **EvalKani** efficiently and thus retrieve ϕ. Then, as $m \in \mu_2(N)$, we have that $m \mod N_t \in \mu_2(N_t)$ with $|\mu_2(N_t)| = 2^{n-t}$, meaning that we have significantly diminished the numbers of possible masks. To ensure the security of M-SIDH, we need for A and B to be such that for all $A_t = \prod_{i=t}^{n_A} p_i$, we have that $A_t \geq \sqrt{B} \Rightarrow n_A - t \geq \lambda$ and similarly for B. This induces that, in the case of SIDH, we need around $n_A + n_B \simeq 4.5\lambda$.

Another important attack on M-SIDH and problem 1 is the *generalized lollipop attack*, as detailed in [10]. It requires that the domain[2] of the mask isogeny

[2] or codomain, using duality.

M-SIDH

Alice(pp)	**Bob**(pp)
$s_A \leftarrow_\$ \mathbb{Z}_A, \alpha \leftarrow_\$ \mu_2(B)$	$s_B \leftarrow_\$ \mathbb{Z}_B, \beta \leftarrow_\$ \mu_2(A)$
$R_A \leftarrow P_A + [s_A]Q_A$	$R_B \leftarrow P_B + [s_B]Q_B$
$\phi_A, E_A \leftarrow$ **KernelToIsogeny**(E, R_A)	$\phi_B, E_B \leftarrow$ **KernelToIsogeny**(E, R_B)
$S_A \leftarrow [\alpha]\phi_A(P_B)$	$S_B \leftarrow [\beta]\phi_B(P_A)$
$T_A \leftarrow [\alpha]\phi_A(Q_B)$	$T_B \leftarrow [\beta]\phi_B(Q_A)$

$$\xrightarrow{E_A, S_A, T_A}$$

$$\xleftarrow{E_B, S_B, T_B}$$

$U_A \leftarrow S_B + [s_A]T_B$	$U_B \leftarrow S_A + [s_B]T_A$
$\psi_A, E_K \leftarrow$ **KernelToIsogeny**(E_B, U_A)	$\psi_B, E_K \leftarrow$ **KernelToIsogeny**(E_A, U_B)
$K \leftarrow \mathsf{KDF}\big(j(E_K)\big)$	$K \leftarrow \mathsf{KDF}\big(j(E_K)\big)$

Fig. 2. M-SIDH protocol

ϕ is defined over $\mathbb{F}_p$ to construct a new unmasked isogeny ψ, use **EvalKani** over ψ to retrieve $\ker(\psi)$ and extract $\ker(\tilde{\phi})$ from $\ker(\psi)$.

To be more specific, let $\phi : E_0 \to E$ be an isogeny of degree d, with E_0 defined over $\mathbb{F}_p$. Let $\langle P, Q\rangle$ be a basis of $E_0[N]$ and S, T to be the masked image of those points, i.e. $\binom{S}{T} = [m]\phi\binom{P}{Q}$. We then consider the following diagram, where we denote as $\phi^{(p)}$ the pushforward $\pi_*\phi$. Because E_0 is defined over $\mathbb{F}_p$, we have that $\pi \in \mathrm{End}(E_0)$ and its pushforward is well-defined. We set $\psi = \phi^{(p)} \circ \widehat{\phi}$.

$$\begin{tikzcd}
& E \arrow[dd, dashed, "\pi"] \arrow[dd, bend left, "\psi"] \\
E_0 \arrow[loop left, dashed, "\pi"] \arrow[ur, "\phi"] \arrow[dr, "\phi^{(p)}"'] & \\
& E^{(p)}
\end{tikzcd}$$

We then use the following lemma.

Lemma 2. *[10, Lemma 3]: Using the above notation, assume that the matrix* $\mathbf{M}_{\widehat{\pi}}$ *is such that* $\widehat{\pi}\binom{P}{Q} = \mathbf{M}_{\widehat{\pi}}\binom{P}{Q}$ *Then, we can compute* $\psi(E[N])$ *as*

$$\psi\begin{pmatrix} S \\ T \end{pmatrix} = dp^{-1}\mathbf{M}_{\widehat{\pi}}\pi\begin{pmatrix} S \\ T \end{pmatrix} \mod N$$

As we can evaluate ψ over $E[N]$ and we have that $\deg(\psi) = d^2 \le N^2$, we can use **EvalKani** over ψ to evaluate ψ over any points and in particular over $E[d]$.

We can then extract $\ker(\widehat{\varphi})$ from $\ker(\psi)[d]$, depending on the relation between d and p, as detailed in [10, section 3.2].

3 Constructing a PKE from the generalized lollipop attack

The core concept behind SILBE is to leverage the generalized lollipop attack over the M-SIDH as a deciphering mechanism, akin to how the original lollipop attack was employed in designing SETA [35]. This endeavor will make usage of all the different isogeny representations that we detailed in section 2.1. SILBE is in fact related to [10, section 4.3] and the idea of M-SIDH with trapdoor curves, although there are substantial changes. The PKE part of SILBE works as follows:

- Setup: We find the adequate β and N to construct a base prime $p = 3^\beta N f + 1$ such that $p = 3 \mod 4$ and $N = \prod_{i=1}^n p_i$ with n big enough such that it is secure. We also compute P_0, Q_0 a basis of $E_0[N]$ and U_0, V_0 a basis of $E_0[3^\beta]$. We compute a matrix $\mathbf{M}_\pi$ that represent the action of π over P_0, Q_0.
- KG: Alice computes a long isogeny between E_0 and E_A. Using **EvalKani**, it retrieves the representing ideal I and use the **RandomEquivalentIdeal** algorithm to find a short connecting isogeny $\phi_A : E_0 \to E_A$. E_A is then used as the public key while ϕ_A is the secret key.
- Enc: Bob computes $\phi_B : E_A \to E_B$ an isogeny. It then sends the masked image by ϕ_B of a basis $E_A[N]$, where the mask is the message m.
- Dec: Using its knowledge of ϕ_A, Alice uses the generalized lollipop attack over $\phi_B \circ \phi_A$ to retrieve $\ker(\widehat{\phi_B})$ and using the discrete logarithm, it retrieves m.

The public parameters of SILBE are constructed using the following Setup algorithm. It uses **EvalImageMatrix**, a small subroutine based on Weil's pairing that given P, Q a basis of $E[N]$, with N smooth and $X, Y \in E[N]$ compute $\mathbf{M}$ the matrix such that $\binom{X}{Y} = \mathbf{M}\binom{P}{Q}$. We discuss more thoroughly how we construct p in section 5. We also denote as $\mathfrak{O}_0$ the efficient evaluation basis of $\mathrm{End}(E_0)$, that we detailed in section 2.1, with E_0 the curve with j-invariant 1728 defined over $\mathbb{F}_p$.

3.1 Key generation

As touched earlier, the key generation of SILBE constructs a long isogeny walk with starting curve E_0. This is done to use the following proposition.

Proposition 1. *[12, proposition B.2.1]: Let $\phi : E \to E'$ be an ℓ^h-isogeny obtained from a non-backtracking random ℓ-isogeny walk over $\mathcal{G}_p^\ell$. Then, for all $\epsilon \in]0, 2]$, the distribution of E' has statistical distance $O(p^{-\epsilon/2})$ to the uniform distribution in the supersingular isogeny graph, provided that $h \geq (1+\epsilon)\log_\ell(p)$.*

Algorithm 1 SILBE.Setup

```
Input: $1^\lambda$
Output: $\mathsf{pp} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big)$ with $p$ a prime, $\langle P_0, Q_0\rangle = E_0[N]$, $\langle U_0, V_0\rangle = E_0[3^\beta]$, $\mathbf{M}_\pi \in \mathrm{GL}_2(N)$ and $t$ an integer such that $3^{\beta t} \geq p^2$.
1: Take $p$ a prime of the form $p = 3^\beta N f + 1$ such that $p = 3 \mod 4$ and $N = \prod_{i=1}^n p_i$
   with $p_i$ distinct odd small prime numbers such that $N \geqslant 3^\beta p^{1/2}\log(p)^2$, $N$ is
   coprime to 3 and $n$ big enough such that for all $N_k = \prod_{i=k}^n p_i$, we have that
   $N_k \geq \sqrt{3^\beta} \Rightarrow n - k \geq \lambda$.
2: $P_0, Q_0 \leftarrow$ CanonicalTorsionBasis$(E_0, N)$
3: $U_0, V_0 \leftarrow$ CanonicalTorsionBasis$(E_0, 3^\beta)$
4: $\mathbf{M}_\pi \leftarrow$ EvalImageMatrix$(E_0, P_0, Q_0, \pi(P_0), \pi(Q_0))$.
5: $t \leftarrow \left\lceil \frac{2\log_2(p)}{\beta\log_2(3)} \right\rceil$
6: $\mathsf{pp} \leftarrow \big(p, N, P_0, Q_0, U_0, V_0, \mathbf{M}_\pi, t\big)$.
7: return pp
```

By constructing a path of length t of 3^β-isogenies $\rho_1, \cdots, \rho_t$, we get that the degree of their composition is greater than p^2 and the end curve distribution will be $O(p^{-1/2})$ statistically close from the uniform distribution, meaning that it will be computationally indistinguishable from an uniform random sampling. We call the end curve E_A. To compute $I_1, \cdots I_t$ the ideals corresponding to $\rho_1, \cdots, \rho_t$. we use the following recursive mechanism:

$$\begin{tikzcd}
E_0 \arrow[r, "\rho_1"] \arrow[r, bend left, "J_1"] \arrow[rrr, bend left=40, "J_{t-1}"] \arrow[rrrr, bend left=50, "J_t"] \arrow[rrrr, bend right, "I_{\phi_A}"'] & E_1 \arrow[r, "\rho_2"] & \cdots \arrow[r, "\rho_{t-1}"] & E_{t-1} \arrow[r, "\rho_t"] & E_A
\end{tikzcd}$$

Fig. 3. Diagram of the Key Generation of SILBE

1. Assume knowledge of $\kappa_i : E_0 \to E_i$ together with its representative ideal J_i such that $n(J_i)$ is prime and coprime to 3. Furthermore, assume knowledge of $\mathfrak{D}_{E_i}$ a T-evaluation basis over E_i with $T \neq 3$ prime. Finally, assume knowledge of I_j with $1 \leq j \leq i$.
2. Using **KernelToIsogeny**, we can construct ρ_{i+1} and find E_{i+1} and using $\mathfrak{D}_{E_i}$ we can find I_{i+1} with the **KernelToIdeal**.
3. Then, we have that $J_i I_{i+1}$ is a $(\mathcal{O}_0, \mathcal{O}_{E_{i+1}})$-ideal, using **RandomEquivalentIdeal**, we find an ideal J_{i+1} such that $n(J_i) \neq n(J_{i+1})$ and $n(J_{i+1}) \in$

$[\sqrt{p}\log(p), \sqrt{p}\log(p)]$ is prime. To speed-up computations, we consider $\widetilde{N} = \prod_{i=1}^{x} p_i$ with x minimal such that $\widetilde{N} \geq p^{1/4}\log(p)^{1/2}$ and ask for $\widetilde{N}^2 - n(J_i)$ to be prime and equal to $1 \mod 4$.

$$\begin{tikzcd}
E_0 \arrow[r, "J_i", "\kappa_i"'] \arrow[dr, "\kappa_{i+1}", "J_{i+1}"'] & E_i \arrow[d, "\rho_{i+1}"', "I_i"] \\
& E_{i+1}
\end{tikzcd}$$

4. Now, using **EvalTorsion** over the above triangle, we evaluate $\kappa_{i+1} = \phi_{J_{i+1}}$ over $\langle P_0, Q_0\rangle = E_0[N]$. We then have constructed a high dimension representation of κ_{i+1}.
5. Using **ConstructKani** over $(P_0, Q_0, \kappa_{i+1}(P_0), \kappa_{i+1}(Q_0))$ in dimension 4 thanks to $\tilde{N}$, we get a Kani's isogeny F_{i+1} and can therefore evaluate κ_{i+1} over any points. This is then used to apply the **PushEndRing** over κ_{i+1} and J_{i+1} to retrieve $\mathfrak{O}_{E_{i+1}}$ a $n(J_{i+1})$-evaluation basis over E_{i+1}.

Using this mechanism, we compute I_i for $i = 1, \cdots, t$. Additionally, we also compute $\mathfrak{O}_{E_A}$ a $n(J_t)$-**EvaluationBasis** of $\mathrm{End}(E_A)$. To speed up the decryption part of SILBE, we use **RandomEquivalentIdeal** over J_t to find another $(\mathcal{O}_0, \mathcal{O}_{E_A})$-ideal I_{ϕ_A} such that $N' - n(I_{\phi_A})^2 3^{2\beta} = 1 \mod 4$ and is a prime number, with $N' = p_1 \cdot \prod_{i=2}^{n} p_i^2$. This ensures that the **EvalKani** in **Generalisedlolli** is performed in dimension 4. The reason behind the choice of N' and not N^2 comes from the fact $N^2 - n(I_{\phi_A})^2 3^{2\beta} = (N - n(I_{\phi_A})3^{\beta})(N + n(I_{\phi_A})3^{\beta})$ and can therefore never be prime.

Once found, we use **EvalTorsion** over $\rho_t \circ \cdots \circ \rho_1$ and $I_1 \cdots I_t$ to evaluate $\phi_A\binom{P_0}{Q_0}$ and use **EvalImageMatrix**, to compute the matrix $\mathbf{M}_{\phi_\mathbf{A}}$ such that $\phi_A\binom{P_0}{Q_0} = \mathbf{M}_{\phi_\mathbf{A}}\binom{P_A}{Q_A}$.

We then set E_A as the public key and $\mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}$ as the secret key. We construct ρ_i in such a way that our walk cannot be backwards. To do so, we use U_i, V_i a basis of $E_i[3^\beta]$ such that $\rho_i(E_{i-1}[3^\beta]) = \langle V_i\rangle$. As we set $\ker(\rho_{i+1}) = \langle U_i + [\eta_{i+1}]V_i\rangle$, we have that it can be any cyclic isogeny of degree 3^β except $\widehat{\rho_i}$.

3.2 Encryption & Decryption

The underlying architecture behind the PKE part of SILBE is given in figure 4

Encryption

As explained, the message space of SILBE is $\mu_2(N) = \left\{x \in \mathbb{Z}_N \middle| x^2 = 1\right\}$. As $N = \prod_{i=1}^{n} p_i$, we have that $|\mu_2(N)| = 2^n$ and we can furthermore construct an efficient mapping between $\{0,1\}^n$ and $\mu_2(N)$ using the Chinese remainder theorem. To encrypt m, Bob starts to compute a random isogeny $\phi_B : E_A \to E_B$ of degree 3^β. Then, similarly to M-SIDH, we compute the image of the N torsion points through this isogeny and mask those points using the message m. The ciphertext is therefore $E_B, [m]\phi_B(P)$ and $[m]\phi_B(Q)$.

Algorithm 2 SILBE.KG

```
Input: $\mathsf{pp} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big)$
Output: $\mathsf{pk}, \mathsf{sk}$ a public/secret key pair.
1: $J_0 \leftarrow \mathcal{O}_0$
2: for $1 \leqslant i \leqslant t$ do
3:     Sample $\eta_i \in_\$ \mathbb{Z}_{3^\beta}$.
4:     $E_i, \rho_i \longleftarrow$ KernelToIsogeny$\big(E_{i-1}, (U_{i-1} + [\eta_i]V_{i-1})\big)$    ▷ Already in $\mathsf{pp}$ if $i = 1$.
5:     $I_i \longleftarrow$ KernelToIdeal$\big(\mathfrak{O}_{E_{i-1}}, (U_{i-1} + [\eta_i]V_{i-1})\big)$
6:     Deterministically compute $U_i, V_i$ a basis of $E_i[3^\beta]$ with $\langle V_i \rangle = \rho_i(E_{i-1}[3^\beta])$.
7:     $J_i \longleftarrow$ RandomEquivalentIdeal$(J_{i-1} I_i)$
8:     if $n(J_i) = n(J_{i-1})$ or and $\widetilde{N}^2 - n(J_i) \neq 1 \mod 4$ or is not prime do go back to line 7.
9:     $S_i, T_i \longleftarrow$ EvalTorsion$(\mathfrak{O}_0, \rho_i \circ \kappa_{i-1}, J_{i-1} I_i, id, J_i, \{P_0, Q_0\})$
10:    $F_i \longleftarrow$ ConstructKani$\big(n(J_i), \widetilde{N}, \widetilde{N}, (P_0, Q_0, S_i, T_i)\big)$
11:    $\mathfrak{O}_{E_i} \longleftarrow$ PushEndRing$(\mathfrak{O}_0, \kappa_i, J_i)$    ▷ $\kappa_i \leftarrow F_i(0, 0, -, 0)_3$
12: $I_{\phi_A} \longleftarrow$ RandomEquivalentIdeal$\big(J_t\big)$
13: if $N' - n(I_{\phi_A})^2 3^{2\beta} \neq 1 \mod 4$ or is not prime do go back to line 12.
14: $K, L \longleftarrow$ EvalTorsion$(\mathcal{O}_0, \rho_t \circ \cdots \circ \rho_1, I_1 \cdots I_t, 1, I_{\phi_A}, P_0, Q_0)$
15: $\mathbf{M}_{\phi_\mathbf{A}} \longleftarrow$ EvalImageMatrix$(E_t, N, P_t, Q_t, K, L)$
16: $\mathsf{pk} \longleftarrow \big(E_t = E_A\big)$
17: $\mathsf{sk} \longleftarrow \big(\mathfrak{O}_{E_t}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}\big)$
18: return $\mathsf{pk}, \mathsf{sk}$.
```

$$\begin{tikzcd}
 & E_A \arrow[r, "\phi_B"] \arrow[d, dashed, "\pi"] & E_B \arrow[d, dashed, "\pi"'] \arrow[d, bend left, "\psi"] \\
E_0 \arrow[loop left, dashed, "\pi"] \arrow[ur, "I_{\phi_A}"] \arrow[dr, "{[\pi]_* I_{\phi_A}}"'] & & \\
 & E_A^{(p)} \arrow[r, "\phi_B^{(p)}"] & E_B^{(p)}
\end{tikzcd}$$

Fig. 4. Diagram of the encryption/decryption of SILBE, Alice in red and Bob in blue

Decryption

As previously stated, we use the generalized lollipop over $\phi_B \circ \phi_A$ to decipher our message. Indeed, using the torsion points in ct, we can define $\binom{S}{T} = [\mathsf{m}]\phi_B \circ \phi_A \binom{P_0}{Q_0}$. Theses points are easily computable using sk as

$$[\mathsf{m}]\phi_B \circ \phi_A \begin{pmatrix} P_0 \\ Q_0 \end{pmatrix} = [\mathsf{m}]\mathbf{M}_{\phi_\mathbf{A}} \phi_B \begin{pmatrix} P_A \\ Q_A \end{pmatrix} = \mathbf{M}_{\phi_\mathbf{A}} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

We modify the generalized lollipop attack of [10, section 4] such that it just computes $\ker(\widehat{\phi_B})$ and not the whole $\ker(\widehat{\phi_B \circ \phi_A})$. This speed up the decryption.

Algorithm 3 SILBE.Enc

```
Input: $\mathsf{pp}, \mathsf{pk}, \mathsf{m} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big), E_A$ with $\mathsf{m} \in \mu_2(N)$
Output: $\mathsf{ct} = (E_B, R_1, R_2)$ with $R_1, R_2 \in E_B[N]$.
1: $P_A, Q_A \longleftarrow$ CanonicalTorsionBasis$(E_A, N)$
2: $U_A, V_A \longleftarrow$ CanonicalTorsionBasis$(E_A, 3^\beta)$
3: Sample $r_B \in_\$ \mathbb{Z}_{3^\beta}$
4: $E_B, \phi_B \longleftarrow$ KernelToIsogeny$\big(E_A, (U_A + [r_B]V_A)\big)$
5: $\binom{R_1}{R_2} \longleftarrow [\mathsf{m}]\phi_B\binom{P_A}{Q_A}$
6: $\mathsf{ct} \longleftarrow (E_B, R_1, R_2)$
7: return ct
```

We consider the following isogeny

$$\psi : E_B \longrightarrow E_B^{(p)}$$

$$\psi = (\phi_B \circ \phi_A)^{(p)} \circ \phi_A \circ \phi_B = \phi_B^{(p)} \circ \phi_A^{(p)} \circ \phi_A \circ \phi_B$$

Using lemma 2, we can evaluate ψ over $E_B[N]$ as

$$\psi\begin{pmatrix} S \\ T \end{pmatrix} = n(I_{\phi_A})3^\beta \mathbf{M}_\pi{}^{-1}\pi\begin{pmatrix} S \\ T \end{pmatrix} = n(I_{\phi_A})3^\beta \mathbf{M}_\pi{}^{-1}\mathbf{M}_{\phi_\mathbf{A}}\pi\begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

We then use **EvalKani** over ψ to evaluate $\widehat{\psi}$ over $E_B^{(p)}[3^\beta]$. Due to the nature of $N' - n(I_{\phi_A})^2 3^{2\beta}$, this is done in dimension 4. We then have, following [10, section 3.2] that $\widehat{\psi}(E_B^{(p)}[3^\beta]) = \ker(\psi)[3^\beta] = \ker(\widehat{\phi_B})$. The reason comes from our good choice of public parameters, as $p - 1 = 0 \mod 3$ and thus $\left(\frac{-p}{3}\right) = -1$, meaning that 3 is inert inside $\mathbb{Z}[\sqrt{-p}]$ and in $\mathbb{Z}[\sqrt{\chi}]$ with $\chi \in \mathrm{End}(E)$ a lollipop endomorphism defined as $\chi = \widehat{\pi} \circ \phi_A^{(p)} \circ \widehat{\phi_A} = [-1]\phi_A \circ \pi \circ \widehat{\phi_A}$ such that $\chi^2 = [-p(\deg \phi_A)^2]$. We thus know $\ker(\widehat{\phi_B})$, so we can thus use **KernelToIsogeny** to compute $\widehat{\phi_B}(R_1) = [\mathsf{m}3^\beta]P_A$ and retrieve m using the discrete logarithm over $E[N]$.

Algorithm 4 SILBE.Dec

```
Input: $\mathsf{pp}, \mathsf{sk}, \mathsf{ct} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big), (\mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}), (E_B, \mathsf{R}_1, \mathsf{R}_2)$
Output: m
1: $P_A, Q_A \longleftarrow$ CanonicalTorsionBasis$(E_A, N)$
2: $U_B, V_B \longleftarrow$ CanonicalTorsionBasis$(E_B^{(p)}, 3^\beta)$
3: $\binom{S}{T} \longleftarrow \mathbf{M}_{\phi_\mathbf{A}}\binom{R_1}{R_2}$
4: $\binom{K}{L} \longleftarrow [n(I_{\phi_A})3^\beta]\mathbf{M}_\pi^{-1}\pi\binom{S}{T}$
5: $G, H \longleftarrow$ EvalKani$\big(n(I_{\phi_A})^2 3^{2\beta}, N, N/p_1, S, T, K, L, U_B, V_B\big)$   ▷ $\widehat{\psi} = F(-, 0, 0, 0)_1$
6: $\widehat{\phi_B} \longleftarrow$ KernelToIsogeny$(E_B, G + H)$   ▷ if $G = H$, take just $G$
7: return $(3^\beta)^{-1} \cdot \big($discretelog$(P_A, \widehat{\phi_B}(R_1), N)\big) \mod N$
```

3.3 Security

First and foremost, we see that SILBE is not IND-CPA secure. Indeed, To distinguish between two known messages m_0 and m_1, we simply have to multiply R_1 and R_2 by m_0 and use **EvalKani** in dimension 8. If we are able to retrieve ϕ_B, then this means that the encrypted message was m_0, as that would induce that $[m_0]R_1 = [m_0^2]\phi_B(P) = \phi_B(P)$. Otherwise, this means that the encrypted message was m_1 with overwelming probability. That mechanism can be used to know if a ciphertext ct is the encryption of a plaintext m or not. This induces that any adversary of SILBE can simulate the oracle $\mathsf{Plaintext\_Check}$. This will be useful in the following proposition.

Proposition 2. *The security of SILBE as an OW-PCA PKE reduces to problem 1 over random curves.*

Proof. Using the previously explained method to simulate $\mathsf{Plaintext\_Check}$, we have that

$$\text{SILBE is OW-PCA secure} \iff \text{SILBE is OW-CPA secure}$$

Following proposition 1, we have that the distribution of the public key E_A is $O(p^{-1/2})$ close from the uniform distribution over supersingular curves, meaning that it is computationally indistinguishable. Let $\mathcal{A}^{\mathsf{OW-CPA}}$ be any adversary for SILBE. We can then construct an algorithm $\mathcal{B}$ that solve problem 1 over random curves with the same advantage. $\mathcal{B}$ is defined as such:

1. $\mathcal{B}$ receives as input (P, Q, S, T) with P, Q the canonical basis of $E[N]$ and $\binom{S}{T} = [\mathsf{m}]\varphi\binom{P}{Q}$ with $\varphi : E \to E'$ an isogeny of degree 3^β.
2. It then calls $\mathcal{A}^{\mathsf{OW\text{-}CPA}}\big(E, (E', S, T)\big)$ and receive $\mathsf{n} \in \mu_2(N)$.
3. It then compute $[\mathsf{n}]S, [\mathsf{n}]T$ and use **EvalKani** in dimension 8 over theses points to retrieve $\ker(\varphi)$. As 3^β is smooth, using **KernelToIsogeny**, it can compute φ.

We see that if $\mathcal{A}^{\mathsf{OW-CPA}}$ succeeds, then so does $\mathcal{B}$, meaning that

$$\mathbb{P}[\mathcal{B} \text{ solve problem 1}] \geq \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}^{\mathsf{OW\text{-}CPA}})$$

Thus, under the assumption that problem 1 over random curves is hard, then SILBE is OW-PCA secure.

4 Extending this PKE into an UPKE

SILBE is thus OW-PCA secure[3] and can be made IND-qCCA using for example $U^{\not\perp}$ variant of the Fujisaki-Okamoto transform, as detailed in [23, section 4.2]. We can thus construct PKE from the generalized lollipop attack. We now make of SILBE an UPKE.

[3] and in reality OW-qPCA secure, as problem 1 is believed to be quantum resistant.

4.1 Design

The idea behind SILBE key update mechanism comes from the fact that our key generation mechanism has two excellent properties, namely that it can be adapted to start over any curve E, provided that we know an isogeny $\phi : E_0 \to E$ and that finding the public key can be done by just using **KernelToIsogeny**, without knowledge of $\phi : E_0 \to E$. Our key update mechanism is therefore an adaptation of the key generation. Its architecture is given in figure 5 and is performed as such:

$$\begin{tikzcd}
E_0 \arrow[d, "I_{\phi_A}"'] \arrow[dr, "J_1"] \arrow[drrr, "J_{t-1}"] \arrow[drrrr, "J_t"] \arrow[drrrr, bend left, "I_{\phi'_A}"] & & & & \\
E_A \arrow[r, "{I_1, \rho_1}"'] & E_1 \arrow[r, "{I_2, \rho_2}"'] & \cdots \arrow[r, "{I_{t-1}, \rho_{t-1}}"'] & E_{t-1} \arrow[r, "{I_t, \rho_t}"'] & {E_A}'
\end{tikzcd}$$

Fig. 5. Diagram of the key update mechanism of SILBE, Alice in red and Bob in blue. Black isogenies are used for the construction of **SILBE.Usk**.

- UG: Generate a seed $\mu \in \{0,1\}^{4\log(p)}$.
- Upk: Use a hash function over μ to generate a sequence of elements in $\mathbb{Z}_{3^\beta}$. Use this sequence to create kernels of an isogeny walk starting at the public key E_A. Thanks to **KernelToIsogeny**, we compute the end curve of that walk, defined as E'_A, the updated public key.
- Usk: Use a hash function over μ to generate a sequence of elements in $\mathbb{Z}_{3^\beta}$. Use this sequence to create kernels of an isogeny walk starting at the public key E_A. Thanks to **KernelToIsogeny**, we compute the end curve of that walk, defined as E'_A. Using the knowledge of $\phi_A : E_0 \to E_A$, we construct, using **EvalKani** and **RandomEquivalentIdeal**, an isogeny $\phi'_A : E_0 \to E'_A$, the updated secret key.

Algorithm 5 SILBE.UG

Input: $\mathsf{pp} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big)$
Output: μ an update.

```
1: Sample $\mu \in_\$ \{0,1\}^{4\log(p)}$        ▷ $4\log(p)$ ensures that $H$ resists quantum attacks.
2: return $\mu$
```

Algorithm 6 SILBE.Upk

```
Input: $\mathsf{pp}, \mathsf{pk}, \mu = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big), E_A$.
Output: $\mathsf{pk}'$ the updated public key.
1: $E_0 \leftarrow E_A$    $U_0, V_0 \longleftarrow$ CanonicalTorsionBasis$(E_A, 3^\beta)$
2: $(\eta_1, \cdots, \eta_t) \leftarrow H(\mu)$    ▷ $\eta_i \in \mathbb{Z}_{3^\beta}$
3: for $1 \leqslant i \leqslant t$ do
4:     $E_i, \rho_i \longleftarrow$ KernelToIsogeny$\big(E_{i-1}, (U_{i-1} + [\eta_i]V_{i-1})\big)$
5:     Deterministically compute $U_i, V_i$ a basis of $E_i[3^\beta]$ with $\langle V_i \rangle = \rho_i(E_{i-1}[3^\beta])$.
6: $\mathsf{pk}' \leftarrow E_t = E'_A$
7: return $\mathsf{pk}'$
```

Algorithm 7 SILBE.Usk

```
Input: $\mathsf{pp}$, $\mathsf{sk}$ , $\mu = (p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big), (\mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}), \mu$
Output: $\mathsf{sk}'$ the updated secret key.
1: $E_0 \leftarrow E_A$    $J_0 \leftarrow I_\phi$    $U_0, V_0 \leftarrow$ CanonicalTorsionBasis$(E_A, 3^\beta)$
2: $(\eta_1, \cdots, \eta_t) \leftarrow H(\mu)$    ▷ $\eta_i \in \mathbb{Z}_{3^\beta}$
3: for $1 \leqslant i \leqslant t$ do
4:     $E_i, \rho_i \longleftarrow$ KernelToIsogeny$\big(E_{i-1}, (U_{i-1} + [\eta_i]V_{i-1})\big)$
5:     $I_i \longleftarrow$ KernelToIdeal$\big(\mathfrak{O}_{E_{i-1}}, (U_i + [\eta_i]V_i)\big)$
6:     Deterministically compute $U_i, V_i$ a basis of $E_i[3^\beta]$ with $\langle V_i \rangle = \rho_i(E_{i-1}[3^\beta])$.
7:     $J_i \longleftarrow$ RandomEquivalentIdeal$(J_{i-1}I_i)$
8:     if $n(J_i) = n(J_{i-1})$ or and $\widetilde{N}^2 - n(J_i) \neq 1 \mod 4$ or is not prime do go back
       to line 7.
9:     $S_i, T_i \longleftarrow$ EvalTorsion$(\mathfrak{O}_0, \rho_i \circ \kappa_{i-1}, J_{i-1}I_i, 1, J_i, P_0, Q_0)$    ▷ Use $\mathbf{M}_\phi$ if $i = 1$
10:    $F_i \longleftarrow$ ConstructKani$(n(J_i), \widetilde{N}, \widetilde{N}, P_0, Q_0, S_i, T_i)$
11:    $\mathfrak{O}_{E_i} \longleftarrow$ PushEndRing$(\mathfrak{O}_0, \kappa_i, J_i)$    ▷ $\kappa_i = F(0, 0, -, 0)_3$
12: $I_{\phi'_A} \longleftarrow$ RandomEquivalentIdeal$\big(J_t\big)$
13: if $N' - n(I_{\phi'_A})^2 3^{2\beta} \neq 1 \mod 4$ or is not prime do go back to line 12.
14: $K, L \longleftarrow$ EvalTorsion$(\mathfrak{O}_0, \kappa_t, J_t, 1, I_{\phi'}, P_0, Q_0)$
15: $\mathbf{M}_{\phi'_\mathbf{A}} \longleftarrow$ EvalImageMatrix$(E_t, N, P_t, Q_t, K, L)$
16: $\mathsf{sk}' \longleftarrow \big(\mathfrak{O}_{E_t}, I_{\phi'_A}, \mathbf{M}_{\phi'_\mathbf{A}}\big)$
17: return $\mathsf{sk}'$.
```

4.2 Security

The reason behind the fact that SILBE remains secure as an UPKE comes from the fact that, in the Random Oracle Model (ROM), we have that **SILBE.Upk** is a one way mechanism such that the distribution of the updated public key E'_A is statistically close from the uniform distribution and thus from the public key distribution E_A given by **SILBE.KG**. Therefore, any adversaries capable of breaking SILBE in the OW-PCA-U scenario are also inherently capable of breaking a fresh instance of SILBE in a OW-PCA scenario. [4] This leads us to the following proposition.

[4] More precisely, it is able to break a fresh instance of SILBE chosen among $\mathsf{poly}(\lambda)$ many of them.

Proposition 3. *In the ROM,*

$$\text{SILBE is OW-PCA secure} \iff \text{SILBE is OW-PCA-U secure}$$

Therefore, under the assumption that the problem 1 is hard over random curves, we have that SILBE is a OW-PCA-U secure UPKE.

Importantly, we could slightly change our update mechanism to not require a hash function. This comes from the fact that it is very similar to the CGL hash function [11]. We can thus adapt [11, section 5] and get that the problem of finding μ such that $\textbf{SILBE.Upk}(E, \mu) = E'$ reduces to the *isogeny walk problem* [13, problem 3] and thus that our key update mechanism is one-way. Nevertheless, we would require some modifications of the public key as we would have to add $V_t \in E_A[3^\beta]$ such that $\langle V_t \rangle = \rho_t(E_{t-1}[3^\beta])$ to ensure that the update long isogeny is not backtracking. To keep the same security level, we would also need to compute a slightly longer isogeny. On that note, due to the size of p, we see that we can shorten the length of our path such that our distribution is not $O(p^{-1/2})$ -statistically close from uniform, but just $O(2^{-\lambda})$, which would be sufficient.

To make of SILBE an IND-CCA-U UPKE, we use the transformation provided in [3, section 4]. It transforms a OW-CPA-U UPKE into an IND-CCA-U UPKE[5] in an IND) in the ROM.[6] To do so, we need to show that SILBE is λ-*spread* [3, definition 7] but this is a direct consequence of proposition 1 and of the fact that $3^\beta \gg 2^\lambda$, as we now show.

5 Parameters & Efficiency

5.1 Finding "SILBE" friendly primes

As we previously explained when detailing SILBE's public parameters, we have that the cross relation between β and N forces N to have many prime factors. To find good N and β, we do as follows:

- If $N \leq 3^\beta \sqrt{p} \log(p) \simeq 3^{3\beta/2} N^{1/2} \big(\log(N) + \beta \log(3)\big)$, we increase the size of N.
- If $N_t \geq 3^{\beta/2}$ and $n - t < \lambda$, we increase the size of β.

Once we have found adequate N and β, we find a good cofactor f such that $p = 3^\beta N f + 1$ is prime. Using this method, we found the following parameters, detailed in table 1.

We see that in SILBE, we need N to have slightly less than 7λ distinct prime divisors.

[5] and in reality, in our case, in an IND-qCCA-U, as we rely on problems that are believed to be quantum resistant.

[6] Using their security definition, we indeed have that SILBE is an OW-CR-CPA.

λ	β	N	f	n	$\log_2(p)$
128	2043	$5 \times 7 \times 11 \times \cdots \times 6863$	1298	881	13013
192	3229	$5 \times 7 \times 11 \times \cdots \times 10789$	1790	1312	20538
256	4461	$5 \times 7 \times 11 \times \cdots \times 14879$	16706	1741	28346

Table 1. Parameters for SILBE

5.2 Efficiency of SILBE

The main drawback with SILBE is its efficiency. This essentially comes from the size of the parameters, together with performing Kani in dimension 4 with relatively large primes. For example, the number of field operations needed to perform the **EvalKani** in **SILBE.Dec** is in the order of $7^5\lambda^5\log(\lambda)^4$, which is, for $\lambda = 128$, around 2^{60}. Nevertheless, we can improve the efficiency of **SILBE.Usk** and **SILBE.KG** as follows:

– We could adapt the **RigorousDoublePath** [12, Algorithm 12] and replace Kani's Lemma by the **KLPT** [25] for key generation and update mechanism. This would require a change of prime p, as we would need for p to be such that $3^\beta N|(p^2-1)$ with $N \geq p^{3/2}$. This is very similar to the primes used in SQISign [15]. Finding such primes would be difficult, which is the reason we chose to present **SILBE.Upk** and **SILBE.KG** using high dimension isogenies.
– We could also speed up the key generation by adapting **RandIsogImages** algorithm of QFESTA [32] to directly construct an isogeny $\phi_A : E_0 \to E_A$, this would nevertheless require additional assumption to ensure that the distribution is computationally indistinguishable from uniform.

6 Further work and conclusion

We thus have constructed SILBE, the first isogeny-based UPKE not relying on group actions. In addition to solving the issues highlighted in [19, section 5], it makes an adequate demonstration of how to combine the multiple isogeny representations to construct new cryptographic schemes.

Further work on SILBE should be directed to improving its efficiency. A pivotal question for exploration is the refinement of computing HD-isogenies as the construction of a higher-dimensional analog akin to $\sqrt{\text{élu}}$ [6] would demonstrably improve SILBE, together with shedding light on novel possibilities to use HD-isogenies in Isogeny Based Cryptography.

References

1. Abou Haidar, C., Libert, B., Passelègue, A.: Updatable public key encryption from dcr: Efficient constructions with stronger security. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 11–22 (2022)

2. Abou Haidar, C., Passelègue, A., Stehlé, D.: Efficient updatable public-key encryption from lattices. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 342–373. Springer (2023)
3. Asano, K., Watanabe, Y.: Updatable public key encryption with strong cca security: Security analysis and efficient generic construction. Cryptology ePrint Archive, Paper 2023/976 (2023), https://eprint.iacr.org/2023/976, https://eprint.iacr.org/2023/976
4. Basso, A., Fouotsa, T.B.: New sidh countermeasures forÂ aÂ more efficient key exchange. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 208–233. Springer Nature Singapore, Singapore (2023)
5. Basso, A., Maino, L., Pope, G.: Festa: Fast encryption from a supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 98–126. Springer Nature Singapore, Singapore (2023)
6. Bernstein, D.J., De Feo, L., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. Open Book Series **4**(1), 39–55 (2020)
7. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key homomorphic prfs and their applications. In: Annual Cryptology Conference. pp. 410–428. Springer (2013)
8. Castryck, W., Decru, T.: An efficient key recovery attack on sidh. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 423–447. Springer (2023)
9. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: Csidh: an efficient post-quantum commutative group action. In: Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24. pp. 395–427. Springer (2018)
10. Castryck, W., Vercauteren, F.: A polynomial time attack on instances of m-sidh and festa. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 127–156. Springer (2023)
11. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. Journal of CRYPTOLOGY **22**(1), 93–113 (2009)
12. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: Sqisignhd: New dimensions in cryptography. Cryptology ePrint Archive, Paper 2023/436 (2023), https://eprint.iacr.org/2023/436, https://eprint.iacr.org/2023/436
13. De Feo, L.: Mathematics of isogeny based cryptography. arXiv preprint arXiv:1711.04062 (2017)
14. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Journal of Mathematical Cryptology **8**(3), 209–247 (2014)
15. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: Sqisign: compact post-quantum signatures from quaternions and isogenies. In: Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26. pp. 64–93. Springer (2020)
16. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over f_p f p. Designs, Codes and Cryptography **78**, 425–440 (2016)
17. Deuring, M.: Die typen der multiplikatorenringe elliptischer funktionenkörper. In: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg. vol. 14, pp. 197–272. Springer Berlin/Heidelberg (1941)

18. Dodis, Y., Karthikeyan, H., Wichs, D.: Updatable public key encryption in the standard model. In: Theory of Cryptography: 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8–11, 2021, Proceedings, Part III 19. pp. 254–285. Springer (2021)
19. Eaton, E., Jao, D., Komlo, C., Mokrani, Y.: Towards post-quantum key-updatable public-key encryption via supersingular isogenies. In: International Conference on Selected Areas in Cryptography. pp. 461–482. Springer (2021)
20. Fouotsa, T.B.: Sidh with masked torsion point images. Cryptology ePrint Archive, Paper 2022/1054 (2022), https://eprint.iacr.org/2022/1054, https://eprint.iacr.org/2022/1054
21. Fouotsa, T.B., Moriya, T., Petit, C.: M-sidh and md-sidh: countering sidh attacks by masking information. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 282–309. Springer (2023)
22. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4. pp. 19–34. Springer (2011)
23. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited (2018)
24. Kani, E.: The number of curves of genus two with elliptic differentials. (1997)
25. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion-isogeny path problem. LMS Journal of Computation and Mathematics **17**(A), 418–432 (2014)
26. Leroux, A.: Quaternion Algebra and isogeny-based cryptography. Ph.D. thesis, Ecole doctorale de l'Institut Polytechnique de Paris (2022)
27. Leroux, A.: Verifiable random function from the deuring correspondence and higher dimensional isogenies. Cryptology ePrint Archive, Paper 2023/1251 (2023), https://eprint.iacr.org/2023/1251, https://eprint.iacr.org/2023/1251
28. Leroux, A., Roméas, M.: Updatable encryption from group actions. Cryptology ePrint Archive (2022)
29. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on sidh. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 448–471. Springer (2023)
30. Milne, J.S.: Abelian varieties. Arithmetic geometry pp. 103–150 (1986)
31. Moriya, T.: Is-cube: An isogeny-based compact kem using a boxed sidh diagram. Cryptology ePrint Archive, Paper 2023/1506 (2023), https://eprint.iacr.org/2023/1506, https://eprint.iacr.org/2023/1506
32. Nakagawa, K., Onuki, H.: Qfesta: Efficient algorithms and parameters for festa using quaternion algebras. Cryptology ePrint Archive, Paper 2023/1468 (2023), https://eprint.iacr.org/2023/1468, https://eprint.iacr.org/2023/1468
33. Robert, D.: Evaluating isogenies in polylogarithmic time (2022)
34. Robert, D.: Breaking sidh in polynomial time. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 472–503. Springer (2023)
35. de Saint Guilhem, C.D., Kutas, P., Petit, C., Silva, J.: Séta: Supersingular encryption from torsion attacks. IACR Cryptol. ePrint Arch. **2019**, 1291 (2019)
36. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009)
37. Vélu, J.: Isogénies entre courbes elliptiques. Comptes-Rendus de l'Académie des Sciences **273**, 238–241 (1971)