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 generalised lollipop attack of Castryck and Vercauteren on the M-SIDH Key exchange by Fouotsa, Moriya and Petit. Doing so, we can in fact make SILBE a post-quantum secure Updatable Public Key Encryption scheme (UPKE). SILBE is in fact the first isogeny-based UPKE which is not based on group actions. Hence, SILBE overcomes the limitations highlighted by Eaton, Jao, Komlo and Mokrani at SAC'21 regarding the design of an SIDH-style UPKE. This is possible by leveraging both the Deuring Correspondence and Kani's Lemma, two central concepts in Isogeny-Based Cryptography.

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

1 Introduction

The notion of Updatable Public Key Encryption (UPKE) was initially introduced in [9] 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 UPKE schemes have been proposed based on discrete logarithm [21], DRC [1], LWE [21,2], and on isogenies [24,33].

In the later case, an in-depth exploration of the question was performed in 2020 by Eaton, Jao, Komlo, and Mokrani in [24]. They proposed two designs of isogeny-based UPKE, respectively based on SIDH [27,17] and CSIDH [11]. For the former protocol, the authors suggested that "a viable construction in practice is hindered by existing mathematical limitations" and only proposed a relaxed variant of UPKE which they named "online UPKE". The online UPKE was then instantiated in this 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 [33].

In the meantime, SIDH was shown insecure in [10,34,41] by leveraging the accessible images of torsion points to construct a high-dimensional isogeny using Kani's Lemma [29] and extract the secret isogeny from the high-dimensional

one. This tool is revolutionary and has enables a breadth of new schemes such as SQISignHD [14] and its 2D variants [4,23,38], FESTA and QFESTA [6,37], IS-CUBE [36] or DeuringVRF [32] and spawn many countermeasures such as M-SIDH [25,26] and binSIDH/terSIDH [5]. Proposed by Fouotsa, Moriya and Petit, the M-SIDH countermeasure prevents the attacks of [10,34,41], at the cost of a lesser efficiency.

The idea in this paper is to explore whether Kani's Lemma could be leveraged to construct an UPKE not based on isogeny group action, and therefore overcome the obstacles encountered by Eaton, Jao, Komlo, and Mokrani in [24]. To do so, we strongly rely on two central tools:

- The first is the *generalised lollipop attack*, as proposed in [12] by Castryck and Vercauteren, and especially how it can be used to attack very specific instances of the M-SIDH.
- The second is the *Deuring correspondence*, that links isogenies between supersingular elliptic curves and ideals between maximal orders of quaternion algebras, and enables new computations, as detailed in Leroux's Thesis [31].

Contributions: Our main contribution is to turn the generalised lollipop attack over M-SIDH into not only a PKE but an UPKE, therefore overcoming the mathematical limitations described in [24] when attempting to design a SIDH based UPKE. Our UPKE, named SILBE[1] for Supersingular Isogeny Lollipop Based Encryption, follows the same principle as SETA [18] where Petit's torsion point attacks were used to design a PKE. However, this adaptation is not without its challenges and necessitates numerous novel technical inputs 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 a supersingular isogeny graph. We also generate examples of secure primes 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 the 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 [26] for further details.

In [12], Castryck and Vercauteren showed 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 $\big(E_1, [\alpha]\phi(P), [\alpha]\phi(Q)\big)$. 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

[1] "syllable" in German.

characteristic p is defined over $\mathbb{F}_p$ with probability $\Theta(p^{-1/2})$. 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 [19]. If the latter problem was solved, it would lead to a sub-exponential quantum algorithm for computing isogenies between supersingular elliptic curves [19], 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, one runs the generalised 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_A by performing a very long walk from E_0 (E_0 set to $j(E_0) = 1728$), 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 sets 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 $\rho \circ \phi_A : E_0 \to E'_A$ 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.

Using this method, we therefore construct an UPKE which we prove to be OW-PCA-U secure, and we rely on the efficient transform detailed in [3], more specifically on [3, Theorem 4], to obtain an IND-CCA-U UPKE. This leads to the very first secure isogeny-based UPKE which is not based on isogeny group actions. In fact, as mentioned earlier, Eaton, Jao, Komlo and Mokrani [24] studied the design of isogeny-based UPKEs, but their non group action based construction had several limitations[2] which are overcome by SILBE.

Outline: The remainder of this paper is organised as follows. In Section 2, we give a quick recall of UPKE, M-SIDH and of the standard algorithms that we use to define SILBE. In Section 3, we detail how we construct the PKE part

[2] These limitations were due to the fact that their secret key update procedure uses the KLPT [30] algorithm and its output is too long to serve as an SIDH secret key.

of SILBE. In Section 4 we explain how we build the key update mechanism of SILBE and provide security arguments for the resulting UPKE. Finally, in Section 5, we discuss how we generate public parameters and discuss SILBE's efficiency.

2 Preliminaries

Throughout this paper, λ denotes the security parameter. We say $f : \mathbb{R} \to \mathbb{R}$ is a negligible function and we denote this by denote $f(x) \leq \mathsf{negl}(x)$ 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 its running time is polynomial in x. Let p be a prime, $\mathbb{F}_p$ be the finite field of characteristic p and $\overline{\mathbb{F}_p}$ be its algebraic closure. Let E and E' be elliptic curves over $\overline{\mathbb{F}_p}$.

2.1 Supersingular isogenies and more

We provide a concise recapitulation of the field and its main ideas. We recommend referring to De Feo's notes [16] and Silverman's book [43] for a general understanding of elliptic curves and isogenies. For insights into the Deuring Correspondence, Leroux's thesis [31] is an excellent resource, while [41] provides invaluable details on Kani's Lemma.

Generalities. An isogeny $\phi : E \to E'$ is 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 isogenies is the product of the degree of the isogenies appearing in the composition. We will consider isogenies up to isomorphism, where 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 (up to isomorphism) *dual isogeny* $\widehat{\phi} : E' \to E$ such that $\phi \circ \widehat{\phi} = [\deg(\phi)]$ on E' and $\widehat{\phi} \circ \phi = [\deg(\phi)]$ on E, with $[n]$ being the scalar multiplication map by $n \in \mathbb{Z}$. The *n-torsion group*, denoted by $E[n]$ is the kernel of the scalar multiplication by n, $E[n] = \ker([n])$. We have that $E[n] \cong \mathbb{Z}_n^2$ when n is co-prime to p. 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 $\phi : E \to E'$, $\ker(\phi) \subset E[\deg(\phi)]$. Let $\phi : E \to F$ and $\psi : E \to F'$ be two isogenies of co-prime degree. The *pushforward* of ψ by ϕ is the isogeny $\phi_*\psi : F \to E'$ whose kernel is 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 an imaginary quadratic field. Otherwise, they are *supersingular* and $\mathrm{End}(E)$ is isomorphic to a maximal order of $\mathbf{B}_{p,\infty}$ the (unique up to isomorphism) 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 on p, and $\mathbf{ij} = -\mathbf{ji}$. 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 corresponds to the maximal order $\mathcal{O}_0 = \mathbb{Z}+\mathbf{i}\mathbb{Z}+\frac{\mathbf{i}+\mathbf{j}}{2}\mathbb{Z}+\frac{1+\mathbf{ij}}{2}\mathbb{Z}$ where $\mathbf{i}$ stands for the endomorphism $(x, y) \to (-x, \sqrt{-1}y)$ and $\mathbf{j}$ stands for the Frobenius endomorphism π.

Supersingularity is an important property: it is preserved by isogenies, all supersingular curves are defined in $\mathbb{F}_{p^2}$ and connected to each other by $\mathbb{F}_{p^2}$-rational isogenies. Importantly, Deuring proved in [20] 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 $\mathcal{O}_R(I)$ being 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 $\phi_I : E \to F$ whose kernel is 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: Finally, a pivotal advancement in the realm of isogenies is Kani's Lemma [29], particularly notable for its role in dismantling SIDH, as detailed in [10,34,41], where it was used to embed isogenies between elliptic curves into higher dimensional isogenies. In this paper, we solely focus on principally polarized abelian varieties and therefore omit the notion of polarization. We refer the interested reader to Milne's book [35]. The only exception is that we denote

the dual of a higher dimensional isogeny ϕ as $\tilde{\phi}$, its polarized dual. We present Kani's Lemma as described in [41, 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 $g' \circ f = f' \circ g$, with $\deg(f) = \deg(f')$ and $\deg(g) = \deg(g')$ coprime. 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 $\deg(F) = \deg(f) + \deg(g)$, $\ker(F) = \left\{ \big(f(P), -g(P)\big) \middle| \ P \in A[D] \right\}$ and $\ker(\widetilde{F}) = \left\{ \big(\tilde{f}(P), g'(P)\big) \middle| \ P \in B[D] \right\}$.*

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

$$\begin{array}{ccccc} & & V & & \\ & \overset{F_1}{\nearrow} & & \overset{\widetilde{F_2}}{\nwarrow} & \\ B \times A' & & \xrightarrow{\quad F \quad} & & A \times B' \end{array}$$

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

2.2 UPKE

We base our definition of UPKE on the notion of symmetric UPKE from [24].

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

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

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

$$\mathbb{P}\left[\mathsf{Dec}\big(\mathsf{sk_i}, \mathsf{Enc}(\mathsf{pk_i}, \mathsf{m})\big) = \mathsf{m} \ \middle| \ \begin{array}{c} (\mathsf{sk}_0, \mathsf{pk}_0) \xleftarrow{\$} \mathsf{KG}(\mathsf{pp}), \mu_\mathsf{i} \xleftarrow{\$} \mathsf{UG}(\mathsf{pp}), \\ \big(\mathsf{sk_i}, \mathsf{pk_i}\big) \xleftarrow{\$} \Big(\mathsf{Usk}(\mathsf{sk}_{\mathsf{i}-1}, \mu_\mathsf{i}), \mathsf{Upk}(\mathsf{pk}_{\mathsf{i}-1}, \mu_\mathsf{i})\Big) \end{array} \right] = 1$$

We make a slight abuse of notation, as all algorithms also take pp as input, but this choice is made to avoid too heavy notations. The desirable security notion an UPKE should achieve are *Forward Security* and *Post-Compromise Security*. The first notion requires that if the adversary learns 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 requires that the adversary is not able to retrieve sk_j for $j > i$ without knowing the update values $\mu_{i+1}, \cdots, \mu_j$.

More formally, the security of an UPKE is captured by security games that use the following oracles and lists. We denote by $\mathsf{Oracles}$ the set of oracles given below.

- $\mathsf{Upd\_list}$ and $\mathsf{Cor\_list}$ are two lists that respectively store the updates specifically 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 returns $\mathsf{pk_{i+1}}$.
- $\mathsf{Given\_Upd}$: 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* receives an index j and returns 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 as inputs. It returns true if the ciphertext is a valid encryption of the plaintext, and false if not.

We use these oracles to define the security notion of *One-Wayness under Plaintext Checking Attack with Updatability* (OW-PCA-U). Here, the adversaries have to decrypt a challenge ciphertext for a chosen uncorrupted key. An UPKE is *OW-PCA-U secure* if for any given $(\mathcal{A}_1, \mathcal{A}_2)$ $\mathsf{poly}(\lambda)$ adversaries, we have that

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

where $\mathcal{G}^{\mathsf{OW\text{-}PCA\text{-}U}}$ is the cryptographic game described in Figure 1.

Similarly to OW-PCA-U security, we also have the notion of *INDistinguishability under Chosen Plaintext Attack with Updatability* (IND-CPA-U) and *INDistinguishability under Chosen Ciphertext Attack with Updatability* (IND-CCA-U). In both cases, both adversaries have access to the $\mathsf{Fresh\_Upd}$, $\mathsf{Given\_Upd}$ and $\mathsf{Corrupt}$ oracles, denoted together as $\mathsf{Oracles}$. Additionally, in IND-CCA-U, both adversaries have access to an additional oracle, $\mathcal{O}_{Dec}$ that decrypts ciphertexts ct given by the adversary.

An UPKE is IND-CPA/CCA-U secure if for any given $\mathsf{poly}(\lambda)$ adversaries $(\mathcal{A}_1, \mathcal{A}_2)$, we have that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA/CCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2) = \left|\mathbb{P}\left[\mathcal{G}_1^{\mathsf{IND\text{-}CPA/CCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2) = 1\right] - \mathbb{P}\left[\mathcal{G}_0^{\mathsf{IND\text{-}CPA/CCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where $\mathcal{G}^{\mathsf{IND\text{-}CPA/CCA\text{-}U}}$ is the cryptographic game described in Figure 2.

2.3 Used Algorithms

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

- **KernelToIsogeny**: Takes as input E and $K \in E[d]$ where E is a supersingular curve, and returns ϕ, the isogeny of degree d whose kernel is generated by K together with its codomain E'. To do so, it factorises ϕ as a composition of prime degree isogenies and uses Vélu's Formulas [44] to compute each of these prime degree isogenies. To be efficient, this requires for d to be smooth.

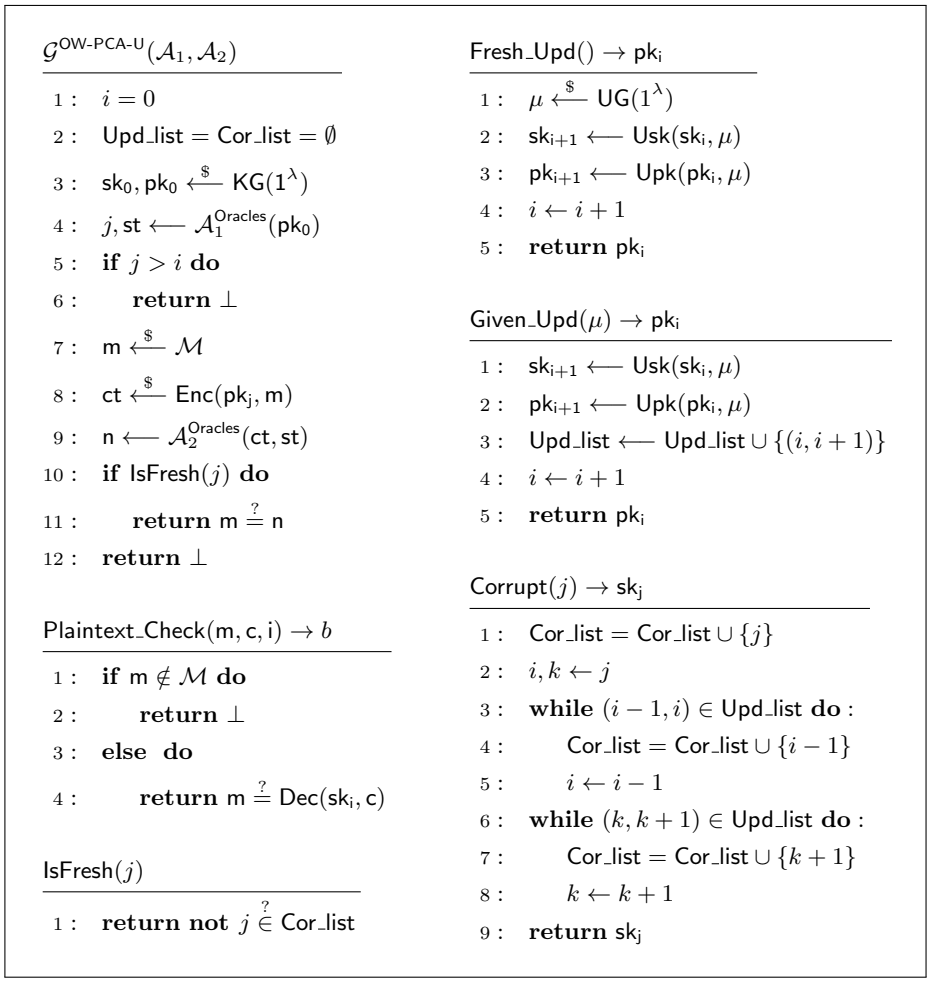


Fig. 1. OW-PCA-U Game

- **CanonicalTorsionBasis**: Takes as input E a supersingular curve and N an integer such that $N|(p^2-1)$ and returns $\langle P, Q\rangle = E[N]$. To do so, one can, for example, sample points at random in $E(\mathbb{F}_{p^2})$ or its quadratic twist and multiplies them by the right co-factor, although more efficient method such as [45] exists. To ensure that this method is deterministic, the sampling is performed deterministically using the Elligator algorithm [8].
- **PushEndRing** [14, Algorithm 8]: Takes as input $\mathfrak{O}_E$ an evaluation basis of $\mathrm{End}(E)$, $\varphi : E \to F$ 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 $\mathrm{End}(F)$. An evaluation basis [14, Definition A.4.1] consists in an isomorphism between

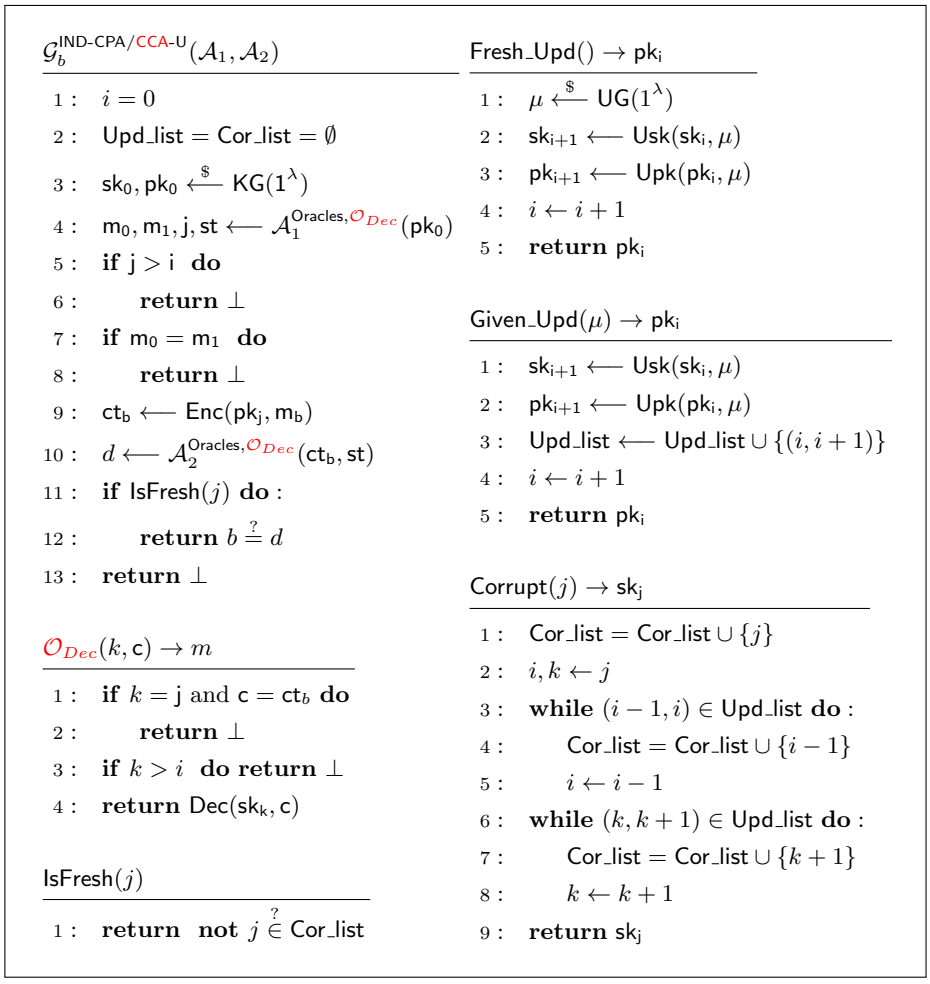


Fig. 2. IND-CPA/CCA-U Game

the endomorphism ring and a maximal order such that every element of the basis is efficiently computable [14, Definition 1.1.1].

- **KernelToIdeal** [14, Algorithm 9]: Takes as input $\mathfrak{O}_E$ a N-evaluation basis of $\mathrm{End}(E)$ and K a generator of the kernel of an isogeny ϕ of *smooth* degree d co-prime to N and return I_ϕ.
- **EvalTorsion** [14, Algorithm 11]: Takes as input $\mathfrak{O}_F$ an evaluation basis of $\mathrm{End}(F)$, $\rho_1 : F \to E$ 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 co-prime to d_1 and d_2. It outputs $\phi_J(P)$, with P any point whose order is co-prime to $d_1 d_2$.

- **RandomEquivalentIdeal** [31, Algorithm 6]: 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 "relatively small" prime, meaning that $n(J) \in [\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$ with extremely high probability, as shown by [31, Lemma 3.2.3 & Lemma 3.2.4].
- **ConstructKani** [40]: 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 depending on $N_1 N_2 - d$. F is in fact the higher dimensional isogeny induced by the following (Kani) diagram:

$$\begin{array}{ccc} E^g & \xrightarrow{\phi^g} & F^g \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\alpha} \\ E^g & \xrightarrow{\phi^g} & F^g \end{array}$$

with $E^g = \overbrace{E \times \cdots \times E}^{g \text{ times}}$, $\phi^g = \big(\overbrace{\phi, \cdots, \phi}^{g \text{ times}}\big)$ and with α an endomorphism of E^g of dimension $1, 2$ or 4 depending on $N_1 N_2 - d$. We also denote by **EvalKani** the algorithm that uses this higher dimensional isogeny to evaluate $\phi(R)$ for any $R \in E$.

2.4 M-SIDH

Following the breaking of SIDH in [10,34,41], some countermeasures were proposed among which Masked SIDH or M-SIDH [25,26]. The central idea comes from the fact that masking the torsion point images in SIDH still enables to compute the pushforwards while protecting against **EvalKani**, as the received torsion points describe the isogeny $[\mathsf{m}]\phi$ whose degree is greater than N. Nevertheless, given $[\mathsf{m}]\phi\binom{P}{Q}$ with P, Q a basis of $E[A]$, we can retrieve $\mathsf{m}^2 \mod A$. 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 A to be such that $\mathbb{Z}_A$ has exponentially 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 M-SIDH that we now describe as presented in [26]. The M-SIDH is given in Figure 3 and its public parameters are as follows:

- $p = ABf - 1$ a prime number such that with $A = \prod_{i=1}^{n_A} p_i$ and $B = \prod_{j=1}^{n_B} q_j$ co-prime 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 can efficiently sample at random over the set $\mu_2(A) = \big\{x \in \mathbb{Z}_A \big| x^2 = 1\big\}$ and $\mu_2(B)$.

It was shown in [26] that the key security of M-SIDH reduces, mutatis mutandis, to the following problem:

M-SIDH

Alice(pp)	**Bob**(pp)
$s_A \leftarrow_\$ \mathbb{Z}_A, \alpha \leftarrow_\$ \mu_2(B)$	$s_B \leftarrow_\$ \mathbb{Z}_B, \beta \leftarrow_\$ \mu_2(A)$
$R_A \leftarrow P_A + [s_A]Q_A$	$R_B \leftarrow P_B + [s_B]Q_B$
$\phi_A, E_A \leftarrow \mathbf{KernelToIsogeny}(E, R_A)$	$\phi_B, E_B \leftarrow \mathbf{KernelToIsogeny}(E, R_B)$
$S_A \leftarrow [\alpha]\phi_A(P_B)$	$S_B \leftarrow [\beta]\phi_B(P_A)$
$T_A \leftarrow [\alpha]\phi_A(Q_B)$	$T_B \leftarrow [\beta]\phi_B(Q_A)$
$\xrightarrow{E_A, S_A, T_A}$	
$\xleftarrow{E_B, S_B, T_B}$	
$U_A \leftarrow S_B + [s_A]T_B$	$U_B \leftarrow S_A + [s_B]T_A$
$\psi_A, E_K \leftarrow \mathbf{KernelToIsogeny}(E_B, U_A)$	$\psi_B, E_K \leftarrow \mathbf{KernelToIsogeny}(E_A, U_B)$
$K \leftarrow \mathsf{KDF}(j(E_K))$	$K \leftarrow \mathsf{KDF}(j(E_K))$

Fig. 3. M-SIDH protocol

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 $N = \prod_{i=1}^{n} p_i$ co-prime to d and let $m \in \mu_2(N)$ be a random element. Given $P, Q, [\mathsf{m}]\phi(P), [\mathsf{m}]\phi(Q)$, compute ϕ.

Importantly, it is not sufficient to ask for n_A and n_B to be around λ, as following [26, 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 $A_t = \prod_{i=t}^{n_A} p_i$, $A_t \geq \sqrt{B} \Rightarrow n_A - t \geq \lambda$ and similarly for B. This induces that, in the case of M-SIDH, we need around $n_A + n_B \simeq 4.5\lambda$.

2.5 Generalised lollipop attack

Recently, Castryck and Vercauteren proposed in [12] a new form of attack, named the *generalised lollipop attack*. It can be used to attack very specific instances of FESTA [6] and M-SIDH. In the latter, it applies when the domain[3] of the masked isogeny ϕ is defined over $\mathbb{F}_p$. When this is the case, its constructs a new unmasked isogeny ψ, uses **EvalKani** over ψ to retrieve $\ker(\psi)$ and extracts $\ker(\widehat{\phi})$ from $\ker(\psi)$.

[3] Or co-domain, using duality.

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 be the masked images of those points, i.e. $\binom{S}{T} = [m]\phi\binom{P}{Q}$. Since E_0 is defined over $\mathbb{F}_p$, we have that the Frobenius π is an endomorphism of E_0. We consider the following diagram, where we denote by $\phi^{(p)} : E_0 \to E^{(p)}$ the pushforward $\pi_*\phi$ of ϕ through the Frobenius π. We set $\psi = \phi^{(p)} \circ \widehat{\phi}$ and use the following lemma.

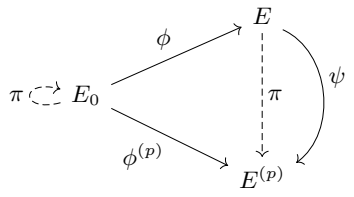


Fig. 4. Diagram of the generalised lollipop over M-SIDH

Lemma 2. *[12, Lemma 3]: Using the above notations, let $\mathbf{M}_{\widehat{\pi}}$ be the matrix such that $\widehat{\pi}\binom{P}{Q} = \mathbf{M}_{\widehat{\pi}}\binom{P}{Q}$. 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 know the evaluation of ψ over $E[N]$ and because $\deg(\psi) = d^2 \leq N^2$, we can use **EvalKani** over ψ to evaluate ψ over any points and in particular over $E[d]$. We can then extract[4] $\ker(\widehat{\phi})$ from $\ker(\psi)[d]$ as detailed in [12, Section 3.2].

3 Constructing a PKE from the generalised lollipop attack

The core concept behind SILBE is to leverage the generalised lollipop attack over M-SIDH as a deciphering mechanism, akin to how the original lollipop attack was employed in designing SETA [18]. This endeavor will make usage of all the different isogeny representations. SILBE is in fact related to [12, 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 $E_0[N]$, U_0, V_0 a basis of $E_0[3^\beta]$ and the matrix $\mathbf{M}_\pi$ that represents the action of π over the basis (P_0, Q_0).

[4] Modulo some minor technical details that are easily manageable.

- KG: Alice computes a long isogeny between E_0 and E_A. Using **KernelToIdeal** and **EvalKani**, it retrieves the representing ideal I and uses 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 a random isogeny $\phi_B : E_A \to E_B$. It then sends the masked images though ϕ_B of a canonical basis of $E_A[N]$, where the mask is the message m.
- Dec: From its knowledge of ϕ_A, Alice uses the generalised lollipop attack over $\phi_B \circ \phi_A$ to retrieve $\ker(\widehat{\phi_B})$. It retrieves m using discrete logarithm computations in $E_B[N]$.

The public parameters of SILBE are constructed using the **SILBE.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 $X, Y \in E[N]$ computes the matrix $\mathbf{M}$ such that $\binom{X}{Y} = \mathbf{M}\binom{P}{Q}$. We discuss more thoroughly how we construct p in Section 5. We also denote by $\mathfrak{O}_0$ the standard efficient evaluation basis of $\mathrm{End}(E_0)$, that is detailed in Section 2.1, with E_0 the curve with j-invariant 1728 defined over $\mathbb{F}_p$.

Algorithm 1 SILBE.Setup

Input: 1^λ

Output: $\mathsf{pp} = \big(p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$ with $p = 3^\beta N f + 1$ 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 co-prime 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: $\mathsf{pp} \leftarrow \big(p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$.
7: **return** pp

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 making use of the following proposition.

Proposition 1. *[14, 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 $\tilde{O}(p^{-\epsilon/2})$ to the uniform distribution in the supersingular isogeny graph, provided that $h \geq (1+\epsilon)\log_\ell(p)$.*

By constructing a path made of t 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 $\tilde{O}(p^{-1/2})$ statistically close from the uniform distribution, which implies that it is computationally indistinguishable from the uniform distribution. 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:

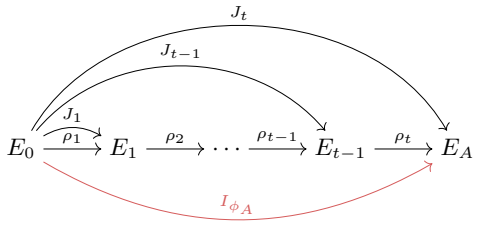
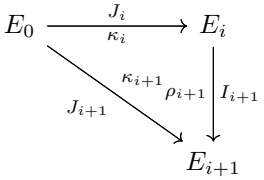


Fig. 5. Diagram of the Key Generation of SILBE

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 co-prime to 3. Furthermore, assume knowledge of $\mathfrak{O}_{E_i}$ a T-evaluation basis[5] 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 $\mathfrak{O}_{E_i}$ we can find I_{i+1} via the **KernelToIdeal**.
3. As 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 $n(J_{i+1}) \in [\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$ 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.



4. 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 can then construct a high dimension representation of κ_{i+1}.

[5] A T-evaluation basis is an evaluation basis that can only evaluate points of order co-prime to T.

5. Using **ConstructKani** over $(P_0, Q_0, \kappa_{i+1}(P_0), \kappa_{i+1}(Q_0))$ in dimension 4, we construct 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 $\mathrm{End}(E_A)$. The dual of this process was independently described by Nakagawa and Onuki [39], who use it to design a new ideal to isogeny algorithm for SQISign.

As a last step, essential for the decryption part of SILBE, we use once again **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 the generalised lollipop is also 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. Note that we need to construct ρ_i in such a way that our walk does not backtrack. To do so, we use U_i, V_i a canonical 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 its kernel foes not intersect that of $\widehat{\rho_i}$.

3.2 Encryption & Decryption

The underlying architecture behind the PKE part of SILBE is given in Figure 6.

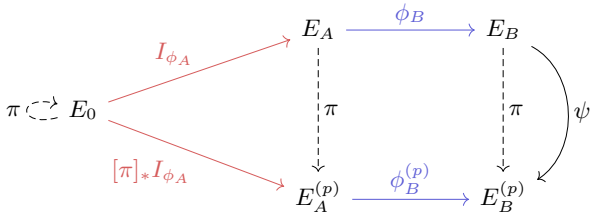


Fig. 6. Diagram of the encryption/decryption of SILBE, Alice in red and Bob in blue

Algorithm 2 SILBE.KG

Input: $\mathsf{pp} = \big(p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$
Output: pk, 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: $E_i, \rho_i \longleftarrow \mathbf{KernelToIsogeny}\big(E_{i-1}, (U_{i-1} + [\eta_i]V_{i-1})\big)$ ▷ In pp if $i = 1$.
5: $I_i \longleftarrow \mathbf{KernelToIdeal}\big(\mathfrak{O}_{E_{i-1}}, (U_{i-1} + [\eta_i]V_{i-1})\big)$
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: $J_i \longleftarrow \mathbf{RandomEquivalentIdeal}(J_{i-1}I_i)$
8: **if** $n(J_i) = n(J_{i-1})$ **or** $\widetilde{N}^2 - n(J_i) \neq 1 \mod 4$ **or** is not prime **do**
9: go back to line 7.
10: $S_i, T_i \longleftarrow \mathbf{EvalTorsion}(\mathfrak{O}_0, \rho_i \circ \kappa_{i-1}, J_{i-1}I_i, id, J_i, \{P_0, Q_0\})$
11: $F_i \longleftarrow \mathbf{ConstructKani}\big(n(J_i), \widetilde{N}, \widetilde{N}, (P_0, Q_0, S_i, T_i)\big)$
12: $\mathfrak{O}_{E_i} \longleftarrow \mathbf{PushEndRing}(\mathfrak{O}_0, \kappa_i, J_i)$ ▷ $\kappa_i(-) \leftarrow F_i(0, 0, -, 0)_3$
13: $I_{\phi_A} \longleftarrow \mathbf{RandomEquivalentIdeal}\big(J_t\big)$
14: **if** $N' - n(I_{\phi_A})^2 3^{2\beta} \neq 1 \mod 4$ **or** is not prime **do** go back to line 13.
15: $K, L \longleftarrow \mathbf{EvalTorsion}(\mathcal{O}_0, \rho_t \circ \cdots \circ \rho_1, I_1 \cdots I_t, 1, I_{\phi_A}, P_0, Q_0)$
16: $\mathbf{M}_{\phi_\mathbf{A}} \longleftarrow \mathbf{EvalImageMatrix}(E_t, N, P_t, Q_t, K, L)$
17: $\mathsf{pk} \longleftarrow \big(E_t = E_A\big)$
18: $\mathsf{sk} \longleftarrow \big(\mathfrak{O}_{E_t}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}\big)$
19: **return** pk, sk.

Encryption As explained earlier, the message space of SILBE is $\mu_2(N) = \big\{x \in \mathbb{Z}_N \big| x^2 = 1\big\}$. As $N = \prod_{i=1}^n p_i$, we have that $|\mu_2(N)| = 2^n$ and we can construct an efficient mapping between $\{0,1\}^n$ and $\mu_2(N)$ using the Chinese remainder theorem. To encrypt m, Bob computes a random isogeny $\phi_B : E_A \to E_B$ of degree 3^β. Then, similarly to M-SIDH, it computes the images of a canonical N-torsion basis (P_A, Q_A) through this isogeny and masks those images using the message m. The ciphertext is therefore $\mathsf{ct} = (E_B, R_1, R_2)$ where $R_1 = [\mathsf{m}]\phi_B(P_A)$ and $R_2 = [\mathsf{m}]\phi_B(Q_A)$.

Algorithm 3 SILBE.Enc

Input: $\mathsf{pp} = \big(p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$, $\mathsf{pk} = E_A$ and a message $\mathsf{m} \in \mu_2(N)$
Output: $\mathsf{ct} = (E_B, R_1, R_2)$ with $R_1, R_2 \in E_B[N]$.

1: $P_A, Q_A \longleftarrow \mathbf{CanonicalTorsionBasis}(E_A, N)$
2: $U_A, V_A \longleftarrow \mathbf{CanonicalTorsionBasis}(E_A, 3^\beta)$
3: Sample $r_B \in_\$ \mathbb{Z}_{3^\beta}$
4: $E_B, \phi_B \longleftarrow \mathbf{KernelToIsogeny}\big(E_A, (U_A + [r_B]V_A)\big)$
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

Decryption As previously stated, we use the generalised 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 generalised lollipop attack of [12, Section 4] such that it just computes $\ker(\widehat{\phi_B})$ and not the whole $\ker(\widehat{\phi_B \circ \phi_A})$. This speeds up the decryption. We consider the isogeny $\psi : E_B \longrightarrow E_B^{(p)}$ given by

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

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

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

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. Following [12, Section 3.2], we have 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 in $\mathbb{Z}[\sqrt{-p}]$ and in $\mathbb{Z}[\sqrt{\chi}]$ with $\chi \in \mathrm{End}(E_A)$ the 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 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: $\mathsf{pp} = \big(p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$, $\mathsf{sk} = (\mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}})$ and $\mathsf{ct} = (E_B, \mathsf{R}_1, \mathsf{R}_2)$
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: $\binom{K}{L} \longleftarrow [n(I_{\phi_A})3^\beta]\mathbf{M}_\pi^{-\mathbf{1}} \pi \binom{S}{T}$
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 $G = H$, just take 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 stress that SILBE is not IND-CPA secure. Indeed, to distinguish between an encryption of m_0 and that of 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 $[m_0]R_1 = [m_0^2]\phi_B(P_A) = \phi_B(P_A)$ and $[m_0]R_2 = [m_0^2]\phi_B(Q_A) = \phi_B(Q_A)$. Otherwise, this means that the encrypted message was m_1 with overwhelming probability. This mechanism can be used to know if a ciphertext ct is that of a plaintext m or not. This induces that any adversary 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 the $\mathsf{Plaintext\_Check}$ oracle, 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 $\tilde{O}(p^{-1/2})$ statistically close to the uniform distribution over supersingular curves. Let $\mathcal{A}^{\mathsf{OW-CPA}}$ be any adversary for SILBE. We can construct an algorithm $\mathcal{B}$ that solve Problem 1 over random curves with the same advantage as $\mathcal{A}^{\mathsf{OW-CPA}}$. $\mathcal{B}$ 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 $\mathcal{A}^{\mathsf{OW-CPA}}\big(E, (E', S, T)\big)$ and receive $\mathsf{n} \in \mu_2(N)$.
3. It then computes $[\mathsf{n}]S, [\mathsf{n}]T$ and uses **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, SILBE is OW-PCA secure.

4 Extending this PKE into an UPKE

4.1 Design

The idea behind SILBE's 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 7 and it is performed as follows:

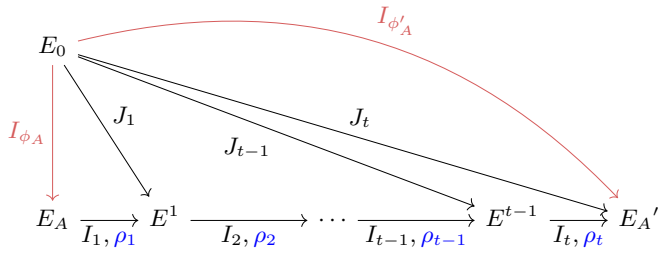


Fig. 7. Diagram of the key update mechanism of SILBE, Alice in red and Bob in blue. Black isogenies are used for the construction of **SILBE.Usk**.

– UG: Generate a random seed $\mu \in \{0,1\}^{4\log(p)}$. The fact that $\mu \in \{0,1\}^{4\log(p)}$ will ensure that the distribution of the updated public key is statistically close to the uniform distribution, as it is the case for a newly generated public key.

– Upk: Use a hash function H over μ to generate a sequence of t 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 E'_A of that walk and set is as the updated public key.

– Usk: Use the hash function H 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 follow the mechanism described in the key generation (Section 3.1) to construct a new relatively short isogeny $\phi'_A : E_0 \to E'_A$, the updated secret key.

Algorithm 5 SILBE.UG

Input: $\mathsf{pp} = \big(p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$
Output: μ an update.
1: Sample $\mu \in_\$ \{0,1\}^{4\log(p)}$
2: **return** μ

Algorithm 6 SILBE.Upk

Input: $\mathsf{pp} = \big(p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$, $\mathsf{pk} = E_A$ and μ.
Output: pk' the updated public key.

$E^0 \leftarrow E_A \qquad U^0, V^0 \longleftarrow$ **CanonicalTorsionBasis**$(E^0, 3^\beta)$
$(\eta_1, \cdots, \eta_t) \leftarrow H(\mu)$ ▷ $\eta_i \in \mathbb{Z}_{3^\beta}$
for $1 \leqslant i \leqslant t$ **do**
  $E^i, \rho_i \longleftarrow$ **KernelToIsogeny**$\big(E^{i-1}, (U^{i-1} + [\eta_i]V^{i-1})\big)$
  Deterministically compute U^i, V^i a basis of $E^i[3^\beta]$ with $\langle V^i \rangle = \rho_i(E^{i-1}[3^\beta])$.
$\mathsf{pk}' \leftarrow E'_A = E^t$
return pk'

Algorithm 7 SILBE.Usk

Input: $\mathsf{pp} = (p, N, \beta, (P_0, Q_0), (U_0, V_0), \mathbf{M}_\pi, t\big)$, $\mathsf{sk} = (\mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}})$ and μ.
Output: sk' the updated secret key.

1: $E^0 \leftarrow E_A \qquad J_0 \leftarrow I_\phi \qquad U^0, V^0 \leftarrow$ **CanonicalTorsionBasis**$(E^0, 3^\beta)$
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** $\widetilde{N}^2 - n(J_i) \neq 1 \mod 4$ **or** is not prime **do**
9: go back to line 7.
10: $S^i, T^i \longleftarrow$ **EvalTorsion**$(\mathfrak{O}_0, \rho_i \circ \kappa_{i-1}, J_{i-1}I_i, 1, J_i, P_0, Q_0)$ ▷ Use $\mathbf{M}_{\phi_\mathbf{A}}$ if $i = 1$
11: $F_i \longleftarrow$ **ConstructKani**$(n(J_i), \widetilde{N}, \widetilde{N}, P_0, Q_0, S^i, T^i)$
12: $\mathfrak{O}_{E_i} \longleftarrow$ **PushEndRing**$(\mathfrak{O}_0, \kappa_i, J_i)$ ▷ $\kappa_i(-) = F_i(0, 0, -, 0)_3$
13: $I_{\phi'_A} \longleftarrow$ **RandomEquivalentIdeal**$\big(J_t\big)$
14: **if** $N' - n(I_{\phi'_A})^2 3^{2\beta} \neq 1 \mod 4$ **or** is not prime **do** go back to line 12.
15: $K, L \longleftarrow$ **EvalTorsion**$(\mathfrak{O}_0, \kappa_t, J_t, 1, I_{\phi'_A}, P_0, Q_0)$
16: $\mathbf{M}_{\phi'_\mathbf{A}} \longleftarrow$ **EvalImageMatrix**(E_t, N, P_t, Q_t, K, L)
17: $\mathsf{sk}' \longleftarrow \big(\mathfrak{O}_{E_t}, I_{\phi'_A}, \mathbf{M}_{\phi'_\mathbf{A}}\big)$
18: **return** sk'

4.2 Security

SILBE is OW-PCA-U secure. This 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 to the uniform distribution and thus, E'_A has the same distribution as a public key E_A returned 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[6]. This leads us to the following proposition.

[6] 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,*

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

In fact, we can see that our update mechanism is very similar to the CGL hash function [13], as the problem of finding μ such that $\mathbf{SILBE.Upk}(E, \mu) = E'$ reduces to the *isogeny walk problem* [16, Problem 3], meaning that our key update mechanism is one-way.

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

Transforming SILBE into an IND-CCA-U UPKE. SILBE is thus an OW-PCA-U UPKE. This is a nice result, but it would be far preferable to have an IND-CPA or an IND-CCA UPKE, a security requirement that we already showed in Section 3.3 SILBE can not reach alone. To remedy this problem, we transform SILBE using an adaptation of the Fujisaki-Okamoto (FO) transform [28] to UPKE. More specifically, we use the transformation detailed in [3, Section 4]. For that transformation, we will need the notion of λ-speadness, as defined in [3, Definition 7], and of One Time Chosen Plaintext Attack Symmetric Key Encryption (OT-CPA SKE), as given in [3, Definition 3].

Theorem 1. *[3, Theorem 4, simplified form] Let Π be an OW-CPA-U UPKE scheme that is λ-spread and let Γ be an OT-CPA SKE scheme. Then, given 4 random oracles, we can derive an UPKE Σ that is IND-CCA-U secure in the ROM.*

Applying this theorem to SILBE, we can derive SILBE*, an IND-CCA-U secure version of SILBE. To so so, we have to show that SILBE is λ-spread, but this is a direct consequence of Proposition 1 and of the fact that $3^\beta \gg 2^\lambda$, as we will show in Section 5.1.

It is essential to highlight that this transformation results in an UPKE where UG and Upk are amalgamated. This amalgamation introduces the possibility for the update token to depend on certain information from the public keys, a change from our initial definition. This unified model of UPKE is well-established in the literature (see, for example, [22,3]). Additionally, it's worth noting that this transformation yields an IND-CCA-CU secure UPKE, a stronger security notion compared to our original IND-CCA-U, which corresponds to an IND-CCA-CR secure UPKE, if we follow [22] naming convention. For a detailed exposition of the construction, please refer to Figure 8 in Appendix A.

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 compute N and β, we proceed as follows.

- Initiate β and N.
- 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 the smallest co-factor f such that $p = 3^{\beta}Nf + 1$ is prime. Using this method, we found the following "SILBE friendly" primes, detailed in Table 1.

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

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

5.2 Efficiency of SILBE

We perform a high level theoretical analysis of the efficiency of SILBE. The efficiency of SILBE mostly depends on the size of the parameters and on the cost of performing Kani's Lemma in dimension 4 over relatively large primes. Using the approximation $\sum_{p \leq x} p^i \simeq \pi(x^{i+1})$, we have that the use of Kani's lemma in SILBE requires around:

- $2^3 3^5 \lambda^5 \log(\lambda)^4$ field operations for either **SILBE.KG** and **SILBE.Usk**.
- $7^5 \lambda^5 \log(\lambda)^4$ field operations for **SILBE.Dec**.

Those values, although perfectly polynomial in λ, are not yet practical. This essentially comes from the size of the parameters, together with performing Kani's Lemma in dimension 4 with relatively large primes. Nevertheless, we could improve the efficiency of some subroutines of SILBE as follows:

- **SILBE.KG**: We could speed up the key generation by adapting the **RandIsogImages** algorithm of QFESTA [37] to construct an isogeny $\phi_A : E_0 \to E_A$ directly. With this, we could efficiently perform **SILBE.KG** using high dimensional isogenies of dimension 2, thus requiring only $7^3\lambda^3\log(\lambda)^2$ field computations.
- **SILBE.Usk**: We can extend the previous idea to also speed up **SILBE.Usk**. More specifically, to efficiently compute the isogenies given by the ideals J_i linking E_0 with the E^i curves, we first compute an endomorphism $\gamma_i \in \mathrm{End}(E_0)$ of norm $n(J_i)\big(\overline{N} - n(J_i)\big) > p$, with $\overline{N} > \sqrt{p}$ a divisor of N. We then use Kani's Lemma once to split this endomorphism and retrieve

an auxiliary isogeny of domain E_0 and degree $\overline{N} - n(J_i)$. This isogeny can then be used to evaluate the isogeny described by J_i by applying Kani's Lemma one more time. Using this method, we can perform **SILBE.Usk** using $2^4 5^3 \lambda^3 \log(\lambda)^2$ field operations.

The potential improvements above would nevertheless require additional assumptions to ensure that the distribution is computationally indistinguishable from uniform.

When it comes to performing **SILBE.Dec** using dimension 2 isogenies, it is far more challenging. This comes from the fact that the isogeny ψ is of degree $3^{2\beta} n(I_{\phi_A})^2 \gg N$. This implies that evaluating ψ requires splitting our high-dimensional isogeny into two parts, since we don't have enough torsion to compute our higher dimensional isogeny in one go. In dimension 4, the isogeny computed is an endomorphism, which facilitates its splitting. In dimension 2, the isogeny computed is not an endomorphism anymore, which makes the splitting method does not extend naturally. It is unclear whether one could split the dimension 2 isogeny (in the context of SILBE) without incurring a huge overhead.

A robust implementation of SILBE necessitates efficient implementations of isogenies of dimension 2 and/or 4 of prime degree in the order of λ. However, to the best of our knowledge, no such comprehensive work exists for either dimensions. Currently, the only efficient implementations are for isogenies of degree 2 [15,14] in dimension 2 and 4 and of degree 3 [42] in dimension 2.

In brief, there some obstacles preventing SILBE from being practically efficient and being implemented today, but with the ongoing effort in improving the computation of high-dimensional isogenies it in very likely that these obstacles are overcome in the nearest future. Eventually, follow-up works will potentially lead to a more efficient scheme. For example, in Section 6, we discuss a possible variant where FESTA is used as the underlying building block and identify some roadblocks in instantiating this variant.

6 Conclusion and further work

We have thus constructed SILBE, the first isogeny-based UPKE not relying on group actions. In addition to solving the issues highlighted in [24, 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 to $\sqrt{\text{élu}}$ [7] would demonstrably improve SILBE, together with shedding light on novel possibilities to use high dimension isogenies in Isogeny Based Cryptography.

One way to gain in efficiency is to make sure that during decryption, the degree of the higher dimension isogeny computed is a power of two. This is not possible in M-SIDH since the order of the points in the public keys need to be

highly composite. A plausible candidate here is FESTA [6] where the torsion points in the public key have order a power of two, hence the higher dimension isogeny computed during decryption will have order a power of two. The obstacle with FESTA is the fact that E_0 being defined over $\mathbb{F}_p$ and the endomorphism ring of E_0 being known to the owner of the secret key are not sufficient for the lollipop attack. In fact, with FESTA, one needs to know a relatively small endomorphism θ of E_0 that fixes one or both cyclic groups generated by points in a particular basis which is dependant on the secret isogeny $\phi_A : E_0 \to E_A$. This can be assured during the key generation, but it is unclear how to assure this during the key update process, since the key update process leads to a brand new public key E'_A, together with its corresponding secret key ϕ'_A. In fact, the endomorphism θ used in the previous secret key is useless, and in some cases, there may not exists any $\theta' \in \mathrm{End}(E)$ that matches the constraints with respect to the new secret key ϕ'_A. We leave further investigations of instantiating SILBE with FESTA for future work.

Acknowledgment

We would like to thank SAC 2024 anonymous reviewers for their valuable and insightful suggestions.

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., Feo, L.D., Dartois, P., Leroux, A., Maino, L., Pope, G., Robert, D., Wesolowski, B.: SQIsign2D-west: The fast, the small, and the safer. Cryptology ePrint Archive, Paper 2024/760 (2024), https://eprint.iacr.org/2024/760, https://eprint.iacr.org/2024/760
5. Basso, A., Fouotsa, T.B.: New SIDH Countermeasures for a More Efficient Key Exchange. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 208–233. Springer Nature Singapore, Singapore (2023)
6. 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)
7. 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)

8. Bernstein, D.J., Hamburg, M., Krasnova, A., Lange, T.: Elligator: elliptic-curve points indistinguishable from uniform random strings. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. pp. 967–980 (2013)
9. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key Homomorphic PRFs and Their Applications. In: Annual Cryptology Conference. pp. 410–428. Springer (2013)
10. 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)
11. 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)
12. 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)
13. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. Journal of CRYPTOLOGY **22**(1), 93–113 (2009)
14. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQISignHD: new dimensions in cryptography. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 3–32. Springer (2024)
15. Dartois, P., Maino, L., Pope, G., Robert, D.: An algorithmic approach to $(2,2)$-isogenies in the theta model and applications to isogeny-based cryptography. Cryptology ePrint Archive, Paper 2023/1747 (2023), https://eprint.iacr.org/2023/1747, https://eprint.iacr.org/2023/1747
16. De Feo, L.: Mathematics of isogeny based cryptography. arXiv preprint arXiv:1711.04062 (2017)
17. 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)
18. De Feo, L., Delpech de Saint Guilhem, C., Fouotsa, T.B., Kutas, P., Leroux, A., Petit, C., Silva, J., Wesolowski, B.: SÉTA: Supersingular encryption from torsion attacks. In: Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27. pp. 249–278. Springer (2021)
19. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Designs, Codes and Cryptography **78**, 425–440 (2016)
20. 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)
21. 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)
22. Dodis, Y., Karthikeyan, H., Wichs, D.: Updatable public key encryption in the standard model. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography. pp. 254–285. Springer International Publishing, Cham (2021)

23. Duparc, M., Fouotsa, T.B.: SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies. Cryptology ePrint Archive, Paper 2024/773 (2024), https://eprint.iacr.org/2024/773, https://eprint.iacr.org/2024/773
24. Eaton, E., Jao, D., Komlo, C., Mokrani, Y.: Towards Post-Quantum Updatable Public-Key Encryption via Supersingular Isogenies. In: International Conference on Selected Areas in Cryptography. pp. 461–482. Springer (2021)
25. 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
26. 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)
27. 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)
28. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part III 38. pp. 96–125. Springer (2018)
29. Kani, E.: The number of curves of genus two with elliptic differentials. Walter de Gruyter, Berlin/New York Berlin, New York (1997)
30. 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)
31. Leroux, A.: Quaternion Algebra and Isogeny-Based Cryptography. Ph.D. thesis, Ecole doctorale de l'Institut Polytechnique de Paris (2022)
32. 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
33. Leroux, A., Roméas, M.: Updatable encryption from group actions. In: International Conference on Post-Quantum Cryptography. pp. 20–53. Springer (2024)
34. 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)
35. Milne, J.S.: Abelian varieties. Arithmetic Geometry pp. 103–150 (1986)
36. 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
37. 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
38. Nakagawa, K., Onuki, H.: SQIsign2D-east: A new signature scheme using 2-dimensional isogenies. Cryptology ePrint Archive, Paper 2024/771 (2024), https://eprint.iacr.org/2024/771, https://eprint.iacr.org/2024/771
39. Onuki, H., Nakagawa, K.: Ideal-to-isogeny algorithm using 2-dimensional isogenies and its application to SQIsign. Cryptology ePrint Archive, Paper 2024/778 (2024), https://eprint.iacr.org/2024/778, https://eprint.iacr.org/2024/778
40. Robert, D.: Evaluating isogenies in polylogarithmic time. Cryptology ePrint Archive, Paper 2022/1068 (2022), https://eprint.iacr.org/2022/1068, https://eprint.iacr.org/2022/1068

41. Robert, D.: Breaking SIDH in polynomial time. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 472–503. Springer (2023)
42. Santos, M.C.R., Costello, C., Smith, B.: Efficient (3,3)-isogenies on fast kummer surfaces. Cryptology ePrint Archive, Paper 2024/144 (2024), https://eprint.iacr.org/2024/144, https://eprint.iacr.org/2024/144
43. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009)
44. Vélu, J.: Isogénies entre courbes elliptiques. Comptes-Rendus de l'Académie des Sciences **273**, 238–241 (1971)
45. Zanon, G.H., Simplicio, M.A., Pereira, G.C., Doliskani, J., Barreto, P.S.: Faster key compression for isogeny-based cryptosystems. IEEE Transactions on Computers **68**(5), 688–701 (2018)

A IND-CCA-CU FO transform

Theorem 2. *[3, Theorem 4] Let* $\Pi := (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{UG}, \mathsf{Upk}, \mathsf{Usk})$ *be an OW-CPA-U UPKE scheme that is* λ*-spread and* $\Gamma := (\mathsf{SEnc}, \mathsf{SDec})$ *be an OT-CPA SKE scheme with* $\mathcal{M}, \mathcal{R}$ *and* $\mathcal{SK}$ *respectively the spaces of plaintexts of* Π*, randomness of* Π*, and secret keys of* Γ*. Let*

- $\mathsf{G} : \{0,1\}^* \to \mathcal{SK}$
- $\mathsf{H} : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \mathcal{SK}$
- $\widehat{\mathsf{G}} : \{0,1\}^* \to \mathcal{SK}$
- $\widehat{\mathsf{H}} : \{0,1\}^* \times \{0,1\}^* \to \mathcal{M}$

be 4 random oracles. Then we can construct an UPKE $\Sigma := (\mathsf{KG}', \mathsf{Enc}', \mathsf{Dec}', \mathsf{Upk}', \mathsf{Usk}')$ *that is IND-CCA-U secure in the ROM.*

The proof of this theorem is detailed in [3, Section 4.3 & Section 4.4]. The construction of Σ is detailed in Figure 8.

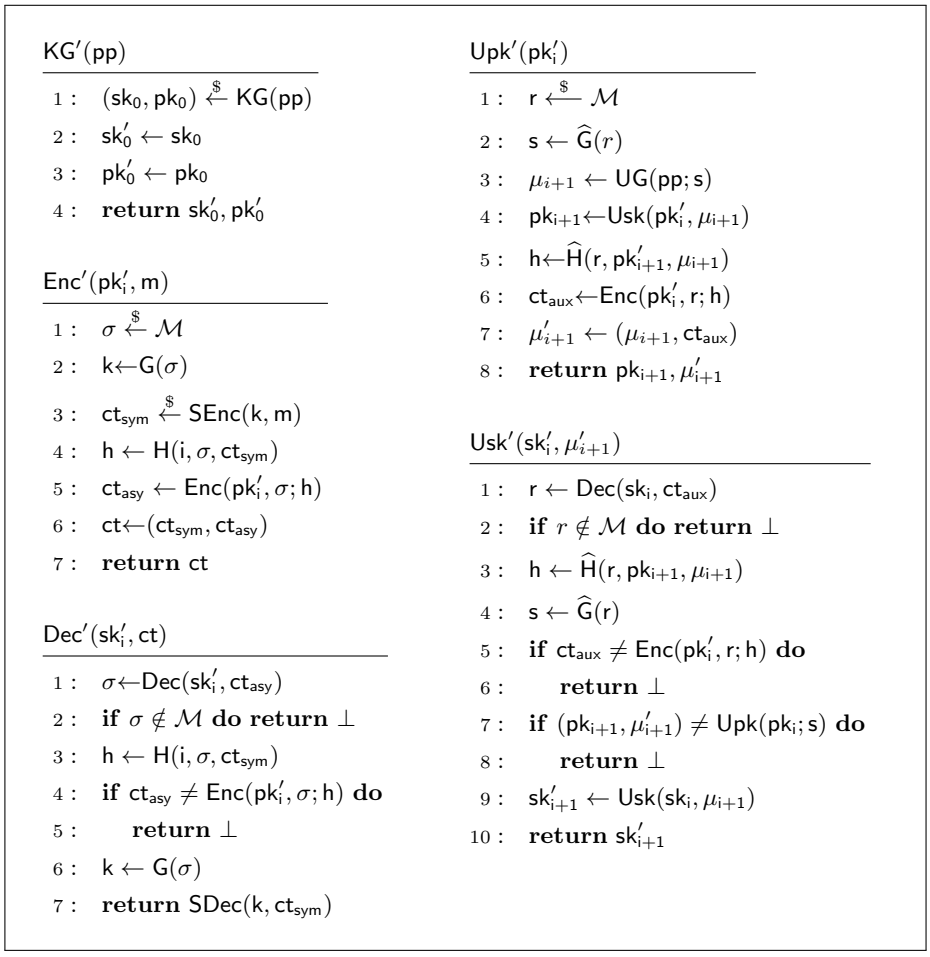


Fig. 8. [3] UPKE-FO-Transform