Verifiable random function from the Deuring correspondence and higher dimensional isogenies

Antonin Leroux,

[1] DGA-MI, Bruz, France antonin.leroux@polytechnique.org
[2] IRMAR, Université de Rennes, France

Abstract. In this paper, we introduce DeuringVUF, a new Verifiable Unpredictable Function (VUF) protocol based on isogenies between supersingular curves.

The most interesting application of this VUF is DeuringVRF a post-quantum Verifiable Random Function (VRF). The main advantage of this new scheme is its compactness, with combined public key and proof size of roughly 450 bytes, which is orders of magnitude smaller than other generic purpose post-quantum VRF constructions. This scheme is also the first post-quantum VRF satisfying unconditional uniqueness. We show that this scheme is practical by providing a first non-optimized C implementation that runs in roughly 18ms for verification and 160ms for evaluation. Up to our knowledge, this is the fastest implementation of a post-quantum VRF.

The function at the heart of our construction is the one that computes the codomain of an isogeny of big prime degree from its kernel. The evaluation can be performed efficiently with the knowledge of the endomorphism ring using a new ideal-to-isogeny algorithm introduced recently by Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert and Wesolowski that uses computation of dimension 2 isogenies between elliptic products to compute effectively the translation through the Deuring correspondence of any ideal. On the other hand, without the knowledge of the endomorphism ring, this computation appears to be hard. The security of our DeuringVUF holds under a new assumption call the one-more isogeny problem (OMIP).

Another application of DeuringVUF is the first hash-and-sign signature based on isogenies in the standard model. While we don't expect the signature in itself to outperform the recent variants of SQIsign, it remains very competitive in both compactness and efficiency while providing a new framework to build isogeny-based signature that could lead to new interesting applications.

We also introduce several new algorithms for the effective Deuring correspondence. In particular, we introduce an algorithm to translate an ideal of norm a big power of a small prime ℓ into the corresponding isogeny of dimension 1 using isogenies between abelian variety of dimension 2 as a tool. This algorithm can be used to improve the SQIsign signature scheme.

1 Introduction

A Verifiable Random Function (VRF) is a way to generate authenticated randomness in a verifiable manner. This notion was introduced in [37] and have found several practical applications in the DNSSEC protocol [25] or in blockchain consensus [7,23,13].

The most widely-used VRF constructions are based on pairings and elliptic curves such as [4] and are not resistant to an attacker that can access a quantum computer. Thus, it is an important problem to devise new schemes that are compact, efficient and resistant to quantum attackers.

Verifiable Unpredictable Functions (VUF) are very closely related to VRF as there is a very simple transform from one to the other in the ROM. Recently, [24] showed that the VRF obtained from this transform have the additional property of being unbiasable, which is often a desirable feature in practice.

In this work, we explore the possibilities offered by isogeny-based cryptography, one of the newest family of post-quantum candidates known for the compactness of its schemes, to build a VUF and thus a VRF. The main tools of isogeny-based cryptography are isogenies, that are maps between abelian varieties. Until very recently, only isogenies between elliptic curves, i.e. varieties of dimension 1, had been really studied. However, isogenies between abelian varieties of higher dimension (namely $2, 4$ and 8) have recently found some surprising applications in the cryptanalysis of the SIDH key-exchange protocol [28]. A series of paper by Castryck and Decru [6], Maino, Martindale, Panny, Pope and Wesolowski [36], and Robert [39] have shown how to use isogenies of higher dimension to break completely SIDH. This breakthrough has produced a small revolution in the field, first by breaking its most famous protocol, and more recently by finding several new constructive applications (see [1,8,11,2] among many examples).

In this article, we follow the example set in [1,11] and explore the combinations of these new techniques with another sub-domain of the field related to the study of the Deuring correspondence, a link between quaternion algebras and isogenies between elliptic curves. As for isogenies of higher dimension, the Deuring correspondence was first explored for its cryptanalitic applications [31,17] before revealing its constructive potential in signature schemes [22,14]. These protocols rely on some complex algorithms to realize effectively the Deuring correspondence: i.e. the translation from isogenies to ideals (their quaternionic counterparts) and vice versa. In particular, we will make good use of a new algorithm introduced in [1] to evaluate efficiently isogeny of arbitrary degree from the corresponding ideal. In this work, we improve the efficiency of some existing algorithms by tackling the case of translating an ideal of norm a big power of a small prime to the corresponding isogeny.

Related Works. There exists several other proposals of quantum-resistant VRF. Lattice-based constructions were the first to appear with [26,42]. These first constructions suffered from huge proof sizes and has been subsequently improved [43,21]. Among those, the recent proposal from [21] appears to be quite practical with a relatively reasonable combined key and proof size of around 20KB. We can also mention [20] that introduces a practical few-times construction.

There are other existing solutions relying on the security of symmetric primitives such as [5] that introduces several construction based on hash functions.

Finally, two proposals based on isogenies have been recently introduced in [32]. The VRF protocols presented in [32] are constructed from isogeny-based group actions and share almost nothing with our new VRF construction apart from the fact that isogenies are involved in both cases. Our construction uses a lot of different techniques and is much more compact. Conceptually, our DeuringVUF schemes is much closer to the recent weak VDF proposal of [16] or the new SQIsign2D-west [1] variant of the SQIsign signature scheme.

Contributions. Our contributions can be summarized in the following manner:

- A new VUF protocol DeuringVUF based on the Deuring correspondence and isogenies between abelian varieties of high dimension. The security of the construction is based on a new hardness assumption that is related to well-studied algorithmic problem of isogeny-based cryptographic.
- A new VRF protocol DeuringVRF (easily derived from DeuringVUF) with unconditional uniqueness, pseudo-randomness and unbiasability in the random oracle model. DeuringVRF is much more compact than all existing post-quantum constructions as exhibited in Table 1. Moreover, DeuringVRF is also unbiasable with no additional cost.
- A C implementation of our proposed scheme based on the code from [1] which is, as far as we know, the most efficient implementation of post-quantum VRF scheme.
- The first hash-and-sign signature scheme based on isogenies in the standard model.
- A new algorithm for the effective Deuring correspondence to translate ideals to their corresponding isogenies using isogenies in dimension 2. This new algorithm relaxes the constraints of previous existing solutions and requires only "SIDH primes" of the form $c2^f3^e - 1$ for the characteristic of the underlying field. These primes are easy to find at any level of security unlike the "SQIsign primes" required by the algorithms used in [14,15]. This new approach appears to be a promising direction to explore in order to improve the efficiency of the SQIsign [14] signature scheme.
- A new algorithm to evaluate an isogeny from its ideal representation with less torsion requirement than existing solutions.

In Table 1, we compare the concrete sizes we obtain for DeuringVRF for the example parameters that we will introduce in Section 4.2 with other existing constructions. We see that our new protocol is much more compact than all existing solutions and also that it is the only one with unconditional uniqueness.

Remark 1. Note that the compactness gap between our isogeny-based VRF and lattice-based VRF is much larger than the gap between SQIsign and lattice-based signatures. This is because the uniqueness feature required by VRF is much harder to get in the lattice setting where things are inherently noisy. This is illustrated by the fact that our VRF is the first one in the post-quantum setting to achieve unconditional uniqueness (as opposed to computational). As such, VRF seems to be one of the most promising application of isogeny-based cryptography as showcased by our new scheme.

	Public Key (bytes)	Proof (bytes)	Unrestricted evaluation	Uniqueness	Assumption	Security level
LB - VRF [20]	3.3K	4.9K	✗	Computational	MSIS/MLWE(Latt.)	128
SL - VRF [5]	48	40K	✓	Computational	LowMC(Hash)	128
LaV [21]	8.81K	10.27K	✓	Computational	MSIS/MLWR(Latt.)	128
CAPYBARA [32]	8.3K	39K	✓	Computational	DDH(Isog.)	< 128
TSUBAKI [32]	5.3K	34K	✓	Computational	sDDH(Isog.)	< 128
DeuringVUF	192	256	✓	Unconditional	$\mathrm{OMIP}_{\mathsf{2dim}}$(Isog.)	128

Table 1: Comparison of the sizes and security properties of several post-quantum VRF schemes with our protocol DeuringVRF for the parameters of Section 4.2.

As far as we know, from the unrestricted primitives mentioned in Table 1, only SL-VRF has been implemented with an evaluation time of 765ms and verification time of 475ms. In comparison, our C implementation achieves 160ms for evaluation and 18ms for verification which makes it the fastest implementation of a post-quantum VRF.

1.1 Technical overview

The high-level idea of our DeuringVUF construction is the following: given a supersingular curve E (the public key), the DeuringVUF function associates the curve E/G to the subgroup G of E. Computing E/G is difficult from the sole knowledge of E and G when the order of G is a big prime, but it can be done efficiently when one knows the endomorphism ring of E (and a few additional information) using the Deuring correspondence.

The main feature of a VUF is the verifiability of the output. In DeuringVUF, the correctness of the result can be proven by embedding the isogeny $E \to E/G$ in a 2-dimensional isogeny using the techniques recently introduced to attack SIDH [6,36,39].

At its heart, our construction exploits the differences between the different known ways of representing a cyclic isogeny.

First, there is the *kernel representation* made of one generator of the kernel. When the kernel is defined over a small field extension, this representation is quite easy to sample from the domain curve and it enables simple verification of the correctness of the computation by evaluating the isogeny on its kernel. However, there is no known efficient algorithm to compute or evaluate an isogeny from the kernel in the generic case. All those properties makes the kernel representation a perfect input to our random function.

Then, there is the *ideal representation* obtained from the Deuring correspondence. This representation is the most powerful one as it allows us to perform all the possible operations efficiently. However, as it also encodes the knowledge of the endomorphism ring of the domain, it essentially contains all the information there is to know about the isogeny, its domain and its codomain. This is why the ideal representation matches exactly the requirements of a secret key/trapdoor.

Finally, there is the 2-*dimensional isogeny representation* (noted $\mathit{2}\mathsf{dim}$ hereafter) introduced recently by Robert [38]. It allows us to evaluate efficiently the isogeny with the help of dimension 2 isogenies without revealing anything on the endomorphism ring. This is ideal for the proof as it provides verifiability when combined with the kernel representation while not leaking anything secret.

Below, we give a more precise description of the various mechanisms and parameters constituting our $\mathsf{DeuringVUF}$ scheme. The notations introduced below are kept throughout the paper.

Parameters. Let p, N be two distinct primes. For a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$, let f be the biggest exponent such that $E[2^f]$ is defined over $\mathbb{F}_{p^2}$. The N-torsion is defined over a $\mathbb{F}_{p^2}$ as well (but on a quadratic twist of E). Finally, let E_0 be some public curve of known endomorphism ring.

Keys. The public keys are made of:

1. a supersingular curve E,
2. a basis $\langle R, S\rangle$ of $E[N]$,

Secret keys are constituted by:

1. an ideal I connecting a fixed maximal order $\mathcal{O}_0$ to $\mathcal{O} \cong \mathrm{End}(E)$ corresponding to some isogeny $\varphi_I : E_0 \to E$.
2. a matrix M keeping track of how the basis R, S was computed.
3. a basis U_0, V_0 of $E_0[2^f]$ and its preimage U, V through the dual of the isogeny $\hat{\varphi}_I$.

Evaluation Mechanism. On input x, the VRF evaluation is as follows: hash x into one element $(r : s) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, compute $R_x = [r]R + [s]S$ and compute $E_x = E/\langle R_x\rangle$. The output is then $v = j(E_x)$. The knowledge of M will enable the efficient computation of the kernel ideal I_x corresponding to the subgroup $\langle R_x\rangle$. Then, the Deuring correspondence can be used to find the curve whose endomorphism ring is isomorphic to the right order of I_x (corresponding to the codomain of an N-isogeny $\varphi_x : E \to E_x$).

The computation of the matrix M and the basis R, S to allow an efficient and correct computation of I_x will be the key of the key generation process.

Proof and Verification Protocol. Proving the correctness of the computation can be done by revealing a representation of the isogeny $\varphi_x : E \to E_x$. If the verifier can check the degree of this isogeny and evaluate it on R_x, then the output must be correct. For that, we propose to use the representation of [38] obtained by embedding φ_x inside a 2^f-isogeny in dimension 2.

The crucial step to compute this high-dimensional representation is the evaluation of some isogenies on well-chosen torsion points. Using the secret key and the ideal I_x, the prover will be able to evaluate these isogenies efficiently using [1, Algorithm 3].

After that, the verification simply consists in checking that the isogeny representation is valid and has kernel R_x. This last part can be done with an $\mathsf{IsogEval}_{\mathit{2}\mathsf{dim}}$ algorithm to evaluate the isogeny from its $\mathit{2}\mathsf{dim}$ representation (see [12]).

Hard Problem and security. The security of our new VRF scheme essentially stems from the problem of computing the codomain of an isogeny from its kernel. The best known algorithm to solve this problem was introduced in [3] and has polynomial complexity in the degree of the isogeny.

Since the pseudo-randomness property of our scheme allows the adversary to evaluate the function on several inputs, the concrete security is based on the $\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}$, a variant of this problem, where the adversary has access to an oracle that computes the codomain and a $\mathit{2}\mathsf{dim}$ isogeny representation on given instances. The goal is then to find the answer for one instance that was not queried to this oracle. This problem has not been used anywhere in cryptography before, but the problem of computing the codomain of an isogeny from its kernel has been studied extensively due to its impact on the efficiency of several schemes in isogeny-based cryptography and has been recently considered for a proposal of weak post-quantum VDF [16].

The formal description of our DeuringVUF scheme can be found in Section 3.1. The concrete protocols include several additional steps to meet the requirements of a cryptographic VUF.

New algorithm for the effective Deuring correspondence. The task of translating an ideal of norm a big power of a small prime ℓ into the corresponding isogeny is at the heart of the SQIsign signature scheme [14]. The algorithm proposed in the original article was improved in [15] and it was used in the version of SQIsign submitted to the NIST.

We propose a new method to solve this problem. The motivation for this new algorithm is to overcome the remaining obstacles of the method from [15] with the new possibilities offered by high-dimensional isogenies. More specifically, the bottleneck in this algorithm is the computation of some endomorphisms. While the best solution in dimension 1 is to require that these endomorphisms have big smooth norm T^2 for some smooth integer T, we can remove this requirement by using embedding the endomorphisms in isogenies of dimension 2. This idea simplifies a lot the choice of parameters by removing the need of so-called "SQIsign primes" (and replacing them by "SIDH primes"), and turns out to be more efficient thanks to the fast formulas to compute chain of $(2,2)$-isogenies in the theta model introduced in [12]. We introduce this new method in Appendix A.

Ackowledgements. We thank the anonymous reviewers for useful remarks on a previous version of this paper. We thank the SQIsign2D-west team for privately sharing their code with us.

The rest of this paper is organized as follows. Section 2 introduces preliminaries on VRFs and the Deuring correspondence. Our VRF construction is introduced and analyzed in Section 3. In Section 3.3, we present all the algorithms required to instantiate the protocols. In Section 4.1, we look at parameters, size and efficiency for the proposed VRF construction.

2 Background material

We call *negligible* a function $f : \mathbb{Z}_{>0} \to \mathbb{R}_{>0}$ if it is asymptotically dominated by $O(x^{-n})$ for all $n > 0$. When a quantity a depending on some parameter x is negligible we will sometimes write $a \leq \mathrm{negl}(x)$.

2.1 Verifiable Unpredictable and Random Functions

A Verifiable Function (VF) consists in the following algorithms:

- $\mathsf{ParamGen}(1^\lambda)$, returns a set of public parameters pp.
- $\mathsf{KeyGen}(pp)$, returns a pair (pk, sk) of public key and secret key from the public parameters.
- $\mathsf{VUFEval}(sk, x) = (v, \pi)$, takes the secret key sk and an input $x \in \{0,1\}^{n_1(\lambda)}$ and computes the output $v \in \{0,1\}^{n_2(\lambda)}$ along with a proof π.
- $\mathsf{Verify}(pk, \pi, x, v)$ takes the VRF public key, proof, input and output and returns 0 or 1.

There are two basic security properties that we want in a verifiable function.

- **Provability**: The verification always returns 1 on correctly generated proof and output from a given input (see Definition 2).
- **Uniqueness**: There does not exist a key and input and two pairs (v_1, π_1) and (v_2, π_2) with $v_1 \neq v_2$ both passing the verification (see Definition 5).

In cryptography, we are interested in Verifiable Functions with additional properties. In particular, in cases where it is hard to compute the output of the function without the secret key. There are two main flavours of this idea each leading to a different kind of verifiable function: VUFs and VRFs.

The weakest one is called **Unpredictability** and it requires an adversary with access to a evaluation oracle under a given public key to produce the output corresponding to an input that wasn't queried to the oracle. A VF satisfying Unpredictability is called a VUF (see Definition 3).

The stronger property is called **Pseudo-randomness**, and it requires that an adversary cannot distinguish between $\mathsf{VRFEval}(sk, x_0)$ and a random value even with access to an oracle computing $\mathsf{VRFEval}(sk, x)$ for $x \neq x_0$, (see Definition 6). A verifiable function satisfying pseudo-randomness is called a VRF.

It is well known that the two notions are equivalent in the ROM as one can obtain a VRF from a VUF by hashing the output of the VUF to get the output of the VRF. We will use this transform to obtain $\mathsf{DeuringVRF}$ from $\mathsf{DeuringVUF}$

Recently, in [24] it was shown that the VRF constructions obtained in this manner also satisfy an additional property called **unbiasability** which is important for practical applications.

2.2 Elliptic curves, quaternion algebras and the Deuring correspondence

Below, we briefly expose the useful features and definitions of the Deuring correspondence. For a more complete treatment of supersingular elliptic curves and quaternion algebras and their link through the Deuring correspondence see [27,30,34,40].

The Deuring correspondence is an equivalence of categories between isogenies of supersingular elliptic curves and the left ideals over maximal order $\mathcal{O}$ of $\mathcal{B}_{p,\infty}$, inducing a bijection between conjugacy classes of supersingular j-invariants and maximal orders (up to equivalence) [30]. Moreover, this bijection is explicitly constructed as $E \to \mathrm{End}(E)$. Hence, given a supersingular curve E_0 with endomorphism ring $\mathcal{O}_0$, the pair (E_1, φ), where E_1 is another supersingular elliptic curve and $\varphi : E_0 \to E_1$ is an isogeny, is sent to a left integral $\mathcal{O}_0$-ideal. The right order of this ideal is isomorphic to $\mathrm{End}(E_1)$. One way of realizing this correspondence is obtained through the kernel ideals defined in [41]. Given an integral left-$\mathcal{O}_0$-ideal I, we define the kernel of I as the subgroup $E_0[I] = \{P \in E_0(\overline{\mathbb{F}}_{p^2}) : \alpha(P) = 0 \text{ for all } \alpha \in I\}$. To I, we associate the isogeny $\varphi_I : E_0 \to E_0/E_0[I]$. Conversely, given an isogeny φ, the corresponding *kernel ideal* is $I_\varphi = \{\alpha \in \mathcal{O}_0 \ : \ \alpha(P) = 0 \text{ for all } P \in \ker(\varphi)\}$. Sometimes, when the kernel of φ is given as a group G generated by a point P, we also write I_G or I_P for this ideal. Two ideals I, J are said to be *equivalent* if $I = J\beta$ for some $\beta \in B_{p,\infty}^\times$ and we write $I \sim J$.

The main properties of the Deuring correspondence are summarized in Table 2.

Notation for duals and conjugates. Dual isogenies are denoted with a hat ($\varphi \to \hat{\varphi}$), and conjugates with an overline ($\gamma \to \overline{\gamma}$). As highlighted in Table 2, under the deuring correspondence, these two operations are equivalent.

On push-forward isogenies and ideals. Given two isogenies φ, ψ of coprime degree. We can define the push-forward of φ by ψ that we denote by $[\psi]_*\varphi$ as the isogeny of degree $\deg \varphi$ and kernel $\psi(\ker \varphi)$. The same can be done for the

Supersingular j-invariants over $\mathbb{F}_{p^2}$ $j(E)$ (up to Galois conjugacy)	Maximal orders in $B_{p,\infty}$ $\mathcal{O} \cong \text{End}(E)$ (up to isomorpshim)
(E_1, φ) with $\varphi : E \to E_1$	I_φ integral left $\mathcal{O}$-ideal and right $\mathcal{O}_1$-ideal
$\theta \in \text{End}(E_0)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$n(I_\varphi)$
$\hat{\varphi}$	$\overline{I_\varphi}$
$\varphi : E \to E_1, \psi : E \to E_1$	Equivalent Ideals $I_\varphi \sim I_\psi$
$\tau \circ \rho : E \to E_1 \to E_2$	$I_{\tau\circ\rho} = I_\rho \cdot I_\tau$

Table 2: The Deuring correspondence, a summary from [14].

push-forward of φ by ψ. This way, we get the following commutative diagram.

$$\begin{tikzcd}
E_3 \arrow[r, "{[\psi]_*\varphi}"] & E_4 \\
E_1 \arrow[u, "\psi"] \arrow[r, "\varphi"] & E_2 \arrow[u, "{[\varphi]_*\psi}"']
\end{tikzcd}$$

Under the Deuring corresponding we can define the push-forward of an ideal I by another ideal J of coprime norm as the ideal $[J]_*I$ corresponding to the push-forward isogeny $[\varphi_J]_*\varphi_I$. Formulas to compute the push-forward ideals are given in [14, Lemma 3].

In this work, we build upon several existing algorithms of the Deuring correspondence. We give precise references for all of them when they appear. Note that a description for all those algorithms can be found in [34, Chapters 3 and 4].

2.3 Isogeny Representation

The formal notion of *isogeny representation* is gaining more and more importance as the variety of existing method to build these representations is expanding. This definition appears at various places in the literature [33,34,11] with some small changes. The common and most important part is the existence of an algorithm to evaluate the isogeny from its representation. The representation is called *efficient* when the size of the representation and the complexity of the evaluation algorithm is polylogarithmic in the degree and field characteristic p.

Since, we are going to work with several family of representations, we will label each of those families with a tag xx. All the data and algorithms associated with the family xx will bear the same tag.

Definition 1. *An efficient isogeny representation* xx *for an isogeny* $\varphi : E \to E'$ *of degree* N *defined over* $\mathbb{F}_q$ *is denoted by* s^φ_{xx}*. It has size* $O\left(\text{polylog}(qN)\right)$*, and there exists the following algorithm:* $\mathsf{IsogEval}_{\mathsf{xx}}$ *that takes* $E, s^\varphi_{\mathsf{xx}}$ *and a point* P *in* $E[\mathbb{F}_{q^k}]$ *in input, and computes* $\varphi(P) \in E'[\mathbb{F}_{q^k}]$ *in time* $O\left(\text{polylog}(q^k N)\right)$.

On existing isogeny representations. There exists several isogeny representation in the literature. A non-exhaustive list of them can be found in [34, chapter 4]. In this work, we will use two of the representations presented there: the *kernel representation* based on the Vélu formulas (which is one of the "historical" isogeny representation) and the *ideal representation* based on the Deuring correspondence.

For the kernel representation, we use the tag ker. The representation $s^{\varphi}_{\mathsf{ker}}$ is made of a generator of $\ker\varphi$. This representation can be quite compact ($O\left(\text{polylog}(p)\right)$) when the kernel points are defined over a small field extension. However, the complexity of $\mathsf{IsogEval}_{\mathsf{ker}}$ is polynomial in the biggest prime factor of the degree which makes it efficient only for smooth degree isogenies. Hence, it does not meet our definition of *efficient* isogeny (but this gap is actually desirable for our construction). The kernel representation has another advantage : it is quite efficient to "sample" when the kernel points of order N are defined over a small extension. By efficient to sample, we mean that, for a given supersingular curve E, it is easy to compute the kernel representation of a random isogeny of degree N (we can even sample uniformly at random from the set of N-isogenies starting from E).

For the *ideal representation* we use the tag id. The representation $s^{\varphi}_{\mathsf{id}}$ is made of a basis of the ideal I_{φ} corresponding to φ under the Deuring correspondence. The ideal representation matches our definition of *efficient*, however it requires to know the endomorphism ring of the domain. When $\text{End}(E)$ is known, we can also efficiently sample ideal representations of uniformly random N-isogenies.

The recent attacks against the scheme SIDH [6,36,39] have introduced a new way to build an isogeny representation, as was noted by Robert in [38], by evaluating φ on a basis of the T-torsion for $T \geq \sqrt{N}$. The evaluation can be performed by computing an isogeny between abelian varieties of dimension $y > 1$ that embeds the isogeny φ. There are different variants of this idea for different values of y. In this work, we will look at the version for $y = 2$. We will call dimension-2 representation (with the tag $\mathsf{\mathit{2}dim}$), the isogeny representation obtained from this principle. In most of this work, we are going to use these representations in a black box manner. We refer the reader to [6,36,38,11,12] to see how to instantiate the required algorithms. We give a brief summary below.

Embedding isogenies in higher dimension isogenies with Kani's lemma. The goal of this paragraph is to explain how one can embed isogenies in higher dimension using Kani's lemma. This result introduced in [29] describe how to build isogenies of dimension $2y$ from isogenies in dimension y.

Lemma 1 (Kani). *Let us consider a commutative diagram of isogenies between principally polarized abelian varieties of dimension g*

$$\begin{tikzcd}
A' \arrow[r, "\varphi'"] & B' \\
A \arrow[u, "\psi"] \arrow[r, "\varphi"] & B \arrow[u, "\psi'"']
\end{tikzcd}$$

where φ and φ' are a-isogenies and ψ and ψ' are b-isogenies for two integers a, b. The isogeny $F : A \times B' \longrightarrow B \times A'$ given in matrix notation by

$$F := \begin{pmatrix} \varphi & \hat{\psi}' \\ -\psi & \hat{\varphi}' \end{pmatrix}$$

is a d-isogeny between abelian varieties of dimension $2g$ with $d = a + b$, for the product polarisations.

If a and b are coprime, the kernel of F is

$$\ker(F) = \{([\deg \varphi]x, \psi' \circ \varphi(x)) \mid x \in A[d]\}.$$

Remark 2. This lemma was first proven in [29, Theorem 2.3]. We are going to use it for $g = 1$ to obtain the *2*dim representation. The idea is that the isogeny F provides a representation for the isogeny $\varphi : A \to B$ since φ can be recovered as $\rho_2 \circ F \circ \rho_1$ where ρ_1 is any embedding morphism from A to $A \times B'$ and ρ_2 is the projection from $B \times A'$ to B.

More details on Kani's Lemma and the ways to compute efficiently isogenies in dimension 2 in the theta model can be found in [12]. Below, we explain more concretely how Kani's Lemma can be applied to get our isogeny representations for an isogeny $\varphi : E_1 \to E_2$.

The 2dim *isogeny representation.* In dimension $g = 1$, Lemma 1 can be applied to embed φ in an isogeny of dimension 2 with $A = E_1$ and $B = E_2$. The degree d is chosen to be 2^f for the smallest exponent f such that $2^f > N$. In that case, the isogeny ψ is any isogeny of degree $2^f - N$ coprime to N of domain E_1. This will define an isogeny diagram of the form

$$\begin{tikzcd}
E_4 \arrow[r, "\varphi'"] & E_3 \\
E_1 \arrow[u, "\psi"] \arrow[r, "\varphi"] & E_2 \arrow[u, "\psi'"']
\end{tikzcd}$$

Nothing is required of ψ other than having the correct domain and degree. We define $s^{\varphi}_{2\mathsf{dim}}$ as P, Q a basis of $E_1[2^f]$, the curve E_3 and the points $\psi' \circ \varphi(P), \psi' \circ \varphi(Q)$. Kani's Lemma tells us that this is enough information to compute the kernel of F, and F can be computed from its kernel. Note that despite our notation, there isn't a unique representation of any given φ (as it suffices to replace P, Q by another basis or replace ψ by an isogeny of the same domain and degree). However, all the representations are equivalent in the sense that we can evaluate φ from any $s^{\varphi}_{2\mathsf{dim}}$ with the same complexity. After the full isogeny F has been computed, the isogeny φ can be evaluated on any point as $\rho_2 \circ F \circ \rho_1$. This is how we get the algorithm $\mathsf{IsogEval}_{2\mathsf{dim}}$ We don't give a full description of this algorithm which can be based on the formulas from [12] and use it as a black-box in the rest of this work.

2.4 Ideal to isogeny translation from isogenies in dimension 2

Recently, Basso et al. introduced as [1, Algorithm 3] an efficient algorithm to evaluate any ideal I corresponding to an isogeny $\varphi_I : E_0 \to E$ on points of E_0.

This algorithm has no restriction on the degree of I, and requires a prime characteristic p of the special form $c2^f - 1$ where c is as small as possible to be efficient. This algorithm requires to computes three chain of $(2,2)$-isogenies of variable length.

In this work, we use this algorithm as a blackbox and denote it as $\mathsf{AnyIdealToIsogeny}$. It takes as input an $\mathcal{O}_0$-ideal and some points and outputs the codomain of the isogeny φ_I represented by I and the image of the torsion points given in input under φ_I.

To make that algorithm efficient, we will take a prime characteristic of the form $p = c2^f - 1$ although not exactly the one used in [1] (more details in Section 4.1).

3 New post-quantum VUF from isogenies

In this section, we provide a generic description of our $\mathsf{DeuringVUF}$ protocol. The algorithms are detailed in Section 3.1. The security of the scheme is analyzed in Section 3.2.

3.1 Formal description

In this section, we give a formal description of the different algorithms that constitutes our $\mathsf{DeuringVUF}$ protocol. We try to provide a high-level presentation to allow the reader to grasp the idea of the construction without wondering too much about the technical details. We postpone the detailed description of the most complicated building blocks to Sections 3.2 and 3.3. We also omit the discussion about parameters; it will be discussed later in Section 4.1.

Henceforth, let us assume that there are three distinct odd primes p, N, N_{sk}, and one exponent f such that all the 2^f and N torsion points of supersingular curves over $\mathbb{F}_{p^2}$ can be defined over $\mathbb{F}_{p^2}$ (possibly over different twists of the same curve), and $2^f > N$. N_{sk} is the degree of the secret key isogeny. Let us write $f' \leq f$ the smallest exponent f' such that $2^{f'} > N$. The $\mathit{2}\mathsf{dim}$-isogeny representation we are going to compute as proof of the evaluation will have degree $2^{f'}$.

There is also a curve E_0 over $\mathbb{F}_{p^2}$ of known endomorphism ring $\mathcal{O}_0$. The public parameters also include a basis (P_0, Q_0) of $E_0[N]$ and the related kernel ideal I_{P_0} together with an endomorphism $\kappa \in \mathcal{O}_0$ such that $\kappa(P_0) = Q_0$ and $\mathrm{tr}(\kappa)^2 - 4n(\kappa)$ is not a square mod N. We write $pp = (p, N, E_0, P_0, Q_0, I_{P_0}, \kappa)$.

The ideal I_{P_0} and the endomorphism κ allow us to compute efficiently the kernel ideal corresponding to subgroups of order N of the form $[x]P_0 + [y]Q_0$ for any x, y. If $\varphi_I : E_0 \to E$ is the secret isogeny corresponding to the secret key ideal I of norm N_{sk}, then the public key basis R, S of $E[N]$ is simply computed as $\varphi_I(P_0), \varphi_I(Q_0)$ rerandomized by a random matrix M. With this rerandomization, the distribution of R, S is independant of the secret, and the knowledge of the matrix M will be enough to reexpress any $[r]R+[s]S$ as $[x]\varphi_I(P_0)+[y]\varphi_I(Q_0)$ which will allow us to compute the ideal of kernel generated by $[r]R+[s]S$ (from the ideal of kernel generated by $[x]P_0 + [y]Q_0$).

We write $f_{\mathsf{in}} : \{0,1\}^{n_1(\lambda)} \to \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ an injective function where $n_1(\lambda)$ is a function of the security parameter λ.

The KeyGen, VUFEval, and Verify algorithms are described as Algorithms 1 to 3 respectively. The key generation algorithms makes use of a so-called deterministic basis of the public key curve E. By this, we mean a basis generated from the curve E by any deterministic algorithm. This is classical in isogeny-based cryptography. The only important part is that this basis can be recomputed from public information.

Algorithm 1 KeyGen(pp)

```
Input: Public parameters $pp$.
Output: A pair of DeuringVUF keys $sk, pk$.
Generate a random $\mathcal{O}_0$-ideal $I$ of norm $N_{\mathsf{sk}}$.
$E, P, Q, U', V' \leftarrow \mathsf{AnyIdealToIsogeny}(I, P_0, Q_0, U_0, V_0)$.
$\theta \leftarrow \varphi_I \circ \kappa_0 \circ \hat{\varphi}_I$, $Q \leftarrow [N_{\mathsf{sk}}]Q$.
Generate a random $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$.
Set $(R, S) \leftarrow ([a]P + [b]Q, [c]P + [d]Q)$.
$s \leftarrow N_{\mathsf{sk}}^{-1} \mod 2^f$
$U', V' \leftarrow [s]U', [s]V'$.
Compute a deterministic basis $U, V$ of $E[2^f]$.
Compute the matrix $M' = \left(\begin{smallmatrix} a' & b' \\ c' & d' \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ such that $U, V \leftarrow ([a']U' + [b']V', [c']U' + [d']V')$.
$U_0, V_0 \leftarrow ([a']U_0 + [b']V_0, [c']U_0 + [d']V_0)$.
return $(sk, pk) = ((E, I, M, U_0, V_0, U, V), (E, R, S))$.
```

As already mentioned in Section 2.3, the algorithm $\mathsf{IsogEval}_{\mathit{2}\mathsf{dim}}$ is used as blackbox in this article and we do not give any details about it. There are, however, several tasks that need more details in the algorithms we have outlined above. We list them below. These missing algorithms will be treated in Sections 3.2 and 3.3.

- The algorithm $\mathsf{IsogenyRepresentation}_{\mathit{2}\mathsf{dim}}$ to compute the $\mathit{2}\mathsf{dim}$ isogeny representation and the codomain.
- The $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}$ algorithm to check that the $\mathit{2}\mathsf{dim}$ representation is correct.

3.2 Security Analysis

In this section, we study the security properties of our VUF scheme.

Algorithm 2 $\mathsf{VUFEval}(sk, x)$

```
Input: A DeuringVUF secret key, and an input $x \in \{0,1\}^{n_1(\lambda)}$.
Output: A proof $\pi$ and the evaluation $v$ of the DeuringVUF function on input $x$.
Parse $sk$ as $E, I, M, U_0, V_0, U, V$.
Set $a, b, c, d$ such that $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$
$(r : s) \leftarrow f_{\mathsf{in}}(x)$.
$\gamma \leftarrow (ra + sc) + (rb + sd)\kappa$
Let $\alpha$ be such that $I_{P_0} = \mathcal{O}_0\langle \alpha, N\rangle$.
$I_x \leftarrow \mathcal{O}_0\langle \alpha\overline{\gamma}, N\rangle$.
$E_x, (E_3, U_3, V_3) \leftarrow \mathsf{IsogenyRepresentation}_{2\mathsf{dim}}(I, I_x, E, U, V, U_0, V_0)$.
$\pi \leftarrow E_3, U_3, V_3$.
$v \leftarrow j(E_x)$.
return $(\pi, v)$.
```

Algorithm 3 $\mathsf{Verify}(pk, \pi, x, v)$

```
Input: A DeuringVUF public key, an input $x \in \{0,1\}^{n_1(\lambda)}$, a proof $\pi$ and an output $v \in \{0,1\}^{n_2(\lambda)}$.
Output: A bit $b$.
Parse $pk$ as $E, R, S$.
if $R = 0_E$ or $S = 0_E$ or $[N]R \neq 0_E$ or $[N]S \neq 0_E$ then
    Return 0
end if
Parse $\pi$ as $E_3, U_3, V_3$ and parse $v$ as $j \in \mathbb{F}_{p^2}$.
Compute $U, V$ a canonical basis of $E[2^f]$.
$s^{\varphi}_{2\mathsf{dim}} \leftarrow E, [2^{f-f'}]U, [2^{f-f'}]V, E_3, [2^{f-f'}]U_3, [2^{f-f'}]V_3$
$(r : s) \leftarrow f_{\mathsf{in}}(x)$
$R_x \leftarrow [r]R + [s]S$
if $R_x = 0_E$ or $\mathsf{IsogVerif}_{2\mathsf{dim}}(s^{\varphi}_{2\mathsf{dim}}, j) = 0$ then
    return 0.
end if
if $r = 0$ then
    $P_{\varphi_x} \leftarrow \mathsf{IsogEval}_{2\mathsf{dim}}(s^{\varphi}_{2\mathsf{dim}}, R)$
else
    $P_{\varphi_x} \leftarrow \mathsf{IsogEval}_{2\mathsf{dim}}(s^{\varphi}_{2\mathsf{dim}}, S)$
end if
$Q_{\varphi_x} \leftarrow \mathsf{IsogEval}_{2\mathsf{dim}}(s^{\varphi}_{2\mathsf{dim}}, R_x)$
if $P_{\varphi_x} = 0_{E_x}$ or $Q_{\varphi_x} \neq 0_{E_x}$ then
    return 0.
end if
return 1.
```

Provability. The first and most basic notions a VUF must satisfy is provability. This notion implies that a correctly generated proof for a given input and key will pass the verification.

Definition 2. *A VUF is said to be provable if, for any $x \in \{0,1\}^{n_1(\lambda)}$, $(sk, pk) \leftarrow$ $\mathsf{KeyGen}(pp)$ and $(\pi, v) \leftarrow \mathsf{VUFEval}(sk, x)$, the following equality is satisfied:*

$$\mathsf{Verify}(pk, \pi, x, v) = 1$$

Using results later proven in this section, we are able to show that DeuringVUF is *provable.*

Proposition 1. DeuringVUF *is provable.*

Proof. In KeyGen, the correctness of AnyIdealToIsogeny, ensures that the isogeny φ_I on which the points are evaluated is the one corresponding to the ideal I. Thus, the curve E is the codomain of φ_I and we have $P, Q = \varphi_I(P_0, Q_0)$ and $U', V' = \varphi_I(U_0, V_0)$.

In VUFEval for an input x, we have $(r : s) = f_{\mathsf{in}}(x)$. Let us write $R_x = [r]R + [s]S$ where R, S are the points of the public key. By definition $(0 : 0)$ is excluded from the projective line and so R_x is a non-trivial point of order N. Then, since $\deg \varphi_x$ is coprime to N, the point $\hat{\varphi}_I(R_x)$ is also a non-zero point of order N. We need to justify that $I_{\hat{\varphi}_I(R_x)} = \mathcal{O}_0\langle \alpha\overline{\gamma}, N\rangle$. For that, it suffices to prove that $\ker \alpha\overline{\gamma} \cap E_0[N] = \langle \hat{\varphi}_I(R_x)\rangle$.

If α is such that $I_{P_0} = \mathcal{O}_0\langle \alpha, N\rangle$, then the principal ideal $\mathcal{O}_0\alpha$ correspond to an endomorphism of E_0 that we write α (by abuse of notation) such that $\alpha(P_0) = 0$ and the multiples of P_0 are the only points of order N in $\ker \alpha$.

After application of the matrix M, we have that $R_x = [ar+cs]P + [br+ds]Q$, thus $\langle \hat{\varphi}_I(R_x)\rangle = \langle [ar+cs]P_0 + [br+ds]Q_0\rangle$. Since κ is such that $Q_0 = \kappa(P_0)$, then the endomorphism $\gamma = (ar+cs) + (br+ds)\kappa$ sends $\langle P_0\rangle$ to $\langle \hat{\varphi}_I(R_x)\rangle$.

Now, note that $n(\gamma)$ is coprime to N. Indeed, the discriminant of the quadratic equation $n(t + z\kappa) = 0 \mod N$ is exactly $\mathrm{tr}(\kappa)^2 - 4n(\kappa)$ which is not a square by definition of κ, and so unless $t, z = 0, 0 \mod N$, there are no solution to this equation. This is not the case here for $ar+cs, br+ds$ as $r, s \neq 0, 0 \mod N$ and the matrix M was chosen to be invertible mod N.

This means (since we can invert $n[\gamma]$) that $\overline{\gamma}$ also sends $\langle \hat{\varphi}_I(R_x)\rangle$ to $\langle P_0\rangle$, and moreover that the preimage of $\langle P_0\rangle$ through $\overline{\gamma}$ is exactly $\langle \hat{\varphi}_I(R_x)\rangle$. Thus, it implies that $\hat{\varphi}_I(R_x)$ is exactly $\ker \alpha\overline{\gamma} \cap E_0[N]$ which proves that the computation of I_x is correct.

$\mathsf{IsogenyRepresentation}_{2\mathsf{dim}}$ will compute the representation for the isogeny $[\varphi_I]_*\varphi_{I_x}$ whose kernel is generated by R_x by what we just showed, and so the evaluation R_φ, S_φ of the points R, S in Verify will satisfy $[r]R_{\varphi_x} + [s]S_{\varphi_x} = 0_{E_x}$ if the *2*dim representation is valid.

This follows from the correctness of $\mathsf{IsogenyRepresentation}_{2\mathsf{dim}}$ and Lemma 1. We just need to show that the input I, I_x, U_0, V_0, U, V is well-formed. For that it suffices to prove that $U_0, V_0 = \hat{\varphi}_I(U, V)$ which is true since U', V' is set to be $[1/N_{\mathsf{sk}}]\varphi_I(U_0, V_0)$. Then, the matrix M' is computed to be the transition matrix from U', V' to U, V. Thus after application of M' on the basis U_0, V_0, we get that $U_0, V_0 = \varphi_I(\hat{U}, V)$.

Since $s^{\varphi}_{\mathsf{2dim}}$ is honestly computed as a valid representation of an isogeny of degree N, the output of $\mathsf{IsogVerif}_{\mathsf{2dim}}$ will be 1 by Proposition 3 and this proves the result. □

Unpredictability. This notion states that it is hard to evaluate the function without the secret key.

Definition 3. *Let $\mathcal{A}$ be an algorithm running in time t and playing the following experiment:*

1. $pp \leftarrow \mathsf{ParamGen}(1^{\lambda})$
2. $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$.
3. $(x^\star, y^\star, \pi^\star) \leftarrow \mathcal{A}^{\mathsf{VUFEval}(\cdot), H_{\mathsf{out}}(\cdot)}(pk)$.
4. *If $x^\star$ was previously queried, then $b \leftarrow 0$.*
5. *Else, $(y, \pi) \leftarrow \mathsf{VUFEval}(sk, x^\star)$, $b \leftarrow (y == y^\star)$.*

The Unpredictability advantage of $\mathcal{A}$ is defined as

$$\mathrm{Adv}^{\mathcal{A}}_{\mathrm{UP}}(t) = \Pr\{b = 1\} \tag{1}$$

The advantage of the protocol is defined as $\mathrm{Adv}_{\mathrm{UP}}(t) = \max_{\mathcal{A}} \mathrm{Adv}^{\mathcal{A}}_{\mathrm{UP}}(t)$

The VUF is unpredictable if

$$\mathrm{Adv}_{\mathrm{UP}}(t) \le \mathrm{negl}(\lambda)$$

when t is in $O(\mathrm{poly}(\lambda))$.

The Unpredictability property of our VUF is based on the hardness of Problem 1 that we introduce below. This problem is defined with respect to an arbitrary isogeny representation labelled with the tag xx. Even though, the security of our scheme only depends on the $\mathsf{2dim}$ representation, there is no harm in introducing the problem in a more general setting. Note that the resulting problem might not be as hard for all isogeny representations. To define Problem 1, we need to define an isogeny oracle in the fashion of the RADIO and RUGDIO introduced in [11]. We call this new oracle a N-FIXDIO$_{\mathsf{xx}}$.

Definition 4. *Given two odd prime $N \neq p$, a FIXed Degree N-Isogeny Oracle* (N-FIXDIO$_{)}$ *takes in input a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, and a point $P \in E[N]$ and outputs the j-invariant $j(E/\langle P\rangle)$ and an isogeny representation $s^{\varphi}_{\mathsf{xx}}$ for the N-isogeny $\varphi : E \to E/\langle P\rangle$.*

Problem 1. **One-More Isogeny Problem** (OMIP$_{\mathsf{xx}}$) Given two odd primes $N \neq p$, let E be a random supersingular elliptic curve over $\mathbb{F}_{p^2}$. Given access to the N-FIXDIO$_{\mathsf{xx}}$ on input E, the goal is to compute the j-invariant of the codomain of an isogeny not given as the output of the N-FIXDIO$_{\mathsf{xx}}$.

We define $\mathrm{Adv}_{\mathrm{OMIP}_{\mathsf{xx}}}(t) = \max_{\mathcal{A}} \Pr\{\mathcal{A}^{N\text{-FIXDIO}_{\mathsf{xx}}(\cdot)}() \text{ solves Pb. } 1\}$ for $\mathcal{A}$ ranging over all algorithms running in time t.

The hardness of Problem 1 underlies the Unpredictability of our VUF as we prove in Proposition 2.

Let us define $\mathcal{S}_p$ to be the set of isomorphism class of supersingular curves over $\mathbb{F}_{p^2}$, and let $\mathcal{D}_{N_{\mathsf{sec}}}$ be the distribution obtained by sampling a curve as the codomain of a uniformly random isogeny of degree N of domain E_0. Let $\Delta_{N_{\mathsf{sec}}}$ be the statistical distance between the uniform distribution on $\mathcal{S}_p$ (denoted by $\mathcal{US}_p$) and $\mathcal{D}_{N_{\mathsf{sec}}}$.

Proposition 2. *The scheme* DeuringVUF *satisfies*

$$\mathrm{Adv}_{\mathrm{UP}}(t) \leq (1 + 2\#\mathcal{S}_p \cdot \Delta_{N_{\mathsf{sec}}}) \cdot \mathrm{Adv}_{\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}}(t')$$

against any adversary $\mathcal{A}$ for some time t' polynomial in t.

Proof. Let us define $\mathrm{Adv}_{\mathrm{UP}^{\$}}(t)$ as the advantage against a modification of the experiment defined in Definition 3 where the distribution of pk is uniform in $\mathcal{S}_p$, and let us define $\mathbb{G}(E)$ the output of the experiment when $pk = E$.

$$\begin{aligned}
&|\mathrm{Adv}_{\mathrm{UP}^{\$}}(t) - \mathrm{Adv}_{\mathrm{UP}}(t)| \\
&\qquad \leq \sum_{E \in \mathcal{S}_p} \Pr\{\mathbb{G}(E) = 1 | pk = E\} \cdot |\Pr\{\mathcal{US}_p = E\} - \Pr\{\mathcal{D}_{N_{\mathsf{sec}}} = E\}| \\
&\qquad \leq \sum_{E \in \mathcal{S}_p} \Pr\{\mathbb{G}(E) = 1 | pk = E\} \cdot \sum_{E \in \mathcal{S}_p} |\Pr\{\mathcal{US}_p = E\} - \Pr\{\mathcal{D}_{N_{\mathsf{sec}}} = E\}| \\
&\qquad \leq 2\#\mathcal{S}_p \cdot \Delta_{N_{\mathsf{sec}}} \cdot \mathrm{Adv}_{\mathrm{UP}^{\$}}(t)
\end{aligned}$$

The proof is concluded by proving that $\mathrm{Adv}_{\mathrm{UP}^{\$}}(t) \leq \mathrm{Adv}_{\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}}(t)$. For that, we build an adversary $\mathcal{C}$ against the $\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}$ from an adversary $\mathcal{A}$ against the modified unpredictability experiment with uniform public key.

This adversary $\mathcal{C}$ is very simple. It sets the public key as the curve E given in the instance of the problem, answers to any evaluation query from $\mathcal{A}$ by using the $\mathit{2}\mathsf{dim}$-FIXDIO. In the end, it gets a value $y^\star$ from $\mathcal{A}$, and it outputs $y^\star$.

By definition the simulation of the unpredictability challenger is perfect, and it is clear that $\mathcal{C}$ solves the problem if and only if $\mathcal{A}$ wins the unpredictability game as f_{in} is injective. This proves the desired result.

By the Ramanujan property of the supersingular isogeny graph, if $\log(p) \approx 2\lambda$, one can show that by taking $N_{\mathsf{sk}} \approx p^2$, one gets $\Delta_{N_{\mathsf{sk}}} = \mathrm{negl}(\lambda)$.

Analysis of the $\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}$. The most obvious way to attack the $\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}$ is to try to compute directly any isogeny of domain E and degree N from its kernel. The best known method is the $\sqrt{\text{é}}\text{lu}$ algorithm from [3]. This algorithm takes $O(\sqrt{\max_{d|N} d})$ (ignoring logarithmic factors) operations over the field of definition of the kernel. Thus, even when $E[N]$ is defined over a small extension (which will be the case in our protocols), the complexity is exponential when N is a prime number. Another approach would be to try to compute the endomorphism ring $\mathrm{End}(E)$ (which would amount to key recovery in the context of our $\mathsf{Deuring\text{-}VUF}$ protocols). As our protocols can run in polynomial-time, the knowledge of the endomorphism ring is obviously enough to break the $\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}$. However, the complexity to compute the endomorphism ring of a random supersingular curve is $O(\sqrt{p})$ (see [18] for instance).

The two methods we described above are rather generic attacks that are not really using the fact that an access to the N-$\mathrm{FIXDIO}_{\mathit{2}\mathsf{dim}}$ is provided in the $\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}$. In particular, the attacker has access to several isogenies of degree N that he can evaluate. One might wonder if there could be a way to "tweak" one of the isogenies given by the N-$\mathrm{FIXDIO}_{\mathit{2}\mathsf{dim}}$ to obtain a new isogeny that would lead to a suitable solution to the $\mathrm{OMIP}_{\mathit{2}\mathsf{dim}}$. However, there does not seem to be an obvious way to do so. The only way to "tweak" an isogeny seems to be to apply some kind of push-forward and realize a commutative diagram where two parallel arrows are isogenies of degree N, one that is the output of the N-FIXDIO and the other one that would be the "tweaked" isogeny. There is nothing to prevent this from happening, however the tweaked isogeny will not have E as domain with overwhelming probability. The only possibility to have E as the domain of the "tweaked" isogeny would be that one of the perpendicular arrows of the commutative diagram is an endomorphism of E. Computing one endomorphism of a random supersingular curve also has complexity $O(\sqrt{p})$ and so this is not possible.

Finally, one might wonder if the access to the N-FIXDIO might help finding endomorphisms. It was argued in [11, Section 6.4] that the RADIO and RUGDIO oracles introduced there should not help to compute some endomorphisms of a given supersingular curve as we already know how to compute efficiently all isogenies of smooth degree. Given that our N-FIXDIO oracle is pretty similar to the RADIO and RUDGIO, the same reasoning applies in our case to justify that the N-FIXDIO should not be of any help.

The $\mathit{2}\mathsf{dim}$ representation reveals more information than isogenies of degree N, as each $\mathit{2}\mathsf{dim}$ representation also embeds an isogeny of degree $2^f - N$. This isogeny can also be chosen to be uniform among all isogenies of the same domain and degree, and so for the same reason that revealing isogenies of degree N is not problematic, this additional information does not seem to help breaking the problem either.

Uniqueness. A VUF scheme satisfy unconditional full uniqueness when there cannot be two possible output for the same input. This is formalized in Definition 5 below.

Definition 5. *A VUF is said to satisfy unconditional full uniqueness when no values pk, v, v', x, π, π' can satisfy* $\mathsf{Verify}(pk, \pi, x, v) = 1$ *and* $\mathsf{Verify}(pk, \pi', x, v') = 1$ *with $v \neq v'$.*

To prove the uniqueness of our scheme, we need to give more details about the verification procedure. In particular, we need to details the $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}$ algorithms.

Verification in dimension 2. The verification in dimension 2 is pretty simple: we need to verify that the provided isogeny representation is well-formed. This means verifying that we can compute F, an isogeny of dimension 2 that represents an isogeny between E and a curve of the correct j-invariant. This part of the verification is handled by the $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}$ algorithm. For uniqueness, we also need to verify that the degree and kernel are correct. These two properties will be verified during the check that $\varphi(R_x) = 0$ performed in Verify (the full justification can be found in the proof of Proposition 4).

Algorithm 4 $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}(E_1, s^{\varphi}_{\mathit{2}\mathsf{dim}}, j)$

Input: A curve E_1, a $\mathit{2}\mathsf{dim}$ representation $s^{\varphi}_{\mathit{2}\mathsf{dim}}$, and a j-invariant j.
Output: A bit b.

```
Parse $s^{\varphi}_{\mathit{2}\mathsf{dim}}$ as $E_1, P_1, Q_1, E_3, P_3, Q_3$.
$G \leftarrow \langle ([N]P_1, P_3), ([N]Q_1, Q_3) \rangle$
if $G$ is not a kernel of a $2^{f'}$-isogeny of dimension 2 then
    return 0.
end if
Compute $F : E_1 \times E_3 \to E_2 \times E_4$ of kernel $G$.
if the computation of $F$ fails in any way then
    return 0.
end if
if $j(E_2) \neq j$ then
    return 0.
else
    return 1.
end if
```

Note that, in the statement below, when the output $\mathsf{IsogVerif}$ is 1, there is no guarantee that the degree of the isogeny represented is N exactly simply that the degree is smaller than 2^f.

Proposition 3. *If* $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}(s^{\varphi}_{\mathit{2}\mathsf{dim}}, j) = 1$*, then* $s^{\varphi}_{\mathit{2}\mathsf{dim}}$ *constitutes a valid* $\mathit{2}\mathsf{dim}$ *isogeny representation for an isogeny* $\varphi : E_1 \to E_2$ *of degree smaller than* $2^{f'}$ *where* $j(E_2) = j$*.*

Conversely, if $s^{\varphi}_{\mathit{2}\mathsf{dim}}$ *is a valid* $\mathit{2}\mathsf{dim}$ *isogeny representation for an isogeny of degree* N *from* E_1 *to* E_2*, then* $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}(E_1, s^{\varphi}_{\mathit{2}\mathsf{dim}}, j(E_2)) = 1$*.*

Proof. When $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}(E_1, s^{\varphi}_{\mathit{2}\mathsf{dim}}, j) = 1$, there exists a $2^{f'}$-isogeny $F =: E_1 \times E_3 \to E_2 \times E_4$ with $j(E_2) = j$. Kani's Lemma imply that we have a valid $\mathit{2}\mathsf{dim}$ representation for the isogeny $\rho_2 \circ F \circ \rho_1 : E_1 \to E_2$ where ρ_1, ρ_2 are defined as in Remark 2, and that the degree of this isogeny must be smaller than $2^{f'}$.

Conversely, when $s_{\mathsf{2dim}}^{\varphi}$ is a valid representation for an isogeny φ of degree N, then by definition, we must have $P_3, Q_3 = \psi' \circ \varphi(P_1, Q_1)$. In this case, the subgroup G agrees exactly with the subgroup defined in Lemma 1. Thus, G is a correct kernel of a dimension 2 isogeny whose codomain is isomorphic to $E_2 \times E_4$. Thus, the output of $\mathsf{IsogVerif}_{\mathsf{2dim}}$ is 1. □

Proposition 4. *The scheme* DeuringVUF *satisfies unconditional full uniqueness.*

Proof. Let us assume that we have a value $v = j$ passing the verification for an input x, a public key pk and proof π. We want to prove that the only possibility is that $j(E') = j(E/\langle R_x \rangle)$.

By Proposition 3, we know that a valid isogeny representation for an isogeny $\varphi_x : E \to E'$ with $j(E') = j$ can be extracted from pk and π if $\mathsf{IsogVerif}_{\mathsf{2dim}}(s_{\mathsf{2dim}}^{\varphi}, j) \neq 0$.

Then, in Verify we check that R, S both have order N and are non-zero, and that R_x is non-zero. Then, in checking that $P_{\varphi_x} \neq 0$, we check that R, S are a basis of $E[N]$. Indeed, if $r = 0$, then R_x and R are colinear, and if not, then S and R_x must be colinear. So if R, S were not a basis (and so were colinear), R_x would be colinear with them too (since R_x is non-zero), and so P_{φ_x} would be also 0. Since it is not, then R, S must be a basis of $E[N]$. Thus, R_x have full order N. And since $\varphi_x(R_x) = 0_{E_x}$, we know that $\langle R_x \rangle \subset \ker \varphi_x$. This implies that N divides the degree of φ. But since φ has degree smaller than $2^{f'}$ and we defined f' to be the smallest exponent such that $2^{f'} > N$, we get that $\deg \varphi$ must be N, and so $\ker \varphi = \langle R_x \rangle$ and so E' must be isomorphic to $E/\langle R_x \rangle$ which proves the result. □

3.3 Isogeny representation computation in dimension 2.

It remains to introduce the algorithm $\mathsf{IsogenyRepresentation}_{\mathsf{2dim}}$. As for KeyGen, the main building block of this algorithm is the AnyIdealToIsogeny from [1].

Algorithm 5 $\mathsf{IsogenyRepresentation}_{\mathsf{2dim}}(I, J, E, P, Q, \hat{\varphi}_I(P), \hat{\varphi}_I(Q))$

Input: I an $\mathcal{O}_0$-ideal of norm N_{sk}, J an $\mathcal{O}_0$-ideal of norm N, a curve E such that $\mathrm{End}(E) \cong \mathcal{O}_R(I)$, a basis U, V of $E[2^f]$ and the image $U_0, V_0 = \hat{\varphi}_I(U, V)$.

Output: E_x,E_3, U_3, V_3 where $\varphi : E \to E_x$ is the pushforward isogeny $[\varphi_I]_* \varphi_J$ of degree N and $U_3, V_3 = \psi \circ \varphi(U, V)$ for some isogeny $\psi : E_x \to E_3$ of degree $2^{f'} - N$.

```
Compute $K$ an $O_0$-ideal of norm $2^{f'} - N$.
$L \leftarrow I \cap J \cap K$.
$E_3, U_3, V_3 \leftarrow \mathsf{AnyIdealToIsogeny}(L, U_0, V_0)$.
$s \leftarrow N_{\mathsf{sk}}^{-1} \mod 2^{f'}$.
$U_3, V_3 \leftarrow [s]U_3, [s]V_3$.
Compute the codomain of $F : E \times E_3 \to E_x \times F_x$ of kernel $[2^{f-f'}]\langle([N]U, U_3), ([N]V, V_3)\rangle$.
return $E_x, (E_3, U_3, V_3)$.
```

The correctness of this algorithm follow from Lemma 1. Indeed, for the output of IsogenyRepresentation to be a correct, we need to verify that F the points $([N_{\mathsf{sk}}]U, U_3), ([N_{\mathsf{sk}}]V, V_3)$ correctly generate the kernel of the isogeny F embedding φ_J.

The ideal L correspond to some composition of isogenies $\psi \circ [\varphi_I]_* \varphi_J$ where $\psi : E_x \to E_3$ is an isogeny of degree $(2^{f'} - N)$ and E_3 is some supersingular curve. By definition of AnyIdealToIsogeny, and $U_0, V_0 = \hat{\varphi_I}(u, V)$, the points U_3, V_3 are thus equal to $[N_{\mathsf{sk}}]\psi \circ [\varphi_I]_* \varphi_J(U, V)$. This proves that the computation of E_3, U_3, V_3 is correct.

Hence, after multiplication by $[s]$, by Lemma 1 the points $([N_{\mathsf{sk}}]U, U_3), ([N_{\mathsf{sk}}]V, V_3)$ correctly generate the kernel of a $2^{f'}$-isogeny of dimension 2 embedding $[\varphi_I]_* \varphi_J$ and so the computation of E_x is correct.

4 Parameters and Performances

In this section, we discuss the choice of parameters to instantiate our Deuring-VUF family as efficiently as possible at a given level of security λ. We propose a concrete set of parameters for $\lambda = 128$ presumably corresponding to the NIST-I level of security.

4.1 Parameter computation.

The main parameter we need to choose is the value of p. After that is done, all the other parameters can be deduced almost directly. Before explaining how to find this prime concretely, let us give a brief reminder on the various constraints and requirements.

A summary of the constraints for security. The generic key recovery attack has complexity $\tilde{O}(\sqrt{p})$. Thus, we need to take $\log(p) \approx 2\lambda$. Similarly, the best known algorithm to compute N-isogenies has complexity $\tilde{O}(\sqrt{N})$. Thus, we need to target $\log(N) \approx 2\lambda$. The $\tilde{O}$ notation hiding polynomial factors in $\log \log p$, we can afford to be a bit below the 2λ threshold for both $\log p$ and $\log N$ without damaging the security.

A summary of the constraints for efficiency. We need to take a prime p of the form $c2^f - 1$ with c as small as possible and such that $N \mid (p-1)/2$ to ensure that the N torsion of supersingular curve can be defined over $\mathbb{F}_p^2$. The complexity of finding such a p is logarithmic. Indeed, the primality of p and of the near primality of $(p-1)/2$ both have a probability of $\approx 1/\log p$ to happen. Thus, it suffices to try a logarithmic number of c, f until a suitable solution is found.

Computation of remaining public parameters. Now that we have specified the choices of all the integral parameters, we need to explain how to compute the remaining public parameters of our scheme. In particular, we need to find a basis P_0, Q_0 of $E_0[N]$, an endomorphism κ such that $\kappa(P_0) = Q_0$ and the kernel ideal I_{P_0}. This operation is not completely trivial as N is a large prime number, but it can be done as we explain next. Let us take as E_0 one of the curves of known endomorphism ring $\mathcal{O}_0$ (for instance the curve of j-invariant 1728 which is supersingular since we have $p = 3 \mod 4$). Our solution uses the fact that we can evaluate efficiently any endomorphisms of E_0 on points of E_0 using the endomorphisms $\pi : (x, y) \to (x^p, y^p)$ and $\iota(x, y) \to (-x, \sqrt{-1}y)$.

First, we can select R_0 as any point of order N. Then, we compute $\alpha \in \mathrm{End}(E_0)$ of norm satisfying $\gcd(n(\alpha), N^2) = N$. Then, we can set $P_0 = \alpha(R_0)$. If $P_0 = 0$ we can try with another R_0 until we have P_0 of order N. When P_0 has order N, the ideal I_{P_0} is equal to $\mathcal{O}_0\langle\overline{\alpha}, N\rangle$. To finish the precomputation, we can take any endomorphism κ of norm coprime to N that is not contained in $\mathbb{Z} + I_{P_0}$ and we set $Q_0 = \kappa(P_0)$. This ensures that P_0, Q_0 is a basis of $E_0[N]$ as if $\kappa(P_0) = \mu P_0$ for some integer μ, then $\kappa - \mu \in I_{P_0}$ and so κ would be contained in $\mathbb{Z} + I_{P_0}$.

4.2 Example parameters for $\lambda = 128$

We now describe concrete parameters for $\lambda = 128$ that are estimated to reach the NIST-I security level. The prime $p = c2^f - 1$ was found after exhausting a small set of values c, f selected to ensure $p < 2^{256}$.

We found the 249-bit prime p

$$p + 1 = 305 \cdot 2^{240}$$

for which the 239-bit prime $N = (p - 1)/1182$ is equal to:

455912313669549255941544617365981512121305066452676634709889925037435729.

An example of endomorphism α to compute I_{P_0} is the endomorphism

$$597586948685359749757390890506743252 + 314328096788165707865159343442888735\pi$$

of norm N and κ can be taken as ι.

4.3 Sizes

In this section, we explain how to compute the sizes given in Table 1 for the parameters given in Section 4.2. We also provide abstract formulas that are true for any security level λ assuming that $\log p = 2\lambda$.

On compression. To reduce the size of the public key and proof, we can use standard compression techniques for elliptic curves and their points.

- Curves can be represented by their j-invariants which are always defined over $\mathbb{F}_{p^2}$ for supersingular curves.

- A basis of the 2^f-torsion can be compressed as two $\mathbb{F}_{p^2}$ elements and 1 bit corresponding to the x-coordinates of the two points of the basis, and one bit indicating the relative sign of these points.
- A point of arbitrary order N defined over $\mathbb{F}_{p^2}$ can always be represented as one element over $\mathbb{F}_{p^2}$ and a bit (representing the x-coordinates of the point and the parity of the y-coordinate).

Computation of key and proof sizes. Using the compressed representations that we described above for all the points and curves involved in our construction, we can deduce the size of keys and proofs of our DeuringVUF protocol. The public key is made of 1 curve, and 2 points of order N. This can be represented by three elements over $\mathbb{F}_{p^2}$ and 2 bits so $12\lambda + 2$ bits. The proof is made of 1 curve, and a basis of the 2^f torsion. This can be represented by 3 elements over $\mathbb{F}_{p^2}$ and 1-bit, so this is $12\lambda + 1$.

Since we need the function f_{in} to be injective to avoid collisions, the input space cannot be bigger than $N + 1$. Thus, we can take the input space has size $n_1(\lambda) = \lfloor \log_2 N \rfloor$.

The output space is one element over $\mathbb{F}_{p^2}$ so this takes 4λ.

These results are summarized in Table 3.

y	Input (bits)	Output (bits)	Public Key (bits)	Proof (bits)
2	2λ	4λ	$12\lambda + 3$	$12\lambda + 1$

Table 3: Size of the inputs, outputs, keys and proofs of the DeuringVUF schemes.

Note that the basis of $E[2^f]$ could further be reduced to 3 scalars modulo 2^f at the cost of an additional pairing computation. However, the sizes being already very compact, it does not seem that the additional computational cost would really be worth it.

4.4 Implementation

We made a C implementation of our scheme basing ourselves on the code from [1] for the algorithms AnyIdealToIsogeny and $\mathsf{IsogEval}_{2\mathsf{dim}}$. The need to handle points of order N on the quadratic twists of supersingular curves required a little bit of adaptation, but apart from that, all the main building blocks were already there. The performance of our implementation are reported in Table 4. Below, we analyze a little bit those results.

The prime characteristic being different, we couldn't use the optimized finite field arithmetic developed by the authors of [1], and so we used a generic finite field arithmetic from fiat-crypto instead. Our results should thus be compared with the performances of [1, Table 3] and not [1, Table 4]. With an optimized finite field arithmetic, we're expecting an improvement factor similar to the one between Table 3 and Table 4 in [1]. In particular, the verification time of DeuringVUF should be within a factor two of SQIsign2D-west's verification.

We also expect an optimized implementation of the evaluation of DeuringVUF to be at least as fast as SQIsign2D-west signature. However, our current implementation of the evaluation is much slower than that. There are two reasons for that, and both are related to the performance of the AnyIdealToIsogenyalgorithm, and in particular the search for the two integers u, v (see [1, Section 4.2] for more details). The first one should be easily addressed whereas it is less obvious if anything can be done for the second one.

The first explanation is that this step requires to reduce the ideal given in input which can be done with the LLL algorithm. The current implementation of LLL from [1] seems to be very slow when the ideal given in input is quite big. In our scheme, it is much bigger than in SQIsign2D-west, and the LLL execution accounts for almost a third of the total running in our current implementation. This operation could definitely be optimized with a proper LLL implementation and we ultimately expect the running time of this step to be negligible compared to the rest of the computation.

The second reason comes from the heuristics detailed in [1, Section 4.2]. This heuristic tells us that the running time of the search procedure for u and v is proportionnal to the ratio $p/2^f$. For the SQIsign2D-west prime at NIST level 1, this ratio is 5, whereas it is 305 in DeuringVUF.

Metric	KeyGen	VUFEval	Verify
Clockcycles (10^6 cycles)	256	371	40
Time (ms)	111	161	17.4

Table 4: Average running time of the C DeuringVUF implementation on an intel core i7 at 2.3GHz with turbo-boost disabled, adapted from the code of [1]

5 Interesting Applications

In this section, we detail two interesting applications of our VUF. The first, and most interesting one is a VRF scheme, the second one is a hash-and-sign style signature scheme. Both schemes are derived almost directly from the VUF.

5.1 Verifiable Random Function

As explained in Section 2.1, one can derive almost directly a VRF from a VUF by adding one extra step in the evaluation algorithm: define the VRF output as the hash of the VUF output with the public key and the input. Then, the proof must also include the VUF output. The whole transform is described precisely in [24, Figure 9].

With this small modification, in the random oracle model, it is possible to extend the unpredictability property of the VUF (which reduces to a computational problem) to the pseudo-randomness required by a VRF (which reduces to a decisional problem). In [24, Theorem 1], Giunta and Stewart prove that the resulting protocol is a VRF satisfying the additional property of unbiasability.

The VRF obtained by applying this transform to our scheme DeuringVUF is called DeuringVRF. The security of this scheme follows in the ROM from Propositions 1 and 2 and [24, Theorem 1]. The public key is the same as DeuringVUF. The proof size is slightly bigger than the one in DeuringVUF as one needs to include the VUF output which adds 4λ bits to the proof. This is how we got the sizes showcased in Table 1. In terms of performances, the overhead of the transform is completely negligible.

Below, we recall the formal definition of pseudo-randomness of a VRF.

Pseudo-randomness. This security notion implies that it is hard to distinguish the output from a random value without knowing the secret key.

Definition 6. *Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an algorithm running in time t and playing the following experiment:*

1. $pp \leftarrow \mathsf{ParamGen}(1^\lambda)$
2. $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$.
3. $(x^\star, st_1) \leftarrow \mathcal{A}_1^{\mathsf{VUFEval}(\cdot), H_{\mathsf{out}}(\cdot)}(pk)$.
4. $(v_0, \pi_0) \leftarrow \mathsf{VUFEval}(sk, x^\star)$.
5. $v_1 \overset{\$}{\leftarrow} \{0,1\}^{n_2(\lambda)}$.
6. $b \overset{\$}{\leftarrow} \{0,1\}$.
7. $b' \leftarrow \mathcal{A}_2^{\mathsf{VUFEval}(\cdot), H_{\mathsf{out}}(\cdot)}(v_b, st)$.

where the query of VUFEval *on $x^\star$ are implicitly forbidden. The pseudo-randomness advantage of $\mathcal{A}$ is defined as*

$$\mathrm{Adv}_{\mathrm{PR}}^{\mathcal{A}}(t) = \Pr\{b = b'\} \tag{2}$$

The advantage of the protocol is defined as $\mathrm{Adv}_{\mathrm{PR}}(t) = \max_{\mathcal{A}} \mathrm{Adv}_{\mathrm{PR}}^{\mathcal{A}}(t)$

The VRF is pseudo-random if

$$\mathrm{Adv}_{\mathrm{PR}}(t) \leq 1/2 + \mathrm{negl}(\lambda)$$

when t is in $O(\mathrm{poly}(\lambda))$.

5.2 Hash-and-sign Signature

A similarly easy transform allow us to derive a signature from a VUF. For that it suffices to have a function to hash the message to the input space of the VUF, and then output the result of the VUF evaluation as the signature. The unforgeability of the resulting signature protocol holds under the unpredictability of the VUF and the collision resistance of the hash function.

The sizes and performances of our VUF (see in Tables 1 and 4), and the discussion in Section 4.4 show that the resulting signature is not much worse (in both size and speed) than the recent SQIsign2D-west signature in all metric.

While the resulting signature in itself does not improve upon the state of the art in isogeny-based signatures (it is less compact, and less efficient than SQIsign's most recent variants), we hope that the radically different way it was obtained (when compared to SQIsign) could still lead to interesting applications. Also, note that the security of this new scheme holds in the standard model and does not require the ROM, which is also a first in isogeny-based cryptography.

6 Prospects, open questions and future work

We have introduced a new DeuringVUF protocol based on isogenies between supersingular curves by making use of two important sub-fields of isogeny-based cryptography: the Deuring correspondence and isogenies between abelian varieties of high dimension. The security of our new problem stands upon a new security assumption, the OMIP. Despite its novelty, this new assumption is related to various well-studied problem in isogeny-based cryptography, and its hardness appears quite plausible. Interestingly, progress in the resolution of this problem could have very positive impacts on the efficiency of other areas of isogeny-based cryptography. Nonetheless, the OMIP, requires more study, in particular in the quantum setting.

We derive two interesting applications from this DeuringVUF. The main one is a VRF that is efficient and more compact (by a good margin) than every other post-quantum VRF protocol, thus proving that VRFs could be one of the most promising case of application of isogeny-based cryptography. The second contribution is the first hash-and-sign signature based on isogenies.

References

1. Basso, A., Dartois, P., de Feo, L., Leroux, A., Maino, L., Pope, G., Robert, D., Wesolowski, B.: SQIsign2D-West the fast, the small, and the safer (2024)
2. Basso, A., Maino, L., Pope, G.: Festa: Fast encryption from supersingular torsion attacks. Cryptology ePrint Archive (2023)
3. Bernstein, D.J., Feo, L.D., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. ANTS (2020)
4. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 416–432. Springer (2003)
5. Buser, M., Dowsley, R., Esgin, M.F., Kasra Kermanshahi, S., Kuchta, V., Liu, J.K., Phan, R.C.W., Zhang, Z.: Post-quantum verifiable random function from symmetric primitives in pos blockchain. In: European Symposium on Research in Computer Security. pp. 25–45. Springer (2022)
6. Castryck, W., Decru, T.: An efficient key recovery attack on sidh. EUROCRYPT (2023)

7. Chen, J., Gorbunov, S., Micali, S., Vlachos, G.: Algorand agreement: Super fast and partition resilient byzantine agreement. Cryptology ePrint Archive, Report 2018/377 (2018), https://eprint.iacr.org/2018/377
8. Chen, M., Leroux, A., Panny, L.: Scallop-hd: group action from 2-dimensional isogenies. In: IACR International Conference on Public-Key Cryptography. pp. 190–216. Springer (2024)
9. Chen, M., Leroux, A., Panny, L.: Scallop-hd: group action from 2-dimensional isogenies. In: IACR International Conference on Public-Key Cryptography. pp. 190–216. Springer (2024)
10. Cornacchia, G.: Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$. Giornale di Matematiche di Battaglini **46**, 33–90 (1908)
11. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQISignHD: New dimensions in cryptography. EUROCRYPT (2024)
12. 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 (2024)
13. David, B., Gaži, P., Kiayias, A., Russell, A.: Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 66–98. Springer (2018)
14. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: Sqisign: compact post-quantum signatures from quaternions and isogenies. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 64–93. Springer (2020)
15. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the deuring correspondence: towards practical and secure sqisign signatures. In: Advances in Cryptology–EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V. pp. 659–690. Springer (2023)
16. Decru, T., Maino, L., Sanso, A.: Towards a quantum-resistant weak verifiable delay function. In: International Conference on Cryptology and Information Security in Latin America. pp. 149–168. Springer (2023)
17. Eisenträger, K., Hallgren, S., Lauter, K., Morrison, T., Petit, C.: Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 329–368. Springer International Publishing (2018)
18. Eisenträger, K., Hallgren, S., Leonardi, C., Morrison, T., Park, J.: Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs. Open Book Series **4**(1), 215–232 (2020)
19. Elkies, N.D.: The existence of infinitely many supersingular primes for every elliptic curve over $\mathbb{Q}$. Inventiones mathematicae **89**(3), 561–567 (1987)
20. Esgin, M.F., Kuchta, V., Sakzad, A., Steinfeld, R., Zhang, Z., Sun, S., Chu, S.: Practical post-quantum few-time verifiable random function with applications to algorand. In: International Conference on Financial Cryptography and Data Security. pp. 560–578. Springer (2021)
21. Esgin, M.F., Steinfeld, R., Liu, D., Ruj, S.: Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs. In: Annual International Cryptology Conference. pp. 484–517. Springer (2023)
22. Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. In: ASIACRYPT (2017)

23. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles. pp. 51–68 (2017)
24. Giunta, E., Stewart, A.: Unbiasable verifiable random functions. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 142–167. Springer (2024)
25. Goldberg, S., Naor, M., Papadopoulos, D., Reyzin, L., Vasant, S., Ziv, A.: Nsec5: provably preventing dnssec zone enumeration. Cryptology ePrint Archive (2014)
26. Goyal, R., Hohenberger, S., Koppula, V., Waters, B.: A generic approach to constructing and proving verifiable random functions. In: Theory of Cryptography Conference. pp. 537–566. Springer (2017)
27. H. Silverman, J.: The Arithmetic of Elliptic Curves, vol. 106 (01 2009)
28. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) Post-Quantum Cryptography. pp. 19–34. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
29. Kani, E.: The number of curves of genus two with elliptic differentials. Journal für die reine und angewandte Mathematik **1997**(485), 93–122 (1997). https://doi.org/10.1515/crll.1997.485.93
30. Kohel, D.: Endomorphism rings of elliptic curves over finite fields. Ph.D. thesis, University of California at Berkeley (1996)
31. Kohel, D., Lauter, K.E., Petit, C., Tignol, J.P.: On the quaternion ℓ-isogeny path problem. IACR Cryptology ePrint Archive **2014**, 505 (2014)
32. Lai, Y.F.: Capybara and tsubaki: Verifiable random functions from group actions and isogenies. Cryptology ePrint Archive (2023)
33. Leroux, A.: A new isogeny representation and applications to cryptography. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part II. pp. 3–35. Springer (2022)
34. Leroux, A.: Quaternion Algebra and isogeny-based cryptography. Ph.D. thesis, Ecole doctorale de l'Institut Polytechnique de Paris (2022)
35. Love, J., Boneh, D.: Supersingular curves with small noninteger endomorphisms. Open Book Series **4**(1), 7–22 (2020)
36. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on sidh. In: EUROCRYPT (2023)
37. Micali, S., Rabin, M., Vadhan, S.: Verifiable random functions. In: 40th annual symposium on foundations of computer science (cat. No. 99CB37039). pp. 120–130. IEEE (1999)
38. Robert, D.: Evaluating isogenies in polylogarithmic time. Cryptology ePrint Archive (2022)
39. Robert, D.: Breaking SIDH in polynomial time. EUROCRYPT (2023)
40. Voight, J.: Quaternion Algebras. Springer Graduate Texts in Mathematics series (2018)
41. Waterhouse, W.C.: Abelian varieties over finite fields. Annales Scientifiques de l'E.N.S (1969)
42. Yamada, S.: Asymptotically compact adaptively secure lattice ibes and verifiable random functions via generalized partitioning techniques. In: Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part III 37. pp. 161–193. Springer (2017)

43. Yang, R., Au, M.H., Zhang, Z., Xu, Q., Yu, Z., Whyte, W.: Efficient lattice-based zero-knowledge arguments with standard soundness: construction and applications. In: Annual International Cryptology Conference. pp. 147–175. Springer (2019)

A Some new algorithms for the Deuring correspondence

A.1 Translating ideals of norm a big prime power of a small power.

The goal of this section is to instantiate an algorithm $\mathsf{IdealToIsogeny}^2_{\ell^\bullet}$ to translate an ideal of norm a power of ℓ into their corresponding isogeny for some small prime ℓ using isogenies of dimension 2 as a tool.

For the rest of this section, we fix an exponent e such that the ℓ^e torsion of supersingular curves can be defined over $\mathbb{F}_{p^2}$. For simplicity, we will target the case where the norm of the input to $\mathsf{IdealToIsogeny}^2_{\ell^\bullet}$ is exactly ℓ^{ne} for some integer n. The generic case can be derived trivially from there.

The outline of this algorithm $\mathsf{IdealToIsogeny}^2_{\ell^\bullet}$ is inspired by the $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ algorithm introduced as [15, Algorithm 5] in the context of the SQIsign signature scheme. When the input has norm ℓ^{ne}, the algorithm $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ consists in n sequential executions of a sub-algorithm $\mathsf{IdealToIsogenyEichler}_{\ell^e}$ ([15, Algorithm 4]) that performs the translation for inputs of norm ℓ^e exactly.

We will keep the same structure for our algorithm $\mathsf{IdealToIsogeny}^2_{\ell^\bullet}$, and we introduce an algorithm $\mathsf{IdealToIsogeny}^2_{\ell^e}$ to replace $\mathsf{IdealToIsogenyEichler}_{\ell^e}$.

We start with a brief summary of the ideas underlying $\mathsf{IdealToIsogenyEichler}_{\ell^e}$ to provide some insights on how and why its dimension 2 counterpart was designed.

Translating ideal to isogenies with isogenies in dimension 1, a summary. The main subtlety in $\mathsf{IdealToIsogenyEichler}_{\ell^e}$ is that each translation of length e "consumes" the ℓ^e torsion points (those points are necessary to express the kernel of the ℓ^e-isogenies to be translated). This is why the algorithm $\mathsf{IdealToIsogeny}$ $\mathsf{Eichler}_{\ell^e}$ performs a "refresh" operation, necessary to all its subsequent executions inside $\mathsf{IdealToIsogeny}^1_{\ell^\bullet}$. In $\mathsf{IdealToIsogenyEichler}_{\ell^e}$, this refresh is done by evaluating some well-chosen endomorphism θ of the domain curve on the ℓ^e torsion. In dimension 1, there is only one way to ensure that this endomorphism θ can be efficiently evaluated: ensure that $\deg\theta | T^2$ where T is a smooth integer such that the T-torsion points are defined over a small field extension. Endomorphisms satisfying these constraintes can be found using the $\mathsf{SpecialEichlerNorm}_T$ algorithm [15, Algorithm 3]. But this algorithm only succeeds when the value of T is quite big ($T \approx p^{5/4}$).

This constraint on the size of T is the main cause of the relative inefficiency of $\mathsf{IdealToIsogeny}^1_{\ell^\bullet}$, because having the T-torsion defined over a small field extension of $\mathbb{F}_{p^2}$ implies a very strong constraint on the two integers p and T. A suitable solution can be always be found, but the smoothness bound of T might not be very small. This smoothness bound in turn impacts the cost of the T-isogenies that must be computed in order to evaluate the endomorphism θ (the smoother T, the faster the computation will be).

This limitation is the main motivation to introduce a variant using isogenies of dimenion 2. The goal of this algorithm is to overcome the obstacle of dimension 1 by exploiting the $\mathsf{2dim}$ isogeny representation.

Translating ideal to isogenies with isogenies in dimension 2, an overview. Our goal is to simplify the computation of the endomorpism θ. To overcome the obstacles encountered with the dimension 1 algorithm, we follow a reasoning that resembles the idea behind the recent SQIsignHD scheme [11]: by embedding θ in an 2^f-isogeny of higher dimension (for some exponent f), we can relax most of the constraints on its degree (and in particular the smoothness). This means that we can get rid of $\mathsf{SpecialEichlerNorm}_T$ and simply look for θ among the endomorphism of small norm in $\mathrm{End}(E)$. The concrete requirements for $\mathsf{IdealToIsogeny}^2_{\ell^e}$ are in fact slightly more complex than that. In the next paragraph, we introduce an algorithm $\mathsf{RandomGoodEndomorphism}$ to find suitable endomorphisms.

In the remaining of this section, we assume $\ell \neq 2$, and we fix and exponent f such that the 2^f torsion of supersingular curves is defined over $\mathbb{F}_{p^2}$.

Finding suitable endomorphisms for the dimension 2 representation. As explained in Section 2.3, the $\mathsf{2dim}$ representation for any isogeny φ of degree a requires a second isogeny β of degree b such that $2^f = a + b$. In our case, the isogeny we want to represent is an endomorphism θ. Following an idea introduced in [9], we propose to choose β as an endomorphism of E as well. With the method described in [9] (see the algorithm $\mathsf{OrientDiamondDim2}$), it is possible to find efficiently two endomorphisms θ, β in the same quadratic order satisfying the norm equation $2^f = n(\theta) + n(\beta)$. Since θ, β are each defined by two coefficients, this is quaternary quadratic equation that can be solved using Cornacchia's algorithm [10] when 2^f is big enough compared to n. In Appendix B, we introduce an algorithm $\mathsf{ExtendedOrientDiamondDim2}$ (which is a simple variant of [9, Algorithm 1]) to handle that task.

As in $\mathsf{SpecialEichlerNorm}$, the endomorphism θ computed by $\mathsf{RandomGoodEndomorphism}$ must satisfy an additional constraint: it can not be contained in the Eichler order $\mathbb{Z} + K$ for some ideal K of norm ℓ given in input. This additional constraint is quite strong and it implies that $\mathsf{RandomGoodEndomorphism}$ will always fail for some maximal order $\mathcal{O}$. Fortunately, it can be shown heuristically that this will only happen with very small probability when we consider a random supersingular curve, and this will be enough for our need. We provide some more insights on these potential failures later in this section.

Before proving the correctness and termination of our algorithm, we need to state a preliminary result. The following lemma, adapted from a result first mentioned by Elkies in [19] tells us that we can always find a non-trivial endomorphism ω of norm smaller than $p^{2/3}$. We have also included a result by Boneh and Love regarding the number of curves having an endomorphism of norm smaller than some bound [35, Proposition A.3].

Algorithm 6 RandomGoodEndomorphism$(\mathcal{O}, K, f)$

```
Input: A maximal order $\mathcal{O}$, an $\mathcal{O}$-ideal $K$ of norm $\ell$, and an integer $f$.
Output: $\theta, \beta \in \mathcal{O} \smallsetminus (2\mathcal{O} \cup \mathbb{Z} + K)$ and $2^f = n(\theta) + n(\beta)$, $\gcd(n(\theta), n(\beta)) = 1$.
Sample random element $\omega$ of norm smaller than $2^f$ in $\mathcal{O}^\perp$ until $\omega$ is not contained
in $\mathbb{Z} + K$ and $D_\omega = \mathrm{tr}(\omega)^2 - 4n(\omega)$ is equal to $5 \mod 8$.
if no such elements $\omega$ can be found then
    Return $\perp$
end if
if ExtendedOrientDiamondDim2$(D_\omega, f) \neq \perp$ then
    $\theta, \beta =$ ExtendedOrientDiamondDim2$(D, f)$.
    Return $\theta, \beta$.
end if
return Return $\perp$.
```

Lemma 2. *For any supersingular curve E, there exists a non-trivial endomorphism $\omega : E \to E$ of degree smaller than $2p^{2/3}$.*

Moreover, given any $B < p^{2/3}$, the number of curves having a non-trivial endomorphism of norm smaller than B is in $O(B^{3/2})$.

Proof. Let $\mathcal{O}$ be isomorphic to $\mathrm{End}(E)$. Let us consider the rank 3 lattice $\mathcal{O}/\mathbb{Z}$. Since the reduced discriminant of $\mathcal{O}$ is p, the determinant of $\mathcal{O}/\mathbb{Z}$ is p^2 and, by Minkowski's theorem, it must contain an element θ of norm smaller than $2p^{2/3}$. The second result regaring the number of curves having a non-trivial endomorphism smaller than B was proven as [35, Proposition A.3]. □

The correctness and complexity of ExtendedOrientDiamondDim2 is stated in Proposition 8 of Appendix B. This proposition holds under Heuristic 1 which is a plausible heuristic assumption (adapted from [9, Heuristic 13]) regarding the distribution of numbers of the form $2^e + D(1+z^2)$ where D is the discriminant of a quadratic imaginary order equal to 1 mod 4. Thus, Proposition 5 holds under the same heuristic assumption.

Proposition 5. *Assuming Heuristic 1, for any $\kappa > 0$, there exists $\eta = \Theta(\log\log(p) + \kappa)$ such that, if $f > 2/3\log(p) + \eta$, then* RandomGoodEndomorphism *will succeed with probability bigger than $1 - 2^{-\kappa}$ on input $\mathcal{O}, K, f$ if the maximal order $\mathcal{O}$ is a uniformly random maximal order in $B_{p,\infty}$, and K is a random $\mathcal{O}$-ideal of norm ℓ.*

Proof. For the algorithm to fail, either no suitable ω was found or ExtendedOrientDiamondDim2 failed on input D_ω, f.

Let us write $\alpha_1, \alpha_2, \alpha_3$ for the three successive minimas of $\mathcal{O}^\perp$. First, note that if $n(\alpha_3) < 2^h$, then there exists an endomorphism of norm smaller than 2^h that is not contained in $\mathbb{Z} + K$. Indeed, if $\alpha_1, \alpha_2, \alpha_3$ we cannot have all three elements $\alpha_1, \alpha_2, \alpha_3$ contained in $\mathbb{Z} + K$ as $\mathcal{O}/\ell\mathcal{O} \cong M_2(\mathbb{Z}/\ell\mathbb{Z})$ and so $\mathcal{O}^\perp/\ell\mathcal{O}^\perp \cong M_2(\mathbb{Z}/\ell\mathbb{Z})/\mathbb{Z}$. Let us now take the smallest $\omega \in \mathcal{O}^\perp$ such that $n(\omega) < A$ and $\omega \notin \mathbb{Z} + K$ and $D_\omega = 5 \mod 8$.

Then, by Proposition 8, the success probability of ExtendedOrientDiamondDim2 on input D_ω, f is upper-bounded by

$$(1 - \lambda_0/log(|D_\omega|))^{\sqrt{2^f/|D_\omega|}}.$$

For D_ω and f big enough, this function will clearly increase as D_ω increases. Thus, for any $\mathcal{O}$ such that $n(\omega) < A$ (which implies $D_\omega < 4A$) we can upper-bound the failure probability of RandomGoodEndomorphism on input $\mathcal{O}$ by

$$(1 - \lambda_0/\log(4A))^{\sqrt{2^{f+2}/A}}.$$

If we write $p(A)$, the probability that a random $\mathcal{O}$ in $B_{p,\infty}$ is such that $n(\omega) < A$, then the conditional probability formula associated to a trivial majoration of any probability by 1 gives us that the probability of failure of RandomGoodEndomorphism on a random input $\mathcal{O}$ is upper-bounded by

$$p(A) + (1 - \lambda_0/\log(4A))^{\sqrt{2^{f+2}/A}}$$

We can now use Lemma 2 to upper-bound $p(A)$.

By Minkowski's second theorem, we know that $n(\alpha_1)n(\alpha_2)n(\alpha_3) \leq \mu_0 p^2$ for some constant μ_0. Thus, if $n(\alpha_3) > A$, then $n(\alpha_1) \leq \mu_1 p/\sqrt{A}$ for some constant μ_1. By Lemma 2, this implies that the number of curves with $n(\alpha_3) < A$ is smaller than $\mu_2 p^{3/2}/A^{3/4}$ for some constant μ_2.

It is sufficient that a constant number of (linearly-independent) endomorphisms in $\mathcal{O}^\perp$ are smaller than A, to get an element ω that is not in $\mathbb{Z} + K$ and of discriminant equal to $5 \mod 8$. Since $\alpha_1, \alpha_2, \alpha_3$ constitutes a basis of elements of norm smaller than A, we obtain enough elements to find ω with constant probability and so we get $p(A) \leq \mu_3\sqrt{p}/A^{3/4}$ for some constant μ_3.

We derive the following upper-bound on the failure probability:

$$\mu_3 \frac{\sqrt{p}}{A^{3/4}} + \left(1 - \frac{\lambda_0}{\log(4A)}\right)^{\sqrt{2^f/A}} \tag{3}$$

Now, it is easily verified that for any $\kappa > 0$, there exists $\eta = \Theta(\kappa + \log\log(p))$ such that if $f > 2/3\log(p) + \eta$, then there exists $a = 2/3\log(p) + \Theta(\kappa)$ smaller than f such that the upper-bound of the failure probability given in Equation 3 is smaller than $2^{-\kappa}$ when $A = 2^a$. □

Potential failures of RandomGoodEndomorphism. We can extract easily the cases where RandomGoodEndomorphism will potentially fail: when the smallest endomorphism of $\mathcal{O}^\perp$ is smaller than usual so all the endomorphisms of norm smaller than 2^f all lie in the same quadratic order that is either either contained in $\mathbb{Z} + K$ or does not have a discriminant equal to $5 \mod 8$. Depending on the value of p and f, this might happen for some maximal orders and values of K. In those cases, RandomGoodEndomorphism will simply fail. The results of Proposition 5 allow us to adjust the value of f to reduce the probability of this bad event as much as possible. Moreover, note that it can be shown that the third successive minima must be in $\Theta(p)$. Thus, when $f > \log(p)$ the proportion of failing maximal order will decrease very quickly to 0. In all of our applications of $\mathsf{RandomGoodEndomorphism}$, it should not be too hard to rerandomize the choice of maximal order, thus these failures should not really be problematic as soon as we are careful to pick a value of f that is not too close to $2/3\log(p)$. We made some experiments by sampling random maximal orders as right orders of $\mathcal{O}_0$-ideals of norm 2^{256} (for a prime $p \approx 2^{256}$) and did not find a single example where the algorithm failed out of thousands of trials.

The full ideal-to-isogeny subroutine in dimension 2. As we explained above, we obtain $\mathsf{IdealToIsogeny}^2_{\ell^e}$ by adapting $\mathsf{IdealToIsogenyEichler}_{\ell^e}$ to use $\mathsf{RandomGoodEndomorphism}$ instead of $\mathsf{SpecialEichlerNorm}$ and compute θ from a $\mathit{2}\mathsf{dim}$ isogeny representation rather than as a T-isogeny. This yields Algorithm 7. We remind the reader that we use in a black-box manner an algorithm $\mathsf{IsogEval}_{\mathit{2}\mathsf{dim}}$ to evaluate isogenies from their $\mathit{2}\mathsf{dim}$-representation.

The algorithm $\mathsf{IdealToIsogeny}^2_{\ell^e}$ also assumes the knowledge of a curve E_0 with its endomorphism ring $\mathrm{End}(E_0)$. For this curve, it is known how to evaluate any endomorphism $\theta_0 \in \mathrm{End}(E_0)$.

Algorithm 7 $\mathsf{IdealToIsogeny}^2_{\ell^e}(\mathcal{O}, I, J, \varphi_J, P)$

```
Input: $I$ a left $\mathcal{O}$-ideal of norm $\ell^e$, an $(\mathcal{O}_0, \mathcal{O})$-ideal $J$ of norm in $\ell^\bullet$ and $\varphi_J : E_0 \to E$
    the corresponding isogeny, the generator $P \in E[\ell^e]$ of $\ker\varphi_K$ s.t $\hat{\varphi}_J = \varphi_{K'} \circ \varphi_K$.
Output: $\varphi_I$ of degree $\ell^e$
Set $K = \overline{J} + \mathcal{O}\ell$.
if $\mathsf{RandomGoodEndomorphism}(\mathcal{O}, K, 2f) = \perp$ then
    Return $\perp$.
end if
Compute $\theta, \beta = \mathsf{RandomGoodEndomorphism}(\mathcal{O}, K, h)$.
Select $\alpha \in I$ s.t. $I = \mathcal{O}\langle \alpha, \ell^e \rangle$.
Compute $C, D$ s.t. $\alpha \cdot (C + D\theta) \in K$ and $\gcd(C, D, \ell) = 1$ using linear algebra.
Compute $R, S$ a basis of $E[2^f]$, $t = \deg\varphi_J^{-1} \mod 2^f$.
Compute $W, X = \hat{\varphi_J}(R, S)$.
Set $\theta_0 = \hat{\varphi}_J \circ \theta \circ \varphi_J \in \mathrm{End}(E_0)$ and compute $U, V = \theta_0(W, X)$.
Set $\beta_0 = \hat{\varphi}_J \circ \beta \circ \varphi_J \in \mathrm{End}(E_0)$ and compute $W, X = \beta_0(W, X)$ and $Y, Z =$
    $\beta_0(U, V)$.
Compute $U, V = [t^2]\varphi_J(U, V)$,$W, X = [t^2]\varphi_J(W, X)$,and $Y, Z = [t^3]\varphi_J(Y, Z)$.
Set $s^\theta_{\mathit{2}\mathsf{dim}} = E, E, U, V, Y, Z$, and $c^{n(\theta)}_{\mathit{2}\mathsf{dim}}(E) = E, R, S, W, X$.
Compute $Q = \mathsf{IsogEval}_{\mathit{2}\mathsf{dim}}(s^\theta_{\mathit{2}\mathsf{dim}}, c^{n(\theta)}_{\mathit{2}\mathsf{dim}}(E), P)$
Compute $\varphi_I$ of kernel $\langle [C]P + [D]Q \rangle$.
return $\varphi_I$.
```

Proposition 6. *Let $\mathcal{O}, I, J, \varphi_J, P$ be the input to* $\mathsf{IdealToIsogeny}^2_{\ell^e}$ *and let $K = \overline{J} + \mathcal{O}\ell$. If* $\mathsf{RandomGoodEndomorphism}(\mathcal{O}, K, h) \neq \perp$*, then* $\mathsf{IdealToIsogeny}^2_{\ell^e}$ *returns the correct output on input $\mathcal{O}, I, K, \varphi_J, P$.*

Proof. By [15, Lemma 8], if the point Q is equal to $\theta(P)$, then the group $\langle [C]P + [D]Q \rangle$ is the kernel of the desired isogeny φ_I. Thus, for our purpose, it suffices to show that Q is indeed equal to $\theta(P)$. By the presumed correctness of $\mathsf{IsogEval}_{\mathit{2}\mathsf{dim}}$, we need to show that the isogeny representation $c^{n(\theta)}_{\mathit{2}\mathsf{dim}}(E)$, $s^{\theta}_{\mathit{2}\mathsf{dim}}$ is correct. First, note that since θ, β are commutative endomorphisms, the commutative diagram they generate only involvde the curve E and we have $\theta' = \theta$ and $\beta' = \beta$. Second, we need to verify that $U, V = \theta(R, S)$, $W, X = \beta(R, S)$ and $Y, Z = \beta \circ \theta(R, S)$. Following the various computations, we see that $U, V = [t^2]\varphi_J \circ \theta_0 \circ \hat{\varphi}_J(R, S)$. By definition of θ_0 this is $[t^2][\deg \varphi_J^2]\theta(R, S) = \theta(R, S)$. The same can be shown for W, X, and for Y, Z with $\beta_0 \circ \theta_0 = [\deg \varphi_J]\hat{\varphi}_J \circ \beta \circ \theta \circ \varphi_J$. This defines a correct $\mathit{2}\mathsf{dim}$ isogeny representation according to the formulas given in Section 2.3 and this concludes the proof. □

The generic ideal-to-isogeny algorithm. We are now ready to introduce the full algorithm $\mathsf{IdealToIsogeny}^2_{\ell^\bullet}$ algorithm. As we said at the beginning of this section, this is a simple generalization of [15, Algorithm 5]. We refer the reader to [15] for the proof of correctness.

Algorithm 8 $\mathsf{IdealToIsogeny}^2_{\ell^\bullet}(I, J, \varphi_J)$

```
Input: $I$ a left $\mathcal{O}$-ideal of norm $\ell^{ne}$, an $(\mathcal{O}_0, \mathcal{O})$-ideal $J$ of norm $\ell^\bullet$ and $\varphi_J : E_0 \to E$
    the corresponding isogeny
Output: $\varphi_I$ of degree $\ell^{ne}$.
Set $J_i = J$, $I_i = I + \ell^f \mathcal{O}$, $I_i' = I_i^{-1} I$, $\mathcal{O}_i = \mathcal{O}$.
Set $\varphi_i$ of degree $\ell^f$ as the isogeny such that $\hat{\varphi}_J = \varphi' \circ \varphi_i$
Set $\varphi_I = [1]_E$ and $E_i = E$.
for $i \in [1, n]$ do
    Compute $P_i \in E_i[\ell^f]$ s.t $\ker \varphi_i = \langle P_i \rangle$.
    Compute $\varphi_{I_i} = \mathsf{IdealToIsogeny}^2_{\ell^e}(\mathcal{O}_i, I_i, J_i, \varphi_I \circ \varphi_J, P_i)$.
    Set $\varphi_i = \hat{\varphi}_{I_i}$, $\varphi_I = \varphi_{I_i} \circ \varphi_I$ and $E_i$ is the codomain of $\varphi_{I_i}$.
    Set $J_i = J_i \cdot I_i$, $\mathcal{O}_i = \mathcal{O}_L(I_i')$, $I_i = I_i' + \ell^f \mathcal{O}_i$ and $I_i' = I_i^{-1} I_i'$.
end for
return $\varphi_I$.
```

A.2 Evaluating isogenies from the ideal representation when the order is not coprime with the degree

In this section, we introduce an algorithms $\mathsf{IsogEvalNonCoprime}^1_{\mathsf{id}}$ to evaluate isogenies of dimension 1 from their ideal representation when the norm is a power of ℓ and the order of the points to be evaluated is not coprime to ℓ.

This algorithm is based on the following idea: given an isogeny $\psi : E_0 \to E$ of arbitrary degree coprime to ℓ, another isogeny $\varphi_J : E_0 \to E$ of degree $\ell^\bullet$, an endomorphism θ of E of norm coprime to ℓ can be used to compute the image of any subgroup $G \subset E_0[G]$ efficiently, assuming a few conditions on θ.

Our algorithm $\mathsf{IsogEvalNonCoprime}^1_{\mathsf{id}}$ is obtained by combining this idea with the algorithm introduced in the proof of Lemma 3 below to obtain the images of an isogeny on given points of smooth order from the images of this isogeny on subgroups of the same order.

Lemma 3. *Let N be an integer coprime to some small prime ℓ. Let E_0, E be two elliptic curves connected by an isogeny $\psi : E_0 \to E$. Assume that $E[2^f]$ is defined over $\mathbb{F}_{p^2}$. There is an algorithm of complexity $O\left(\text{poly}(\log(p) + f)\right)$ that takes $G_1, G_2, G_3, H_1, H_2, H_3$ where G_1, G_2, G_3 are three subgroups of order 2^f such that $G_i \cap G_j = \{0\}$ for all $1 \leq i < j \leq 3$ and $H_i = \psi(G_i)$ for $i = 1, 2, 3$, a point $P \in E_0[2^f]$ and computes $\psi(P)$ up to sign.*

Proof. Let P_i, Q_i be the respective generators of G_i, H_i for $i = 1, 2, 3$. We know there exists λ_i such that $\varphi(P_i) = [\lambda_i]Q_i$. By assumption that $G_1 \cap G_2 = \{0\}$, the two points Q_1, Q_2 form a basis of $E[2^f]$. By solving a bidimensional discrete logarithm in $E[2^f]$, one can find μ_1, μ_2 such that $Q_3 = [\mu_1]Q_1 + [\mu_2]Q_2$. Doing the same on E_0, we obtain $P_3 = [\nu_1]P_1 + [\nu_2]P_2$. Then, we get that $\lambda_3/\lambda_i = \nu_i/\mu_i$ for $i = 1, 2$ ($\mu_i \neq 0$ since $H_3 \cap H_1 = \{0\}, H_3 \cap H_2 = \{0\}$ because ψ has degree coprime to ℓ). Thus, the three values $R_i = \lambda_3 \psi(P_i)$ can be computed for $i = 1, 2, 3$.

Then, computing the discrete logarithm of $e(P_1, P_2)$ and $e(R_1, R_2)$, we get the scalar $\lambda_3^2 N \mod 2^f$. Dividing by N and computing a squareroot s of the result mod 2^f, we get $S_i = s^{-1} R_i = \pm \psi(P_i)$. Then, to evaluate any point $P \in E_0[2^f]$ is suffices to find a, b such that $P = aP_1 + bP_2$ and to output $aS_1 + bS_2$.

It is clear that all the operations above can be performed in $O\left(\text{poly}(\log(p) + f)\right)$. □

The algorithm $\mathsf{IsogEvalNonCoprime}^1_{\mathsf{id}}$ uses several building blocks of the Deuring correspondence based on isogenies in dimension 1. There is the $\mathsf{SpecialEichlerNorm}_{T^2}$ algorithm (see [15, Algorithm 3]) to compute endomorphisms of norm dividing T^2 in any maximal order barred of an Eichler order of level ℓ, and the $\mathsf{IdealToKernel}_D$ and $\mathsf{IdealToIsogeny}_{\mathsf{D}}$ algorithms to translate an ideal of norm D in the corresponding kernel and isogeny respectively (see [34, section 4.2.1]). The integer T is an implicit parameter of Algorithm 9.

Proposition 7. *(Heuristic)* $\mathsf{IsogEvalNonCoprime}^{id}_1$ *is correct and terminates with constant probability when $T > p^{5/4}$.*

Proof. The heuristics involved in this proof are the same than in the proofs of correctness and termination of $\mathsf{KLPT}_{2^\bullet}$, $\mathsf{IdealToKernel}_{2^f}$ $\mathsf{IdealToIsogeny}^1_{\ell^\bullet}$ and $\mathsf{SpecialEichlerNorm}_T$ (see [34] and [15] for statements and proofs regarding the termination and correctness of these algorithms).

The termination of $\mathsf{IsogEvalNonCoprime}_1$ follows from the termination of all the building blocks. The condition on T and the constant success probability both come from [15, Proposition 6].

Let us now prove correctness. After the ideal J and the isogeny φ_J have been computed together with the point Q. The steps 5 to 15 are essentially the same that are performed in $\mathsf{IdealToIsogenyEichler}_{\ell^e}$. We refer the reader to the proof of

Algorithm 9 $\mathsf{IsogEvalNonCoprime}^1_{\mathsf{id}}(I, P)$

```
Input: $I$ a left $\mathcal{O}_0$-ideal of norm $N$ coprime to 2, a point $P \in E_0[2^f]$.
Output: $\pm\varphi_I(P)$
 1: Set $\mathcal{O} = \mathcal{O}_R(I)$.
 2: Compute $J = \mathsf{KLPT}_{2^\bullet}(I)$.
 3: Compute $\varphi_J = \mathsf{IdealToIsogeny}^1_{2^\bullet}(\mathcal{O}_0, 1, J)$.
 4: Compute $Q$ the kernel of the dual of the last isogeny composing $\varphi_J$.
 5: Set $K = \overline{J} + \mathcal{O}2$.
 6: Compute $\theta = \mathsf{SpecialEichlerNorm}_T(\mathcal{O}, K)$ of norm dividing $T^2$.
 7: Take any $n_1|T$ and $n_2|T$ s.t. $n_1 n_2 = n(\theta)$. Compute $H_1 = \mathcal{O}\langle\theta, n_1\rangle$ and $H_2 = \mathcal{O}\langle\overline{\theta}, n_2\rangle$.
 8: Compute $L_j = [J]^* H_j$, and $\varphi_j = [\varphi_J]_* \mathsf{IdealToIsogeny}_{n_j}(L_j)$ for $j \in \{1, 2\}$.
 9: Compute $Q' = \hat{\varphi}_2 \circ \varphi_1(Q)$.
10: Compute $I_1, I_2, I_3$ three $\mathcal{O}_0$-ideals of norm $2^f$ such that $I_i + \mathcal{O}_0 2 \neq I_j + \mathcal{O}_0 2$ for $1 \leq i < j \leq 3$.
11: for $i = 1, 2, 3$ do
12:     Compute $\langle P_i \rangle = \mathsf{IdealToKernel}_{2^f}(I_i)$.
13:     Compute $\alpha_i$ such that $[I]_* I_i = \mathcal{O}_0\langle\alpha_i, 2^f\rangle$.
14:     Compute $C_i, D_i$ s.t. $\alpha_i \cdot (C_i + D_i\theta) \in K$ and $\gcd(C_i, D_i, 2) = 1$ using linear algebra.
15:     Compute the kernel $Q_i = [C_i]Q + [D_i]Q'$.
16: end for
17: Compute $\mu_1, \mu_2$ such that $Q_3 = [\mu_1]Q_1 + [\mu_2]Q_2$.
18: Compute $\nu_1, \nu_2$ such that $P_3 = [\nu_1]P_1 + [\nu_2]P_2$.
19: Set $R_3 = Q_3$ and $R_i = (\mu_i/\nu_i \mod 2^f)Q_i$ for $i = 1, 2$.
20: Compute $\lambda$ such that $e_{2^f}(P_1, P_2)^\lambda = e_{2^f}(R_1, R_2)$.
21: Compute $s = \sqrt{\lambda/N}^{-1} \mod 2^f$.
22: Compute $S_i = [s]R_i$ for $i = 1, 2, 3$.
23: Compute $a, b$ such that $P = [a]P_1 + [b]P_2$.
24: return $[a]S_1 + [b]S_2$.
```

[15, Proposition 6] for a proof that the subgroup $\langle Q_i \rangle$ is the kernel of the ideal $[I]_* I_i$ for $i = 1, 2, 3$. This means that $\langle Q_i \rangle = \varphi_I(\langle P_i \rangle)$ for $i = 1, 2, 3$.

Steps 17 to 22 correspond to the algorithm described in the proof of Lemma 3. We refer the reader to this proof to show that $S_1, S_2, S_3 = \pm\varphi_I(P_1, P_2, P_3)$. Then, since $P = [a]P_1 + [b]P_2$, we get $[a]S_1 + [b]S_2 = \pm\varphi_I(P)$.

□

B On the ExtendedOrientDiamondDim2 algorithm.

In this section, we give some details on the ExtendedOrientDiamondDim2 algorithm and provide some statement on this algorithm. We remind the reader that this algorithm is almost identical to the OrientDiamondDim2 algorithm ([9, Algorithm 1]). The only real difference being that the exponent h is given in input.

Algorithm 10 ExtendedOrientDiamondDim2$(D_{\mathfrak{O}}, h)$

Input: An imaginary quadratic order $\mathfrak{O}$ with discriminant $D_{\mathfrak{O}} = 5 \mod 8$.
Output: $\theta, \beta \in \mathfrak{O}$ such that $n(\theta) + n(\beta)$ is a 2-power and $\gcd(n(\theta), n(\beta)) = 1$.

```
Set $x := 0$, $y := 0$.
for $z \in \times[1, \lfloor\sqrt{\frac{2^{h+1}}{|D_{\mathfrak{O}}|} - 1}\rfloor]$ do
    $M := 2^{h+2} + D_{\mathfrak{O}}(z^2 + 1)$.
    if $M$ is a prime such that $M \equiv 1 \bmod 4$ or $M = 2M'$ where $M'$ is a prime such that $M' \equiv 1 \bmod 4$ then
        Use Cornacchia's algorithm to find $X, Y$ such that $X^2 + Y^2 = M$.
        Set $x = (X - D_{\mathfrak{O}})/2$ and $y = (Y - zD_{\mathfrak{O}})/2$.
        break
    end if
end for
if $x = 0$ and $y = 0$ then
    Return $\perp$.
end if
$\theta := x + \frac{D_{\mathfrak{O}} + \sqrt{D_{\mathfrak{O}}}}{2}$, $\beta := y + z\frac{D_{\mathfrak{O}} + \sqrt{D_{\mathfrak{O}}}}{2}$.
return $\theta, \beta$.
```

We reuse the heuristic introduced in [9, Heuristic 13] as Heuristic 1 below. Assuming that heuristic, and following the proof provided of [9, Proposition 14], we can prove Proposition 8.

Heuristic 1. *Let $h, D_{\mathfrak{O}} = 5 \mod 8$ be the inputs to Algorithm 10. If z are sampled as random integers then the integers $2^{e+2} + D_{\mathfrak{O}}(1 + z^2)$ behave like random odd integers of the same size that are either congruent to 1 modulo 4 or equal to 2 times an integer that is congruent to 1 modulo 4.*

Proposition 8. *Assuming Heuristic 1,* ExtendedOrientDiamondDim2 *is correct, and there exists a constant λ_1 such that the computation will succeed with probability at least:*

$$1 - (1 - \lambda_0 / \log(|D_{\mathfrak{O}}|))^{\sqrt{2^h/|D_{\mathfrak{O}}|}}.$$